

Справочник JavaScript API Яндекс.Карт

Версия 2.1.79

8.07.2021



Справочник JavaScript API Яндекс.Карт. Версия 2.1.79. Версия

Дата подготовки документа: 8.07.2021

Этот документ является составной частью технической документации Яндекса.

© 2008—2021 ООО «ЯНДЕКС». Все права защищены.

Предупреждение об исключительных правах и конфиденциальной информации

Исключительные права на все результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации юридических лиц, товаров, работ, услуг и предприятий, которым предоставляется правовая охрана (интеллектуальную собственность), используемые при разработке, поддержке и эксплуатации службы Справочник JavaScript API Яндекс.Карт, включая, но не ограничиваясь, программы для ЭВМ, базы данных, изображения, тексты, другие произведения, а также изобретения, полезные модели, товарные знаки, знаки обслуживания, коммерческие обозначения и фирменные наименования, принадлежат ООО «ЯНДЕКС» либо его лицензиарам.

Использование результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации в целях, не связанных с разработкой, поддержкой и эксплуатацией службы Справочник JavaScript API Яндекс.Карт, не допускается без получения предварительного согласия правообладателя. Настоящий документ содержит конфиденциальную информацию ООО «ЯНДЕКС». Использование конфиденциальной информации в целях, не связанных с разработкой, поддержкой и эксплуатацией службы Справочник JavaScript API Яндекс.Карт, а равно как и разглашение таковой, не допускается. При этом под разглашением понимается любое действие или бездействие, в результате которых конфиденциальная информация в любой возможной форме (устной, письменной, иной форме, в том числе с использованием технических средств) становится известной третьим лицам без согласия обладателя такой информации либо вопреки трудовому или гражданско-правовому договору.

Отношения ООО «ЯНДЕКС» с лицами, привлекаемыми для разработки, поддержки и эксплуатации службы Справочник JavaScript API Яндекс.Карт, регулируются законодательством Российской Федерации и заключаемыми в соответствии с ним трудовыми и/или гражданско-правовыми договорами (соглашениями). Нарушение требований об охране результатов интеллектуальной деятельности и приравненных к ним средств индивидуализации, а равно как и конфиденциальной информации, влечет за собой дисциплинарную, гражданско-правовую, административную или уголовную ответственность в соответствии с законодательством Российской Федерации.

Контактная информация

ООО «ЯНДЕКС»

<https://www.yandex.ru>

Тел.: +7 495 739 7000

Email: pr@yandex-team.ru

Главный офис: 119021, Россия, г. Москва, ул. Льва Толстого, д. 16

Содержание

Справочник API Яндекс.Карт.....	11
Balloon.....	11
Описание событий.....	16
behavior.....	16
behavior.DbClickZoom.....	16
behavior.Drag.....	18
behavior.LeftMouseButtonMagnifier.....	20
behavior.MultiTouch.....	22
behavior.RightMouseButtonMagnifier.....	23
behavior.RouteEditor.....	25
behavior.Ruler.....	28
behavior.ScrollZoom.....	31
behavior.storage.....	32
borders.....	33
borders.load.....	33
Circle.....	35
Описание полей.....	47
clusterer.....	47
clusterer.addon.....	47
clusterer.Balloon.....	48
clusterer.Hint.....	51
Clusterer.....	53
Описание полей.....	59
Описание событий.....	60
Описание методов.....	60
ClusterPlacemark.....	63
Описание полей.....	75
Описание методов.....	76
Collection.....	76
Описание событий.....	79
Описание методов.....	79
collection.....	81
collection.Item.....	81
control.....	84
control.Button.....	84
control.FullscreenControl.....	93
control.GeolocationControl.....	100
control.ListBox.....	105
control.ListBoxItem.....	114
control.Manager.....	119
control.RouteButton.....	126
control.RouteEditor.....	131
control.RoutePanel.....	136
control.RulerControl.....	139
control.SearchControl.....	143
control.storage.....	157
control.TrafficControl.....	158
control.TypeSelector.....	166
control.ZoomControl.....	174
coordSystem.....	178
coordSystem.cartesian.....	178
coordSystem.geo.....	179

data.....	180
data.Manager.....	180
domEvent.....	184
domEvent.manager.....	184
domEvent.MultiPointer.....	188
domEvent.MultiTouch.....	189
domEvent.Pointer.....	189
domEvent.Touch.....	190
DomEvent.....	191
Описание методов.....	193
event.....	194
event.Group.....	194
event.Manager.....	195
event.Mapper.....	198
Event.....	199
Описание методов.....	201
findOrganization.....	203
formatter.....	203
Описание методов.....	204
geocode.....	205
GeocodeResult.....	207
Описание методов.....	212
geolocation.....	213
Описание методов.....	213
geometry.....	215
geometry.base.....	215
geometry.Circle.....	233
geometry.json.....	237
geometry.LineString.....	240
geometry.pixel.....	246
geometry.Point.....	263
geometry.Polygon.....	266
geometry.Rectangle.....	273
geometryEditor.....	277
geometryEditor.Circle.....	277
geometryEditor.LineString.....	279
geometryEditor.model.....	294
geometryEditor.Point.....	308
geometryEditor.Polygon.....	310
geometryEditor.view.....	326
GeoObject.....	328
Описание полей.....	345
Описание событий.....	347
geoObject.....	348
geoObject.addon.....	348
geoObject.Balloon.....	350
geoObject.Hint.....	354
geoObject.Sequence.....	357
GeoObjectCollection.....	362
Описание событий.....	368
Описание методов.....	369
geoQuery.....	371
GeoQueryResult.....	374
Описание методов.....	381
geoXml.....	413
geoXml.load.....	413
getZoomRange.....	414
graphics.....	415

graphics.style.....	415
Hint.....	417
hotspot.....	420
hotspot.layer.....	420
hotspot.Layer.....	430
hotspot.ObjectSource.....	445
Hotspot.....	453
interactivityModel.....	456
interactivityModel.storage.....	456
layer.....	456
layer.storage.....	456
layer.tile.....	457
layer.tileContainer.....	460
Layer.....	465
Описание методов.....	470
LayerCollection.....	473
Описание методов.....	475
layout.....	478
layout.Image.....	478
layout.ImageWithContent.....	482
layout.PieChart.....	486
layout.storage.....	492
layout.templateBased.....	492
LoadingObjectManager.....	497
Описание полей.....	504
Описание методов.....	505
Map.....	507
Описание полей.....	517
Описание событий.....	521
Описание методов.....	522
map.....	533
map.action.....	533
map.addon.....	543
map.Balloon.....	544
map.behavior.....	547
map.Container.....	552
map.Converter.....	556
map.Copyrights.....	558
map.GeoObjects.....	561
map.Hint.....	567
map.layer.....	569
map.margin.....	573
map.pane.....	578
map.ZoomRange.....	582
MapEvent.....	583
mapType.....	586
mapType.storage.....	586
MapType.....	586
Описание методов.....	587
meta.....	588
Описание полей.....	589
modules.....	590
modules.define.....	590
modules.isDefined.....	592
modules.require.....	593
Monitor.....	594
Описание методов.....	595
multiRouter.....	598

multiRouter.bicycle.....	598
multiRouter.driving.....	611
multiRouter.Editor.....	634
multiRouter.EditorAddon.....	646
multiRouter.masstransit.....	656
multiRouter.MultiRoute.....	694
multiRouter.MultiRouteModel.....	704
multiRouter.pedestrian.....	710
multiRouter.ViaPoint.....	731
multiRouter.ViaPointModel.....	735
multiRouter.WayPoint.....	738
multiRouter.WayPointModel.....	743
ObjectManager.....	746
Описание полей.....	752
Описание методов.....	752
objectManager.....	760
objectManager.addon.....	760
objectManager.Balloon.....	762
objectManager.ClusterCollection.....	766
objectManager.Hint.....	773
objectManager.ObjectCollection.....	777
objectManager.OverlayCollection.....	784
option.....	796
option.Manager.....	796
option.presetStorage.....	801
overlay.....	811
overlay.Circle.....	811
overlay.hotspot.....	816
overlay.html.....	840
overlay.Pin.....	864
overlay.Placemark.....	869
overlay.Polygon.....	875
overlay.Polyline.....	880
overlay.Rectangle.....	884
overlay.storage.....	889
pane.....	891
pane.EventsPane.....	891
pane.MovablePane.....	894
pane.StaticPane.....	897
Panorama.....	900
Описание методов.....	902
panorama.....	903
panorama.Base.....	903
panorama.createPlayer.....	909
panorama.isSupported.....	910
panorama.locate.....	910
panorama.Manager.....	911
panorama.Player.....	914
Placemark.....	923
Описание полей.....	936
Polygon.....	936
Описание полей.....	948
Polyline.....	948
Описание полей.....	959
Popup.....	959
projection.....	961
projection.Cartesian.....	961
projection.sphericalMercator.....	962

projection.wgs84Mercator.....	963
ready.....	964
Rectangle.....	965
Описание полей.....	977
regions.....	977
regions.load.....	977
RemoteObjectManager.....	979
Описание полей.....	987
Описание событий.....	987
Описание методов.....	987
route.....	990
router.....	993
router.addon.....	993
router.Editor.....	993
router.Path.....	997
router.Route.....	1004
router.Segment.....	1010
router.ViaPoint.....	1014
router.WayPoint.....	1020
shape.....	1027
shape.Circle.....	1027
shape.LineString.....	1028
shape.MultiGeometry.....	1029
shape.MultiPolygon.....	1031
shape.Polygon.....	1032
shape.Rectangle.....	1034
shape.storage.....	1035
suggest.....	1036
SuggestView.....	1036
Описание полей.....	1040
Описание событий.....	1040
Описание методов.....	1041
template.....	1041
template.filtersStorage.....	1041
Template.....	1042
Описание методов.....	1044
templateLayoutFactory.....	1045
Описание методов.....	1046
traffic.....	1046
traffic.provider.....	1046
util.....	1055
util.AsyncStorage.....	1055
util.augment.....	1059
util.bind.....	1059
util.bounds.....	1060
util.cursor.....	1067
util.defineClass.....	1070
util.Dragger.....	1072
util.extend.....	1076
util.hd.....	1076
util.math.....	1078
util.pixelBounds.....	1079
util.requireCenterAndZoom.....	1083
util.Storage.....	1084
vow.....	1086
Описание методов.....	1086
vow.Deferred.....	1088
vow.Promise.....	1089

Интерфейсы.....	1093
IBalloon.....	1093
IBalloonLayout.....	1095
IBalloonManager.....	1097
IBalloonOwner.....	1100
IBaseCircleGeometry.....	1101
IBaseGeometry.....	1103
IBaseLinearRingGeometry.....	1104
IBaseLineStringGeometry.....	1106
IBasePointGeometry.....	1109
IBasePolygonGeometry.....	1110
IBaseRectangleGeometry.....	1113
IBehavior.....	1115
ICanvasTile.....	1117
IChild.....	1119
IChildOnMap.....	1120
ICircleGeometry.....	1121
ICircleGeometryAccess.....	1124
ICollection.....	1127
IContainerPane.....	1129
IControl.....	1131
IControlParent.....	1133
ICoordSystem.....	1134
ICopyrightsAccessor.....	1138
ICopyrightsProvider.....	1139
ICustomizable.....	1141
IDataManager.....	1142
IDomEvent.....	1143
IDomEventEmitter.....	1146
IDomTile.....	1149
IEvent.....	1151
IEventController.....	1154
IEventEmitter.....	1155
IEventGroup.....	1155
IEventManager.....	1157
IEventPane.....	1161
IEventTrigger.....	1163
IEventWorkflowController.....	1164
IExpandableControlLayout.....	1166
IFreezable.....	1169
IGeocodeProvider.....	1170
IGeometry.....	1173
IGeometryEditor.....	1175
IGeometryEditorChildModel.....	1177
IGeometryEditorModel.....	1178
IGeometryEditorRootModel.....	1179
IGeometryJson.....	1180
IGeoObject.....	1181
IGeoObjectCollection.....	1185
IGeoObjectPopupData.....	1191
IGeoObjectSequence.....	1192
IGroupControlLayout.....	1195
IHint.....	1198
IHintManager.....	1200
IHintOwner.....	1201
IHotspot.....	1202
IHotspotLayerObject.....	1205
IHotspotObjectSource.....	1209

IHotspotShape.....	1211
Iterator.....	1215
ILayer.....	1216
ILayout.....	1219
ILinearRingGeometryAccess.....	1224
ILineStringGeometry.....	1229
ILineStringGeometryAccess.....	1232
IMapAction.....	1237
IMapObjectCollection.....	1239
IMapState.....	1240
IMultiRouteModelJson.....	1241
IMultiRouteParams.....	1241
IMultiRouteReferencePoint.....	1244
IMultiRouterRouteBalloon.....	1244
IOptionManager.....	1245
IOverlay.....	1249
IPane.....	1254
IPanorama.....	1256
IPanoramaConnection.....	1259
IPanoramaConnectionArrow.....	1259
IPanoramaConnectionMarker.....	1260
IPanoramaGraph.....	1261
IPanoramaGraphEdge.....	1262
IPanoramaGraphNode.....	1263
IPanoramaMarker.....	1263
IPanoramaMarkerIcon.....	1264
IPanoramaMarkerIconSet.....	1265
IPanoramaTileLevel.....	1266
IParentOnMap.....	1267
IPixelCircleGeometry.....	1268
IPixelGeometry.....	1269
IPixelLineStringGeometry.....	1271
IPixelMultiLineGeometry.....	1273
IPixelMultiPolygonGeometry.....	1275
IPixelPointGeometry.....	1279
IPixelPolygonGeometry.....	1281
IPixelRectangleGeometry.....	1285
IPointGeometry.....	1287
IPointGeometryAccess.....	1289
IPolygonGeometry.....	1290
IPolygonGeometryAccess.....	1293
IPopup.....	1300
IPopupManager.....	1303
IPositioningContext.....	1307
IProjection.....	1308
IPromiseProvider.....	1309
IRatioMap.....	1310
IRectangleGeometry.....	1310
IRectangleGeometryAccess.....	1313
IRoutePanel.....	1315
ISearchControlLayout.....	1319
ISearchProvider.....	1322
ISelectableControl.....	1325
ISelectableControlLayout.....	1327
IShape.....	1330
ISuggestProvider.....	1333
ISuggestViewLayout.....	1334
ITile.....	1335

ITrafficControlLayout.....	1336
ITrafficProvider.....	1340
ITransportProperties.....	1341
IZoomControlLayout.....	1342

Справочник API Яндекс.Карт

Справочник содержит описание программного интерфейса JavaScript API версии 2.1.79.

Дата выпуска версии 03 июня 2021 г.

Изменено:

- Обновлён логотип.

Исправлено:

- Правки слоя пробок.
- Правки блока "Открыть в Яндекс.Картах".
- Правки вёрстки.

Balloon

Расширяет [IBalloon](#), [Popup](#).

Балун — всплывающее облако, которое может отображать любое HTML-содержимое. Обычно балун существует на карте в единственном экземпляре, управление им происходит через специальные менеджеры (например, [карты](#), [геообъектов](#), [слоев активных областей](#) и т. д.). Не создавайте их самостоятельно, если в этом нет необходимости.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Balloon(map[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Ссылка на объект карты.
options	—	Тип: Object Опции.
options.autoPan	true	Тип: Boolean Сдвигать карту, чтобы отобразить открывшийся балун.
options.autoPanCheckZoomRange	false	Тип: Boolean Включает автомасштабирование при невозможности отобразить карту после перемещения на том же масштабе.

Параметр	Значение по умолчанию	Описание
options.autoPanDuration	500	Тип: Number Длительность перемещения к точке балуна (в мс).
options.autoPanMargin	34	Тип: Number Number[] Отступ или отступы от краев видимой области карты при выполнении действия autoPan. Значение может быть задано в виде одного числа (равный отступ для всех сторон), в виде двух чисел (для вертикальные и горизонтальные отступы) и в виде четырех чисел (в порядке: верхний, правый, нижний и левый отступ). Следует иметь в виду, данное значение будет суммироваться со значением рассчитываемым в менеджере отступов map.margin.Manager .
options.autoPanUseMapMargin	true	Тип: Boolean Нужно ли учитывать отступы карты map.margin.Manager при выполнении действия autoPan.
options.closeButton	true	Тип: Boolean Флаг наличия кнопки закрытия.
options.closeTimeout	700	Тип: Number Задержка закрытия (в мс).
options.contentLayout	—	Тип: Function String Макет для содержимого балуна. (Тип: конструктор объекта с интерфейсом ILayout или ключ макета).
options.interactivityModel	—	Тип: String Ключ модели интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
options.layout	islands#balloon	Тип: Function String Внешний макет балуна. (Тип: конструктор объекта с интерфейсом ILayout или ключ макета).

Параметр	Значение по умолчанию	Описание
<code>options.maxHeight</code>	—	Тип: Number Максимальная высота, в пикселах.
<code>options.maxWidth</code>	—	Тип: Number Максимальная ширина, в пикселах.
<code>options.minHeight</code>	—	Тип: Number Минимальная высота, в пикселах.
<code>options.minWidth</code>	—	Тип: Number Минимальная ширина, в пикселах.
<code>options.offset</code>	—	Тип: Number[] Дополнительное смещение позиции относительно точки привязки.
<code>options.openTimeout</code>	150	Тип: Number Задержка открытия (в мс).
<code>options.pane</code>	'balloon'	Тип: String Ключ пейна, в который помещается оверлей балуна.
<code>options.panelContentLayout</code>	null	Тип: Function String Макет для содержимого балуна в режиме панели. Если опция не задана, то используется значение опции <code>contentLayout</code> . (Тип: конструктор объекта с интерфейсом ILayout или ключ макета).
<code>options.panelMaxHeightRatio</code>	—	Тип: Number Максимальная высота балуна-панели. Задаётся как коэффициент отношения к высоте карты: числом от 0 до 1.
<code>options.panelMaxMapArea</code>	—	Тип: Number Максимальная площадь карты, при которой балун будет отображаться в виде панели. Можно отключить режим панели, задав значение 0, и наоборот, всегда отображать балун в виде панели, задав значение <i>Infinity</i> .

Параметр	Значение по умолчанию	Описание
<code>options.shadow</code>	true	Тип: Boolean Флаг наличия тени.
<code>options.shadowLayout</code>	—	Тип: Function String Макет тени. (Тип: конструктор объекта с интерфейсом ILayout или ключ макета).
<code>options.shadowOffset</code>	—	Тип: Number[] Дополнительное смещение позиции тени относительно точки привязки.
<code>options.zIndex</code>	—	Тип: String z-index балуна.

* Обязательный параметр/опция.

Пример:

```
// Создание независимого экземпляра балуна и отображение его в центре карты.
var balloon = new ymaps.Balloon(myMap);
// Здесь родительскими устанавливаются опции карты,
// где содержатся значения по умолчанию для обязательных опций.
balloon.options.setParent(myMap.options);
// Открываем балун в центре карты:
balloon.open(myMap.getCenter());
```

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
<code>options</code>	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
<code>autopanbegin</code>	Начало автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event .
<code>autopanend</code>	Окончание автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event .
<code>beforeuserclose</code>	Событие, предшествующее событию Balloon.userclose . Позволяет отменить пользовательское действие посредством вызова метода <code>preventDefault</code> . Экземпляр класса Event .
<code>close</code>	Закрытие инфо-объекта. Унаследовано от IPopup .
<code>open</code>	Открытие инфо-объекта. Унаследовано от IPopup .

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
userclose	Закрытие балуна пользователем. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
autoPan()	vow.Promise	Перемещает карту таким образом, чтобы балун оказался видимым. Унаследован от IBalloon .
close([force])	vow.Promise	Закрывает инфо-объект. Унаследован от IPopup .
getData()		Возвращает данные инфо-объекта. Унаследован от IPopup .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopup .
getOverlaySync()	IOverlay	Возвращает оверлей, если тот существует. Унаследован от IPopup .
getPosition()		Возвращает координаты инфо-объекта. Унаследован от IPopup .
isOpen()	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopup .
open([position[, data]])	vow.Promise	Открывает инфо-объект в указанной позиции. Если инфо-объект уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией IProjection , которая содержится в опциях. Унаследован от IPopup .
setData(data)	vow.Promise	Задаёт инфо-объекту новые данные. Унаследован от IPopup .
setPosition(position)	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopup .

Описание событий

autopanbegin

Начало автоматического перемещения центра карты, вызванного методом `autoPan`. Экземпляр класса [Event](#).

autopanend

Окончание автоматического перемещения центра карты, вызванного методом `autoPan`. Экземпляр класса [Event](#).

beforeuserclose

Событие, предшествующее событию [Balloon.userclose](#). Позволяет отменить пользовательское действие посредством вызова метода `preventDefault`. Экземпляр класса [Event](#).

userclose

Закрытие балуна пользователем. Экземпляр класса [Event](#).

behavior

behavior.DblClickZoom

Расширяет [IBehavior](#).

Поведение "масштабирование карты двойным щелчком кнопки мыши".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
behavior.DblClickZoom([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции.
options.centering	true	Тип: Boolean true - при двойном щелчке мышью карта масштабируется так, что точка под курсором мыши становится новым центром карты; false - точка под курсором мыши сохраняет свое положение при масштабировании двойным щелчком.
options.duration	200	Тип: Number Время анимации масштабирования двойным щелчком кнопки мыши (0 - анимации нет).

Параметр	Значение по умолчанию	Описание
options.useMapMargin	true	Тип: Boolean Нужно ли при центрировании учитывать отступы, которые были рассчитаны в менеджере отступов map.margin.Manager .

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
disable	Выключение поведения. Унаследовано от IBehavior .
enable	Включение поведения. Унаследовано от IBehavior .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение. Унаследован от IBehavior .
enable()		Включает поведение. Унаследован от IBehavior .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .

Имя	Возвращает	Описание
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

behavior.Drag

Расширяет [IBehavior](#).

Поведение "перетаскивание карты с помощью мыши либо одиночного касания".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
behavior.Drag([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции.
options.actionCursor	'grabbing'	Тип: String Курсор для поведения behavior.Drag при перетаскивании карты.
options.cursor	'grab'	Тип: String Курсор для поведения behavior.Drag при наведении на карту.
options.inertia	true	Тип: Boolean Включает кинетическую инерцию движения в конце перетаскивания.
options.inertiaDuration	400	Тип: Number String Продолжительность инерции в мс. Строковое значение "auto" задает продолжительность инерции пропорционально расстоянию перетаскивания.

Параметр	Значение по умолчанию	Описание
options.tremor	2	Тип: Integer Минимальное смещение курсора после нажатия кнопки мыши, после которого начинает двигаться карта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
disable	Выключение поведения. Унаследовано от IBehavior .
enable	Включение поведения. Унаследовано от IBehavior .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение. Унаследован от IBehavior .
enable()		Включает поведение. Унаследован от IBehavior .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .

Имя	Возвращает	Описание
<code>isEnabled()</code>	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior .
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

behavior.LeftMouseButtonMagnifier

Расширяет [IBehavior](#).

Поведение "масштабирование карты при выделении области левой кнопкой мыши".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
behavior.LeftMouseButtonMagnifier([options])
```

Создает поведение "масштабирование карты при выделении области левой кнопкой мыши".

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.actionCursor</code>	'crosshair'	Тип: String Курсор во время выбора области для увеличения при включенном поведении behavior.LeftMouseButtonMagnifier . Список всех доступных курсоров описан в Менеджере курсоров .
<code>options.cursor</code>	'zoom'	Тип: String Курсор при включенном поведении behavior.LeftMouseButtonMagnifier . Список всех доступных курсоров описан в Менеджере курсоров .
<code>options.duration</code>	300	Тип: Number Продолжительность анимации увеличения области при использовании поведения behavior.LeftMouseButtonMagnifier , мс.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
disable	Выключение поведения. Унаследовано от IBehavior .
enable	Включение поведения. Унаследовано от IBehavior .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение. Унаследован от IBehavior .
enable()		Включает поведение. Унаследован от IBehavior .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior .

Имя	Возвращает	Описание
setParent(parent)	IChildOnMap	<p>Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта.</p> <p>Унаследован от IChildOnMap.</p>

behavior.MultiTouch

Расширяет [IBehavior](#).

Поведение "масштабирование карты мультисенсорным касанием".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
behavior.MultiTouch([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции.</p>
options.tremor	2	<p>Тип: Number</p> <p>Минимальное движение по экрану устройства (в пикселях), при котором сработает поведение behavior.MultiTouch.</p>

Поля

Имя	Тип	Описание
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>

События

Имя	Описание
disable	<p>Выключение поведения.</p> <p>Унаследовано от IBehavior.</p>
enable	<p>Включение поведения.</p> <p>Унаследовано от IBehavior.</p>

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение. Унаследован от IBehavior .
enable()		Включает поведение. Унаследован от IBehavior .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

behavior.RightMouseButtonMagnifier

Расширяет [IBehavior](#).

Поведение "масштабирование карты при выделении области правой кнопкой мыши".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
behavior.RightMouseButtonMagnifier([options])
```

Создает поведение "масштабирование карты при выделении области правой кнопкой мыши".

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции.
options.actionCursor	'crosshair'	Тип: String Курсор во время выбора области для увеличения при включенном поведении behavior.LeftMouseButtonMagnifier . Список всех доступных курсоров описан в Менеджере курсоров .
options.duration	300	Тип: Number Продолжительность анимации при выделении области при помощи поведения behavior.RightMouseButtonMagnifier , мс.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
disable	Выключение поведения. Унаследовано от IBehavior .
enable	Включение поведения. Унаследовано от IBehavior .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение. Унаследован от IBehavior .
enable()		Включает поведение. Унаследован от IBehavior .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

behavior.RouteEditor

Расширяет [IBehavior](#).

Поведение "редактор маршрутов".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
behavior.RouteEditor()
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
disable	Выключение поведения. Унаследовано от IBehavior .

Имя	Описание
enable	Включение поведения. Унаследовано от IBehavior .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
routechange	Событие смены маршрута вследствие вызова метода <code>setState</code> или включения/выключения поведения. Старый и новый маршрут можно получить через свойства события 'oldRoute' и 'newRoute' соответственно.

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение. Унаследован от IBehavior .
enable()		Включает поведение. Унаследован от IBehavior .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getRoute()	router.Route	Возвращает маршрут.
getState()	String	Возвращает текущее состояние редактора маршрутов в закодированном виде.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Имя	Возвращает	Описание
<code>setState(state)</code>		Восстанавливает состояние редактора маршрутов из закодированной строки. При установлении нового маршрута будет вызвано событие <code>routechange</code> .

Описание событий

`routechange`

Событие смены маршрута вследствие вызова метода `setState` или включения/выключения поведения. Старый и новый маршрут можно получить через свойства события `'oldRoute'` и `'newRoute'` соответственно.

Описание методов

`getRoute`

```
{router.Route} getRoute()
```

Возвращает маршрут.

`getState`

```
{String} getState()
```

Возвращает текущее состояние редактора маршрутов в закодированном виде.

`setState`

```
{ } setState(state)
```

Восстанавливает состояние редактора маршрутов из закодированной строки. При установлении нового маршрута будет вызвано событие `routechange`.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>state *</code>	—	Тип: String null Закодированное состояние редактора маршрутов. Состояние задается в следующем формате: <code>rt=точка1~точка2~точка3~точка4...&via=индексы-via-points</code> Например, если закодированная строка выглядит так: <code>"rt=50,30~42,35~45,32~40,30&via=1,2"</code> , это значит что путевыми точками маршрута должны быть "50,30" и "40,30", а промежуточными "42,35" и "45,32". Если параметр <code>via</code> не указывать, то маршрут будет состоять только из путевых точек.

* Обязательный параметр/опция.

Пример:

```
var behavior = map.behaviors.get('routeEditor');
behavior.events.add('routechange', function (e) {
    var newRoute = e.get('newRoute');
    alert(newRoute.getLength()); // Длина нового маршрута.
    setTimeout(function () {
        // Удаляем маршрут.
        behavior.setState(null);
    }, 1000);
});
behavior.setState('rt=55.874872,37.562677~55.92517867214157,37.62725433916199~55.920011490602526,37.6629269628905&via=1');
```

behavior.Ruler

Расширяет [IBehavior](#).

Поведение "Линейка". Позволяет отмечать на карте точки и отображать расстояния между ними.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
behavior.Ruler([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции.
options.balloonAutoPan	true	Тип: Boolean Нужно ли автопозиционирование карты при открытии балуна линейки.
options.balloonAutoPanUseMapMargin	true	Тип: Boolean Нужно ли учитывать отступы карты map.margin.Manager при выполнении действия autoPan для балуна линейки.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
geometry	geometry.LineString	Геометрия поведения "Линейка".
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
disable	Выключение поведения. Унаследовано от IBehavior .

Имя	Описание
enable	Включение поведения. Унаследовано от IBehavior .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
close()	Boolean	Удаляет все точки линейки. Если текущее количество точек больше двух, то будет выведен запрос на подтверждение действия.
disable()		Выключает поведение. Унаследован от IBehavior .
enable()		Включает поведение. Унаследован от IBehavior .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getState()	String	Состояние линейки описывается строкой, состоящей из последовательностей, разделяемых символом "~". Каждая последовательность представляет собой подстроку вида "долгота,широта" и описывает приращение координат относительно предыдущей точки линейки.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior .

Имя	Возвращает	Описание
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .
<code>setState(state)</code>		Восстанавливает состояние линейки из закодированной строки. Смотрите <code>behavior.Ruler.getState</code> .

Описание полей

geometry

```
{geometry.LineString} geometry
```

Геометрия поведения "Линейка".

Пример:

```
// Задаем координаты первой точки линейки.  
myMap.behaviors.get('ruler').geometry.set(0, [0, 0]);
```

Описание методов

close

```
{Boolean} close()
```

Удаляет все точки линейки. Если текущее количество точек больше двух, то будет выведен запрос на подтверждение действия.

Возвращает true, если действие произведено успешно.

getState

```
{String} getState()
```

Состояние линейки описывается строкой, состоящей из последовательностей, разделяемых символом "~". Каждая последовательность представляет собой подстроку вида "долгота,широта" и описывает приращение координат относительно предыдущей точки линейки.

Возвращает текущее состояние линейки в закодированном виде.

setState

```
{ } setState(state)
```

Восстанавливает состояние линейки из закодированной строки. Смотрите `behavior.Ruler.getState`.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>state</code> *	—	Тип: String Закодированное состояние линейки.

* Обязательный параметр/опция.

behavior.ScrollZoom

Расширяет [IBehavior](#).

Поведение "масштабирование карты колесом мыши".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
behavior.ScrollZoom([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.maximumDelta</code>	5	Тип: Number Максимальное возможное изменение, на которое можно изменить зум карты при одном событии скролла (непрерывным вращением колеса мыши). При достижении этого лимита карта перерисовывается (показываются тайлы для достигнутого уровня масштаба).
<code>options.speed</code>	5	Тип: Number Скорость увеличения/уменьшения масштаба карты при включенном поведении behavior.ScrollZoom , уровней масштаба в секунду.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
<code>options</code>	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
disable	Выключение поведения. Унаследовано от IBehavior .
enable	Включение поведения. Унаследовано от IBehavior .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение. Унаследован от IBehavior .
enable()		Включает поведение. Унаследован от IBehavior .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

behavior.storage

Статический объект.

Экземпляр класса [util.Storage](#)

Хранилище классов поведений карты. Добавление нового поведения на карту производится через это хранилище.

По умолчанию в хранилище добавлены следующие поведения:

- "drag" - перемещение карты при нажатой левой кнопке мыши либо одиночным касанием [behavior.Drag](#);
- "scrollZoom" - изменение масштаба колесом мыши [behavior.ScrollZoom](#) (только для десктопных браузеров);
- "dblClickZoom" - масштабирование карты двойным щелчком кнопки мыши [behavior.DblClickZoom](#);
- "multiTouch" - масштабирование карты мультисенсорным управлением (например, на сенсорном экране) [behavior.MultiTouch](#) (только для мобильных браузеров);
- "rightMouseButtonMagnifier" - увеличение области, выделенной правой кнопкой мыши (только для настольных браузеров), [behavior.RightMouseButtonMagnifier](#) (только для десктопных браузеров);
- "leftMouseButtonMagnifier" - увеличение области, выделенной левой кнопкой мыши либо одиночным касанием, [behavior.LeftMouseButtonMagnifier](#);
- "ruler" - измерение расстояния [behavior.Ruler](#).
- "routeEditor" - редактор маршрутов [behavior.RouteEditor](#);

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.

borders

borders.load

Статическая функция.

Предоставляет доступ к геометрии различных регионов и стран мира. Если на сайте используется политика CSP, то для работы с Регионами в директиве 'connect-src' необходимо указать домены: <https://api-maps.yandex.ru> <https://suggest-maps.yandex.ru> https://*.maps.yandex.net <https://yandex.ru>

Возвращает объект-promise.

```
{ vow.Promise } borders.load(region[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>region *</code>	—	<p>Тип: String</p> <p>ISO_3166-1 код страны (RU, UA, BY, KZ) для загрузки административно-территориального деления, либо '001' для загрузки геометрии границ стран мира, либо AQ для загрузки геометрии границ Антарктиды.</p>
<code>options</code>	—	<p>Тип: Object</p> <p>Опции отображения.</p>
<code>options.disputedBorders</code>	—	<p>Тип: String</p> <p>Двухбуквенный код государства, официальная позиция которого будет использоваться для определения административного подчинения спорных территорий. Допустимые значения: 'RU', 'UA', 'UN'. По умолчанию совпадает с кодом страны, который указан при загрузке API. Неподдерживаемые коды стран приводятся к RU. Для региона '001' (границы стран мира) поддерживается код 'UN' — мировые границы согласно позиции ООН.</p>
<code>options.lang</code>	—	<p>Тип: String</p> <p>Язык (ru, uk, en, be, kk, tr).</p>
<code>options.quality</code>	1	<p>Тип: Number</p> <p>Уровень качества. Доступные значения:</p> <ul style="list-style-type: none"> • 0 - минимальное качество; • 1 - стандартное качество; • 2 - повышенное качество; • 3 - высокое качество. <p>Уровень качества влияет на точность представления кривых и объем файла данных.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
// Отобразим на карте регионы с помощью ObjectManager.
ymaps.borders.load('RU', {
  lang: 'en'
```

```
}).then(function (geojson) {  
    var features = geojson.features.map(function (feature) {  
        feature.id = feature.properties.iso3166;  
        return feature;  
    });  
    var objectManager = new ymaps.ObjectManager();  
    objectManager.add(features);  
    myMap.geoObjects.add(objectManager);  
});
```

2.

```
// Добавим на карту регионы через GeoObject.  
ymaps.borders.load('RU', {  
    lang: 'en'  
}).then(function (geojson) {  
    for (var i = 0; i < geojson.features.length; i++) {  
        var geoObject = new ymaps.GeoObject(geojson.features[i]);  
        myMap.geoObjects.add(geoObject);  
    }  
});
```

Circle

Расширяет [GeoObject](#).

Круг. Представляет собой геообъект с геометрией [geometry.Circle](#).

См.: [GeoObject](#) [geometry.Circle](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Circle(geometry[, properties[, options]])
```

Создает экземпляр круга.

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: ICircleGeometry Number [] Object Ссылка на объект геометрии точки или массив, первый элемент которого координаты центра круга, второй - радиус в метрах, или объект с параметрами геометрии.

Параметр	Значение по умолчанию	Описание
properties	—	<p>Тип: Object IDataManager</p> <p>Данные круга. Соответствуют данным класса GeoObject. Могут задаваться как экземпляр класса, реализующего интерфейс IDataManager, либо в виде хэша. При значениях опций по умолчанию кругу трактуются следующие поля данных:</p> <ul style="list-style-type: none">• <code>hintContent</code> - содержимое всплывающей подсказки круга;• <code>balloonContent</code> - содержимое балуна круга;• <code>balloonContentHeader</code> - содержимое заголовка балуна круга;• <code>balloonContentBody</code> - содержимое основной части балуна круга;• <code>balloonContentFooter</code> - содержимое нижней части балуна круга. <p>Поле <code>balloonContent</code> является кратким обозначением для поля <code>balloonContentBody</code>, но при одновременном задании <code>balloonContentBody</code> более приоритетен. Также вы можете дополнить данные круга своими собственными полями и использовать их везде, где это возможно. Например, в макете круга или макете балуна.</p>

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции круга. Используя этот параметр можно задавать как опции самого круга, так и опции его составных частей:</p> <ul style="list-style-type: none"> Опции балуна круга с префиксом <code>balloon</code>. Опции всплывающей подсказки круга с префиксом <code>hint</code>. Опции редактора геометрии круга с префиксом <code>editor</code>. Смотрите описание класса geometryEditor.Circle. Опции геометрии могут задаваться без префикса. Смотрите описание класса IGeometry геометрии geometry.Circle.
options.circleOverlay	"default#circle"	<p>Тип: String Function</p> <p>Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра:</p> <ul style="list-style-type: none"> <code>geometry</code>: IPixelCircleGeometry - непосредственно пиксельная геометрия; <code>data</code>: Object - данные оверлея; <code>options</code>: Object - опции оверлея. <p>Функция возвращает объект vow.Promise.</p>
options.cursor	"pointer"	<p>Тип: String</p> <p>Вид курсора над кругом.</p>
options.draggable	false	<p>Тип: Boolean</p> <p>Определяет возможность перетаскивания круга.</p>

Параметр	Значение по умолчанию	Описание
<code>options.fill</code>	<code>true</code>	Тип: Boolean Наличие заливки фигуры.
<code>options.fillColor</code>	<code>"0066ff99"</code>	Тип: String Цвет заливки.
<code>options.fillImageHref</code>	—	Тип: String Фоновое изображение. При включении данной опции в режиме заливки stretch значение опции <code>fillColor</code> игнорируется.
<code>options.fillMethod</code>	<code>'stretch'</code>	Тип: String Тип заливки фоном. Может принимать одно из двух значений: <ul style="list-style-type: none"> <code>stretch</code> - фоновое изображение растягивается по размеру оверлея. <code>tile</code> - Фоновое изображение повторяется без изменений размера. Аналог <code>background-repeat</code> в css. Можно использовать для заливки фигуры неким шаблоном.
<code>options.fillOpacity</code>	<code>1</code>	Тип: Number Прозрачность заливки.
<code>options.hasBalloon</code>	<code>true</code>	Тип: Boolean Определяет наличие поля <code>balloon</code> у круга.
<code>options.hasHint</code>	<code>true</code>	Тип: Boolean Определяет наличие поля <code>hint</code> у круга.
<code>options.hideIconOnBalloonOpen</code>	<code>true</code>	Тип: Boolean Скрывать иконку при открытии балуна.
<code>options.interactiveZIndex</code>	<code>false</code>	Тип: Boolean Включает режим автоматического изменения <code>z-index</code> круга в зависимости от его состояния.

Параметр	Значение по умолчанию	Описание
options.interactivityModel	"default#geoObject"	<p>Тип: String</p> <p>Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage.</p>
options.opacity	1	<p>Тип: Number</p> <p>Прозрачность.</p>
options.openBalloonOnClick	true	<p>Тип: Boolean</p> <p>Определяет, показывать ли балун при щелчке на круге.</p>
options.openEmptyBalloon	false	<p>Тип: Boolean</p> <p>Определяет, показывать ли пустой балун при щелчке на круге.</p>
options.openEmptyHint	false	<p>Тип: Boolean</p> <p>Определяет, показывать ли пустой хинт при наведении указателя мыши на круг.</p>
options.openHintOnHover	true	<p>Тип: Boolean</p> <p>Определяет, показывать ли хинт при наведении указателя мыши на круг.</p>
options.outline	true	<p>Тип: Boolean</p> <p>Наличие обводки круга.</p>
options.pane	"areas"	<p>Тип: String</p> <p>Ключ пейна, в который помещается оверлей круга.</p>
options.strokeColor	"0066ffff"	<p>Тип: String String[]</p> <p>Цвет линии или обводки. Можно задать несколько значений для множественной обводки.</p>
options.strokeOpacity	1	<p>Тип: Number Number[]</p> <p>Прозрачность линии или обводки. Можно задать несколько значений для множественной обводки.</p>

Параметр	Значение по умолчанию	Описание
options.strokeStyle	—	<p>Тип: String Object[String[] Object[]]</p> <p>Стиль линии или обводки. Доступные стили перечислены в объекте graphics.style.stroke.</p>
options.strokeWidth	1	<p>Тип: Number Number[]</p> <p>Толщина линии или обводки. Можно задать несколько значений для множественной обводки.</p>
options.syncOverlayInit	false	<p>Тип: Boolean</p> <p>Включает синхронное добавление оверлея на карту. По умолчанию добавление оверлея осуществляется асинхронно, что позволяет предотвратить зависания браузера при добавлении на карту большого числа геообъектов. Однако, асинхронное добавление не позволяет получать доступ к оверлею сразу после добавления круга на карту.</p>
options.useMapMarginInDragging	true	<p>Тип: Boolean</p> <p>При перетаскивании объекта к краю карты происходит автоматическое изменение центра карты. Нужно ли учитывать отступы карты при автоматическом смещении центра карты map.margin.Manager.</p>
options.visible	true	<p>Тип: Boolean</p> <p>Определяет видимость круга.</p>
options.zIndex	—	<p>Тип: Number</p> <p>z-index круга в обычном состоянии. Наименее приоритетный.</p>
options.zIndexActive	—	<p>Тип: Number</p> <p>z-index круга с открытым балуном. Наиболее приоритетный.</p>
options.zIndexDrag	—	<p>Тип: Number</p> <p>z-index круга при перетаскивании.</p>
options.zIndexHover	—	<p>Тип: Number</p> <p>z-index круга при наведении на него указателя мыши.</p>

* Обязательный параметр/опция.

Пример:

```
// Создаем геодезический круг радиусом 1000 километров.
var circle = new ymaps.Circle([[50, 75], 1000000], {}, {
  geodesic: true
});
// Добавляем круг на карту.
myMap.geoObjects.add(circle);
```

Поля

Имя	Тип	Описание
balloon	geoObject.Balloon	Балун геообъекта. Унаследовано от GeoObject .
editor	geometryEditor.Circle	Редактор геометрии "Круг".
events	event.Manager	Менеджер событий. Унаследовано от GeoObject .
geometry	geometry.Circle	Геометрия типа "Круг".
hint	geoObject.Hint	Хинт геообъекта. Унаследовано от GeoObject .
indices	ArrayBuffer	
options	option.Manager	Менеджер опций геообъекта. Унаследовано от GeoObject .
properties	data.Manager	Менеджер данных геообъекта. Унаследовано от GeoObject .
state	data.Manager	Состояние геообъекта. Определяется следующими полями: <ul style="list-style-type: none"> • <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. • <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; • <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается. Унаследовано от GeoObject .
vertices	ArrayBuffer	

События

Имя	Описание
balloonclose	Закрытие балуна. Экземпляр класса Event . Унаследовано от GeoObject .
balloonopen	Открытие балуна на геообъекте. Экземпляр класса Event . Унаследовано от GeoObject .

Имя	Описание
beforedrag	<p>Событие, предшествующее событию drag. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p> <p>Унаследовано от GeoObject.</p>
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p> <p>Унаследовано от GeoObject.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent - оригинальное событие редактора геометрии. <p>Унаследовано от GeoObject.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Закрытие хинта. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>

Имя	Описание
hintopen	Открытие хинта на геообъекте. Экземпляр класса Event . Унаследовано от GeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mousemove	Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent . Унаследовано от IDomEventEmitter .

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. <p>Унаследован от IChildOnMap.</p>
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. <p>Унаследован от IChildOnMap.</p>

Описание полей

editor

```
{geometryEditor.Circle} editor
```

Редактор геометрии "Круг".

geometry

```
{geometry.Circle} geometry
```

Геометрия типа "Круг".

indices

```
{ArrayBuffer} indices
```

vertices

```
{ArrayBuffer} vertices
```

clusterer

clusterer.addon

clusterer.addon.balloon

Примечание: Конструктор класса `clusterer.addon.balloon` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Методы

Методы

Имя	Возвращает	Описание
<code>get(clusterer)</code>	<code>IPopupManager</code>	Возвращает менеджер балуна кластеризатора.

Описание методов

get

```
{IPopupManager} get(clusterer)
```

Возвращает менеджер балуна кластеризатора.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>clusterer *</code>	—	Тип: <code>Clusterer</code> Кластеризатор

* Обязательный параметр/опция.

Пример:

```
ymaps.clusterer.addon.balloon.get(clusterer)
```

clusterer.addon.hint

Примечание: Конструктор класса `clusterer.addon.hint` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

[Методы](#)**Методы**

Имя	Возвращает	Описание
get(clusterer)	IPopupManager	Возвращает менеджер хинта кластеризатора.

Описание методов**get**

```
{IPopupManager} get(clusterer)
```

Возвращает менеджер хинта кластеризатора.

Параметры:

Параметр	Значение по умолчанию	Описание
clusterer *	—	Тип: Clusterer Кластеризатор

* Обязательный параметр/опция.

Пример:

```
ymaps.clusterer.addon.hint.get(clusterer)
```

clusterer.Balloon

Расширяет [IBalloonManager](#).

Менеджер балуна кластеризатора. Позволяет управлять балуном кластера, открывать и скрывать его. Использует внутри себя менеджер балуна карты [map.Balloon](#). Кластеризаторы содержат экземпляр этого класса, доступный как `myClusterer.balloon`. Не создавайте экземпляры этого класса без необходимости.

См.: [Balloon](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
clusterer.Balloon(clusterer)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>clusterer</code> *	—	Тип: Clusterer Кластеризатор.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
<code>open</code>		Открытие балуна на кластере. Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на менеджер балуна кластеризатора; <code>cluster</code> - ссылка на объект-кластер. Экземпляр класса Event .

События

Имя	Описание
<code>autopanbegin</code>	Начало автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
<code>autopanend</code>	Окончание автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
<code>beforeuserclose</code>	Событие, предшествующее событию Balloon.userclose . Позволяет отменить пользовательское действие посредством вызова метода <code>preventDefault</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
<code>close</code>	Закрытие балуна. <ul style="list-style-type: none"> <code>target</code> - ссылка на менеджер балуна кластеризатора; <code>cluster</code> - ссылка на объект-кластер. Экземпляр класса Event .

Имя	Описание
open	Открытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. Унаследовано от IPopupManager .
userclose	Заккрытие балуна пользователем. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .

Методы

Имя	Возвращает	Описание
autoPan()	vow.Promise	Перемещает карту таким образом, чтобы балун оказался видимым. Унаследован от IBalloonManager .
close([force])	vow.Promise	Закрывает инфо-объект. Унаследован от IPopupManager .
destroy()		Отключает менеджер инфо-объекта. Унаследован от IPopupManager .
getData()	<code>Object null</code>	Возвращает данные инфо-объекта или <code>null</code> . Унаследован от IPopupManager .
getOptions()	IOptionManager null	Возвращает менеджер опций или <code>null</code> . Унаследован от IPopupManager .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopupManager .
getOverlaySync()	IOverlay null	Возвращает оверлей, если тот существует. Унаследован от IPopupManager .
getPosition()	<code>Number[] null</code>	Возвращает координаты инфо-объекта или <code>'null'</code> . Унаследован от IPopupManager .

Имя	Возвращает	Описание
<code>isOpen()</code>	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopupManager .
<code>setData(data)</code>	vow.Promise	Задаёт инфо-объекту новые данные. Унаследован от IPopupManager .
<code>setOptions(options)</code>	vow.Promise	Задаёт инфо-объекту новые опции. Унаследован от IPopupManager .
<code>setPosition(position)</code>	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopupManager .

Описание полей

open

open

Открытие балуна на кластере. Имена полей, доступных через метод [Event.get](#):

- target - ссылка на менеджер балуна кластеризатора;
- cluster - ссылка на объект-кластер.

Экземпляр класса [Event](#).

Пример:

```
clusterer.balloon.events.add('open', function (e) {
  var clusterPlacemark = e.get('cluster');
});
```

Описание событий

close

Закрытие балуна.

- target - ссылка на менеджер балуна кластеризатора;
- cluster - ссылка на объект-кластер.

Экземпляр класса [Event](#).

clusterer.Hint

Расширяет [IHintManager](#).

Менеджер хинта кластеризатора. Позволяет управлять хинтом кластера, открывать и скрывать его. Использует внутри себя менеджер хинтов карты [map.Hint](#). Кластеризаторы содержат экземпляр этого класса, доступный как `myClusterer.hint`. Не создавайте экземпляры этого класса без необходимости.

См.: [Hint](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
clusterer.Hint(clusterer)
```

Параметры:

Параметр	Значение по умолчанию	Описание
clusterer *	—	Тип: Clusterer Кластеризатор.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
close	Заккрытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. Унаследовано от IPopupManager .
open	Открытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. Унаследовано от IPopupManager .

Методы

Имя	Возвращает	Описание
close ([force])	vow.Promise	Закрывает инфо-объект. Унаследован от IPopupManager .
destroy ()		Отключает менеджер инфо-объекта. Унаследован от IPopupManager .
getData ()	Object null	Возвращает данные инфо-объекта или null. Унаследован от IPopupManager .

Имя	Возвращает	Описание
getOptions()	IOptionManager null	Возвращает менеджер опций или null. Унаследован от IPopupManager .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopupManager .
getOverlaySync()	IOverlay null	Возвращает оверлей, если тот существует. Унаследован от IPopupManager .
getPosition()	Number[] null	Возвращает координаты инфо-объекта или `null`. Унаследован от IPopupManager .
isOpen()	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopupManager .
open([position[, data[, options]])	vow.Promise	Открывает инфо-объект в указанной позиции. Унаследован от IPopupManager .
setData(data)	vow.Promise	Задаёт инфо-объекту новые данные. Унаследован от IPopupManager .
setOptions(options)	vow.Promise	Задаёт инфо-объекту новые опции. Унаследован от IPopupManager .
setPosition(position)	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopupManager .

Clusterer

Расширяет [IChildOnMap](#), [ICustomizable](#), [IEventEmitter](#), [IParentOnMap](#).

Кластеризатор геообъектов. Кластеризует объекты в видимой области карты. Если объект не попадает в видимую область карты, он не будет добавлен на карту. Обратите внимание, что кластеризатор не поддерживает реакцию на изменение координат объектов (программно, или в результате драга). Если у объекта в кластеризаторе необходимо изменить координаты, следует удалить объект из кластеризатора, поменять его геометрию, а затем добавить обратно.

См.: [ClusterPlacemark](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Clusterer([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции. Опции для дочерних объектов-кластеров задаются с префиксом "cluster". См. ClusterPlacemark .
options.gridSize	64	Тип: Number Размер ячейки кластеризации в пикселях. Значение должно быть равно 2^n (в область 256 на 256 пикселей должно уместиться равное количество ячеек).
options.groupByCoordinates	false	Тип: Boolean Специальный режим работы кластеризатора при котором кластеры образуются только из геобъектов с одинаковыми координатами.
options.hasBalloon	true	Тип: Boolean Флаг наличия у кластеризатора поля <code>.balloon</code> . Если при клике на кластер не нужно открывать балун, рекомендуется установить эту опцию в значение false, чтобы избежать лишних инициализаций;
options.hasHint	true	Тип: Boolean Флаг наличия у кластеризатора поля <code>.hint</code> . Если при наведении на кластер не нужно показывать всплывающую подсказку, рекомендуется установить эту опцию в значение false, чтобы избежать лишних инициализаций;

Параметр	Значение по умолчанию	Описание
options.margin	10	<p>Тип: Number Number[]</p> <p>Число или массив чисел, задающие отступ для центра кластера относительно ячеек кластеризации. Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это вертикальные и горизонтальные отступы соответственно. Если задан массив из 4х чисел, то это отступы top, right, bottom, left.</p>
options.maxZoom	Infinity	<p>Тип: Number[]</p> <p>Максимальный коэффициент масштабирования карты, на котором происходит кластеризация объектов. Даже если кластеризация отключена, будут показаны только объекты в видимой области карты.</p>
options.minClusterSize	2	<p>Тип: Number</p> <p>Минимальное количество объектов, образующих кластер.</p>
options.preset	—	<p>Тип: String</p> <p>Ключ предустановленных опций кластеризатора. Список ключей, доступных в пакете <code>package.clusters</code>, содержится в описании option.presetStorage.</p>
options.showInAlphabeticalOrder	false	<p>Тип: Boolean</p> <p>Показывать метки в балуне в алфавитном порядке при нажатии на кластер. Геообъекты кластера сортируются по специальным полям в данных этих геообъектов - <code>clusterCaption</code> (или <code>balloonContentHeader</code>, если предыдущее поле не определено). По умолчанию геообъекты показываются в порядке добавления в кластеризатор.</p>
options.useMapMargin	true	<p>Тип: Boolean</p> <p>Нужно ли учитывать отступы карты map.margin.Manager при приближении карты после клика на кластере.</p>

Параметр	Значение по умолчанию	Описание
<code>options.viewportMargin</code>	128	<p>Тип: Number Number[]</p> <p>Отступ для области, в которой производится кластеризация. Кластеризация проводится для видимой области карты. С помощью данной опции область кластеризации расширяется по отношению к видимой области карты.</p>
<code>options.zoomMargin</code>	0	<p>Тип: Number Number[]</p> <p>Отступы от границ видимой области карты, которые соблюдаются при приближении карты после клика на кластере. Рекомендуется устанавливать значение опции в соответствии с размером иконок кластеров и меток. Например, если метка попадает в видимую область карты только нижним концом ножки, стоит выставить ненулевой отступ <code>top</code>, чтобы метка оставалась полностью видна после того, как кластер распался. Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы <code>top</code>, <code>right</code>, <code>bottom</code>, <code>left</code>.</p>

Пример:

```
// Создание кластеризатора
// Создание карты, требующей кластеризации геообъектов
var map = new ymaps.Map('mapsID', { center: [56.034, 36.992], zoom: 8 });
// Создание массива геообъектов
myGeoObjects = [];
myGeoObjects[0] = new ymaps.GeoObject({
  geometry: { type: "Point", coordinates: [56.034, 36.992] },
  properties: {
    clusterCaption: 'Geo object №1',
    balloonContentBody: 'Balloon content for geo object №1'
  }
});
myGeoObjects[1] = new ymaps.GeoObject({
  geometry: { type: "Point", coordinates: [56.021, 36.983] },
  properties: {
    clusterCaption: 'Geo object №2',
    balloonContentBody: 'Balloon content for geo object №2'
  }
});
// Создание кластера и запрещение масштабирования карты при щелчке по кластеру
var clusterer = new ymaps.Clusterer({ clusterDisableClickZoom: true });
clusterer.add(myGeoObjects);
map.geoObjects.add(clusterer);
```

Поля

Имя	Тип	Описание
<code>balloon</code>	<code>clusterer.Balloon</code>	Балун кластеризатора.

Имя	Тип	Описание
balloonclose		<p>Заккрытие балуна.</p> <ul style="list-style-type: none"> target - ссылка на менеджер балуна кластеризатора; cluster - ссылка на объект-кластер. <p>Экземпляр класса Event.</p>
balloonopen		<p>Открытие балуна на кластере. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> target - ссылка на менеджер балуна кластеризатора; cluster - ссылка на объект-кластер. <p>Экземпляр класса Event.</p>
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
hint	clusterer.hint	Хинт кластеризатора.
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>

События

Имя	Описание
hintclose	Заккрытие хинта. Экземпляр класса Event .
hintopen	Открытие хинта на кластере. Экземпляр класса Event .
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. <p>Унаследовано от IChild.</p>

Методы

Имя	Возвращает	Описание
add(objects)	Clusterer	Добавляет геообъект или массив геообъектов в кластеризатор.
createCluster(center, geoObjects)	IGeoObject	Функция для создания кластера кластеризатором. Вызывается кластеризатором непосредственно в процессе кластеризации. Принимает на вход центр кластера и массив геообъектов, попадающих в данный кластер. Возвращает кластер, который потом будет добавлен на карту. Если требуется, чтобы кластеризатор создавал пользовательские объекты-кластеры, следует переопределить этот метод у кластеризатора.
getBounds()	Number[][] null	Возвращает географические координаты прямоугольной области, охватывающей все элементы кластера.
getClusters()	IGeoObject[]	Метод для получения текущего массива объектов-кластеров. Обратите внимание, что объекты-кластеры меняются при изменении масштаба карты или при сдвиге центра карты. Если необходимо выполнять операции над всеми кластерами, корректнее переопределить метод createCluster и добавлять необходимые операции в его вызов.
getGeoObjects()	IGeoObject[]	Возвращает массив геообъектов, добавленных в кластеризатор.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getObjectState(geoObject)	Object	Функция для получения информации о текущем состоянии объекта, добавленного в кластеризатор.

Имя	Возвращает	Описание
<code>getParent()</code>	<code>IParentOnMap</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>remove(objects)</code>	<code>Clusterer</code>	Удаляет геообъекты из кластеризатора.
<code>removeAll()</code>	<code>Clusterer</code>	Удаляет все геообъекты из кластеризатора.
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание полей

balloon

```
{clusterer.Balloon} balloon
```

Балун кластеризатора.

balloonclose

```
balloonclose
```

Закрытие балуна.

- `target` - ссылка на менеджер балуна кластеризатора;
- `cluster` - ссылка на объект-кластер.

Экземпляр класса Event.

balloonopen

```
balloonopen
```

Открытие балуна на кластере. Имена полей, доступных через метод [Event.get](#):

- `target` - ссылка на менеджер балуна кластеризатора;
- `cluster` - ссылка на объект-кластер.

Экземпляр класса Event.

Пример:

```
clusterer.events.add('balloonopen', function (e) {
    var clusterPlacemark = e.get('cluster');
});
```

hint

```
{clusterer.hint} hint
```

Хинт кластеризатора.

Описание событий**hintclose**

Закрытие хинта. Экземпляр класса [Event](#).

hintopen

Открытие хинта на кластере. Экземпляр класса [Event](#).

Описание методов**add**

```
{Clusterer} add(objects)
```

Добавляет геообъект или массив геообъектов в кластеризатор.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
objects *	—	Тип: IGeoObject IGeoObject[] Массив геообъектов или единичный геообъект.

* Обязательный параметр/опция.

Пример:

```
var myPlacemark = new ymaps.Placemark([35, 21]);
clusterer.add(myPlacemark);
var placemarks = [
  new ymaps.Placemark([44, 55]),
  new ymaps.Placemark([34, 45])
];
clusterer.add(placemarks);
```

createCluster

```
{IGeoObject} createCluster(center, geoObjects)
```

Функция для создания кластера кластеризатором. Вызывается кластеризатором непосредственно в процессе кластеризации. Принимает на вход центр кластера и массив геообъектов, попадающих в данный кластер. Возвращает кластер, который потом будет добавлен на карту. Если требуется, чтобы кластеризатор создавал пользовательские объекты-кластеры, следует переопределить этот метод у кластеризатора.

Возвращает объект-кластер. По умолчанию создает экземпляры класса [ClusterPlacemark](#).

Параметры:

Параметр	Значение по умолчанию	Описание
<code>center *</code>	—	Тип: <code>Number[]</code> Центр кластера в геокоординатах.
<code>geoObjects *</code>	—	Тип: <code>IGeoObject[]</code> Массив плейсмарков в кластере.

* Обязательный параметр/опция.

Пример:

```
// Зададим всплывающую подсказку кластерам в зависимости от состава.
var clusterer = new ymaps.Clusterer();
clusterer.createCluster = function (center, geoObjects) {
    // Создаем метку-кластер с помощью стандартной реализации метода.
    var clusterPlacemark = ymaps.Clusterer.prototype.createCluster.call(this, center, geoObjects),
        geoObjectsLength = clusterPlacemark.getGeoObjects().length,
        hintContent;
    if (geoObjectsLength < 10) {
        hintContent = 'Мало меток';
    } else if (geoObjectsLength < 100) {
        hintContent = 'Нормально так меток';
    } else {
        hintContent = 'Меток навалом';
    }
    clusterPlacemark.properties.set('hintContent', hintContent);
    return clusterPlacemark;
};
```

getBounds

```
{Number[][]|null} getBounds()
```

Возвращает географические координаты прямоугольной области, охватывающей все элементы кластера.

Пример:

```
var clusterer = new ymaps.Clusterer();
clusterer.add(myPlacemarks);
map.setBounds(clusterer.getBounds());
map.geoObjects.add(clusterer);
```

getClusters

```
{IGeoObject[]} getClusters()
```

Метод для получения текущего массива объектов-кластеров. Обратите внимание, что объекты-кластеры меняются при изменении масштаба карты или при сдвиге центра карты. Если необходимо выполнять операции над всеми кластерами, корректнее переопределить метод `createCluster` и добавлять необходимые операции в его вызов.

Возвращает массив кластеров, добавленных на карту в данный момент.

Пример:

```
map.events.add('boundschange', function () {
    var clusters = clusterer.getClusters();
    alert('Карта была перемещена и сейчас показывается ' + clusters.length + ' кластеров.'));
```

getGeoObjects

```
{IGeoObject[]} getGeoObjects()
```

Возвращает массив геообъектов, добавленных в кластеризатор.

Пример:

```
// Посчитаем количество объектов, попавших в видимую область карты.
var geoObjects = clusterer.getGeoObjects(),
    shownObjectsCounter = 0;
for (var i = 0, l = geoObjects.length; i < l; i++) {
    if (clusterer.getObjectState(geoObjects[i]).isShown) {
        shownObjectsCounter++;
    }
}
alert('Сейчас на карте показаны ' + shownObjectsCounter + ' меток из ' + geoObjects.length + '.');
```

getObjectState

```
{Object} getObjectState(geoObject)
```

Функция для получения информации о текущем состоянии объекта, добавленного в кластеризатор.

Возвращает объект, содержащий следующие поля:

- isShown - признак, указывающий, находится ли объект в видимой области карты;
- cluster - ссылка на кластер, в который добавлен объект;
- isClustered - признак, указывающий, попал ли объект в состав кластера.

Параметры:

Параметр	Значение по умолчанию	Описание
geoObject *	—	Тип: IGeoObject Геообъект, для которого нужно получить состояние.

* Обязательный параметр/опция.

Пример:

```
// Открытие балуна кластера с выбранным объектом.
// Получим данные о состоянии объекта внутри кластера.
var geoObjectState = clusterer.getObjectState(myGeoObjects[1]);
// Проверяем, находится ли объект в видимой области карты.
if (geoObjectState.isShown) {
    // Если объект попадает в кластер, открываем балун кластера с нужным выбранным объектом.
    if (geoObjectState.isClustered) {
        geoObjectState.cluster.state.set('activeObject', myGeoObjects[1]);
        clusterer.balloon.open(geoObjectState.cluster);
    } else {
        // Если объект не попал в кластер, открываем его собственный балун.
        myGeoObjects[1].balloon.open();
    }
}
```

remove

```
{Clusterer} remove(objects)
```

Удаляет геообъекты из кластеризатора.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
objects *	—	Тип: IGeoObject IGeoObject[] Массив геообъектов.

* Обязательный параметр/опция.

Пример:

```
var myPlacemark = new ymaps.Placemark([35, 21],
    placemarks = [
        new ymaps.Placemark([44, 55]),
        new ymaps.Placemark([34, 45]),
        ...
    ];
clusterer.add(myPlacemark).add(placemarks);
...
// Удалим первые 10 объектов массива из кластеризатора.
clusterer.remove(placemarks.slice(0, 10));
```

removeAll

```
{Clusterer} removeAll()
```

Удаляет все геообъекты из кластеризатора.

Возвращает ссылку на себя.

Пример:

```
clusterer.add(placemark1);
clusterer.add(placemark2);
clusterer.removeAll();
```

ClusterPlacemark

Расширяет [IGeoObject](#), [collection.Item](#).

Кластер геообъектов. Используется по умолчанию в [Clusterer](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
ClusterPlacemark(geometry, properties[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: Number[] Object IPointGeometry Координаты метки или хэш с описанием геометрии, или ссылка на объект точечной геометрии.
properties *	—	Тип: IDataManager Данные кластера.
properties.geoObjects *	—	Тип: IGeoObject[] Массив геообъектов, которые находятся в данном кластере.

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции кластера. Помимо частных опций, балун кластера поддерживает те же опции, что и Balloon . Опции для балуна кластера указываются с префиксом 'balloon'.

Параметр	Значение по умолчанию	Описание
<code>options.balloonContentLayout</code>	<code>'cluster#balloonTwoColumns'</code>	<p>Тип: Function String</p> <p>Макет балуна кластера в обычном режиме. Можно передать конструктор объекта с интерфейсом <code>ILayout</code> или ключ одного из стандартных макетов. Каждый стандартный макет имеет некоторые собственные опции. В стандартных макетах по умолчанию используются поля данных геообъектов: <code>clusterCaption</code>, <code>balloonContentHeader</code>, <code>balloonContent</code> и <code>balloonContentFooter</code> (см. Placemark).</p> <ul style="list-style-type: none"> <code>'cluster#balloonTwoColumns'</code> - Макет с двумя колонками. В левой колонке находится список названий меток (поле данных метки <code>clusterCaption</code> или <code>balloonContentHeader</code>, если предыдущее поле не определено). В правой колонке располагается вся информация о геообъекте. Опции макета: <ul style="list-style-type: none"> <code>balloonLeftColumnWidth</code> — ширина поля со списком объектов в балуне кластера в пикселях. Значение по умолчанию: 125 <code>'cluster#balloonCarousel'</code> - Информация о геообъекте располагается в центре. По бокам размещаются кнопки перехода к предыдущему и следующему геообъекту. В нижней части балуна находится меню быстрой навигации. Опции макета: <ul style="list-style-type: none"> <code>balloonCycling</code> — цикличность списка при навигации боковыми стрелками. Значение по умолчанию: <code>true</code> <code>balloonPagerSize</code> — количество элементов навигации в нижней панели. Значение по умолчанию зависит от типа устройства. Для мобильных телефонов — это 4, а для планшетов и персональных компьютеров — 9. <code>balloonPagerType</code> — тип нижней панели навигации. Может принимать значения <code>'numeric'</code> и <code>'marker'</code>. <ul style="list-style-type: none"> <code>numeric</code> — отображение номера геообъекта в списке; <code>marker</code> — отображение маркеров без номеров. Рекомендуется использовать при количестве элементов в кластере меньше или равном значению <code>options.balloonPagerSize</code>.

Параметр	Значение по умолчанию	Описание
<code>options.balloonContentLayoutHeight</code>	—	<p>Тип: Number</p> <p>Высота макета контента балуна кластера. По умолчанию опция балуна <code>balloonMaxHeight</code> для кластера не установлена, так как все стандартные макеты балунов кластеров имеют определенные размеры. Стандартное значение зависит от макета.</p> <ul style="list-style-type: none"> 'cluster#balloonTwoColumns' - 210 пикселей 'cluster#balloonCarousel' - 177 пикселей 'cluster#balloonAccordion' - 283 пикселей
<code>options.balloonContentLayoutWidth</code>	—	<p>Тип: Number</p> <p>Ширина макета контента балуна кластера. По умолчанию опция балуна <code>balloonMaxWidth</code> для кластера не установлена, так как все стандартные макеты балунов кластеров имеют определенные размеры. Стандартное значение зависит от макета.</p> <ul style="list-style-type: none"> 'cluster#balloonTwoColumns' - 475 пикселей 'cluster#balloonCarousel' - 308 пикселей 'cluster#balloonAccordion' - 305 пикселей <p>Опция не используется в режиме панели.</p>

Параметр	Значение по умолчанию	Описание
<code>options.balloonItemContentLayout</code>	—	<p>Тип: <code>ILayout String</code></p> <p>Макет с информацией о геообъекте. Стандартное значение зависит от макета.</p> <ul style="list-style-type: none"> В макете <code>'cluster#balloonTwoColumns'</code> отображается справа от списка. Стандартное значение <code>'cluster#balloonTwoColumnsItemContent'</code>. В макете <code>'cluster#balloonCarousel'</code> отображается в центре. Стандартно значение <code>'cluster#balloonCarouselItemContent'</code>. В макете <code>'cluster#balloonAccordion'</code> отображает после клика по элементу списка. Стандартно значение <code>'cluster#balloonAccordionItemContent'</code>. <p>Набор полей, которые поступают в данный макет, отличаются от родительского и соответствует полям, которые поступают в обычный макет балуна геообъекта. Еще были добавлены поля <code>ownerProperties</code>, <code>ownerOptions</code> и <code>ownerState</code> для доступа к данным кластера.</p>
<code>options.balloonPanelContentLayout</code>	<code>null</code>	<p>Тип: <code>Function String</code></p> <p>Макет балуна кластера в режиме "панель". Можно передать конструктор объекта с интерфейсом <code>ILayout</code>. Доступные значения такие же, как и у опции <code>'balloonContentLayout'</code>. Если значение равно <code>null</code>, то применяется значение опции <code>'balloonContentLayout'</code>.</p>
<code>options.cursor</code>	<code>'pointer'</code>	<p>Тип: <code>String</code></p> <p>Курсор над меткой кластера.</p>

Параметр	Значение по умолчанию	Описание
<code>options.disableClickZoom</code>	false	Тип: Boolean Флаг, запрещающий увеличение коэффициента масштабирования карты при клике на кластер.
<code>options.hideIconOnBalloonOpen</code>	true	Тип: Boolean Скрывать иконку при открытии балуна.
<code>options.iconColor</code>	—	Тип: String Цвет иконки кластера. Эта опция применяется для стандартных иконок в браузерах, поддерживающих SVG.
<code>options.iconContentLayout</code>	'cluster#iconContent'	Тип: Function String Макет содержимого метки кластера. (Тип: конструктор объекта с интерфейсом ILayout или ключ макета). Если у метки не нужно отображать содержимое, значение опции нужно выставить в null.
<code>options.iconLayout</code>	'cluster#icon'	Тип: Function String Макет метки кластера (Тип: конструктор объекта с интерфейсом ILayout или ключ макета).

Параметр	Значение по умолчанию	Описание
options.icons	—	<p>Тип: Object[]</p> <p>Массив, описывающий иконки для стандартной реализации кластера. Описание иконки представляет из себя объект с полями</p> <ul style="list-style-type: none"> • href - ссылка на картинку; • size - массив из двух чисел - размер иконки в пикселях; • offset - смещение иконки относительно точки привязки объекта; • shape - необязательное поле. Объект, реализующий интерфейс IShape или JSON-описание геометрии. Позволяет задавать описание геометрии иконки. Если параметр отсутствует, то активной для событий (наведение мыши, клик) будет считаться прямоугольная область вокруг иконки.
options.iconShape	—	<p>Тип: IGeometryJson null</p> <p>Фигура активной области метки кластера. Задается в виде JSON-описания пиксельной геометрии иконки. Эту опцию нужно использовать при создании своих HTML макетов. Координаты геометрии фигуры отсчитываются от точки привязки.</p>
options.interactivityModel	'default#geoObject'	<p>Тип: String</p> <p>Модель интерактивности кластера. Доступные ключи и их значения перечислены в описании interactivityModel.storage.</p>
options.numbers	[10, 100]	<p>Тип: Number[]</p> <p>Массив, описывающий граничные значения для размеров кластеров. Количество иконок, описанных в опции "icons", должно быть на 1 больше, чем чисел в данном массиве.</p>

Параметр	Значение по умолчанию	Описание
<code>options.openBalloonOnClick</code>	<code>true</code>	Тип: Boolean Опция, позволяющая запретить открытие балуна при клике на кластере. По умолчанию открытие балуна разрешено;
<code>options.openEmptyHint</code>	<code>false</code>	Тип: Boolean Опция, включающая показ пустых всплывающих подсказок. По умолчанию пустые хинты кластера не показываются.
<code>options.openHintOnHover</code>	<code>true</code>	Тип: Boolean Опция, позволяющая запретить показ всплывающей подсказки при наведении на кластер. По умолчанию показ хинтов разрешен.
<code>options.zIndexHover</code>	—	Тип: Number Значение <code>zIndex</code> , которое выставляется метке кластера при наведении.

* Обязательный параметр/опция.

Примеры:

1.

```
// Переменная с описанием двух видов иконок кластеров.
var clusterIcons = [
  {
    href: 'small.png',
    size: [40, 40],
    // Отступ, чтобы центр картинки совпадал с центром кластера.
    offset: [-20, -20]
  },
  {
    href: 'big.png',
    size: [60, 60],
    offset: [-30, -30],
    // Можно задать геометрию активной области метки.
    // Если геометрия не задана, активной областью будет
    // прямоугольник.
    shape: {
      type: 'Circle',
      coordinates: [0, 0],
      radius: 30
    }
  }
],
// При размере кластера до 100 будет использована 'small.png'.
// При размере кластера больше 100 будет использована 'big.png'.
clusterNumbers = [100],
// Сделаем макет содержимого иконки кластера,
// в котором цифры будут раскрашены в белый цвет.
MyIconContentLayout = ymaps.templateLayoutFactory.createClass(
  '<div style="color: #FFFFFF; font-weight: bold;">{{ properties.geoObjects.length }}</div>'
),
var clusterer = new ymaps.Clusterer({
  // Если опции для кластеров задаются через кластеризатор,
  // необходимо указывать их с префиксами "cluster".
  clusterIcons: clusterIcons,
  clusterNumbers: clusterNumbers,
  clusterIconContentLayout: MyIconContentLayout
});
```

2.

```
// Создание кластеризатора с макетом-каруселью.
var clusterer = new ymaps.Clusterer({
    clusterDisableClickZoom: true,
    // Используем макет "карусель"
    clusterBalloonContentLayout: "cluster#balloonCarousel",
    // Запрещаем закичивание списка при постраничной навигации.
    clusterBalloonCycling: false,
    // Настройка внешнего вида панели навигации.
    // Элементами панели навигации будут маркеры.
    clusterBalloonPagerType: "marker",
    // Количество элементов в панели.
    clusterBalloonPagerSize: 6
});
```

3.

```
// Создание кластеризатора с макетом-аккордеоном
var clusterer = new ymaps.Clusterer({
    clusterDisableClickZoom: true,
    // Используем макет "аккордеон"
    clusterBalloonContentLayout: "cluster#balloonAccordion"
});
```

4.

```
// Создание кластеризатора с произвольным HTML-макетом иконки кластера.
var clusterer = new ymaps.Clusterer({
    // Зададим макет метки кластера.
    clusterIconLayout: ymaps.templateLayoutFactory.createClass('<div
class="clusterIcon">{{ properties.geoObjects.length }}</div>'),
    // Чтобы метка была кликабельной, переопределим ее активную область.
    clusterIconShape: {
        type: 'Rectangle',
        coordinates: [[0, 0], [20, 20]]
    }
});
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	data.Manager	Состояние кластера. Определяется следующими полями: <ul style="list-style-type: none"> <code>activeObject</code> - Ссылка на активный объект кластера. Активным объектом является тот, который в данный момент выбран в балуне кластера.

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code>.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getBounds()	<code>Number[][] null</code>	<p>Возвращает географические координаты прямоугольной области, охватывающей все элементы кластера.</p>
getGeoObjects()	IGeoObject[]	<p>Метод является упрощенным вызовом <code>cluster.properties.get('geoObjects')</code>;</p>
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.</p> <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	<p>Метод предоставляет синхронный доступ к оверлею.</p> <p>Унаследован от IGeoObject.</p>

Имя	Возвращает	Описание
<code>getParent()</code>	<code>IParentOnMap</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>onAddToMap(map)</code>		Функция, которая вызывается при добавлении элемента на карту. Для выполнения дополнительных действий при добавлении объекта на карту переопределите эту функцию. Унаследован от <code>collection.Item</code> .
<code>onRemoveFromMap(oldMap)</code>		Функция, которая вызывается при удалении элемента с карты. Для выполнения дополнительных действий при удалении объекта с карты переопределите эту функцию. Унаследован от <code>collection.Item</code> .
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание полей

state

```
{data.Manager} state
```

Состояние кластера. Определяется следующими полями:

- `activeObject` - Ссылка на активный объект кластера. Активным объектом является тот, который в данный момент выбран в балуне кластера.

Пример:

```
var geoObjects = cluster.properties.get('geoObjects');
// При открытии балуна кластера будет выбран третий объект в списке.
cluster.state.set('activeObject', geoObjects[2]);
```

Описание методов

getBounds

```
{Number[][]|null} getBounds()
```

Возвращает географические координаты прямоугольной области, охватывающей все элементы кластера.

getGeoObjects

```
{IGeoObject[]} getGeoObjects()
```

Метод является упрощенным вызовом `cluster.properties.get('geoObjects')`;

Возвращает массив геообъектов, образующих кластер.

Collection

Расширяет [ICollection](#), [collection.Item](#).

Базовая реализация коллекции объектов на карте.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Collection([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции коллекции.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
add	Произошло добавление дочернего объекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"><code>child</code> - добавленный дочерний элемент.

Имя	Описание
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. <p>Унаследовано от IChild.</p>
remove	<p>Произошло удаление дочернего объекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> child - удаленный дочерний элемент.

Методы

Имя	Возвращает	Описание
add(child)	Collection	Добавляет элемент в коллекцию.
each(callback[, context])	Collection	Вызывает функцию-обработчик для всех элементов коллекции.
filter(filterFunction)	Object[]	Вызывает функцию-фильтр для всех элементов коллекции. При возвращении фильтром ненулевого значения элемент коллекции попадает в конечный массив.
get(index)	Object	Возвращает элемент коллекции или null, если номер находится вне диапазона номеров коллекции.
getAll()	Object[]	Возвращает массив со всеми элементами коллекции.
getIterator()	Iterator	Возвращает итератор для перебора элементов коллекции.

Имя	Возвращает	Описание
getLength()	Number	Возвращает количество элементов коллекции.
getMap()	Map	Возвращает карту, которой принадлежит элемент коллекции. Унаследован от collection.Item .
getParent()	IParentOnMap	Возвращает родительский объект. Унаследован от collection.Item .
indexOf(childToFind)	Number	Возвращает порядковый номер объекта в коллекции либо -1, если объект не найден.
onAddToMap(map)		Функция, которая вызывается при добавлении элемента на карту. Для выполнения дополнительных действий при добавлении объекта на карту переопределите эту функцию. Унаследован от collection.Item .
onRemoveFromMap(oldMap)		Функция, которая вызывается при удалении элемента с карты. Для выполнения дополнительных действий при удалении объекта с карты переопределите эту функцию. Унаследован от collection.Item .
remove(child)	Collection	Удаляет элемент из коллекции.
removeAll()	Collection	Удаляет все элементы коллекции.
setParent(parent)	collection.Item	Устанавливает родителя для заданного элемента коллекции. Унаследован от collection.Item .

Описание событий

add

Произошло добавление дочернего объекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `child` - добавленный дочерний элемент.

remove

Произошло удаление дочернего объекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `child` - удаленный дочерний элемент.

Описание методов

add

```
{Collection} add(child)
```

Добавляет элемент в коллекцию.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>child *</code>	—	Тип: collection.Item Элемент, который необходимо добавить.

* Обязательный параметр/опция.

each

```
{Collection} each(callback[, context])
```

Вызывает функцию-обработчик для всех элементов коллекции.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>callback *</code>	—	Тип: Function Функция-обработчик. На вход получает элемент коллекции. При возвращении функцией значения <code>false</code> перебор останавливается.
<code>context</code>	—	Тип: Object Контекст вызываемой функции.

* Обязательный параметр/опция.

filter

```
{Object[]} filter(filterFunction)
```

Вызывает функцию-фильтр для всех элементов коллекции. При возвращении фильтром ненулевого значения элемент коллекции попадает в конечный массив.

Возвращает массив из элементов, которые попали в выборку.

Параметры:

Параметр	Значение по умолчанию	Описание
filterFunction *	—	Тип: Function Функция, которая является фильтром для объектов коллекции. Первым параметром принимает элемент из коллекции. Должна возвращать булево значение.

* Обязательный параметр/опция.

get

```
{Object} get(index)
```

Возвращает элемент коллекции или null, если номер находится вне диапазона номеров коллекции.

Параметры:

Параметр	Значение по умолчанию	Описание
index *	—	Тип: Number Порядковый номер элемента в коллекции.

* Обязательный параметр/опция.

getAll

```
{Object[]} getAll()
```

Возвращает массив со всеми элементами коллекции.

getIterator

```
{Iterator} getIterator()
```

Возвращает итератор для перебора элементов коллекции.

getLength

```
{Number} getLength()
```

Возвращает количество элементов коллекции.

indexOf

```
{Number} indexOf(childToFind)
```

Возвращает порядковый номер объекта в коллекции либо -1, если объект не найден.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>childToFind</code> *	—	Тип: Object Искомый объект.

* Обязательный параметр/опция.

remove

```
{Collection} remove(child)
```

Удаляет элемент из коллекции.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>child</code> *	—	Тип: <code>collection.Item</code> Элемент, который необходимо удалить.

* Обязательный параметр/опция.

removeAll

```
{Collection} removeAll()
```

Удаляет все элементы коллекции.

Возвращает ссылку на себя.

collection

collection.Item

Расширяет [IChildOnMap](#), [ICustomizable](#), [IEventEmitter](#), [IParentOnMap](#).

Базовый класс элемента коллекции объектов карты.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
collection.Item([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции объекта.

Пример:

```
// Пример реализации собственного элемента управления на основе наследования от collection.Item.
// Элемент управления отображает имя объекта, расположенного в центре карты.
var map = new ymaps.Map('map', {
  center: [55.819543, 37.611619],
  zoom: 6
});
// Создание пользовательского класса.
var CustomControl = function (options) {
  CustomControl.superclass.constructor.call(this, options);
};
// И наследование от collection.Item.
ymaps.util.defineClass(CustomControl, ymaps.collection.Item, {
  onAddToMap: function (map) {
    CustomControl.superclass.onAddToMap.call(this, map);
    // Создание HTML-элемента с текстом.
    this.getParent().getChildElement(this).then(this._onChildElementGet, this);
  },
  onRemoveFromMap: function (oldMap) {
    CustomControl.superclass.onRemoveFromMap.call(this, oldMap);
  },
  _onChildElementGet: function (parentElementContainer) {
    // Вы можете создать представление DOM для элемента управления здесь
    // и добавьте его в качестве дочернего элемента в parentElementContainer.
    // ...
  }
});
var customControl = new CustomControl();
map.controls.add(customControl, {top: 10, left: 10});
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IParentOnMap .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. <p>Унаследовано от IChild.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает карту, которой принадлежит элемент коллекции.
getParent()	IParentOnMap	Возвращает родительский объект.
onAddToMap(map)		Функция, которая вызывается при добавлении элемента на карту. Для выполнения дополнительных действий при добавлении объекта на карту переопределите эту функцию.
onRemoveFromMap(oldMap)		Функция, которая вызывается при удалении элемента с карты. Для выполнения дополнительных действий при удалении объекта с карты переопределите эту функцию.
setParent(parent)	collection.Item	Устанавливает родителя для заданного элемента коллекции.

Описание методов

getMap

```
{Map} getMap()
```

Возвращает карту, которой принадлежит элемент коллекции.

getParent

```
{IParentOnMap} getParent()
```

Возвращает родительский объект.

onAddToMap

```
{ } onAddToMap(map)
```

Функция, которая вызывается при добавлении элемента на карту. Для выполнения дополнительных действий при добавлении объекта на карту переопределите эту функцию.

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Карта, на которую добавлен объект.

* Обязательный параметр/опция.

onRemoveFromMap

```
{ } onRemoveFromMap(oldMap)
```

Функция, которая вызывается при удалении элемента с карты. Для выполнения дополнительных действий при удалении объекта с карты переопределите эту функцию.

Параметры:

Параметр	Значение по умолчанию	Описание
oldMap *	—	Тип: Map Карта, с которой удален объект.

* Обязательный параметр/опция.

setParent

```
{collection.Item} setParent(parent)
```

Устанавливает родителя для заданного элемента коллекции.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
parent *	—	Тип: IParentOnMap Родительский объект.

* Обязательный параметр/опция.

control

control.Button

Расширяет [ICustomizable](#), [ISelectableControl](#).

Элемент управления "Кнопка". Стандартный макет кнопки изменяет свой внешний вид в зависимости от размеров карты. Если карта имеет большую ширину, то кнопка принимает вид "картинка + текст". Если карта среднего размера, кнопка принимает вид "текст". Если карта небольшого размера, то в макете кнопки отображается только иконка. Если для кнопки не задана иконка, то во всех состояниях будет отображаться текст и наоборот.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.Button([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object string Параметры кнопки или строка - содержимое кнопки в виде HTML.
parameters.data	—	Тип: Object Данные кнопки.
parameters.data.content	—	Тип: String Содержимое кнопки в виде HTML.
parameters.data.image	—	Тип: String URL иконки кнопки. Стандартный макет кнопки рассчитан на иконку размером 16x16 пикселей.
parameters.data.title	—	Тип: String Текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.
parameters.options	—	Тип: Object Опции кнопки.
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли кнопка свои размеры в менеджере отступов карты map.margin.Manager .

Параметр	Значение по умолчанию	Описание
parameters.options.float	"right"	<p>Тип: String</p> <p>Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируется только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.</p>
parameters.options.floatIndex	0	<p>Тип: Number</p> <p>Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".</p>

Параметр	Значение по умолчанию	Описание
parameters.options.layout	—	<p>Тип: ISelectableControlLayout String</p> <p>Конструктор макета элемента управления, реализующий интерфейс ISelectableControlLayout или ключ макета в хранилище layout.storage. В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> control - ссылка на элемент управления; options - менеджер опций элемента управления control.Button.options; data - менеджер данных элемента управления control.Button.data; state - менеджер состояния элемента управления control.Button.state. <p>Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует на интерфейсные события макета и меняет значения полей control.Button.state в зависимости от полученных команд.</p>
parameters.options.maxWidth	90	<p>Тип: Number Number[]</p> <p>Максимальная ширина кнопки в различных состояниях. Если задано число, то считается, что кнопка имеет одинаковые максимальные размеры во всех состояниях. Если задан массив, то он будет трактоваться как максимальная ширина кнопки в различных состояниях - от меньшего к большему. Количество доступных состояний задается в экземпляре класса control.Manager через опцию states. Этот класс обычно является полем Map.controls. По умолчанию у элементов управления есть три состояния - ['small', 'medium', 'large'].</p>

Параметр	Значение по умолчанию	Описание
parameters.options.position	—	<p>Тип: Object</p> <p>Объект, описывающий позицию элемента управления. При указании опции position значение опции float автоматически трактуется как "none".</p>
parameters.options.position.bottom	'auto'	<p>Тип: Number String</p> <p>Положение относительно нижнего края карты.</p>
parameters.options.position.left	'auto'	<p>Тип: Number String</p> <p>Положение относительно левого края карты.</p>
parameters.options.position.right	'auto'	<p>Тип: Number String</p> <p>Положение относительно правого края карты.</p>
parameters.options.position.top	'auto'	<p>Тип: Number String</p> <p>Положение относительно верхнего края карты.</p>
parameters.options.selectOnClick	true	<p>Тип: Boolean</p> <p>Опция, описывающая поведение кнопки.</p> <ul style="list-style-type: none"> • true — кнопка становится "нажатой" после клика, то есть меняется ее внешний вид и значение поля control.Button.state устанавливается в 'selected'; • false - кнопка не меняет свой внешний вид и состояние после клика на нее.

Параметр	Значение по умолчанию	Описание
<code>parameters.options.size</code>	'auto'	<p>Тип: String</p> <p>Параметр, отвечающий за внешний вид стандартного макета кнопки. Может принимать значения:</p> <ul style="list-style-type: none"> 'auto' - стандартный макет кнопки автоматически изменяет свой внешний вид в зависимости от размеров карты и от количества добавленных элементов управления; 'small' - в макете кнопки отображается иконка, независимо от размера карты; 'medium' - в макете кнопки отображается только текст, независимо от размера карты; 'large' - в макете кнопки всегда отображается иконка и текст, независимо от размера карты.
<code>parameters.options.visible</code>	true	<p>Тип: Boolean</p> <p>Признак того, что элемент управления отображается.</p>
<code>parameters.state</code>	—	<p>Тип: Object</p> <p>Объект, описывающий состояние кнопки.</p>
<code>parameters.state.enabled</code>	true	<p>Тип: Boolean</p> <p>Признак, активна ли кнопка.</p>
<code>parameters.state.selected</code>	false	<p>Тип: Boolean</p> <p>Признак, нажата ли кнопка.</p>

Примеры:**1.**

```
// Пример 1.
// Создадим кнопку и добавим ее на карту.
```

```

var button = new ymaps.control.Button({
  data: {
    // Зададим иконку для кнопки.
    image: 'images/button.jpg',
    // Текст на кнопке.
    content: 'Save',
    // Текст всплывающей подсказки.
    title: 'Click to save the route'
  },
  options: {
    // Зададим опции кнопки.
    selectOnClick: false,
    // У кнопки будет три состояния: иконка, текст и текст + иконка.
    // Зададим три значения ширины кнопки для всех состояний.
    maxWidth: [30, 100, 150]
  }
});
map.controls.add(button, { float: 'right', floatIndex: 100 });

```

2.

```

// Пример 2
// Создание кнопки с пользовательским макетом.
var button = new ymaps.control.Button({
  data: {
    content: 'Red button',
    title: 'Press the button'
  },
  options: {
    layout: ymaps.templateLayoutFactory.createClass(
      // Если кнопка не нажата, применяется CSS стиль 'myButton'.
      // Если кнопка нажата, к ней применяются CSS-стили 'myButton' и 'myButtonSelected'.

      "<div class='myButton {% if state.selected %}myButtonSelected{% endif %}' title='{{ data.title }}'>" +
      "{{ data.content }}" +
      "</div>"
    ),
    // Чтобы другие элементы управления корректно позиционировались по горизонтали,
    // нужно обязательно задать максимальную ширину для макета.
    maxWidth: 150
  }
});
map.controls.add(button, { float: 'left', floatIndex: 0 });

// Можно задать позиционирование относительно краев карты. В этом случае
// значение опции maxWidth не влияет на позиционирование
// элементов управления.
map.controls.add(button, { float: 'none', position: {left: '5px', top: '5px'} });

```

Поля

Имя	Тип	Описание
data	data.Manager	<p>Данные кнопки. Имена полей, доступных через метод <code>data.Manager.get</code>:</p> <ul style="list-style-type: none"> <code>image</code> - иконка кнопки, если есть; <code>content</code> - содержимое кнопки в виде HTML; <code>title</code> - текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от IControl.</p>
press		<p>Событие, сигнализирующее о том, что кнопка была нажата. В отличие от <code>click</code> генерируется только если состояние <code>isEnabled</code> == <code>true</code>. Экземпляр класса <code>Event</code>.</p>

Имя	Тип	Описание
state	data.Manager	<p>Состояние кнопки. Имена полей, доступных через метод <code>data.Manager.get</code>:</p> <ul style="list-style-type: none"> <code>selected</code> - признак, нажата ли кнопка; <code>enabled</code> - признак, активна ли кнопка. <code>size</code> - размер, который в данный момент выставлен кнопке;

События

Имя	Описание
click	Нажатие на кнопку. Экземпляр класса Event .
deselect	<p>Элемент управления не выбран.</p> <p>Унаследовано от ISelectableControl.</p>
disable	<p>Элемент управления стал недоступным.</p> <p>Унаследовано от ISelectableControl.</p>
enable	<p>Элемент управления стал доступным.</p> <p>Унаследовано от ISelectableControl.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
select	<p>Элемент управления выбран.</p> <p>Унаследовано от ISelectableControl.</p>

Методы

Имя	Возвращает	Описание
deselect()		<p>Отменяет выбор элемента управления (выключает).</p> <p>Унаследован от ISelectableControl.</p>
disable()		<p>Делает элемент управления недоступным (запрещены пользовательские действия).</p> <p>Унаследован от ISelectableControl.</p>

Имя	Возвращает	Описание
<code>enable()</code>		Делает элемент управления доступным (разрешены пользовательские действия). Унаследован от ISelectableControl .
<code>getMap()</code>	Map	Возвращает ссылку на карту.
<code>getParent()</code>	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
<code>isEnabled()</code>	Boolean	Возвращает true - элемент управления доступен, false - недоступен. Унаследован от ISelectableControl .
<code>isSelected()</code>	Boolean	Возвращает true - элемент управления выбран, false - не выбран. Унаследован от ISelectableControl .
<code>select()</code>		Выбирает (включает) элемент управления. Унаследован от ISelectableControl .
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание полей

data

```
{data.Manager} data
```

Данные кнопки. Имена полей, доступных через метод `data.Manager.get`:

- `image` - иконка кнопки, если есть;
- `content` - содержимое кнопки в виде HTML;
- `title` - текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.

press

```
press
```

Событие, сигнализирующее о том, что кнопка была нажата. В отличие от `click` генерируется только если состояние `isEnabled == true`. Экземпляр класса `Event`.

state

```
{data.Manager} state
```

Состояние кнопки. Имена полей, доступных через метод `data.Manager.get`:

- `selected` - признак, нажата ли кнопка;
- `enabled` - признак, активна ли кнопка.
- `size` - размер, который в данный момент выставлен кнопке;

Пример:

```
var button = new ymaps.control.Button('Редактировать');  
// Зададим состояние кнопки "нажата" -  
// аналогично вызову метода button.select();  
button.state.set('selected', true);
```

Описание событий**click**

Нажатие на кнопку. Экземпляр класса [Event](#).

Описание методов**getMap**

```
{Map} getMap()
```

Возвращает ссылку на карту.

control.FullscreenControl

Расширяет [control.Button](#).

Элемент управления "Полноэкранный режим". Задать свойство `z-index` контейнеру карты, находящейся в "полноэкранном режиме", можно с помощью опции `Map.options.fullscreenZIndex`. Ключ элемента управления в хранилище [control.storage](#) — `"fullscreenControl"`.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.FullscreenControl([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.
parameters.data	—	Тип: Object Объект, описывающий данные элемента управления.
parameters.data.title	—	Тип: String Текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.

Параметр	Значение по умолчанию	Описание
parameters.options	—	Тип: Object Опции элемента управления.
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли элемент управления свои размеры в менеджере отступов карты map.margin.Manager .
parameters.options.float	"right"	Тип: String Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируются только по значениям опций left, right, bottom, top относительно границ карты.
parameters.options.floatIndex	300	Тип: Number Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none". Также смотрите описание опции position.

Параметр	Значение по умолчанию	Описание
parameters.options.layout	—	<p>Тип: ISelectableControlLayout String</p> <p>Конструктор макета элемента управления, реализующий интерфейс ISelectableControlLayout или ключ макета в хранилище layout.storage. В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> control - ссылка на элемент управления; options — менеджер опций элемента управления control.FullscreenControl.options; data — менеджер данных элемента управления control.FullscreenControl.data; state - менеджер состояния элемента управления control.FullscreenControl.state. <p>Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует на интерфейсные события макета.</p>
parameters.options.maxWidth	28	<p>Тип: Number Number[]</p> <p>Максимальная ширина элемента управления в различных состояниях. Если задано число, то считается, что элемент управления имеет одинаковые максимальные размеры во всех состояниях. Если задан массив, то он будет трактоваться как максимальная ширина в различных состояниях - от меньшего к большему. Количество доступных состояний задается в экземпляре класса control.Manager через опцию states. Этот класс обычно является полем Map.controls. По умолчанию у элементов управления есть три состояния - ['small', 'medium', 'large']. По умолчанию элемент управления не изменяет своего размера, всегда имея вид кнопки с иконкой.</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.position</code>	—	Тип: Object Объект, описывающий позицию элемента управления. При указании опции <code>position</code> значение опции <code>float</code> автоматически трактуется как <code>"none"</code> .
<code>parameters.options.position.bottom</code>	<code>'auto'</code>	Тип: Number String Положение относительно нижнего края карты.
<code>parameters.options.position.left</code>	<code>'auto'</code>	Тип: Number String Положение относительно левого края карты.
<code>parameters.options.position.right</code>	<code>'auto'</code>	Тип: Number String Положение относительно правого края карты.
<code>parameters.options.position.top</code>	<code>'auto'</code>	Тип: Number String Положение относительно верхнего края карты.
<code>parameters.options.visible</code>	<code>true</code>	Тип: Boolean Признак того, что элемент управления отображается.
<code>parameters.state</code>	—	Тип: Object Объект, описывающий состояние элемента управления.
<code>parameters.state.enabled</code>	<code>true</code>	Тип: Boolean Признак, активна ли кнопка.
<code>parameters.state.selected</code>	<code>false</code>	Тип: Boolean Признак, нажата ли кнопка.

Пример:

```
// Добавим элемент управления на карту и сразу переведем
// её в «полноэкранный режим».
var fullscreenControl = new ymaps.control.FullscreenControl();
myMap.controls.add(fullscreenControl);
fullscreenControl.enterFullscreen();
```

Поля

Имя	Тип	Описание
data	data.Manager	<p>Данные кнопки. Имена полей, доступных через метод <code>data.Manager.get</code>:</p> <ul style="list-style-type: none"> <code>image</code> - иконка кнопки, если есть; <code>content</code> - содержимое кнопки в виде HTML; <code>title</code> - текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши. <p>Унаследовано от control.Button.</p>
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от IControl.</p>
press		<p>Событие, сигнализирующее о том, что кнопка была нажата. В отличие от <code>click</code> генерируется только если состояние <code>isEnabled</code> == <code>true</code>. Экземпляр класса <code>Event</code>.</p> <p>Унаследовано от control.Button.</p>
state	data.Manager	<p>Состояние элемента управления. Имена полей, доступных через метод <code>data.Manager.get</code>:</p> <ul style="list-style-type: none"> <code>fullscreen</code> — признак, находится ли карта в полноэкранном режиме.

События

Имя	Описание
click	<p>Нажатие на кнопку. Экземпляр класса Event.</p> <p>Унаследовано от control.Button.</p>
deselect	<p>Элемент управления не выбран.</p> <p>Унаследовано от ISelectableControl.</p>
disable	<p>Элемент управления стал недоступным.</p> <p>Унаследовано от ISelectableControl.</p>
enable	<p>Элемент управления стал доступным.</p> <p>Унаследовано от ISelectableControl.</p>
fullscreenenter	<p>Карта переключена в полноэкранный режим. Экземпляр класса Event.</p>
fullscreenexit	<p>Карта вышла из полноэкранного режима. Экземпляр класса Event.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
select	Элемент управления выбран. Унаследовано от ISelectableControl .

Методы

Имя	Возвращает	Описание
deselect()		Отменяет выбор элемента управления (выключает). Унаследован от ISelectableControl .
disable()		Делает элемент управления недоступным (запрещены пользовательские действия). Унаследован от ISelectableControl .
enable()		Делает элемент управления доступным (разрешены пользовательские действия). Унаследован от ISelectableControl .
enterFullscreen()		Позволяет переводить карту в полноэкранный режим.
exitFullscreen()		Позволяет вывести карту из полноэкранного режима.
getMap()	Map	Возвращает ссылку на карту. Унаследован от control.Button .
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .

Имя	Возвращает	Описание
<code>isEnabled()</code>	Boolean	Возвращает true - элемент управления доступен, false - недоступен. Унаследован от ISelectableControl .
<code>isSelected()</code>	Boolean	Возвращает true - элемент управления выбран, false - не выбран. Унаследован от ISelectableControl .
<code>select()</code>		Выбирает (включает) элемент управления. Унаследован от ISelectableControl .
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание полей

state

```
{data.Manager} state
```

Состояние элемента управления. Имена полей, доступных через метод `data.Manager.get`:

- `fullscreen` — признак, находится ли карта в полноэкранном режиме.

Описание событий

fullscreenenter

Карта переключена в полноэкранный режим. Экземпляр класса [Event](#).

fullscreenexit

Карта вышла из полноэкранного режима. Экземпляр класса [Event](#).

Описание методов

enterFullscreen

```
{ } enterFullscreen()
```

Позволяет переводить карту в полноэкранный режим.

exitFullscreen

```
{ } exitFullscreen()
```

Позволяет вывести карту из полноэкранного режима.

control.GeolocationControl

Расширяет [control.Button](#).

Элемент управления "геолокация". Позволяет отразить положение пользователя на карте. Ключ элемента управления в хранилище [control.storage](#) — "geolocationControl".

См.: [geolocation](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.GeolocationControl([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.
parameters.data	—	Тип: Object Объект, описывающий данные элемента управления.
parameters.data.image	'geolocation'	Тип: String URL иконки кнопки.
parameters.data.title	—	Тип: String Текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.
parameters.options	—	Тип: Object Опции элемента управления.
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли элемент управления свои размеры в менеджере отступов карты map.margin.Manager .

Параметр	Значение по умолчанию	Описание
parameters.options.float	"right"	<p>Тип: String</p> <p>Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируются только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.</p>
parameters.options.floatIndex	300	<p>Тип: Number</p> <p>Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".</p>
parameters.options.maxWidth	28	<p>Тип: Number Number[]</p> <p>Максимальная ширина элемента управления в различных состояниях. Если задано число, то считается, что элемент управления имеет одинаковые максимальные размеры во всех состояниях. Если задан массив, то он будет трактоваться как максимальная ширина в различных состояниях - от меньшего к большему. Количество доступных состояний задается в экземпляре класса control.Manager через опцию states. Этот класс обычно является полем Map.controls. По умолчанию элемент управления не изменяет своего размера, всегда имея вид кнопки с иконкой.</p>
parameters.options.noPlacemark	false	<p>Тип: Boolean</p> <p>При значении true метка-местоположение не будет показана на карте, а так же не произойдет автоматической смены центра и масштабирования карты.</p>
parameters.options.position	—	<p>Тип: Object</p> <p>Объект, описывающий позицию элемента управления. При указании опции position значение опции float автоматически трактуется как "none".</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.position.bottom</code>	'auto'	Тип: Number String Положение относительно нижнего края карты.
<code>parameters.options.position.left</code>	'auto'	Тип: Number String Положение относительно левого края карты.
<code>parameters.options.position.right</code>	'auto'	Тип: Number String Положение относительно правого края карты.
<code>parameters.options.position.top</code>	'auto'	Тип: Number String Положение относительно верхнего края карты.
<code>parameters.options.visible</code>	true	Тип: Boolean Признак того, что элемент управления отображается.
<code>parameters.state</code>	—	Тип: Object Объект, описывающий состояние элемента управления.
<code>options.useMapMargin</code>	true	Тип: Boolean Нужно ли учитывать отступы карты <code>map.margin.Manager</code> при ее центрировании.

Пример:

```
// Добавим элемент управления с собственной меткой геолокации на карте.
var geolocationControl = new ymaps.control.GeolocationControl({
  options: {noPlacemark: true}
});
geolocationControl.events.add('locationchange', function (event) {
  var position = event.get('position'),
  // При создании метки можно задать ей любой внешний вид.
  locationPlacemark = new ymaps.Placemark(position);

  myMap.geoObjects.add(locationPlacemark);
  // Установим новый центр карты в текущее местоположение пользователя.
  myMap.panTo(position);
});
myMap.controls.add(geolocationControl);
```

Поля

Имя	Тип	Описание
data	data.Manager	<p>Данные кнопки. Имена полей, доступных через метод <code>data.Manager.get</code>:</p> <ul style="list-style-type: none"> <code>image</code> - иконка кнопки, если есть; <code>content</code> - содержимое кнопки в виде HTML; <code>title</code> - текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши. <p>Унаследовано от control.Button.</p>
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от IControl.</p>
press		<p>Событие, сигнализирующее о том, что кнопка была нажата. В отличие от <code>click</code> генерируется только если состояние <code>isEnabled</code> == <code>true</code>. Экземпляр класса <code>Event</code>.</p> <p>Унаследовано от control.Button.</p>
state	data.Manager	<p>Состояние кнопки. Имена полей, доступных через метод <code>data.Manager.get</code>:</p> <ul style="list-style-type: none"> <code>selected</code> - признак, нажата ли кнопка; <code>enabled</code> - признак, активна ли кнопка. <code>size</code> - размер, который в данный момент выставлен кнопке; <p>Унаследовано от control.Button.</p>

События

Имя	Описание
click	<p>Нажатие на кнопку. Экземпляр класса Event.</p> <p>Унаследовано от control.Button.</p>
deselect	<p>Элемент управления не выбран.</p> <p>Унаследовано от ISelectableControl.</p>
disable	<p>Элемент управления стал недоступным.</p> <p>Унаследовано от ISelectableControl.</p>
enable	<p>Элемент управления стал доступным.</p> <p>Унаследовано от ISelectableControl.</p>

Имя	Описание
locationchange	<p>Событие определения положения пользователя. Список полей события, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>position</code> — местоположение пользователя в географических координатах. <code>geoObjects</code> — экземпляр класса GeoObjectCollection, в котором находится объект, обозначающий текущее местоположение пользователя.
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
select	<p>Элемент управления выбран.</p> <p>Унаследовано от ISelectableControl.</p>

Методы

Имя	Возвращает	Описание
deselect()		<p>Отменяет выбор элемента управления (выключает).</p> <p>Унаследован от ISelectableControl.</p>
disable()		<p>Делает элемент управления недоступным (запрещены пользовательские действия).</p> <p>Унаследован от ISelectableControl.</p>
enable()		<p>Делает элемент управления доступным (разрешены пользовательские действия).</p> <p>Унаследован от ISelectableControl.</p>
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от control.Button.</p>

Имя	Возвращает	Описание
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
isEnabled()	Boolean	Возвращает true - элемент управления доступен, false - недоступен. Унаследован от ISelectableControl .
isSelected()	Boolean	Возвращает true - элемент управления выбран, false - не выбран. Унаследован от ISelectableControl .
select()		Выбирает (включает) элемент управления. Унаследован от ISelectableControl .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание событий

locationchange

Событие определения положения пользователя. Список полей события, доступных через метод [Event.get](#):

- `position` — местоположение пользователя в географических координатах.
- `geoObjects` — экземпляр класса [GeoObjectCollection](#), в котором находится объект, обозначающий текущее местоположение пользователя.

control.ListBox

Расширяет [ICollection](#), [IControl](#), [ICustomizable](#).

Класс для создания элемента управления в виде раскрывающегося списка. Стандартный макет выпадающего списка изменяет свой внешний вид в зависимости от размеров карты. Если карта имеет большую ширину, то в заголовке выпадающего списка отображается текст. Если карта небольшого размера, то в заголовке списка отображается только иконка. Если для кнопки не задана иконка, то во всех состояниях будет отображаться текст и наоборот.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.ListBox([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры выпадающего списка.
parameters.data	—	Тип: Object Данные.
parameters.data.content	—	Тип: String Заголовок списка.
parameters.data.image	—	Тип: String URL иконки кнопки. Стандартный макет кнопки рассчитан на иконку размером 16x16 пикселей.
parameters.data.title	—	Тип: String Текст всплывающей подсказки, которая появляется при наведении на список курсора мыши.
parameters.items	—	Тип: IControl[] Массив дочерних элементов списка.
parameters.options	—	Тип: Object Опции элемента управления.
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли элемент управления свои размеры в менеджере отступов карты map.margin.Manager .
parameters.options.collapseOnBlur	true	Тип: Boolean Флаг, включающий сворачивание списка, когда кнопка теряет фокус (например, когда пользователь кликнул на другом элементе страницы).
parameters.options.expandOnClick	true	Тип: Boolean Флаг, разрешающий автоматическое разворачивание/сворачивание списка по клику.

Параметр	Значение по умолчанию	Описание
<code>parameters.options.float</code>	"right"	Тип: String Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируется только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.
<code>parameters.options.floatIndex</code>	0	Тип: Number Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".

Параметр	Значение по умолчанию	Описание
parameters.options.layout	—	<p>Тип: Function String</p> <p>Конструктор макета элемента управления, реализующий интерфейсы ISelectableControlLayout и IGroupControlLayout или ключ макета в хранилище layout.storage. В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> control - ссылка на элемент управления; options - менеджер опций элемента управления control.ListBox.options; data - менеджер данных элемента управления control.ListBox.data; state - менеджер состояния элемента управления control.ListBox.state. <p>Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует на интерфейсные события макета и меняет значения полей control.ListBox.state в зависимости от полученных команд.</p>
parameters.options.maxWidth	90	<p>Тип: Number Number[]</p> <p>Максимальная ширина кнопки списка при различных состояниях. Если задано число, то считается, что кнопка имеет одинаковые максимальные размеры во всех состояниях. Если задан массив, то он будет трактоваться как максимальная ширина кнопки в различных состояниях - от меньшего к большему. Количество доступных состояний задается в экземпляре класса control.Manager через опцию states. Этот класс обычно является полем Map.controls. По умолчанию у элементов управления есть три состояния - ['small', 'medium', 'large'].</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.popupFloat</code>	—	Тип: String Сторона, по которой нужно выравнивать попап элемента управления. Может принимать два значения: "left" и "right". По умолчанию определяется согласно опции "float".
<code>parameters.options.position</code>	—	Тип: Object Объект, описывающий позицию элемента управления. При указании опции position значение опции float автоматически трактуется как "none".
<code>parameters.options.position.bottom</code>	'auto'	Тип: Number String Положение относительно нижнего края карты.
<code>parameters.options.position.left</code>	'auto'	Тип: Number String Положение относительно левого края карты.
<code>parameters.options.position.right</code>	'auto'	Тип: Number String Положение относительно правого края карты.
<code>parameters.options.position.top</code>	'auto'	Тип: Number String Положение относительно верхнего края карты.
<code>parameters.options.visible</code>	true	Тип: Boolean Признак того, что элемент управления отображается.
<code>parameters.state</code>	—	Тип: Object Состояние выпадающего списка.
<code>parameters.state.expanded</code>	false	Тип: Boolean Признак, развернут ли список.

Примеры:**1.**

```
// Пример 1.
// Обработка нажатия на элементы списка.
var cityList = new ymaps.control.ListBox({
  data: {
    content: 'Выберите город'
  },
  // ...
});
```

```

    items: [
      new ymaps.control.ListBoxItem('Москва'),
      new ymaps.control.ListBoxItem('Новосибирск'),
      new ymaps.control.ListBoxItem({options: {type: 'separator'}}),
      new ymaps.control.ListBoxItem('Нью-Йорк'),
    ]
  });
cityList.get(0).events.add('click', function () {
  map.setCenter([55.752736, 37.606815]);
});
cityList.get(1).events.add('click', function () {
  map.setCenter([55.026366, 82.907803]);
});
cityList.get(3).events.add('click', function () {
  map.setCenter([40.695537, -73.97552]);
});
map.controls.add(cityList, { floatIndex: 0 });

```

2.

```

// Пример 2
// Создание пользовательского списка.
// В этом примере используется jQuery, загруженный из http://yandex.st/jquery/1.6.4/jquery.min.js

// По умолчанию раскрывающийся список реагирует на событие "click" и автоматически
// изменяет свое состояние на расширенное или свернутое.
var MyListBoxLayout = ymaps.templateLayoutFactory.createClass(
  '<div id="my-listbox-header" >{{ data.title }}</div >' +
  // Этот элемент будет служить контейнером для дочерних элементов списка.
  '<div id="my-list-box" style="display: {% if state.expanded %}block{% else %}none{% endif %};" >' +
  '</div >', {

    build: function() {
      MyListBoxLayout.superclass.build.call(this);
      this.childContainerElement = $('#my-list-box').get(0);
      // Каждый раз, когда мы перестраиваемся, мы генерируем событие
      // это означает, что контейнер для дочерних элементов изменился.
      // Формат события описан в интерфейсе IGroupControlLayout.
      this.events.fire('childcontainerchange', {
        newChildContainerElement: this.childContainerElement,
        oldChildContainerElement: null
      });
    },

    // Переопределим метод, который требует интерфейс IGroupControlLayout.
    getChildContainerElement: function () {
      return this.childContainerElement;
    }
  }
),

// Создадим список и выставим созданный макет через опции.
listBox = new ymaps.control.ListBox({options: {layout: MyListBoxLayout}});

```

3.

```

// Пример 3.
// Использование элемента управления ListBox в качестве фильтра
// для отображения объектов на карте (поддерживается мультिवыбор).
// Объекты добавляются на карту с помощью ObjectManager.

// Создание выпадающего списка с 5 элементами.
var listBoxItems = ['Школа', 'Аптека', 'Магазин', 'Больница', 'Бар']
.map(function(title) {
  return new ymaps.control.ListBoxItem({
    data: {
      content: title
    },
    state: {
      selected: true
    }
  });
});

// Теперь создаем выпадающий список из 5 пунктов.
var listBoxControl = new ymaps.control.ListBox({
  data: {
    content: 'Filter',
    title: 'Filter'
  },
  items: listBoxItems,
  state: {
    // Указывает, что список расширен.
    expanded: true,
    filters: listBoxItems.reduce(function(filters, filter) {
      filters[filter.data.get('content')] = filter.isSelected();
      return filters;
    }, {})
  }
});

map.controls.add(listBoxControl);

```

```
// Добавление отслеживания к индикатору для проверки того, выбран ли элемент списка.
listBoxControl.events.add(['select', 'deselect'], function(e) {
    var listBoxItem = e.get('target');
    var filters = ymaps.util.extend({}, listBoxControl.state.get('filters'));
    filters[listBoxItem.data.get('content')] = listBoxItem.isSelected();
    listBoxControl.state.set('filters', filters);
});

// Отслеживание изменений в элементе управления.список.государственное поле.
var filterMonitor = new ymaps.Monitor(listBoxControl.state);

filterMonitor.add('filters', function(filters) {
    // Применение фильтра к ObjectManager.
    objectManager.setFilter(getFilterFunction(filters));
});

function getFilterFunction(categories){
    return function(obj){
        var content = obj.properties.balloonContent;
        return categories[content]
    }
}
```

Поля

Имя	Тип	Описание
data	data.Manager	Данные выпадающего списка. Имена полей, доступных через метод data.Manager.get : <ul style="list-style-type: none"> <code>content</code> - заголовок выпадающего списка; <code>title</code> - текст всплывающей подсказки, которая появляется при наведении на список курсора мыши.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от IControl .
state	data.Manager	Состояние выпадающего списка. Имена полей, доступных через метод data.Manager.get : <ul style="list-style-type: none"> <code>expanded</code> - признак, раскрыт ли список; <code>size</code> - размер, который в данный момент выставлен списку;

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от ICollection .
click	Щелчок по заголовку списка. Экземпляр класса Event .
collapse	Список закрыт. Экземпляр класса Event .
expand	Список открыт. Экземпляр класса Event .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .

Имя	Описание
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .
press	Событие, сигнализирующее о том, что кнопка была нажата. В отличие от <code>click</code> генерируется только если состояние <code>isEnabled == true</code> . Экземпляр класса Event .
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection .

Методы

Имя	Возвращает	Описание
add(object)	ICollection	Добавляет в коллекцию дочерний объект. Унаследован от ICollection .
collapse()	control.ListBox	Сворачивает список.
expand()	control.ListBox	Раскрывает список.
getIterator()	IIterator	Возвращает итератор по коллекции. Унаследован от ICollection .
getMap()	Map	Возвращает ссылку на карту.
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
isExpanded()	Boolean	Возвращает признак того, находится ли контрол в развернутом состоянии.
remove(object)	ICollection	Удаляет из коллекции дочерний объект. Унаследован от ICollection .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание полей

data

```
{data.Manager} data
```

Данные выпадающего списка. Имена полей, доступных через метод `data.Manager.get`:

- `content` - заголовок выпадающего списка;
- `title` - текст всплывающей подсказки, которая появляется при наведении на список курсора мыши.

Пример:

```
// Добавим на карту выпадающий список и будем менять его всплывающую подсказку
// в зависимости от того, свернут список или развернут.

// Создадим группу слушателей событий.
var listBoxListener = listBox.events.group()
    .add('expand', function () {
        listBox.data.set('title', 'Список развернут.');
```

state

```
{data.Manager} state
```

Состояние выпадающего списка. Имена полей, доступных через метод `data.Manager.get`:

- `expanded` - признак, раскрыт ли список;
- `size` - размер, который в данный момент выставлен списку;

Пример:

```
// Создадим и добавим на карту список, который изначально открыт.
var listBox = new ymaps.control.ListBox();
listBox.state.set('expanded', true);
map.controls.add(listBox);
```

Описание событий

click

Щелчок по заголовку списка. Экземпляр класса [Event](#).

collapse

Список закрыт. Экземпляр класса [Event](#).

expand

Список открыт. Экземпляр класса [Event](#).

press

Событие, сигнализирующее о том, что кнопка была нажата. В отличие от `click` генерируется только если состояние `isEnabled == true`. Экземпляр класса [Event](#).

Описание методов

collapse

```
{control.ListBox} collapse()
```

Сворачивает список.

Возвращает ссылку на себя.

expand

```
{control.ListBox} expand()
```

Раскрывает список.

Возвращает ссылку на себя.

getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

isExpanded

```
{Boolean} isExpanded()
```

Возвращает признак того, находится ли контрол в развернутом состоянии.

control.ListBoxItem

Расширяет [ICustomizable](#), [ISelectableControl](#).

Элемент раскрывающегося списка.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.ListBoxItem([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object String Параметры элемента либо строка - HTML-содержимое элемента.
parameters.data	—	Тип: Object Данные элемента.
parameters.data.content	—	Тип: String Содержимое элемента.

Параметр	Значение по умолчанию	Описание
<code>parameters.options</code>	—	<p>Тип: Object</p> <p>Опции элемента управления.</p>
<code>parameters.options.layout</code>	<code>'islands#listBoxItemLayout'</code>	<p>Тип: Function String</p> <p>Конструктор макета элемента управления, реализующий интерфейс ISelectableControlLayout или ключ макета в хранилище layout.storage. Это базовый макет, который в зависимости от значения опции <code>type</code> выбирает, какой из подмакетов отобразить - <code>options.selectableLayout</code> или <code>options.separatorLayout</code>. В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> <code>control</code> - ссылка на элемент управления; <code>options</code> - менеджер опций элемента управления <code>control.ListBoxItem.options</code>; <code>data</code> - менеджер данных элемента управления <code>control.ListBoxItem.data</code>; <code>state</code> - менеджер состояния элемента управления <code>control.ListBoxItem.state</code>. <p>Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует на интерфейсные события макета и меняет значения полей <code>control.ListBoxItem.state</code> в зависимости от полученных команд.</p>
<code>parameters.options.selectableLayout</code>	<code>'islands#listBoxItemSelectableLayout'</code>	<p>Тип: Function String</p> <p>Конструктор макета элемента списка, реализующий интерфейс ISelectableControlLayout или ключ макета в хранилище layout.storage. Применяется к элементам с опцией <code>type='item'</code>. Опция стандартной реализации макета элемента списка.</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.selectOnClick</code>	<code>true</code>	<p>Тип: Boolean</p> <p>Флаг, разрешающий автоматическое выделение элемента списка по клику.</p> <ul style="list-style-type: none"> <code>true</code> — элемент списка выделяется после клика и меняет значение поля <code>control.ListBoxItem.state</code> на <code>'selected'</code>; <code>false</code> — элемент списка не меняет свой внешний вид и состояние после клика.
<code>parameters.options.separatorLayout</code>	<code>'islands#listBoxItemSeparatorLayout'</code>	<p>Тип: Function String</p> <p>Конструктор макета разделителя элементов списка, реализующий интерфейс <code>IControlLayout</code> или ключ макета в хранилище <code>layout.storage</code>. Применяется к элементам с опцией <code>type='separator'</code>. Опция стандартной реализации макета элемента списка.</p>
<code>parameters.options.type</code>	<code>'selectable'</code>	<p>Тип: String</p> <p>Тип пункта меню. В зависимости от значения этой опции макет элемента списка инстанцирует один из подмакетов - <code>options.selectableLayout</code> или <code>options.separatorLayout</code>. Возможные значения:</p> <ul style="list-style-type: none"> <code>'selectable'</code> - элемент списка, выделяется галочкой справа от содержимого; <code>'separator'</code> - разделитель.
<code>parameters.options.visible</code>	<code>true</code>	<p>Тип: Boolean</p> <p>Признак того, что элемент управления отображается.</p>
<code>parameters.state</code>	—	<p>Тип: Object</p> <p>Объект, описывающий состояние пункта меню.</p>
<code>parameters.state.selected</code>	<code>false</code>	<p>Тип: Boolean</p> <p>Признак, выбран ли пункт.</p>

Поля

Имя	Тип	Описание
data	data.Manager	Данные элемента списка. Имена полей, доступных через метод <code>data.Manager.get</code> : <ul style="list-style-type: none"> <code>content</code> - содержимое элемента списка в виде HTML; <code>title</code> - текст всплывающей подсказки.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
state	data.Manager	Состояние элемента списка. Имена полей, доступных через метод <code>data.Manager.get</code> : <ul style="list-style-type: none"> <code>selected</code> - признак, выбран ли элемент списка; <code>enabled</code> - признак, активен ли элемент списка.

События

Имя	Описание
click	Выбор элемента списка.
deselect	Элемент управления не выбран. Унаследовано от ISelectableControl .
disable	Элемент управления стал недоступным. Унаследовано от ISelectableControl .
enable	Элемент управления стал доступным. Унаследовано от ISelectableControl .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .
select	Элемент управления выбран. Унаследовано от ISelectableControl .

Методы

Имя	Возвращает	Описание
deselect()		Отменяет выбор элемента управления (выключает). Унаследован от ISelectableControl .
disable()		Делает элемент управления недоступным (запрещены пользовательские действия). Унаследован от ISelectableControl .
enable()		Делает элемент управления доступным (разрешены пользовательские действия). Унаследован от ISelectableControl .
getMap()	Map	Возвращает ссылку на карту.
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
isEnabled()	Boolean	Возвращает true - элемент управления доступен, false - недоступен. Унаследован от ISelectableControl .
isSelected()	Boolean	Возвращает true - элемент управления выбран, false - не выбран. Унаследован от ISelectableControl .
select()		Выбирает (включает) элемент управления. Унаследован от ISelectableControl .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание полей

data

```
{data.Manager} data
```

Данные элемента списка. Имена полей, доступных через метод `data.Manager.get`:

- `content` - содержимое элемента списка в виде HTML;
- `title` - текст всплывающей подсказки.

state

```
{data.Manager} state
```

Состояние элемента списка. Имена полей, доступных через метод `data.Manager.get`:

- `selected` - признак, выбран ли элемент списка;
- `enabled` - признак, активен ли элемент списка.

Описание событий

click

Выбор элемента списка.

Описание методов

getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

control.Manager

Менеджер элементов управления.

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
control.Manager(map[, controls[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: Map Экземпляр карты.
<code>controls</code>	—	Тип: <code>String[] IControl[]</code> Элементы управления, которые требуется разместить на карте изначально.
<code>options</code>	—	Тип: <code>Object</code> Опции менеджера.

Параметр	Значение по умолчанию	Описание
<code>options.margin</code>	10	Тип: Number Отступ элементов управления друг от друга и от краёв карты. Задаётся в пикселях.
<code>options.pane</code>	—	Тип: <code>IPane</code> Контейнер для всех элементов управления.
<code>options.states</code>	['small', 'medium', 'large']	Тип: String[] Массив размеров, начиная от наименьшего к наибольшему.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	<code>event.Manager</code>	Менеджер событий.
<code>options</code>	<code>option.Manager</code>	Опции менеджера.
<code>state</code>	<code>data.Manager</code>	Состояние менеджера. Имена полей, доступных через метод <code>data.Manager.get</code> : <ul style="list-style-type: none"> <code>size</code> - состояние в котором находятся элементы управления.

Методы

Имя	Возвращает	Описание
<code>add(control[, options])</code>	<code>control.Manager</code>	Добавляет элемент управления в менеджер.
<code>each(callback, context)</code>	<code>control.Manager</code>	Вызывает функцию-обработчик для всех элементов управления.
<code>get(index)</code>	<code>IControl</code> null	Возвращает элемент управления или null, если элемент не найден.
<code>getChildElement(control)</code>	<code>vow.Promise</code>	Возвращает объект-обещание, который подтверждается HTML-элементом, в который должен разместиться дочерний элемент.
<code>getContainer()</code>	HTMLElement	Возвращает контейнер, в который будут добавлены элементы контроля.

Имя	Возвращает	Описание
<code>getMap()</code>	<code>Map</code>	Возвращает ссылку на карту.
<code>indexOf(childToFind)</code>	<code>Integer</code>	Возвращает -1, если элемент управления не найден, либо индекс элемента в менеджере.
<code>remove(control)</code>	<code>control.Manager</code>	Удаление элемента управления из менеджера.

Описание полей

events

```
{event.Manager} events
```

Менеджер событий.

options

```
{option.Manager} options
```

Опции менеджера.

state

```
{data.Manager} state
```

Состояние менеджера. Имена полей, доступных через метод `data.Manager.get`:

- `size` - состояние в котором находятся элементы управления.

Описание методов

add

```
{control.Manager} add(control[, options])
```

Добавляет элемент управления в менеджер.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>control *</code>	—	<p>Тип: <code>IControl String</code></p> <p>Элементы управления, заданные экземплярами классов, реализующих интерфейс <code>IControl</code> либо ключами.</p> <p>Возможные значения ключей:</p> <ul style="list-style-type: none"> "fullscreenControl" — кнопка разворачивания карты на весь экран <code>control.FullscreenControl</code>; "geolocationControl" — кнопка определения местоположения пользователя <code>control.GeolocationControl</code>; "routeEditor" — кнопка включения и отключения поведения "редактор маршрута" <code>control.RouteEditor</code>; "rulerControl" — кнопка включения и отключения поведения "линейка" <code>control.RulerControl</code>; "searchControl" — поисковая строка <code>control.SearchControl</code>. "trafficControl" — панель пробок <code>control.TrafficControl</code>; "typeSelector" — панель переключения типа карты <code>control.TypeSelector</code>; "zoomControl" — ползунок масштаба <code>control.ZoomControl</code>; "routeButtonControl" — панель для построения маршрутов <code>control.RouteButton</code>, "routePanelControl" — панель маршрутизации <code>control.RoutePanel</code>. <p>Кроме того, вы можете задавать один из предустановленных наборов элементов управления с помощью специальных ключей:</p> <ul style="list-style-type: none"> "smallMapDefaultSet" — базовый набор элементов управления, оптимизированный для карт небольшого размера и экранов телефонов. Состоит из элементов управления "zoomControl", "searchControl", "typeSelector"

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	<p>Тип: Object</p> <p>Опции элемента управления.</p>
<code>options.float</code>	"right"	<p>Тип: String</p> <p>Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируются только по значениям опций left, right, bottom, top относительно границ карты.</p>
<code>options.floatIndex</code>	0	<p>Тип: Number</p> <p>Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".</p>
<code>options.position</code>	—	<p>Тип: Object</p> <p>Объект, описывающий позицию элемента управления. При указании опции position значение опции float автоматически трактуется как "none".</p>
<code>options.position.bottom</code>	'auto'	<p>Тип: Number String</p> <p>Положение относительно нижнего края карты. Работает только при float = none.</p>
<code>options.position.left</code>	'auto'	<p>Тип: Number String</p> <p>Положение относительно левого края карты. Работает только при float = none.</p>

Параметр	Значение по умолчанию	Описание
<code>options.position.right</code>	'auto'	Тип: Number String Положение относительно правого края карты. Работает только при float = none.
<code>options.position.top</code>	'auto'	Тип: Number String Положение относительно верхнего края карты. Работает только при float = none.

* Обязательный параметр/опция.

Пример:

```
map.controls
  .add('zoomControl')
  .add('typeSelector');
```

each

```
{control.Manager} each(callback, context)
```

Вызывает функцию-обработчик для всех элементов управления.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>callback *</code>	—	Тип: Function Функция-обработчик. На вход получает элемент коллекции. При возвращении функцией значения false перебор останавливается.
<code>context *</code>	—	Тип: Object Контекст вызываемой функции.

* Обязательный параметр/опция.

get

```
{IControl|null} get(index)
```

Возвращает элемент управления или null, если элемент не найден.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index *</code>	—	Тип: <code>Number String</code> Индекс искомого элемента или его ключ.

* Обязательный параметр/опция.

`getChildElement`

```
{vow.Promise} getChildElement(control)
```

Возвращает объект-обещание, который подтверждается HTML-элементом, в который должен разместиться дочерний элемент.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>control *</code>	—	Тип: <code>IControl</code> Элемент управления.

* Обязательный параметр/опция.

`getContainer`

```
{HTMLElement} getContainer()
```

Возвращает контейнер, в который добавляются элементы управления.

`getMap`

```
{Map} getMap()
```

Возвращает ссылку на карту.

`indexOf`

```
{Integer} indexOf(childToFind)
```

Возвращает -1, если элемент управления не найден, либо индекс элемента в менеджере.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>childToFind *</code>	—	Тип: <code>String IControl</code> Элемент управления или его ключ.

* Обязательный параметр/опция.

`remove`

```
{control.Manager} remove(control)
```

Удаление элемента управления из менеджера.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
control *	—	Тип: IControl String Удаляемый элемент управления, либо его ключ.

* Обязательный параметр/опция.

control.RouteButton

Расширяет [IControl](#), [ICustomizable](#).

Кнопка для построения маршрутов. При нажатии на кнопку появляется всплывающее окно, которое содержит поля ввода начальной и конечной точек маршрута. Элемент управления доступен в хранилище [control.storage](#) по ключу "routeButtonControl".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.RouteButton([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.
parameters.lazy	true	Тип: Boolean Если значение выставлено в true, будет включен режим ленивой загрузки – модули, которые необходимы для работы с маршрутизацией, начнут загружаться только тогда, когда пользователь впервые откроет панель.
parameters.options	—	Тип: Object Опции элемента управления. Используйте префикс 'routePanel', чтобы задать опции интерфейса IRoutePanel .
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли элемент управления свои размеры в менеджере отступов карты map.margin.Manager .

Параметр	Значение по умолчанию	Описание
parameters.options.autofocus	true	<p>Тип: Boolean</p> <p>Указывает, должна ли панель маршрутизации автоматически получать фокус при открытии всплывающего окна.</p>
parameters.options.collapseOnBlur	true	<p>Тип: Boolean</p> <p>Флаг, разрешающий сворачивание панели при потере фокуса на элементе управления.</p>
parameters.options.float	"right"	<p>Тип: String</p> <p>Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируется только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.</p>
parameters.options.floatIndex	0	<p>Тип: Number</p> <p>Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".</p>
parameters.options.popupAnimate	true	<p>Тип: Boolean</p> <p>Флаг, указывающий, использовать ли анимацию при раскрытии/сворачивании всплывающего окна.</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.popupFloat</code>	'auto'	<p>Тип: Boolean</p> <p>Определяет, по какому краю кнопки будет выравниваться всплывающее окно. Может принимать значения:</p> <ul style="list-style-type: none"> 'auto' - всплывающее окно выравнивается в зависимости от своих размеров и размеров карты; 'right' – всплывающее окно выравнивается по правому краю кнопки; 'right' – всплывающее окно выравнивается по левому краю кнопки;
<code>parameters.options.popupWidth</code>	'210px'	<p>Тип: String</p> <p>CSS-ширина всплывающего окна. Может задаваться как в пикселях, так и в процентах. Обратите внимание, что если указывать в процентах, ширина будет определяться относительно размеров кнопки, а не карты. Минимальное значение 176px, максимальное – 400px.</p>
<code>parameters.options.position</code>	—	<p>Тип: Object</p> <p>Объект, описывающий позицию элемента управления.</p>
<code>parameters.options.position.bottom</code>	'auto'	<p>Тип: Number String</p> <p>Положение относительно нижнего края карты.</p>
<code>parameters.options.position.left</code>	10	<p>Тип: Number String</p> <p>Положение относительно левого края карты.</p>
<code>parameters.options.position.right</code>	'auto'	<p>Тип: Number String</p> <p>Положение относительно правого края карты.</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.position.top</code>	108	Тип: Number String Положение относительно верхнего края карты.
<code>parameters.options.size</code>	'auto'	Тип: String Параметр, отвечающий за внешний вид элемента управления. Может принимать значения: <ul style="list-style-type: none"> 'auto' — макет автоматически изменяет свой внешний вид в зависимости от размеров карты и от количества добавленных элементов управления; 'small' — всегда показывается только иконка элемента управления, независимо от размеров карты; 'medium' — всегда отображается только подпись элемента управления; 'large' — отображаются и иконка, и подпись.
<code>parameters.options.visible</code>	true	Тип: Boolean Признак того, что элемент управления отображается.
<code>parameters.state</code>	—	Тип: Object Объект, описывающий состояние элемента управления.
<code>parameters.state.expanded</code>	false	Тип: Boolean Показывается ли панель маршрута в данный момент.

Примеры:**1.**

```
// Пример 1.
// Создадим элемент управления "Маршруты" и добавим его на карту.
var routeButton = new ymaps.control.RouteButton({
  options: {
    size: "small"
  }
});
```

```

    }
  });
  myMap.controls.add(routeButton);

```

2.

```

// Пример 2
// Добавим элемент управления в левый угол карты
// и зададим начальную и конечную точки маршрута.
myMap.controls.add('routeButtonControl', {
  size: "large",
  float: "left",
  floatIndex: 1000,
});
myMap.controls.get('routeButtonControl').routePanel.state.set({
  fromEnabled: false,
  from: "moscow",
  to: "saint petersburg",
  type: "auto"
});

```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от IControl .
routePanel	IRoutePanel	

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание полей

routePanel

```
{IRoutePanel} routePanel
```

control.RouteEditor

Расширяет [control.Button](#).

Элемент управления "Редактор маршрутов". Ключ элемента управления в хранилище [control.storage](#) — "routeEditor".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.RouteEditor([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.
parameters.data	—	Тип: Object Объект, описывающий данные элемента управления.
parameters.data.image	'routes'	Тип: String URL иконки кнопки.
parameters.data.title	—	Тип: String Текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.
parameters.options	—	Тип: Object Опции элемента управления.
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли элемент управления свои размеры в менеджере отступов карты map.margin.Manager .

Параметр	Значение по умолчанию	Описание
parameters.options.float	"right"	<p>Тип: String</p> <p>Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируются только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.</p>
parameters.options.floatIndex	100	<p>Тип: Number</p> <p>Приоритет расположения элемента управления. Элемент с максимальным приоритетом.</p>
parameters.options.layout	—	<p>Тип: ISelectableControlLayout String</p> <p>Конструктор макета элемента управления, реализующий интерфейс ISelectableControlLayout или ключ макета в хранилище layout.storage. В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> control - ссылка на элемент управления; options — менеджер опций элемента управления control.FullscreenControl.options; data — менеджер данных элемента управления control.FullscreenControl.data; state - менеджер состояния элемента управления control.FullscreenControl.state. <p>Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует на интерфейсные события макета</p>

Параметр	Значение по умолчанию	Описание
parameters.options.maxWidth	28	<p>Тип: Number Number[]</p> <p>Максимальная ширина элемента управления в различных состояниях. Если задано число, то считается, что элемент управления имеет одинаковые максимальные размеры во всех состояниях. Если задан массив, то он будет трактоваться как максимальная ширина в различных состояниях - от меньшего к большему. Количество доступных состояний задается в экземпляре класса control.Manager через опцию states. Этот класс обычно является полем Map.controls. По умолчанию у элементов управления есть три состояния - ['small', 'medium', 'large']. По умолчанию элемент управления не изменяет своего размера, всегда имея вид кнопки с иконкой.</p>
parameters.options.position	—	<p>Тип: Object</p> <p>Объект, описывающий позицию элемента управления. При указании опции position значение опции float автоматически трактуется как "none".</p>
parameters.options.position.bottom	'auto'	<p>Тип: Number String</p> <p>Положение относительно нижнего края карты.</p>
parameters.options.position.left	'auto'	<p>Тип: Number String</p> <p>Положение относительно левого края карты.</p>
parameters.options.position.right	'auto'	<p>Тип: Number String</p> <p>Положение относительно правого края карты.</p>
parameters.options.position.top	'auto'	<p>Тип: Number String</p> <p>Положение относительно верхнего края карты.</p>
parameters.options.visible	true	<p>Тип: Boolean</p> <p>Признак того, что элемент управления отображается.</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.state</code>	—	Тип: Object Объект, описывающий состояние элемента управления.

Пример:

```
// Добавляем контрол на карту.
map.controls.add('routeEditor');
```

Поля

Имя	Тип	Описание
<code>data</code>	<code>data.Manager</code>	Данные кнопки. Имена полей, доступных через метод <code>data.Manager.get</code> : <ul style="list-style-type: none"> <code>image</code> - иконка кнопки, если есть; <code>content</code> - содержимое кнопки в виде HTML; <code>title</code> - текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши. Унаследовано от <code>control.Button</code> .
<code>events</code>	<code>IEventManager</code>	Менеджер событий. Унаследовано от <code>IEventEmitter</code> .
<code>options</code>	<code>IOptionManager</code>	Менеджер опций. Унаследовано от <code>IControl</code> .
<code>press</code>		Событие, сигнализирующее о том, что кнопка была нажата. В отличие от <code>click</code> генерируется только если состояние <code>isEnabled == true</code> . Экземпляр класса <code>Event</code> . Унаследовано от <code>control.Button</code> .
<code>state</code>	<code>data.Manager</code>	Состояние кнопки. Имена полей, доступных через метод <code>data.Manager.get</code> : <ul style="list-style-type: none"> <code>selected</code> - признак, нажата ли кнопка; <code>enabled</code> - признак, активна ли кнопка. <code>size</code> - размер, который в данный момент выставлен кнопке; Унаследовано от <code>control.Button</code> .

События

Имя	Описание
<code>click</code>	Нажатие на кнопку. Экземпляр класса <code>Event</code> . Унаследовано от <code>control.Button</code> .
<code>deselect</code>	Элемент управления не выбран. Унаследовано от <code>ISelectableControl</code> .

Имя	Описание
disable	Элемент управления стал недоступным. Унаследовано от ISelectableControl .
enable	Элемент управления стал доступным. Унаследовано от ISelectableControl .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
select	Элемент управления выбран. Унаследовано от ISelectableControl .

Методы

Имя	Возвращает	Описание
deselect()		Отменяет выбор элемента управления (выключает). Унаследован от ISelectableControl .
disable()		Делает элемент управления недоступным (запрещены пользовательские действия). Унаследован от ISelectableControl .
enable()		Делает элемент управления доступным (разрешены пользовательские действия). Унаследован от ISelectableControl .
getMap()	Map	Возвращает ссылку на карту. Унаследован от control.Button .
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
getRoute()	router.Route	Возвращает маршрут.

Имя	Возвращает	Описание
isEnabled()	Boolean	Возвращает true - элемент управления доступен, false - недоступен. Унаследован от ISelectableControl .
isSelected()	Boolean	Возвращает true - элемент управления выбран, false - не выбран. Унаследован от ISelectableControl .
select()		Выбирает (включает) элемент управления. Унаследован от ISelectableControl .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание методов

getRoute

```
{router.Route} getRoute()
```

Возвращает маршрут.

control.RoutePanel

Расширяет [IControl](#), [ICustomizable](#).

Панель маршрутизации. Ключ в хранилище [control.storage](#) – 'routePanelControl'.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.RoutePanel([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.

Параметр	Значение по умолчанию	Описание
parameters.options	—	<p>Тип: Object</p> <p>Опции элемента управления. Для задания основных опций IRoutePanel используйте префикс 'routePanel'. Обратите внимание, параметр <code>routePanelAdjustMapMargin</code> по умолчанию принимает значение true.</p>
parameters.options.autofocus	true	<p>Тип: Boolean</p> <p>Должна ли панель автоматически получать фокус после ее добавления на карту.</p>
parameters.options.float	"left"	<p>Тип: String</p> <p>Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируется только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.</p>
parameters.options.floatIndex	0	<p>Тип: Number</p> <p>Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".</p>
parameters.options.maxWidth	'210px'	<p>Тип: String</p> <p>CSS-ширина всплывающего окна. Может задаваться как в пикселях, так и в процентах. Обратите внимание, что если указывать в процентах, ширина будет определяться относительно размеров кнопки, а не карты. Минимальное значение 176px, максимальное – 400px.</p>
parameters.options.position	—	<p>Тип: Object</p> <p>Объект, описывающий позицию элемента управления.</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.position.bottom</code>	—	Тип: Number String Положение относительно нижнего края карты.
<code>parameters.options.position.left</code>	—	Тип: Number String Положение относительно левого края карты.
<code>parameters.options.position.right</code>	—	Тип: Number String Положение относительно правого края карты.
<code>parameters.options.position.top</code>	—	Тип: Number String Положение относительно верхнего края карты.
<code>parameters.options.showHeader</code>	false	Тип: Boolean Показывать ли заголовок панели.
<code>parameters.options.title</code>	'Routes'	Тип: String Заголовок панели. Отображается только в том случае, если опция 'showHeader' выставлена в true.
<code>parameters.options.visible</code>	true	Тип: Boolean Признак того, что элемент управления отображается.
<code>parameters.state</code>	—	Тип: Object Объект, описывающий состояние элемента управления.

Примеры:**1.**

```
// Пример 1.
// Добавим на карту панель маршрутизации.
myMap.controls.add('routePanelControl')
```

2.

```
// Пример 2
// Создадим панель маршрутизации шириной в 300 пикселей.
// В качестве заголовка панели установим начальную точку.
myMap.controls.add('routePanelControl', {
  maxWidth: 300,
});
var routePanel = myMap.controls.get('routePanelControl').routePanel;
routePanel.options.set('adjustMapMargin', true);
routePanel.state.set({
  fromEnabled: false,
  from: "Москва",
});
```

```
to: "Санкт-Петербург",
type: "auto"
});
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от IControl .
routePanel	IRoutePanel	Панель маршрутизации.

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание полей

routePanel

```
{IRoutePanel} routePanel
```

Панель маршрутизации.

control.RulerControl

Расширяет [control.Button](#).

Элемент управления "Линейка". Ключ элемента управления в хранилище [control.storage](#) — "rulerControl".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.RulerControl([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.
parameters.data	—	Тип: Object Объект, описывающий данные элемента управления.
parameters.options	—	Тип: Object Опции элемента управления.
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли элемент управления свои размеры в менеджере отступов карты map.margin.Manager .
parameters.options.position	—	Тип: Object Объект, описывающий позицию элемента управления. При указании опции position значение опции float автоматически трактуется как "none".
parameters.options.position.bottom	30	Тип: Number String Положение относительно нижнего края карты.
parameters.options.position.left	'auto'	Тип: Number String Положение относительно левого края карты.
parameters.options.position.right	10	Тип: Number String Положение относительно правого края карты.
parameters.options.position.top	'auto'	Тип: Number String Положение относительно верхнего края карты.

Параметр	Значение по умолчанию	Описание
parameters.options.scaleLine	true	Тип: Boolean Флаг, указывающий, отображать или не отображать масштабную линейку справа от кнопки линейки.
parameters.options.visible	true	Тип: Boolean Признак того, что элемент управления отображается.
parameters.state	—	Тип: Object Объект, описывающий состояние элемента управления.

Поля

Имя	Тип	Описание
data	data.Manager	Данные кнопки. Имена полей, доступных через метод data.Manager.get : <ul style="list-style-type: none"> <code>image</code> - иконка кнопки, если есть; <code>content</code> - содержимое кнопки в виде HTML; <code>title</code> - текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши. Унаследовано от control.Button .
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от IControl .
press		Событие, сигнализирующее о том, что кнопка была нажата. В отличие от <code>click</code> генерируется только если состояние <code>isEnabled</code> == <code>true</code> . Экземпляр класса <code>Event</code> . Унаследовано от control.Button .
state	data.Manager	Состояние кнопки. Имена полей, доступных через метод data.Manager.get : <ul style="list-style-type: none"> <code>selected</code> - признак, нажата ли кнопка; <code>enabled</code> - признак, активна ли кнопка. <code>size</code> - размер, который в данный момент выставлен кнопке; Унаследовано от control.Button .

События

Имя	Описание
click	Нажатие на кнопку. Экземпляр класса Event . Унаследовано от control.Button .
deselect	Элемент управления не выбран. Унаследовано от ISelectableControl .
disable	Элемент управления стал недоступным. Унаследовано от ISelectableControl .
enable	Элемент управления стал доступным. Унаследовано от ISelectableControl .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
select	Элемент управления выбран. Унаследовано от ISelectableControl .

Методы

Имя	Возвращает	Описание
deselect()		Отменяет выбор элемента управления (выключает). Унаследован от ISelectableControl .
disable()		Делает элемент управления недоступным (запрещены пользовательские действия). Унаследован от ISelectableControl .
enable()		Делает элемент управления доступным (разрешены пользовательские действия). Унаследован от ISelectableControl .
getMap()	Map	Возвращает ссылку на карту. Унаследован от control.Button .

Имя	Возвращает	Описание
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
isEnabled()	Boolean	Возвращает true - элемент управления доступен, false - недоступен. Унаследован от ISelectableControl .
isSelected()	Boolean	Возвращает true - элемент управления выбран, false - не выбран. Унаследован от ISelectableControl .
select()		Выбирает (включает) элемент управления. Унаследован от ISelectableControl .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

control.SearchControl

Расширяет [IControl](#), [ICustomizable](#).

Элемент управления "Поиск по карте". Позволяет обрабатывать поисковый запрос пользователя и отображать результат в панели и на карте.

Каждый результат поиска представляется в панели элемента управления в виде двухстрочного блока. Для формирования блока используются поля name и description объекта-результата геокодирования.

Ключ элемента управления в хранилище [control.storage](#) — "searchControl".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.SearchControl([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.
parameters.data	—	Тип: Object Объект, описывающий данные элемента управления.

Параметр	Значение по умолчанию	Описание
<code>parameters.options</code>	—	<p>Тип: Object</p> <p>Опции элемента управления.</p>
<code>parameters.options.adjustMapMargin</code>	false	<p>Тип: Boolean</p> <p>Регистрирует ли элемент управления свои размеры в менеджере отступов карты <code>map.margin.Manager</code>.</p>
<code>parameters.options.boundedBy</code>	—	<p>Тип: Number[][]</p> <p>Прямоугольная область на карте, где предположительно находится искомый объект. При ранжировании объекты, находящиеся внутри указанной области, будут получать больший приоритет.</p>
<code>parameters.options.fitMaxWidth</code>	false	<p>Тип: Boolean</p> <p>Растягивать ли поисковую строку до максимального размера.</p>
<code>parameters.options.float</code>	"right"	<p>Тип: String</p> <p>Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируется только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.</p>
<code>parameters.options.floatIndex</code>	200	<p>Тип: Number</p> <p>Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.formLayout</code>	<code>'islands#searchControlFormLayout'</code>	<p>Тип: <code>ILayout String</code></p> <p>Конструктор макета формы для реализации поиска в макете элемента управления по умолчанию или ключ в хранилище <code>layout.storage</code>.</p> <p>В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none">• <code>control</code> - ссылка на элемент управления;• <code>options</code> - менеджер опций элемента управления <code>control.SearchControl.options</code>;• <code>data</code> - менеджер данных элемента управления <code>control.SearchControl.data</code>;• <code>state</code> - менеджер состояния элемента управления <code>control.SearchControl.state</code>.
<code>parameters.options.kind</code>	<code>'house'</code>	<p>Тип: <code>String</code></p> <p>Вид топонима (только для обратного геокодирования). Список возможных значений:</p> <ul style="list-style-type: none">• <code>house</code> - дом;• <code>street</code> - улица;• <code>metro</code> - станция метро;• <code>district</code> - район города;• <code>locality</code> - населенный пункт (город/поселок/деревня/село/...).

Параметр	Значение по умолчанию	Описание
parameters.options.layout	'islands#searchControlLayout'	<p>Тип: ISearchControlLayout String</p> <p>Макет элемента управления.</p> <p>В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> control - ссылка на элемент управления; options - менеджер опций элемента управления control.SearchControl.options; data - менеджер данных элемента управления control.SearchControl.data; state - менеджер состояния элемента управления control.SearchControl.state. <p>Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует на интерфейсные события макета и меняет значения полей control.SearchControl.state в зависимости от полученных команд.</p>
parameters.options.maxWidth	[30, 72, 315]	<p>Тип: Number Number[]</p> <p>Максимальная ширина элемента управления в различных состояниях. Если задано число, то считается, что элемент управления имеет одинаковые максимальные размеры во всех состояниях. Если задан массив, то он будет трактоваться как максимальная ширина в различных состояниях - от меньшего к большему. Количество доступных состояний задается в экземпляре класса control.Manager через опцию states. Этот класс обычно является полем Map.controls. Для состояния 'large' ширина может быть в пределах от 280 до 660 пикселей. По умолчанию у элементов управления есть три состояния - ['small', 'medium', 'large'].</p>

Параметр	Значение по умолчанию	Описание
parameters.options.noCentering	false	<p>Тип: Boolean</p> <p>false - автоматически располагать центр карты так, чтобы объект был виден целиком. Значение true - не изменять центр карты при показе найденного объекта. Если указать noCentering = true и noPlacemark = true, то при щелчке по результату поиска никаких видимых изменений на карте не произойдет.</p>
parameters.options.noPlacemark	false	<p>Тип: Boolean</p> <p>false - автоматически добавлять в центр найденного объекта метку с открытым балуном, true - не добавлять. Если указать noCentering = true и noPlacemark = true, то при щелчке по результату поиска никаких видимых изменений на карте не произойдет. Опция не работает с провайдером yandex#search.</p>
parameters.options.noPopup	false	<p>Тип: Boolean</p> <p>true - не показывать выпадающий список результатов, false - показывать.</p>
parameters.options.noSelect	false	<p>Тип: Boolean</p> <p>При значении false будет автоматически показываться результат поиска, если он является единственным найденным объектом, true — результат не будет выбираться.</p>
parameters.options.noSuggestPanel	false	<p>Тип: Boolean</p> <p>true - не показывать панель с поисковыми подсказками, false - показывать.</p>
parameters.options.placeholderContent	—	<p>Тип: String</p> <p>Текст-подсказка, показываемый в поле ввода элемента управления.</p>

Параметр	Значение по умолчанию	Описание
parameters.options.popupItemLayout	'islands#searchControlPopupItemLayout'	<p>Тип: ILayout String</p> <p>Конструктор макета результата поиска в выпадающем списке в макете элемента управления по умолчанию или ключ в хранилище layout.storage.</p> <p>В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none">• control - ссылка на элемент управления;• options - менеджер опций элемента управления <code>control.SearchControl.options</code>;• data - менеджер данных элемента управления <code>control.SearchControl.data</code>;• state - менеджер состояния элемента управления control.SearchControl.state.

Параметр	Значение по умолчанию	Описание
<code>parameters.options.popupLayout</code>	<code>'islands#searchControlPopupLayout'</code>	<p>Тип: <code>ILayout String</code></p> <p>Конструктор макета выпадающего списка с результатами поиска в макете элемента управления по умолчанию или ключ в хранилище <code>layout.storage</code>.</p> <p>В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> <code>control</code> - ссылка на элемент управления; <code>options</code> - менеджер опций элемента управления <code>control.SearchControl.options</code>; <code>data</code> - менеджер данных элемента управления <code>control.SearchControl.data</code>; <code>state</code> - менеджер состояния элемента управления <code>control.SearchControl.state</code>.
<code>parameters.options.position</code>	—	<p>Тип: <code>Object</code></p> <p>Объект, описывающий позицию элемента управления. При указании опции <code>position</code> значение опции <code>float</code> автоматически трактуется как <code>"none"</code>.</p>
<code>parameters.options.position.bottom</code>	<code>'auto'</code>	<p>Тип: <code>Number String</code></p> <p>Положение относительно нижнего края карты.</p>
<code>parameters.options.position.left</code>	<code>'auto'</code>	<p>Тип: <code>Number String</code></p> <p>Положение относительно левого края карты.</p>
<code>parameters.options.position.right</code>	<code>'auto'</code>	<p>Тип: <code>Number String</code></p> <p>Положение относительно правого края карты.</p>
<code>parameters.options.position.top</code>	<code>'auto'</code>	<p>Тип: <code>Number String</code></p> <p>Положение относительно верхнего края карты.</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.provider</code>	'yandex#map'	<p>Тип: <code>IGeocodeProvider String</code></p> <p>Провайдер геокодирования. Можно воспользоваться одним из стандартных провайдеров:</p> <ul style="list-style-type: none"> 'yandex#map' - поиск топонимов на карте; 'yandex#search' - поиск по топонимам и организациям. Для поиска по организациям не работают следующие опции: «noPlacemark», «noCentering», «noSelect», «strictBounds», «kind». Кроме того, по умолчанию на карте показывается 20 объектов, если не указана опция «results» для элемента управления.
<code>parameters.options.searchCoordOrder</code>	'latlong'	<p>Тип: <code>String</code></p> <p>Определяет каким образом нужно интерпретировать координаты в запросе. По умолчанию координаты будут обрабатываться как широта-долгота. Опция не работает с провайдером <code>yandex#search</code>.</p>
<code>parameters.options.size</code>	'auto'	<p>Тип: <code>String</code></p> <p>Параметр, отвечающий за внешний вид элемента управления. Может принимать значения:</p> <ul style="list-style-type: none"> 'small' — всегда показывать кнопку с иконкой; 'medium' — всегда показывать кнопку с текстом; 'large' — всегда показывать полную поисковую форму; 'auto' — производить автоматический выбор размера элемента управления в зависимости от наличия свободного пространства в тулбаре.

Параметр	Значение по умолчанию	Описание
parameters.options.strictBounds	—	Тип: Boolean Искать только внутри области, заданной опцией <code>boundedBy</code> . Объекты вне указанной области попадать в выдачу не будут.
parameters.options.useMapBounds	—	Тип: Boolean Флаг, позволяющий учитывать при поиске границы видимой области карты. При значении <code>true</code> рассчитанная область видимости имеет больший приоритет, чем заданная через <code>boundedBy</code> .
parameters.options.zoomMargin	0	Тип: Number Отступы от границ видимой области карты при показе найденных результатов. Для того чтобы опция применилась, значение <code>noCentering</code> должно быть <code>false</code> . Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы <code>top</code> , <code>right</code> , <code>bottom</code> , <code>left</code> . При включенной опции <code>"useMapMargin"</code> значение <code>"zoomMargin"</code> складывается со значениями, которые были рассчитаны в менеджере отступов map.margin.Manager .
parameters.state	—	Тип: Object Объект, описывающий состояние элемента управления.
options.useMapMargin	true	Тип: Boolean Нужно ли учитывать отступы карты map.margin.Manager при показе результатов поиска на карте.

Примеры:**1.**

```
// Пример 1.
// Создание элемента управления и добавление его на карту.
var searchControl = new ymaps.control.SearchControl({
  options: {
    float: 'right',
    floatIndex: 100,
    noPlacemark: true
  }
});
myMap.controls.add(searchControl);
```

2.

```
// Пример 2
// Если элемент управления уже добавлен на карту по ключу из хранилища control.storage,
// то его экземпляр можно получить с помощью метода control.Manager.get
// менеджера элементов управления control.Manager.

var searchControl = myMap.controls.get('searchControl');
searchControl.events.add('submit', function () {
    console.log('request: ' + searchControl.getRequestString());
}, this);
```

3.

```
// Пример 3.
// Включим поиск по организациям в элементе управления, уже добавленном на карту.
// Если элемент управления уже добавлен на карту по ключу из хранилища control.storage,
// то его экземпляр можно получить с помощью метода control.Manager.get.

var searchControl = myMap.controls.get('searchControl');
searchControl.options.set('provider', 'yandex#search');
```

4.

```
// Пример 4.
// Создадим элемент управления «поиск по карте» с включенным поиском по организациям.

var searchControl = new ymaps.control.SearchControl({
    options: {
        float: 'left',
        provider: 'yandex#search'
    }
});
```

5.

```
// Пример 5.
// Увеличим значение максимальной ширины для поисковой строки в состоянии 'large'.
map.options.set({
    // Максимальная ширина для трех состояний: [30, 72, 315].
    // Изменим ширину для состояния 'large'.
    searchControlMaxWidth: [30, 72, 500],
    // Расширим поисковую строку до максимального размера.
    fitMaxWidth: true
});
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от IControl .
state	data.Manager	Состояние элемента управления. Имена полей, доступных через метод data.Manager.get : <ul style="list-style-type: none"> <code>size</code> — текущий размер элемента управления; <code>results</code> — массив, содержащий результаты поиска; <code>currentIndex</code> — индекс текущего выбранного элемента; <code>found</code> — общее количество найденных результатов; <code>request</code> — текущий активный запрос; <code>correction</code> — исправленный запрос; <code>noSuggestPanel</code> - флаг, скрывать ли панель поисковых подсказок.

События

Имя	Описание
clear	Событие очистки результатов поиска. Экземпляр класса Event .
error	Событие ошибки получения результатов поиска с сервера. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>error</code> — информация об ошибке.
load	Событие получения результатов поиска с сервера. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>skip</code> — сколько элементов было пропущено; <code>count</code> — количество загруженных элементов.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .
resultselect	Событие выбора результата поиска. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>index</code> — индекс выбранного результата.
resultshow	Событие показа результата поиска. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>index</code> — индекс выбранного результата.
submit	Событие отправления поискового запроса на сервер. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
clear()		Очищает результаты поиска, а также содержимое поисковой строки элемента управления.
getMap()	Map	Возвращает ссылку на карту.
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
getRequestString()	String	Возвращает поисковый запрос.

Имя	Возвращает	Описание
<code>getResponseMetaData()</code>	Object	Возвращает метаданные геопоиска.
<code>getResult(index)</code>	vow.Promise	Получение результатов поиска.
<code>getResultsArray()</code>	Object[]	Возвращает массив, содержащий все текущие результаты.
<code>getResultsCount()</code>	Integer	Возвращает количество найденных результатов.
<code>getSelectedIndex()</code>	Integer	Возвращает индекс выбранного элемента.
<code>hideResult()</code>		Скрывает результат, показанный на карте.
<code>search(request, options)</code>	vow.Promise	Осуществляет поиск.
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .
<code>showResult(index)</code>	control.SearchControl	Отображает результат с заданным индексом.

Описание полей

state

```
{data.Manager} state
```

Состояние элемента управления. Имена полей, доступных через метод `data.Manager.get`:

- `size` — текущий размер элемента управления;
- `results` — массив, содержащий результаты поиска;
- `currentIndex` — индекс текущего выбранного элемента;
- `found` — общее количество найденных результатов;
- `request` — текущий активный запрос;
- `correction` — исправленный запрос;
- `noSuggestPanel` - флаг, скрывать ли панель поисковых подсказок.

Описание событий

clear

Событие очистки результатов поиска. Экземпляр класса [Event](#).

error

Событие ошибки получения результатов поиска с сервера. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `error` — информация об ошибке.

load

Событие получения результатов поиска с сервера. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `skip` — сколько элементов было пропущено;
- `count` — количество загруженных элементов.

resultselect

Событие выбора результата поиска. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `index` — индекс выбранного результата.

resultshow

Событие показа результата поиска. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `index` — индекс выбранного результата.

submit

Событие отправления поискового запроса на сервер. Экземпляр класса [Event](#).

Описание методов**clear**

```
{ } clear()
```

Очищает результаты поиска, а также содержимое поисковой строки элемента управления.

getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

getRequestString

```
{String} getRequestString()
```

Возвращает поисковый запрос.

getResponseMetaData

```
{Object} getResponseMetaData()
```

Возвращает метаданные геопоиска.

getResult

```
{vow.Promise} getResult(index)
```

Получение результатов поиска.

Возвращает объект типа [vow.Promise](#).

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index *</code>	—	Тип: Integer Индекс результата, начинается с 0.

* Обязательный параметр/опция.

getResultsArray

```
{Object[]} getResultsArray()
```

Возвращает массив, содержащий все текущие результаты.

getResultsCount

```
{Integer} getResultsCount()
```

Возвращает количество найденных результатов.

getSelectedIndex

```
{Integer} getSelectedIndex()
```

Возвращает индекс выбранного элемента.

hideResult

```
{ } hideResult()
```

Скрывает результат, показанный на карте.

Пример:

```
// Если мы показали какой-то результат на карте с помощью control.SearchControl.showResult,  
// или он был показан автоматически при поиске, мы можем скрыть его, например,  
// по клику на кнопку.  
var myButton = new ymaps.control.Button("Hide results");  
myButton.events.add('click', function () {  
    searchControl.hideResult();  
}, this);  
myMap.controls.add(myButton, { selectOnClick: false });
```

search

```
{vow.Promise} search(request, options)
```

Осуществляет поиск.

Возвращает объект типа [vow.Promise](#).

Параметры:

Параметр	Значение по умолчанию	Описание
<code>request *</code>	—	Тип: String Запрос.

Параметр	Значение по умолчанию	Описание
<code>options</code> *	—	Тип: Object Дополнительные опции провайдера.

* Обязательный параметр/опция.

Пример:

```
// Попробуем найти Москву и вывести поле name
// первого найденного результата в консоль.
searchControl.search('Москва').then(function () {
    var geoObjectsArray = searchControl.getResultsArray();
    if (geoObjectsArray.length) {
        // Выводит свойство name первого геообъекта из результатов запроса.
        console.log(geoObjectsArray[0].properties.get('name'));
    }
});
```

showResult

```
{control.SearchControl} showResult(index)
```

Отображает результат с заданным индексом.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index</code> *	—	Тип: Integer Индекс результата, начинается с 0.

* Обязательный параметр/опция.

Пример:

```
// Мы хотим показывать всегда первый результат,
// вне зависимости от количества найденных
// объектов на карте (1 или более).
var searchControl = new ymaps.control.SearchControl({
    // Опция отключает автоматический выбор результата поиска.
    options: { noSelect: true }
});
searchControl.events.add('load', function (event) {
    // Проверяем, что это событие не "дозагрузки" результатов и
    // по запросу найден хотя бы один результат.
    if (!event.get('skip') && searchControl.getResultsCount()) {
        searchControl.showResult(0);
    }
});
```

control.storage

Статический объект.

Экземпляр класса [util.Storage](#)

Хранилище элементов управления карты. Содержит соответствие ключей элементов управления и их конструкторов.

По умолчанию в хранилище добавлены следующие элементы управления:

- "rulerControl" — линейка и масштабный отрезок [control.RulerControl](#);

- "searchControl" — поиск на карте [control.SearchControl](#);
- "trafficControl" — панель пробок [control.TrafficControl](#);
- "typeSelector" — панель переключения типа карты [control.TypeSelector](#);
- "zoomControl" — ползунок масштаба [control.ZoomControl](#);
- "geolocationControl" — элемент управления геолокацией [control.GeolocationControl](#);
- "routeEditor" — редактор маршрутов [control.RouteEditor](#).
- "fullscreenControl" — элемент управления «полноэкранным режимом» [control.FullscreenControl](#).
- "routeButtonControl" — панель для построения маршрутов [control.RouteButton](#),
- "routePanelControl" — панель маршрутизации [control.RoutePanel](#).

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.

control.TrafficControl

Расширяет [IControl](#), [ICustomizable](#).

Панель управления пробками на карте.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.TrafficControl([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.
parameters.options	—	Тип: Object Опции элемента управления.
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли элемент управления свои размеры в менеджере отступов карты map.margin.Manager .

Параметр	Значение по умолчанию	Описание
<code>parameters.options.collapseOnBlur</code>	true	Тип: Boolean Флаг, разрешающий сворачивание панели при потере фокуса на элементе управления. Например при клике на документ.
<code>parameters.options.float</code>	"right"	Тип: String Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируется только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.
<code>parameters.options.floatIndex</code>	100	Тип: Number Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".

Параметр	Значение по умолчанию	Описание
parameters.options.layout	—	<p>Тип: <code>Function String</code></p> <p>Макет элемента управления. В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> <code>control</code> - ссылка на элемент управления; <code>options</code> - менеджер опций элемента управления <code>control.TrafficControl.options</code>; <code>data</code> — менеджер данных элемента управления <code>control.TrafficControl.data</code>; <code>state</code> — менеджер состояний элемента управления <code>control.TrafficControl.state</code>; <p>Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует на интерфейсные события макета и меняет значения полей <code>control.TrafficControl.state</code> в зависимости от полученных команд. (Тип: конструктор объекта с интерфейсом <code>ITrafficControlLayout</code> или ключ макета).</p>
parameters.options.maxWidth	[26, 195, 195]	<p>Тип: <code>Number Number[]</code></p> <p>Максимальная ширина кнопки в различных состояниях. Если задано число, то считается, что элемент управления имеет одинаковые максимальные размеры во всех состояниях. Если задан массив, то он будет трактоваться как максимальная ширина в различных состояниях - от меньшего к большему. Количество доступных состояний задается в экземпляре класса <code>control.Manager</code> через опцию <code>states</code>. Этот класс обычно является полем <code>Map.controls</code>. По умолчанию у элементов управления есть три состояния - ['small', 'medium', 'large'].</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.position</code>	—	<p>Тип: Object</p> <p>Объект, описывающий позицию элемента управления. При указании опции <code>position</code> значение опции <code>float</code> автоматически трактуется как <code>"none"</code>.</p>
<code>parameters.options.position.bottom</code>	<code>'auto'</code>	<p>Тип: Number String</p> <p>Положение относительно нижнего края карты.</p>
<code>parameters.options.position.left</code>	<code>'auto'</code>	<p>Тип: Number String</p> <p>Положение относительно левого края карты.</p>
<code>parameters.options.position.right</code>	<code>'auto'</code>	<p>Тип: Number String</p> <p>Положение относительно правого края карты.</p>
<code>parameters.options.position.top</code>	<code>'auto'</code>	<p>Тип: Number String</p> <p>Положение относительно верхнего края карты.</p>
<code>parameters.options.size</code>	<code>'auto'</code>	<p>Тип: String</p> <p>Параметр, отвечающий за внешний вид стандартного макета элемента управления пробками. Может принимать значения:</p> <ul style="list-style-type: none"> <code>'auto'</code> — макет автоматически изменяет свой внешний вид в зависимости от размеров карты и от количества добавленных элементов управления; <code>'small'</code> - в макете кнопки отображается только иконка светофора, независимо от размера карты; <code>'large'</code> - в макете кнопки всегда отображается иконка светофора и текст, независимо от размера карты.

Параметр	Значение по умолчанию	Описание
parameters.options.visible	true	Тип: Boolean Признак того, что элемент управления отображается.
parameters.state	—	Тип: Object Состояние элемента управления.
parameters.state.providerKey	'traffic#actual'	Тип: String Ключ провайдера пробок, показываемых на карте. <ul style="list-style-type: none"> 'traffic#actual' - пробки "сейчас" 'traffic#archive' - пробки "обычно"
parameters.state.trafficShown	false	Тип: Boolean Показаны ли данные по пробкам на карте.

Пример:

```
// Добавление на карту элемента управления пробками
// с включенными пробками "сейчас".
var trafficControl = new ymaps.control.TrafficControl({state: {trafficShown: true}});
map.controls.add(trafficControl, {top:
```

Поля

Имя	Тип	Описание
data	data.Manager	Данные панели.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от IControl .
state	data.Manager	Состояние панели. Имена полей, доступных через метод data.Manager.get : <ul style="list-style-type: none"> <code>trafficShown</code> - признак, показан ли на карте провайдер пробок; <code>providerKey</code> - ключ провайдера, который показывает панель. Может принимать значения 'traffic#actual' или 'traffic#archive'; <code>expanded</code> - признак, развернута ли панель.

События

Имя	Описание
collapse	Панель пробок свернута. Экземпляр класса Event .

Имя	Описание
expand	Панель пробок развернута. Экземпляр класса Event .
hidetraffic	Пробки скрыты. Экземпляр класса Event .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
providerkeychange	Сменился ключ провайдера. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> newProviderKey - новое значение ключа провайдера; oldProviderKey – старое значение ключа.
showtraffic	Показаны пробки. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
collapse()		Свернуть панель пробок.
expand()		Развернуть панель пробок.
getMap()	Map	Возвращает ссылку на карту.
getParent()	IControlParent null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
getProvider([key])	ITrafficProvider	Возвращает экземпляр провайдера пробок.
hideTraffic()		Скрыть провайдер пробок с карты.
isExpanded()	Boolean	Возвращает признак, развернута ли панель.
isTrafficShown()	Boolean	Возвращает признак, показан ли провайдер пробок на карте.

Имя	Возвращает	Описание
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .
showTraffic()		Показать провайдер пробок на карте.

Описание полей

data

```
{data.Manager} data
```

Данные панели.

state

```
{data.Manager} state
```

Состояние панели. Имена полей, доступных через метод `data.Manager.get`:

- `trafficShown` - признак, показан ли на карте провайдер пробок;
- `providerKey` - ключ провайдера, который показывает панель. Может принимать значения `'traffic#actual'` или `'traffic#archive'`;
- `expanded` - признак, развернута ли панель.

Описание событий

collapse

Панель пробок свернута. Экземпляр класса [Event](#).

expand

Панель пробок развернута. Экземпляр класса [Event](#).

hidetraffic

Пробки скрыты. Экземпляр класса [Event](#).

providerkeychange

Сменился ключ провайдера. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `newProviderKey` - новое значение ключа провайдера;
- `oldProviderKey` – старое значение ключа.

showtraffic

Показаны пробки. Экземпляр класса [Event](#).

Описание методов

collapse

```
{ } collapse()
```

Свернуть панель пробок.

expand

```
{ } expand()
```

Развернуть панель пробок.

getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

getProvider

```
{ITrafficProvider} getProvider([key])
```

Возвращает экземпляр провайдера пробок.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key</code>	—	Тип: String Ключ провайдера информации о пробках. Список доступных ключей: <ul style="list-style-type: none">'traffic#actual' - провайдер пробок "сейчас";'traffic#archive' — провайдер пробок "обычно". Если параметр не передан, возвращается текущий провайдер.

Пример:

```
// Добавим на карту контрол пробок.  
map.controls.add('trafficControl');  
// Провайдер пробок "сейчас" при открытии будет показывать слой дорожных событий.  
map.controls.get('trafficControl').getProvider('traffic#actual').state.set('infoLayerShown', true);
```

hideTraffic

```
{ } hideTraffic()
```

Скрыть провайдер пробок с карты.

isExpanded

```
{Boolean} isExpanded()
```

Возвращает признак, развернута ли панель.

isTrafficShown

```
{Boolean} isTrafficShown()
```

Возвращает признак, показан ли провайдер пробок на карте.

showTraffic

```
{ } showTraffic()
```

Показать провайдер пробок на карте.

control.TypeSelector

Расширяет [control.ListBox](#).

Элемент управления "Типы карты". В данный элемент управления можно добавлять элементы списка, которые описывают типы карты, а также дополнительные элементы. Ключ элемента управления в хранилище [control.storage](#) — "typeSelector".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.TypeSelector([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: String[] MapType [] Object Объект с описаниями параметров элемента управления. Если передан массив, то он трактуется как массив типов карты.

Параметр	Значение по умолчанию	Описание
<code>parameters.mapTypes</code>	—	<p>Тип: <code>String[] MapType[]</code></p> <p>Массив конструкторов типов карты или ключей. Если параметр отсутствует, в элемент будет добавлен стандартный набор типов карты. Список доступных типов карты:</p> <ul style="list-style-type: none"> • <code>'yandex#map'</code> - тип карты "схема"; • <code>'yandex#satellite'</code> - тип карты "спутник"; • <code>'yandex#hybrid'</code> - тип карты "гибрид".
<code>parameters.options</code>	—	<p>Тип: <code>Object</code></p> <p>Опции элемента управления.</p>
<code>parameters.options.adjustMapMargin</code>	<code>false</code>	<p>Тип: <code>Boolean</code></p> <p>Регистрирует ли элемент управления свои размеры в менеджере отступов карты <code>map.margin.Manager</code>.</p>
<code>parameters.options.collapseOnBlur</code>	<code>true</code>	<p>Тип: <code>Boolean</code></p> <p>Флаг, разрешающий сворачивание списка при потере фокуса на кнопке. Например при клике на документ.</p>
<code>parameters.options.collapseTimeout</code>	<code>3000</code>	<p>Тип: <code>Number</code></p> <p>Временной интервал, по истечении которого раскрытый список автоматически закрывается.</p>
<code>parameters.options.expandOnClick</code>	<code>true</code>	<p>Тип: <code>Boolean</code></p> <p>Флаг, разрешающий автоматическое разворачивание/сворачивание списка по клику.</p>

Параметр	Значение по умолчанию	Описание
parameters.options.float	"right"	<p>Тип: String</p> <p>Сторона, по которой нужно выравнивать элемент управления. Может принимать три значения: "left", "right" или "none". При значении "left" или "right" элементы управления выстраиваются друг за другом, начиная от левого или правого края карты соответственно. При значении "none" элементы управления позиционируется только по значениям опций left, right, bottom, top относительно границ карты. Также смотрите описание опции position.</p>
parameters.options.floatIndex	200	<p>Тип: Number</p> <p>Приоритет расположения элемента управления. Элемент с максимальным приоритетом находится ближе к указанному в свойстве float краю карты. Не работает при float = "none".</p>

Параметр	Значение по умолчанию	Описание
parameters.options.layout	—	<p>Тип: Function String</p> <p>Конструктор макета элемента управления, реализующий интерфейсы ISelectableControlLayout и IGroupControlLayout или ключ макета в хранилище layout.storage. В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> control - ссылка на элемент управления; options - менеджер опций элемента управления control.ListBox.options; data - менеджер данных элемента управления control.ListBox.data; state - менеджер состояния элемента управления control.ListBox.state. <p>Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует на интерфейсные события макета и меняет значения полей control.ListBox.state в зависимости от полученных команд.</p>
parameters.options.maxWidth	[30, 65, 85]	<p>Тип: Number Number[]</p> <p>Максимальная ширина кнопки списка при различных состояниях. Если задано число, то считается, что кнопка имеет одинаковые максимальные размеры во всех состояниях. Если задан массив, то он будет трактоваться как максимальная ширина кнопки в различных состояниях - от меньшего к большему.</p>

Параметр	Значение по умолчанию	Описание
<code>parameters.options.panoramasItemMode</code>	'ifMercator'	<p>Тип: String</p> <p>Показывает или скрывает элемент "Панорамы". Возможные значения:</p> <ul style="list-style-type: none"> 'on' — элемент "Панорамы" всегда показывается. 'ifMercator' — элемент "Панорамы" показывается, только если проекция карты является проекцией меркатора. 'off' — элемент "Панорамы" никогда не показывается. <p>Элемент "Панорамы" доступен только при использовании стандартного макета.</p>
<code>parameters.options.position</code>	—	<p>Тип: Object</p> <p>Объект, описывающий позицию элемента управления. При указании опции position значение опции float автоматически трактуется как "none".</p>
<code>parameters.options.position.bottom</code>	'auto'	<p>Тип: Number String</p> <p>Положение относительно нижнего края карты.</p>
<code>parameters.options.position.left</code>	'auto'	<p>Тип: Number String</p> <p>Положение относительно левого края карты.</p>
<code>parameters.options.position.right</code>	'auto'	<p>Тип: Number String</p> <p>Положение относительно правого края карты.</p>
<code>parameters.options.position.top</code>	'auto'	<p>Тип: Number String</p> <p>Положение относительно верхнего края карты.</p>
<code>parameters.options.visible</code>	true	<p>Тип: Boolean</p> <p>Признак того, что элемент управления отображается.</p>

Параметр	Значение по умолчанию	Описание
parameters.state	—	Тип: Object Состояние элемента управления.
parameters.state.expanded	false	Тип: Boolean Признак, развернут ли список.

Пример:

```
map.controls.add(new ymaps.control.TypeSelector(['yandex#map', 'yandex#hybrid']));
```

Поля

Имя	Тип	Описание
data	data.Manager	Данные элемента управления.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от IControl .
state	data.Manager	Состояние выпадающего списка. Имена полей, доступных через метод data.Manager.get : <ul style="list-style-type: none"> <code>expanded</code> - признак, раскрыт ли список; <code>size</code> - размер, который в данный момент выставлен списку; Унаследовано от control.ListBox .

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от ICollection .
click	Щелчок по заголовку списка. Экземпляр класса Event . Унаследовано от control.ListBox .
collapse	Список закрыт. Экземпляр класса Event . Унаследовано от control.ListBox .
expand	Список открыт. Экземпляр класса Event . Унаследовано от control.ListBox .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
press	<p>Событие, сигнализирующее о том, что кнопка была нажата. В отличие от <code>click</code> генерируется только если состояние <code>isEnabled == true</code>. Экземпляр класса Event.</p> <p>Унаследовано от control.ListBox.</p>
remove	<p>Произошло удаление дочернего объекта.</p> <p>Унаследовано от ICollection.</p>

Методы

Имя	Возвращает	Описание
add(object)	ICollection	<p>Добавляет в коллекцию дочерний объект.</p> <p>Унаследован от ICollection.</p>
addMapType(mapType[, positionIndex])	control.TypeSelector	Добавляет тип карты в список.
collapse()	control.ListBox	<p>Сворачивает список.</p> <p>Унаследован от control.ListBox.</p>
expand()	control.ListBox	<p>Раскрывает список.</p> <p>Унаследован от control.ListBox.</p>
getIterator()	IIterator	<p>Возвращает итератор по коллекции.</p> <p>Унаследован от ICollection.</p>
getMap()	Map	Возвращает ссылку на карту.
getParent()	IControlParent null	<p>Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.</p> <p>Унаследован от IControl.</p>
isExpanded()	Boolean	<p>Возвращает признак того, находится ли контрол в развернутом состоянии.</p> <p>Унаследован от control.ListBox.</p>
remove(object)	ICollection	<p>Удаляет из коллекции дочерний объект.</p> <p>Унаследован от ICollection.</p>

Имя	Возвращает	Описание
<code>removeAllMapTypes()</code>	<code>control.TypeSelector</code>	Удаляет все типы карты из элемента управления.
<code>removeMapType(mapType)</code>	<code>control.TypeSelector</code>	Удаляет тип карты.
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IControl</code> .

Описание полей**data**

```
{data.Manager} data
```

Данные элемента управления.

Описание методов**addMapType**

```
{control.TypeSelector} addMapType(mapType[, positionIndex])
```

Добавляет тип карты в список.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>mapType *</code>	—	Тип: <code>String MapType</code> Тип карты.
<code>positionIndex</code>	—	Тип: <code>Integer</code> Позиция в списке (если не указана, новый тип карты добавляется в конец списка). Список значений <code>positionIndex</code> по умолчанию для стандартных типов карты: <ul style="list-style-type: none"> 'yandex#map' - 5; 'yandex#satellite' - 10; 'yandex#hybrid' - 15.

* Обязательный параметр/опция.

Примеры:

1.

```
var typeSelector = new ymaps.control.TypeSelector([]);
typeSelector.addMapType('yandex#map', 1);
typeSelector.addMapType('yandex#hybrid', 0);
```

2.

```
// Если используется стандартный набор типов карты, и мы
// хотим добавить свой из хранилища mapType.storage между
// типами «спутник» и «схема».
var typeSelector = myMap.controls.get('typeSelector');
typeSelector.addMapType('my#mapType', 6);
```

getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

removeAllMapTypes

```
{control.TypeSelector} removeAllMapTypes()
```

Удаляет все типы карты из элемента управления.

Возвращает ссылку на себя.

removeMapType

```
{control.TypeSelector} removeMapType(mapType)
```

Удаляет тип карты.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>mapType *</code>	—	Тип: String MapType Тип карты.

* Обязательный параметр/опция.

control.ZoomControl

Расширяет [IControl](#), [ICustomizable](#).

Элемент управления масштабом карты. Ключ элемента управления в хранилище `control.storage` — "zoomControl".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
control.ZoomControl([parameters])
```

Параметры:

Параметр	Значение по умолчанию	Описание
parameters	—	Тип: Object Параметры элемента управления.
parameters.data	—	Тип: Object Данные элемента управления.
parameters.options	—	Тип: Object Опции элемента управления.
parameters.options.adjustMapMargin	false	Тип: Boolean Регистрирует ли элемент управления свои размеры в менеджере отступов карты map.margin.Manager .
parameters.options.layout	—	Тип: IZoomControlLayout String Конструктор макета элемента управления или его ключ в хранилище layout.storage . В конструктор макета передается объект, содержащий поля: <ul style="list-style-type: none"> control - ссылка на элемент управления; options - менеджер опций элемента управления control.ZoomControl.options; data - менеджер данных элемента управления control.ZoomControl.data; state - менеджер состояния элемента управления control.ZoomControl.state.
parameters.options.position	—	Тип: Object Объект, описывающий позицию элемента управления.
parameters.options.position.bottom	'auto'	Тип: Number String Положение относительно нижнего края карты.

Параметр	Значение по умолчанию	Описание
<code>parameters.options.position.left</code>	10	Тип: Number String Положение относительно левого края карты.
<code>parameters.options.position.right</code>	'auto'	Тип: Number String Положение относительно правого края карты.
<code>parameters.options.position.top</code>	108	Тип: Number String Положение относительно верхнего края карты.
<code>parameters.options.size</code>	'auto'	Тип: String Параметр, отвечающий за внешний вид элемента управления. Может принимать значения: <ul style="list-style-type: none"> 'small' — всегда показывать маленький элемент управления масштабом карты; 'large' — всегда показывать большой элемент управления масштабом карты; 'auto' — производить автоматический выбор размера элемента управления в зависимости от высоты контейнера карты.
<code>parameters.options.visible</code>	true	Тип: Boolean Признак того, что элемент управления отображается.
<code>parameters.options.zoomDuration</code>	200	Тип: Number Время анимации при смене уровня масштабирования.
<code>parameters.options.zoomStep</code>	1	Тип: Number Шаг изменения уровня масштабирования карты.

Параметр	Значение по умолчанию	Описание
<code>parameters.state</code>	—	Тип: Object Объект, описывающий состояние элемента управления.

Примеры:**1.**

```
// Пример 1.
// Создадим элемент управления масштабом маленького размера и добавим его на карту.
var zoomControl = new ymaps.control.ZoomControl({
  options: {
    size: "small"
  }
});
myMap.controls.add(zoomControl);
```

2.

```
// Пример 2
// Добавление большого элемента управления масштабом карты по ключу из control.storage.
myMap.controls.add('zoomControl', {
  size: "large"
});
```

Поля

Имя	Тип	Описание
<code>events</code>	<code>IEventManager</code>	Менеджер событий. Унаследовано от <code>IEventEmitter</code> .
<code>options</code>	<code>IOptionManager</code>	Менеджер опций. Унаследовано от <code>IControl</code> .

События

Имя	Описание
<code>optionschange</code>	Изменение в опциях объекта. Унаследовано от <code>ICustomizable</code> .
<code>parentchange</code>	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от <code>ICChild</code> .

Методы

Имя	Возвращает	Описание
<code>getMap()</code>	<code>Map</code>	Возвращает ссылку на карту.

Имя	Возвращает	Описание
<code>getParent()</code>	<code>IControlParent</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IControl .
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IControl .

Описание методов**getMap**

```
{Map} getMap()
```

Возвращает ссылку на карту.

coordSystem**coordSystem.cartesian**

Статический объект.

Экземпляр класса [ICoordSystem](#)

Класс, описывающий геометрию декартовой плоскости. Используется как координатная система по умолчанию при построении геодезических линий в нестандартных проекциях.

[Методы](#)**Методы**

Имя	Возвращает	Описание
<code>getDistance(point1, point2)</code>	Number	Возвращает кратчайшее (вдоль геодезической линии) расстояние между двумя заданными точками (в метрах).

Имя	Возвращает	Описание
<code>solveDirectProblem(startPoint, direction, distance)</code>	Object	<p>Решает так называемую первую (прямую) геодезическую задачу: где мы окажемся, если выйдем из указанной точки в указанном направлении и пройдем, не сворачивая, указанное расстояние. Решением прямой геодезической задачи являются следующие данные:</p> <ul style="list-style-type: none"> • конечная точка, • конечное направление, • функция пути, • функция, позволяющая для произвольного момента времени указать, в какой точке мы будем находиться и в каком направлении двигаться.
<code>solveInverseProblem(startPoint, endPoint[, reverseDirection])</code>	Object	<p>Решает так называемую вторую (обратную) геодезическую задачу: построить кратчайший маршрут между двумя точками на картографируемой поверхности и определить расстояние и направление движения. Обратите внимание: на карте земной поверхности кратчайшие маршруты изображаются кривыми линиями. Для геообъектов в API можно включить режим отображения кратчайших расстояний между точками с помощью опции <code>geodesic</code>.</p>

coordSystem.geo

Статический объект.

Экземпляр класса `ICoordSystem`

Объект, описывающий геометрию земной поверхности. Позволяет строить кратчайшие маршруты (геодезические линии) между точками земной поверхности и находить расстояния. Используется для отображения геообъектов с использованием геодезических линий.

Методы

Методы

Имя	Возвращает	Описание
<code>getDistance(point1, point2)</code>	Number	Возвращает кратчайшее (вдоль геодезической линии) расстояние между двумя заданными точками (в метрах).
<code>solveDirectProblem(startPoint, direction, distance)</code>	Object	<p>Решает так называемую первую (прямую) геодезическую задачу: где мы окажемся, если выйдем из указанной точки в указанном направлении и пройдем, не сворачивая, указанное расстояние. Решением прямой геодезической задачи являются следующие данные:</p> <ul style="list-style-type: none"> • конечная точка, • конечное направление, • функция пути, • функция, позволяющая для произвольного момента времени указать, в какой точке мы будем находиться и в каком направлении двигаться.
<code>solveInverseProblem(startPoint, endPoint[, reverseDirection])</code>	Object	<p>Решает так называемую вторую (обратную) геодезическую задачу: построить кратчайший маршрут между двумя точками на картографируемой поверхности и определить расстояние и направление движения. Обратите внимание: на карте земной поверхности кратчайшие маршруты изображаются кривыми линиями. Для геообъектов в API можно включить режим отображения кратчайших расстояний между точками с помощью опции <code>geodesic</code>.</p>

data

data.Manager

Расширяет [IDataManager](#), [IFreezable](#).

Менеджер произвольных данных.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
data.Manager(data)
```

Параметры:

Параметр	Значение по умолчанию	Описание
data *	—	Тип: Object Данные.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий объекта. Унаследовано от IFreezable .

События

Имя	Описание
change	Изменение внутреннего состояния объекта. Унаследовано от IFreezable .

Методы

Имя	Возвращает	Описание
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable .
get(path[, defaultValue])	Object	Возвращает значение поля данных с заданным именем.
getAll()	Object	Возвращает объект, содержащий все поля данных.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .

Имя	Возвращает	Описание
<code>set(path[, value])</code>	<code>data.Manager</code>	<p>Устанавливает значения для указанных полей. Поддерживаются две сигнатуры:</p> <ul style="list-style-type: none"> один аргумент, представляющий собой объект вида {"имя": "значение"}; два аргумента, первый из которых — имя поля, второй — значение. <p>Имя может адресовать вложенные поля, т.е. содержать '.'.</p>
<code>setAll()</code>	<code>data.Manager</code>	<p>Полностью перезаписывает все поля данных. Равносильно последовательному вызову методов <code>unsetAll</code> и <code>set</code>, но обеспечивает большую производительность.</p>
<code>unfreeze()</code>	<code>IFreezable</code>	<p>Переводит объект в активный режим.</p> <p>Унаследован от <code>IFreezable</code>.</p>
<code>unset(path)</code>	<code>data.Manager</code>	Очищает заданные поля данных.
<code>unsetAll()</code>	<code>data.Manager</code>	Очищает все поля данных.

Описание методов

get

```
{Object} get(path[, defaultValue])
```

Возвращает значение поля данных с заданным именем.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>path *</code>	—	<p>Тип: String</p> <p>Строка с именем поля данных. Имя может адресовать вложенные поля, т.е. содержать '.'.</p>

Параметр	Значение по умолчанию	Описание
<code>defaultValue</code>	—	Тип: Object Значение по умолчанию.

* Обязательный параметр/опция.

getAll

```
{Object} getAll()
```

Возвращает объект, содержащий все поля данных.

set

```
{data.Manager} set(path[, value])
```

Устанавливает значения для указанных полей. Поддерживаются две сигнатуры:

- один аргумент, представляющий собой объект вида {"имя": "значение"};
- два аргумента, первый из которых — имя поля, второй — значение.

Имя может адресовать вложенные поля, т.е. содержать '.'.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>path</code> *	—	Тип: Object string Строка, содержащая имя поля данных, либо объект вида {"имя поля": "значение"}.
<code>value</code>	—	Тип: Object Значение, если в качестве первого аргумента передается строка, содержащая имя поля.

* Обязательный параметр/опция.

Пример:

```
var balloonData = new ymaps.data.Manager({  
  balloonContentHeader: 'Заголовок балуна.'  
});  
balloonData.set({  
  balloonContentBody: 'Содержимое балуна.',  
  balloonContentFooter: '<a href="//ya.ru">test</a>'  
});
```

setAll

```
{data.Manager} setAll()
```

Полностью перезаписывает все поля данных. Равносильно последовательному вызову методов `unsetAll` и `set`, но обеспечивает большую производительность.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>data *</code>	—	Тип:

* Обязательный параметр/опция.

unset

```
{data.Manager} unset(path)
```

Очищает заданные поля данных.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>path *</code>	—	Тип: String String[] Имя поля, либо массив имён.

* Обязательный параметр/опция.

unsetAll

```
{data.Manager} unsetAll()
```

Очищает все поля данных.

Возвращает ссылку на себя.

domEvent

domEvent.manager

Статический объект.

Предоставляет единый интерфейс для работы с событиями DOM-элементов во всех браузерах и на всех устройствах. Для устройств, которые не поддерживают мышь, события будут транслироваться.

- Событие начала касания (touchstart/pointerdown) с одной точкой прикосновения транслируется в последовательность событий mouseenter, mousemove и mousedown;
- Событие перемещения касания (touchmove/pointermove) с одной точкой прикосновения транслируется в событие mousemove;
- События завершения касания (touchend/pointerup) или отмены (touchcancel/pointercancel) транслируются в последовательность событий 'mouseup', 'mousemove' и 'mouseleave', если ранее произошло событие начала касания с одной точкой прикосновения;
- Быстрое поступление событий начала и завершения с одной точкой прикосновения без перемещения транслируется в событие click;
- Быстрое поступление двух событий click транслируется в событие dblclick;
- Если между событиями начала и завершения с одной точкой прикосновения без перемещения была продолжительная задержка, то это транслируется в событие contextmenu.

Также поддерживаются специальные события для обработки нескольких одновременных касаний.

- multitouchstart отправляется при поступлении события начала касания с двумя и более точками прикосновений;
- multitouchmove отправляется при поступлении события перемещения касания с двумя и более точками прикосновений;
- multitouchend отправляется при поступлении события завершения касания, если ранее было отправлено событие multitouchstart;
- При добавлении/удалении точки прикосновения будет отправлено событие multitouchend и событие multitouchstart, если оставшееся количество точек больше или равно двум.

Менеджер работы с событиями DOM-элементов.

Методы

Примеры:

1.

```
// Прослушивание событий одного элемента DOM.
var block = document.getElementById('block');
ymaps.domEventManager
    .add(block, 'click', function (event) {
        // событие 'click'.
        console.log(event.get('type'));
    })
    .add(block, 'mouseleave', function (event) {
        // событие 'mouseleave'.
        console.log(event.get('type'));
    })
// Установка одного слушателя для нескольких событий.
.add(block, ['mousedown', 'mouseup'], function (event) {
    // события 'mousedown' / 'mouseup'.
    console.log(event.get('type'));
});
```

2.

```
// Использование контейнера событий.
var block = document.getElementById('block'),
    domEventsGroup = ymaps.domEvent.manager.group(block);
domEventsGroup.add('click', function (event) {
    console.log(event.get('type')); // click
    // Удаляем все прослушивания событий.
    domEventsGroup.removeAll();
});
```

3.

```
// Выполняем listener в контексте конкретного объекта.
var block = document.getElementById('block');
// Задаем класс.
var someClass = function () {
    this.property = 'value';
};
// Создаем класс.
var someObj = new someClass();
ymaps.domEvent.manager.add(block, 'click', function(event) {
    // Выводит 'click'.
    console.log(this.property + ' ' + event.get('type'));
}, someObj);
```

4.

```
// На устройствах, которые поддерживают сенсорный ввод, можно прослушивать специальные события multitouch*
var block = document.getElementById('block');
ymaps.domEvent.manager
    .add(block, ['multitouchstart', 'multitouchmove', 'multitouchend'], function (event) {
        console.log(event.get('type')); // multitouchstart / multitouchmove / multitouchend
        // Запрещаем сдвиг и масштабирование страницы пальцами.
        event.callMethod('preventDefault');
    });
```

Методы

Имя	Статический	Возвращает	Описание
<code>domEvent.manager.add(htmlElement, types, callback[, context[, capture]])</code>		<code>domEvent.manager</code>	Добавляет слушателя DOM-событий объекта.
<code>domEvent.manager.group(htmlElement[, capture])</code>		<code>event.Group</code>	Возвращает группу слушателей событий для заданного DOM-элемента.
<code>domEvent.manager.remove(htmlElement, types, callback[, context[, capture]])</code>		<code>domEvent.manager</code>	Удаляет слушателя DOM-событий объекта.

Описание методов**add**

```
{domEvent.manager} <static> domEvent.manager.add(htmlElement, types, callback[, context[, capture]])
```

Добавляет слушателя DOM-событий объекта.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>htmlElement *</code>	—	Тип: HTMLElement Document DOM-элемент, события которого необходимо прослушивать.
<code>types *</code>	—	Тип: String String[] Тип или типы событий.
<code>callback *</code>	—	Тип: Function Функция-обработчик события.
<code>context</code>	—	Тип: Object Контекст выполнения функции-обработчика.
<code>capture</code>	—	Тип: Boolean Признак того, что событие необходимо отслеживать на фазе захвата.

* Обязательный параметр/опция.

group

```
{event.Group} <static> domEvent.manager.group(htmlElement[, capture])
```

Возвращает группу слушателей событий для заданного DOM-элемента.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>htmlElement *</code>	—	Тип: HTMLElement Document DOM-элемент.
<code>capture</code>	—	Тип: Boolean Признак того, что событие необходимо отслеживать на фазе захвата.

* Обязательный параметр/опция.

remove

```
{domEvent.manager} <static> domEvent.manager.remove(htmlElement, types, callback[, context[, capture]])
```

Удаляет слушателя DOM-событий объекта.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>htmlElement *</code>	—	Тип: HTMLElement Document DOM-элемент, события которого прослушиваются.
<code>types *</code>	—	Тип: String String[] Тип или типы событий.
<code>callback *</code>	—	Тип: Function String Функция-обработчик события либо уникальный id пары callback-context.
<code>context</code>	—	Тип: Object Контекст выполнения функции-обработчика.
<code>capture</code>	—	Тип: Boolean Признак того, что событие необходимо отслеживать на фазе захвата.

* Обязательный параметр/опция.

domEvent.MultiPointer

Расширяет IMultiTouchEvent.

Объект, описывающий multitouch-событие, которое было вызвано несколькими событиями PointerEvent. Предоставляет прокси-методы для доступа к полям и методам DOM-события.

[Конструктор](#) | [Методы](#)

Конструктор

```
domEvent.MultiPointer(originalEvent[, type])
```

Создаёт объект, описывающий multitouch-событие.

Параметры:

Параметр	Значение по умолчанию	Описание
originalEvent *	—	Тип: Object pointer-событие.
type	—	Тип: String Тип события. Если не указан, считается равным 'multi' + originalEvent.type .

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
get(name)	Object	Возвращает значение свойства оригинального события. Значения свойств кэшируются.

Описание методов

get

```
{Object} get(name)
```

Возвращает значение свойства оригинального события. Значения свойств кэшируются.

Параметры:

Параметр	Значение по умолчанию	Описание
name *	—	Тип: String Имя свойства.

* Обязательный параметр/опция.

domEvent.MultiTouch

Расширяет IMultiTouchEvent.

Объект-событие. Предоставляет прокси-методы для доступа к полям и методам DOM-события. Возвращаемые значения нормализованы с учётом особенностей различных браузеров.

[Конструктор](#) | [Методы](#)

Конструктор

```
domEvent.MultiTouch(originalEvent[, type])
```

Создаёт объект-событие, описывающий multitouch-событие.

Параметры:

Параметр	Значение по умолчанию	Описание
originalEvent *	—	Тип: Object multitouch-событие.
type	—	Тип: String Тип события. Если не указан, считается равным 'multi' + originalEvent.type.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
get(name)	Object	Возвращает значение свойства оригинального события. Значения свойств кэшируются.

Описание методов

get

```
{Object} get(name)
```

Возвращает значение свойства оригинального события. Значения свойств кэшируются.

Параметры:

Параметр	Значение по умолчанию	Описание
name *	—	Тип: String Имя свойства.

* Обязательный параметр/опция.

domEvent.Pointer

Расширяет IMultiTouchEvent.

Объект-событие. Предоставляет прокси-методы для доступа к полям и методам DOM-события (единочное касание экрана).

[Конструктор](#) | [Методы](#)

Конструктор

```
domEvent.Pointer(originalEvent[, type])
```

Создаёт объект, описывающий pointer-событие.

Параметры:

Параметр	Значение по умолчанию	Описание
originalEvent *	—	Тип: Object DOM-событие.
type	—	Тип: String Тип события. Если не указан, берётся из originalEvent.type.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
get(name)	Object	Возвращает значение свойства оригинального события. Значения свойств кэшируются.

Описание методов

get

```
{Object} get(name)
```

Возвращает значение свойства оригинального события. Значения свойств кэшируются.

Параметры:

Параметр	Значение по умолчанию	Описание
name *	—	Тип: String Имя свойства.

* Обязательный параметр/опция.

domEvent.Touch

Расширяет IMultiTouchEvent.

Объект-событие. Предоставляет прокси-методы для доступа к полям и методам DOM-события. Возвращаемые значения нормализованы с учётом особенностей различных браузеров.

[Конструктор](#) | [Методы](#)

Конструктор

```
domEvent.Touch(originalEvent[, type])
```

Создаёт объект-событие, описывающий touch-событие (одиночное касание экрана).

Параметры:

Параметр	Значение по умолчанию	Описание
<code>originalEvent</code> *	—	Тип: Object DOM-событие.
<code>type</code>	—	Тип: String Тип события. Если не указан, берётся из <code>originalEvent.type</code> .

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
<code>get(name)</code>	Object	Возвращает значение свойства оригинального события. Значения свойств кэшируются.

Описание методов

get

```
{Object} get(name)
```

Возвращает значение свойства оригинального события. Значения свойств кэшируются.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>name</code> *	—	Тип: String Имя свойства.

* Обязательный параметр/опция.

DomEvent

Расширяет [IDomEvent](#).

DOM-событие в системе API Яндекс.Карт. Предоставляет прокси-методы для доступа к полям и методам исходного DOM-события. Возвращаемые значения нормализованы с учётом особенностей различных браузеров. Также переопределено свойство `position`, возвращается массив вида `[pageX, pageY]`.

[Конструктор](#) | [Методы](#)

Конструктор

```
DomEvent(originalEvent[, type])
```

Создаёт DOM-событие в системе API Яндекс.Карт.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>originalEvent</code> *	—	Тип: DomEvent DOM-событие.
<code>type</code>	—	Тип: Object Тип события. Если не указан, берётся из <code>originalEvent.type</code> .

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
<code>allowMapEvent()</code>		Разрешает распространение данного события на карту. Унаследован от IEvent .
<code>callMethod(name)</code>		Вызывает указанный метод у исходного события. Второй и последующие аргументы передаются методу при вызове. Унаследован от IEvent .
<code>get(name)</code>	Object	Возвращает значение некоторого свойства. Сначала проверяется, не было ли свойство задано через <code>set</code> , затем проверяется наличие свойства в <code>domEvent.overrideStorage</code> . Если не найдено — то проверяется в <code>originalEvent</code> . Значения свойств кэшируются.
<code>getSourceEvent()</code>	IDomEvent	Возвращает исходное DOM-событие. Унаследован от IDomEvent .
<code>isDefaultPrevented()</code>	Boolean	Возвращает true, если реакция по умолчанию на событие отменена. Унаследован от IEvent .
<code>isImmediatePropagationStopped()</code>	Boolean	Возвращает true, если обработка события прервана. Унаследован от IEvent .

Имя	Возвращает	Описание
isMapEventAllowed()	Boolean	Возвращает true, если событие карты разрешено. Унаследован от IEvent .
isPropagationStopped()	Boolean	Возвращает true, если распространение события прервано, Унаследован от IEvent .
preventDefault()		Отменяет реакцию по умолчанию на событие в рамках системы событий API Яндекс.Карт. Вызов этого метода не влияет на обработку браузером действия по умолчанию исходного DOM-события. Унаследован от IDomEvent .
stopImmediatePropagation()		Прекращает обработку события в системе событий API Яндекс.Карт. Т.е. после вызова данного метода ни один обработчик данного события не будет вызван. Вызов этого метода не влияет на обработку исходного DOM-события на уровне браузера. Унаследован от IDomEvent .
stopPropagation()		Прекращает распространение DOM-события в системе событий API Яндекс.Карт. Вызов этого метода не влияет на распространение исходного DOM-события по DOM-дереву. Унаследован от IDomEvent .

Описание методов

get

```
{Object} get(name)
```

Возвращает значение свойства. Сначала проверяется, не было ли свойство задано через set, затем проверяется наличие свойства в domEvent.overrideStorage. Если не найдено — то проверяется в originalEvent. Значения свойств кэшируются.

Параметры:

Параметр	Значение по умолчанию	Описание
name *	—	Тип: String Имя свойства.

* Обязательный параметр/опция.

event

event.Group

Расширяет [IEventManager](#).

Группа слушателей событий.

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
event.Group(events)
```

Создает группу слушателей событий.

Параметры:

Параметр	Значение по умолчанию	Описание
events *	—	Тип: IEventManager Менеджер событий, для которого создана группа.

* Обязательный параметр/опция.

Пример:

```
// Создадим группу слушателей.  
var listeners = events.group()  
  .add('click', function () {  
    alert('click');  
  })  
  .add('dblclick', function () {  
    alert('dblclick');  
  })  
  .add('dblclick', function () {  
    alert('dblclick');  
  });  
// ...  
// Когда хранящиеся в контейнере обработчики событий  
// нам больше не нужны, просто очистим группу.  
listeners.removeAll();
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий, для которого создана группа.

Методы

Имя	Возвращает	Описание
add(types, callback[, context[, priority]])	IEventManager	Добавляет слушатель события. Унаследован от IEventManager .
getLength()	Integer	Возвращает текущее количество подписчиков в группе.
remove(types, callback[, context[, priority]])	IEventManager	Удаляет слушатель события из группы. Унаследован от IEventManager .

Имя	Возвращает	Описание
<code>removeAll()</code>	<code>IEventGroup</code>	Удаляет всех слушателей событий из группы. Унаследован от <code>IEventGroup</code> .

Описание полей

events

```
{IEventManager} events
```

Менеджер событий, для которого создана группа.

Описание методов

getLength

```
{Integer} getLength()
```

Возвращает текущее количество подписчиков в группе.

event.Manager

Расширяет `IEventManager`.

Менеджер событий. При помощи менеджера событий можно подписываться на события и отписываться от них, а также инициировать сами события. Менеджер реализует возможность построения иерархии распространения событий с помощью метода `event.Manager.setParent`.

Распространение события происходит в три фазы:

- 1. Событие получают непосредственные подписчики;
- 2. Событие получают вышестоящие в иерархии объекты посредством перебрасывания события на родительском менеджере событий;
- 3. Событие получают обработчики действия по умолчанию посредством служебного события `type + 'defaultaction'`; действие по умолчанию выполняется, только если поле `target` события совпадает с контекстом менеджера событий.

Также менеджер позволяет добавлять обработчики событий с указанием приоритета. При бросании событий обработчики будут вызваны в порядке убывания приоритета.

Подписки с одинаковыми параметрами `callback`, `context`, но с разными приоритетами считаются разными. При удалении подписки необходимо указывать тот же приоритет, что и при добавлении.

[Конструктор](#) | [Методы](#)

Конструктор

```
event.Manager([params])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>params</code>	—	Тип: <code>Object</code> Параметры менеджера событий.

Параметр	Значение по умолчанию	Описание
<code>params.context</code>	—	Тип: <code>Object</code> Контекстный объект менеджера событий.
<code>params.controllers</code>	—	Тип: <code>IEventWorkflowController[]</code> Контроллер или контроллеры менеджера событий.
<code>params.parent</code>	—	Тип: <code>IEventManager</code> Родительский менеджер событий.

Методы

Имя	Возвращает	Описание
<code>add(types, callback[, context[, priority]])</code>	<code>IEventManager</code>	Добавляет новую подписку. Унаследован от <code>IEventManager</code> .
<code>createEventObject(type, event, target)</code>	<code>Event</code>	Функция, которая создает объект-событие. Вызывается в методе <code>fire</code> , когда переданный объект не является экземпляром класса <code>Event</code> или его наследником.
<code>fire(type[, event])</code>	<code>event.Manager</code>	Бросает событие.
<code>getParent()</code>	<code>IEventManager</code> <code>null</code>	Возвращает ссылку на родительский менеджер событий. Унаследован от <code>IEventManager</code> .
<code>group()</code>	<code>IEventGroup</code>	Возвращает группу слушателей событий, ассоциированную с данным менеджером событий. Унаследован от <code>IEventManager</code> .
<code>once(types, callback[, context[, priority]])</code>	<code>IEventManager</code>	Добавляет слушателя, который вызовет функцию-обработчик только 1 раз. Унаследован от <code>IEventManager</code> .

Имя	Возвращает	Описание
<code>remove(types, callback[, context[, priority]])</code>	<code>IEventManager</code>	Удаляет существующую подписку. Унаследован от <code>IEventManager</code> .
<code>setParent(parent)</code>	<code>event.Manager</code>	Устанавливает родительский менеджер событий.

Описание методов

createEventObject

```
{Event} createEventObject(type, event, target)
```

Функция, которая создает объект-событие. Вызывается в методе `fire`, когда переданный объект не является экземпляром класса `Event` или его наследником.

Возвращает объект-событие.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>type *</code>	—	Тип: String Тип события.
<code>event *</code>	—	Тип: Object Объект, описывающий событие.
<code>target *</code>	—	Тип: Object Объект, на котором произошло событие.

* Обязательный параметр/опция.

fire

```
{event.Manager} fire(type[, event])
```

Бросает событие.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>type *</code>	—	Тип: String Тип события.

Параметр	Значение по умолчанию	Описание
event	—	Тип: Object Event Событие. Если передан хэш с данными, то для него будет вызван метод <code>createEventObject</code> , и дальнейшие действия будут производиться с новым созданным событием.

* Обязательный параметр/опция.

setParent

```
{event.Manager} setParent(parent)
```

Устанавливает родительский менеджер событий.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
parent *	—	Тип: IEventManager null Родительский менеджер событий.

* Обязательный параметр/опция.

event.Mapper

Расширяет [IEventTrigger](#).

Маппер событий. Позволяет управлять распространением событий по иерархии менеджеров событий.

[Конструктор](#) | [Методы](#)

Конструктор

```
event.Mapper(targetEventManager, mappingTable)
```

Параметры:

Параметр	Значение по умолчанию	Описание
targetEventManager *	—	Тип: IEventManager Менеджер событий, на который маппер распространяет иницилируемые на нем события.

Параметр	Значение по умолчанию	Описание
<code>mappingTable</code> *	—	<p>Тип: Object</p> <p>Таблица правил маппинга. Представляет собой хэш, ключами которого являются типы событий, а значениями - соответствующие функции маппинга, либо Boolean значения. Функция маппинга для конкретного типа событий получает на вход экземпляр события, инициированного на маппере, а возвращать должна экземпляр события для распространения далее по иерархии, либо null, если распространение необходимо запретить. Boolean значения трактуются следующим образом:</p> <ul style="list-style-type: none"> • true - события данного типа распространяются по иерархии в неизменном виде; • false - события данного типа по иерархии не распространяются. <p>Дополнительно в таблице доступен ключ "*" для правила обработки по умолчанию.</p>

* Обязательный параметр/опция.

Пример:

```
// Создаем и устанавливаем маппер событий, который преобразует событие "click" корневой
// коллекции геообъектов в событие "geoobjectclick" самой карты.
var mapper = new ymaps.event.Mapper(myMap.events, {
  "*": false,
  "click": function (event) {
    return new ymaps.Event({
      type: "geoobjectclick",
      target: map,
      originalTarget: event.get("target")
    }, event);
  }
});
myMap.geoObjects.events.setParent(mapper);
```

Методы

Имя	Возвращает	Описание
<code>fire(type[, eventObject])</code>	<code>IEventTrigger</code>	<p>Иницирует событие.</p> <p>Унаследован от <code>IEventTrigger</code>.</p>

Event

Расширяет `IEvent`.

Событие. Предоставляет методы для доступа к полям и методам объекта `originalEvent` с возможностью их переопределения.

[Конструктор](#) | [Методы](#)

Конструктор

```
Event(originalEvent[, sourceEvent])
```

Создает событие.

Параметры:

Параметр	Значение по умолчанию	Описание
originalEvent *	—	Тип: Object Исходные данные.
sourceEvent	—	Тип: IEvent Исходное событие.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
allowMapEvent()		Разрешает распространение данного события на карту. Унаследован от IEvent .
callMethod(name)	Object	Вызывает указанный метод. Операция эквивалентна поиску поля через <code>get</code> и вызову с передачей в качестве контекста <code>originalEvent</code> . Все аргументы после первого передаются в качестве параметров вызываемому методу.
get(name)	Object	Возвращает значение поля из <code>originalEvent</code> . <code>originalEvent</code> всегда имеет следующие поля: <ul style="list-style-type: none"><code>type</code> - строковый тип события;<code>target</code> - ссылка на объект, породивший событие.
getSourceEvent()	IEvent null	Возвращает исходное событие. Унаследован от IEvent .

Имя	Возвращает	Описание
isDefaultPrevented()	Boolean	Проверяет, отменена ли реакция по умолчанию на событие в рамках системы событий API Яндекс.Карт.
isImmediatePropagationStopped()	Boolean	Определяет, прекращено ли распространение события в системе событий API Яндекс.Карт.
isMapEventAllowed()	Boolean	Возвращает true, если событие карты разрешено. Унаследован от IEvent .
isPropagationStopped()	Boolean	Проверяет, остановлено ли распространение события по иерархии объектов и коллекций в системе событий API Яндекс.Карт.
preventDefault()		Отменяет реакцию по умолчанию на событие в рамках системы событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву.
stopImmediatePropagation()		Прекращает распространение события в системе событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву.
stopPropagation()		Прекращает распространение события по иерархии объектов и коллекций в системе событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву.

Описание методов

callMethod

```
{Object} callMethod(name)
```

Вызывает указанный метод. Операция эквивалентна поиску поля через `get` и вызову с передачей в качестве контекста `originalEvent`. Все аргументы после первого передаются в качестве параметров вызываемому методу.

Возвращает значение.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>name *</code>	—	Тип: String Имя метода.

* Обязательный параметр/опция.

get

```
{Object} get(name)
```

Возвращает значение поля из `originalEvent`. `originalEvent` всегда имеет следующие поля:

- `type` - строковый тип события;
- `target` - ссылка на объект, породивший событие.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>name *</code>	—	Тип: String Имя свойства.

* Обязательный параметр/опция.

Пример:

```
// Синхронизируем два объекта между собой.  
object1.events.add(["add", "remove"], function (event) {  
    object2[event.get("type")](event.get("child"));  
});
```

isDefaultPrevented

```
{Boolean} isDefaultPrevented()
```

Проверяет, отменена ли реакция по умолчанию на событие в рамках системы событий API Яндекс.Карт.

Возвращает `true`, если реакция на событие по умолчанию отменена, `false` - в противном случае.

isImmediatePropagationStopped

```
{Boolean} isImmediatePropagationStopped()
```

Определяет, прекращено ли распространение события в системе событий API Яндекс.Карт.

Возвращает `true` - распространение прекращено, `false` - нет

isPropagationStopped

```
{Boolean} isPropagationStopped()
```

Проверяет, остановлено ли распространение события по иерархии объектов и коллекций в системе событий API Яндекс.Карт.

Возвращает `true` - распространение по иерархии отменено, `false` - нет.

preventDefault

```
{ } preventDefault()
```

Отменяет реакцию по умолчанию на событие в рамках системы событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву.

stopImmediatePropagation

```
{ } stopImmediatePropagation()
```

Прекращает распространение события в системе событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву.

stopPropagation

```
{ } stopPropagation()
```

Прекращает распространение события по иерархии объектов и коллекций в системе событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву.

findOrganization

Статическая функция.

Находит организацию по идентификатору. Результат предоставляется в виде объекта [@GeoObject](#).

Возвращает объект-promise.

```
{ vow.Promise } findOrganization(organizationId)
```

Параметры:

Параметр	Значение по умолчанию	Описание
organizationId *	—	Тип: String Идентификатор организации.

* Обязательный параметр/опция.

Пример:

```
// Выполняет поиск организации с идентификатором "1124715036".  
// Результат добавляется на карту, и его балун открывается.  
ymaps.findOrganization('1124715036').then(  
  function (orgGeoObject) {  
    map.geoObjects.add(orgGeoObject);  
    orgGeoObject.balloon.open();  
  },  
  function (err) {  
    // обработка ошибок  
  }  
);
```

formatter

Статический объект.

Статический класс, содержащий методы форматирования единиц измерения с учётом текущего языка.

[Методы](#)

Методы

Имя	Возвращает	Описание
<code>distance(value[, significantDigits])</code>	String	Возвращает строковое представление расстояния, отформатированное согласно локальным правилам и переведённое в нужную систему единиц измерения (например, для локали en_US расстояние будет представлено в милях).
<code>duration(value[, significantDigits])</code>	String	Возвращает строковое представление интервала времени.

Описание методов**distance**

```
{String} distance(value[, significantDigits])
```

Возвращает строковое представление расстояния, отформатированное согласно локальным правилам и переведённое в нужную систему единиц измерения (например, для локали en_US расстояние будет представлено в милях).

Параметры:

Параметр	Значение по умолчанию	Описание
<code>value *</code>	—	Тип: Number Длина в метрах.
<code>significantDigits</code>	2	Тип: Integer Число значащих цифр в ответе.

* Обязательный параметр/опция.

duration

```
{String} duration(value[, significantDigits])
```

Возвращает строковое представление интервала времени.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>value *</code>	—	Тип: Number Интервал времени в секундах.

Параметр	Значение по умолчанию	Описание
significantDigits	2	Тип: Integer Число значащих цифр в ответе.

* Обязательный параметр/опция.

geocode

Статическая функция.

Обработывает запросы геокодирования. Результат запроса может быть представлен в формате JSON или в виде объекта [GeoObjectCollection](#). Формат ответа геокодера описан в разделе [Геокодирование](#).

См.: [GeocodeResult](#)

Возвращает объект-promise.

```
{ vow.Promise } geocode(request[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
request *	—	Тип: String Number[] Адрес, для которого необходимо определить координаты (прямое геокодирование) или координаты, для которых необходимо определить адрес (обратное геокодирование).
options	—	Тип: Object Опции.
options.boundedBy	—	Тип: Number[][] Прямоугольная область на карте, где предположительно находится искомый объект.
options.json	false	Тип: Boolean Если true, то в функцию-обработчик передается JSON. В противном случае в функцию обработчик передается объект, содержащий в поле geoObjects результаты геокодирования в виде коллекции GeoObjectCollection . При геокодировании посредством геокодера 'yandex#map' в коллекции будут находиться объекты GeocodeResult .

Параметр	Значение по умолчанию	Описание
<code>options.kind</code>	'house'	<p>Тип: String</p> <p>Вид топонима (только для обратного геокодирования).</p> <p>Список возможных значений:</p> <ul style="list-style-type: none"> house - дом; street - улица; metro - станция метро; district - район города; locality - населенный пункт (город/поселок/деревня/село/...).
<code>options.provider</code>	'yandex#map'	<p>Тип: <code>IGeocodeProvider</code> String</p> <p>Провайдер геокодирования. Можно воспользоваться одним из стандартных провайдеров:</p> <ul style="list-style-type: none"> 'yandex#map' - поиск по карте.
<code>options.results</code>	10	<p>Тип: Integer</p> <p>Максимальное количество возвращаемых результатов.</p>
<code>options.searchCoordOrder</code>	—	<p>Тип: String</p> <p>Определяет каким образом нужно интерпретировать координаты в запросе.</p>
<code>options.skip</code>	0	<p>Тип: Integer</p> <p>Число результатов, которое необходимо пропустить.</p>
<code>options.strictBounds</code>	false	<p>Тип: Boolean</p> <p>Искать только внутри области, заданной опцией <code>boundedBy</code>.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
// Осуществляет поиск объекта с именем "Москва".
// Полученный результат сразу отображается на карте.
var myGeocoder = ymaps.geocode("Moscow");
myGeocoder.then(
  function (res) {
    map.geoObjects.add(res.geoObjects);
    // Выведем в консоль данные, полученные в результате геокодирования объекта.
    console.log(res.geoObjects.get(0).properties.get('metaDataProperty').getAll());
  },
  function (err) {
    console.log(err);
  }
);
```

```
function (err) {
    // Обработка ошибки.
}
);
```

2.

```
// Реализует интерфейс IGeocodeProvider.
var randomPointProvider = {
    geocode: function (request, options) {
        var deferred = ymaps.vow.defer(),
            geoObjects = new ymaps.GeoObjectCollection(),
            results = options.results || 10;

        for (var i = 0; i < results; i++) {
            geoObjects.add(new ymaps.GeoObject({
                geometry: {
                    type: "Point",
                    coordinates: [(Math.random() - 0.5) * 180, (Math.random() - 0.5) * 360]
                },
                properties: {
                    name: request + ' ' + i,
                    description: request + '\s description ' + i,
                    balloonContentBody: '<p>' + request + ' ' + i + '</p>'
                }
            }));
        }

        var response = {
            geoObjects: geoObjects, // search output geo objects
            // Ответная метайнформация.
            metaData: {
                geocoder: {
                    request: request, // processed request string
                    found: results, // number of results found
                    results: results, // number of results returned
                    skip: options.skip || 0 // number of results skipped
                }
            }
        };

        // Асинхронная обработка.
        setTimeout(function () {
            deferred.resolve(response);
        }, 0);

        return deferred.promise();
    }
};

var myGeocoder = ymaps.geocode("Moscow", { provider: randomPointProvider });

myGeocoder.then(
    function (res) {
        map.geoObjects.add(res.geoObjects);
    },
    function (err) {
        // handling errors
    }
);
```

GeocodeResult

Расширяет [IGeoObject](#).

Результат геокодирования.

См.: [Placemark geocode geolocation control.SearchControl](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
GeocodeResult()
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .

Имя	Тип	Описание
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getAddressLine()	String	Возвращает строку с адресом объекта.

Имя	Возвращает	Описание
getAdministrativeAreas()	String[]	Возвращает административно-территориальные образования, в которые входит объект (федеральный округ, регион, район), не больше двух.
getCountry()	String null	Возвращает государство, которому принадлежит топоним (если применимо).
getCountryCode()	String null	Возвращает код государства, которому принадлежит топоним (если применимо), в виде двухбуквенного кода ISO 3166.
getLocalities()	String[]	Возвращает населённый пункт и, опционально, образование внутри населённого пункта, которым принадлежит топоним.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getPremise()	String null	Возвращает наименование здания, если оно есть (например, название терминала аэропорта).

Имя	Возвращает	Описание
<code>getPremiseNumber()</code>	<code>String null</code>	Возвращает номер здания (включая корпус, владение и прочие дополнительные признаки).
<code>getThoroughfare()</code>	<code>String null</code>	Возвращает путь сообщения (улица, шоссе, проезд и т.д.), которому принадлежит топоним (если применимо).
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение <code>null</code> , то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание методов

`getAddressLine`

```
{String} getAddressLine()
```

Возвращает строку с адресом объекта.

`getAdministrativeAreas`

```
{String[]} getAdministrativeAreas()
```

Возвращает административно-территориальные образования, в которые входит объект (федеральный округ, регион, район), не больше двух.

`getCountry`

```
{String|null} getCountry()
```

Возвращает государство, которому принадлежит топоним (если применимо).

`getCountryCode`

```
{String|null} getCountryCode()
```

Возвращает код государства, которому принадлежит топоним (если применимо), в виде двухбуквенного кода ISO 3166.

`getLocalities`

```
{String[]} getLocalities()
```

Возвращает населённый пункт и, опционально, образование внутри населённого пункта, которым принадлежит топоним.

getPremise

```
{String|null} getPremise()
```

Возвращает наименование здания, если оно есть (например, название терминала аэропорта).

getPremiseNumber

```
{String|null} getPremiseNumber()
```

Возвращает номер здания (включая корпус, владение и прочие дополнительные признаки).

getThoroughfare

```
{String|null} getThoroughfare()
```

Возвращает путь сообщения (улица, шоссе, проезд и т.д.), которому принадлежит топоним (если применимо).

geolocation

Статический объект.

Предоставляет информацию о местоположении пользователя.

Методы

Методы

Имя	Возвращает	Описание
get([options])	vow.Promise	Пытается определить местоположение пользователя. Возвращает объект-promise, который будет либо подтверждён объектом с полем geoObjects, либо отклонён с сообщением об ошибке. Поле geoObjects является экземпляром GeoObjectCollection . В коллекцию будет добавлен объект, обозначающий текущее местоположение пользователя.

Описание методов

get

```
{vow.Promise} get([options])
```

Пытается определить местоположение пользователя. Возвращает объект-promise, который будет либо подтверждён объектом с полем geoObjects, либо отклонён с сообщением об ошибке. Поле geoObjects является экземпляром [GeoObjectCollection](#). В коллекцию будет добавлен объект, обозначающий текущее местоположение пользователя.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип:
<code>options.autoReverseGeocode</code>	<code>true</code>	Тип: <code>true</code> - автоматически прогеокодировать положение пользователя, <code>false</code> - вернуть как есть. В случае автоматического геокодирования объект, обозначающий текущее местоположение пользователя, будет иметь структуру, аналогичную результату выполнения geocode .
<code>options.mapStateAutoApply</code>	<code>false</code>	Тип: <code>true</code> - при добавлении на карту автоматически выставить центр и уровень масштабирования карты так, чтобы показать текущее местоположение пользователя, <code>false</code> - ничего не делать.
<code>options.provider</code>	<code>'auto'</code>	Тип: Провайдер геолокации. Допустимые значения: <code>'yandex'</code> - геолокация по данным Яндекса на основе ip пользователя, <code>'browser'</code> - встроенная браузерная геолокация, <code>'auto'</code> - провести геолокацию всеми доступными способами и выбрать лучшее значение.
<code>options.timeout</code>	<code>30000</code>	Тип: Время ожидания ответа в мс.
<code>options.useMapMargin</code>	<code>true</code>	Тип: Boolean Нужно ли учитывать отступы карты map.margin.Manager при автоматическом выставлении центра и уровня масштаба карте.

Примеры:

1.

```
ymaps.geolocation.get({
```

```
// Выставляем опцию для определения положения по ip
provider: 'yandex',
// Карта автоматически отцентрируется по положению пользователя.
mapStateAutoApply: true
}).then(function (result) {
  myMap.geoObjects.add(result.geoObjects);
});
```

2.

```
ymaps.geolocation.get({
  // Зададим способ определения геолокации
  // на основе ip пользователя.
  provider: 'yandex',
  // Включим автоматическое геокодирование результата.
  autoReverseGeocode: true
}).then(function (result) {
  // Выведем результат геокодирования.
  console.log(result.geoObjects.get(0).properties.get('metaDataProperty'));
});
```

geometry

geometry.base

geometry.base.Circle

Расширяет [IBaseCircleGeometry](#).

Базовая геометрия "Круг".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.base.Circle([coordinates[, radius]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates	null	Тип: Number[] null Координаты центра круга.
radius	0	Тип: Number Радиус круга.

Пример:

```
var myCircle = new ymaps.geometry.base.Circle([0, 0], 10);
myCircle.events.add('change', function () {
  alert('Геометрия изменилась');
});
myCircle.freeze();
myCircle.setCoordinates([10, 10]);
myCircle.setRadius(20);
// В этот момент будет сгенерировано единственное событие, и будет выведено сообщение.
myCircle.unfreeze();
```

Поля

Имя	Тип	Описание
events	event.Manager	Менеджер событий геометрии.

События

Имя	Описание
change	<p>Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> oldCoordinates - старые координаты центра; newCoordinates - новые координаты центра; oldRadius - старый радиус; newRadius - новый радиус. <p>Унаследовано от ICircleGeometryAccess.</p>

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	<p>Проверяет, лежит ли переданная точка внутри круга.</p> <p>Унаследован от ICircleGeometryAccess.</p>
freeze()	IFreezable	<p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от IFreezable.</p>
getBounds()	Number[][] null	<p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.</p> <p>Унаследован от IBaseGeometry.</p>
getClosest(anchorPosition)	Object	<p>Ищет на окружности точку, ближайшую к anchorPosition.</p> <p>Унаследован от ICircleGeometryAccess.</p>
getCoordinates()	Number[][] null	<p>Возвращает координаты центра круга.</p> <p>Унаследован от ICircleGeometryAccess.</p>
getRadius()	Number	<p>Возвращает радиус круга.</p> <p>Унаследован от ICircleGeometryAccess.</p>

Имя	Возвращает	Описание
getType()	String	Возвращает строку "Circle". Унаследован от IBaseCircleGeometry .
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
setCoordinates(coordinates)	ICircleGeometryAccess	Задаёт координаты центра круга. Унаследован от ICircleGeometryAccess .
setRadius(radius)	ICircleGeometryAccess	Задаёт радиус круга. Унаследован от ICircleGeometryAccess .
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

Описание полей

events

```
{event.Manager} events
```

Менеджер событий геометрии.

geometry.base.LinearRing

Расширяет [IBaseLinearRingGeometry](#).

Базовая геометрия "Замкнутый контур".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.base.LinearRing([coordinates[, fillRule]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates	[]	Тип: Number[][] Координаты геометрии.

Параметр	Значение по умолчанию	Описание
<code>fillRule</code>	"evenOdd"	<p>Тип: String</p> <p>Строковый идентификатор, определяющий алгоритм заливки многоугольника. Может принимать одно из двух значений:</p> <ul style="list-style-type: none">• <code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.• <code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.

Пример:

```
var linearRing = new ymaps.geometry.base.LinearRing([
  [0, 0], [0, 10], [10, 10], [10, 0], [0, 0]
]);
//...
linearRing.set(1, [5, 10]);
```

Поля

Имя	Тип	Описание
events	event.Manager	Менеджер событий геометрии.

События

Имя	Описание
change	<p>Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>oldCoordinates</code> — старые координаты; <code>newCoordinates</code> - новые координаты; <code>oldFillRule</code> - старый алгоритм заливки; <code>newFillRule</code> - новые алгоритм заливки. <p>Унаследовано от ILinearRingGeometryAccess.</p>

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	<p>Проверяет, лежит ли переданная точка внутри контура.</p> <p>Унаследован от ILinearRingGeometryAccess.</p>
freeze()	IFreezable	<p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от IFreezable.</p>
get(index)	Number[]	<p>Возвращает координаты точки с заданным индексом.</p> <p>Унаследован от ILinearRingGeometryAccess.</p>
getBounds()	Number[][] null	<p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.</p> <p>Унаследован от IBaseGeometry.</p>

Имя	Возвращает	Описание
getChildGeometry(index)	IPointGeometryAccess	Создает и возвращает объект IPointGeometryAccess для заданной вершины ломаной линии. Унаследован от ILinearRingGeometryAccess .
getClosest(anchorPosition)	Object	Ищет на контуре точку, ближайшую к <code>anchorPosition</code> . Унаследован от ILinearRingGeometryAccess .
getCoordinates()	Number[][]	Возвращает массив координат геометрии. Унаследован от ILinearRingGeometryAccess .
getFillRule()	String	Возвращает идентификатор алгоритма заливки. Унаследован от ILinearRingGeometryAccess .
getLength()	Integer	Возвращает количество точек в геометрии. Унаследован от ILinearRingGeometryAccess .
getType()	String	Возвращает строку "LinearRing". Унаследован от IBaseLinearRingGeometry .
insert(index, coordinates)	ILinearRingGeometryAccess	Добавляет новую точку с заданным индексом. Унаследован от ILinearRingGeometryAccess .
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
remove(index)	Number[]	Удаляет точку с заданным индексом. Унаследован от ILinearRingGeometryAccess .
set(index, coordinates)	ILinearRingGeometryAccess	Задаёт координаты точки с заданным индексом. Унаследован от ILinearRingGeometryAccess .
setCoordinates(coordinates)	ILinearRingGeometryAccess	Задаёт массив координат геометрии. Унаследован от ILinearRingGeometryAccess .

Имя	Возвращает	Описание
<code>setFillRule(fillRule)</code>	ILinearRingGeometryAccess	Задаёт алгоритм заливки контура. Унаследован от ILinearRingGeometryAccess .
<code>splice(index, number)</code>	<code>Number[][]</code>	Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра <code>number</code> . Унаследован от ILinearRingGeometryAccess .
<code>unfreeze()</code>	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

Описание полей

events

```
{event.Manager} events
```

Менеджер событий геометрии.

geometry.base.LinearRing.fromEncodedCoordinates

Статическая функция.

Создает геометрию [geometry.base.LinearRing](#) на основе строки закодированных в Base64 координат.

Возвращает геометрию.

```
{ geometry.base.LinearRing } geometry.base.LinearRing.fromEncodedCoordinates(encodedCoordinates)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>encodedCoordinates *</code>	—	Тип: String Закодированные в Base64 координаты вершин контура.

* Обязательный параметр/опция.

geometry.base.LinearRing.toEncodedCoordinates

Статическая функция.

Возвращает строку закодированных в Base64 координат для заданного объекта геометрии.

```
{ String } geometry.base.LinearRing.toEncodedCoordinates(geometry)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geometry *</code>	—	Тип: <code>geometry.base.LinearRing</code> Геометрия.

* Обязательный параметр/опция.

`geometry.base.LineString`

Расширяет `IBaseLineStringGeometry`.

Базовая геометрия "Ломаная линия".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.base.LineString([coordinates])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coordinates</code>	<code>[]</code>	Тип: <code>Number[][]</code> Координаты геометрии.

Пример:

```
var lineString = new ymaps.geometry.base.LineString([
  [30, 50], [31, 51], [32, 52]
]);
//...
lineString.set(1, [20, 40]).remove(2);
```

Поля

Имя	Тип	Описание
<code>events</code>	<code>event.Manager</code>	Менеджер событий геометрии.

События

Имя	Описание
<code>change</code>	Изменение координат. Экземпляр класса <code>Event</code> . Имена полей, доступных через метод <code>Event.get</code> : <ul style="list-style-type: none"><code>oldCoordinates</code> — старые координаты;<code>newCoordinates</code> - новые координаты; Унаследовано от <code>ILineStringGeometryAccess</code> .

Методы

Имя	Возвращает	Описание
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable .
get(index)	Number[]	Возвращает координаты точки с заданным индексом. Унаследован от ILineStringGeometryAccess .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IBaseGeometry .
getChildGeometry(index)	IPointGeometryAccess	Создает и возвращает объект IPointGeometryAccess для заданной вершины ломаной линии. Унаследован от ILineStringGeometryAccess .
getClosest(anchorPosition)	Object	Ищет на ломаной линии точку, ближайшую к anchorPosition. Унаследован от ILineStringGeometryAccess .
getCoordinates()	Number[][]	Возвращает массив координат геометрии. Унаследован от ILineStringGeometryAccess .
getLength()	Integer	Возвращает количество точек в геометрии. Унаследован от ILineStringGeometryAccess .
getType()	String	Возвращает строку "LineString". Унаследован от IBaseLineStringGeometry .
insert(index, coordinates)	ILineStringGeometryAccess	Добавляет новую точку с заданным индексом. Унаследован от ILineStringGeometryAccess .

Имя	Возвращает	Описание
<code>isFrozen()</code>	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
<code>remove(index)</code>	Number[]	Удаляет точку с заданным индексом. Унаследован от ILineStringGeometryAccess .
<code>set(index, coordinates)</code>	ILineStringGeometryAccess	Задаёт координаты точки с заданным индексом. Унаследован от ILineStringGeometryAccess .
<code>setCoordinates(coordinates)</code>	ILineStringGeometryAccess	Задаёт массив координат геометрии. Унаследован от ILineStringGeometryAccess .
<code>splice(index, number)</code>	Number[][]	Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра number. Унаследован от ILineStringGeometryAccess .
<code>unfreeze()</code>	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

Описание полей

events

```
{event.Manager} events
```

Менеджер событий геометрии.

geometry.base.LineString.fromEncodedCoordinates

Статическая функция.

Создает геометрию [geometry.base.LineString](#) на основе строки закодированных в Base64 координат.

Возвращает геометрию.

```
{ geometry.base.LineString } geometry.base.LineString.fromEncodedCoordinates(encodedCoordinates)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>encodedCoordinates</code> *	—	Тип: String Закодированные в Base64 координаты вершин ломаной линии.

* Обязательный параметр/опция.

geometry.base.LineString.toEncodedCoordinates

Статическая функция.

Возвращает строку закодированных в Base64 координат для заданного объекта геометрии.

```
{ String } geometry.base.LineString.toEncodedCoordinates(geometry)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geometry</code> *	—	Тип: geometry.base.LineString Геометрия.

* Обязательный параметр/опция.

geometry.base.Point

Расширяет [IBasePointGeometry](#).

Базовая геометрия "Точка".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.base.Point(\[coordinates\])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coordinates</code>	null	Тип: Number[] null Координаты точки.

Пример:

```
var point = new ymaps.geometry.base.Point([30, 50]);  
  
// Точка всегда будет соответствовать центру карты.  
map.events.add('boundschange', function (e) {  
  if (e.get('newCenter') !== e.get('oldCenter')) {  
    point.setCoordinates(e.get('newCenter'));  
  }  
});
```

Поля

Имя	Тип	Описание
<code>events</code>	event.Manager	Менеджер событий геометрии.

События

Имя	Описание
change	<p>Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>oldCoordinates</code> — старые координаты; <code>newCoordinates</code> - новые координаты; <p>Унаследовано от IPointGeometryAccess.</p>

Методы

Имя	Возвращает	Описание
getBounds()	<code>Number[][] null</code>	<p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.</p> <p>Унаследован от IBaseGeometry.</p>
getCoordinates()	<code>Number[] null</code>	<p>Возвращает координаты точки.</p> <p>Унаследован от IPointGeometryAccess.</p>
getType()	<code>String</code>	<p>Возвращает строку "Point".</p> <p>Унаследован от IBasePointGeometry.</p>
setCoordinates(coordinates)	IPointGeometryAccess	<p>Задаёт координаты точки.</p> <p>Унаследован от IPointGeometryAccess.</p>

Описание полей**events**

```
{event.Manager} events
```

Менеджер событий геометрии.

geometry.base.Polygon

Расширяет [IBasePolygonGeometry](#).

Базовая геометрия "Многоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.base.Polygon([coordinates[, fillRule]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coordinates</code>	<code>[]</code>	Тип: <code>Number[][][]</code> Координаты геометрии.
<code>fillRule</code>	<code>"evenOdd"</code>	Тип: <code>String</code> Строковый идентификатор, определяющий алгоритм заливки многоугольника. Может принимать одно из двух значений: <ul style="list-style-type: none"><code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.<code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.

Пример:

```
var polygon = new ymaps.geometry.base.Polygon([
  // Внешний контур.
  [
    [0, 0], [0, 5], [5, 5], [5, 0], [0, 0]
  ],
  // Внутренний контур.
  [
    [1, 1], [1, 2], [2, 2], [2, 1], [1, 1]
  ]
]);
```

Поля

Имя	Тип	Описание
events	event.Manager	Менеджер событий геометрии.

События

Имя	Описание
change	<p>Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">oldCoordinates — старые координаты;newCoordinates - новые координаты;oldFillRule - старый алгоритм заливки;newFillRule - новые алгоритм заливки. <p>Унаследовано от IPolygonGeometryAccess.</p>

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	<p>Проверяет, лежит ли переданная точка внутри многоугольника.</p> <p>Унаследован от IPolygonGeometryAccess.</p>
freeze()	IFreezable	<p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от IFreezable.</p>
get(index)	Number[][]	<p>Возвращает координаты контура с заданным индексом.</p> <p>Унаследован от IPolygonGeometryAccess.</p>

Имя	Возвращает	Описание
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IBaseGeometry .
getChildGeometry(index)	ILinearRingGeometryAccess	Создает и возвращает объект ILinearRingGeometryAccess для заданного контура. Унаследован от IPolygonGeometryAccess .
getClosest(anchorPosition)	Object	Ищет на контуре многоугольника точку, ближайшую к anchorPosition. Унаследован от IPolygonGeometryAccess .
getCoordinates()	Number[][][]	Возвращает массив координат геометрии. Унаследован от IPolygonGeometryAccess .
getFillRule()	String	Возвращает идентификатор алгоритма заливки. Унаследован от IPolygonGeometryAccess .
getLength()	Integer	Возвращает количество контуров в геометрии. Унаследован от IPolygonGeometryAccess .
getType()	String	Возвращает строку "Polygon". Унаследован от IBasePolygonGeometry .
insert(index, path)	IPolygonGeometryAccess	Добавляет новый контур с заданным индексом. Унаследован от IPolygonGeometryAccess .

Имя	Возвращает	Описание
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
remove(index)	ILinearRingGeometryAccess	Удаляет контур с заданным индексом. Унаследован от IPolygonGeometryAccess .
set(index, path)	IPolygonGeometryAccess	Задаёт координаты контура с заданным индексом. Унаследован от IPolygonGeometryAccess .
setCoordinates(coordinates)	IPolygonGeometryAccess	Задаёт массив координат геометрии. Унаследован от IPolygonGeometryAccess .
setFillRule(fillRule)	IPolygonGeometryAccess	Задаёт алгоритм заливки многоугольника. Унаследован от IPolygonGeometryAccess .
splice(index, number)	ILinearRingGeometryAccess	Удаляет определенное число контуров начиная с заданного индекса. При этом на место удаляемых контуров могут быть добавлены новые. Координаты новых контуров можно передавать дополнительными аргументами после параметра number. Унаследован от IPolygonGeometryAccess .
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

Описание полей

events

```
{event.Manager} events
```

Менеджер событий геометрии.

geometry.base.Polygon.fromEncodedCoordinates

Статическая функция.

Создает геометрию [geometry.base.Polygon](#) на основе строки закодированных в Base64 координат.

Возвращает геометрию.

```
{ geometry.base.Polygon } geometry.base.Polygon.fromEncodedCoordinates(encodedCoordinates)
```

Параметры:

Параметр	Значение по умолчанию	Описание
encodedCoordinates *	—	Тип: String Закодированные в Base64 координаты вершин многоугольника.

* Обязательный параметр/опция.

geometry.base.Polygon.toEncodedCoordinates

Статическая функция.

Возвращает строку закодированных в Base64 координат для заданного объекта геометрии.

```
{ String } geometry.base.Polygon.toEncodedCoordinates(geometry)
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: geometry.base.Polygon Геометрия.

* Обязательный параметр/опция.

geometry.base.Rectangle

Расширяет [IBaseRectangleGeometry](#).

Базовая геометрия "Прямоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.base.Rectangle(\[coordinates\])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates	null	Тип: Number[][] null Массив, содержащий координаты двух противоположных углов прямоугольника.

Пример:

```
var rectangle = new ymaps.geometry.base.Rectangle([  
  [30, 50], [31, 51]  
]);
```

Поля

Имя	Тип	Описание
events	event.Manager	Менеджер событий геометрии.

События

Имя	Описание
change	<p>Изменение координат углов. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> oldCoordinates - старые координаты углов; newCoordinates - новые координаты углов. <p>Унаследовано от IRectangleGeometryAccess.</p>

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	<p>Проверяет, лежит ли переданная точка внутри прямоугольника.</p> <p>Унаследован от IRectangleGeometryAccess.</p>
freeze()	IFreezable	<p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от IFreezable.</p>
getBounds()	Number[][] null	<p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.</p> <p>Унаследован от IBaseGeometry.</p>
getClosest(anchorPosition)	Object	<p>Ищет на контуре прямоугольника точку, ближайшую к anchorPosition.</p> <p>Унаследован от IRectangleGeometryAccess.</p>
getCoordinates()	Number[][]	<p>Возвращает координаты двух противоположных углов прямоугольника.</p> <p>Унаследован от IRectangleGeometryAccess.</p>

Имя	Возвращает	Описание
getType()	String	Возвращает строку "Rectangle". Унаследован от IBaseRectangleGeometry .
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
setCoordinates(coordinates)	IRectangleGeometryAccess	Задаёт координаты двух противоположных углов прямоугольника. Унаследован от IRectangleGeometryAccess .
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

Описание полей

events

```
{event.Manager} events
```

Менеджер событий геометрии.

geometry.Circle

Расширяет [ICircleGeometry](#).

Геометрия "Круг".

См.: [Circle](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.Circle([coordinates[, radius[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates	null	Тип: Number[] null Координаты центра круга.
radius	0	Тип: Number Радиус круга в метрах.
options	—	Тип: Object Опции геометрии.

Параметр	Значение по умолчанию	Описание
<code>options.geodesic</code>	<code>false</code>	Тип: Boolean Включает отображение с использованием геодезических линий.
<code>options.pixelRendering</code>	<code>"jumpy"</code>	Тип: String Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений: <ul style="list-style-type: none"> <code>jumpy</code> - отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещаться во время движения карты; <code>static</code> - отображение всегда располагается в начальном мире и не перемещается при движении карты.
<code>options.projection</code>	—	Тип: <code>IProjection</code> Проекция.

Пример:

```
// Создаем экземпляр класса геометрии круга (указываем координаты и радиус в метрах).
var circleGeometry = new ymaps.geometry.Circle([30, 50], 10),
// Создаем экземпляр класса геообъекта и передаем нашу геометрию в конструктор.
circleGeoObject = new ymaps.GeoObject({ geometry: circleGeometry });

// Изменяем радиус геометрии через свойство geometry геообъекта.
circleGeoObject.geometry.setRadius(5)
// Или напрямую.
circleGeometry.setRadius(5);
// Также доступ к circleGeometry можно получить через circleGeoObject.geometry.
```

Поля

Имя	Тип	Описание
<code>events</code>	<code>IEventManager</code>	Менеджер событий. Унаследовано от <code>IEventEmitter</code> .
<code>options</code>	<code>IOptionManager</code>	Менеджер опций. Унаследовано от <code>ICustomizable</code> .

События

Имя	Описание
change	<p>Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> oldCoordinates - старые координаты центра; newCoordinates - новые координаты центра; oldRadius - старый радиус; newRadius - новый радиус. <p>Унаследовано от ICircleGeometryAccess.</p>
mapchange	<p>Сменилась карта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IGeometry.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
pixelgeometrychange	<p>Изменилась пиксельная геометрия. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> pixelGeometry - новая пиксельная геометрия IPixelGeometry. <p>Унаследовано от IGeometry.</p>

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	<p>Проверяет, лежит ли переданная точка внутри круга.</p> <p>Унаследован от ICircleGeometryAccess.</p>
freeze()	IFreezable	<p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от IFreezable.</p>

Имя	Возвращает	Описание
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно. Унаследован от IGeometry .
getClosest(anchorPosition)	Object	Ищет на окружности точку, ближайшую к anchorPosition. Унаследован от ICircleGeometryAccess .
getCoordinates()	Number[][] null	Возвращает координаты центра круга. Унаследован от ICircleGeometryAccess .
getMap()	Map null	Возвращает текущую карту. Унаследован от IGeometry .
getPixelGeometry([options])	IPixelGeometry	Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты. Унаследован от IGeometry .
getRadius()	Number	Возвращает радиус круга. Унаследован от ICircleGeometryAccess .
getType()	String	Возвращает строку "Circle". Унаследован от ICircleGeometry .
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
setCoordinates(coordinates)	ICircleGeometryAccess	Задает координаты центра круга. Унаследован от ICircleGeometryAccess .
setMap(map)		Задает карту. Унаследован от IGeometry .
setRadius(radius)	ICircleGeometryAccess	Задает радиус круга. Унаследован от ICircleGeometryAccess .

Имя	Возвращает	Описание
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

geometry.json

geometry.json.circle

Расширяет [IGeometryJson](#).

Объект, описывающий JSON-представление геометрии "Круг".

[Конструктор](#) | [Поля](#)

Конструктор

```
geometry.json.circle()
```

Пример:

```
var geometryJson = {  
  type: "Circle",  
  coordinates: [1, 2],  
  radius: 100  
};
```

Поля

Имя	Тип	Описание
coordinates	Number[] null	Координаты центра круга.
radius	Number	Радиус круга.
type	String	Идентификатор типа геометрии "Круг". Всегда должен принимать значение "Circle".

Описание полей

coordinates

```
{Number[]|null} coordinates
```

Координаты центра круга.

radius

```
{Number} radius
```

Радиус круга.

type

```
{String} type
```

Идентификатор типа геометрии "Круг". Всегда должен принимать значение "Circle".

geometry.json.lineString

Расширяет [IGeometryJson](#).

Объект, описывающий JSON-представление геометрии "Ломаная линия".

[Конструктор](#) | [Поля](#)

Конструктор

```
geometry.json.lineString()
```

Пример:

```
var geometryJson = {  
  type: "LineString",  
  coordinates: [[1, 2], [3, 5], [7, 11]]  
};
```

Поля

Имя	Тип	Описание
coordinates	Number[][]	Координаты ломаной линии.
type	String	Идентификатор типа геометрии "Ломаная линия". Всегда должен принимать значение "LineString".

Описание полей

coordinates

```
{Number[][]} coordinates
```

Координаты ломаной линии.

type

```
{String} type
```

Идентификатор типа геометрии "Ломаная линия". Всегда должен принимать значение "LineString".

geometry.json.Point

Расширяет [IGeometryJson](#).

Объект, описывающий JSON-представление геометрии "Точка".

[Конструктор](#) | [Поля](#)

Конструктор

```
geometry.json.Point()
```

Пример:

```
var geometryJson = {  
  type: "Point",  
  coordinates: [1, 2]  
};
```

Поля

Имя	Тип	Описание
type	String	Идентификатор типа геометрии. Унаследовано от IGeometryJson .

geometry.json.polygon

Расширяет [IGeometryJson](#).

Объект, описывающий JSON-представление геометрии "Многоугольник".

[Конструктор](#) | [Поля](#)

Конструктор

```
geometry.json.polygon()
```

Пример:

```
var geometryJson = {  
  type: "Polygon",  
  coordinates: [  
    [[0, 0], [7, 11]],  
    [[1, 2], [3, 5]]  
  ]  
};
```

Поля

Имя	Тип	Описание
coordinates	Number[][][]	Координаты многоугольника.
fillRule	String	Идентификатор алгоритма заливки многоугольника. Может принимать одно из двух значений: <ul style="list-style-type: none">• <code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.• <code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.
type	String	Идентификатор типа геометрии. Унаследовано от IGeometryJson .

Описание полей

coordinates

```
{Number[][][]} coordinates
```

Координаты многоугольника.

fillRule

```
{String} fillRule
```

Идентификатор алгоритма заливки многоугольника. Может принимать одно из двух значений:

- `evenOdd` - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.
- `nonZero` - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.

geometry.json.rectangle

Расширяет [IGeometryJson](#).

Объект, описывающий JSON-представление геометрии "Прямоугольник".

[Конструктор](#) | [Поля](#)

Конструктор

```
geometry.json.rectangle()
```

Пример:

```
var geometryJson = {  
  type: "Rectangle",  
  coordinates: [[1, 2], [3, 5]]  
};
```

Поля

Имя	Тип	Описание
coordinates	<code>Number[][] null</code>	Координаты двух противоположных углов прямоугольника.
type	<code>String</code>	Идентификатор типа геометрии "Прямоугольник". Всегда должен принимать значение "Rectangle".

Описание полей

coordinates

```
{Number[][]|null} coordinates
```

Координаты двух противоположных углов прямоугольника.

type

```
{String} type
```

Идентификатор типа геометрии "Прямоугольник". Всегда должен принимать значение "Rectangle".

geometry.LineString

Расширяет [ILineStringGeometry](#).

Геометрия "Ломаная линия".

См.: [Polyline](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.LineString([coordinates[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates	<code>[]</code>	Тип: <code>Number[][]</code> Координаты геометрии.
options	<code>—</code>	Тип: <code>Object</code> Опции геометрии.
options.coordRendering	<code>—</code>	Тип: <code>String</code> Строковый идентификатор, определяющий алгоритм пересчета координат геометрии в пиксельные координаты. Для геометрии "Ломаная линия" может принимать одно из двух значений: <ul style="list-style-type: none"><code>shortestPath</code> - алгоритм, учитывающий зацикленность проекции по осям, и формирующий пиксельные координаты так, чтобы расстояние между двумя соседними точками было минимальным;<code>straightPath</code> - алгоритм, не учитывающий зацикленность проекции;
options.geodesic	<code>false</code>	Тип: <code>Boolean</code> Включает отображение с использованием геодезических линий.

Параметр	Значение по умолчанию	Описание
<code>options.pixelRendering</code>	"jumpy"	<p>Тип: String</p> <p>Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений:</p> <ul style="list-style-type: none"> <code>jumpy</code> - отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещается во время движения карты; <code>static</code> - отображение всегда располагается в начальном мире и не перемещается при движении карты.
<code>options.projection</code>	—	<p>Тип: IProjection</p> <p>Проекция.</p>
<code>options.simplification</code>	true	<p>Тип: Boolean</p> <p>Включает симплификацию при рендеринге пиксельной геометрии.</p>

Пример:

```
// Создает экземпляр геометрии точки (задает координаты).
var lineStringGeometry = new ymaps.geometry.LineString([
  [30, 50], [31, 51], [32, 52]
]);
// Создание экземпляра объекта geo и передача нашей геометрии конструктору.
var lineStringGeoObject = new ymaps.GeoObject({ geometry: lineStringGeometry });

lineStringGeometry.events.add('change', function (e) {
  alert([e.get('newCoordinates'), e.get('oldCoordinates')]);
});

// Изменение вершин с помощью свойства "геометрия" геообъекта (установка новых координат для второй точки на линии).
lineStringGeoObject.geometry
  .set(1, [20, 40])
  .remove(2);
// Или напрямую.
lineStringGeometry
  .set(1, [20, 40])
  .remove(2);
// Вы также можете получить доступ к lineStringGeometry через lineStringGeoObject.геометрия.
```

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
<code>options</code>	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>

События

Имя	Описание
change	Изменение координат. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>oldCoordinates</code> — старые координаты; <code>newCoordinates</code> - новые координаты; Унаследовано от ILineStringGeometryAccess .
mapchange	Сменилась карта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IGeometry .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
pixelgeometrychange	Изменилась пиксельная геометрия. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>pixelGeometry</code> - новая пиксельная геометрия IPixelGeometry. Унаследовано от IGeometry .

Методы

Имя	Возвращает	Описание
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable .
get(index)	<code>Number[]</code>	Возвращает координаты точки с заданным индексом. Унаследован от ILineStringGeometryAccess .
getBounds()	<code>Number[][]</code> или <code>null</code>	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно. Унаследован от IGeometry .

Имя	Возвращает	Описание
getChildGeometry(index)	IPointGeometryAccess	Создает и возвращает объект IPointGeometryAccess для заданной вершины ломаной линии. Унаследован от ILineStringGeometryAccess .
getClosest(anchorPosition)	Object	Ищет на ломаной линии точку, ближайшую к <code>anchorPosition</code> . Унаследован от ILineStringGeometryAccess .
getCoordinates()	Number[][]	Возвращает массив координат геометрии. Унаследован от ILineStringGeometryAccess .
getDistance([from, to])	Number	Возвращает длину указанного отрезка, либо всей линии, если ограничители не заданы.
getLength()	Integer	Возвращает количество точек в геометрии. Унаследован от ILineStringGeometryAccess .
getMap()	Map null	Возвращает текущую карту. Унаследован от IGeometry .
getPixelGeometry([options])	IPixelGeometry	Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты. Унаследован от IGeometry .
getType()	String	Возвращает строку "LineString". Унаследован от ILineStringGeometry .
insert(index, coordinates)	ILineStringGeometryAccess	Добавляет новую точку с заданным индексом. Унаследован от ILineStringGeometryAccess .
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
remove(index)	Number[]	Удаляет точку с заданным индексом. Унаследован от ILineStringGeometryAccess .

Имя	Возвращает	Описание
<code>set(index, coordinates)</code>	ILineStringGeometryAccess	Задаёт координаты точки с заданным индексом. Унаследован от ILineStringGeometryAccess .
<code>setCoordinates(coordinates)</code>	ILineStringGeometryAccess	Задаёт массив координат геометрии. Унаследован от ILineStringGeometryAccess .
<code>setMap(map)</code>		Задаёт карту. Унаследован от IGeometry .
<code>splice(index, number)</code>	<code>Number[][]</code>	Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра <code>number</code> . Унаследован от ILineStringGeometryAccess .
<code>unfreeze()</code>	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

Описание методов

`getDistance`

```
{Number} getDistance([from[, to]])
```

Возвращает длину указанного отрезка, либо всей линии, если ограничители не заданы.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>from</code>	0	Тип: <code>Number</code> Начиная с какой точки считать длину.
<code>to</code>	—	Тип: <code>Number</code> До какой точки считать длину. Если не указано, берётся последняя точка.

Пример:

```
var lineStringGeometry = new ymaps.geometry.LineString([[30, 50], [31, 51], [32, 52]]),
var geoObject = new ymaps.GeoObject({ geometry: lineStringGeometry });
```

```
myMap.geoObjects.add(geoObject);  
// Длина всей линии.  
console.log(geoObject.geometry.getDistance());  
// Длина отрезка от первой до второй точки.  
console.log(geoObject.geometry.getDistance(0, 1));
```

geometry.LineString.fromEncodedCoordinates

Статическая функция.

Создает геометрию [geometry.LineString](#) на основе строки закодированных в Base64 координат.

Возвращает геометрию.

```
{ geometry.LineString } geometry.LineString.fromEncodedCoordinates(encodedCoordinates)
```

Параметры:

Параметр	Значение по умолчанию	Описание
encodedCoordinates *	—	Тип: String Закодированные в Base64 координаты вершин ломаной линии.

* Обязательный параметр/опция.

Пример:

```
var base64Coords = "gMPJAYDw-gJAQg8AQEIPAEBCDwBAQg8A";  
var geometry = ymaps.geometry.LineString.fromEncodedCoordinates(base64Coords);  
var polyline = new ymaps.Polyline(geometry);
```

geometry.LineString.toEncodedCoordinates

Статическая функция.

Возвращает строку закодированных в Base64 координат для заданного объекта геометрии.

```
{ String } geometry.LineString.toEncodedCoordinates(geometry)
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: geometry.LineString Геометрия.

* Обязательный параметр/опция.

geometry.pixel

geometry.pixel.Circle

Расширяет [IPixelCircleGeometry](#).

Пиксельная геометрия "Круг".

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
geometry.pixel.Circle(coordinates, radius[, metaData])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coordinates</code> *	—	Тип: <code>Number[] null</code> Координаты центра круга.
<code>radius</code> *	—	Тип: <code>Number null</code> Радиус круга.
<code>metaData</code>	—	Тип: <code>Object</code> Метаданные.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	<code>IEventManager</code>	Менеджер событий. Унаследовано от <code>IEventEmitter</code> .

Методы

Имя	Возвращает	Описание
<code>equals(geometry)</code>	<code>Boolean</code>	Возвращает <code>true</code> , если переданная геометрия эквивалентна данной. Унаследован от <code>IPixelGeometry</code> .
<code>getBounds()</code>	<code>Number[] null</code>	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от <code>IBaseGeometry</code> .
<code>getCoordinates()</code>	<code>Number[]</code>	Возвращает координаты центра круга. Унаследован от <code>IPixelCircleGeometry</code> .
<code>getMetaData()</code>	<code>Object</code>	Возвращает метаданные пиксельной геометрии. Унаследован от <code>IPixelGeometry</code> .

Имя	Возвращает	Описание
getRadius()	Number	Возвращает радиус круга. Унаследован от IPixelCircleGeometry .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry .
scale(factor)	IPixelGeometry	Создает отмасштабированную копию геометрии. Унаследован от IPixelGeometry .
shift(offset)	IPixelGeometry	Создает копию геометрии сдвинутую на заданную величину. Унаследован от IPixelGeometry .

geometry.pixel.LineString

Расширяет [IPixelLineStringGeometry](#).

Пиксельная геометрия "Ломаная линия".

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
geometry.pixel.LineString(coordinates[, metaData])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates *	—	Тип: Number[][] Координаты линии.
metaData	—	Тип: Object Метаданные.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
equals(geometry)	Boolean	Возвращает true, если переданная геометрия эквивалентна данной. Унаследован от IPixelGeometry .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IBaseGeometry .
getClosest(anchorPosition)	Object	Ищет на ломаной линии точку, ближайшую к anchorPosition. Унаследован от IPixelLineStringGeometry .
getCoordinates()	Number[][]	Возвращает координаты линии. Унаследован от IPixelLineStringGeometry .
getLength()	Integer	Возвращает количество точек в геометрии. Унаследован от IPixelLineStringGeometry .
getMetaData()	Object	Возвращает метаданные пиксельной геометрии. Унаследован от IPixelGeometry .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry .
scale(factor)	IPixelGeometry	Создает отмасштабированную копию геометрии. Унаследован от IPixelGeometry .
shift(offset)	IPixelGeometry	Создает копию геометрии сдвинутую на заданную величину. Унаследован от IPixelGeometry .

geometry.pixel.MultiLineString

Расширяет [IPixelMultiLineGeometry](#).

Пиксельная геометрия "Множество линий".

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
geometry.pixel.MultiLineString(coordinates[, metaData])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates *	—	Тип: Number[][] Координаты линий.
metaData	—	Тип: Object Метаданные.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
equals(geometry)	Boolean	Возвращает true, если переданная геометрия эквивалентна данной. Унаследован от IPixelGeometry .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IBaseGeometry .

Имя	Возвращает	Описание
getClosest(anchorPosition)	Object	Ищет на контуре точку, ближайшую к anchorPosition. Унаследован от IPixelMultiLineGeometry .
getCoordinates()	Number[][][]	Возвращает координаты мультилиний. Унаследован от IPixelMultiLineGeometry .
getLength()	Integer	Возвращает количество линий в мультилинии. Унаследован от IPixelMultiLineGeometry .
getMetaData()	Object	Возвращает метаданные пиксельной геометрии. Унаследован от IPixelGeometry .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry .
scale(factor)	IPixelGeometry	Создает отмасштабированную копию геометрии. Унаследован от IPixelGeometry .
shift(offset)	IPixelGeometry	Создает копию геометрии сдвинутую на заданную величину. Унаследован от IPixelGeometry .

geometry.pixel.MultiPolygon

Расширяет [IPixelMultiPolygonGeometry](#).

Пиксельная геометрия "Многоугольник из нескольких фигур".

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
geometry.pixel.MultiPolygon(coordinates, fillRule[, metaData])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates *	—	Тип: Number[][][] Координаты многоугольников.

Параметр	Значение по умолчанию	Описание
fillRule *	—	<p>Тип: String</p> <p>Строковый идентификатор, определяющий алгоритм заливки многоугольников. Может принимать одно из двух значений:</p> <ul style="list-style-type: none">• <code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.• <code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.
metaData	—	<p>Тип: Object</p> <p>Метаданные.</p>

Параметр	Значение по умолчанию	Описание
metaData.convex	false	Тип: Boolean Признак выпуклости многоугольника. true — выпуклый, false — невыпуклый. Для выпуклых многоугольников быстрее рассчитывается попадание точки в многоугольник.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет лежит ли переданная точка внутри мультимногоугольника. Унаследован от IPixelMultiPolygonGeometry .
equals(geometry)	Boolean	Возвращает true, если переданная геометрия эквивалентна данной. Унаследован от IPixelGeometry .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IBaseGeometry .
getClosest(anchorPosition)	Object	Ищет на контуре мультимногоугольника точку, ближайшую к anchorPosition. Унаследован от IPixelMultiPolygonGeometry .

Имя	Возвращает	Описание
getCoordinates()	Number[][][]	Возвращает координаты мультимногоугольника. Унаследован от IPixelMultiPolygonGeometry .

Имя	Возвращает	Описание
getFillRule()	String	<p>Возвращает строковый идентификатор, определяющий алгоритм заливки мультимногоугольника. Идентификатор может иметь одно из двух значений:</p> <ul style="list-style-type: none">• <code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.• <code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри. <p>Унаследован от IPixelMultiPolygonGeometry.</p>

Имя	Возвращает	Описание
getLength()	Integer	Возвращает количество многоугольников в мультимногоугольнике. Унаследован от IPixelMultiPolygonGeometry .
getMetaData()	Object	Возвращает метаданные пиксельной геометрии. Унаследован от IPixelGeometry .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry .
scale(factor)	IPixelGeometry	Создает отмасштабированную копию геометрии. Унаследован от IPixelGeometry .
shift(offset)	IPixelGeometry	Создает копию геометрии сдвинутую на заданную величину. Унаследован от IPixelGeometry .

geometry.pixel.Point

Расширяет [IPixelPointGeometry](#).

Пиксельная геометрия "Точка".

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
geometry.pixel.Point(position[, metaData])
```

Параметры:

Параметр	Значение по умолчанию	Описание
position *	—	Тип: Number[] null Координаты точки.
metaData	—	Тип: Object Метаданные.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
equals(geometry)	Boolean	Возвращает true, если переданная геометрия эквивалентна данной. Унаследован от IPixelGeometry .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IBaseGeometry .
getCoordinates()	Number[]	Возвращает координаты точки. Унаследован от IPixelPointGeometry .
getMetaData()	Object	Возвращает метаданные пиксельной геометрии. Унаследован от IPixelGeometry .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry .
scale(factor)	IPixelGeometry	Создает отмасштабированную копию геометрии. Унаследован от IPixelGeometry .
shift(offset)	IPixelGeometry	Создает копию геометрии сдвинутую на заданную величину. Унаследован от IPixelGeometry .

geometry.pixel.Polygon

Расширяет [IPixelPolygonGeometry](#).

Пиксельная геометрия "Многоугольник".

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
geometry.pixel.Polygon(coordinates, fillRule[, metaData])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates *	—	<p>Тип: Number[]</p> <p>Координаты многоугольника.</p>
fillRule *	—	<p>Тип: String</p> <p>Строковый идентификатор, определяющий алгоритм заливки многоугольника. Может принимать одно из двух значений:</p> <ul style="list-style-type: none"> • <code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи. • <code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.
metaData	—	<p>Тип: Object</p> <p>Метаданные.</p>

Параметр	Значение по умолчанию	Описание
metaData.convex	false	Тип: Boolean Признак выпуклости многоугольника. true — выпуклый, false — невыпуклый. Для выпуклых многоугольников быстрее рассчитывается попадание точки в многоугольник.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри многоугольника. Унаследован от IPixelPolygonGeometry .
equals(geometry)	Boolean	Возвращает true, если переданная геометрия эквивалентна данной. Унаследован от IPixelGeometry .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IBaseGeometry .
getClosest(anchorPosition)	Object	Ищет на контуре многоугольника точку, ближайшую к anchorPosition. Унаследован от IPixelPolygonGeometry .

Имя	Возвращает	Описание
getCoordinates()	Number[][]	<p>Возвращает координаты многоугольника.</p> <p>Унаследован от IPixelPolygonGeometry.</p>

Имя	Возвращает	Описание
getFillRule()	String	<p>Возвращает строковый идентификатор, определяющий алгоритм заливки многоугольника. Идентификатор может принимать одно из двух значений:</p> <ul style="list-style-type: none">• <code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.• <code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри. <p>Унаследован от IPixelPolygonGeometry.</p>

Имя	Возвращает	Описание
getLength()	Integer	Возвращает количество контуров в многоугольнике. Унаследован от IPixelPolygonGeometry .
getMetaData()	Object	Возвращает метаданные пиксельной геометрии. Унаследован от IPixelGeometry .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry .
scale(factor)	IPixelGeometry	Создает отмасштабированную копию геометрии. Унаследован от IPixelGeometry .
shift(offset)	IPixelGeometry	Создает копию геометрии сдвинутую на заданную величину. Унаследован от IPixelGeometry .

geometry.pixel.Rectangle

Расширяет [IPixelRectangleGeometry](#).

Пиксельная геометрия "Прямоугольник".

[Конструктор](#) | [Поля](#) | [Методы](#)

Конструктор

```
geometry.pixel.Rectangle([coordinates[, metaData]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates	null	Тип: Number[][] null Координаты двух противоположных углов прямоугольника.
metaData	—	Тип: Object Метаданные.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
equals(geometry)	Boolean	Возвращает true, если переданная геометрия эквивалентна данной. Унаследован от IPixelGeometry .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IBaseGeometry .
getClosest(anchorPosition)	Object	Ищет на прямоугольнике точку, ближайшую к anchorPosition. Унаследован от IPixelRectangleGeometry .
getCoordinates()	Number[][]	Возвращает координаты двух противоположных углов прямоугольника. Унаследован от IPixelRectangleGeometry .
getMetaData()	Object	Возвращает метаданные пиксельной геометрии. Унаследован от IPixelGeometry .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry .
scale(factor)	IPixelGeometry	Создает отмасштабированную копию геометрии. Унаследован от IPixelGeometry .
shift(offset)	IPixelGeometry	Создает копию геометрии сдвинутую на заданную величину. Унаследован от IPixelGeometry .

geometry.Point

Расширяет [IPointGeometry](#).

Геометрия "Точка".

См.: [Placemark](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.Point([position[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
position	null	Тип: Number[] Координаты точки.
options	—	Тип: Object Опции геометрии.
options.pixelRendering	"jumpy"	Тип: String Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений: <ul style="list-style-type: none">• <code>jumpy</code> - отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещается во время движения карты;• <code>static</code> - отображение всегда располагается в начальном мире и не перемещается при движении карты.
options.projection	—	Тип: IProjection Проекция.

Пример:

```
// Создаем инстанцию геометрии точки (указываем координаты).
var pointGeometry = new ymaps.geometry.Point([30, 50]),
// Создаем инстанцию геообъекта и передаем нашу геометрию в конструктор.
var placemark = new ymaps.GeoObject({ geometry: pointGeometry });

// Изменяем вершины через свойство geometry геообъекта.
placemark.geometry.setCoordinates([20, 40]);
// Или напрямую.
pointGeometry.setCoordinates([20, 40]);
// Также доступ к pointGeometry можно получить через placemark.geometry.
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Имя	Тип	Описание
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
change	Изменение координат. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>oldCoordinates</code> — старые координаты; <code>newCoordinates</code> - новые координаты; Унаследовано от IPointGeometryAccess .
mapchange	Сменилась карта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IGeometry .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
pixelgeometrychange	Изменилась пиксельная геометрия. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>pixelGeometry</code> - новая пиксельная геометрия IPixelGeometry. Унаследовано от IGeometry .

Методы

Имя	Возвращает	Описание
getBounds()	<code>Number[][] null</code>	Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно. Унаследован от IGeometry .
getCoordinates()	<code>Number[][] null</code>	Возвращает координаты точки. Унаследован от IPointGeometryAccess .

Имя	Возвращает	Описание
getMap()	Map null	Возвращает текущую карту. Унаследован от IGeometry .
getPixelGeometry([options])	IPixelGeometry	Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты. Унаследован от IGeometry .
getType()	String	Возвращает строку "Point". Унаследован от IPointGeometry .
setCoordinates(coordinates)	IPointGeometryAccess	Задает координаты точки. Унаследован от IPointGeometryAccess .
setMap(map)		Задает карту. Унаследован от IGeometry .

geometry.Polygon

Расширяет [IPolygonGeometry](#).

Геометрия "Многоугольник".

См.: [Polygon](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.Polygon([coordinates[, fillRule[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates	[]	Тип: Number[][][] Координаты геометрии. Трехмерный массив, элементами которого являются двумерные массивы координат контуров многоугольника. Первый элемент описывает внешний контур, остальные - внутренние.

Параметр	Значение по умолчанию	Описание
fillRule	"evenOdd"	<p>Тип: String</p> <p>Строковый идентификатор, определяющий алгоритм заливки многоугольника. Может принимать одно из двух значений:</p> <ul style="list-style-type: none">• evenOdd - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.• nonZero - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.
options	—	<p>Тип: Object</p> <p>Опции геометрии.</p>

Параметр	Значение по умолчанию	Описание
<code>options.coordRendering</code>	"shortestPath"	<p>Тип: String</p> <p>Строковый идентификатор, определяющий алгоритм пересчета координат геометрии в пиксельные координаты. Может принимать одно из двух значений:</p> <ul style="list-style-type: none"> <code>shortestPath</code> - алгоритм, учитывающий зацикленность проекции по осям, и формирующий пиксельные координаты так, чтобы расстояние между двумя соседними точками было минимальным; <code>straightPath</code> - алгоритм, не учитывающий зацикленность проекции;
<code>options.geodesic</code>	false	<p>Тип: Boolean</p> <p>Включает отображение с использованием геодезических линий.</p>
<code>options.pixelRendering</code>	"jumpy"	<p>Тип: String</p> <p>Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений:</p> <ul style="list-style-type: none"> <code>jumpy</code> - отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещается во время движения карты; <code>static</code> - отображение всегда располагается в начальном мире и не перемещается при движении карты.
<code>options.projection</code>	—	<p>Тип: IProjection</p> <p>Проекция.</p>
<code>options.simplification</code>	true	<p>Тип: Boolean</p> <p>Включает симплификацию при рендеринге пиксельной геометрии.</p>

Пример:

```
// Создаем инстанцию геометрии многоугольника (указываем координаты вершин контуров).
var polygonGeometry = new ymaps.geometry.Polygon([
  // Внешний контур.
  [
    [0, 0], [0, 5], [5, 5], [5, 0], [0, 0]
```



```

    ],
    // Внутренний контур.
    [
        [1, 1], [1, 2], [2, 2], [2, 1], [1, 1]
    ]
  ]),
  // Создаем инстанцию геообъекта и передаем нашу геометрию в конструктор.
  polygonGeoObject = new ymaps.GeoObject({ geometry: polygonGeometry });

  // Также доступ к polygonGeometry можно получить через polygonGeoObject.geometry.

```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
change	Изменение координат. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>oldCoordinates</code> — старые координаты; <code>newCoordinates</code> - новые координаты; <code>oldFillRule</code> - старый алгоритм заливки; <code>newFillRule</code> - новые алгоритм заливки. Унаследовано от IPolygonGeometryAccess .
mapchange	Сменилась карта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IGeometry .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
pixelgeometrychange	Изменилась пиксельная геометрия. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>pixelGeometry</code> - новая пиксельная геометрия IPixelGeometry. Унаследовано от IGeometry .

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	<p>Проверяет, лежит ли переданная точка внутри многоугольника.</p> <p>Унаследован от IPolygonGeometryAccess.</p>
freeze()	IFreezable	<p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от IFreezable.</p>
get(index)	Number[][]	<p>Возвращает координаты контура с заданным индексом.</p> <p>Унаследован от IPolygonGeometryAccess.</p>
getBounds()	Number[][] null	<p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.</p> <p>Унаследован от IGeometry.</p>
getChildGeometry(index)	ILinearRingGeometryAccess	<p>Создает и возвращает объект ILinearRingGeometryAccess для заданного контура.</p> <p>Унаследован от IPolygonGeometryAccess.</p>
getClosest(anchorPosition)	Object	<p>Ищет на контуре многоугольника точку, ближайшую к anchorPosition.</p> <p>Унаследован от IPolygonGeometryAccess.</p>
getCoordinates()	Number[][][]	<p>Возвращает массив координат геометрии.</p> <p>Унаследован от IPolygonGeometryAccess.</p>
getFillRule()	String	<p>Возвращает идентификатор алгоритма заливки.</p> <p>Унаследован от IPolygonGeometryAccess.</p>
getLength()	Integer	<p>Возвращает количество контуров в геометрии.</p> <p>Унаследован от IPolygonGeometryAccess.</p>

Имя	Возвращает	Описание
<code>getMap()</code>	<code>Map</code> null	Возвращает текущую карту. Унаследован от IGeometry .
<code>getPixelGeometry([options])</code>	IPixelGeometry	Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты. Унаследован от IGeometry .
<code>getType()</code>	String	Возвращает строку "Polygon". Унаследован от IPolygonGeometry .
<code>insert(index, path)</code>	IPolygonGeometryAccess	Добавляет новый контур с заданным индексом. Унаследован от IPolygonGeometryAccess .
<code>isFrozen()</code>	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
<code>remove(index)</code>	ILinearRingGeometryAccess	Удаляет контур с заданным индексом. Унаследован от IPolygonGeometryAccess .
<code>set(index, path)</code>	IPolygonGeometryAccess	Задаёт координаты контура с заданным индексом. Унаследован от IPolygonGeometryAccess .
<code>setCoordinates(coordinates)</code>	IPolygonGeometryAccess	Задаёт массив координат геометрии. Унаследован от IPolygonGeometryAccess .
<code>setFillRule(fillRule)</code>	IPolygonGeometryAccess	Задаёт алгоритм заливки многоугольника. Унаследован от IPolygonGeometryAccess .
<code>setMap(map)</code>		Задаёт карту. Унаследован от IGeometry .

Имя	Возвращает	Описание
<code>splice(index, number)</code>	ILinearRingGeometryAccess []	Удаляет определенное число контуров начиная с заданного индекса. При этом на место удаляемых контуров могут быть добавлены новые. Координаты новых контуров можно передавать дополнительными аргументами после параметра <code>number</code> . Унаследован от IPolygonGeometryAccess .
<code>unfreeze()</code>	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

geometry.Polygon.fromEncodedCoordinates

Статическая функция.

Создает геометрию [geometry.Polygon](#) на основе строки закодированных в Base64 координат.

Возвращает геометрию.

```
{ geometry.Polygon } geometry.Polygon.fromEncodedCoordinates(encodedCoordinates)
```

Параметры:

Параметр	Значение по умолчанию	Описание
encodedCoordinates *	—	Тип: String Закодированные в Base64 координаты вершин ломаной линии.

* Обязательный параметр/опция.

Пример:

```
var base64Coords =
  "AAAAAAAAAAAAAAAAAQEtMAEBLTAAAAAAAAAAAAAMC0s__AtLP_AAAAAA==;QEIPAEBCDwAAAAAAAAAQEIPAEBCDwAAAAAAAAAMC98P_AvFD_AAAAAA==",
  geometry = ymaps.geometry.Polygon.fromEncodedCoordinates(base64Coords);
  polygon = new ymaps.Polygon(geometry);
```

geometry.Polygon.toEncodedCoordinates

Статическая функция.

Возвращает строку закодированных в Base64 координат для заданного объекта геометрии.

```
{ String } geometry.Polygon.toEncodedCoordinates(geometry)
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: geometry.Polygon Геометрия.

* Обязательный параметр/опция.

geometry.Rectangle

Расширяет [IRectangleGeometry](#).

Геометрия "Прямоугольник".

См.: [Rectangle](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometry.Rectangle([coordinates[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
coordinates	null	Тип: Number[][] null Массив, содержащий координаты двух противоположных углов прямоугольника.
options	—	Тип: Object Опции геометрии.

Параметр	Значение по умолчанию	Описание
<code>options.coordRendering</code>	—	<p>Тип: String</p> <p>Строковый идентификатор, определяющий алгоритм пересчета координат геометрии в пиксельные координаты. Для геометрии "Прямоугольник" может принимать одно из трех значений:</p> <ul style="list-style-type: none">• <code>shortestPath</code> - алгоритм, учитывающий зацикленность проекции по осям, и формирующий пиксельные координаты так, чтобы расстояние между противоположными углами было минимальным;• <code>straightPath</code> - алгоритм, не учитывающий зацикленность проекции;• <code>boundsPath</code> - алгоритм, трактующий координаты углов прямоугольника, как координаты соответственно нижнего и верхнего углов ограничивающей области. При расчете диагонали по зацикленным осям проекции всегда выбирается направление обхода против часовой стрелки.
<code>options.geodesic</code>	false	<p>Тип: Boolean</p> <p>Включает отображение с использованием геодезических линий.</p>

Параметр	Значение по умолчанию	Описание
options.pixelRendering	"jumpy"	<p>Тип: String</p> <p>Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений:</p> <ul style="list-style-type: none"> <code>jumpy</code> - отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещаться во время движения карты; <code>static</code> - отображение всегда располагается в начальном мире и не перемещается при движении карты.
options.projection	—	<p>Тип: IProjection</p> <p>Проекция.</p>

Пример:

```
// Создаем инстанцию геометрии прямоугольника (указываем координаты противоположных углов).
var rectangleGeometry = new ymaps.geometry.Rectangle([[30, 50], [31, 51]]),
// Создаем инстанцию геообъекта и передаем нашу геометрию в конструктор.
rectangleGeoObject = new ymaps.GeoObject({ geometry: rectangleGeometry });
// Изменяем координаты через свойство geometry геообъекта.
rectangleGeoObject.geometry.setCoordinates([[10, 20], [51, 71]]);
// Или напрямую.
rectangleGeometry.setCoordinates([[10, 20], [51, 71]]);
// Также доступ к rectangleGeometry можно получить через rectangleGeoObject.geometry.
```

Поля

Имя	Тип	Описание
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>

События

Имя	Описание
change	<p>Изменение координат углов. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>oldCoordinates</code> - старые координаты углов; <code>newCoordinates</code> - новые координаты углов. <p>Унаследовано от IRectangleGeometryAccess.</p>

Имя	Описание
mapchange	<p>Сменилась карта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IGeometry.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
pixelgeometrychange	<p>Изменилась пиксельная геометрия. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> pixelGeometry - новая пиксельная геометрия IPixelGeometry. <p>Унаследовано от IGeometry.</p>

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	<p>Проверяет, лежит ли переданная точка внутри прямоугольника.</p> <p>Унаследован от IRectangleGeometryAccess.</p>
freeze()	IFreezable	<p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от IFreezable.</p>
getBounds()	Number[][] null	<p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.</p> <p>Унаследован от IGeometry.</p>
getClosest(anchorPosition)	Object	<p>Ищет на контуре прямоугольника точку, ближайшую к anchorPosition.</p> <p>Унаследован от IRectangleGeometryAccess.</p>
getCoordinates()	Number[][]	<p>Возвращает координаты двух противоположных углов прямоугольника.</p> <p>Унаследован от IRectangleGeometryAccess.</p>

Имя	Возвращает	Описание
getMap()	Map null	Возвращает текущую карту. Унаследован от IGeometry .
getPixelGeometry([options])	IPixelGeometry	Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты. Унаследован от IGeometry .
getType()	String	Возвращает строку "Rectangle". Унаследован от IRectangleGeometry .
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable .
setCoordinates(coordinates)	IRectangleGeometryAccess	Задаёт координаты двух противоположных углов прямоугольника. Унаследован от IRectangleGeometryAccess .
setMap(map)		Задаёт карту. Унаследован от IGeometry .
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable .

geometryEditor

geometryEditor.Circle

Расширяет [IGeometryEditor](#).

Редактор геометрии "Круг".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometryEditor.Circle(geometry[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: ICircleGeometry Геометрия "Круг".

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции редактора геометрии.
options.drawingCursor	"arrow"	Тип: Boolean Курсор мыши в режиме рисования.
options.drawOver	true	Тип: Boolean Разрешает ставить точки поверх объектов карты в режиме рисования.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
geometry	IGeometry	Редактируемая геометрия. Унаследовано от IGeometryEditor .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
state	IDataManager	Менеджер состояния редактора геометрии. Поля данных, доступные посредством методов get и set: <ul style="list-style-type: none"> • <code>editing</code> - определяет включен ли редактор. Тип - Boolean. Значение по умолчанию - false. • <code>drawing</code> - определяет включен ли режим рисования. Тип - Boolean. Значение по умолчанию - false.

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
statechange	Изменение состояния редактора геометрии. Экземпляр класса Event . Унаследовано от IGeometryEditor .

Методы

Имя	Возвращает	Описание
startDrawing()	vow.Promise	Включает режим рисования круга.

Имя	Возвращает	Описание
startEditing()		Включает режим редактирования. Унаследован от IGeometryEditor .
stopDrawing()	vow.Promise	Отключает режим рисования круга.
stopEditing()		Отключает режим редактирования. Унаследован от IGeometryEditor .

Описание полей

state

```
{IDataManager} state
```

Менеджер состояния редактора геометрии.

Поля данных, доступные посредством методов get и set:

- editing - определяет включен ли редактор. Тип - Boolean. Значение по умолчанию - false.
- drawing - определяет включен ли режим рисования. Тип - Boolean. Значение по умолчанию - false.

Описание методов

startDrawing

```
{vow.Promise} startDrawing()
```

Включает режим рисования круга.

Возвращает объект-promise.

stopDrawing

```
{vow.Promise} stopDrawing()
```

Отключает режим рисования круга.

Возвращает объект-promise.

geometryEditor.LineString

Расширяет [IGeometryEditor](#).

Редактор геометрии "Ломаная линия".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometryEditor.LineString(geometry[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	<p>Тип: ILineStringGeometry</p> <p>Геометрия "Ломаная линия".</p>
options	—	<p>Тип: Object</p> <p>Опции редактора геометрии.</p>
options.dblClickHandler	—	<p>Тип: Function</p> <p>Обработчик двойного щелчка по вершине. Принимает ссылку на модель редактируемой вершины. По умолчанию, обработчик определен функцией, удаляющей соответствующую вершину.</p>
options.drawingCursor	"arrow"	<p>Тип: Boolean</p> <p>Курсор мыши в режиме добавления новых вершин.</p>
options.drawOver	true	<p>Тип: Boolean</p> <p>Разрешает ставить точки поверх объектов карты в режиме добавления новых вершин.</p>
options.edgeInteractiveOptions	true	<p>Тип: Boolean</p> <p>Позволяет использовать для промежуточных меток опции с постфиксами, привязанными к текущему состоянию метки. Доступны следующие постфиксы:</p> <ul style="list-style-type: none"> • <code>Hover</code> — опции с данным постфиксом применяются при наведении на метку указателя мыши; • <code>Drag</code> — опции с данным постфиксом применяются при перетаскивании метки. <p>Примеры таких опций: <code>edgeLayoutHover</code>, <code>edgeIconImageSizeActive</code>, <code>edgeIconImageShapeHover</code> и т. д. Если вам не требуется менять опции промежуточных меток в зависимости от их состояния, то нужно отключить данную опцию.</p>

Параметр	Значение по умолчанию	Описание
<code>options.edgeLayout</code>	—	<p>Тип: Function</p> <p>Класс макета для промежуточных меток.</p>
<code>options.maxPoints</code>	Infinity	<p>Тип: Number</p> <p>Максимальное допустимое число вершин в ломаной линии.</p>
<code>options.menuManager</code>	—	<p>Тип: Function</p> <p>Диспетчер контекстного меню, открывающегося при щелчке по вершине. Принимает два аргумента:</p> <ul style="list-style-type: none"> массив объектов, описывающих пункты контекстного меню для этой вершины; ссылку на модель редактируемой вершины. <p>Диспетчер может изменять данные в переданном массиве. Должен возвращать массив объектов, описывающих пункты контекстного меню.</p>
<code>options.minPoints</code>	0	<p>Тип: Number</p> <p>Минимально допустимое число вершин полилинии.</p>
<code>options.useAutoPanInDrawing</code>	true	<p>Тип: Boolean</p> <p>Включается режим автоматического передвижения карты при перетаскивании вершины на границе.</p>
<code>options.useMapMarginInDrawing</code>	true	<p>Тип: Boolean</p> <p>Нужно ли учитывать отступы карты в режиме рисования.</p>

Параметр	Значение по умолчанию	Описание
options.vertexInteractiveOptions	true	<p>Тип: Boolean</p> <p>Позволяет использовать для меток вершин опции с постфиксами, привязанными к текущему состоянию вершины. Доступны следующие постфиксы:</p> <ul style="list-style-type: none"> Hover — опции с данным постфиксом применяются при наведении на вершину метки указателя мыши; Drag — опции с данным постфиксом применяются при перетаскивании вершины. Active — опции с данным постфиксом применяются когда для вершины открыто контекстное меню. <p>Примеры таких опций: vertexLayoutHover, vertexIconImageSizeActive, vertexIconImageShapeHover и т. д. Если вам не требуется менять опции меток вершин в зависимости от их состояния, то нужно отключить данную опцию.</p>
options.vertexLayout	—	<p>Тип: Function</p> <p>Класс макета для меток на вершинах ломаной линии.</p>

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
geometry	IGeometry	<p>Редактируемая геометрия.</p> <p>Унаследовано от IGeometryEditor.</p>
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Тип	Описание
state	IDataManager	<p>Менеджер состояния редактора геометрии.</p> <p>Поля данных, доступные посредством методов <code>get</code> и <code>set</code>:</p> <ul style="list-style-type: none"> <code>editing</code> - определяет включен ли редактор. Тип - <code>Boolean</code>. Значение по умолчанию - <code>false</code>. <code>drawing</code> - определяет включен ли режим добавления новых вершин. Тип - <code>Boolean</code>. Значение по умолчанию - <code>false</code>. <code>drawingFrom</code> - определяет то, как добавляются новые точки в режиме добавления. Может принимать одно из двух строковых значений: <code>"begin"</code> - точки добавляются в начало ломаной линии, <code>"end"</code> - точки добавляются в конец. Значение по умолчанию - <code>"end"</code>.

События

Имя	Описание
beforeedgedrag	<p>Событие, предшествующее событию <code>"edgedrag"</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; <code>edgeModel</code> - ссылка на модель перетаскиваемой метки; <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие <code>"edgedrag"</code> будет отменено.</p>

Имя	Описание
beforeedgedragstart	<p>Событие, предшествующее событию "edgedragstart". Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.• <code>edgeModel</code> - ссылка на модель перетаскиваемой метки;• <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "edgedragstart", будет отменено.</p>
beforevertexadd	<p>Событие, предшествующее событию vertexadd. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>parentModel</code> - ссылка на родительскую, для добавляемой вершины, модель данных;• <code>vertexIndex</code> - индекс добавляемой вершины;• <code>globalPixels</code> - координаты добавляемой вершины в глобальных пиксельных координатах. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none">• <code>setGlobalPixels</code> - метод позволяет скорректировать значения координат добавляемой вершины. В качестве аргумента принимает новые глобальные пиксельные координаты вершины в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "vertexadd" будет отменено.</p>

Имя	Описание
beforevertexdrag	<p>Событие, предшествующее событию vertexdrag. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;• vertexModel - ссылка на модель перетаскиваемой вершины;• globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none">• setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "vertexdrag" будет отменено.</p>
beforevertexdragstart	<p>Событие, предшествующее событию vertexdragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.• vertexModel - ссылка на модель перетаскиваемой вершины;• globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "vertexdragstart", будет отменено.</p>

Имя	Описание
beforevertexdraw	<p>Событие, предшествующее событию vertexdraw. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>parentModel</code> - ссылка на родительскую, для добавляемой вершины, модель данных; <code>vertexIndex</code> - индекс добавляемой вершины; <code>globalPixels</code> - координаты добавляемой вершины в глобальных пиксельных координатах. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setGlobalPixels</code> - метод позволяет скорректировать значения координат добавляемой вершины. В качестве аргумента принимает новые глобальные пиксельные координаты вершины в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "vertexdraw" будет отменено.</p>
drawingstart	Включение режима добавления новых вершин. Экземпляр класса Event .
drawingstop	Отключение режима добавления новых вершин. Экземпляр класса Event .
edgedrag	<p>Перетаскивание промежуточной метки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; <code>edgeModel</code> - ссылка на модель перетаскиваемой метки; <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах.
edgedragend	<p>Завершение перетаскивания промежуточной метки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>edgeModel</code> - ссылка на модель перетаскиваемой метки; <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах.

Имя	Описание
edgedragstart	Начало перетаскивания промежуточной метки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. • <code>edgeModel</code> - ссылка на модель перетаскиваемой метки; • <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах.
editingstart	Включение режима редактирования вершин. Экземпляр класса Event .
editingstop	Отключение режима редактирования вершин. Экземпляр класса Event .
framingstart	Включение режима масштабирования. Экземпляр класса Event .
framingstop	Отключение режима масштабирования. Экземпляр класса Event .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
statechange	Изменение состояния редактора геометрии. Экземпляр класса Event . Унаследовано от IGeometryEditor .
vertexadd	Добавление новой вершины. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • <code>parentModel</code> - ссылка на родительскую, для добавленной вершины, модель данных; • <code>vertexIndex</code> - индекс добавленной вершины; • <code>globalPixels</code> - координаты добавленной вершины в глобальных пиксельных координатах.
vertexdrag	Перетаскивание вершины. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>vertexModel</code> - ссылка на модель перетаскиваемой вершины; • <code>globalPixels</code> - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

Имя	Описание
vertexdragend	<p>Завершение перетаскивания вершины. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> vertexModel - ссылка на модель перетаскиваемой вершины; globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах.
vertexdragstart	<p>Начало перетаскивания вершины. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. vertexModel - ссылка на модель перетаскиваемой вершины; globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах.
vertexdraw	<p>Рисование новой вершины. Данное событие, как правило, предшествует событию добавления вершины и возникает в момент движения мышью при включенном режиме добавления новых вершин. На основе данных, передаваемых в данном событии, отображаются направляющие линии в режиме добавления новых вершин. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> parentModel - ссылка на родительскую, для добавляемой вершины, модель данных; vertexIndex - индекс добавляемой вершины; globalPixels - координаты добавляемой вершины в глобальных пиксельных координатах.

Методы

Имя	Возвращает	Описание
getModel()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом модели в момент ее фактического создания, либо отклоняется с одним из следующих кодов ошибки:</p> <ul style="list-style-type: none"> Canceled - режим редактирования отключен до момента фактического создания модели; Editor wasn't started - режим редактирования не включен.

Имя	Возвращает	Описание
<code>getModelSync()</code>	<code>geometryEditor.model.RootLine</code> или <code>null</code>	Возвращает модель данных редактора или <code>null</code> , если на момент вызова она отсутствует.
<code>getView()</code>	<code>vow.Promise</code>	Возвращает объект-обещание, который подтверждается объектом отображения в момент его фактического создания, либо отклоняется с одним из следующих кодов ошибки: <ul style="list-style-type: none"> • <code>Canceled</code> - режим редактирования отключен до момента фактического создания отображения; • <code>Editor wasn't started</code> - режим редактирования не включен.
<code>getViewSync()</code>	<code>geometryEditor.view.Path</code> или <code>null</code>	Возвращает отображение редактора или <code>null</code> , если на момент вызова оно отсутствует.
<code>startDrawing()</code>	<code>vow.Promise</code>	Включает режим добавления новых вершин в ломаную линию. Включение происходит асинхронно.
<code>startEditing()</code>	<code>vow.Promise</code>	Включает режим редактирования. Включение происходит асинхронно.
<code>startFraming()</code>	<code>vow.Promise</code>	Включает режим масштабирования для ломаной линии. Включение происходит асинхронно.
<code>stopDrawing()</code>		Отключает режим добавления новых вершин в ломаную линию.
<code>stopEditing()</code>		Отключает режим редактирования.
<code>stopFraming()</code>		Отключает режим масштабирования.

Описание полей

state

```
{IDataManager} state
```

Менеджер состояния редактора геометрии.

Поля данных, доступные посредством методов `get` и `set`:

- `editing` - определяет включен ли редактор. Тип - `Boolean`. Значение по умолчанию - `false`.
- `drawing` - определяет включен ли режим добавления новых вершин. Тип - `Boolean`. Значение по умолчанию - `false`.
- `drawingFrom` - определяет то, как добавляются новые точки в режиме добавления. Может принимать одно из двух строковых значений: `"begin"` - точки добавляются в начало ломаной линии, `"end"` - точки добавляются в конец. Значение по умолчанию - `"end"`.

Описание событий

`beforeedgedrag`

Событие, предшествующее событию `"edgedrag"`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset` - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- `edgeModel` - ссылка на модель перетаскиваемой метки;
- `globalPixels` - координаты перетаскиваемой метки в глобальных пиксельных координатах.

Имена методов, доступных через метод [Event.callMethod](#):

- `setPixelOffset` - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие `"edgedrag"` будет отменено.

`beforeedgedragstart`

Событие, предшествующее событию `"edgedragstart"`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.
- `edgeModel` - ссылка на модель перетаскиваемой метки;
- `globalPixels` - координаты перетаскиваемой метки в глобальных пиксельных координатах.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие `"edgedragstart"`, будет отменено.

`beforevertexadd`

Событие, предшествующее событию `vertexadd`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `parentModel` - ссылка на родительскую, для добавляемой вершины, модель данных;
- `vertexIndex` - индекс добавляемой вершины;
- `globalPixels` - координаты добавляемой вершины в глобальных пиксельных координатах.

Имена методов, доступных через метод [Event.callMethod](#):

- `setGlobalPixels` - метод позволяет скорректировать значения координат добавляемой вершины. В качестве аргумента принимает новые глобальные пиксельные координаты вершины в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие `"vertexadd"` будет отменено.

Пример:

```
// Корректируем координаты событий добавления вершин так, чтобы
// они не выходили за границы квадрата со стороной
// 100 пикселей и центром, совпадающим с центром карты.
polyline.editor.events.add(["beforevertexdraw", "beforevertexadd"],
function (event) {
    var mapGlobalPixelCenter = geoMap.getGlobalPixelCenter(),
```

```
globalPixels = event.get("globalPixels"),
pixelBounds = [
    [mapGlobalPixelCenter[0] - 100, mapGlobalPixelCenter[1] - 100],
    [mapGlobalPixelCenter[0] + 100, mapGlobalPixelCenter[1] + 100]
];
event.callMethod("setGlobalPixels", [
    Math.max(Math.min(globalPixels[0], pixelBounds[1][0]), pixelBounds[0][0]),
    Math.max(Math.min(globalPixels[1], pixelBounds[1][1]), pixelBounds[0][1])
]);
});
```

beforevertexdrag

Событие, предшествующее событию vertexdrag. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- vertexModel - ссылка на модель перетаскиваемой вершины;
- globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

Имена методов, доступных через метод [Event.callMethod](#):

- setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "vertexdrag" будет отменено.

Пример:

```
// Инвертируем сдвиг при перетаскивании вершины.
polyline.editor.events.add("beforevertexdrag", function (event) {
    var pixelOffset = event.get("pixelOffset");
    event.callMethod("setPixelOffset", [-pixelOffset[0], -pixelOffset[1]]);
});
```

beforevertexdragstart

Событие, предшествующее событию vertexdragstart. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.
- vertexModel - ссылка на модель перетаскиваемой вершины;
- globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие "vertexdragstart", будет отменено.

beforevertexdraw

Событие, предшествующее событию vertexdraw. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- parentModel - ссылка на родительскую, для добавляемой вершины, модель данных;
- vertexIndex - индекс добавляемой вершины;
- globalPixels - координаты добавляемой вершины в глобальных пиксельных координатах.

Имена методов, доступных через метод [Event.callMethod](#):

- setGlobalPixels - метод позволяет скорректировать значения координат добавляемой вершины. В качестве аргумента принимает новые глобальные пиксельные координаты вершины в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "vertexdraw" будет отменено.

drawingstart

Включение режима добавления новых вершин. Экземпляр класса [Event](#).

drawingstop

Отключение режима добавления новых вершин. Экземпляр класса [Event](#).

edgedrag

Перетаскивание промежуточной метки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- edgeModel - ссылка на модель перетаскиваемой метки;
- globalPixels - координаты перетаскиваемой метки в глобальных пиксельных координатах.

edgedragend

Завершение перетаскивания промежуточной метки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- edgeModel - ссылка на модель перетаскиваемой метки;
- globalPixels - координаты перетаскиваемой метки в глобальных пиксельных координатах.

edgedragstart

Начало перетаскивания промежуточной метки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.
- edgeModel - ссылка на модель перетаскиваемой метки;
- globalPixels - координаты перетаскиваемой метки в глобальных пиксельных координатах.

editingstart

Включение режима редактирования вершин. Экземпляр класса [Event](#).

editingstop

Отключение режима редактирования вершин. Экземпляр класса [Event](#).

framingstart

Включение режима масштабирования. Экземпляр класса [Event](#).

framingstop

Отключение режима масштабирования. Экземпляр класса [Event](#).

vertexadd

Добавление новой вершины. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- parentModel - ссылка на родительскую, для добавленной вершины, модель данных;
- vertexIndex - индекс добавленной вершины;
- globalPixels - координаты добавленной вершины в глобальных пиксельных координатах.

vertexdrag

Перетаскивание вершины. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;

- `vertexModel` - ссылка на модель перетаскиваемой вершины;
- `globalPixels` - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

vertexdragend

Завершение перетаскивания вершины. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `vertexModel` - ссылка на модель перетаскиваемой вершины;
- `globalPixels` - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

vertexdragstart

Начало перетаскивания вершины. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.
- `vertexModel` - ссылка на модель перетаскиваемой вершины;
- `globalPixels` - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

vertexdraw

Рисование новой вершины. Данное событие, как правило, предшествует событию добавления вершины и возникает в момент движения мышью при включенном режиме добавления новых вершин. На основе данных, передаваемых в данном событии, отображаются направляющие линии в режиме добавления новых вершин. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `parentModel` - ссылка на родительскую, для добавляемой вершины, модель данных;
- `vertexIndex` - индекс добавляемой вершины;
- `globalPixels` - координаты добавляемой вершины в глобальных пиксельных координатах.

Описание методов

getModel

```
{vow.Promise} getModel()
```

Возвращает объект-обещание, который подтверждается объектом модели в момент ее фактического создания, либо отклоняется с одним из следующих кодов ошибки:

- `Canceled` - режим редактирования отключен до момента фактического создания модели;
- `Editor wasn't started` - режим редактирования не включен.

getModelSync

```
{geometryEditor.model.RootLineString|null} getModelSync()
```

Возвращает модель данных редактора или `null`, если на момент вызова она отсутствует.

getView

```
{vow.Promise} getView()
```

Возвращает объект-обещание, который подтверждается объектом отображения в момент его фактического создания, либо отклоняется с одним из следующих кодов ошибки:

- `Canceled` - режим редактирования отключен до момента фактического создания отображения;
- `Editor wasn't started` - режим редактирования не включен.

getViewSync

```
{geometryEditor.view.Path|null} getViewSync()
```

Возвращает отображение редактора или null, если на момент вызова оно отсутствует.

startDrawing

```
{vow.Promise} startDrawing()
```

Включает режим добавления новых вершин в ломаную линию. Включение происходит асинхронно.

Возвращает объект-обещание, который подтверждается при включении режима добавления новых вершин, либо отклоняется с одним из следующих кодов ошибки:

- Canceled - режим добавления новых вершин отключен до момента его фактического старта.

startEditing

```
{vow.Promise} startEditing()
```

Включает режим редактирования. Включение происходит асинхронно.

Возвращает объект-обещание, который подтверждается в момент фактического старта режима редактирования, либо отклоняется с одним из следующих кодов ошибки:

- Canceled - режим редактирования отключен до момента его фактического старта.

startFraming

```
{vow.Promise} startFraming()
```

Включает режим масштабирования для ломаной линии. Включение происходит асинхронно.

Возвращает Promise, который подтверждается в момент фактического старта режима масштабирования.

stopDrawing

```
{ } stopDrawing()
```

Отключает режим добавления новых вершин в ломаную линию.

stopEditing

```
{ } stopEditing()
```

Отключает режим редактирования.

stopFraming

```
{ } stopFraming()
```

Отключает режим масштабирования.

geometryEditor.model

geometryEditor.model.ChildLinearRing

Примечание: Конструктор класса `geometryEditor.model.ChildLinearRing` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [geometryEditor.model.ChildLineString](#).

Модель дочернего замкнутого контура. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [Методы](#)

Поля

Имя	Тип	Описание
editor	IGeometryEditor	Редактор геометрии. Унаследовано от IGeometryEditorChildModel .
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
geometry	IBaseGeometry	Геометрия модели. Унаследовано от IGeometryEditorChildModel .

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel .
getAllVerticesNumber()	Integer	Возвращает суммарное количество вершин редактируемой геометрии. Унаследован от geometryEditor.model.ChildLineString .
getEdgeModels()	geometryEditor.model.Edge[]	Возвращает массив моделей промежуточных меток. Унаследован от geometryEditor.model.ChildLineString .
getIndex()	Integer	Возвращает индекс дочерней ломаной в родительской модели. Унаследован от geometryEditor.model.ChildLineString .
getParent()	IGeometryEditorModel	Возвращает родительскую модель данных. Унаследован от IGeometryEditorChildModel .
getPixels()	Number[]	Возвращает пиксельные данные модели. Унаследован от IGeometryEditorModel .
getVertexModels()	geometryEditor.model.ChildVertex[]	Возвращает массив моделей дочерних вершин. Унаследован от geometryEditor.model.ChildLineString .

Имя	Возвращает	Описание
setIndex(index)		Задаёт индекс дочерней ломаной в родительской модели. Унаследован от geometryEditor.model.ChildLineString .
setPixels(pixels)		Задаёт пиксельные данные модели. Унаследован от IGeometryEditorChildModel .
spliceVertices(start, deleteCount)	Number[][]	Удаляет определенное число вершин ломаной начиная с заданного индекса. При этом на место удаляемых вершин могут быть добавлены новые. Координаты новых вершин в глобальных пикселях можно передавать дополнительными аргументами после параметра deleteCount. Унаследован от geometryEditor.model.ChildLineString .

geometryEditor.model.ChildLineString

Примечание: Конструктор класса `geometryEditor.model.ChildLineString` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeometryEditorChildModel](#).

Модель дочерней ломаной линии. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [Методы](#)

Поля

Имя	Тип	Описание
editor	IGeometryEditor	Редактор геометрии. Унаследовано от IGeometryEditorChildModel .
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
geometry	IBaseGeometry	Геометрия модели. Унаследовано от IGeometryEditorChildModel .

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel .

Имя	Возвращает	Описание
<code>getAllVerticesNumber()</code>	Integer	Возвращает суммарное количество вершин редактируемой геометрии.
<code>getEdgeModels()</code>	<code>geometryEditor.model.Edge[]</code>	Возвращает массив моделей промежуточных меток.
<code>getIndex()</code>	Integer	Возвращает индекс дочерней ломаной в родительской модели.
<code>getParent()</code>	<code>IGeometryEditorModel</code>	Возвращает родительскую модель данных. Унаследован от <code>IGeometryEditorChildModel</code> .
<code>getPixels()</code>	Number[]	Возвращает пиксельные данные модели. Унаследован от <code>IGeometryEditorModel</code> .
<code>getVertexModels()</code>	<code>geometryEditor.model.ChildVertex[]</code>	Возвращает массив моделей дочерних вершин.
<code>setIndex(index)</code>		Задаёт индекс дочерней ломаной в родительской модели.
<code>setPixels(pixels)</code>		Задаёт пиксельные данные модели. Унаследован от <code>IGeometryEditorChildModel</code> .
<code>spliceVertices(start, deleteCount)</code>	Number[][]	Удаляет определенное число вершин ломаной начиная с заданного индекса. При этом на место удаляемых вершин могут быть добавлены новые. Координаты новых вершин в глобальных пикселях можно передавать дополнительными аргументами после параметра <code>deleteCount</code> .

Описание методов

getAllVerticesNumber

```
{Integer} getAllVerticesNumber()
```

Возвращает суммарное количество вершин редактируемой геометрии.

getEdgeModels

```
{geometryEditor.model.Edge[]} getEdgeModels()
```

Возвращает массив моделей промежуточных меток.

getIndex

```
{Integer} getIndex()
```

Возвращает индекс дочерней ломаной в родительской модели.

getVertexModels

```
{geometryEditor.model.ChildVertex[]} getVertexModels()
```

Возвращает массив моделей дочерних вершин.

setIndex

```
{ } setIndex(index)
```

Задаёт индекс дочерней ломаной в родительской модели.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index *</code>	—	Тип: Integer Индекс дочерней вершины.

* Обязательный параметр/опция.

spliceVertices

```
{Number[][]} spliceVertices(start, deleteCount)
```

Удаляет определенное число вершин ломаной начиная с заданного индекса. При этом на место удаляемых вершин могут быть добавлены новые. Координаты новых вершин в глобальных пикселях можно передавать дополнительными аргументами после параметра `deleteCount`.

Возвращает массив координат удаленных вершин.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>start *</code>	—	Тип: Integer Индекс, начиная с которого производится удаление и добавление вершин.
<code>deleteCount *</code>	—	Тип: Integer Количество удаляемых вершин.

* Обязательный параметр/опция.

geometryEditor.model.ChildVertex

Примечание: Конструктор класса `geometryEditor.model.ChildVertex` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeometryEditorChildModel](#).

Модель дочерней вершины. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [Методы](#)

Поля

Имя	Тип	Описание
editor	IGeometryEditor	Редактор геометрии. Унаследовано от IGeometryEditorChildModel .
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
geometry	IBaseGeometry	Геометрия модели. Унаследовано от IGeometryEditorChildModel .

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel .
getAllVerticesNumber()	Integer	Возвращает суммарное количество вершин редактируемой геометрии.
getIndex()	Integer	Возвращает индекс дочерней вершины в родительской модели.
getNextVertex()	geometryEditor.model.ChildVertex null	Возвращает ссылку на модель следующей вершины.
getParent()	IGeometryEditorModel	Возвращает родительскую модель данных. Унаследован от IGeometryEditorChildModel .
getPixels()	Number[]	Возвращает пиксельные данные модели. Унаследован от IGeometryEditorModel .
getPrevVertex()	geometryEditor.model.ChildVertex null	Возвращает ссылку на модель предыдущей вершины.
setGlobalPixels(pixels)		Задаёт глобальные пиксельные координаты вершины.
setIndex(index)		Задаёт индекс дочерней вершины в родительской модели.

Имя	Возвращает	Описание
<code>setNextVertex(nextVertex)</code>		Устанавливает ссылку на модель следующей вершины.
<code>setPixels(pixels)</code>		Задаёт пиксельные данные модели. Унаследован от <code>IGeometryEditorChildModel</code> .
<code>setPrevVertex(prevVertex)</code>		Устанавливает ссылку на модель предыдущей вершины.

Описание методов

`getAllVerticesNumber`

```
{Integer} getAllVerticesNumber()
```

Возвращает суммарное количество вершин редактируемой геометрии.

`getIndex`

```
{Integer} getIndex()
```

Возвращает индекс дочерней вершины в родительской модели.

`getNextVertex`

```
{geometryEditor.model.ChildVertex|null} getNextVertex()
```

Возвращает ссылку на модель следующей вершины.

`getPrevVertex`

```
{geometryEditor.model.ChildVertex|null} getPrevVertex()
```

Возвращает ссылку на модель предыдущей вершины.

`setGlobalPixels`

```
{ } setGlobalPixels(pixels)
```

Задаёт глобальные пиксельные координаты вершины.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>pixels</code> *	—	Тип: <code>Number[]</code> Глобальные пиксельные координаты вершины.

* Обязательный параметр/опция.

setIndex

```
{ } setIndex(index)
```

Задаёт индекс дочерней вершины в родительской модели.

Параметры:

Параметр	Значение по умолчанию	Описание
index *	—	Тип: Integer Индекс дочерней вершины.

* Обязательный параметр/опция.

setNextVertex

```
{ } setNextVertex(nextVertex)
```

Устанавливает ссылку на модель следующей вершины.

Параметры:

Параметр	Значение по умолчанию	Описание
nextVertex *	—	Тип: geometryEditor.model.ChildVertex null Модель следующей вершины.

* Обязательный параметр/опция.

setPrevVertex

```
{ } setPrevVertex(prevVertex)
```

Устанавливает ссылку на модель предыдущей вершины.

Параметры:

Параметр	Значение по умолчанию	Описание
prevVertex *	—	Тип: geometryEditor.model.ChildVertex null Модель предыдущей вершины.

* Обязательный параметр/опция.

geometryEditor.model.Edge

Примечание: Конструктор класса `geometryEditor.model.Edge` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeometryEditorRootModel](#).

Модель промежуточной метки. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [Методы](#)

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel .
getNextVertex()	geometryEditor.model.ChildVertex или null	Возвращает ссылку на модель следующей вершины.
getPixels()	Number[]	Возвращает пиксельные данные модели. Унаследован от IGeometryEditorModel .
getPrevVertex()	geometryEditor.model.ChildVertex или null	Возвращает ссылку на модель предыдущей вершины.
setNextVertex(nextVertex)		Устанавливает ссылку на модель следующей вершины.
setPrevVertex(prevVertex)		Устанавливает ссылку на модель предыдущей вершины.

Описание методов

getNextVertex

```
{geometryEditor.model.ChildVertex|null} getNextVertex()
```

Возвращает ссылку на модель следующей вершины.

getPrevVertex

```
{geometryEditor.model.ChildVertex|null} getPrevVertex()
```

Возвращает ссылку на модель предыдущей вершины.

setNextVertex

```
{ } setNextVertex(nextVertex)
```

Устанавливает ссылку на модель следующей вершины.

Параметры:

Параметр	Значение по умолчанию	Описание
nextVertex *	—	Тип: geometryEditor.model.ChildVertex null Модель следующей вершины.

* Обязательный параметр/опция.

setPrevVertex

```
{ } setPrevVertex(prevVertex)
```

Устанавливает ссылку на модель предыдущей вершины.

Параметры:

Параметр	Значение по умолчанию	Описание
prevVertex *	—	Тип: geometryEditor.model.ChildVertex null Модель предыдущей вершины.

* Обязательный параметр/опция.

geometryEditor.model.EdgeGeometry

Примечание: Конструктор класса `geometryEditor.model.EdgeGeometry` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeometry](#).

Геометрия промежуточной метки. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [События](#) | [Методы](#)

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
mapchange	<p>Сменилась карта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IGeometry.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
pixelgeometrychange	<p>Изменилась пиксельная геометрия. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> pixelGeometry - новая пиксельная геометрия IPixelGeometry. <p>Унаследовано от IGeometry.</p>

Методы

Имя	Возвращает	Описание
getBounds()	Number[][] null	<p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.</p> <p>Унаследован от IGeometry.</p>
getMap()	Map null	<p>Возвращает текущую карту.</p> <p>Унаследован от IGeometry.</p>
getPixelGeometry([options])	IPixelGeometry	<p>Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.</p> <p>Унаследован от IGeometry.</p>
getType()	String	<p>Возвращает идентификатор типа геометрии.</p> <p>Унаследован от IBaseGeometry.</p>
setMap(map)		<p>Задает карту.</p> <p>Унаследован от IGeometry.</p>

geometryEditor.model.RootLineString

Примечание: Конструктор класса `geometryEditor.model.RootLineString` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeometryEditorRootModel](#).

Модель корневой ломаной линии. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [Методы](#)

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel .
getAllVerticesNumber()	Integer	Возвращает суммарное количество вершин редактируемой геометрии.
getPixels()	Number[]	Возвращает пиксельные данные модели. Унаследован от IGeometryEditorModel .
getVertexModels()	geometryEditor.model.ChildVertex	Возвращает массив моделей дочерних вершин.
spliceVertices(start, deleteCount)	Number[][]	Удаляет определенное число вершин ломаной начиная с заданного индекса. При этом на место удаляемых вершин могут быть добавлены новые. Координаты новых вершин в глобальных пикселях можно передавать дополнительными аргументами после параметра <code>deleteCount</code> .

Описание методов

[getAllVerticesNumber](#)

```
{Integer} getAllVerticesNumber()
```

Возвращает суммарное количество вершин редактируемой геометрии.

[getVertexModels](#)

```
{geometryEditor.model.ChildVertex[]} getVertexModels()
```

Возвращает массив моделей дочерних вершин.

spliceVertices

```
{Number[][]} spliceVertices(start, deleteCount)
```

Удаляет определенное число вершин ломаной начиная с заданного индекса. При этом на место удаляемых вершин могут быть добавлены новые. Координаты новых вершин в глобальных пикселях можно передавать дополнительными аргументами после параметра deleteCount.

Возвращает массив координат удаленных вершин.

Параметры:

Параметр	Значение по умолчанию	Описание
start *	—	Тип: Integer Индекс, начиная с которого производится удаление и добавление вершин.
deleteCount *	—	Тип: Integer Количество удаляемых вершин.

* Обязательный параметр/опция.

geometryEditor.model.RootPolygon

Примечание: Конструктор класса `geometryEditor.model.RootPolygon` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeometryEditorRootModel](#).

Модель корневого полигона. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [Методы](#)

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel .
getAllVerticesNumber()	Integer	Возвращает суммарное количество вершин редактируемой геометрии.
getPathModels()	geometryEditor.model.ChildLineModel	Возвращает массив моделей дочерних контуров.

Имя	Возвращает	Описание
<code>getPixels()</code>	<code>Number[]</code>	Возвращает пиксельные данные модели. Унаследован от IGeometryEditorModel .
<code>splicePaths(start, deleteCount)</code>	<code>Number[][]</code>	Удаляет определенное число контуров многоугольника начиная с заданного индекса. При этом на место удаляемых контуров могут быть добавлены новые. Координаты новых контуров в глобальных пикселях можно передавать дополнительными аргументами после параметра <code>deleteCount</code> .

Описание методов

`getAllVerticesNumber`

```
{Integer} getAllVerticesNumber()
```

Возвращает суммарное количество вершин редактируемой геометрии.

`getPathModels`

```
{geometryEditor.model.ChildLinearRing[]} getPathModels()
```

Возвращает массив моделей дочерних контуров.

`splicePaths`

```
{Number[][]} splicePaths(start, deleteCount)
```

Удаляет определенное число контуров многоугольника начиная с заданного индекса. При этом на место удаляемых контуров могут быть добавлены новые. Координаты новых контуров в глобальных пикселях можно передавать дополнительными аргументами после параметра `deleteCount`.

Возвращает массив координат удаленных контуров.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>start</code> *	—	Тип: Integer Индекс, начиная с которого производится удаление и добавление контуров.
<code>deleteCount</code> *	—	Тип: Integer Количество удаляемых контуров.

* Обязательный параметр/опция.

geometryEditor.Point

Расширяет [IGeometryEditor](#).

Редактор геометрии "Точка".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometryEditor.Point(geometry[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPointGeometry Геометрия "Точка".
options	—	Тип: Object Опции редактора геометрии.
options.dblClickHandler	—	Тип: Function Обработчик двойного щелчка по вершине. Принимает ссылку на модель редактируемой вершины. По умолчанию, обработчик определен функцией, удаляющей соответствующую вершину.
options.drawingCursor	"arrow"	Тип: Boolean Курсор мыши в режиме рисования.
options.drawOver	true	Тип: Boolean Разрешает ставить точки поверх объектов карты в режиме рисования.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
geometry	IGeometry	Редактируемая геометрия. Унаследовано от IGeometryEditor .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

Имя	Тип	Описание
state	IDataManager	<p>Менеджер состояния редактора геометрии.</p> <p>Поля данных, доступные посредством методов get и set:</p> <ul style="list-style-type: none"> • <code>editing</code> - определяет включен ли редактор. Тип - Boolean. Значение по умолчанию - false. • <code>drawing</code> - определяет включен ли режим рисования. Тип - Boolean. Значение по умолчанию - false.

События

Имя	Описание
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
statechange	<p>Изменение состояния редактора геометрии. Экземпляр класса Event.</p> <p>Унаследовано от IGeometryEditor.</p>

Методы

Имя	Возвращает	Описание
startDrawing()	vow.Promise	Включает режим рисования точки.
startEditing()		<p>Включает режим редактирования.</p> <p>Унаследован от IGeometryEditor.</p>
stopDrawing()	vow.Promise	Отключает режим рисования точки.
stopEditing()		<p>Отключает режим редактирования.</p> <p>Унаследован от IGeometryEditor.</p>

Описание полей

state

```
{IDataManager} state
```

Менеджер состояния редактора геометрии.

Поля данных, доступные посредством методов get и set:

- `editing` - определяет включен ли редактор. Тип - Boolean. Значение по умолчанию - false.
- `drawing` - определяет включен ли режим рисования. Тип - Boolean. Значение по умолчанию - false.

Описание методов

startDrawing

```
{vow.Promise} startDrawing()
```

Включает режим рисования точки.

Возвращает объект-promise.

stopDrawing

```
{vow.Promise} stopDrawing()
```

Отключает режим рисования точки.

Возвращает объект-promise.

geometryEditor.Polygon

Расширяет [IGeometryEditor](#).

Редактор геометрии "Многоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geometryEditor.Polygon(geometry[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPolygonGeometry Геометрия "Многоугольник".
options	—	Тип: Object Опции редактора геометрии.
options.dblClickHandler	—	Тип: Function Обработчик двойного щелчка по вершине. Принимает ссылку на модель редактируемой вершины. По умолчанию, обработчик определен функцией, удаляющей соответствующую вершину.
options.drawingCursor	"arrow"	Тип: Boolean Курсор мыши в режиме добавления новых вершин.
options.drawOver	true	Тип: Boolean Разрешает ставить точки поверх объектов карты в режиме добавления новых вершин.

Параметр	Значение по умолчанию	Описание
options.edgeInteractiveOptions	true	<p>Тип: Boolean</p> <p>Позволяет использовать для промежуточных меток опции с постфиксами, привязанными к текущему состоянию метки. Доступны следующие постфиксы:</p> <ul style="list-style-type: none"> • Drag — опции с данным постфиксом применяются при перетаскивании метки. • Hover — опции с данным постфиксом применяются при наведении на метку указателя мыши; <p>Примеры таких опций: edgeLayoutHover, edgeIconImageSizeActive, edgeIconImageShapeHover и т. д. Если вам не требуется менять опции промежуточных меток в зависимости от их состояния, то нужно отключить данную опцию.</p>
options.edgeLayout	—	<p>Тип: Function</p> <p>Класс макета для промежуточных меток.</p>
options.maxPoints	Infinity	<p>Тип: Number</p> <p>Максимально допустимое число вершин в многоугольнике.</p>

Параметр	Значение по умолчанию	Описание
<code>options.menuManager</code>	—	<p>Тип: Function</p> <p>Диспетчер контекстного меню, открывающегося при щелчке по вершине. Принимает два аргумента:</p> <ul style="list-style-type: none">• массив объектов, описывающих пункты контекстного меню для этой вершины;• ссылку на модель редактируемой вершины. <p>Диспетчер может изменять данные в переданном массиве. Должен возвращать массив объектов, описывающих пункты контекстного меню.</p>
<code>options.minPoints</code>	0	<p>Тип: Number</p> <p>Минимальное допустимое число вершин в многоугольнике.</p>
<code>options.useAutoPanInDrawing</code>	true	<p>Тип: Boolean</p> <p>Включается режим автоматического передвижения карты при перетаскивании вершины на границе.</p>
<code>options.useMapMarginInDrawing</code>	true	<p>Тип: Boolean</p> <p>Нужно ли учитывать отступы карты в режиме рисования.</p>

Параметр	Значение по умолчанию	Описание
<code>options.vertexInteractiveOptions</code>	true	<p>Тип: Boolean</p> <p>Позволяет использовать для меток вершин опции с постфиксами, привязанными к текущему состоянию вершины. Доступны следующие постфиксы:</p> <ul style="list-style-type: none"> Hover — опции с данным постфиксом применяются при наведении на вершину метки указателя мыши; Drag — опции с данным постфиксом применяются при перетаскивании вершины. Active — опции с данным постфиксом применяются когда для вершины открыто контекстное меню. <p>Примеры таких опций: <code>vertexLayoutHover</code>, <code>vertexIconImageSizeActive</code>, <code>vertexIconImageShapeHover</code> и т. д. Если вам не требуется менять опции меток вершин в зависимости от их состояния, то нужно отключить данную опцию.</p>
<code>options.vertexLayout</code>	—	<p>Тип: Function</p> <p>Класс макета для меток на вершинах многоугольника.</p>

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	<code>IEventManager</code>	<p>Менеджер событий.</p> <p>Унаследовано от <code>IEventEmitter</code>.</p>
<code>geometry</code>	<code>IGeometry</code>	<p>Редактируемая геометрия.</p> <p>Унаследовано от <code>IGeometryEditor</code>.</p>
<code>options</code>	<code>IOptionManager</code>	<p>Менеджер опций.</p> <p>Унаследовано от <code>ICustomizable</code>.</p>

Имя	Тип	Описание
state	IDataManager	<p>Менеджер состояния редактора геометрии.</p> <p>Поля данных, доступные посредством методов <code>get</code> и <code>set</code>:</p> <ul style="list-style-type: none"> <code>editing</code> - определяет включен ли редактор. Тип - <code>Boolean</code>. Значение по умолчанию - <code>false</code>. <code>drawing</code> - определяет включен ли режим добавления новых вершин. Тип - <code>Boolean</code>. Значение по умолчанию - <code>false</code>. <code>drawingFrom</code> - определяет то, как добавляются новые точки в режиме добавления. Может принимать одно из двух строковых значений: <code>"begin"</code> - точки добавляются в начало многоугольника, <code>"end"</code> - точки добавляются в конец. <code>drawingFromIndex</code> - индекс вершины многоугольника, после которой будут добавляться новые точки в режиме добавления. Данное поле доступно только на запись, т.к. факт записи приводит к изменению порядка вершин в многоугольнике, так чтобы точка с заданным индексом стала последней по счету. Тип - целое число. <code>drawingPath</code> - индекс контура многоугольника, в который добавляются новые точки в режиме добавления. Тип - целое число. Значение по умолчанию - <code>0</code>.

События

Имя	Описание
beforeedgedrag	<p>Событие, предшествующее событию <code>"edgedrag"</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; <code>edgeModel</code> - ссылка на модель перетаскиваемой метки; <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие <code>"edgedrag"</code> будет отменено.</p>

Имя	Описание
beforeedgedragstart	<p>Событие, предшествующее событию "edgedragstart". Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.• <code>edgeModel</code> - ссылка на модель перетаскиваемой метки;• <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "edgedragstart", будет отменено.</p>
beforevertexadd	<p>Событие, предшествующее событию vertexadd. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>parentModel</code> - ссылка на родительскую, для добавляемой вершины, модель данных;• <code>vertexIndex</code> - индекс добавляемой вершины;• <code>globalPixels</code> - координаты добавляемой вершины в глобальных пиксельных координатах. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none">• <code>setGlobalPixels</code> - метод позволяет скорректировать значения координат добавляемой вершины. В качестве аргумента принимает новые глобальные пиксельные координаты вершины в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "vertexadd" будет отменено.</p>

Имя	Описание
beforevertexdrag	<p>Событие, предшествующее событию vertexdrag. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;• vertexModel - ссылка на модель перетаскиваемой вершины;• globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none">• setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "vertexdrag" будет отменено.</p>
beforevertexdragstart	<p>Событие, предшествующее событию vertexdragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.• vertexModel - ссылка на модель перетаскиваемой вершины;• globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "vertexdragstart", будет отменено.</p>

Имя	Описание
beforevertexdraw	<p>Событие, предшествующее событию vertexdraw. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>parentModel</code> - ссылка на родительскую, для добавляемой вершины, модель данных; <code>vertexIndex</code> - индекс добавляемой вершины; <code>globalPixels</code> - координаты добавляемой вершины в глобальных пиксельных координатах. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setGlobalPixels</code> - метод позволяет скорректировать значения координат добавляемой вершины. В качестве аргумента принимает новые глобальные пиксельные координаты вершины в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "vertexdraw" будет отменено.</p>
drawingstart	Включение режима добавления новых вершин. Экземпляр класса Event .
drawingstop	Отключение режима добавления новых вершин. Экземпляр класса Event .
edgedrag	<p>Перетаскивание промежуточной метки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; <code>edgeModel</code> - ссылка на модель перетаскиваемой метки; <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах.
edgedragend	<p>Завершение перетаскивания промежуточной метки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>edgeModel</code> - ссылка на модель перетаскиваемой метки; <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах.

Имя	Описание
edgedragstart	Начало перетаскивания промежуточной метки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. • <code>edgeModel</code> - ссылка на модель перетаскиваемой метки; • <code>globalPixels</code> - координаты перетаскиваемой метки в глобальных пиксельных координатах.
editingstart	Включение режима редактирования вершин. Экземпляр класса Event .
editingstop	Отключение режима редактирования вершин. Экземпляр класса Event .
framingstart	Включение режима масштабирования. Экземпляр класса Event .
framingstop	Отключение режима масштабирования. Экземпляр класса Event .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
statechange	Изменение состояния редактора геометрии. Экземпляр класса Event . Унаследовано от IGeometryEditor .
vertexadd	Добавление новой вершины. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • <code>parentModel</code> - ссылка на родительскую, для добавленной вершины, модель данных; • <code>vertexIndex</code> - индекс добавленной вершины; • <code>globalPixels</code> - координаты добавленной вершины в глобальных пиксельных координатах.
vertexdrag	Перетаскивание вершины. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>vertexModel</code> - ссылка на модель перетаскиваемой вершины; • <code>globalPixels</code> - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

Имя	Описание
vertexdragend	<p>Завершение перетаскивания вершины. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> vertexModel - ссылка на модель перетаскиваемой вершины; globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах.
vertexdragstart	<p>Начало перетаскивания вершины. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. vertexModel - ссылка на модель перетаскиваемой вершины; globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах.
vertexdraw	<p>Рисование новой вершины. Данное событие, как правило, предшествует событию добавления вершины и возникает в момент движения мышью при включенном режиме добавления новых вершин. На основе данных, передаваемых в данном событии, отображаются направляющие линии в режиме добавления новых вершин. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> parentModel - ссылка на родительскую, для добавляемой вершины, модель данных; vertexIndex - индекс добавляемой вершины; globalPixels - координаты добавляемой вершины в глобальных пиксельных координатах.

Методы

Имя	Возвращает	Описание
getModel()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом модели в момент ее фактического создания, либо отклоняется с одним из следующих кодов ошибки:</p> <ul style="list-style-type: none"> Canceled - режим редактирования отключен до момента фактического создания модели; Editor wasn't started - режим редактирования не включен.

Имя	Возвращает	Описание
<code>getModelSync()</code>	<code>geometryEditor.model.RootPoly</code> null	Возвращает модель редактора или null, если на момент вызова она отсутствует.
<code>getView()</code>	<code>vow.Promise</code>	Возвращает объект-обещание, который подтверждается объектом отображения в момент его фактического создания, либо отклоняется с одним из следующих кодов ошибки: <ul style="list-style-type: none"> • Canceled - режим редактирования отключен до момента фактического создания отображения; • Editor wasn't started - режим редактирования не включен.
<code>getViewSync()</code>	<code>geometryEditor.view.MultiPath</code> null	Возвращает отображение редактора или null, если на момент вызова оно отсутствует.
<code>startDrawing()</code>	<code>vow.Promise</code>	Включает режим добавления новых вершин в многоугольник. Включение происходит асинхронно.
<code>startEditing()</code>	<code>vow.Promise</code>	Включает режим редактирования многоугольника (для добавления новых вершин). Включение происходит асинхронно.
<code>startFraming()</code>	<code>vow.Promise</code>	Включает режим масштабирования для многоугольника. Включение происходит асинхронно.
<code>stopDrawing()</code>		Отключает режим добавления новых вершин в многоугольник.
<code>stopEditing()</code>		Отключает режим редактирования.
<code>stopFraming()</code>		Отключает режим масштабирования.

Описание полей**state**

```
{IDataManager} state
```

Менеджер состояния редактора геометрии.

Поля данных, доступные посредством методов `get` и `set`:

- `editing` - определяет включен ли редактор. Тип - `Boolean`. Значение по умолчанию - `false`.
- `drawing` - определяет включен ли режим добавления новых вершин. Тип - `Boolean`. Значение по умолчанию - `false`.
- `drawingFrom` - определяет то, как добавляются новые точки в режиме добавления. Может принимать одно из двух строковых значений: `"begin"` - точки добавляются в начало многоугольника, `"end"` - точки добавляются в конец.
- `drawingFromIndex` - индекс вершины многоугольника, после которой будут добавляться новые точки в режиме добавления. Данное поле доступно только на запись, т.к. факт записи приводит к изменению порядка вершин в многоугольнике, так чтобы точка с заданным индексом стала последней по счету. Тип - целое число.
- `drawingPath` - индекс контура многоугольника, в который добавляются новые точки в режиме добавления. Тип - целое число. Значение по умолчанию - 0.

Описание событий

`beforeedgedrag`

Событие, предшествующее событию `"edgedrag"`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset` - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- `edgeModel` - ссылка на модель перетаскиваемой метки;
- `globalPixels` - координаты перетаскиваемой метки в глобальных пиксельных координатах.

Имена методов, доступных через метод [Event.callMethod](#):

- `setPixelOffset` - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие `"edgedrag"` будет отменено.

`beforeedgedragstart`

Событие, предшествующее событию `"edgedragstart"`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.
- `edgeModel` - ссылка на модель перетаскиваемой метки;
- `globalPixels` - координаты перетаскиваемой метки в глобальных пиксельных координатах.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие `"edgedragstart"`, будет отменено.

`beforevertexadd`

Событие, предшествующее событию `vertexadd`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `parentModel` - ссылка на родительскую, для добавляемой вершины, модель данных;
- `vertexIndex` - индекс добавляемой вершины;
- `globalPixels` - координаты добавляемой вершины в глобальных пиксельных координатах.

Имена методов, доступных через метод [Event.callMethod](#):

- `setGlobalPixels` - метод позволяет скорректировать значения координат добавляемой вершины. В качестве аргумента принимает новые глобальные пиксельные координаты вершины в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "vertexadd" будет отменено.

Примеры:

1.

```
// Запрещаем добавлять новые вершины на расстоянии более 100 пикселей от центра карты.
polygon.editor.events.add("beforevertexadd", function (event) {
    var mapGlobalPixelCenter = geoMap.getGlobalPixelCenter(),
        globalPixels = event.get("globalPixels"),
        vector = [mapGlobalPixelCenter[0] - globalPixels[0], mapGlobalPixelCenter[1] - globalPixels[1]],
        vectorLength = Math.sqrt(vector[0] * vector[0] + vector[1] * vector[1]);
    if (dist > 100) {
        event.preventDefault();
    }
});
```

2.

```
// Корректируем координаты событий добавления вершин так,
// чтобы они не выходили за границы квадрата со стороной
// 100 пикселей и центром, совпадающим с центром карты.
polygon.editor.events.add(["beforevertexdraw", "beforevertexadd"], function (event) {
    var mapGlobalPixelCenter = geoMap.getGlobalPixelCenter(),
        globalPixels = event.get("globalPixels"),
        pixelBounds = [
            [mapGlobalPixelCenter[0] - 100, mapGlobalPixelCenter[1] - 100],
            [mapGlobalPixelCenter[0] + 100, mapGlobalPixelCenter[1] + 100]
        ];
    event.callMethod("setGlobalPixels", [
        Math.max(Math.min(globalPixels[0], pixelBounds[1][0]), pixelBounds[0][0]),
        Math.max(Math.min(globalPixels[1], pixelBounds[1][1]), pixelBounds[0][1])
    ]);
});
```

beforevertexdrag

Событие, предшествующее событию vertexdrag. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- vertexModel - ссылка на модель перетаскиваемой вершины;
- globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

Имена методов, доступных через метод [Event.callMethod](#):

- setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "vertexdrag" будет отменено.

beforevertexdragstart

Событие, предшествующее событию vertexdragstart. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.
- vertexModel - ссылка на модель перетаскиваемой вершины;
- globalPixels - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие "vertexdragstart", будет отменено.

Пример:

```
// Запрещаем перетаскивание вершины с индексом 0 в контуре с тем же индексом.
polygon.editor.events.add("beforevertexdragstart", function (event) {
    var vertexModel = event.get("vertexModel"),
        pathModel = vertexModel.getParent();
    if (pathModel.getIndex() == 0 && vertexModel.getIndex() == 0) {
        event.preventDefault();
    }
});
```

```
});
```

beforevertexdraw

Событие, предшествующее событию vertexdraw. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- parentModel - ссылка на родительскую, для добавляемой вершины, модель данных;
- vertexIndex - индекс добавляемой вершины;
- globalPixels - координаты добавляемой вершины в глобальных пиксельных координатах.

Имена методов, доступных через метод [Event.callMethod](#):

- setGlobalPixels - метод позволяет скорректировать значения координат добавляемой вершины. В качестве аргумента принимает новые глобальные пиксельные координаты вершины в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "vertexdraw" будет отменено.

drawingstart

Включение режима добавления новых вершин. Экземпляр класса [Event](#).

drawingstop

Отключение режима добавления новых вершин. Экземпляр класса [Event](#).

edgedrag

Перетаскивание промежуточной метки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- edgeModel - ссылка на модель перетаскиваемой метки;
- globalPixels - координаты перетаскиваемой метки в глобальных пиксельных координатах.

edgedragend

Завершение перетаскивания промежуточной метки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- edgeModel - ссылка на модель перетаскиваемой метки;
- globalPixels - координаты перетаскиваемой метки в глобальных пиксельных координатах.

edgedragstart

Начало перетаскивания промежуточной метки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.
- edgeModel - ссылка на модель перетаскиваемой метки;
- globalPixels - координаты перетаскиваемой метки в глобальных пиксельных координатах.

editingstart

Включение режима редактирования вершин. Экземпляр класса [Event](#).

editingstop

Отключение режима редактирования вершин. Экземпляр класса [Event](#).

framingstart

Включение режима масштабирования. Экземпляр класса [Event](#).

framingstop

Отключение режима масштабирования. Экземпляр класса [Event](#).

vertexadd

Добавление новой вершины. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `parentModel` - ссылка на родительскую, для добавленной вершины, модель данных;
- `vertexIndex` - индекс добавленной вершины;
- `globalPixels` - координаты добавленной вершины в глобальных пиксельных координатах.

vertexdrag

Перетаскивание вершины. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset` - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- `vertexModel` - ссылка на модель перетаскиваемой вершины;
- `globalPixels` - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

vertexdragend

Завершение перетаскивания вершины. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `vertexModel` - ссылка на модель перетаскиваемой вершины;
- `globalPixels` - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

vertexdragstart

Начало перетаскивания вершины. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.
- `vertexModel` - ссылка на модель перетаскиваемой вершины;
- `globalPixels` - координаты перетаскиваемой вершины в глобальных пиксельных координатах.

vertexdraw

Рисование новой вершины. Данное событие, как правило, предшествует событию добавления вершины и возникает в момент движения мышью при включенном режиме добавления новых вершин. На основе данных, передаваемых в данном событии, отображаются направляющие линии в режиме добавления новых вершин. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `parentModel` - ссылка на родительскую, для добавляемой вершины, модель данных;
- `vertexIndex` - индекс добавляемой вершины;
- `globalPixels` - координаты добавляемой вершины в глобальных пиксельных координатах.

Описание методов**getModel**

```
{vow.Promise} getModel()
```

Возвращает объект-обещание, который подтверждается объектом модели в момент ее фактического создания, либо отклоняется с одним из следующих кодов ошибки:

- `Canceled` - режим редактирования отключен до момента фактического создания модели;

- Editor wasn't started - режим редактирования не включен.

getModelSync

```
{geometryEditor.model.RootPolygon|null} getModelSync()
```

Возвращает модель редактора или null, если на момент вызова она отсутствует.

getView

```
{vow.Promise} getView()
```

Возвращает объект-обещание, который подтверждается объектом отображения в момент его фактического создания, либо отклоняется с одним из следующих кодов ошибки:

- Canceled - режим редактирования отключен до момента фактического создания отображения;
- Editor wasn't started - режим редактирования не включен.

getViewSync

```
{geometryEditor.view.MultiPath|null} getViewSync()
```

Возвращает отображение редактора или null, если на момент вызова оно отсутствует.

startDrawing

```
{vow.Promise} startDrawing()
```

Включает режим добавления новых вершин в многоугольник. Включение происходит асинхронно.

Возвращает объект-обещание, который подтверждается при включении режима добавления новых вершин, либо отклоняется с одним из следующих кодов ошибки:

- Canceled - режим добавления новых вершин отключен до момента его фактического старта.

startEditing

```
{vow.Promise} startEditing()
```

Включает режим редактирования многоугольника (для добавления новых вершин). Включение происходит асинхронно.

Возвращает объект-обещание, который подтверждается в момент фактического старта режима редактирования, либо отклоняется с одним из следующих кодов ошибки:

- Canceled - режим редактирования отключен до момента его фактического старта.

startFraming

```
{vow.Promise} startFraming()
```

Включает режим масштабирования для многоугольника. Включение происходит асинхронно.

Возвращает Promise, который подтверждается в момент фактического старта режима масштабирования.

stopDrawing

```
{ } stopDrawing()
```

Отключает режим добавления новых вершин в многоугольник.

stopEditing

```
{ } stopEditing()
```

Отключает режим редактирования.

stopFraming

```
{ } stopFraming()
```

Отключает режим масштабирования.

geometryEditor.view**geometryEditor.view.Edge**

Примечание: Конструктор класса `geometryEditor.view.Edge` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Отображение промежуточной метки. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Методы](#)**Методы**

Имя	Возвращает	Описание
getPlacemark()	GeoObject	Возвращает ссылку на геообъект промежуточной метки. В данных метки, под именем "model", доступна ссылка на модель данных промежуточной метки.

Описание методов**getPlacemark**

```
{GeoObject} getPlacemark()
```

Возвращает ссылку на геообъект промежуточной метки. В данных метки, под именем "model", доступна ссылка на модель данных промежуточной метки.

geometryEditor.view.MultiPath

Примечание: Конструктор класса `geometryEditor.view.MultiPath` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Отображение множества контуров. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Методы](#)**Методы**

Имя	Возвращает	Описание
getEdgePlacemarks()	GeoObjectCollection	Возвращает коллекцию промежуточных меток дочерних отображений контуров.

Имя	Возвращает	Описание
<code>getPathViews()</code>	<code>geometryEditor.view.Path[]</code>	Возвращает массив отображений дочерних контуров.
<code>getVertexPlacemarks()</code>	<code>GeoObjectCollection</code>	Возвращает коллекцию меток вершин дочерних отображений контуров.

Описание методов

getEdgePlacemarks

```
{GeoObjectCollection} getEdgePlacemarks()
```

Возвращает коллекцию промежуточных меток дочерних отображений контуров.

getPathViews

```
{geometryEditor.view.Path[]} getPathViews()
```

Возвращает массив отображений дочерних контуров.

getVertexPlacemarks

```
{GeoObjectCollection} getVertexPlacemarks()
```

Возвращает коллекцию меток вершин дочерних отображений контуров.

geometryEditor.view.Path

Примечание: Конструктор класса `geometryEditor.view.Path` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Отображение контура. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

Методы

Методы

Имя	Возвращает	Описание
<code>getEdgePlacemarks()</code>	<code>GeoObjectCollection</code>	Возвращает коллекцию промежуточных меток, входящих в контур.
<code>getEdgeViews()</code>	<code>geometryEditor.view.Edge[]</code>	Возвращает массив отображений промежуточных меток.
<code>getVertexPlacemarks()</code>	<code>GeoObjectCollection</code>	Возвращает коллекцию меток вершин, входящих в контур.
<code>getVertexViews()</code>	<code>geometryEditor.view.Vertex[]</code>	Возвращает массив отображений дочерних вершин.

Описание методов

getEdgePlacemarks

```
{GeoObjectCollection} getEdgePlacemarks()
```

Возвращает коллекцию промежуточных меток, входящих в контур.

getEdgeViews

```
{geometryEditor.view.Edge[]} getEdgeViews()
```

Возвращает массив отображений промежуточных меток.

getVertexPlacemarks

```
{GeoObjectCollection} getVertexPlacemarks()
```

Возвращает коллекцию меток вершин, входящих в контур.

getVertexViews

```
{geometryEditor.view.Vertex[]} getVertexViews()
```

Возвращает массив отображений дочерних вершин.

geometryEditor.view.Vertex

Примечание: Конструктор класса `geometryEditor.view.Vertex` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Отображение вершины. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

Методы

Методы

Имя	Возвращает	Описание
<code>getPlacemark()</code>	<code>GeoObject</code>	Возвращает геообъект метки, обозначающей вершину. В данных метки, под именем "model", доступна ссылка на модель данных вершины.

Описание методов

getPlacemark

```
{GeoObject} getPlacemark()
```

Возвращает геообъект метки, обозначающей вершину. В данных метки, под именем "model", доступна ссылка на модель данных вершины.

GeoObject

Расширяет `IGeoObject`.

Геообъект. В зависимости от типа геометрии может отображаться как метка, линия, полигон и т.д. Вы также можете использовать вспомогательные классы для упрощенного создания геообъектов с определенным типом геометрии.

См.: [Placemark](#) [Polyline](#) [Polygon](#) [Circle](#) [Rectangle](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
GeoObject([feature[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
feature	—	Тип: Object Описание геообъекта.
feature.geometry	—	Тип: IGeometry IGeometryJson Геометрия геообъекта. Может задаваться непосредственно объектом, либо в виде объекта JSON-представления геометрии. Примеры классов и объектов, которые могут представлять геометрию - geometry.Point , geometry.json.Point .

Параметр	Значение по умолчанию	Описание
<code>feature.properties</code>	—	<p>Тип: <code>IDataManager Object</code></p> <p>Данные геообъекта. Могут задаваться как экземпляр класса, реализующего интерфейс <code>IDataManager</code>, либо в виде хэша. При значениях опций по умолчанию геообъектом трактуются следующие поля данных:</p> <ul style="list-style-type: none">• <code>iconContent</code> - содержимое иконки геообъекта;• <code>iconCaption</code> — подпись иконки геообъекта;• <code>hintContent</code> - содержимое всплывающей подсказки геообъекта;• <code>balloonContent</code> - содержимое балуна геообъекта;• <code>balloonContentHeader</code> - содержимое заголовка балуна геообъекта;• <code>balloonContentBody</code> - содержимое основной части балуна геообъекта;• <code>balloonContentFooter</code> - содержимое нижней части балуна геообъекта. <p>Поле <code>balloonContent</code> является кратким обозначением для поля <code>balloonContentBody</code>, но при одновременном задании <code>balloonContentBody</code> более приоритетен. Также вы можете дополнить данные геообъекта своими собственными полями и использовать их везде, где это возможно. Например в макете метки или макете балуна.</p>

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции геообъекта. Используя этот параметр можно задавать как опции самого геообъекта, так и опции его составных частей:</p> <ul style="list-style-type: none"> Опции балуна геообъекта с префиксом <code>balloon</code>. Опции всплывающей подсказки геообъекта с префиксом <code>hint</code>. Опции редактора геометрии геообъекта с префиксом <code>editor</code>. Тип редактора и перечень доступных опций зависит от типа геометрии геообъекта. Смотрите описание классов geometryEditor.LineString, geometryEditor.Polygon, geometryEditor.Point. Опции геометрии могут задаваться без префикса. Смотрите описание классов IGeometry геометрий geometry.Point, geometry.Polygon и др.
options.circleOverlay	"default#circle"	<p>Тип: String Function</p> <p>Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра:</p> <ul style="list-style-type: none"> <code>geometry</code>: IPixelCircleGeometry - непосредственно пиксельная геометрия; <code>data</code>: Object - данные оверлея; <code>options</code>: Object - опции оверлея. <p>Функция возвращает объект vow.Promise.</p>
options.cursor	"pointer"	<p>Тип: String</p> <p>Вид курсора над геообъектом.</p>

Параметр	Значение по умолчанию	Описание
<code>options.draggable</code>	false	Тип: Boolean Определяет возможность перетаскивания геообъекта.
<code>options.fill</code>	true	Тип: Boolean Наличие заливки фигуры.
<code>options.fillColor</code>	"0066ff99"	Тип: String Цвет заливки.
<code>options.fillImageHref</code>	—	Тип: String Фоновое изображение. При включении данной опции в режиме заливки stretch значение опции <code>fillColor</code> игнорируется.
<code>options.fillMethod</code>	'stretch'	Тип: String Тип заливки фоном. Может принимать одно из двух значений: <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - Фоновое изображение повторяется без изменений размера. Аналог background-repeat в css. Можно использовать для заливки фигуры неким шаблоном.
<code>options.fillOpacity</code>	1	Тип: Number Прозрачность заливки.
<code>options.hasBalloon</code>	true	Тип: Boolean Определяет наличие поля balloon у геообъекта.
<code>options.hasHint</code>	true	Тип: Boolean Определяет наличие поля hint у геообъекта.
<code>options.hideIconOnBalloonOpen</code>	true	Тип: Boolean Скрывать иконку при открытии балуна.

Параметр	Значение по умолчанию	Описание
<code>options.iconCaptionMaxWidth</code>	188	Тип: Number Максимальная ширина подписи метки.
<code>options.iconColor</code>	—	Тип: String Цвет иконки. Эта опция применяется для стандартных иконок в браузерах, поддерживающих SVG.
<code>options.iconContentLayout</code>	—	Тип: Function String Макет содержимого иконки. (Тип: конструктор объекта с интерфейсом <code>ILayout</code> или его ключ в хранилище).
<code>options.iconContentOffset</code>	—	Тип: Number[] Пиксельный сдвиг содержимого иконки относительно родительского элемента. Используется в макете <code>default#imageWithContent</code> .
<code>options.iconContentPadding</code>	—	Тип: Number[] Отступ для содержимого в иконке.
<code>options.iconContentSize</code>	—	Тип: Number[] Размер содержимого. Используется в макете <code>default#imageWithContent</code> .
<code>options.iconImageClipRect</code>	<code>[[0, 0], [{imageWidth}, {imageHeight}]]</code>	Тип: Number[] Прямоугольная область (указывается левый верхний и правый нижний углы), которая будет вырезана из исходного графического файла и смасштабирована под размеры иконки (например для использования спрайтов). По умолчанию исходная картинка подставляется полностью.
<code>options.iconImageHref</code>	—	Тип: String URL графического файла иконки. Применяется только в комбинации с макетами (<code>iconLayout</code>) <code>'default#image'</code> и <code>'default#imageWithContent'</code> .
<code>options.iconImageOffset</code>	—	Тип: Number[] Пиксельный сдвиг изображения иконки внутри родительского элемента.

Параметр	Значение по умолчанию	Описание
<code>options.iconImageShape</code>	—	<p>Тип: <code>IShape</code> <code>null</code></p> <p>Фигура активной области. Если не задана, то автоматически будет рассчитана прямоугольная фигура на основе размера и смещения картинки. Координаты геометрии фигуры отсчитываются от точки привязки. Применяется только в комбинации с макетами (<code>iconLayout</code>) <code>'default#image'</code> и <code>'default#imageWithContent'</code>.</p>
<code>options.iconImageSize</code>	—	<p>Тип: <code>Number</code>[]</p> <p>Размер иконки в пикселях.</p>
<code>options.iconLayout</code>	—	<p>Тип: <code>Function</code> <code>String</code></p> <p>Макет иконки. (Тип: конструктор объекта с интерфейсом <code>ILayout</code> или его ключ в хранилище).</p>
<code>options.iconMaxHeight</code>	—	<p>Тип: <code>Number</code></p> <p>Максимальная высота иконки с содержимым.</p>
<code>options.iconMaxWidth</code>	—	<p>Тип: <code>Number</code></p> <p>Максимальная ширина иконки с содержимым.</p>
<code>options.iconOffset</code>	<code>[0, 0]</code>	<p>Тип: <code>Number</code>[]</p> <p>Пиксельное смещение левого верхнего угла иконки относительно географической точки привязки метки.</p>
<code>options.iconShadow</code>	<code>false</code>	<p>Тип: <code>Boolean</code></p> <p>Флаг наличия тени у иконки.</p>
<code>options.iconShadowImageClipRect</code>	<code>[[0, 0], [{imageWidth}, {imageHeight}]]</code>	<p>Тип: <code>Number</code>[][]</p> <p>Прямоугольная область (указывается левый верхний и правый нижний углы), которая будет вырезана из исходного графического файла и смасштабирована под размеры тени (например для использования спрайтов). По умолчанию исходная картинка подставляется полностью.</p>

Параметр	Значение по умолчанию	Описание
options.iconShadowImageHref	—	<p>Тип: String</p> <p>URL графического файла тени иконки. Применяется только в комбинации с макетами (iconShadowLayout) 'default#image' и 'default#imageWithContent'.</p>
options.iconShadowImageOffset	—	<p>Тип: Number[]</p> <p>Пиксельный сдвиг изображения тени иконки внутри родительского элемента.</p>
options.iconShadowImageSize	—	<p>Тип: Number[]</p> <p>Размер тени иконки.</p>
options.iconShadowLayout	—	<p>Тип: Function String</p> <p>Макет тени иконки. (Тип: конструктор объекта с интерфейсом ILayout или его ключ в хранилище).</p>
options.iconShadowOffset	—	<p>Тип: Number[]</p> <p>Пиксельное смещение тени иконки относительно заданной позиции.</p>
options.interactiveZIndex	—	<p>Тип: Boolean</p> <p>Включает режим автоматического изменения z-index геообъекта в зависимости от его состояния. По умолчанию принимает значение true для геометрии geometry.Point и false для других геометрий.</p>
options.interactivityModel	"default#geoObject"	<p>Тип: String</p> <p>Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage.</p>

Параметр	Значение по умолчанию	Описание
options.lineStringOverlay	"default#polyline"	<p>Тип: String Function</p> <p>Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра:</p> <ul style="list-style-type: none"> • geometry: IPixelLineStringGeometry - непосредственно пиксельная геометрия; • data: IDataManager или Object - данные оверлея; • options: Object - опции оверлея. <p>Функция возвращает объект vow.Promise.</p>
options.opacity	1	<p>Тип: Number</p> <p>Прозрачность.</p>
options.openBalloonOnClick	true	<p>Тип: Boolean</p> <p>Определяет показывать ли балун при щелчке на геообъекте.</p>
options.openEmptyBalloon	false	<p>Тип: Boolean</p> <p>Определяет показывать ли пустой балун при щелчке на геообъекте.</p>
options.openEmptyHint	false	<p>Тип: Boolean</p> <p>Определяет показывать ли пустой хинт при наведении указателя мыши на геообъект.</p>
options.openHintOnHover	true	<p>Тип: Boolean</p> <p>Определяет показывать ли хинт при наведении указателя мыши на геообъект.</p>
options.outline	true	<p>Тип: Boolean</p> <p>Наличие обводки фигуры.</p>

Параметр	Значение по умолчанию	Описание
options.pane	—	<p>Тип: String</p> <p>Ключ пейна, в который помещается оверлей геообъекта. Значение по умолчанию определяется текущим оверлеем.</p>
options.pointOverlay	"default#placemark"	<p>Тип: String Function</p> <p>Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра:</p> <ul style="list-style-type: none"> geometry: IPixelPointGeometry — непосредственно пиксельная геометрия; data: IDataManager или Object - данные оверлея; options: Object - опции оверлея. <p>Функция возвращает объект vow.Promise.</p>
options.polygonOverlay	"default#polygon"	<p>Тип: String Function</p> <p>Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра:</p> <ul style="list-style-type: none"> geometry: IPixelPolygonGeometry - непосредственно пиксельная геометрия; data: IDataManager или Object - данные оверлея; options: Object - опции оверлея. <p>Функция возвращает объект vow.Promise.</p>
options.preset	—	<p>Тип: String</p> <p>Ключ предустановленных опций геообъекта. Список ключей содержится в описании option.presetStorage.</p>

Параметр	Значение по умолчанию	Описание
options.rectangleOverlay	"default#rectangle"	<p>Тип: String Function</p> <p>Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра:</p> <ul style="list-style-type: none"> geometry: IPixelCircleGeometry - непосредственно пиксельная геометрия; data: IDataManager или Object - данные оверлея; options: Object - опции оверлея. <p>Функция возвращает объект vow.Promise.</p>
options.setMapCursorInDragging	false	<p>Тип: Boolean</p> <p>true — при перетаскивании объекта выставлять курсор «сжатая рука» и на объект, и на карту, false — только на сам объект.</p>
options.strokeColor	"0066ffff"	<p>Тип: String String[]</p> <p>Цвет линии или обводки. Можно задать несколько значений для множественной обводки.</p>
options.strokeOpacity	1	<p>Тип: Number Number[]</p> <p>Прозрачность линии или обводки. Можно задать несколько значений для множественной обводки.</p>
options.strokeStyle	—	<p>Тип: String Object String[] Object[]</p> <p>Стиль линии или обводки. Можно задать несколько значений для множественной обводки. Доступные стили перечислены в объекте graphics.style.stroke.</p>
options.strokeWidth	1	<p>Тип: Number Number[]</p> <p>Толщина линии или обводки. Можно задать несколько значений для множественной обводки.</p>

Параметр	Значение по умолчанию	Описание
<code>options.syncOverlayInit</code>	false	Тип: Boolean Включает синхронное добавление оверлея на карту. По умолчанию добавление оверлея осуществляется асинхронно, что позволяет предотвратить зависания браузера при добавлении на карту большого числа геообъектов. Однако, асинхронное добавление не позволяет получать доступ к оверлею сразу после добавления геообъекта на карту.
<code>options.useMapMarginInDragging</code>	true	Тип: Boolean При перетаскивании объекта к краю карты происходит автоматическое изменение центра карты. Нужно ли учитывать отступы карты при автоматическом смещении центра карты map.margin.Manager .
<code>options.visible</code>	true	Тип: Boolean Определяет видимость геообъекта.
<code>options.zIndex</code>	—	Тип: Number z-index геообъекта в обычном состоянии. Наименее приоритетный.
<code>options.zIndexActive</code>	—	Тип: Number z-index геообъекта с открытым балуном. Наиболее приоритетный.
<code>options.zIndexDrag</code>	—	Тип: Number z-index геообъекта при перетаскивании.
<code>options.zIndexHover</code>	—	Тип: Number z-index геообъекта при наведении на него указателя мыши.

Примеры:**1.**

```
var myGeoObject = new ymaps.GeoObject({
  // Определение типа геометрии "Полилиния".
  geometry: {
    type: "LineString",
    coordinates: [
      [55.75, 37.61], [52.51, 13.38]
    ]
  },
  // Определение данных геообъекта.
  properties: {
    hintContent: "Москва-Берлин"
  }
});
```

```

    }, {
      // Включение режима отображения в виде геодезических кривых.
      geodesic: true,
      // Установка ширины до 5 пикселей.
      strokeWidth: 5,
      // Установка цвета линии.
      strokeColor: "#F008"
    });
    // Добавление геообъекта на карту.
    geoMap.geoObjects.add(myGeoObject);

```

2.

```

var myGeoObject = new ymaps.GeoObject({
  // Определение геометрии "Point".
  geometry: {
    type: "Point",
    coordinates: [55.75, 37.61]
  },
  // Определение данных геообъекта.
  properties: {
    hintContent: "Москва",
    balloonContentHeader: "Москва",
    balloonContentBody: "Столица России",
    population: 11848762
  }
}, {
  // Установка предустановки для метки с точкой и без содержимого.
  preset: "islands#redDotIcon",
  // Включение перетаскивания.
  draggable: true,
  // Переопределение макета для содержимого в нижней части воздушного шара.
  balloonContentFooterLayout: ymaps.templateLayoutFactory.createClass(
    'population: {{ properties.population }}', coordinates: {{ geometry.coordinates }}'
  ),
  // Отключение задержки для закрытия всплывающей подсказки.
  hintCloseTimeout: null
});
// Добавление геообъекта на карту.
geoMap.geoObjects.add(myGeoObject);

```

Поля

Имя	Тип	Описание
balloon	geoObject.Balloon	Балун геообъекта.
editor	IGeometryEditor	Редактор геометрии геообъекта.
events	event.Manager	Менеджер событий.
geometry	IGeometry null	Геометрия геообъекта.
hint	geoObject.Hint	Хинт геообъекта.
options	option.Manager	Менеджер опций геообъекта.
properties	data.Manager	Менеджер данных геообъекта.
state	data.Manager	Состояние геообъекта. Определяется следующими полями: <ul style="list-style-type: none"> <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается.

События

Имя	Описание
balloonclose	Заккрытие балуна. Экземпляр класса Event .
balloonopen	Открытие балуна на геообъекте. Экземпляр класса Event .
beforedrag	<p>Событие, предшествующее событию drag. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p>
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>originalEvent</code> - оригинальное событие редактора геометрии.
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>originalEvent</code>: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Заккрытие хинта. Экземпляр класса Event.</p>
hintopen	<p>Открытие хинта на геообъекте. Экземпляр класса Event.</p>

Имя	Описание
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> clientX - координата касания X относительно видимой области браузера; clientY - координата касания Y относительно видимой области браузера; pageX - координата касания X касания относительно начала документа; pageY - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

balloon

```
{geoObject.Balloon} balloon
```

Балун геообъекта.

editor

```
{IGeometryEditor} editor
```

Редактор геометрии геообъекта.

Пример:

```
// Начало редактирования геометрии геообъекта.
geoObject.editor.startEditing();
// ...
// Завершение редактирования.
```

```
geoObject.editor.stopEditing();
```

events

```
{event.Manager} events
```

Менеджер событий.

geometry

```
{IGeometry|null} geometry
```

Геометрия геообъекта.

Пример:

```
// При изменении координат геометрии геообъекта устанавливаем карте границы так,  
// чтобы охватить геообъект целиком.  
myMap.geoObjects.add(myGeoObject);  
myGeoObject.geometry.events.add("change", function () {  
    myMap.setBounds(myGeoObject.geometry.getBounds());  
});
```

hint

```
{geoObject.Hint} hint
```

Хинт геообъекта.

options

```
{option.Manager} options
```

Менеджер опций геообъекта.

properties

```
{data.Manager} properties
```

Менеджер данных геообъекта.

Пример:

```
// При изменении данных выводим пользовательский идентификатор  
// (если у объекта поле "synchronized" принимает значение false).  
myGeoObject.properties.events.add("change", function () {  
    if (!myGeoObject.properties.get("synchronized")) {  
        console.log(myGeoObject.properties.get("myID"));  
    }  
});
```

state

```
{data.Manager} state
```

Состояние геообъекта. Определяется следующими полями:

- active: Boolean - признак того, что на геообъекте открыт балун.
- hover: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши;
- drag: Boolean - признак того, что в данный момент геообъект перетаскивается.

Описание событий

balloonclose

Закрытие балуна. Экземпляр класса [Event](#).

balloonopen

Открытие балуна на геообъекте. Экземпляр класса [Event](#).

beforedrag

Событие, предшествующее событию drag. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- position - координаты относительно документа. Массива вида [pageX, pageY];
- pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

Имена методов, доступных через метод [Event.callMethod](#):

- setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "drag" будет отменено.

Пример:

```
// Разрешаем перетаскивание геообъекта только вдоль горизонтальной оси.
geoObject.events.add("beforedrag", function (event) {
    var originalOffset = event.get("pixelOffset");
    event.callMethod("setPixelOffset", [originalOffset[0], 0]);
});
```

beforedragstart

Событие, предшествующее событию dragstart. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- position - координаты относительно документа. Массива вида [pageX, pageY];
- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие "dragstart", будет отменено.

drag

Перетаскивание геообъекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- position - координаты относительно документа. Массива вида [pageX, pageY];
- pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге;
- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

dragend

Завершение перетаскивания геообъекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- position - координаты относительно документа. Массива вида [pageX, pageY];
- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

dragstart

Начало перетаскивания геообъекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- position - координаты относительно документа. Массива вида [pageX, pageY];
- domEvent - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

editorstatechange

Изменение состояния редактора геометрии геообъекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- originalEvent - оригинальное событие редактора геометрии.

hintclose

Закрытие хинта. Экземпляр класса [Event](#).

hintopen

Открытие хинта на геообъекте. Экземпляр класса [Event](#).

geoObject

geoObject.addon

geoObject.addon.balloon

Примечание: Конструктор класса `geoObject.addon.balloon` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Модуль, который предоставляет возможность использовать балун у геообъекта. Добавляет интерфейс [IBalloonOwner](#) геообъекту [GeoObject](#). При подключении `package.full` (стандартный набор модулей) доступен по умолчанию. Если [GeoObject](#) подключается отдельно, то необходимо явно указать этот модуль в загрузчике. Если [geoObject.addon.balloon](#) подключается отдельно после создания [GeoObject](#), то интерфейс [IBalloonOwner](#) не будет добавлен. Тогда для инициализации менеджера балуна нужно будет использовать метод `geoObject.addon.balloon#get`.

[Методы](#)**Методы**

Имя	Возвращает	Описание
get(geoObject)	IPopupManager	Возвращает менеджер балуна геообъекта.

Описание методов**get**

```
{IPopupManager} get(geoObject)
```

Возвращает менеджер балуна геообъекта.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geoObject *</code>	—	Тип: IGeoObject Геообъект

* Обязательный параметр/опция.

Пример:

```
ymaps.geoObject.addon.balloon.get(geoObject)
```

`geoObject.addon.editor`

Примечание: Конструктор класса `geoObject.addon.editor` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Модуль, который предоставляет возможность использовать редактор у геообъекта. Добавляет поле `GeoObject#editor` в [GeoObject](#). При подключении `package.full` (стандартный набор модулей) доступен по умолчанию. Если [GeoObject](#) подключается отдельно, то необходимо явно указать этот модуль в загрузчике. Если [geoObject.addon.editor](#) подключается отдельно после создания [GeoObject](#), то поле `GeoObject#editor` не будет добавлено. Тогда для редактора нужно будет использовать метод `geoObject.addon.editor#get`.

Методы

Методы

Имя	Возвращает	Описание
<code>create(geoObject)</code>		Функция, создающая у переданного геообъекта поле <code>editor</code> .
<code>get(geoObject)</code>	IGeometryEditor	Возвращает редактор геообъекта.

Описание методов

`create`

```
{ } create(geoObject)
```

Функция, создающая у переданного геообъекта поле `editor`.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geoObject *</code>	—	Тип: IGeoObject Геообъект.

* Обязательный параметр/опция.

`get`

```
{IGeometryEditor} get(geoObject)
```

Возвращает редактор геообъекта.

Параметры:

Параметр	Значение по умолчанию	Описание
geoObject *	—	Тип: IGeoObject Геообъект

* Обязательный параметр/опция.

Пример:

```
ymaps.geoObject.addon.editor.get(geoObject)
```

geoObject.addon.hint

Примечание: Конструктор класса `geoObject.addon.hint` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Модуль, который предоставляет возможность использовать хинт у геообъекта. Добавляет [IHintOwner](#) интерфейс геообъекту ([GeoObject](#)). При подключении `package.full` (стандартный набор модулей) доступен по умолчанию. Если [GeoObject](#) подключается отдельно, то необходимо явно указать этот модуль в загрузчике. Если [geoObject.addon.hint](#) подключается отдельно после создания [GeoObject](#), то интерфейс [IHintOwner](#) не будет добавлен. Тогда для инициализации менеджера балуна нужно будет использовать метод `geoObject.addon.hint#get`.

[Методы](#)**Методы**

Имя	Возвращает	Описание
get(geoObject)	IPopupManager	Возвращает менеджер хинта геообъекта.

Описание методов**get**

```
{IPopupManager} get(geoObject)
```

Возвращает менеджер хинта геообъекта.

Параметры:

Параметр	Значение по умолчанию	Описание
geoObject *	—	Тип: IGeoObject Геообъект

* Обязательный параметр/опция.

Пример:

```
ymaps.geoObject.addon.hint.get(geoObject)
```

geoObject.Balloon

Расширяет [IBalloonManager](#).

Менеджер балуна геообъекта. Позволяет управлять балуном геообъекта, открывать и скрывать его. Передает балуну данные в формате [IGeoObjectPopupData](#). Использует внутри себя менеджер балуна карты [map.Balloon](#). Геообъекты содержат экземпляр этого класса, доступный как `myGeoObject.balloon`. Не создавайте экземпляры этого класса без необходимости.

См.: [Balloon GeoObject.balloon](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geoObject.Balloon(geoObject)
```

Параметры:

Параметр	Значение по умолчанию	Описание
geoObject *	—	Тип: Object Геообъект.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
autopanbegin	Начало автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
autopanend	Окончание автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
beforeuserclose	Событие, предшествующее событию Balloon.userclose . Позволяет отменить пользовательское действие посредством вызова метода <code>preventDefault</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
close	Закрытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. Унаследовано от IPopupManager .

Имя	Описание
open	Открытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. Унаследовано от IPopupManager .
userclose	Закрытие балуна пользователем. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .

Методы

Имя	Возвращает	Описание
autoPan()	vow.Promise	Перемещает карту таким образом, чтобы балун оказался видимым. Унаследован от IBalloonManager .
close([force])	vow.Promise	Закрывает инфо-объект. Унаследован от IPopupManager .
destroy()		Отключает менеджер инфо-объекта. Унаследован от IPopupManager .
getData()	Object null	Возвращает данные инфо-объекта или null. Унаследован от IPopupManager .
getOptions()	IOptionManager null	Возвращает менеджер опций или null. Унаследован от IPopupManager .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopupManager .
getOverlaySync()	IOverlay null	Возвращает оверлей, если тот существует. Унаследован от IPopupManager .
getPosition()	Number[] null	Возвращает координаты инфо-объекта или 'null'. Унаследован от IPopupManager .

Имя	Возвращает	Описание
<code>isOpen()</code>	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopupManager .
<code>open([position[, data[, options]])</code>	vow.Promise	Открывает балун геообъекта.
<code>setData([data])</code>	vow.Promise	Устанавливает новые пользовательские данные.
<code>setOptions(options)</code>	vow.Promise	Задаёт инфо-объекту новые опции. Унаследован от IPopupManager .
<code>setPosition(position)</code>	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopupManager .

Описание методов

open

```
{vow.Promise} open([position[, data[, options]])
```

Открывает балун геообъекта.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
position	—	Тип: <code>Number[]</code> Координаты открытия. По умолчанию: точка на геообъекте, ближайшая к текущему центру карты. Проекцию координат можно указать в опциях, иначе используется проекция геообъекта.
data	—	Тип: <code>Object</code> Данные, добавляемые в поле <code>userData</code> объекта данных, передаваемых балуну.
options	—	Тип: <code>Object</code> Опции.

setData

```
{vow.Promise} setData([data])
```

Устанавливает новые пользовательские данные.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>data</code>	—	Тип: Object Данные, добавляемые в поле <code>userData</code> объекта данных, передаваемых балуну.

geoObject.Hint

Расширяет [IHintManager](#).

Менеджер хинта геообъекта. Позволяет управлять хинтом геообъекта, открывать и скрывать его. Передаёт хинту данные в формате [IGeoObjectPopupData](#). Использует внутри себя менеджер хинтов карты [map.Hint](#). Геообъекты содержат экземпляр этого класса, доступный как `myGeoObject.hint`. Не создавайте экземпляры этого класса без необходимости.

См.: [Hint GeoObject.hint](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geoObject.Hint(geoObject)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geoObject *</code>	—	Тип: Object Геообъект.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
<code>close</code>	Заккрытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"><code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. Унаследовано от IPopupManager .

Имя	Описание
open	<p>Открытие инфо-объекта. Названия полей доступны через метод Event.get:</p> <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. <p>Унаследовано от IPopupManager.</p>

Методы

Имя	Возвращает	Описание
close([force])	vow.Promise	<p>Закрывает инфо-объект.</p> <p>Унаследован от IPopupManager.</p>
destroy()		<p>Отключает менеджер инфо-объекта.</p> <p>Унаследован от IPopupManager.</p>
getData()	<code>Object null</code>	<p>Возвращает данные инфо-объекта или <code>null</code>.</p> <p>Унаследован от IPopupManager.</p>
getOptions()	IOptionManager <code>null</code>	<p>Возвращает менеджер опций или <code>null</code>.</p> <p>Унаследован от IPopupManager.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание вернуть оверлей.</p> <p>Унаследован от IPopupManager.</p>
getOverlaySync()	IOverlay <code>null</code>	<p>Возвращает оверлей, если тот существует.</p> <p>Унаследован от IPopupManager.</p>
getPosition()	<code>Number[]</code> <code>null</code>	<p>Возвращает координаты инфо-объекта или <code>'null'</code>.</p> <p>Унаследован от IPopupManager.</p>
isOpen()	<code>Boolean</code>	<p>Возвращает состояние инфо-объекта: открыт/закрыт.</p> <p>Унаследован от IPopupManager.</p>
open([position[, data[, options]]])	vow.Promise	Открывает хинт геообъекта.
setData([data])	vow.Promise	Устанавливает новые пользовательские данные.
setOptions(options)	vow.Promise	<p>Задаёт инфо-объекту новые опции.</p> <p>Унаследован от IPopupManager.</p>

Имя	Возвращает	Описание
<code>setPosition(position)</code>	<code>vow.Promise</code>	Задаёт инфо-объекту новую позицию. Унаследован от <code>IPopupManager</code> .

Описание методов

open

```
{vow.Promise} open([position[, data[, options]])
```

Открывает хинт геообъекта.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>position</code>	—	Тип: <code>Number[]</code> Координаты открытия. По умолчанию: геометрический центр тяжести геообъекта. Проекцию координат можно указать в опциях, иначе используется проекция геообъекта.
<code>data</code>	—	Тип: <code>Object</code> Данные, добавляемые в поле <code>userData</code> объекта данных, передаваемых хинту.
<code>options</code>	—	Тип: <code>Object</code> Опции.

setData

```
{vow.Promise} setData([data])
```

Устанавливает новые пользовательские данные.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>data</code>	—	Тип: Object Данные, добавляемые в поле <code>userData</code> объекта данных, передаваемых хинту.

geoObject.Sequence

Расширяет [IGeoObject](#), [IGeoObjectSequence](#).

Неизменяемая коллекция геообъектов. Позволяет группировать геообъекты для добавления на карту, задания опций и т.д. Коллекция также является геообъектом.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
geoObject.Sequence([feature[, options]])
```

Создает неизменяемую коллекцию геообъектов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>feature</code>	—	Тип: Object Описание коллекции. Функционально соответствует описанию геообъекта. См.: GeoObject
<code>feature.children</code>	—	Тип: IGeoObject [] Массив дочерних геообъектов.
<code>feature.geometry</code>	—	Тип: IGeometry Object Геометрия коллекции.
<code>feature.properties</code>	—	Тип: IDataManager Object Данные коллекции.
<code>options</code>	—	Тип: Object Опции коллекции. Здесь можно задавать все опции, описанные в классе GeoObject . Значения опций будут применяться как к самой коллекции, так и к ее дочерним объектам, если для них данные опции не определены.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
boundschange	Изменение координат географической области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса Event . Унаследовано от IGeoObjectSequence .
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. Унаследовано от IParentOnMap .

Имя	Описание
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
pixelboundschange	<p>Изменение пиксельных координат области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса Event.</p> <p>Унаследовано от IGeoObjectSequence.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
each(callback[, context])		<p>Для каждого дочернего геообъекта, вызывает функцию-обработчик.</p> <p>Унаследован от IGeoObjectSequence.</p>
get(index)	IGeoObject	<p>Возвращает дочерний геообъект с заданным индексом.</p> <p>Унаследован от IGeoObjectSequence.</p>
getBounds()	<code>Number[][]</code> null	<p>Возвращает географические координаты области, охватывающей коллекции и ее дочерние геообъекты.</p> <p>Унаследован от IGeoObjectSequence.</p>
getIterator()	IIterator	<p>Возвращает итератор по дочерним геообъектам коллекции.</p> <p>Унаследован от IGeoObjectSequence.</p>
getLength()	Integer	<p>Возвращает длину коллекции.</p> <p>Унаследован от IGeoObjectSequence.</p>
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.</p> <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	<p>Метод предоставляет синхронный доступ к оверлею.</p> <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	<p>Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.</p> <p>Унаследован от IChildOnMap.</p>

Имя	Возвращает	Описание
getPixelBounds()	Number[][] null	Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и ее дочерние геообъекты. Унаследован от IGeoObjectSequence .
indexOf(object)	Integer	Возвращает индекс дочернего геообъекта. Если геообъект в коллекции не найден, возвращается -1. Унаследован от IGeoObjectSequence .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

GeoObjectCollection

Расширяет [IGeoObject](#), [IGeoObjectCollection](#).

Коллекция геообъектов. Позволяет группировать геообъекты для добавления на карту, задания опций и т.д. Коллекция также является геообъектом.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
GeoObjectCollection([feature[, options]])
```

Создает коллекцию геообъектов.

Параметры:

Параметр	Значение по умолчанию	Описание
feature	—	Тип: Object Описание коллекции. Функционально соответствует описанию геообъекта. См.: GeoObject
feature.children	—	Тип: IGeoObject[] Массив дочерних геообъектов.

Параметр	Значение по умолчанию	Описание
<code>feature.geometry</code>	—	Тип: IGeometry Object Геометрия коллекции.
<code>feature.properties</code>	—	Тип: IDataManager Object Данные коллекции.
<code>options</code>	—	Тип: Object Опции коллекции. Здесь можно задавать все опции, описанные в классе GeoObject . Значения опций будут применяться как к самой коллекции, так и к ее дочерним объектам, если для них данные опции не определены.

Пример:

```
// Создание коллекции геообъектов и настройка параметров.
var myGeoObjects = new ymaps.GeoObjectCollection({}, {
  preset: "islands#redCircleIcon",
  strokeWidth: 4,
  geodesic: true
});

// Добавление меток и полилинии в коллекцию.
myGeoObjects.add(new ymaps.Placemark([13.38, 52.51]));
myGeoObjects.add(new ymaps.Placemark([30.30, 50.27]));
myGeoObjects.add(new ymaps.Polyline([[13.38, 52.51], [30.30, 50.27]]));

// Добавление коллекции на карту.
myMap.geoObjects.add(myGeoObjects);
// Установка центра и масштаба карты таким образом, чтобы вся коллекция была видна.
myMap.setBounds(myGeoObjects.getBounds());
```

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
<code>geometry</code>	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
<code>options</code>	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
<code>properties</code>	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
<code>state</code>	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
add	<p>Добавлен (вставлен) дочерний геообъект. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>index</code>: Integer - индекс добавленного геообъекта; • <code>child</code>: IGeoObject - ссылка на добавленный геообъект. <p>Унаследовано от IGeoObjectCollection.</p>
boundschange	<p>Изменение координат географической области, охватывающей коллекцию и все ее дочерние геообъекты. Экземпляр класса Event.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>originalEvent</code>: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldMap</code> - старая карта; • <code>newMap</code> - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>overlay</code>: IOverlay null - ссылка на оверлей; • <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> • <code>oldParent</code> - старый родитель; • <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
pixelboundschange	<p>Изменение пиксельных координат области, охватывающей коллекцию и все ее дочерние геообъекты. Экземпляр класса Event.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
remove	<p>Удален дочерний геообъект. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>index</code>: <code>Integer</code> — индекс удаленного геообъекта; • <code>child</code>: IGeoObject — ссылка на удаленный геообъект. <p>Унаследовано от IGeoObjectCollection.</p>
set	<p>В коллекцию записан новый дочерний геообъект. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>index</code>: <code>Integer</code> - индекс геообъекта; • <code>child</code>: IGeoObject - ссылка на новый геообъект; • <code>prevChild</code>: IGeoObject — ссылка на предыдущее значение для данного индекса. <p>Унаследовано от IGeoObjectCollection.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
add(child)	GeoObjectCollection	Добавляет геообъект в коллекцию.
each(callback[, context])		Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.
get(index)	IGeoObject	Возвращает дочерний геообъект с заданным индексом. Унаследован от IGeoObjectCollection .
getBounds()	<code>Number[][]</code> null	Возвращает географические координаты области, охватывающей коллекцию и все ее дочерние геообъекты.
getIterator()	Iterator	Возвращает итератор по коллекции.
getLength()	<code>Integer</code>	Возвращает количество геообъектов в коллекции.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getPixelBounds()	<code>Number[][]</code> null	Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и ее дочерние геообъекты.

Имя	Возвращает	Описание
indexOf(object)	Integer	Возвращает индекс дочернего геообъекта. Если геообъект в коллекции не найден, возвращается -1. Унаследован от IGeoObjectCollection .
remove(child)	GeoObjectCollection	Удаляет геообъект из коллекции.
removeAll()	GeoObjectCollection	Удаляет из коллекции все геообъекты.
set(index, child)	GeoObjectCollection	Записывает в коллекцию новый дочерний геообъект.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .
splice(index, number)	GeoObjectCollection	Удаляет геообъекты из коллекции. Если необходимо, вставляет на их место другие. Объекты, которые будут вставлены вместо удаляемых, передаются как дополнительные параметры (после параметра number).
toArray()	IGeoObject[]	Возвращает массив, содержащий все геообъекты коллекции на момент вызова.

Описание событий

boundschange

Изменение координат географической области, охватывающей коллекцию и все ее дочерние геообъекты. Экземпляр класса [Event](#).

pixelboundschange

Изменение пиксельных координат области, охватывающей коллекцию и все ее дочерние геообъекты. Экземпляр класса [Event](#).

Описание методов

add

```
{GeoObjectCollection} add(child)
```

Добавляет геообъект в коллекцию.

Возвращает ссылку на коллекцию.

Параметры:

Параметр	Значение по умолчанию	Описание
child *	—	Тип: IGeoObject Дочерний геообъект.

* Обязательный параметр/опция.

each

```
{ } each(callback[, context])
```

Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Параметры:

Параметр	Значение по умолчанию	Описание
callback *	—	Тип: Function Функция-обработчик.
context	—	Тип: Object Контекст исполнения функции.

* Обязательный параметр/опция.

getBounds

```
{Number[][]|null} getBounds()
```

Возвращает географические координаты области, охватывающей коллекцию и все ее дочерние геообъекты.

Пример:

```
// Устанавливаем центр и масштаб карты так, чтобы отобразить всю коллекцию целиком.  
myMap.setBounds(myCollection.getBounds());
```

getIterator

```
{IIterator} getIterator()
```

Возвращает итератор по коллекции.

Пример:

```
// Найдем в коллекции геообъект с геометрией "Ломаная линия".
```

```
var iterator = myGroup.getIterator(),
    object;
while ((object = iterator.getNext()) != iterator.STOP_ITERATION) {
    if (object.geometry.getType() == "LineString") {
        break;
    }
}
```

getLength

```
{Integer} getLength()
```

Возвращает количество геообъектов в коллекции.

getPixelBounds

```
{Number[][]|null} getPixelBounds()
```

Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и ее дочерние геообъекты.

remove

```
{GeoObjectCollection} remove(child)
```

Удаляет геообъект из коллекции.

Возвращает ссылку на коллекцию.

Параметры:

Параметр	Значение по умолчанию	Описание
child *	—	Тип: IGeoObject Дочерний геообъект.

* Обязательный параметр/опция.

removeAll

```
{GeoObjectCollection} removeAll()
```

Удаляет из коллекции все геообъекты.

Возвращает ссылку на коллекцию.

set

```
{GeoObjectCollection} set(index, child)
```

Записывает в коллекцию новый дочерний геообъект.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
index *	—	Тип: Integer Индекс.

Параметр	Значение по умолчанию	Описание
<code>child *</code>	—	Тип: <code>IGeoObject</code> Дочерний геообъект.

* Обязательный параметр/опция.

splice

```
{GeoObjectCollection} splice(index, number)
```

Удаляет геообъекты из коллекции. Если необходимо, вставляет на их место другие. Объекты, которые будут вставлены вместо удаляемых, передаются как дополнительные параметры (после параметра `number`).

Возвращает коллекцию удаленных геообъектов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index *</code>	—	Тип: <code>Integer</code> Индекс геообъекта, с которого начинается удаление.
<code>number *</code>	—	Тип: <code>Integer</code> Количество удаляемых геообъектов.

* Обязательный параметр/опция.

Пример:

```
// Удаляет второй объект.  
myGeoObjects.splice(1, 1);  
// Вставляет новый объект obj во вторую позицию.  
myGeoObjects.splice(1, 0, obj);  
// Заменяет второй объект на новый объект obj.  
myGeoObjects.splice(1, 1, obj);
```

toArray

```
{IGeoObject[]} toArray()
```

Возвращает массив, содержащий все геообъекты коллекции на момент вызова.

geoQuery

Статическая функция.

Формирует набор данных из указанного источника и создает на его основе экземпляр `GeoQueryResult`.

Возвращает результат, содержащий данные из источника.

```
{ GeoQueryResult } geoQuery(source)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>source</code> *	—	<p>Тип: Object</p> <p>Источник геообъектов:</p> <ul style="list-style-type: none"> • <code>IGeoObject</code> - объект, реализующий соответствующий интерфейс; • <code>IGeoObject[]</code> - массив объектов, реализующих соответствующий интерфейс; • <code>ICollection</code> - коллекция объектов, реализующих интерфейс <code>IGeoObject</code>; • <code>ICollection[]</code> - массив коллекций объектов, реализующих интерфейс <code>IGeoObject</code>; • <code>vow.Promise</code> — объект-промис, передающий в функцию-обработчик источник данных для <code>geoQuery</code>; . Также в функцию-обработчик может быть передан объект с полем <code>geoObjects</code>, содержащий источник данных для <code>geoQuery</code>; • <code>GeoQueryResult</code> - объект класса <code>GeoQueryResult</code>. • <code>String Object</code> - строка или объект с JSON-описанием объектов; <p>JSON-описание объектов строится по следующему принципу (см. пример ниже). Объектом может быть сущность или коллекция сущностей. Коллекция сущностей представляет из себя объект с полями:</p> <ul style="list-style-type: none"> • <code>type</code> — тип объекта. Значение поля должно быть равно <code>"FeatureCollection"</code>; • <code>features</code> - массив дочерних сущностей коллекции. Дочерние объекты могут быть сущностями или вложенными коллекциями сущностей. <p>Сущность представляет собой объект с полями:</p> <ul style="list-style-type: none"> • <code>type</code> — тип объекта. Значение поля должно быть равно <code>"Feature"</code>; • <code>geometry</code> — геометрия объекта. Содержит поля <code>type</code> и <code>coordinates</code>. Соответствует параметру, передаваемому в конструктор объекта <code>ymaps.GeoObject</code>; • <code>options</code> - опции геообъекта; • <code>properties</code> - данные геообъекта.

* Обязательный параметр/опция.

Примеры:

1.

```
// Создание GeoQueryResult из одиночного геообъекта.  
var placemark = new ymaps.Placemark([34, 56]);  
ymaps.geoQuery(placemark).addToMap(myMap);
```

2.

```
// Создание GeoQueryResult из массива геообъектов.  
var objects = [  
    new ymaps.Placemark([34, 56]),  
    new ymaps.Rectangle([[34, 56], [36, 57]])  
];  
ymaps.geoQuery(objects).addToMap(myMap);
```

3.

```
// Создание GeoQueryResult из коллекции геообъектов.  
var result = ymaps.geoQuery(myMap.geoObjects).searchIntersect(myMap);  
alert("Количество геообъектов в видимой области карты: " + result.getLength());
```

4.

```
// Создание GeoQueryResult из vow.Deferred.  
var result = ymaps.geoQuery(ymaps.geocode('Сыромятнический переулк')).searchInside(myGeoBounds);  
// Поскольку источник данных асинхронный, нужно дождаться обработки результата.  
result.then(function () {  
    alert('Количество объектов, лежащих внутри заданной области: ' + result.getLength());  
});
```

5.

```
// Создание GeoQueryResult из JSON.  
var result = ymaps.geoQuery({  
    type: 'FeatureCollection',  
    features: [  
        {  
            type: 'Feature',  
            geometry: {  
                type: 'Circle',  
                coordinates: [15, 15],  
                radius: 100  
            }  
        },  
        {  
            type: 'Feature',  
            geometry: {  
                type: 'LineString',  
                coordinates: [[15, 16], [66, 23]]  
            }  
        },  
        {  
            type: 'FeatureCollection',  
            features: [  
                {  
                    type: 'Feature',  
                    geometry: {  
                        type: 'Point',  
                        coordinates: [12, 41]  
                    },  
                    properties: {  
                        name: 'point'  
                    },  
                    options: {  
                        preset: 'islands#yellowIcon'  
                    }  
                }  
            ]  
        }  
    ]  
});  
// Добавление не точечных объектов на карту как есть.  
result.search('geometry.type != "Point").addToMap(myMap);  
// Точечные объекты добавим на карту через кластеризатор.  
myMap.geoObjects.add(result.search('geometry.type == "Point").clusterize());
```

GeoQueryResult

Расширяет [IPromiseProvider](#).

Результат выборки.

[Конструктор](#) | [Методы](#)

Конструктор

```
GeoQueryResult(source)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>source</code> *	—	<p>Тип: Object</p> <p>Источник геообъектов:</p> <ul style="list-style-type: none"> IGeoObject - объект, реализующий соответствующий интерфейс; IGeoObject[] - массив объектов, реализующих соответствующий интерфейс; ICollection - коллекция объектов, реализующих интерфейс IGeoObject; ICollection[] - массив коллекций объектов, реализующих интерфейс IGeoObject; IPromiseProvider — объект-промис, передающий в функцию-обработчик источник данных для geoQuery. Также в функцию-обработчик может быть передан объект с полем geoObjects, содержащий источник данных для geoQuery; GeoQueryResult - объект класса GeoQueryResult. String Object - строка или объект с JSON-описанием объектов; Object[] - массив с JSON-описаниями геометрий. Элемент массива представляет из себя объект с полями type и coordinates. При описании круга также необходимо поле radius. При описании многоугольника можно указывать необязательное поле fillRule. <p>JSON-описание объектов строится по следующему принципу (см. пример ниже). Объектом может быть сущность или коллекция сущностей. Коллекция сущностей представляет из себя объект с полями:</p> <ul style="list-style-type: none"> type — тип объекта. Значение поля должно быть равно "FeatureCollection"; features - массив дочерних сущностей коллекции. Дочерние объекты могут быть сущностями или вложенными коллекциями сущностей. <p>Сущность представляет собой объект с полями:</p> <ul style="list-style-type: none"> type — тип объекта. Значение поля должно быть равно "Feature"; geometry — геометрия объекта. Содержит поля type и coordinates. Соответствует параметру, передаваемому в конструктор объекта ymaps.GeoObject; options - опции геообъекта; properties - данные геообъекта.

* Обязательный параметр/опция.

Примеры:

1.

```
// Создание GeoQueryResult из одиночного геообъекта.
var placemark = new ymaps.Placemark([34, 56]);
ymaps.geoQuery(placemark).addToMap(ymap);
```

2.

```
// Создание GeoQueryResult из массива геообъектов.
var objects = [
  new ymaps.Placemark([34, 56]),
  new ymaps.Rectangle([[34, 56], [36, 57]])
];
ymaps.geoQuery(objects).addToMap(ymap);
```

3.

```
// Создание GeoQueryResult из коллекции геообъектов.
var result = ymaps.geoQuery(ymap.geoObjects).searchIntersect(ymap);
alert("Количество геообъектов в видимой области карты: " + result.getLength());
```

4.

```
// Создание GeoQueryResult из vow.Deferred.
var result = ymaps.geoQuery(ymaps.geocode('Сыромятинский переулок')).searchInside(myGeoBounds);
// Поскольку источник данных асинхронный, нужно дождаться обработки результата.
result.then(function () {
  alert('Количество объектов, лежащих внутри заданной области: ' + result.getLength());
});
```

5.

```
// Создание GeoQueryResult из JSON.
var result = ymaps.geoQuery({
  type: 'FeatureCollection',
  features: [
    {
      type: 'Feature',
      geometry: {
        type: 'Circle',
        coordinates: [15, 15],
        radius: 100
      }
    },
    {
      type: 'Feature',
      geometry: {
        type: 'LineString',
        coordinates: [[15, 16], [66, 23]]
      }
    }
  ],
  // Коллекции объектов могут быть вложенными.
  {
    type: 'FeatureCollection',
    features: [
      {
        type: 'Feature',
        geometry: {
          type: 'Point',
          coordinates: [12, 41]
        },
        properties: {
          name: 'point'
        },
        options: {
          preset: 'islands#yellowIcon'
        }
      }
    ]
  }
]
});
// Добавление не точечных объектов на карту как есть.
result.search('geometry.type != "Point").addToMap(ymap);
// Точечные объекты добавим на карту через кластеризатор.
ymap.geoObjects.add(result.search('geometry.type == "Point").clusterize());
```

Методы

Имя	Возвращает	Описание
add(source)	GeoQueryResult	Добавляет объекты из источника к объектам результата. Не изменяет исходный объект, а создает новый, содержащий объединенный набор геообъектов.
addEvents(events, callback, context)	GeoQueryResult	Навешивает обработчики событий на элементы выборки.
addTo(collection)	GeoQueryResult	Метод для добавления объектов в коллекцию геообъектов.
addToMap(map)	GeoQueryResult	Метод для добавления объектов на карту.
applyBoundsToMap(map[, options])	GeoQueryResult	Метод, позволяющий установить видимую область карты так, чтобы видны были все объекты из выборки.
clusterize([options])	Clusterer	Метод создает кластеризатор и добавляет в него объекты из выборки. В случае, если данные выборки еще не готовы, они будут добавлены в кластеризатор сразу после обработки, а возвращенный кластеризатор изначально будет пуст. В кластеризатор будут добавлены только объекты с геометрией "Point".
each(callback, context)	GeoQueryResult	Возвращает ссылку на себя.
get(index)	IGeoObject	Возвращает элемент выборки по индексу.
getBounds()	Number[][] null	Возвращает географические координаты области, охватывающей объекты результата.
getCenter(map)	Number[]	Метод, возвращающий центр области, охватывающей объекты результата, в географических координатах.
getCentralObject(map)	IGeoObject null	Метод для получения объекта, наиболее близкого к центру видимой области карты.

Имя	Возвращает	Описание
getClosestTo(object)	IGeoObject null	Метод, возвращающий объект выборки, ближайший к указанному. Если на вход подается объект, уже находящийся в выборке, то вернется другой объект выборки, ближайший к указанному. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.
getExtreme(key)	Number	Возвращает максимальные и минимальные значения координат среди координат объектов выборки.
getExtremeObject(key)	IGeoObject	Метод, возвращающий объект с минимальной или максимальной координатой среди координат объектов выборки.
getGlobalPixelBounds(map)	Number[][] null	Метод, возвращающий глобальные пиксельные координаты области для текущего зума карты.
getGlobalPixelCenter(map)	Number[]	Возвращает центр области, охватывающей объекты результата, в глобальных пиксельных координатах для текущего зума карты.
getIterator()	IIterator	Возвращает итератор по объектам результата.
getLength()	Number	Возвращает количество элементов в результате.
getMaxZoom(map[, options])	Number	Метод, вычисляющий максимальный коэффициент масштабирования, при котором в видимую область карты попадают все объекты.
getParent()	GeoQueryResult null	Возвращает ссылку на родительскую выборку, если текущая выборка была создана в результате изменения другого объекта GeoQueryResult .
indexOf(item)	Number	Возвращает индекс элемента в выборке. Если элемент не найден, возвращается -1.

Имя	Возвращает	Описание
<code>intersect(result)</code>	GeoQueryResult	Метод создает новую выборку, содержащую общие элементы для двух других выборок.
<code>isReady()</code>	Boolean	Возвращает признак, готовы результаты выборки или еще обрабатываются.
<code>map(callback[, context])</code>	GeoQueryResult	Метод, вызывающий метод <code>callback</code> для всех элементов выборки и формирующий новую выборку на основе полученных результатов.
<code>remove(objects)</code>	GeoQueryResult	Удаляет объекты из результата. Не изменяет исходный объект, а создает новый, содержащий результирующий набор геообъектов.
<code>removeEvents(events, callback, context)</code>		Удаляет подписку на событие с объекта. Обратите внимание, что для корректной отписки передаваемые аргументы должны быть точно такие же, как при подписке через метод <code>addEvents</code> .
<code>removeFrom(collection)</code>	GeoQueryResult	Метод для удаления объектов из коллекции.
<code>removeFromMap(map)</code>	GeoQueryResult	Метод для удаления объектов с карты.
<code>reverse()</code>	GeoQueryResult	Переставляет элементы выборки в обратном порядке и возвращает новую выборку.
<code>search(condition)</code>	GeoQueryResult	Метод для поиска объектов выбоки, удовлетворяющих условиям.
<code>searchContaining(object)</code>	GeoQueryResult	Метод, создающий новую выборку из объектов, содержащих указанный объект. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.

Имя	Возвращает	Описание
searchInside(object)	GeoQueryResult	Метод, создающий новую выборку из объектов, целиком входящих в указанный объект. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.
searchIntersect(object[, options])	GeoQueryResult	Метод, создающий новую выборку из объектов, пересекающих указанный объект. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.
setOptions(key[, value])	GeoQueryResult	Метод, позволяющий задать значение опций всем элементам выборки.
setProperties(path, value)	GeoQueryResult	Метод, позволяющий задать значение поля <code>properties</code> всем элементам выборки.
slice(begin[, end])	GeoQueryResult	Метод, возвращающий срез выборки.
sort(comparator)		Метод для сортировки объектов выборки. Не изменяет исходную выборку, а создает новую, содержащую упорядоченные объекты.
sortByDistance(object)	GeoQueryResult	Метод, позволяющий получить выборку, содержащую объекты, отсортированные по расстоянию до указанного объекта. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов. Не изменяет исходную выборку.
then([onFulfill, onReject, context])	GeoQueryResult	Подписка на "обещание".
unsetOptions(keys)	GeoQueryResult	Метод, позволяющий обнулять значение опций всем элементам выборки.
unsetProperties(path)	GeoQueryResult	Метод позволяющий обнулить значение поля <code>properties</code> у всех элементов выборки.

Описание методов

add

```
{GeoQueryResult} add(source)
```

Добавляет объекты из источника к объектам результата. Не изменяет исходный объект, а создает новый, содержащий объединенный набор геообъектов.

Возвращает новый объект с объединенным набором геообъектов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>source</code> *	—	<p>Тип: Object</p> <p>Источник геообъектов:</p> <ul style="list-style-type: none"> IGeoObject - объект, реализующий соответствующий интерфейс; IGeoObject[] - массив объектов, реализующих соответствующий интерфейс; ICollection - коллекция объектов, реализующих интерфейс IGeoObject; ICollection[] - массив коллекций объектов, реализующих интерфейс IGeoObject; vow.Promise — объект-промис, передающий в функцию-обработчик источник данных для geoQuery;. Также в функцию-обработчик может быть передан объект с полем geoObjects, содержащий источник данных для geoQuery; GeoQueryResult - объект класса GeoQueryResult. String Object - строка или объект с JSON-описанием объектов; Object[] - массив с JSON-описаниями геометрий. Элемент массива представляет из себя объект с полями type и coordinates. При описании круга также необходимо поле radius. При описании многоугольника можно указывать необязательное поле fillRule. <p>JSON-описание объектов строится по следующему принципу (см. пример ниже). Объектом может быть сущность или коллекция сущностей. Коллекция сущностей представляет из себя объект с полями:</p> <ul style="list-style-type: none"> type — тип объекта. Значение поля должно быть равно "FeatureCollection"; features - массив дочерних сущностей коллекции. Дочерние

* Обязательный параметр/опция.

Примеры:

1.

```
// Добавление в GeoQueryResult одиночного геообъекта.
var placemark = new ymaps.Placemark([34, 56]);
myGeoQueryResult.add(placemark);
```

2.

```
// Добавление в GeoQueryResult массива геообъектов.
var objects = [
    new ymaps.Placemark([34, 56]),
    new ymaps.Rectangle([[34, 56], [36, 57]])
];
// Обратите внимание, что в результате будет получен другой объект GeoQueryResult,
// а старый останется без изменений.
var newResult = myGeoQueryResult.add(objects);
```

3.

```
// Добавление в GeoQueryResult коллекции геообъектов.
// Обратите внимание, что в результате будет получен другой объект GeoQueryResult,
// а старый останется без изменений.
var newResult = myGeoQueryResult.add(myMap.geoObjects).searchIntersect(myBounds);
alert("Количество геообъектов в указанной области: " + myGeoQueryResult.getLength());
```

4.

```
// Добавление в GeoQueryResult результатов геокодирования.
// Обратите внимание, что в результате будет получен другой объект GeoQueryResult,
// а старый останется без изменений.
var newResult = myGeoQueryResult.add(ymaps.geocode('Сыромятнический переулок'));
newResult.searchInside(myGeoBounds);
// Поскольку источник данных асинхронный, нужно дождаться обработки результата.
result.then(function () {
    // Исходный объект все еще существует.
    alert('Было объектов до добавления: ' + myGeoQueryResult.getLength());
    // В новом объекте уже и старые геообъекты, и добавленные.
    alert('Стало объектов после добавления: ' + result.getLength());
});
```

5.

```
// Добавление объектов из JSON-строки.
var result = ymaps.geoQuery(myMap.geoObjects),
var extendedResult = result.add('{"type": "Feature", "geometry": { "type": "Point", "coordinates": [15, 64] }}');
extendedResult.addToMap(myMap);
```

addEvents

```
{GeoQueryResult} addEvents(events, callback, context)
```

Навешивает обработчики событий на элементы выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>events</code> *	—	Тип: String String[] Тип или массив типов событий, на которые совершается подписка.
<code>callback</code> *	—	Тип: Function Функция-обработчик.

Параметр	Значение по умолчанию	Описание
<code>context *</code>	—	Тип: <code>Object</code> Контекст выполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

```
ymaps.geoQuery(map.geoObjects).search('geometry.type="Circle"').addEvents('click', function () {  
    alert('Вы кликнули по кругу!');  
});});
```

addTo

```
{GeoQueryResult} addTo(collection)
```

Метод для добавления объектов в коллекцию геообъектов.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>collection *</code>	—	Тип: <code>ICollection</code> Коллекция, в которую добавляются объекты.

* Обязательный параметр/опция.

Пример:

```
// Покажем на карте объекты северного полушария.  
var result1 = ymaps.geoQuery(placemarks).search('lat > 0').addTo(myMap.geoObjects);
```

addToMap

```
{GeoQueryResult} addToMap(map)
```

Метод для добавления объектов на карту.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map *</code>	—	Тип: <code>Map</code> Карта, на которую добавляются объекты.

* Обязательный параметр/опция.

Пример:

```
// Покажем на карте объекты северного полушария.  
var result1 = ymaps.geoQuery(placemarks).search('lat > 0').addToMap(myMap);
```

applyBoundsToMap

```
{GeoQueryResult} applyBoundsToMap(map[, options])
```

Метод, позволяющий установить видимую область карты так, чтобы видны были все объекты из выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: Map Карта.
<code>options</code>	—	Тип: Object Опции.
<code>options.checkZoomRange</code>	<code>false</code>	Тип: Boolean Проверка возможности установить указанный коэффициент масштабирования. Если значение опции равно <code>true</code> , метод вызывается асинхронно. Производится обращение к серверу, который возвращает интервал допустимых для данного центра значений коэффициентов масштабирования. После этого выставляется указанный центр и корректный зум.
<code>options.duration</code>	<code>0</code>	Тип: Number Время анимации в миллисекундах.
<code>options.preciseZoom</code>	<code>false</code>	Тип: Boolean Возможность использования дробных коэффициентов масштабирования.
<code>options.timingFunction</code>	<code>'linear'</code>	Тип: String Тайминг-функция. Аналогично значению CSS свойства <code>transition-timing-function</code> . Полный список значений: http://www.w3.org/TR/css3-transitions/#transition-timing-function_tag

Параметр	Значение по умолчанию	Описание
<code>options.useMapMargin</code>	<code>true</code>	Тип: Boolean Нужно ли учитывать отступы карты map.margin.Manager .
<code>options.zoomMargin</code>	<code>0</code>	Тип: Number Number[] Отступы от границ видимой области карты. Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы top, right, bottom, left. При включенной опции "useMapMargin" значение "zoomMargin" складывается со значениями, которые были рассчитаны в менеджере отступов map.margin.Manager .

* Обязательный параметр/опция.

Примеры:

1.

```
// Сделаем запрос на геокодирование и выставим карте видимую область так,
// чтобы на ней были видны сразу все результаты.
var result = ymaps.geoQuery(ymaps.geocode('улица Ленина')).applyBoundsToMap(ymap);
// Обратите внимание, что запрос на сервер асинхронный, нужно дождаться результата.
result.then(function () {
    alert("Результаты получены и видимая область карты скорректирована.");
}, function () {
    alert("Произошла ошибка.");});
```

2.

```
// Для синхронных запросов выставление новой области карты происходит сразу.
var result = ymaps.geoQuery(objects).applyBoundsToMap(ymap);
alert('Видимая область карты изменена.');;
```

clusterize

```
{Clusterer} clusterize([options])
```

Метод создает кластеризатор и добавляет в него объекты из выборки. В случае, если данные выборки еще не готовы, они будут добавлены в кластеризатор сразу после обработки, а возвращенный кластеризатор изначально будет пуст. В кластеризатор будут добавлены только объекты с геометрией "Point".

Возвращает кластеризатор.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции, передаваемые в конструктор объекта <code>Clusterer</code> .

Пример:

```
// Выберем только кафе и добавим их в кластеризатор.
var clusterer = ymaps.geoQuery(objects).search('properties.type="Cafe").clusterize();
myMap.geoObjects.add(clusterer);
```

each

```
{GeoQueryResult} each(callback, context)
```

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>callback *</code>	—	Тип: Function Функция-обработчик. Вызывается для каждого элемента выборки и получает его на вход.
<code>context *</code>	—	Тип: Object Контекст выполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

```
// Скроем красные метки в видимой области карты.
ymaps.geoQuery(placemarks).searchIntersect(myMap).each(function(pm) {
  if (pm.options.get('preset') == 'islands#redIcon') {
    myMap.geoObjects.remove(pm);
  }
});
```

get

```
{IGeoObject} get(index)
```

Возвращает элемент выборки по индексу.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index *</code>	—	Тип: Number Индекс элементов.

* Обязательный параметр/опция.

Примеры:**1.**

```
// Пример при синхронной обработке.
var result = ymaps.geoQuery(placemark).sort('lat'),
// Самый южный объект.
southObject = result.get(0),
// Самый северный объект.
northObject = result.get(result.getLength() - 1);
```

2.

```
// Пример при асинхронной обработке.
var result = ymaps.geoQuery(ymaps.geocode('кафе Москва')).sort('lat');
// Нужно дождаться готовности результата для дальнейшей обработки.
result.then(function () {
    // Самый южный объект.
    var southObject = result.get(0),
    // Самый северный объект.
    var northObject = result.get(result.getLength() - 1);
});
```

getBounds

```
{Number[][]|null} getBounds()
```

Возвращает географические координаты области, охватывающей объекты результата.

Примеры:**1.**

```
// Пример синхронного добавления объектов.
// Устанавливаем центр и масштаб карты так, чтобы были видны все объекты, добавленные на карту.
var myResult = ymaps.geoQuery(myMap.geoObjects);
myMap.setBounds(myResult.getBounds(), { checkZoomRange: true });
```

2.

```
// Пример асинхронного добавления объектов.
// Нужно дождаться готовности данных и только потом вызывать метод getBounds.
var myResult = ymaps.geoQuery(ymaps.geocode('р. Звероножка')).then(function () {
    myMap.setBounds(myResult.getBounds(), { checkZoomRange: true });
});
```

getCenter

```
{Number[]} getCenter(map)
```

Метод, возвращающий центр области, охватывающей объекты результата, в географических координатах.

Возвращает координаты центра области в географических координатах.

Параметры:

Параметр	Значение по умолчанию	Описание
<i>map</i> *	—	Тип: Map Карта, для которой нужно произвести расчеты.

* Обязательный параметр/опция.

Пример:

```
// Сместим центр карты в центр области, охватывающей объекты.
myMap.setCenter(ymaps.geoQuery(objects).getCenter());
```


getCentralObject

```
{IGeoObject|null} getCentralObject(map)
```

Метод для получения объекта, наиболее близкого к центру видимой области карты.

Возвращает ссылку на геообъект или null в случае пустой выборки.

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Карта.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример работы при синхронных операциях.  
// Откроем балун у объекта, ближайшего к центру видимой области карты.  
ymaps.geoQuery(objects).getCentralObject(myMap).balloon.open();
```

2.

```
// Пример использования асинхронных операций.  
var result = ymaps.geoQuery(ymaps.geocode('Улица Строителей'));  
// Отправили запрос на геокодирование,  
// теперь нужно дождаться результата.  
result.then(function () {  
    result.getCentralObject(myMap).balloon.open();  
});
```

getClosestTo

```
{IGeoObject|null} getClosestTo(object)
```

Метод, возвращающий объект выборки, ближайший к указанному. Если на вход подается объект, уже находящийся в выборке, то вернется другой объект выборки, ближайший к указанному. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.

Возвращает объект выборки, ближайший к указанному или null, если объект невозможно найти.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>object *</code>	—	<p>Тип: Object</p> <p>Объект, относительно которого будет производится поиск. Может принимать следующие значения:</p> <ul style="list-style-type: none"> IGeoObject - объект, реализующий интерфейс <code>IGeoObject</code>; IGeometry - объект, реализующий интерфейс <code>IGeometry</code>; Map - карта. В этом случае опорным объектом будет прямоугольная граница карты; Number[] - координаты точки; Object - Json-описание геометрии. Содержит поля <code>type</code> и <code>coordinates</code>. При описании круга также необходимо поле <code>radius</code>. При описании многоугольника можно указывать необязательное поле <code>fillRule</code>.

* Обязательный параметр/опция.

Примеры:

1.

```
// Примеры использования метода при различных входных данных.
var result = ymaps.geoQuery(objects).addToMap(myMap);

// 1. IGeoObject.
// Можно вести поиск относительно стороннего объекта.
var polyline = new ymaps.Polyline([[35, 65], [35, 66], [34, 62], [34, 63]]);
myMap.geoObjects.add(polyline);
var closestObject = result.getClosestTo(polyline);
// Можно искать по объекту из этой же выборки.
var closestToFirst = result.getClosestTo(result.get(0));

// 2. IGeometry.
var closestToGeometry = result.getClosestTo(placemark.geometry);

// 3. Карта.
```

```
// Найдём объект, ближайший к границе видимой области карты.  
var edgeObject = result.getClosestTo(myMap);  
  
// 4. Объект, ближайший к точке.  
var closestObject = result.getClosestTo([34, 53]);
```

2.

```
// Пример использования при асинхронных операциях.  
var result = уmaps.geoQuery(умaps.geocode('Париж')).addToMap(myMap);  
// Дождёмся ответа от сервера и получим объект, ближайший к точке.  
result.then(function () {  
    var closestObject = result.getClosestTo([34, 65]);  
    // Если ответ пуст, то ближайший объект не найдётся.  
    if (closestObject) {  
        closestObject.balloon.open();  
    }  
});
```

getExtreme

```
{Number} getExtreme(key)
```

Возвращает максимальные и минимальные значения координат среди координат объектов выборки.

Параметры:

Параметр	Значение по умолчанию	Описание
key *	—	Тип: String Ключ для получения данных. Может принимать следующие значения: <ul style="list-style-type: none">• top;• right;• bottom;• left.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример работы при синхронных операциях.  
alert('Самая северная координата: ', уmaps.geoQuery(myMap.geoObjects).getExtreme('top'));
```

2.

```
// Пример работы при асинхронных операциях.  
var result = уmaps.geoQuery(умaps.geocode('Новгород'));  
// Дождёмся ответа от сервера и получим самую северную координату.  
result.then(function () {  
    alert('Самая северная координата в ответе: ' + result.getExtreme('top'));  
});
```

getExtremeObject

```
{IGeoObject} getExtremeObject(key)
```

Метод, возвращающий объект с минимальной или максимальной координатой среди координат объектов выборки.

Возвращает объект с требуемой координатой.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key *</code>	—	Тип: String Ключ для поиска объекта. Может принимать следующие значения: <ul style="list-style-type: none">• top;• right;• bottom;• left.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример работы при синхронных операциях.  
// Откроем балун на самом северном объекте.  
var topObject = ymaps.geoQuery(myMap.geoObjects).getExtremeObject('top');  
topObject.balloon.open();
```

2.

```
// Пример работы при асинхронных операциях.  
var result = ymaps.geoQuery(ymaps.geocode('море Лаптевых'));  
// Нужно дождаться ответа от сервера, чтобы работать с данными.  
result.then(function () {  
    var topObject = result.getExtremeObject('top');  
    // Если ответ пуст, то объект не найден.  
    if (topObject) {  
        topObject.balloon.open();  
    }  
});
```

getGlobalPixelBounds

```
{Number[][]|null} getGlobalPixelBounds(map)
```

Метод, возвращающий глобальные пиксельные координаты области для текущего зума карты.

Возвращает глобальные пиксельные координаты области, охватывающей объекты результата.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map *</code>	—	Тип: Map Карта, для которой ведутся расчеты.

* Обязательный параметр/опция.

Пример:

```
var result = ymaps.geoQuery(placemarks).search('properties.type="shop"').getGlobalPixelBounds(myMap);
```

```
if (Math.abs(result[0][0] - result[1][0]) > myMap.container.getSize()[0]) {  
    alert('Объекты не поместятся на карту по ширине!');  
}
```

getGlobalPixelCenter

```
{Number[]} getGlobalPixelCenter(map)
```

Возвращает центр области, охватывающей объекты результата, в глобальных пиксельных координатах для текущего зума карты.

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Карта, для которой нужно произвести расчеты.

* Обязательный параметр/опция.

Пример:

```
// Посчитаем номер тайла, на который приходится центр области, охватывающей результат.  
var globalPixelCenter = ymaps.geoQuery(objects).getGlobalPixelCenter(myMap),  
    tileNumber = [  
        Math.floor(globalPixelCenter[0] / 256),  
        Math.floor(globalPixelCenter[1] / 256)  
    ];  
alert('Номер центрального тайла: ' + tileNumber[0] + ' ' + tileNumber[1]);
```

getIterator

```
{IIterator} getIterator()
```

Возвращает итератор по объектам результата.

Примеры:

1.

```
// Использование итератора с синхронными операциями.  
// Поиск элементов, соответствующих координатам щелчка.  
myMap.events.add('click', function (event) {  
    var iterator = ymaps.geoQuery(myMap.geoObjects)  
        .searchContaining(event.getCoordinates())  
        .getIterator(),  
        obj;  
    while ((obj = iterator.getNext()) !== iterator.STOP_ITERATION) {  
        // Совершаем необходимые действия над геообъектом.  
    }  
});
```

2.

```
// Использование итератора при асинхронных операциях.  
// Создадим результат из запроса к геокодеру.  
var result = ymaps.geoQuery(ymaps.geocode("Староколпакский переулок")).search("lat > 20");  
// Обратите внимание, что мы только отправили запрос к серверу, ответ придет чуть позже.  
// Поскольку запрос асинхронный, нужно дождаться его готовности перед получением результата.  
result.then(function () {  
    var iterator = result.getIterator(),  
        var obj;  
    while ((obj = iterator.getNext()) !== iterator.STOP_ITERATION) {  
        // Совершаем необходимые действия над геообъектом.  
    }  
});
```

getLength

```
{Number} getLength()
```

Возвращает количество элементов в результате.

Примеры:

1.

```
var result = ymaps.geoQuery(myMap.geoObject).searchIntersect(myPolygon);
alert('Количество геообъектов, пересекающих многоугольник: ' + result.getLength());
```

2.

```
var result = ymaps.geoQuery(ymaps.geocode('Иваново')).searchInside(myMap);
// Поскольку мы отправили запрос на сервер, сразу посчитать количество элементов
// в результате нельзя. Нужно дождаться ответа.
result.then(function () {
    alert('Количество объектов в видимой области карты: ' + result.getLength());
});
```

getMaxZoom

```
{Number} getMaxZoom(map[, options])
```

Метод, вычисляющий максимальный коэффициент масштабирования, при котором в видимую область карты попадают все объекты.

Возвращает максимальный коэффициент масштабирования карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: Map Карта, для которой ведется расчет.
<code>options</code>	—	Тип: Object Опции.
<code>options.useMapMargin</code>	<code>true</code>	Тип: Boolean Нужно ли учитывать отступы карты map.margin.Manager .

* Обязательный параметр/опция.

Пример:

```
// Вычислим максимальный коэффициент масштабирования,
// при котором видны все объекты выборки
// и выставим ограничение для карты.
var maxZoom = ymaps.geoQuery(objects).getMaxZoom();
myMap.options.set('maxZoom', maxZoom);
```

getParent

```
{GeoQueryResult|null} getParent()
```

Возвращает ссылку на родительскую выборку, если текущая выборка была создана в результате изменения другого объекта `GeoQueryResult`.

Пример:

```
ymaps.geoQuery(objectsArray)
// Сначала выберем объекты с типом "кафе" и зададим для них стили.
```

```
.search('properties.type == "cafe"')
.setOptions('preset', 'islands#yellowDotIcon')
// А затем, не прерывая цепочку вызовов,
// вернемся к исходной выборке и зададим стили для другой группы объектов.
.getParent()
.search('properties.type == "shop"')
.setOptions('preset', 'islands#greenDotIcon');
```

indexOf

```
{Number} indexOf(item)
```

Возвращает индекс элемента в выборке. Если элемент не найден, возвращается -1.

Параметры:

Параметр	Значение по умолчанию	Описание
item *	—	Тип: IGeoObject Искомый объект.

* Обязательный параметр/опция.

Пример:

```
// Отсортируем выборку по полю name.
var result = ymaps.geoQuery(polygons).sort('properties.name');
alert('Новая позиция первого элемента: ' + result.indexOf(polygons[0]));
```

intersect

```
{GeoQueryResult} intersect(result)
```

Метод создает новую выборку, содержащую общие элементы для двух других выборок.

Возвращает новую выборку, содержащую результат пересечения.

Параметры:

Параметр	Значение по умолчанию	Описание
result *	—	Тип: GeoQueryResult Выборка, с которой пересекают исходную.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример пересечения при синхронных операциях.
var result = ymaps.geoQuery(placemarks),
    var greenObjects = result.search('properties.color="green"'),
    var roundObjects = result.search('properties.shape="round"'),
    var greenRoundObjects = greenObjects.intersect(roundObjects);
alert('Количество круглых зеленых объектов: ' + greenRoundObjects.getLength());
```

2.

```
// Пример пересечения при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('Ивановка')),
    var filteredByLat = result.search('lat > 56'),
    var filteredByLong = result.search('long > 36'),
    var intersectedResult = filteredByLat.intersect(filteredByLong);
// Поскольку исходный запрос асинхронный, нужно дождаться готовности данных.
intersectedResult.then(function () {
    alert('Количество объектов с именем "Ивановка" +
```

```
'с координатами более [56, 36]: ' +
intersectedResult.getLength());
});
```

isReady

```
{Boolean} isReady()
```

Возвращает признак, готовы результаты выборки или еще обрабатываются.

Пример:

```
var result = ymaps.geoQuery(ymaps.geocode('Иваново'));
if (!result.isReady()) {
  result.then(function () {
    // Обработка данных.
  });
} else {
  // Обработка данных.
}
```

map

```
{GeoQueryResult} map(callback[, context])
```

Метод, вызывающий метод callback для всех элементов выборки и формирующий новую выборку на основе полученных результатов.

Возвращает новую выборку.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>callback *</code>	—	Тип: Function Функция-обработчик. Принимает на вход элемент выборки. Возвращает экземпляр IGeoObject .
<code>context</code>	—	Тип: Object Контекст выполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

```
// Добавим на карту только объекты-окружности.
var circlesResult = ymaps.geoQuery(objects).search('geometry.type="Circle"').addToMap(ymap),
// Также добавим на карту метки, обозначающие центры окружностей.
var centers = circlesResult.map(function (object) {
  return new ymaps.Placemark(object.geometry.getCenter());
}).addToMap(ymap);
```

remove

```
{GeoQueryResult} remove(objects)
```

Удаляет объекты из результата. Не изменяет исходный объект, а создает новый, содержащий результирующий набор геообъектов.

Возвращает новый объект с результирующим набором геообъектов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>objects</code> *	—	<p>Тип: <code>Object</code></p> <p>Объекты могут быть представлены в разном виде:</p> <ul style="list-style-type: none">• <code>IGeoObject</code> - объект, реализующий соответствующий интерфейс;• <code>IGeoObject[]</code> - массив объектов, реализующих соответствующий интерфейс;• <code>ICollection</code> - коллекция объектов, реализующих интерфейс <code>IGeoObject</code>;• <code>ICollection[]</code> - массив коллекций объектов, реализующих интерфейс <code>IGeoObject</code>;• <code>GeoQueryResult</code> - объект класса <code>GeoQueryResult</code>. Обратите внимание, что при асинхронных операциях нужно дождаться готовности результата для его корректного удаления;• <code>vow.Promise</code> - объект класса <code>vow.Deferred</code>. Должен быть зарезовлен массивом геообъектов или объектом с полем <code>geoObjects</code>.

* Обязательный параметр/опция.

Примеры:

1.

```
var objects = [  
  new ymaps.Placemark([34, 56]),  
  new ymaps.Rectangle([[34, 56], [36, 57]])  
];
```

```

    ];
    var result = ymaps.geoQuery(objects);
    // Обратите внимание, что в результате будет получен другой объект GeoQueryResult,
    // а старый останется без изменений.
    var newResult = result.remove(objects[1]);

```

2.

```

// Удаление из GeoQueryResult массива геообъектов.
var objects = [
    new ymaps.Placemark([34, 56]),
    new ymaps.Rectangle([34, 56], [36, 57]),
    new ymaps.Placemark([35, 64])
];
var result = ymaps.geoQuery(objects);
// Обратите внимание, что в результате будет получен другой объект GeoQueryResult,
// а старый останется без изменений.
var newResult = result.remove([objects[1], objects[2]]);

```

3.

```

// Удаление коллекции геообъектов из GeoQueryResult.
// Добавим на карту объекты, которых там еще нет.
myGeoQueryResult.remove(myMap.geoObjects).addToMap(myMap);

```

4.

```

// Удаление из GeoQueryResult данных другого GeoQueryResult.
var result1 = ymaps.geoQuery(placemarks).search('properties.color="green"');
var result2 = ymaps.geoQuery(placemarks).search('properties.shape="circle"');
var result3 = result1.remove(result2);
alert('Количество зеленых некруглых фигур: ' + result3.getLength());

```

removeEvents

```

{} removeEvents(events, callback, context)

```

Удаляет подписку на событие с объекта. Обратите внимание, что для корректной отписки передаваемые аргументы должны быть точно такие же, как при подписке через метод `addEvents`.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>events</code> *	—	Тип: String String[] Тип или массив типов событий, на которые была сделана подписка.
<code>callback</code> *	—	Тип: Function Функция-обработчик, которую указали при подписке.
<code>context</code> *	—	Тип: Object Контекст, который был указан при подписке.

* Обязательный параметр/опция.

Пример:

```

var callback = function () {
    alert('Вы кликнули по кругу!');
};
ymaps.geoQuery(map.geoObjects).search('geometry.type="Circle"').addEvents('click', callback);
// ...

```

```
ymaps.geoQuery(map.geoObjects).search('geometry.type="Circle"]').removeEvents('click', callback);
```

removeFrom

```
{GeoQueryResult} removeFrom(collection)
```

Метод для удаления объектов из коллекции.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>collection *</code>	—	Тип: <code>ICollection</code> Коллекция, из которой нужно удалить объекты.

* Обязательный параметр/опция.

Пример:

```
// Покажем на карте все объекты.  
var result1 = ymaps.geoQuery(placemarks).addTo(myMap.geoObjects),  
// А затем скроем объекты из северного полушария.  
result2 = result1.search('lat > 0').removeFrom(myMap.geoObjects);
```

removeFromMap

```
{GeoQueryResult} removeFromMap(map)
```

Метод для удаления объектов с карты.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map *</code>	—	Тип: <code>Map</code> Карта, с которой нужно удалить объекты.

* Обязательный параметр/опция.

Пример:

```
// Покажем на карте все объекты.  
var result1 = ymaps.geoQuery(placemarks).addToMap(myMap),  
// А затем скроем объекты из северного полушария.  
var result2 = result1.search('lat > 0').removeFromMap(myMap);
```

reverse

```
{GeoQueryResult} reverse()
```

Переставляет элементы выборки в обратном порядке и возвращает новую выборку.

Возвращает новую выборку с элементами в обратном порядке.

Примеры:

1.

```
// Использование при синхронных запросах.  
var result = ymaps.geoQuery(ymap.geoObjects).sort('x'),  
    invertedResult = result.reverse();
```

2.

```
// Использование при асинхронных запросах.  
var result = ymaps.geoQuery(ymaps.geocode('село Добрые пчелы')).sort('x'),  
    invertedResult = result.reverse();  
invertedResult.then(function () {  
    // Результат получен. Готов к использованию.  
});
```

search

```
{GeoQueryResult} search(condition)
```

Метод для поиска объектов выбоки, удовлетворяющих условиям.

Возвращает новую выборку, содержащую результаты поиска.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>condition *</code>	—	<p>Тип: String Function</p> <p>Строковый шаблон для поиска или функция-фильтр. Строковый шаблон имеет структуру "<fieldname> <condition> <expression>". Возможные значения <fieldname>:</p> <ul style="list-style-type: none"> 'lat' - широта. Поддерживается только для точечных объектов; 'lng', 'long' - долгота. Поддерживается только для точечных объектов; 'x' - глобальные пиксельные координаты по оси x. Поддерживается только для точечных объектов; 'y' - глобальные пиксельные координаты по оси y. Поддерживается только для точечных объектов; 'geometry.type' - тип геометрии. Результат выполнения метода <code>geometry.getType()</code>. 'geometry.coordinates.<index>' - координаты. Результат выполнения метода <code>geometry.getCoordinates()</code>. Для получения доступа к координате следует указывать ее индексы через точку - 'geometry.coordinates.0.1'. 'properties.<path>' - значение поля данных; 'options.<key>' - значение опций. <p>Допустимые значения <condition>:</p> <ul style="list-style-type: none"> '>' '>=' '==', '='" '!=' '<=' '<' 'like', 'regex' - соответствие значения <fieldname> некоторому регулярному выражению. <p>Допустимые значения <expression>:</p> <ul style="list-style-type: none"> Число; Строка - обязательно указывается в кавычках; true; false;

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример поиска объектов при синхронных операциях.
var result = ymaps.geoQuery(ymap.geoObjects);

// Поиск объектов с определенным типом геометрии. Обратите внимание, что
// значение поля указано в кавычках - это строка.
result.search('geometry.type = "Circle"')
// Поиск по координате.
  .search('geometry.coordinates. 0 > 100')
// Поиск по широте.
  .search('lat < 0')
// Поиск по пиксельным координатам.
  .search('x >= 100')
// Поиск по значению поля в "properties".
  .search('properties.name != null').search('properties.name rlike "(.) \\\\"')
  .search('properties.author.name = "Степан"')
// Поиск по значению опции.
  .search('options.visible = true');
```

2.

```
// Использование метода при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('Лось'));
// Для работы с результатом нужно дождаться его готовности.
result.then(function () {
  result.search('properties.description regexp "*поселок*"')
    .addToMap(ymap)
    .applyBoundsToMap(ymap);
});
```

searchContaining

```
{GeoQueryResult} searchContaining(object)
```

Метод, создающий новую выборку из объектов, содержащих указанный объект. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.

Возвращает новую выборку, содержащую искомые объекты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>object *</code>	—	<p>Тип: Object</p> <p>Объект, относительно которого будет производится поиск. Может принимать следующие значения:</p> <ul style="list-style-type: none"> <code>IGeoObject</code> - объект, реализующий интерфейс <code>IGeoObject</code>; <code>IGeometry</code> - объект, реализующий интерфейс <code>IGeometry</code>; Map - карта. В этом случае опорным объектом будет прямоугольная граница карты; <code>Number[]</code> - координаты точки; Если передан один параметр, трактуется как точка. Если передано два аргумента, трактуется как центр окружности; <code>Number[][]</code> - координаты прямоугольной области; Object - Json-описания геометрии.

* Обязательный параметр/опция.

Примеры:

1.

```
// Примеры использования метода при различных входных данных.
var result = ymaps.geoQuery(objects).addToMap(myMap);

// 1. IGeoObject.
var polygon = new ymaps.Polygon([[35, 65], [35, 66], [34, 62], [34, 63], [35, 65]]);
myMap.geoObjects.add(polygon);
var objectsContainingPolygon = result.searchContaining(polygon);

// 2. IGeometry.
var objectsContainingGeometry = result.searchContaining(polygon.geometry);

// 3. Map.
var objectsContainingMapBounds = result.searchContaining(myMap);
```

2.

```
// Пример использования при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('кафе'))
  .addToMap(ymap);
// Дождемся ответа от сервера перед обработкой результатов.
result.then(function () {
  var areas = result.map(function (object) {
    return new ymaps.Circle(object.geometry.getCoordinates(), 100);
  })
  .setOptions('visible', false)
  .addToMap(ymap);
  myMap.events.add('click', function (event) {
    if (areas.searchContaining(event.getCoordinates()).getLength()) {
      alert('Вы кликнули около кафе.');
```

searchInside

```
{GeoQueryResult} searchInside(object)
```

Метод, создающий новую выборку из объектов, целиком входящих в указанный объект. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.

Возвращает новую выборку, содержащую искомые объекты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>object *</code>	—	<p>Тип: Object</p> <p>Объект, относительно которого будет производится поиск. Может принимать следующие значения:</p> <ul style="list-style-type: none"> <code>IGeoObject</code> - объект, реализующий интерфейс <code>IGeoObject</code>; <code>IGeometry</code> - объект, реализующий интерфейс <code>IGeometry</code>; <code>Map</code> - карта. В этом случае опорным объектом будет прямоугольная граница карты;

* Обязательный параметр/опция.

Примеры:

1.

```
// Примеры использования метода при различных входных данных.
var result = ymaps.geoQuery(objects).addToMap(ymap);

// 1. IGeoObject.
var polygon = new ymaps.Polygon([[35, 65], [35, 66], [34, 62], [34, 63], [35, 65]]);
myMap.geoObjects.add(polygon);
var objectsInsidePolygon = result.searchInside(polygon);

// 2. IGeometry.
```



```
var objectsInsideGeometry = result.searchInside(polygon.geometry);

// 3. Карта.
// Найдём объекты, целиком лежащие в видимой области карты.
var visibleObject = result.searchInside(myMap);
```

2.

```
// Пример использования при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('Иваново'))
    .setOptions('visible', false)
    .addToMap(myMap);
// Дождёмся ответа от сервера и покажем объекты, попадающие в видимую область карты.
result.then(function () {
    result.searchInside(myMap).setOptions('visible', true);
});
```

searchIntersect

```
{GeoQueryResult} searchIntersect(object[, options])
```

Метод, создающий новую выборку из объектов, пересекающих указанный объект. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.

Возвращает новую выборку, содержащую искомые объекты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>object *</code>	—	<p>Тип: Object</p> <p>Объект, относительно которого будет производится поиск. Может принимать следующие значения:</p> <ul style="list-style-type: none"> <code>IGeoObject</code> - объект, реализующий интерфейс <code>IGeoObject</code>; <code>IGeometry</code> - объект, реализующий интерфейс <code>IGeometry</code>; <code>Map</code> - карта. В этом случае опорным объектом будет прямоугольная граница карты;
<code>options</code>	—	<p>Тип: Object</p> <p>Опции.</p>

Параметр	Значение по умолчанию	Описание
<code>options.considerOccurance</code>	<code>true</code>	<p>Тип: Object</p> <p>Флаг, показывающий, считаем ли мы пересечением, когда одна фигура полностью содержится в другой.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
// Примеры использования метода при различных входных данных.
var result = ymaps.geoQuery(objects).addToMap(ymap);

// 1. IGeoObject.
var polygon = new ymaps.Polygon([[35, 65], [35, 66], [34, 62], [34, 63], [35, 65]]);
ymap.geoObjects.add(polygon);
var objectsIntersectPolygon = result.searchIntersect(polygon);

// 2. IGeometry.
var objectsIntersectGeometry = result.searchIntersect(polygon.geometry);

// 3. Карта.
// При этом будем искать только объекты, пересекающие непосредственно
// границы карты. То есть объекты, попадающие целиком в видимую область карты,
// не попадут в финальную выборку.
var objectsIntersectMapBounds = result.searchIntersect(ymap, {considerOccurance: false});
```

2.

```
// Пример использования при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('cafe'))
    .setOptions('visible', false)
    .addToMap(ymap);

// Дождемся ответа от сервера перед обработкой результатов.
result.then(function () {
    result.searchIntersect(ymap).setOptions('visible', true);
});
```

setOptions

```
{GeoQueryResult} setOptions(key[, value])
```

Метод, позволяющий задать значение опций всем элементам выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key</code> *	—	<p>Тип: String Object</p> <p>Название опции или хэш с опциями и их значениями.</p>
<code>value</code>	—	<p>Тип: Object</p> <p>Значение опции.</p>

* Обязательный параметр/опция.

Пример:

```
var result = ymaps.geoQuery(placemarks);
// Сделаем видимыми элементы, попадающие в прямоугольную область.
result.searchIntersect(myBounds).setOptions('visible', true);

// Зададим опции с помощью хэша.
result.setOptions({zIndex: 10, fillColor: '#ff0005'});
```

setProperties

```
{GeoQueryResult} setProperties(path, value)
```

Метод, позволяющий задать значение поля `properties` всем элементам выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>path</code> *	—	Тип: String Имя поля, которому присваивается значение. Может содержать ".".
<code>value</code> *	—	Тип: Object Значение поля.

* Обязательный параметр/опция.

Пример:

```
var result = ymaps.geoQuery(objects);
// Пометим элементы, попадающие в области.
result.searchIntersect(myBounds1).setProperties('intersectBounds', true);
result.searchIntersect(myBounds2).setProperties('intersectBounds', true);
// ...
result.search('properties.intersectBounds = true').addToMap(myMap);
```

slice

```
{GeoQueryResult} slice(begin[, end])
```

Метод, возвращающий срез выборки.

Возвращает новую выборку, содержащую элементы среза.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>begin</code> *	—	Тип: Number Индекс начального элемента в срезе.

Параметр	Значение по умолчанию	Описание
<code>end</code>	—	Тип: Number Индекс элемента в выборке, на котором срез завершится. При этом последним в новом срезе будет элемент с индексом <code>end-1</code> .

* Обязательный параметр/опция.

Примеры:

1.

```
// Выполним срез при синхронной обработке.  
var result = уmaps.geoQuery(уmaps.geoObjects).slice(0, 10);  
alert('Количество элементов в новой выборке:' + result.getLength());
```

2.

```
// Взятие среза при асинхронной обработке.  
var result = уmaps.geoQuery(уmaps.geocode('кафе Москва')).slice(0, 10);  
// Сразу после запроса результат еще не готов.  
alert('Пока выборка пуста. Количество элементов в выборке: ' + result.getLength());  
// Дождемся готовности результата и посмотрим, сколько элементов будет в выборке.  
result.then(function () {  
    alert('Ответ получен. Количество данных в выборке: ' + result.getLength());  
});
```

sort

```
{ } sort(comparator)
```

Метод для сортировки объектов выборки. Не изменяет исходную выборку, а создает новую, содержащую упорядоченные объекты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>comparator *</code>	—	<p>Тип: String Function</p> <p>Строка с шаблоном сортировки или функция-компаратор. Строка-шаблон может быть представлена в виде '<code><fieldname> [<order>=asc]</code>'. Возможные значения <code><fieldname></code>:</p> <ul style="list-style-type: none"> 'lat' - широта. Поддерживается только для точечных объектов; 'lng', 'long' - долгота. Поддерживается только для точечных объектов; 'x' - глобальные пиксельные координаты по оси x. Поддерживается только для точечных объектов; 'y' - глобальные пиксельные координаты по оси y. Поддерживается только для точечных объектов; 'geometry.type' - тип геометрии. Результат выполнения метода <code>geometry.getType()</code>. 'geometry.coordinates.<index>' - координаты. Результат выполнения метода <code>geometry.getCoordinates()</code>. Для получения доступа к координате следует указывать ее индексы через точку - <code>'geometry.coordinates.0.1'</code>. 'properties.<path>' - значение поля данных; 'options.<key>' - значение опций. <p>Возможные значения необязательного параметра <code><order></code>: <code>parameter</code>:</p> <ul style="list-style-type: none"> 'asc' - сортировка по возрастанию; 'desc' - сортировка по убыванию. <p>Функция-компаратор принимает на вход два элемента выборки. Возвращаемые значения: если первый объект больше второго, функция возвращает значение <code>> 0</code>. Если первый объект равен второму, функция возвращает <code>0</code>. Если первый объект меньше второго, функция возвращает значение <code>< 0</code>.</p>

* Обязательный параметр/опция.

Пример:

```
// Пример использования searchContaining при синхронных операциях.
var result = ymaps.geoQuery(myMap.geoObjects);
result.sort('lat').sort('x')
  .sort('properties.name desc')
  .sort('options.preset')
  .sort(function (a, b) {
    if (a.properties.get('name') == b.properties.get('name')) {
      return a.geometry.getCoordinates()[0] - b.geometry.getCoordinates()[0];
    } else {
      return (a.properties.get('name') > b.properties.get('name')) ? 1 : -1;
    }
  });
```

sortByDistance

```
{GeoQueryResult} sortByDistance(object)
```

Метод, позволяющий получить выборку, содержащую объекты, отсортированные по расстоянию до указанного объекта. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов. Не изменяет исходную выборку.

Возвращает новую упорядоченную выборку.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>object *</code>	—	<p>Тип: Object</p> <p>Объект, до которого будет считаться расстояние. Может принимать следующие значения:</p> <ul style="list-style-type: none"> <code>IGeoObject</code> - объект, реализующий интерфейс <code>IGeoObject</code>; <code>IGeometry</code> - объект, реализующий интерфейс <code>IGeometry</code>; Map - карта. В этом случае опорным объектом будет прямоугольная граница карты; <code>Number[]</code> - координаты точки; <code>Number[][]</code> - координаты прямоугольной области; Object - Json-описание геометрии. Содержит поля <code>type</code> и <code>coordinates</code>. При описании круга также необходимо поле <code>radius</code>. При описании многоугольника можно указывать необязательное поле <code>fillRule</code>.

* Обязательный параметр/опция.

Примеры:

1.

```
// Примеры использования метода при различных входных данных.
var result = ymaps.geoQuery(objects).addToMap(ymap);

// 1. IGeoObject.
var polyline = new ymaps.Polyline([[35, 65], [35, 66], [34, 62], [34, 63]]);
ymap.geoObjects.add(polyline);
var sortedByPolyline = result.sortByDistance(polyline);

// 2. IGeometry.
var sortedByGeometry = result.sortByDistance(placemark.geometry);
```

```
// 3. Map.
var sortedByMapBounds = result.sortByDistance(myMap);

// 4. Выборка, отсортированная по точке.
var sortedByPoint = result.sortByDistance([34, 53]);
```

2.

```
// Пример использования при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('Париж')).addToMap(myMap).sortByDistance([45, 64]);
// Дождемся ответа от сервера и получим ближайший и наиболее удаленный
// объект по отношению к точке.
result.then(function () {
    alert('Ближайший объект имеет координаты ' + result.get(0).geometry.getCoordinates());
    alert('Наиболее удаленный объект имеет координаты ' + result.get(result.getLength() -
1).geometry.getCoordinates());
});
```

then

```
{GeoQueryResult} then([onFulfill[, onReject, context])
```

Подписка на "обещание".

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>onFulfill</code>	—	Тип: Function Функция-обработчик, которая вызывается, если обещание выполнено.
<code>onReject</code>	—	Тип: Function Функция-обработчик, которая вызывается, если обещание не выполнено (произошла ошибка).
<code>context</code> *	—	Тип: Object Контекст выполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

```
var result = ymaps.geoQuery(ymaps.geocode('река Лена'));
result.then(function () {
    alert('Количество найденных объектов: ' + result.getLength());
}, function () {
    alert('Произошла ошибка.');
```

unsetOptions

```
{GeoQueryResult} unsetOptions(keys)
```

Метод, позволяющий обнулять значение опций всем элементам выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>keys *</code>	—	Тип: String String Название или массив названий опций, которые должны быть аннулированы.

* Обязательный параметр/опция.

Пример:

```
result.unsetOptions('visible');
```

unsetProperties

```
{GeoQueryResult} unsetProperties(path)
```

Метод позволяющий обнулить значение поля `properties` у всех элементов выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>path *</code>	—	Тип: String Имя поля, значение которого нужно обнулить. Может содержать ".".

* Обязательный параметр/опция.

Пример:

```
var result = ymaps.geoQuery(objects);  
// Пометим элементы, попадающие в первую область, но не попадающие во вторую.  
result.searchIntersect(myBounds1).setProperties('intersectBounds', true);  
result.searchIntersect(myBounds2).unsetProperties('intersectBounds', true);  
// ...  
result.search('properties.intersectBounds = true').addToMap(myMap);
```

geoXml

geoXml.load

Статическая функция.

Загружает XML-файл с географическими данными и преобразует его в коллекцию [GeoObjectCollection](#).

Сформированная коллекция может быть передана заданной функции для последующей обработки.

Поддерживаемые форматы XML-данных: YMapsML, KML, GPX. Для самой верхней коллекции геообъектов GPX-файла доступны следующие пресеты:

- '`grx#interactive`' - обеспечивает вывод информации о точке маршрута при клике. Также при использовании этого пресета в макете балуна становятся доступными следующие свойства геообъекта: `time`, `velocity`, `trackName`, `trackDescription`, `pointName`, `pointDescription`, `lon`, `lat`, `sym`. Используется по умолчанию.
- '`grx#plain`' - элементы `grx`-коллекции ведут себя как обычные геообъекты;

Возвращает объект-promise. Если xml-файл по указанному урлу будет успешно загружен, promise будет подтверждён и получит в качестве параметров объект со следующими полями:

- `geoObjects` - коллекция геообъектов [GeoObjectCollection](#);
- `mapState` - описание состояния карты [IMapState](#) (только для YMapsML);

```
{ vow.Promise } geoXml.load(url)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>url</code> *	—	Тип: String URL файла с географическими данными.

* Обязательный параметр/опция.

Пример:

```
// Создаем и инициализируем карту.
// Загрузка и отображение файла ymapsml из сервиса My Maps
ymaps.geoXml.load('http://maps.yandex.ru/export/usermaps/HNQ5uTUgbjy6L0dW2uReUjSoXb1Ad7jw/')
    .then(function (res) {
        // Добавление элементов файла ymapsml на карту
        map.geoObjects.add(res.geoObjects);
        // Установка границ и типа карты..
        // границы res.mapState.getBounds() применяются к карте асинхронно,
        // потому что нам нужно получить информацию о доступных уровнях масштабирования для этих границ
        // res.mapState.getType() применяется синхронно.
        if (res.mapState) {
            res.mapState.applyToMap(map).then(function () {
                alert('Boundaries applied to the map ' + res.mapState.getBounds().toString());
            });
        }
        // Если информация о границах не указана в repr:View в файле YMapsML,
        // мы можем применить gml:boundedBy для верхнего элемента ymaps: GeoObjectCollection
        else if (res.geoObjects.properties.get('boundedBy')) {
            map.setBounds(res.geoObjects.properties.get('boundedBy'), {
                checkZoomRange: true
            });
        }
    });

// Загрузка и отображение файла KML
ymaps.geoXml.load('https://sandbox.api.maps.yandex.net/examples/ru/2.1/geoxml_display/geoObjects.kml')
    .then(function (res) {
        map.geoObjects.add(res.geoObjects);
    });

// Загрузка и отображение файла GPX
ymaps.geoXml.load('https://sandbox.api.maps.yandex.net/examples/ru/2.1/geoxml_display/geoObjects.gpx')
    .then(function (res) {
        res.geoObjects.options.set({
            balloonContentBodyLayout: ymaps.templateLayoutFactory.createClass(
                // Балун будет показывать только имя геообъекта свойство и скорость
                '<b>{{ properties.name }}</b> {{ properties.velocity }}'
            )
        });
        map.geoObjects.add(res.geoObjects);
        // Метаданные о границах из файла GPX хранятся в свойствах коллекции res.geoObjects.
        // Применим эти границы к карте.
        if (res.geoObjects.properties.get('boundedBy')) {
            map.setBounds(res.geoObjects.properties.get('boundedBy'), {
                checkZoomRange: true
            });
        }
    });
});
```

getZoomRange

Статическая функция.

Определяет доступный диапазон уровней масштабирования в указанной точке для указанного типа карты.

Возвращает Promise объект, который будет подтверждён массивом из двух чисел — максимальный и минимальный уровень масштабирования в данной точке.

```
{ vow.Promise } getZoomRange(mapType, coords, customizable)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>mapType</code> *	—	Тип: String MapType Тип карты. Строка-ключ из mapType.storage , либо экземпляр класса MapType .
<code>coords</code> *	—	Тип: Number[] Координаты точки, для которой нужно узнать диапазон доступных уровней масштабирования.
<code>customizable</code> *	—	Тип: ICustomizable =null Объект, который содержит менеджер опций. Опции объекта будут учитываться при получении результата.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пусть мы хотим инициализировать карту на максимальном масштабе.
var myMap;
ymaps.getZoomRange('yandex#map', [55.750516, 37.615924]).then(function (result) {
    myMap = new ymaps.Map('mapContainer', {
        center: [55.750516, 37.615924],
        zoom: result[1]
    });
});
```

2.

```
// Инициализируем карту с помощью геокодера, с центром на доме 16 по улице Льва Толстого
// на максимально возможном масштабе.
var myMap;
ymaps.geocode("Москва, Льва Толстого, 16").then(function (geoData) {
    var coords = geoData.geoObjects.get(0).geometry.getCoordinates();
    ymaps.getZoomRange('yandex#map', coords).then(function (zoomRange) {
        myMap = new ymaps.Map('mapContainer', {
            center: coords,
            zoom: zoomRange[1]
        });
    });
});
```

graphics

graphics.style

graphics.style.color

Статический объект.

Задаёт цвет графической фигуры в форматах `#RGB`,`#RGBA`,`#RRGGBB`,`#RRGGBBAA`,`rgb(r,g,b)`,`rgba(r,g,b,a)`

Пример:

```
strokeColor: '#F00';
strokeColor: '#FF0000';
strokeColor: '#FF0000AA';
strokeColor: 'rgba(255,0,0,1)'
```

graphics.style.stroke

Статический объект.

Позволяет изменить стиль линии. Задавать значение можно как используя ключи описанные ниже, так и в формате массива. Следует учесть, что в режиме отображения VML(IE<9) могут быть использованы только ключи. Также в стиле линии можно указать смещение начала пунктира через поле offset.

Поля**Пример:**

```
strokeStyle: 'dot';
strokeStyle: [1,2];
strokeStyle: {
  style: 'dot',
  offset: 10
}
```

Поля

Имя	Тип	Описание
dash		Тире
dashdot		Длинное тире-короткое тире
dot		Точки
longdash		Длинные тире
longdashdot		Очень длинное тире - точка
longdashdotdot		Длинное тире-точка-точка
shortdash		Короткие тире
shortdashdot		Тире-точка
shortdashdotdot		Тире-точка-точка
shortdot		Точки через двойной интервал
solid		Сплошная линия

Описание полей**dash**

```
dash
```

Тире

dashdot

```
dashdot
```

Длинное тире-короткое тире

dot`dot`

Точки

longdash`longdash`

Длинные тире

longdashdot`longdashdot`

Очень длинное тире - точка

longdashdotdot`longdashdotdot`

Длинное тире-точка-точка

shortdash`shortdash`

Короткие тире

shortdashdot`shortdashdot`

Тире-точка

shortdashdotdot`shortdashdotdot`

Тире-точка-точка

shortdot`shortdot`

Точки через двойной интервал

solid`solid`

Сплошная линия

HintРасширяет [IHint](#), [Popup](#).

Хинт - всплывающая подсказка, которая может отображать любое HTML-содержимое. Обычно хинт существует на карте в единственном экземпляре, управление им происходит через специальные менеджеры ([maps](#), [geo objects](#), [hotspot layers](#) и т.д.). Не создавайте их самостоятельно, если в этом нет необходимости.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Hint(map[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Ссылка на объект карты.
options	—	Тип: Object Опции.
options.closeTimeout	700	Тип: Number Задержка закрытия (в мс).
options.contentLayout	—	Тип: Function String Макет для содержимого хинта. (Тип: конструктор объекта с интерфейсом ILayout).
options.fitPane	true	Тип: Boolean Флаг, заставляющий инфо-объекта сдвигать свою позицию, чтобы не выходить за границы контейнера.
options.holdByMouse	true	Тип: Boolean Флаг, отменяющий закрытие хинта, находящегося под курсором.
options.interactivityModel	—	Тип: String Ключ модели интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
options.layout	islands#hint	Тип: Function String Внешний макет хинта. (Тип: конструктор объекта с интерфейсом ILayout).

Параметр	Значение по умолчанию	Описание
<code>options.offset</code>	—	Тип: <code>Number[]</code> Дополнительное смещение позиции относительно точки привязки.
<code>options.openTimeout</code>	150	Тип: <code>Number</code> Задержка открытия (в мс).
<code>options.pane</code>	'outerHint'	Тип: <code>String</code> Ключ пейзажа, в который помещается оверлей хинта.
<code>options.zIndex</code>	—	Тип: <code>String</code> z-index хинта.

* Обязательный параметр/опция.

Пример:

```
// Создаем независимый экземпляр хинта и показываем его с собственным макетом в центре карты с помощью указания гео-координат.
(new ymaps.Hint(ymap, {
  projection: ymaps.projection.wgs84Mercator,
  layout: ymaps.templateLayoutFactory.createClass('{{ content }}')
})).open(ymap.getCenter(), 'Привет');
```

Поля

Имя	Тип	Описание
<code>events</code>	<code>IEventManager</code>	Менеджер событий. Унаследовано от <code>IEventEmitter</code> .
<code>options</code>	<code>IOptionManager</code>	Менеджер опций. Унаследовано от <code>ICustomizable</code> .

События

Имя	Описание
<code>close</code>	Закрытие инфо-объекта. Унаследовано от <code>IPopup</code> .
<code>open</code>	Открытие инфо-объекта. Унаследовано от <code>IPopup</code> .
<code>optionschange</code>	Изменение в опциях объекта. Унаследовано от <code>ICustomizable</code> .

Методы

Имя	Возвращает	Описание
<code>close([force])</code>	<code>vow.Promise</code>	Закрывает инфо-объект. Унаследован от <code>IPopup</code> .

Имя	Возвращает	Описание
getData()		Возвращает данные инфо-объекта. Унаследован от IPopup .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopup .
getOverlaySync()	IOverlay	Возвращает оверлей, если тот существует. Унаследован от IPopup .
getPosition()		Возвращает координаты инфо-объекта. Унаследован от IPopup .
isOpen()	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopup .
open([position[, data]])	vow.Promise	Открывает инфо-объект в указанной позиции. Если инфо-объект уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией IProjection , которая содержится в опциях. Унаследован от IPopup .
setData(data)	vow.Promise	Задаёт инфо-объекту новые данные. Унаследован от IPopup .
setPosition(position)	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopup .

hotspot

hotspot.layer

hotspot.layer.addon

hotspot.layer.addon.balloon

Примечание: Конструктор класса `hotspot.layer.addon.balloon` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

[Методы](#)

Методы

Имя	Возвращает	Описание
<code>get(layer)</code>	<code>IPopupManager</code>	Возвращает менеджер балуна хотспотного слоя.

Описание методов

get

```
{IPopupManager} get(layer)
```

Возвращает менеджер балуна хотспотного слоя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>layer *</code>	—	Тип: <code>hotspot.Layer</code> Слой активных областей.

* Обязательный параметр/опция.

Пример:

```
ymaps.hotspot.layer.addon.balloon.get(layer)
```

hotspot.layer.addon.hint

Примечание: Конструктор класса `hotspot.layer.addon.hint` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Методы

Методы

Имя	Возвращает	Описание
<code>get(layer)</code>	<code>IPopupManager</code>	Возвращает менеджер хинта хотспотного слоя.

Описание методов

get

```
{IPopupManager} get(layer)
```

Возвращает менеджер хинта хотспотного слоя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>layer *</code>	—	Тип: <code>hotspot.Layer</code> Слой активных областей.

* Обязательный параметр/опция.

Пример:

```
ymaps.hotspot.layer.addon.hint.get(layer)
```

hotspot.layer.Balloon

Расширяет [IBalloonManager](#).

Менеджер балуна хотспотного слоя. Позволяет управлять балуном хотспотного слоя, открывать и скрывать его. Использует внутри себя менеджер балуна карты [map.Balloon](#). Хотспотные слои содержат экземпляр этого класса, доступный как `myHotspotLayer.balloon`. Не создавайте экземпляры этого класса без необходимости.

См.: [Balloon hotspot.Layer.balloon](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
hotspot.layer.Balloon(hotspotLayer)
```

Параметры:

Параметр	Значение по умолчанию	Описание
hotspotLayer *	—	Тип: Object Слой активных областей.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
autopanbegin	Начало автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
autopanend	Окончание автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
beforeuserclose	Событие, предшествующее событию Balloon.userclose . Позволяет отменить пользовательское действие посредством вызова метода <code>preventDefault</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .

Имя	Описание
close	<p>Заккрытие инфо-объекта. Названия полей доступны через метод Event.get:</p> <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. <p>Унаследовано от IPopupManager.</p>
open	<p>Открытие инфо-объекта. Названия полей доступны через метод Event.get:</p> <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. <p>Унаследовано от IPopupManager.</p>
userclose	<p>Заккрытие балуна пользователем. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. <p>Унаследовано от IBalloonManager.</p>

Методы

Имя	Возвращает	Описание
autoPan()	vow.Promise	<p>Перемещает карту таким образом, чтобы балун оказался видимым.</p> <p>Унаследован от IBalloonManager.</p>
close([force])	vow.Promise	<p>Закрывает инфо-объект.</p> <p>Унаследован от IPopupManager.</p>
destroy()		<p>Отключает менеджер инфо-объекта.</p> <p>Унаследован от IPopupManager.</p>
getData()	<code>Object null</code>	<p>Возвращает данные инфо-объекта или <code>null</code>.</p> <p>Унаследован от IPopupManager.</p>
getOptions()	IOptionManager null	<p>Возвращает менеджер опций или <code>null</code>.</p> <p>Унаследован от IPopupManager.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание вернуть оверлей.</p> <p>Унаследован от IPopupManager.</p>
getOverlaySync()	IOverlay null	<p>Возвращает оверлей, если тот существует.</p> <p>Унаследован от IPopupManager.</p>

Имя	Возвращает	Описание
<code>getPosition()</code>	<code>Number[] null</code>	Возвращает координаты инфо-объекта или <code>null</code> . Унаследован от IPopupManager .
<code>isOpen()</code>	<code>Boolean</code>	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopupManager .
<code>open([position[, data[, options]])</code>	<code>vow.Promise</code>	Открывает балун в указанной позиции.
<code>setData(data)</code>	<code>vow.Promise</code>	Задаёт инфо-объекту новые данные. Унаследован от IPopupManager .
<code>setOptions(options)</code>	<code>vow.Promise</code>	Задаёт инфо-объекту новые опции. Унаследован от IPopupManager .
<code>setPosition(position)</code>	<code>vow.Promise</code>	Задаёт инфо-объекту новую позицию. Унаследован от IPopupManager .

Описание методов

open

```
{vow.Promise} open([position[, data[, options]])
```

Открывает балун в указанной позиции.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>position</code>	—	Тип: <code>Number[]</code> Координаты открытия в глобальных пиксельных координатах.
<code>data</code>	—	Тип: <code>Object</code> Данные.
<code>options</code>	—	Тип: <code>Object</code> Опции.

hotspot.layer.Hint

Расширяет [IHintManager](#).

Менеджер хинта слоя активных областей. Позволяет управлять хинтом слоя активных областей, открывать и скрывать его. Использует внутри себя менеджер хинтов карты [map.Hint](#). Слои активных областей содержат экземпляр этого класса, доступный как `myHotspotLayer.hint`. Не создавайте экземпляры этого класса без необходимости.

См.: [Hint hotspot.Layer.hint](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
hotspot.layer.Hint(hotspotLayer)
```

Параметры:

Параметр	Значение по умолчанию	Описание
hotspotLayer *	—	Тип: Object Слой активных областей.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
close	Заккрытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"><code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. Унаследовано от IPopupManager .
open	Открытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"><code>target</code> - ссылка на объект, на котором был открыт инфо-объект. Унаследовано от IPopupManager .

Методы

Имя	Возвращает	Описание
close ([force])	vow.Promise	Закрывает инфо-объект. Унаследован от IPopupManager .

Имя	Возвращает	Описание
destroy()		Отключает менеджер инфо-объекта. Унаследован от IPopupManager .
getData()	Object null	Возвращает данные инфо-объекта или null. Унаследован от IPopupManager .
getOptions()	IOptionManager null	Возвращает менеджер опций или null. Унаследован от IPopupManager .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopupManager .
getOverlaySync()	IOverlay null	Возвращает оверлей, если тот существует. Унаследован от IPopupManager .
getPosition()	Number[] null	Возвращает координаты инфо-объекта или `null`. Унаследован от IPopupManager .
isOpen()	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopupManager .
open([position[, data[, options]])	vow.Promise	Открывает всплывающую подсказку в указанной позиции.
setData(data)	vow.Promise	Задаёт инфо-объекту новые данные. Унаследован от IPopupManager .
setOptions(options)	vow.Promise	Задаёт инфо-объекту новые опции. Унаследован от IPopupManager .
setPosition(position)	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopupManager .

Описание методов

open

```
{vow.Promise} open([position[, data[, options]])
```

Открывает всплывающую подсказку в указанной позиции.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
position	—	Тип: Number[] Координаты открытия в глобальных пиксельных координатах.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.

hotspot.layer.Object

Расширяет [IHotspotLayerObject](#).

Объект хотспотного слоя.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
hotspot.layer.Object(shape, feature, options)
```

Создаёт объект хотспотного слоя.

Параметры:

Параметр	Значение по умолчанию	Описание
shape *	—	Тип: IShape Фигура активной области.
feature *	—	Тип: Object Описание объекта GeoObject ;
options *	—	Тип: Object Опции объекта.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mousemove	Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code>.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры, описывающей активную область. Экземпляр класса Event.</p> <p>Унаследовано от IHotspotLayerObject.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getGeometry()	Object	<p>Возвращает реальную геометрию объекта.</p> <p>Унаследован от IHotspotLayerObject.</p>

Имя	Возвращает	Описание
getHotspot()	IHotspot	Возвращает объект описывающий активную область. Унаследован от IHotspotLayerObject .
getId()	Number	Возвращает идентификатор объекта. Унаследован от IHotspotLayerObject .
getProperties()	Object	Возвращает данные объекта. Унаследован от IHotspotLayerObject .
setGeometry(geometry)		Задаёт реальную геометрию объекта. Унаследован от IHotspotLayerObject .
setId(id)		Задаёт идентификатор объекта. Унаследован от IHotspotLayerObject .
setProperties(properties)		Задаёт данные объекта. Унаследован от IHotspotLayerObject .

hotspot.Layer

Расширяет [IChildOnMap](#), [ICustomizable](#).

Слой активных областей.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
hotspot.Layer(objectSource[, options])
```

Создаёт слой активных областей. Каждая отдельная область в составе слоя представляет собой объект - [hotspot.layer.Object](#).

Параметры:

Параметр	Значение по умолчанию	Описание
objectSource *	—	Тип: IHotspotObjectSource Источник объектов слоя.
options	—	Тип: Object Опции слоя. Опции для балунов Balloon и хинтов Hint хотспотного слоя нужно указывать с префиксами 'balloon' и 'hint'.

Параметр	Значение по умолчанию	Описание
<code>options.cursor</code>	'pointer'	Тип: String Тип курсора.
<code>options.dontChangeCursor</code>	false	Тип: Boolean Опция, позволяющая запретить изменение курсора при наведении на объект слоя. По умолчанию курсоры изменяются.
<code>options.hasBalloon</code>	true	Тип: Boolean Флаг, показывающий, есть ли у слоя балуны. Если значение флага false, у слоя не будет создано поле <code>.balloon</code> .
<code>options.hasHint</code>	true	Тип: Boolean Флаг, показывающий, есть ли у слоя хинты. Если значение флага false, у слоя не будет создано поле <code>.hint</code> .
<code>options.interactivityModel</code>	'default#layer'	Тип: String Модель интерактивности слоя. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
<code>options.openBalloonOnClick</code>	true	Тип: Boolean Опция, позволяющая запретить открытие балуна при клике на хотспотном объекте. По умолчанию открытие балуна разрешено;
<code>options.openEmptyBalloon</code>	false	Тип: String Открывать балун с пустым содержимым.
<code>options.openEmptyHint</code>	false	Тип: String Показывать всплывающую подсказку с пустым содержимым.
<code>options.openHintOnHover</code>	true	Тип: Boolean Опция, позволяющая запретить показ хинта при наведении на хотспотный объект. По умолчанию показ хинтов разрешен.

Параметр	Значение по умолчанию	Описание
options.pane	'events'	Тип: IEventPane Контейнер элементов карты, на котором слушаются события.
options.zIndex	—	Тип: Number Значение zIndex слоя. Когда активный объект ищется в пределах одного слоя, то при одновременном попадании курсора в два объекта выбирается объект с наибольшим значением zIndex. При дальнейшей работе - например при определении, фигура какого слоя является более приоритетной, используется значение zIndex слоя. Обратите внимание, что zIndex слоя будет определять взаимное расположение объектов слоя и одиночных объектов, добавленных в определенные пейны карты. Например, если вы хотите наложить хотспотные объекты поверх площадных объектов, стоит задать хотспотному слою zIndex=201. См. map.pane.Manager .

* Обязательный параметр/опция.

Пример:

```
// Создаем источник хотспотных данных. Не указываем значение ключа,
// поэтому имя функции-обработчика (padding jsonp запроса) будет генерироваться динамически.
var objectSource = new ymaps.hotspot.ObjectSource('tiles/%c'),
    hotspotLayer = new ymaps.hotspot.Layer(objectSource, {
      zIndex: 100,
      // Разрешаем показывать балун без содержимого.
      showEmptyBalloon: true
    });
```

Поля

Имя	Тип	Описание
balloon	hotspot.layer.Balloon	Балун хотспотного слоя.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
hint	hotspot.layer.Hint	Хинт хотспотного слоя.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
addtomap	Слой добавлен на карту.
balloonclose	Заккрытие балуна. Экземпляр класса Event .

Имя	Описание
balloonopen	Открытие балуна на хотспотном слое. Экземпляр класса Event .
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
hintclose	Закрытие хинта. Экземпляр класса Event .
hintopen	Открытие хинта на хотспотном слое. Экземпляр класса Event .

Имя	Описание
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Имена полей, доступных через метод <code>IMultiTouchEvent.get</code>:</p> <ul style="list-style-type: none"> • <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Имена полей, доступных через метод <code>IMultiTouchEvent.get</code>:</p> <ul style="list-style-type: none"> <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие; <code>coords</code> - географические координаты точки, в которой произошло событие; <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Имена полей, доступных через метод <code>IMultiTouchEvent.get</code>:</p> <ul style="list-style-type: none"> <code>activeObject</code> (hotspot.layer.Object) — объект слоя, на котором произошло событие; <code>coords</code> - географические координаты точки, в которой произошло событие; <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. <p>Унаследовано от IChild.</p>
removefrommap	Слой удален с карты.
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> activeObject (hotspot.layer.Object) — объект слоя, на котором произошло событие; coords - географические координаты точки, в которой произошло событие; globalPixels - координаты события в глобальных пикселях от левого верхнего угла мира; pagePixels - координаты события в пикселях от левого верхнего угла страницы; clientPixels - координаты события в пикселях от левого верхнего угла окна браузера; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту.
getObjectInPosition(coords)	vow.Promise	Метод, позволяющий получить объект слоя в указанной географической точке на текущем масштабе. Если данные для тайла, в который попадает точка, не загружены, вызов метода спровоцирует их загрузку. После этого поиск будет производиться по загруженным данным.

Имя	Возвращает	Описание
getObjectsInPosition(coords)	vow.Promise	Метод, позволяющий получить объект слоя в указанной географической точке на текущем масштабе. Если данные для тайла, в который попадает точка, не подгружены, вызов метода спровоцирует их загрузку. После этого поиск будет производиться по загруженным данным.
getObjectSource()	IHotspotObjectSource	Возвращает источник объектов слоя активных областей.
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .
update()		Обновляет слой активных областей. После выполнения этой команды загруженные ранее объекты удаляются из контейнера и запрашиваются новые данные.

Описание полей

balloon

```
{hotspot.layer.Balloon} balloon
```

Балун хотспотного слоя.

hint

```
{hotspot.layer.Hint} hint
```

Хинт хотспотного слоя.

Описание событий

addtomap

Слой добавлен на карту.

balloonclose

Закрытие балуна. Экземпляр класса [Event](#).

balloonopen

Открытие балуна на хотспотном слое. Экземпляр класса [Event](#).

click

Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

contextmenu

Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

dblclick

Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

hintclose

Закрытие хинта. Экземпляр класса [Event](#).

hintopen

Открытие хинта на хотспотном слое. Экземпляр класса [Event](#).

mousedown

Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mouseenter

Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mouseleave

Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mousemove

Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mouseup

Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса Event. Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

multitouchend

Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent` с информацией о прикосновениях. Имена полей, доступных через метод `IMultiTouchEvent.get`:

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

multitouchmove

Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent` с информацией о прикосновениях. Имена полей, доступных через метод `IMultiTouchEvent.get`:

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

multitouchstart

Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent` с информацией о прикосновениях. Имена полей, доступных через метод `IMultiTouchEvent.get`:

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

removefrommap

Слой удален с карты.

wheel

Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса Event. Имена полей, доступных через метод [Event.get](#):

- `activeObject` ([hotspot.layer.Object](#)) — объект слоя, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

Описание методов

getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

getObjectInPosition

```
{vow.Promise} getObjectInPosition(coords)
```

Метод, позволяющий получить объект слоя в указанной географической точке на текущем масштабе. Если данные для тайла, в который попадает точка, не подгружены, вызов метода спровоцирует их загрузку. После этого поиск будет производиться по загруженным данным.

Возвращает Promise, который будет разрешен объектом слоя, в который попадает заданная точка. Если точка попадает в несколько объектов слоя, вернется объект с максимальным значением `zIndex`.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coords</code> *	—	Тип: <code>Number[]</code> Географические координаты точки.

* Обязательный параметр/опция.

getObjectsInPosition

```
{vow.Promise} getObjectsInPosition(coords)
```

Метод, позволяющий получить объект слоя в указанной географической точке на текущем масштабе. Если данные для тайла, в который попадает точка, не подгружены, вызов метода спровоцирует их загрузку. После этого поиск будет производиться по загруженным данным.

Возвращает Promise, который будет разрешен массивом объектов слоя, в которые попадает заданная точка.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coords</code> *	—	Тип: Number[] Географические координаты точки.

* Обязательный параметр/опция.

`getObjectSource`

```
{IHotspotObjectSource} getObjectSource()
```

Возвращает источник объектов слоя активных областей.

Пример:

```
// Обновление источника данных для слоя активных областей.  
hotspotLayer.getObjectSource().setTileUrlTemplate('newSource/?%c');  
hotspotLayer.update();
```

`update`

```
{ } update()
```

Обновляет слой активных областей. После выполнения этой команды загруженные ранее объекты удаляются из контейнера и запрашиваются новые данные.

`hotspot.ObjectSource`

Расширяет [IHotspotObjectSource](#).

Стандартная реализация интерфейса [IHotspotObjectSource](#). Работает со стандартным форматом ответа сервера.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
hotspot.ObjectSource(tileUrlTemplate[, keyTemplate[, options]])
```

Возвращает источник данных для слоя активных областей.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tileUrlTemplate</code> *	—	Тип: String Function Возвращает шаблон URL данных для тайла. Поддерживаются специальные конструкции и возможность использования функции по аналогии с Layer .

Параметр	Значение по умолчанию	Описание
<code>keyTemplate</code>	—	<p>Тип: String Function</p> <p>Шаблон идентификатора тайла. Используется для формирования padding в JSONP-запросе за данными тайла. Задается по аналогии с предыдущим параметром. Все символы, не являющиеся буквой, цифрой или символом '_', будут заменены на '_'. Если параметр не задан, то padding jsonp-запроса будет сгенерирован автоматически. Примеры преобразований при tileNumber=[3, 1], zoom=9:</p> <ul style="list-style-type: none"> 'myCallback=%x' => 'myCallback_3' '%c' => 'x_3_y_1_z_9' 'callback2_%c' => 'callback2_x_3_y_1_z_9' 'callback%test' => 'callback_test'
<code>options</code>	—	<p>Тип: Object</p> <p>Опции источника данных.</p>
<code>options.bounds</code>	null	<p>Тип: Bounds</p> <p>Область на карте, для которой есть данные, или null, если данные есть везде. Опция для стандартной реализации метода restrict.</p>
<code>options.maxZoom</code>	0	<p>Тип: Bounds</p> <p>Максимальный уровень масштабирования, для которого делаются запросы за данными. Опция для стандартной реализации метода restrict.</p>
<code>options.minZoom</code>	0	<p>Тип: Bounds</p> <p>Минимальный уровень масштабирования, для которого делаются запросы за данными. Опция для стандартной реализации метода restrict.</p>
<code>options.noCache</code>	false	<p>Тип: Boolean</p> <p>false — использовать встроенный кэш браузера, true — не использовать (к каждому URL будет автоматически приписываться случайный GET-параметр, чтобы исключить кэширование тайлов браузером).</p>

* Обязательный параметр/опция.

Пример:

```
// Пример стандартного формата ответа сервера.
hotspot_callback({
  // Ответ может содержать поле "data" или "error".
  "data": {
    "type": "FeatureCollection",
    // Массив активных областей.
    "features": [{
      "type": "Feature",
      // Данные объекта.
      // Передаются в конструктор hotspot.layer.Object в качестве данных.
      "properties": {
        "hintContent": "Содержимое всплывающей подсказки.",
        "balloonContentBody": "Содержимое балуна.",
        "balloonContentHeader": "Заголовок балуна.",
        "balloonContentFooter": "Нижняя часть балуна.",
        // Можно задавать свойство balloonContent вместо Body/Header/Footer

        // Обязательное поле, описывающее hotspot.layer.Object.
        "HotspotMetaData": {
          // Идентификатор активной области.
          // hotspot.layer.Object.getId по умолчанию возвращает значение этого поля.
          "id": 10469893,
          "zIndex": 10,
          // Данные, на основе которых создается геометрия hotspot.layer.Object.
          // Обязательное поле.
          // Стандартная реализация позволяет передавать следующие типы геометрий:
          // "Rectangle" - прямоугольник.
          // Будет создана геометрия geometry.pixel.Rectangle.
          // "Polygon" - многоконтурный многоугольник.
          // Будет создана геометрия geometry.pixel.Polygon.
          // "MultiPolygon" - сложная фигура, состоящая из нескольких многоконтурных многоугольников.
          // Будет создана геометрия geometry.pixel.MultiPolygon.
          // "ConvexPolygon"
          // - многоконтурный многоугольник. Все контуры должны быть выпуклыми. Все контуры должны быть выпуклыми.
          // Работает быстрее, чем "Polygon".
          // Будет создана геометрия geometry.pixel.Polygon.
          // "MultiConvexPolygon"
          // - сложная фигура, состоящая из нескольких многоконтурных многоугольников.
          // Все контуры должны быть выпуклыми. Работает быстрее, чем "MultiPolygon".
          // Будет создана геометрия geometry.pixel.MultiPolygon.
          "RenderedGeometry": {
            "type": "Polygon",

            // Координаты передаются в пиксельных координатах, отсчитываемых от верхнего левого угла тайла.
            "coordinates": [
              // Первый контур многоугольника.
              [
                [-315, 280], [32, 442], [141, 208], [-206, 46], [-315, 280]
              ],
              // Второй контур многоугольника.
              [
                [-186, 155], [-238, 265], [-152, 306], [-100, 196], [-186, 155]
              ]
            ]
          }
        }
      },
      // Фактическая геометрия объекта.
      // Необязательное поле;
      // Передается в hotspot.layer.Object и доступна через метод hotspot.layer.Object.getGeometry.
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          // Первый контур многоугольника.
          [
            [29.176096525, 40.904183940],
            [29.177027467, 40.903854324],
            [29.177319900, 40.904329679],
            [29.176389040, 40.904659406],
            [29.176096525, 40.904183940]
          ],
          // Второй контур многоугольника.
          [
            [29.176442530, 40.904437683],
            [29.176303055, 40.904213830],
            [29.176533525, 40.904131168],
            [29.176673032, 40.904355010],
            [29.176442530, 40.904437683]
          ]
        ]
      }
    ]
  },
  {
    "type": "Feature",
    "properties": {
      // Описание следующей активной области.
    }
  }
});
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .

Методы

Имя	Возвращает	Описание
cancelLastRequest()		Отменяет последний запрос за данными. Унаследован от IHotspotObjectSource .
getKey(tileNumber, zoom)	String	Возвращает имя callback-функции (padding) в JSONP-запросе, если оно статическое, либо null, если можно использовать динамический JSONP-callback. Шаблоны поддерживают те же подстановки, что и в шаблоне url тайла. Все символы, не являющиеся буквой, цифрой или символом '_', будут заменены на '_'.
getKeyTemplate()	String	Возвращает шаблон идентификатора тайла.
getTileUrl(tileNumber, zoom)	String	Возвращает URL данных для конкретного тайла.
getTileUrlTemplate()	String	Возвращает шаблон URL данных для тайла.
parseResponse(layer, res, callback, tileNumber, zoom)		Разбирает ответ сервера. Возвращает в callback массив объектов-активных областей, сформированных по JSON-описанию.

Имя	Возвращает	Описание
<code>requestObjects(layer, tileNumber, zoom, callback)</code>		Строит массив объектов <code>IHotspotLayerObject</code> , соответствующих определенному слою, тайлу и коэффициенту масштабирования карты, и передает его в функцию <code>callback</code> . Унаследован от <code>IHotspotObjectSource</code> .
<code>restrict(layer, tileNumber, zoom)</code>	Boolean	Метод, который вызывается перед тем, как отправить запрос за данными для тайла. Если метод возвращает <code>true</code> , то запрос на сервер посылаться не будет, и в качестве ответа вернется пустой массив объектов. Стандартная реализация метода проверяет, лежит ли параметр <code>zoom</code> в пределах <code>[options.minZoom, options.maxZoom]</code> . А также проверяет, попадает ли центр запрашиваемого тайла в указанные <code>options.bounds</code> . Если <code>options.bounds=null</code> , то проверка для этого параметра не осуществляется.
<code>setKeyTemplate(template)</code>		Задаёт новый шаблон идентификатора тайла.
<code>setTileUrlTemplate(template)</code>		Задаёт новый шаблон URL данных для тайла.

Описание методов

getKey

```
{String} getKey(tileNumber, zoom)
```

Возвращает имя callback-функции (padding) в JSONP-запросе, если оно статическое, либо `null`, если можно использовать динамический JSONP-callback. Шаблоны поддерживают те же подстановки, что и в шаблоне url тайла. Все символы, не являющиеся буквой, цифрой или символом '_', будут заменены на '_'.

Возвращает идентификатор тайла. Используется для формирования padding в JSONP-запросе за данными тайла.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tileNumber *</code>	—	Тип: Number[] Номер тайла (тайловые координаты).
<code>zoom *</code>	—	Тип: Integer Уровень масштабирования.

* Обязательный параметр/опция.

getKeyTemplate

```
{String} getKeyTemplate()
```

Возвращает шаблон идентификатора тайла.

getTileUrl

```
{String} getTileUrl(tileNumber, zoom)
```

Возвращает URL данных для конкретного тайла.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tileNumber *</code>	—	Тип: Number[] Номер тайла (тайловые координаты).
<code>zoom *</code>	—	Тип: Integer Уровень масштабирования.

* Обязательный параметр/опция.

Пример:

```
var hotspotObjectSource = new ymaps.hotspot.ObjectSource('dataSource/?%c');
hotspotObjectSource.getTileUrl = function (tileNumber, zoom) {
    if (zoom > 10) {
        // Для крупных масштабов используем выставленный путь к данным.
        return ymaps.hotspot.ObjectSource.prototype.call(this, tileNumber, zoom);
    } else {
        // Для мелких масштабов используем другой путь.
        return 'otherSource/getHotspots.xml?z=' + zoom + '<codeph>&</codeph>x=' + tileNumber[0] + '<codeph>&</codeph>y=' + tileNumber[1];
    }
};
```

getTileUrlTemplate

```
{String} getTileUrlTemplate()
```

Возвращает шаблон URL данных для тайла.

parseResponse

```
{ } parseResponse(layer, res, callback, tileNumber, zoom)
```

Разбирает ответ сервера. Возвращает в callback массив объектов-активных областей, сформированных по JSON-описанию.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>layer *</code>	—	Тип: <code>hotspot.Layer</code> Слой, которому принадлежат объекты.
<code>res *</code>	—	Тип: <code>Object</code> Ответ сервера.
<code>callback *</code>	—	Тип: <code>Function</code> Функция-обработчик.
<code>tileNumber *</code>	—	Тип: <code>Number[]</code> Номер тайла, для которого пришел ответ.
<code>zoom *</code>	—	Тип: <code>Number</code> Коэффициент масштабирования, для которого пришел ответ. массива объектов.

* Обязательный параметр/опция.

restrict

```
{Boolean} restrict(layer, tileNumber, zoom)
```

Метод, который вызывается перед тем, как отправить запрос за данными для тайла. Если метод возвращает `true`, то запрос на сервер посылаться не будет, и в качестве ответа вернется пустой массив объектов. Стандартная реализация метода проверяет, лежит ли параметр `zoom` в пределах `[options.minZoom, options.maxZoom]`. А также проверяет, попадает ли центр запрашиваемого тайла в указанные `options.bounds`. Если `options.bounds=null`, то проверка для этого параметра не осуществляется.

Возвращает `true` - тайл вышел за пределы области данных (для данного тайла данных нет), `false` - не вышел (данные есть).

Параметры:

Параметр	Значение по умолчанию	Описание
<code>layer *</code>	—	Тип: <code>hotspot.Layer</code> Слой активных областей.

Параметр	Значение по умолчанию	Описание
<code>tileNumber *</code>	—	Тип: Number[] Номер тайла.
<code>zoom *</code>	—	Тип: Integer Уровень масштабирования.

* Обязательный параметр/опция.

Пример:

```
// Пример переопределения метода "restrict"
// Предположим, что есть только данные по Мурманску и Новосибирску.
var myMap = new ymaps.Map('map', {center: [32.5, 68.9], zoom: 9});
var geoBounds = [
  [[31.729958, 69.369182], [34.203324, 68.666473]], // Murmansk
  [[82.179084, 55.341085], [83.725642, 54.670738]] // Novosibirsk
];
var projection = myMap.options.get('projection');
var myHotspotSource = new ymaps.hotspot.ObjectSource('http://www.myDomain.ru/tiles/?%c', '%c');

myHotspotSource.restrict = function(layer, tileNumber, zoom) {
  // Вычисление пиксельных границ городов для данного уровня масштабирования.
  var boundsFromPoints = ymaps.util.bounds.fromPoints;
  var toGlobalPixels = projection.toGlobalPixels;
  var pixelBounds = [
    boundsFromPoints(
      toGlobalPixels(geoBounds[0][0], zoom),
      toGlobalPixels(geoBounds[0][1], zoom)
    ),
    boundsFromPoints(
      toGlobalPixels(geoBounds[1][0], zoom),
      toGlobalPixels(geoBounds[1][1], zoom)
    )
  ];
  // Вычисление границ пикселей плитки
  var leftTop = [tileNumber[0] * 256, tileNumber[1] * 256];
  var tileBounds = [leftTop, [leftTop[0] + 256, leftTop[1] + 256]];
  var intersects = ymaps.util.bounds.intersects;
  // Если границы пикселей тайла пересекаются с границами пикселей указанных областей,
  // мы должны отправить запрос на получение данных.
  if (intersects(pixelBounds[0], tileBounds) || (intersects(pixelBounds[1], tileBounds))) {
    return false;
  }

  // Для всех остальных тайлов этот источник не содержит никаких данных.
  return true;
}
```

setKeyTemplate

```
{ } setKeyTemplate(template)
```

Задаёт новый шаблон идентификатора тайла.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>template *</code>	—	Тип: String Шаблон идентификатора.

* Обязательный параметр/опция.

setTileUrlTemplate

```
{ } setTileUrlTemplate(template)
```


Задаёт новый шаблон URL данных для тайла.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>template</code> *	—	Тип: String Шаблон URL

* Обязательный параметр/опция.

Hotspot

Расширяет [IHotspot](#).

Активная область.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Hotspot(shape[, zIndex])
```

Активная область.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>shape</code> *	—	Тип: IShape Фигура активной области.
<code>zIndex</code>	0	Тип: Number zIndex активной области.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .

События

Имя	Описание
<code>click</code>	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getShape()	IShape	<p>Возвращает фигуру активной области.</p> <p>Унаследован от IHotspot.</p>
getZIndex()	Number	<p>Возвращает <code>zIndex</code> активной области.</p> <p>Унаследован от IHotspot.</p>
setShape(shape)		<p>Задаёт фигуру активной области.</p> <p>Унаследован от IHotspot.</p>
setZIndex(zIndex)		<p>Задаёт <code>z-index</code> активной области.</p> <p>Унаследован от IHotspot.</p>

interactivityModel

interactivityModel.storage

Статический объект.

Экземпляр класса [util.Storage](#)

Хранилище моделей интерактивности. Модели интерактивности позволяют объектам по-разному обрабатывать DOM-события. Список доступных ключей моделей интерактивности:

- 'default#opaque' - объект генерирует все DOM-события и не прокидывает их на карту. Поведения карты не будут работать при наведении или клике на объекты с данной моделью интерактивности.
- 'default#geoObject' - объект генерирует все DOM-события. На карту прокидываются события 'wheel', 'mousedown', 'dblclick', 'contextmenu', 'multitouchstart', 'multitouchmove' и 'multitouchend'. Если на карте включены поведения 'scrollZoom', 'dblClickZoom' или 'magnifier', они будут работать через объекты с данной моделью интерактивности, в отличие от объектов с моделью 'default#opaque'.
- 'default#layer' - объект генерирует все DOM-события. На карту прокидываются события 'wheel', 'mousedown', 'contextmenu', 'multitouchstart', 'multitouchmove' и 'multitouchend'. Если на карте включены поведения 'scrollZoom', 'drag' или 'magnifier', они будут работать через объекты с данной моделью интерактивности.
- 'default#transparent' - объект генерирует все DOM-события, а затем прокидывает их на карту.
- 'default#silent' - объект не генерирует DOM-события, но прокидывает их на карту.

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.

layer

layer.storage

Статический объект.

Экземпляр класса [util.Storage](#)

Хранилище слоев.

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.

Имя	Возвращает	Описание
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.

layer.tile

layer.tile.CanvasTile

Расширяет [ICanvasTile](#).

Картиночный canvas-тайл. Умеет отрисовывать указанное изображение через метод drawImage у 2d контекста canvas-элемента.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
layer.tile.CanvasTile(url[, options[, renderOptions]])
```

Создаёт картиночный canvas-тайл. Доступен по ключу "default#canvas" в хранилище классов тайлов.

Параметры:

Параметр	Значение по умолчанию	Описание
url *	—	Тип: String URL изображения.
options	—	Тип: Object Опции.
options.notFoundTile	null	Тип: String null Опция, указывающая по какому url нужно загрузить картинку, если картинка для тайла не загрузилась. Если значение null, то будет показан стандартный тайл с текстовым сообщением. Для прозрачных тайлов опция notFoundTile не применяется, и вместо незагруженных тайлов ничего показываться не будет.
options.tileAnimationDuration	—	Тип: Number Продолжительность анимации прозрачности изображения. Значение по умолчанию зависит от браузера.
renderOptions	—	Тип: Object Параметры рендеринга.

Параметр	Значение по умолчанию	Описание
<code>renderOptions.tileNumber</code>	—	Тип: Number[]
<code>renderOptions.tileZoom</code>	—	Тип: Number

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	<code>IEventManager</code>	Менеджер событий. Унаследовано от <code>IEventEmitter</code> .

События

Имя	Описание
<code>ready</code>	Событие готовности тайла. Унаследовано от <code>ITile</code> .

Методы

Имя	Возвращает	Описание
<code>destroy()</code>		Уничтожает тайл. Унаследован от <code>ITile</code> .
<code>isReady()</code>	Boolean	Проверяет готовность тайла. Унаследован от <code>ITile</code> .
<code>renderAt(context, canvasSize, bounds[, animate])</code>		Отрисовывает картиночный тайл на 2d-контексте объекта canvas. Унаследован от <code>ICanvasTile</code> .

layer.tile.DomTile

Расширяет `IDomTile`.

Картиночный DOM-тайл. Умеет отрисовывать указанное изображение через css свойство background у DOM-элемента.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
layer.tile.DomTile(url[, options[, renderOptions]])
```

Создаёт картиночный dom-тайл. Доступен по ключу "default#dom" в хранилище классов тайлов. Создаёт картиночный dom-тайл.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>url *</code>	—	Тип: String URI изображения

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции.
options.notFoundTile	null	Тип: String null Опция, указывающая по какому url нужно загрузить картинку, если картинка для тайла не загрузилась. Если значение null, то будет показан стандартный тайл с текстовым сообщением. Для прозрачных тайлов опция notFoundTile не применяется, и вместо незагруженных тайлов ничего показываться не будет.
options.tileAnimationDuration	—	Тип: Number Продолжительность анимации прозрачности изображения при рисовании в мс (применяется только в браузерах, поддерживающих CSS Transition для свойства opacity). Значение по умолчанию зависит от браузера.
renderOptions	—	Тип: Object Параметры рендеринга.
renderOptions.tileNumber	—	Тип: Number[]
renderOptions.tileZoom	—	Тип: Number

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
ready	Событие готовности тайла. Унаследовано от ITile .

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает тайл. Унаследован от ITile .

Имя	Возвращает	Описание
<code>isReady()</code>	Boolean	Проверяет готовность тайла. Унаследован от ITile .
<code>renderAt(context, clientBounds, animate)</code>		Добавляет тайл в родительский HTML-элемент. Унаследован от IDomTile .

layer.tileContainer

layer.tileContainer.CanvasContainer

Расширяет [IChildOnMap](#).

Контейнер тайлов на canvas.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
layer.tileContainer.CanvasContainer(layer[, options])
```

Создаёт контейнер тайлов на canvas. Доступен по ключу "default#canvas" в хранилище классов тайловых контейнеров.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>layer *</code>	—	Тип: ILayer Слой.
<code>options</code>	—	Тип: Object Опции контейнера.
<code>options.notFoundTile</code>	null	Тип: String null Опция, указывающая по какому url нужно загрузить картинку, если картинка для тайла не загрузилась. Если значение null, то будет показан стандартный тайл с текстовым сообщением. Для прозрачных тайлов опция notFoundTile не применяется, и вместо незагруженных тайлов ничего показываться не будет.
<code>options.tileClass</code>	'default#canvas'	Тип: ICanvasTile Класс тайлов, используемых контейнером. Должен реализовывать интерфейс ICanvasTile .

Параметр	Значение по умолчанию	Описание
options.tileTransparent	false	Тип: Boolean Флаг, показывающий, являются ли тайлы контейнера прозрачными.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту.
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getTile(tileNumber, tileZoom, priority)	ICanvasTile	Фабричная функция создания тайлов.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание методов

getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

getTile

```
{ICanvasTile} getTile(tileNumber, tileZoom, priority)
```

Фабричная функция создания тайлов.

Возвращает экземпляр тайла.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tileNumber</code> *	—	Тип: Number[] Номер тайла.
<code>tileZoom</code> *	—	Тип: Number Масштаб тайла.
<code>priority</code> *	—	Тип: Number Приоритет загрузки.

* Обязательный параметр/опция.

layer.tileContainer.DomContainer

Расширяет [IChildOnMap](#).

Контейнер тайлов типа IDomTile

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
layer.tileContainer.DomContainer(layer[, options])
```

Создаёт контейнер DOM-тайлов. Доступен по ключу "default#dom" в хранилище классов тайловых контейнеров.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>layer</code> *	—	Тип: ILayer Слой.
<code>options</code>	—	Тип: Object Опции контейнера.

Параметр	Значение по умолчанию	Описание
options.notFoundTile	null	Тип: String null Опция, указывающая по какому url нужно загрузить картинку, если картинка для тайла не загрузилась. Если значение null, то будет показан стандартный тайл с текстовым сообщением. Для прозрачных тайлов опция notFoundTile не применяется, и вместо незагруженных тайлов ничего показываться не будет.
options.tileClass	'default#dom'	Тип: IDomTile Класс тайлов, используемых контейнером. Должен реализовывать интерфейс IDomTile .
options.tileTransparent	false	Тип: Boolean Флаг, показывающий, являются ли тайлы контейнера прозрачными.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
ready	Событие готовности всех тайлов.

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту.

Имя	Возвращает	Описание
<code>getParent()</code>	<code>IParentOnMap</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>getTile(tileNumber, tileZoom, priority)</code>	<code>IDomTile</code>	Фабричная функция создания тайлов.
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание событий

ready

Событие готовности всех тайлов.

Описание методов

getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

getTile

```
{IDomTile} getTile(tileNumber, tileZoom, priority)
```

Фабричная функция создания тайлов.

Возвращает экземпляр тайла.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tileNumber *</code>	—	Тип: Number[] Номер тайла.
<code>tileZoom *</code>	—	Тип: Number Масштаб тайла.
<code>priority *</code>	—	Тип: Number Приоритет загрузки.

* Обязательный параметр/опция.

Layer

Расширяет [ILayer](#), [IParentOnMap](#), [IPositioningContext](#).

Тайловый слой. Позволяет отображать на карте подложку, состоящую из тайлов.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Layer(tileUrlTemplate[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
tileUrlTemplate *	—	<p>Тип: String Function</p> <p>Строковый шаблон URL тайла, либо функция, которая генерирует URL тайла. Для строкового шаблона поддерживается подстановка:</p> <ul style="list-style-type: none">• %с заменяется на x=номер[0]&y=номер[1]&z=уровень масштабирования.• %x — заменяется на номер[1];• %y — заменяется на номер[1];• %z — заменяется на уровень масштабирования;• %l — заменяется на lang=язык;• %d или %d n- заменяется на число от 1 до n, в зависимости от номера тайла; n - количество доменов. Используется для распределения нагрузки между несколькими доменами. В качестве n нужно указывать степени двойки (2, 4, 16 и т.д.). Если шаблон имеет %d, то n=4. <p>Функция-шаблон получает на вход три параметра:</p> <ul style="list-style-type: none">• tileNumber - массив из двух чисел, номера тайла по x и y;• tileZoom - уровень масштабирования;• Возвращает строку-URL.

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.brightness</code>	0.5	Тип: Number Яркость слоя. Задается числом от 0 до 1. 0 соответствует черному цвету, 1 - белому.
<code>options.notFoundTile</code>	null	Тип: String null Опция, указывающая по какому url нужно загрузить картинку, если картинка для тайла не загрузилась. Если значение null, то будет показан стандартный тайл с текстовым сообщением. Для прозрачных тайлов опция notFoundTile не применяется, и вместо незагруженных тайлов ничего показываться не будет.
<code>options.pane</code>	'ground'	Тип: IPane string Указатель на пейн слоя или ключ из map.pane.Manager .
<code>options.projection</code>	—	Тип: Object Проекция слоя.
<code>options.tileSize</code>	[256, 256]	Тип: Number[] Размер тайлов слоя.
<code>options.tileTransparent</code>	false	Тип: Boolean Флаг, показывающий, являются ли тайлы слоя прозрачными.
<code>options.zIndex</code>	constants.zIndex.layer	Тип: Number zIndex слоя в контейнере слоёв.

* Обязательный параметр/опция.

Пример:

```
// Добавляет на карту слой OSM.
map.layers.add(new ymaps.Layer('http://tile.openstreetmap.org/%z/%x/%y.png', {
  projection: ymaps.projection.sphericalMercator
}));
map.copyrights.add('© OpenStreetMap contributors, CC-BY-SA');
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
brightnesschange	Событие изменения яркости слоя. Унаследовано от ILayer .
copyrightschange	Событие изменения доступной информации об авторских правах. Унаследовано от ILayer .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IParentOnMap .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .
tileloadchange	Событие изменения состояния загрузки тайла. Поля данных: <ul style="list-style-type: none"> <code>readyTileNumber</code>-количество готовых тайлов. Тайл считается готовым, когда она загружен и отрисован. Тип: <code>Number</code>. <code>totalTileNumber</code>-общее количество видимых тайлов. Тип: <code>Number</code>. Унаследовано от ILayer .
zoomrangechange	Событие изменения доступной информации о диапазоне уровней масштабирования. Унаследовано от ILayer .

Методы

Имя	Возвращает	Описание
clientPixelsToNumber(clientPixelPoint, tileZoom)	Number[]	Возвращает номер тайла, в который попадает указанная точка при заданном уровне масштаба тайлов.
fromClientPixels(clientPixelPoint)	Number[]	Преобразует клиентские пиксельные координаты в глобальные. Унаследован от IPositioningContext .
getBrightness()	Number	Необязательный метод. Унаследован от ILayer .
getCopyrights(coords, zoom)	vow.Promise	Необязательный метод. Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Унаследован от ILayer .
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getPane()	IPane	Возвращает контейнер, в котором находится слой.
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getTileSize(zoom)	Number[]	Возвращает размеры тайла по горизонтали и вертикали для указанного коэффициента масштабирования.
getTileStatus()	Object	Возвращает общее число видимых тайлов и число готовых. Тайл считается готовым, когда она загружен и отрисован.
getTileUrl(tileNumber, tileZoom)	String null	Возвращает URL тайла по его номеру и уровню масштабирования или null если для запрашиваемого участка нет данных.

Имя	Возвращает	Описание
getTileUrlTemplate()	String Function	Возвращает строковый шаблон URL тайла или функцию, которая его генерирует.
getZoom()	Number	Возвращает текущий уровень масштабирования, в котором работает контекст позиционирования. Унаследован от IPositioningContext .
getZoomRange(point)	vow.Promise	Необязательный метод. Определяет доступный в указанной точке диапазон уровней масштабирования. Если данные есть, то возвращенный объект-promise будет подтвержден и передаст в качестве результата массив из двух чисел - минимальный и максимальный доступный в точке уровень масштабирования. Если данных нет, то promise отклоняется с указанием ошибки. Унаследован от ILayer .
numberToClientBounds(tileNumber, tileZoom)	Number[][]	Преобразует номер и уровень масштаба тайла в занимаемую тайлом область в клиентских координатах родительского контейнера.
restrict(number, tileZoom)	Integer[] null	Накладывает ограничения на зону видимости тайлов (в том числе за цикленность карты по осям x, y).
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .
setTileUrlTemplate(tileUrlTemplate)		
toClientPixels(globalPixelPoint)	Number[]	Преобразует глобальные пиксельные координаты в клиентские. Унаследован от IPositioningContext .

Имя	Возвращает	Описание
<code>update()</code>		Удаляет старые тайлы и запрашивает новые.

Описание методов

`clientPixelsToNumber`

```
{Number[]} clientPixelsToNumber(clientPixelPoint, tileZoom)
```

Возвращает номер тайла, в который попадает указанная точка при заданном уровне масштаба тайлов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>clientPixelPoint</code> *	—	Тип: Number Точка в клиентских пиксельных координатах.
<code>tileZoom</code> *	—	Тип: Number Уровень масштабирования тайла.

* Обязательный параметр/опция.

`getPane`

```
{IPane} getPane()
```

Возвращает контейнер, в котором находится слой.

`getTileSize`

```
{Number[]} getTileSize(zoom)
```

Возвращает размеры тайла по горизонтали и вертикали для указанного коэффициента масштабирования.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>zoom</code> *	—	Тип: Number Значение коэффициента масштабирования.

* Обязательный параметр/опция.

Пример:

```
// Покажем тайлы для большего уровня масштабирования,  
// растянутые в два раза до размера 512x512 пикселей.  
// Например, для экономии трафика.  
var layer = new ymaps.Layer(' ', {  
  projection: ymaps.projection.sphericalMercator  
});  
layer.getTileUrl = function (tileNumber, zoom) {
```

```
return [
  'http://tile.openstreetmap.org',
  Math.max(zoom - 1, 0), tileNumber[0], tileNumber[1]
].join('/') + '.png';
}
layer.getTileSize = function (zoom) {
  if (zoom == 0) {
    return [256, 256];
  }
  return [512, 512];
}
map.copyrights.add('© OpenStreetMap contributors, CC-BY-SA');
```

getTileStatus

```
{Object} getTileStatus()
```

Возвращает общее число видимых тайлов и число готовых. Тайл считается готовым, когда она загружен и отрисован.

Возвращает объект, содержащий следующие поля:

- readyTileNumber-количество готовых тайлов. Тип: Number.
- totalTileNumber - общее количество тайлов. Тип: Number.

getTileUrl

```
{String|null} getTileUrl(tileNumber, tileZoom)
```

Возвращает URL тайла по его номеру и уровню масштабирования или null если для запрашиваемого участка нет данных.

Параметры:

Параметр	Значение по умолчанию	Описание
tileNumber *	—	Тип:
tileZoom *	—	Тип:

* Обязательный параметр/опция.

Пример:

```
// Определяет функцию для формирования URL тайла
var layer = new ymaps.Layer('');
layer.getTileUrl = function (tileNumber, zoom) {
  return [
    'http://tile.openstreetmap.org',
    zoom, tileNumber[0], tileNumber[1]
  ].join('/') + '.png';
}
```

getTileUrlTemplate

```
{String|Function} getTileUrlTemplate()
```

Возвращает строковый шаблон URL тайла или функцию, которая его генерирует.

numberToClientBounds

```
{Number[][]} numberToClientBounds(tileNumber, tileZoom)
```

Преобразует номер и уровень масштаба тайла в занимаемую тайлом область в клиентских координатах родительского контейнера.

Возвращает область в клиентских пиксельных координатах.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tileNumber *</code>	—	Тип: Integer[] Номер тайла.
<code>tileZoom *</code>	—	Тип: Integer Уровень масштабирования тайла.

* Обязательный параметр/опция.

restrict

```
{Integer[]|null} restrict(number, tileZoom)
```

Накладывает ограничения на зону видимости тайлов (в том числе зацикленность карты по осям x, y).

Возвращает новый номер тайла с учётом ограничений либо null, если тайл не входит в зону видимости.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>number *</code>	—	Тип: Integer[] Номер тайла.
<code>tileZoom *</code>	—	Тип: Integer Уровень масштабирования тайла.

* Обязательный параметр/опция.

setTileUrlTemplate

```
{ } setTileUrlTemplate(tileUrlTemplate)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tileUrlTemplate *</code>	—	Тип: String Function Строковый шаблон URL тайла, либо функция, которая его генерирует.

* Обязательный параметр/опция.

update

```
{ } update()
```

Удаляет старые тайлы и запрашивает новые.

Параметры:

Параметр	Значение по умолчанию	Описание
updateBounds *	—	Тип:

* Обязательный параметр/опция.

LayerCollection

Расширяет [ILayer](#), [IMapObjectCollection](#).

Коллекция слоев.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
LayerCollection([options])
```

Коллекция слоев.

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции слоя.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от ICollection .
brightnesschange	Событие изменения яркости слоя. Унаследовано от ILayer .
copyrightschange	Событие изменения доступной информации об авторских правах. Унаследовано от ILayer .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"><code>oldMap</code> - старая карта;<code>newMap</code> - новая карта. Унаследовано от IParentOnMap .

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection .
tileloadchange	Событие изменения состояния загрузки тайла. Поля данных: <ul style="list-style-type: none"> <code>readyTileNumber</code>-количество готовых тайлов. Тайл считается готовым, когда она загружен и отрисован. Тип: <code>Number</code>. <code>totalTileNumber</code>-общее количество видимых тайлов. Тип: <code>Number</code>. Унаследовано от ILayer .
zoomrangechange	Событие изменения доступной информации о диапазоне уровней масштабирования. Унаследовано от ILayer .

Методы

Имя	Возвращает	Описание
add(child)	LayerCollection	Добавляет в коллекцию дочерний объект.
each(callback[, context])		Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.
getBrightness()	<code>Number</code>	Возвращает яркость слоя в виде числа от 0 до 1.
getCopyrights([coords[, zoom]])	vow.Promise	Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Если точка и коэффициент масштабирования не указаны, будут использованы центр и коэффициент масштабирования карты.
getIterator()	Iterator	Возвращает итератор по коллекции. Унаследован от ICollection .

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getZoomRange([coords])	vow.Promise	Определяет доступный в указанной точке диапазон уровней масштабирования. Если данные есть, то возвращенный объект-promise будет подтвержден и передаст в качестве результата массив из двух чисел - минимальный и максимальный доступный в точке уровень масштабирования. Если данных нет, то promise отклоняется с указанием ошибки. Если у коллекции нет ни одного потомка, предоставляющего информацию о диапазоне уровней масштабирования, то promise будет отклонён с сообщением 'noProvider'.
remove(child)	LayerCollection	Удаляет из коллекции дочерний объект.
removeAll()	Collection	Удаляет все элементы коллекции.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание методов

add

```
{LayerCollection} add(child)
```

Добавляет в коллекцию дочерний объект.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>child *</code>	—	Тип: <code>ILayer string</code> Слой (ключ из хранилища <code>layer.storage</code> или экземпляр класса <code>ILayer</code>), который нужно добавить.

* Обязательный параметр/опция.

Пример:

```
// Допустим, мы хотим добавить в свою коллекцию несколько слоев.
var layerCollection = new ymaps.LayerCollection();
var customLayer = new ymaps.Layer('http://tile.openstreetmap.org/%z/%x/%y.png', {
  projection: ymaps.projection.sphericalMercator
});
// Слой можно задать ключом из layer.storage.
var satelliteLayer = 'yandex#satellite';
// Добавим слои в нашу коллекцию.
layerCollection
  .add(customLayer)
  .add(satelliteLayer);
```

each

```
{ } each(callback[, context])
```

Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>callback *</code>	—	Тип: <code>Function</code> Функция-обработчик.
<code>context</code>	—	Тип: <code>Object</code> Контекст исполнения функции.

* Обязательный параметр/опция.

getBrightness

```
{Number} getBrightness()
```

Возвращает яркость слоя в виде числа от 0 до 1.

getCopyrights

```
{vow.Promise} getCopyrights([coords[, zoom]])
```

Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Если точка и коэффициент масштабирования не указаны, будут использованы центр и коэффициент масштабирования карты.

Возвращает `Promise` объект, который будет подтверждён массивом строк или DOM-элементов с информацией об авторских правах.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coords</code>	—	Тип: Number[] Точка на карте, для которой запрашивается информация об авторских правах.
<code>zoom</code>	—	Тип: Number Уровень масштабирования, для которого запрашивается информация об авторских правах.

Пример:

```
// Пусть у нас есть сервис, умеющий по координатам
// и уровню масштабирования возвращать копирайты.
myLayer.getCopyrights = function (coords, zoom) {
  var deferred = ymaps.vow.defer();
  $.ajax('url/to/copyrights/provider?ll=' +
    (coords || map.getCenter()).join(',') + '&z=' +
    (zoom || map.getZoom()),
    function (res) {
      deferred.resolve(res || []);
    });
  return deferred.promise();
};
```

getZoomRange

```
{vow.Promise} getZoomRange([coords])
```

Определяет доступный в указанной точке диапазон уровней масштабирования. Если данные есть, то возвращенный объект-promise будет подтвержден и передаст в качестве результата массив из двух чисел - минимальный и максимальный доступный в точке уровень масштабирования. Если данных нет, то promise отклоняется с указанием ошибки. Если у коллекции нет ни одного потомка, предоставляющего информацию о диапазоне уровней масштабирования, то promise будет отклонён с сообщением 'noProvider'.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coords</code>	—	Тип: Number[] Координаты точки. Если не указана, берётся текущий центр карты.

Пример:

```
// Допустим, наш слой отрисован для масштабов 2-15 по всей земле.
myLayer.getZoomRange = function () {
  return ymaps.vow.resolve([2, 15]);
}
```

remove

```
{LayerCollection} remove(child)
```

Удаляет из коллекции дочерний объект.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>child *</code>	—	Тип: ILayer String Слой (строка-ключ из layer.storage или экземпляр класса ILayer), который нужно удалить.

* Обязательный параметр/опция.

removeAll

```
{Collection} removeAll()
```

Удаляет все элементы коллекции.

Возвращает ссылку на себя.

layout

layout.Image

Расширяет [ILayout](#).

Класс для создания макетов, содержащих картинку.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
layout.Image(data)
```

Создает экземпляр макета-картинки.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>data *</code>	—	Тип: ILayout Данные макета.
<code>data.options</code>	—	Тип: ILayout Опции макета.
<code>data.options.imageClipRect</code>	—	Тип: <code>Number[][]</code> Координаты отображаемой области исходного изображения в пикселях.

Параметр	Значение по умолчанию	Описание
data.options.imageHref	—	Тип: String URL графического файла.
data.options.imageOffset	—	Тип: Number[] Смещение картинки относительно точки привязки.
data.options.imageSize	—	Тип: Number[] Размеры слоя с картинкой.
data.options.shape	—	Тип: IShape Object null Фигура активной области. Может быть задана экземпляром класса, реализующего интерфейс IShape или JSON-описанием пиксельной геометрии иконки. Если не задана, то автоматически будет рассчитана прямоугольная фигура на основе размера и смещения картинки. Координаты геометрии фигуры отсчитываются от точки привязки.

* Обязательный параметр/опция.

Пример:

```
// Создадим круглую метку радиусом 20 пикселей.
var placemark = new ymaps.Placemark([59.936952, 30.343334], null, {
  iconLayout: 'default#image',
  iconImageHref: './images/roundImage.png',
  iconImageSize: [40, 40],
  iconImageOffset: [-20, -20],
  // Определим интерактивную область над картинкой.
  iconShape: {
    type: 'Circle',
    coordinates: [0, 0],
    radius: 20
  }
});
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
emptinesschange	<p>Изменение признака пустоты макета. Экземпляр класса Event.</p> <p>Унаследовано от ILayout.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
parentelementchange	<p>Изменение родительского элемента. Экземпляр класса Event.</p> <p>Унаследовано от ILayout.</p>
shapechange	<p>Изменение фигуры области охватывающей макет. Экземпляр класса Event.</p> <p>Унаследовано от ILayout.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
destroy()		<p>Деструктор. Вызывается при завершении работы с макетом.</p> <p>Унаследован от ILayout.</p>
getData()	Object	<p>Возвращает объект данных макета.</p> <p>Унаследован от ILayout.</p>

Имя	Возвращает	Описание
getParentElement()	HTMLElement	Возвращает родительский HTML-элемент. Унаследован от ILayout .
getShape()	IShape null	Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета. Унаследован от ILayout .
isEmpty()	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от ILayout .
setData(data)		Задаёт данные макета. Унаследован от ILayout .
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от ILayout .

layout.ImageWithContent

Расширяет [layout.Image](#).

Класс для создания макетов, состоящих из картинки и содержимого.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
layout.ImageWithContent(data)
```

Создаёт экземпляр макета-картинки с содержимым.

Параметры:

Параметр	Значение по умолчанию	Описание
data *	—	Тип: ILayout Данные макета.
data.options	—	Тип: ILayout Опции макета.

Параметр	Значение по умолчанию	Описание
data.options.contentLayout	—	Тип: Function String Макет содержимого. (Тип: конструктор объекта с интерфейсом ILayout или его ключ в хранилище).
data.options.contentOffset	—	Тип: Number[] Смещение слоя с содержимым относительно слоя с картинкой.
data.options.contentSize	—	Тип: Number[] Размеры слоя с содержимым.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты макета. Экземпляр класса Event . Унаследовано от ILayout .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
parentelementchange	<p>Изменение родительского элемента. Экземпляр класса Event.</p> <p>Унаследовано от ILayout.</p>
shapechange	<p>Изменение фигуры области охватывающей макет. Экземпляр класса Event.</p> <p>Унаследовано от ILayout.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
destroy()		<p>Деструктор. Вызывается при завершении работы с макетом.</p> <p>Унаследован от ILayout.</p>
getData()	Object	<p>Возвращает объект данных макета.</p> <p>Унаследован от ILayout.</p>
getParentElement()	HTMLElement	<p>Возвращает родительский HTML-элемент.</p> <p>Унаследован от ILayout.</p>
getShape()	IShape null	<p>Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.</p> <p>Унаследован от ILayout.</p>

Имя	Возвращает	Описание
<code>isEmpty()</code>	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от ILayout .
<code>setData(data)</code>		Задаёт данные макета. Унаследован от ILayout .
<code>setParentElement(parent)</code>		Добавляет макет в DOM-дерево. Унаследован от ILayout .

layout.PieChart

Расширяет [layout.templateBased.Base](#).

Макет в виде круговой диаграммы. Доступен в хранилище макетов по ключу 'default#pieChart'. Макет может использоваться как инструмент визуализации произвольных данных, так и в связке с другими визуальными компонентами API, такими как метки, кластеризатор, менеджер объектов.

Примечание: Так как диаграммы рисуются с использованием технологии SVG, данный макет не будет работать в браузерах, не поддерживающих SVG, в частности, IE8.

См.: [Placemark Clusterer](#) [ObjectManager](#) [RemoteObjectManager](#) [LoadingObjectManager](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
layout.PieChart(data)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>data *</code>	—	Тип: Object Данные макета.
<code>data.option.pieChartRadius</code>	$25 + 2 * \text{Math.log}(\text{sum})$	Тип: Number Function Радиус диаграммы в пикселях. Может задаваться числом либо функцией, принимающей в качестве параметров <code>properties</code> и <code>options</code> макета и возвращающей радиус диаграммы в виде числа. По умолчанию радиус определяется как $25 + 2 * \text{Math.log}(\text{sum})$ пикселей, где <code>sum</code> — сумма весов секторов;
<code>data.options</code>	—	Тип: IOptionManager Опции отрисовки макета.

Параметр	Значение по умолчанию	Описание
data.options.pieChartCaptionMaxWidth	200	Тип: Number Максимальный размер подписи (iconCaption), в пикселях.
data.options.pieChartCoreFillStyle	white	Тип: String Стиль заливки центральной части. Задается строкой, кодирующей цвет заливки;
data.options.pieChartCoreRadius	pieChartRadius - 15	Тип: Number Function Радиус центральной части макета, в которой отображается контент. Задается аналогично pieChartRadius — числом или функцией. По умолчанию принимает значение на 15 пикселей меньше pieChartRadius;
data.options.pieChartStrokeStyle	white	Тип: String Стиль линий-разделителей секторов и внешней обводки диаграммы. Задается строкой, кодирующей цвет линии;
data.options.pieChartStrokeWidth	2	Тип: Number Ширина линий-разделителей секторов и внешней обводки диаграммы в пикселях. Задается целым числом.
data.properties *	—	Тип: IDataManager Object Свойства геообъекта, который визуализируется макетом.
data.properties.data *	—	Тип: Object[] Function Статистические данные, на основе которых строится макет. Должны представлять собой массив JSON-объектов с полями weight и color (вес и цвет сектора) либо функцию, возвращающую такой массив.

Параметр	Значение по умолчанию	Описание
data.properties.geoObjects	—	Тип: IGeoObject[] Массив геообъектов в составе кластера, который требуется визуализировать. Используется в случае отсутствия <code>properties.data</code> . В этом случае макет проходит по всем геообъектам, определяет значение опции <code>iconColor</code> каждого объекта и на основе этих данных строит диаграмму.
data.properties.iconCaption	""	Тип: String Подпись к диаграмме.
data.properties.iconContent	The sum of all the sectors	Тип: String Значение, которое пишется в центре диаграммы. Если не задано — выводится сумма весов секторов.

* Обязательный параметр/опция.

Пример:

```
var geoObject = new ymaps.Placemark([55.25, 37.43], {
    // Данные, на основе которых будет формироваться диаграмма.
    data: [
        { weight: 5, color: '#224080' },
        { weight: 3, color: '#408022' },
        { weight: 2, color: '#802240' }
    ],
    {
        iconLayout: 'default#pieChart',
        // Можно переопределить опции макета с префиксом icon.
        iconPieChartCoreRadius: 15
    }
});

myMap.geoObjects.add(geoObject);
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
emptinesschange	<p>Изменение признака пустоты макета. Экземпляр класса Event.</p> <p>Унаследовано от ILayout.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
parentelementchange	<p>Изменение родительского элемента. Экземпляр класса Event.</p> <p>Унаследовано от ILayout.</p>
shapechange	<p>Изменение фигуры области охватывающей макет. Экземпляр класса Event.</p> <p>Унаследовано от ILayout.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
build()		<p>Строит экземпляр макета на основе шаблона и добавляет его в родительский HTML-элемент.</p> <p>Унаследован от layout.templateBased.Base.</p>
clear()		<p>Удаляет содержимое макета из DOM.</p> <p>Унаследован от layout.templateBased.Base.</p>
destroy()		<p>Деструктор. Вызывается при завершении работы с макетом.</p> <p>Унаследован от ILayout.</p>
getData()	Object	<p>Возвращает объект данных макета.</p> <p>Унаследован от ILayout.</p>
getParentElement()	HTMLElement	<p>Возвращает родительский HTML-элемент.</p> <p>Унаследован от ILayout.</p>

Имя	Возвращает	Описание
getShape()	IShape null	<p>Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. По умолчанию пытается построить фигуру через опцию shape - <code>this.getData().options.get('shape')</code>. В опции можно задать как инстанцию класса, реализующего интерфейс IShape, так и JSON-описание пиксельной геометрии фигуры. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.</p> <p>Унаследован от layout.templateBased.Base.</p>
isEmpty()	Boolean	<p>Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.</p> <p>Унаследован от ILayout.</p>
onSublayoutSizeChange(sublayoutInfo, nodeSizeByContent)		<p>Автоматически вызывается при изменении размеров вложенного макета, добавленного с параметром 'observeSize'. Используется для переопределения в конкретных классах макетов для реакции на изменение размеров содержимого.</p> <p>Унаследован от layout.templateBased.Base.</p>
rebuild()		<p>Перестраивает макет.</p> <p>Унаследован от layout.templateBased.Base.</p>
setData(data)		<p>Задаёт данные макета.</p> <p>Унаследован от ILayout.</p>
setParentElement(parent)		<p>Добавляет макет в DOM-дерево.</p> <p>Унаследован от ILayout.</p>

layout.storage

Статический объект.

Экземпляр класса [util.AsyncStorage](#)

Хранилище классов лейаутов.

[Методы](#)

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.
define(key[, depends, resolveCallback[, context]])	util.AsyncStorage	Определяет в хранилище асинхронное значение.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
isDefined(key)	Boolean	Проверка доступности ключа в хранилище.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.
require(keys[, successCallback[, errorCallback[, context]])	vow.Promise	Асинхронный запрос на получение значений из хранилища.

layout.templateBased

layout.templateBased.Base

Расширяет [ILayout](#).

Базовый класс макета на основе шаблонов. Этот класс используется фабрикой макетов как базовый для создания пользовательских макетов.

См.: [templateLayoutFactory](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
layout.templateBased.Base(data)
```

Параметры:

Параметр	Значение по умолчанию	Описание
data *	—	Тип: Object Набор разнородных данных, на основе которых строится макет.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты макета. Экземпляр класса Event . Унаследовано от ILayout .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mousemove	Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> . Унаследовано от IDomEventEmitter .
multitouchmove	Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code> , которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля: <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. Унаследовано от IDomEventEmitter .
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code> , которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля: <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. Унаследовано от IDomEventEmitter .
parentelementchange	Изменение родительского элемента. Экземпляр класса Event . Унаследовано от ILayout .
shapechange	Изменение фигуры области охватывающей макет. Экземпляр класса Event . Унаследовано от ILayout .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
build()		Строит экземпляр макета на основе шаблона и добавляет его в родительский HTML-элемент.

Имя	Возвращает	Описание
<code>clear()</code>		Удаляет содержимое макета из DOM.
<code>destroy()</code>		Деструктор. Вызывается при завершении работы с макетом. Унаследован от ILayout .
<code>getData()</code>	Object	Возвращает объект данных макета. Унаследован от ILayout .
<code>getParentElement()</code>	HTMLElement	Возвращает родительский HTML-элемент. Унаследован от ILayout .
<code>getShape()</code>	IShape null	Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. По умолчанию пытается построить фигуру через опцию shape - <code>this.getData().options.get('shape')</code> . В опции можно задать как инстанцию класса, реализующего интерфейс IShape , так и JSON-описание пиксельной геометрии фигуры. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.
<code>isEmpty()</code>	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от ILayout .
<code>onSublayoutSizeChange(sublayoutInfo, nodeSizeByContent)</code>		Автоматически вызывается при изменении размеров вложенного макета, добавленного с параметром 'observeSize'. Используется для переопределения в конкретных классах макетов для реакции на изменение размеров содержимого.
<code>rebuild()</code>		Перестраивает макет.
<code>setData(data)</code>		Задаёт данные макета. Унаследован от ILayout .

Имя	Возвращает	Описание
<code>setParentElement(parent)</code>		Добавляет макет в DOM-дерево. Унаследован от ILayout .

Описание методов

build

```
{ } build()
```

Строит экземпляр макета на основе шаблона и добавляет его в родительский HTML-элемент.

clear

```
{ } clear()
```

Удаляет содержимое макета из DOM.

getShape

```
{IShape|null} getShape()
```

Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. По умолчанию пытается построить фигуру через опцию `shape` - `this.getData().options.get('shape')`. В опции можно задать как инстанцию класса, реализующего интерфейс `IShape`, так и JSON-описание пиксельной геометрии фигуры. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.

Пример:

```
// Создадим метку и зададим ей круглую интерактивную область.
var MyLayoutClass = ymaps.templateLayoutFactory.createClass('<div class="imageIcon">{{name}}</div>'),
    myPlacemark = new ymaps.Placemark([22, 34], {name: 'Кафе Маяк'}, {
      iconLayout: MyLayoutClass,
      iconShape: {type: 'Circle', coordinates: [0, 0], radius: 20}
    });
```

onSublayoutSizeChange

```
{ } onSublayoutSizeChange(sublayoutInfo, nodeSizeByContent)
```

Автоматически вызывается при изменении размеров вложенного макета, добавленного с параметром `'observeSize'`. Используется для переопределения в конкретных классах макетов для реакции на изменение размеров содержимого.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>sublayoutInfo</code> *	—	Тип: Object Информация о вложенном макете.
<code>nodeSizeByContent</code> *	—	Тип: Object Новые размеры элемента с учётом его содержимого. Доступные поля: <code>'width'</code> , <code>'height'</code> , <code>'scrollIX'</code> , <code>'scrollIY'</code> .

* Обязательный параметр/опция.

rebuild

```
{ } rebuild()
```

Перестраивает макет.

LoadingObjectManager

Расширяет [ICustomizable](#), [IEventEmitter](#), [IGeoObject](#), [IParentOnMap](#).

Менеджер объектов, осуществляющий их оптимальную подгрузку с сервера. Позволяет оптимально загружать, отображать, кластеризовать и управлять видимостью объектов. Менеджер отправляет запрос за данными по указанному url в JSONP-формате. Описание формата соответствует формату добавляемых в ObjectManager объектов (см. [ObjectManager.add](#)). Обратите внимание, что у объектов, отрисованных на карте через данный менеджер, нельзя включать режимы редактирования и перетаскивания.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
LoadingObjectManager(urlTemplate[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
urlTemplate *	—	Тип: String шаблон URL данных. Поддерживаются специальные конструкции по аналогии с Layer . Также поддерживаются подстановки: <ul style="list-style-type: none">• %b заменяется на массив географических координат, описывающих прямоугольную область, для которой требуется загрузить данные.• %t заменяется на массив номеров тайлов, описывающих прямоугольную область, для которой требуется загрузить данные.

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции. <ul style="list-style-type: none">Можно задавать все опции, указанные в описании Clusterer, за исключением опций hasBalloon и hasHint.Опции для кластеров задаются с префиксом cluster. Список опций указан в описании класса ClusterPlacemark;Опции для одиночных объектов задаются с префиксом geoObject. Список опций определен в классе GeoObject. Обратите внимание, менеджер не учитывает опцию 'visible'.
options.clusterize	false	Тип: Boolean Флаг, показывающий, нужно ли кластеризовать объекты. Обратите внимание, что на данный момент кластеризация работает только для точечных объектов. При включенном режиме кластеризации все неточечные объекты будут игнорироваться.
options.loadTileSize	256	Тип: Number Размер тайла для загрузки данных.
options.paddingParamName	'callback'	Тип: Boolean Имя GET-параметра, который содержит значение jsonp-колбека.

Параметр	Значение по умолчанию	Описание
<code>options.paddingTemplate</code>	null	<p>Тип: String</p> <p>Шаблон для jsonp-колбека. Поддерживает те же подстановки, что и <code>urlTemplate</code>. Все символы, не являющиеся буквой или цифрой, будут заменены на <code>'_'</code>. Если параметр не задан, то имя jsonp-колбека будет сгенерировано автоматически. Примеры преобразований при <code>tileNumber=[3, 1], zoom=9</code>:</p> <ul style="list-style-type: none"> <code>'myCallback=%x' => 'myCallback_3'</code> <code>'%c' => 'x_3_y_1_z_9'</code> <code>'callback2_%c' => 'callback2_x_3_y_1_z_9'</code> <code>'callback%test' => 'callback_test'</code> <code>'callback_%b' => 'callback_85_0841__180_0000_85_0841_180_0000'</code> <p>Обратите внимание, что если не использовать в значении опции подстановки, то это может привести к ошибке. Все запросы будут обращаться к одной callback-функции.</p>
<code>options.splitRequests</code>	false	<p>Тип: Boolean</p> <p>Разделять запросы за данными на запросы за одиночными тайлами. По умолчанию запросы делаются за данными для прямоугольной области, содержащей несколько тайлов.</p>
<code>options.syncOverlayInit</code>	false	<p>Тип: Boolean</p> <p>Флаг, разрешающий создавать оверлеи для объектов синхронно. Обратите внимание, что при синхронном создании оверлея нужно самостоятельно обеспечить загрузку нужного класса, реализующего интерфейс <code>IOverlay</code>. По умолчанию оверлеи создаются асинхронно, при этом класс оверлея загружается по требованию.</p>

Параметр	Значение по умолчанию	Описание
<code>options.viewportMargin</code>	128	<p>Тип: <code>Number Number[]</code></p> <p>Отступ для области, в которой показываются объекты. С помощью данной опции область показа объектов расширяется по отношению к видимой области карты.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
var objectManager = new ymaps.LoadingObjectManager('http://myServer.com/tile?bbox=%b', {
  // Включаем кластеризацию.
  clusterize: true,
  // Опции кластеров задаются с префиксом 'cluster'.
  clusterHasBalloon: false,
  // Опции геообъектов задаются с префиксом 'geoObject'.
  geoObjectOpenBalloonOnClick: false
});

// Опции можно задавать напрямую в дочерние коллекции.
objectManager.clusters.options.set({
  preset: 'islands#grayClusterIcons',
  hintContentLayout: ymaps.templateLayoutFactory.createClass('Группа объектов')
});
objectManager.objects.options.set('preset', 'islands#grayIcon');
```

2.

```
Пример ответа для LoadingObjectManager
jsonp_callback({
  // Ответ содержит поля error и data. Если возникает ошибка, то поле "error"
  // содержит код ошибки или описание.
  error: null,
  data: {
    type: 'FeatureCollection',
    features: [
      {
        type: 'Feature',
        geometry: {
          type: 'Point',
          coordinates: [55, 35]
        },
        id: 23,
        properties: {
          balloonContent: 'Содержание балуна метки',
          iconContent: 'Содержание метки'
        },
        options: {
          preset: 'islands#yellowIcon'
        }
      }
    ]
  }
});
```

Поля

Имя	Тип	Описание
<code>clusters</code>	<code>objectManager.ClusterCollection</code>	Коллекция кластеров, сгенерированных менеджером.
<code>events</code>	<code>IEventManager</code>	<p>Менеджер событий.</p> <p>Унаследовано от <code>IDomEventEmitter</code>.</p>
<code>geometry</code>	<code>IGeometry null</code>	<p>Геометрия геообъекта.</p> <p>Унаследовано от <code>IGeoObject</code>.</p>
<code>objects</code>	<code>objectManager.ObjectCollection</code>	Коллекция объектов, добавленных в слой.

Имя	Тип	Описание
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getBounds()	Number[][] null	Вычисляет границы области в геокоординатах, охватывающей все загруженные объекты в составе менеджера.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getObjectState(id)	Object	Получение информации о текущем состоянии объекта, добавленного в менеджер.

Имя	Возвращает	Описание
<code>getOverlay()</code>	<code>vow.Promise</code>	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от <code>IGeoObject</code> .
<code>getOverlaySync()</code>	<code>IOverlay</code> null	Метод предоставляет синхронный доступ к оверлею. Унаследован от <code>IGeoObject</code> .
<code>getParent()</code>	<code>IParentOnMap</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>getPixelBounds()</code>	<code>Number[][]</code> null	Вычисляет границы области в глобальных пиксельных координатах, охватывающей все загруженные объекты в составе менеджера.
<code>getTileUrl()</code>	<code>String</code> null	Возвращает URL тайла с данными.
<code>getUrlTemplate()</code>	<code>String</code>	Возвращает URL шаблона данных.
<code>reloadData()</code>		Метод, удаляющий все загруженные ранее данные и отправляющий запрос за новыми данными.
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .
<code>setUrlTemplate(urlTemplate)</code>		

Описание полей

clusters

```
{objectManager.ClusterCollection} clusters
```

Коллекция кластеров, сгенерированных менеджером.

Пример:

```
objectManager.objects.events.add('click', function (e) {
    var objectId = e.get('objectId');
    objectManager.objects.balloon.open(objectId);
});
```

objects

```
{objectManager.ObjectCollection} objects
```

Коллекция объектов, добавленных в слой.

Пример:

```
objectManager.objects.events.add('click', function (e) {
    var objectId = e.get('objectId');
    objectManager.objects.balloon.open(objectId);
});
```

Описание методов**getBounds**

```
{Number[][]|null} getBounds()
```

Вычисляет границы области в геокоординатах, охватывающей все загруженные объекты в составе менеджера.

Возвращает массив координат области или null, если менеджер не добавлен на карту.

getObjectState

```
{Object} getObjectState(id)
```

Получение информации о текущем состоянии объекта, добавленного в менеджер.

Возвращает объект, содержащий следующие поля:

- found - признак, указывающий, существует ли объект с переданным идентификатором. Type: Boolean.
- isShown - признак, указывающий, находится ли объект в видимой области карты. Type: Boolean.
- cluster - JSON-описание кластера, в который добавлен объект. Помимо обязательных полей, содержит поле properties.geoObjects с массивом объектов, попавших в кластер. Поле возвращается только при включенной кластеризации;
- isClustered – признак, указывающий, попал ли объект в состав кластера. Поле возвращается только при включенной кластеризации; Type: Boolean.
- isFilteredOut - признак, указывающий, прошел ли объект фильтрацию. Если фильтр не задан или объект прошел фильтрацию, значение поля будет false. Type: Boolean.

Параметры:

Параметр	Значение по умолчанию	Описание
id *	—	Тип: Object Идентификатор объекта, для которого нужно получить состояние.

* Обязательный параметр/опция.

Пример:

```
// Открытие балуна кластера с выбранным объектом.
```

```
// Получим данные о состоянии объекта внутри кластера.
var objectState = objectManager.getObjectState(objects[1].id);
// Проверяем, находится ли объект в видимой области карты.
if (objectState.found && objectState.isShown) {
    // Если объект попадает в кластер, открываем балун кластера с нужным выбранным объектом.
    if (objectState.isClustered) {
        objectManager.clusters.state.set('activeObject', objects[1]);
        objectManager.clusters.balloon.open(objectState.cluster.id);
    } else {
        // Если объект не попал в кластер, открываем его собственный балун.
        objectManager.objects.balloon.open(objects[i].id);
    }
}
```

getPixelBounds

```
{Number[][]|null} getPixelBounds()
```

Вычисляет границы области в глобальных пиксельных координатах, охватывающей все загруженные объекты в составе менеджера.

Возвращает массив координат области или null, если менеджер не добавлен на карту.

getTileUrl

```
{String|null} getTileUrl()
```

Возвращает URL тайла с данными.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>parameters</code> *	—	Тип:

* Обязательный параметр/опция.

Пример:

```
var objectManager = new ymaps.LoadingObjectManager('http://myServer.com/tile?bbox=%b');
objectManager.getTileUrl = function (parameters) {
    var boundingBox = parameters.boundingBox.join('~');
    return this.getUrlTemplate().replace(/%b/g, boundingBox);
};
```

getUrlTemplate

```
{String} getUrlTemplate()
```

Возвращает URL шаблона данных.

reloadData

```
{ } reloadData()
```

Метод, удаляющий все загруженные ранее данные и отправляющий запрос за новыми данными.

setUrlTemplate

```
{ } setUrlTemplate(urlTemplate)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>uriTemplate</code> *	—	Тип: String шаблон URL данных.

* Обязательный параметр/опция.

Map

Расширяет [IDomEventEmitter](#).

Класс для создания и управления картой.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Map(parentElement, state[, options])
```

Создает экземпляр карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>parentElement</code> *	—	Тип: Object String Ссылка на HTML-элемент, в котором размещается карта, или идентификатор этого HTML-элемента.
<code>state</code> *	—	Тип: Object Параметры карты.
<code>state.behaviors</code>	['default']	Тип: String[] Включенные поведения карты. По умолчанию включаются: перетаскивание карты, масштабирование карты мультисенсорным касанием и двойным кликом на устройствах с сенсорным экраном; перетаскивание карты мышью, масштабирование карты двойным щелчком и выделением области правой кнопкой мыши на остальных устройствах. Можно указывать любые ключи, поддерживаемые <code>behavior.Manager</code> .

Параметр	Значение по умолчанию	Описание
<code>state.bounds</code>	—	<p>Тип: <code>Number[][]</code></p> <p>Геокоординаты начальной области показа карты. При инициализации карты может быть задана либо пара <code>state.zoom-state.center</code>, либо данная область. Если совместно со <code>state.bounds</code> будут заданы параметры <code>state.zoom</code>, <code>state.center</code>, они будут проигнорированы.</p>
<code>state.center</code>	—	<p>Тип: <code>Number[]</code></p> <p>Геокоординаты центра карты. Должны быть заданы совместно со <code>state.zoom</code>.</p>
<code>state.controls</code>	<code>['default']</code>	<p>Тип: <code>String[]</code></p> <p>Добавленные по умолчанию элементы управления карты. Можно указывать любые ключи, поддерживаемые <code>control.Manager</code>. Список доступных ключей и наборы элементов управления по умолчанию описаны в <code>control.Manager.add</code>.</p>
<code>state.margin</code>	—	<p>Тип: <code>Number Number[]</code></p> <p>Отступы от краёв карты. Передаются в <code>map.margin.Manager.setDefaultMargin</code>.</p>
<code>state.type</code>	<code>'yandex#map'</code>	<p>Тип: <code>String MapType</code></p> <p>Тип карты. Может быть ключом или экземпляром класса <code>MapType</code>. Список доступных ключей:</p> <ul style="list-style-type: none"> • <code>'yandex#map'</code> - тип карты "схема"; • <code>'yandex#satellite'</code> - тип карты "спутник"; • <code>'yandex#hybrid'</code> - тип карты "гибрид".
<code>state.zoom</code>	—	<p>Тип: <code>Number</code></p> <p>Коэффициент масштабирования карты. Должен быть задан совместно с опцией <code>state.center</code>.</p>

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции карты. Через опции карты можно задавать настройки как самой карты, так и добавленных на неё объектов:</p> <ul style="list-style-type: none">• опции поведений карты;• опции балуна карты с префиксом balloon;• опции хинта карты с префиксом hint;• опции геообъектов с префиксом geoObject;• опции слоев с префиксом layer;• опции слоев активных областей с префиксом hotspotLayer; <p>Опции, которые трактует непосредственно сама карта, перечислены ниже.</p>

Параметр	Значение по умолчанию	Описание
<code>options.autoFitToViewport</code>	'ifNull'	<p>Тип: String</p> <p>Автоматическое слежение за контейнером карты. По умолчанию карта перестроится автоматически в случае, когда она инициализируется из скрытого контейнера, либо мы программно поменяли его размеры, иначе необходимо вызвать <code>map.container.fitToViewport</code>. Доступны следующие значения:</p> <ul style="list-style-type: none"> 'none' — не следить за изменением отображения контейнера при инициализации из скрытого контейнера или при программном изменении его размеров; 'ifNull' — как только контейнер получит CSS значение «display», отличное от «none», автоматически запустится <code>map.container.fitToViewport</code>, чтобы подстроить карту под него. После слежение будет остановлено; 'always' — постоянно следить за изменением состояния отображения контейнера карты.
<code>options.avoidFractionalZoom</code>	true	<p>Тип: Boolean</p> <p>true - карта не будет останавливаться на дробных значениях коэффициента масштабирования, false - будет.</p> <p>Примечание: Для десктопных браузеров данная опция по умолчанию принимает значение true, для мобильных - false.</p>
<code>options.exitFullscreenByEsc</code>	true	<p>Тип: Boolean</p> <p>Включение возможности выхода из «полноэкранного режима» с помощью нажатия на клавишу ESC. True — выходить из «полноэкранного режима» по нажатию на ESC, false — не выходить.</p>

Параметр	Значение по умолчанию	Описание
<code>options.fullscreenZIndex</code>	10000	Тип: Number Значение z-index контейнера карты в «полноэкранном режиме».
<code>options.mapAutoFocus</code>	true	Тип: Boolean true - при клике на карту будет убираться фокус с текущего активного dom-элемента, false - при клике на карту фокус будет оставаться в активном элементе.
<code>options.maxAnimationZoomDifference</code>	5	Тип: Number Максимальная разница между текущим и выставляемым коэффициентом масштабирования карты, при котором масштабирование будет происходить плавно.
<code>options.maxZoom</code>	23	Тип: Number Максимальный коэффициент масштабирования карты.
<code>options.minZoom</code>	0	Тип: Number Минимальный коэффициент масштабирования карты.
<code>options.nativeFullscreen</code>	false	Тип: Boolean Использовать нативный «полноэкранный режим», если возможно. При переходе в нативный «полноэкранный режим» браузер сначала запросит у пользователя подтверждение. В случае согласия пользователя браузер открывает карту на весь экран и скрывает свои управляющие элементы. Пользователь может выйти из этого режима нажав ESC. Опции <code>fullscreenZIndex</code> и <code>exitFullscreenByEsc</code> в нативном «полноэкранном режиме» ни на что не влияют.
<code>options.projection</code>	<code>ymaps.projection.wgs84Mercator</code>	Тип: IProjection Проекция карты.

Параметр	Значение по умолчанию	Описание
<code>options.restrictMapArea</code>	false	<p>Тип: Boolean Number[]</p> <p>Устанавливает область просмотра карты так, чтобы пользователь не мог выйти за пределы этой области. True — установить область в соответствии с текущей областью отображения, false — отключить опцию. Если необходимо ограничить область с известными координатами, нужно указать эту прямоугольную область.</p>
<code>options.suppressMapOpenBlock</code>	false	<p>Тип: Boolean</p> <p>Нужно ли скрывать предложение открыть текущую карту в Яндекс.Картах, максимально сохранив всю имеющуюся информацию о ней. True - скрывать, false - не скрывать. Ссылка на Яндекс.Карты отображается в левом нижнем углу карты.</p>
<code>options.suppressObsoleteBrowserNotifier</code>	false	<p>Тип: Boolean</p> <p>Нужно ли в устаревших версиях браузеров скрывать уведомление с предложением обновить браузер. True - скрывать, false - не скрывать. Уведомление отображается в левом нижнем углу карты.</p>
<code>options.yandexMapAutoSwitch</code>	true	<p>Тип: Boolean</p> <p>true — включить автоматическое переключение на Народную Карту там, где схема недостаточно подробна, false — выключить. Эта опция работает только при значениях параметра lang=ru_RU и lang=uk_UA.</p>
<code>options.yandexMapDisablePoiInteractivity</code>	false	<p>Тип: boolean</p> <p>True — отключить интерактивность POI (точек интереса) в подложке карты, false — включить.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
// Инициализация карты с заданным центром и уровнем масштабирования.
var myMap = new ymaps.Map('map', {
  center: [55.74954, 37.621587],
  zoom: 10
});
```

2.

```
// Инициализация карты по известной области просмотра.
// Полагаем, что на странице подключен jQuery.
var $mapElement = $('#map'),
    myMap = new ymaps.Map(
        $mapElement[0],
        ymaps.util.bounds.getCenterAndZoom(
            [[55.7, 37.6], [55.8, 37.7]],
            [$mapElement.width(), $mapElement.height()]
        )
    );
```

3.

```
// Инициализация карты из результата геокодирования
var myMap;
ymaps.geocode('Москва').then(function (res) {
    myMap = new ymaps.Map('map', {
        center: res.geoObjects.get(0).geometry.getCoordinates(),
        zoom :
    });
});
```

Поля

Имя	Тип	Описание
action	map.action.Manager	Менеджер действий карты.
balloon	map.Balloon	Балун карты.
behaviors	map.behavior.Manager	Менеджер поведений карты. Позволяет включать и выключать поведения, а также предоставляет доступ к их методам и свойствам.
container	map.Container	Контейнер карты.
controls	control.Manager	Элементы управления картой.
converter	map.Converter	Преобразователь пиксельных точек карты из глобальных в локальные и наоборот.
copyrights	map.Copyrights	Менеджер размещённой на карте информации об авторских правах.
cursors	util.cursor.Manager	Менеджер курсоров карты.
events	event.Manager	Менеджер событий карты. Поддерживает подписку с приоритетами. Бросает события типа MapEvent .
geoObjects	map.GeoObjects	Менеджер геообъектов карты.
hint	map.Hint	Хинт карты.
layers	map.layer.Manager	Менеджер слоев карты.
margin	map.margin.Manager	Менеджер отступов карты.
options	option.Manager	Опции карты.
panes	map.pane.Manager	Менеджер контейнеров объектов карты.
zoomRange	map.ZoomRange	Объект, предоставляющий доступ к информации о доступных в точке уровнях масштабирования.

События

Имя	Описание
actionbegin	Началось новое плавное движение карты. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> action - запущенное действие.
actionbreak	Событие, возникающее при досрочном прекращении шага действия (например, из-за выполнения другого действия или нового шага того же действия). Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> action - действие.
actionend	Окончание плавного движения карты. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> action - остановленное действие.
actiontick	Начало исполнения нового шага плавного движения (например, сдвиг карты пользователем или шаг анимации плавного масштабирования). Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> action - выполняемое в данный момент действие; tick – описание шага действия в виде объекта с полями globalPixelCenter, zoom, duration и timingFunction.
actiontickcomplete	Окончание исполнения шага плавного движения. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> action - выполняемое в данный момент действие; tick – описание шага действия в виде объекта с полями globalPixelCenter, zoom, duration и timingFunction.
balloonclose	Закрытие балуна. Экземпляр класса Event .
balloonopen	Открытие балуна на карте. Экземпляр класса Event .
boundschange	Событие изменения области просмотра карты (в результате изменения центра или уровня масштабирования). Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> oldCenter - предыдущий центр карты в геокоординатах; Новый центр карты в геокоординатах. oldZoom - предыдущий уровень масштабирования; newZoom - новый уровень масштабирования; oldGlobalPixelCenter - предыдущий центр карты в глобальных пикселах; newGlobalPixelCenter - новый центр карты в глобальных пикселах; oldBounds - предыдущая область показа; newBounds - новая область показа.

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
destroy	<p>Карта уничтожена.</p>
hintclose	<p>Заккрытие хинта. Экземпляр класса Event.</p>
hintopen	<p>Открытие хинта на карте. Экземпляр класса Event.</p>
marginchange	<p>Изменились отступы карты. Экземпляр класса Event.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	Изменились опции карты.
sizechange	<p>Изменился размер карты. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>oldSize</code> - предыдущий размер карты; <code>newSize</code> - новый размер карты.
typechange	Изменился тип карты. Экземпляр класса Event .
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает карту.

Имя	Возвращает	Описание
<code>getBounds([options])</code>	<code>Number[][]</code>	Возвращает двумерный массив геокоординат левого нижнего и правого верхнего углов области показа карты.
<code>getCenter([options])</code>	<code>Number[]</code>	Возвращает географические координаты текущего центра карты.
<code>getGlobalPixelCenter([options])</code>	<code>Number[]</code>	Возвращает глобальные пиксельные координаты текущего центра карты.
<code>getPanoramaManager()</code>	<code>vow.Promise.<panorama.Manager></code>	Возвращает <code>vow.Promise</code> , который будет разрешен менеджером панорамы карты. Promise отклоняется при возникновении ошибки.
<code>getType()</code>	<code>String MapType</code>	Возвращает текущий тип карты.
<code>getZoom()</code>	<code>Number</code>	Возвращает текущий коэффициент масштабирования карты.
<code>panTo(center[, options])</code>	<code>vow.Promise</code>	Устанавливает центр карты. Если передан массив точек, то карта будет перемещаться от одной точки к другой.
<code>setBounds(bounds[, options])</code>	<code>vow.Promise</code>	Позиционирует карту для показа переданного района.
<code>setCenter(center[, zoom[, options]])</code>	<code>vow.Promise</code>	Устанавливает центр и коэффициент масштабирования карты. Центр задается в географических координатах.
<code>setGlobalPixelCenter(globalPixelCenter[, zoom[, options]])</code>	<code>vow.Promise</code>	Устанавливает центр и коэффициент масштабирования карты. Центр задается в глобальных пиксельных координатах.
<code>setType(type[, options])</code>	<code>vow.Promise</code>	Устанавливает тип карты.
<code>setZoom(zoom[, options])</code>	<code>vow.Promise</code>	Устанавливает коэффициент масштабирования карты.

Описание полей

action

```
{map.action.Manager} action
```

Менеджер действий карты.

Пример:

```
var myAction = new ymaps.map.action.Single({
  center: [0, 0],
  zoom: 4,
  duration: 1000,
  timingFunction: "ease-in"
});

myMap.action.execute(myAction);
```

balloon

```
{map.Balloon} balloon
```

Балун карты.

behaviors

```
{map.behavior.Manager} behaviors
```

Менеджер поведений карты. Позволяет включать и выключать поведения, а также предоставляет доступ к их методам и свойствам.

Пример:

```
// Включим масштабирование колесом мыши
myMap.behaviors.enable('scrollZoom');
```

container

```
{map.Container} container
```

Контейнер карты.

Пример:

```
// Подгоним размер карты под новый размер контейнера
// (например, если изменилась верстка страницы или карта была инициализирована
// в скрытом состоянии)
map.container.fitToViewport();
```

controls

```
{control.Manager} controls
```

Элементы управления картой.

Пример:

```
myMap.controls.add('zoomControl', {
  float: 'none',
  position: {
    right: 40,
    top: 5
  }
});
```

converter

```
{map.Converter} converter
```

Преобразователь пиксельных точек карты из глобальных в локальные и наоборот.

Пример:

```
// Преобразуем координаты мыши в географические.
var projection = map.options.get('projection');
$('#map').bind('click', function (e) {
    console.log(projection.fromGlobalPixels(
        map.converter.pageToGlobal([e.pageX, e.pageY]), map.getZoom()
    ));
});
```

copyrights

```
{map.Copyrights} copyrights
```

Менеджер размещённой на карте информации об авторских правах.

Пример:

```
// Добавим информацию об авторе.
map.copyrights.add('© Вася Пупкин');
```

cursors

```
{util.cursor.Manager} cursors
```

Менеджер курсоров карты.

Пример:

```
// Установим на карте курсор "знак вопроса".
var accessor = map.cursors.push('help');
// ...
// Удалим курсор.
accessor.remove();
```

events

```
{event.Manager} events
```

Менеджер событий карты. Поддерживает подписку с приоритетами. Бросает события типа [MapEvent](#).

Примеры:**1.**

```
// Поставим метку по клику над картой.
map.events.add('click', function (e) {
    // Географические координаты точки клика можно узнать
    // посредством вызова метода .get('coords').
    var position = e.get('coords');
    map.geoObjects.add(new ymaps.Placemark(position));
});
```

2.

```
// Будем отслеживать центр и зум карты при плавных перемещениях.
map.events.add('actiontick', function () {
    var state = map.action.getCurrentState();
    console.log(state.zoom, state.globalPixelCenter);
});
```

geoObjects

```
{map.GeoObjects} geoObjects
```

Менеджер геообъектов карты.

hint

```
{map.Hint} hint
```

Хинт карты.

layers

```
{map.layer.Manager} layers
```

Менеджер слоев карты.

См.: [Layer](#)

Пример:

```
// Добавим слой пользовательских тайлов на карту.  
map.layers.add(new ymaps.Layer('http://some.server/tiles?&c'));
```

margin

```
{map.margin.Manager} margin
```

Менеджер отступов карты.

options

```
{option.Manager} options
```

Опции карты.

Пример:

```
// Запретим всем объектам на карте открывать балуны по щелчку мыши.  
map.options.set('openBalloonOnClick', false);
```

panes

```
{map.pane.Manager} panes
```

Менеджер контейнеров объектов карты.

Пример:

```
// Добавим свой элемент в контейнер элементов управления карты.  
$('<div><input type="button" value="Click!"/></div>')  
  .css({ position: 'absolute', left: '5px', top: '50px'})  
  .appendTo(map.panes.get('controls').getElement());
```

zoomRange

```
{map.ZoomRange} zoomRange
```

Объект, предоставляющий доступ к информации о доступных в точке уровнях масштабирования.

Пример:

```
// Узнаем максимальный доступный в точке [20, 30] уровень  
// масштабирования карты.  
map.zoomRange.get([20, 30]).then(function (zoomRange) {  
  alert(zoomRange[1]);  
});
```

Описание событий

actionbegin

Началось новое плавное движение карты. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - запущенное действие.

actionbreak

Событие, возникающее при досрочном прекращении шага действия (например, из-за выполнения другого действия или нового шага того же действия). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - действие.

actionend

Окончание плавного движения карты. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - остановленное действие.

actiontick

Начало исполнения нового шага плавного движения (например, сдвиг карты пользователем или шаг анимации плавного масштабирования). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - выполняемое в данный момент действие;
- tick – описание шага действия в виде объекта с полями globalPixelCenter, zoom, duration и timingFunction.

Пример:

```
// Отслеживает все перемещения карты, даже перетаскивание пользователем
// и плавное масштабирование.
map.events.add('actiontick', function (e) {
    var tick = e.get('tick');
    console.log('Сейчас карта переместится в точку (' +
        map.options.get('projection').fromGlobalPixels(tick.globalPixelCenter, tick.zoom).join(',') +
        ') в течение ' + e.get('tick').duration + ' миллисекунд');
});
```

actiontickcomplete

Окончание исполнения шага плавного движения. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - выполняемое в данный момент действие;
- tick – описание шага действия в виде объекта с полями globalPixelCenter, zoom, duration и timingFunction.

balloonclose

Закрытие балуна. Экземпляр класса [Event](#).

balloonopen

Открытие балуна на карте. Экземпляр класса [Event](#).

boundschange

Событие изменения области просмотра карты (в результате изменения центра или уровня масштабирования). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldCenter` - предыдущий центр карты в геокоординатах;
- Новый центр карты в геокоординатах.
- `oldZoom` - предыдущий уровень масштабирования;
- `newZoom` - новый уровень масштабирования;
- `oldGlobalPixelCenter` - предыдущий центр карты в глобальных пикселах;
- `newGlobalPixelCenter` - новый центр карты в глобальных пикселах;
- `oldBounds` - предыдущая область показа;
- `newBounds` - новая область показа.

Пример:

```
// Будем отслеживать изменение уровня масштабирования карты.  
map.events.add('boundschange', function (event) {  
    if (event.get('newZoom') != event.get('oldZoom')) {  
        alert('Уровень масштабирования изменился.');    }  
});
```

destroy

Карта уничтожена.

hintclose

Закрытие хинта. Экземпляр класса [Event](#).

hintopen

Открытие хинта на карте. Экземпляр класса [Event](#).

marginchange

Изменились отступы карты. Экземпляр класса [Event](#).

optionschange

Изменились опции карты.

sizechange

Изменился размер карты. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldSize` - предыдущий размер карты;
- `newSize` - новый размер карты.

typechange

Изменился тип карты. Экземпляр класса [Event](#).

Описание методов**destroy**

```
{}
```

`destroy()`

Уничтожает карту.

getBounds

```
{Number[][][]} getBounds([options])
```

Возвращает двумерный массив геокоординат левого нижнего и правого верхнего углов области показа карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.useMapMargin</code>	false	Тип: Boolean Нужно ли учитывать отступы, которые были рассчитаны в менеджере отступов map.margin.Manager .

getCenter

```
{Number[]} getCenter([options])
```

Возвращает географические координаты текущего центра карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.useMapMargin</code>	false	Тип: Boolean Нужно ли учитывать отступы, которые были рассчитаны в менеджере отступов map.margin.Manager .

getGlobalPixelCenter

```
{Number[]} getGlobalPixelCenter([options])
```

Возвращает глобальные пиксельные координаты текущего центра карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.

Параметр	Значение по умолчанию	Описание
<code>options.useMapMargin</code>	<code>false</code>	Тип: Boolean Нужно ли учитывать отступы, которые были рассчитаны в менеджере отступов <code>map.margin.Manager</code> .

Пример:

```
// Сдвинем карту на 10 пикселей влево.  
var position = map.getGlobalPixelCenter();  
map.setGlobalPixelCenter([ position[0] - 10, position[1] ]);
```

getPanoramaManager

```
{vow.Promise.<panorama.Manager>} getPanoramaManager()
```

Возвращает `vow.Promise`, который:

- будет **разрешен** объектом `<panorama.Manager>` — менеджером панорамы карты;
- либо **отклонен** с ошибкой.

getType

```
{String|MapType} getType()
```

Возвращает текущий тип карты.

getZoom

```
{Number} getZoom()
```

Возвращает текущий коэффициент масштабирования карты.

panTo

```
{vow.Promise} panTo(center[, options])
```

Устанавливает центр карты. Если передан массив точек, то карта будет перемещаться от одной точки к другой.

Возвращает объект-promise. В случае корректного выполнения действия будет возвращен `resolve` без значения. В случае ошибки будет возвращен `reject` с описанием ошибки.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>center</code> *	—	Тип: <code>Number[] Object[]</code> Центр карты или массив точек для последовательного обхода.
<code>options</code>	—	Тип: <code>Object</code> Опции.

Параметр	Значение по умолчанию	Описание
<code>options.checkZoomRange</code>	false	<p>Тип: Boolean</p> <p>Проверка возможности установить текущий коэффициент масштабирования при изменении центра карты. Если значение опции равно true, метод вызывается асинхронно. Производится обращение к серверу, который возвращает интервал допустимых для данного центра значений коэффициентов масштабирования. После этого выставляется указанный центр и корректный зум.</p>
<code>options.delay</code>	1000	<p>Тип: Number</p> <p>Время задержки между перемещениями в миллисекундах.</p>
<code>options.duration</code>	500	<p>Тип: Number</p> <p>Время анимации в миллисекундах.</p>
<code>options.flying</code>	true	<p>Тип: Boolean</p> <p>Разрешает уменьшать и затем увеличивать зум карты при перемещении между точками. Если расстояние между точками не превышает двух размеров контейнера карты, то карта плавно переместит центр в указанную точку без изменения масштаба. Если расстояние больше, то карта выставит себе меньший коэффициент масштабирования, затем сместится к конечной точке и выставит требуемый коэффициент масштабирования (эмуляция полета над картой).</p>

Параметр	Значение по умолчанию	Описание
<code>options.safe</code>	<code>true</code>	<p>Тип: Boolean</p> <p>Безопасное смещение карты. Используется при значении опции <code>flying</code> равной <code>false</code>. Если расстояние между точками не превышает двух размеров контейнера карты, то карта плавно переместит центр в указанную точку. Если расстояние больше, то произойдет мгновенный переход в выбранную точку. При помощи этого режима удобно перемещать карту без отображения незагруженных тайлов.</p>
<code>options.timingFunction</code>	<code>'ease-in-out'</code>	<p>Тип: String</p> <p>Тайминг-функция. Аналогично значению CSS свойства <code>transition-timing-function</code>. Полный список значений: http://www.w3.org/TR/css3-transitions/#transition-timing-function_tag</p>
<code>options.useMapMargin</code>	<code>false</code>	<p>Тип: Boolean</p> <p>Нужно ли учитывать отступы, которые были рассчитаны в менеджере отступов <code>map.margin.Manager</code>.</p>

* Обязательный параметр/опция.

Пример:

```
// Перелет из Калининграда во Владивосток через Москву.
map.setCenter([54.704815, 20.466380], 10);
map.panTo([
  [55.751574, 37.573856],
  [43.134091, 131.928478]
]).then(function () {
  alert('Прилетели!');
}, function (err) {
  alert('Произошла ошибка ' + err);
}, this);
```

setBounds

```
{vow.Promise} setBounds(bounds[, options])
```

Позиционирует карту для показа переданного района.

Возвращает объект-promise. В случае корректного выполнения действия будет возвращен `resolve` без значения. В случае ошибки будет возвращен `reject` с описанием ошибки.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>bounds *</code>	—	<p>Тип: Number[]</p> <p>Границы области видимости, заданные координатами левого нижнего и правого верхнего углов области.</p>
<code>options</code>	—	<p>Тип: Object</p> <p>Опции.</p>
<code>options.checkZoomRange</code>	false	<p>Тип: Boolean</p> <p>Проверка возможности установить указанный коэффициент масштабирования. Если значение опции равно true, метод вызывается асинхронно. Производится обращение к серверу, который возвращает интервал допустимых для данного центра значений коэффициентов масштабирования. После этого выставляется указанный центр и корректный зум.</p>
<code>options.duration</code>	0	<p>Тип: Number</p> <p>Время анимации в миллисекундах.</p>
<code>options.preciseZoom</code>	false	<p>Тип: Boolean</p> <p>Возможность использования дробных коэффициентов масштабирования.</p>
<code>options.timingFunction</code>	'linear'	<p>Тип: String</p> <p>Тайминг-функция. Аналогично значению CSS свойства transition-timing-function. Полный список значений: http://www.w3.org/TR/css3-transitions/#transition-timing-function_tag</p>

Параметр	Значение по умолчанию	Описание
<code>options.useMapMargin</code>	false	Тип: Boolean Нужно ли учитывать отступы, которые были рассчитаны в менеджере отступов <code>map.margin.Manager</code> .
<code>options.zoomMargin</code>	0	Тип: Number Number[] Отступы от границ видимой области карты. Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы top, right, bottom, left. При включенной опции "useMapMargin" значение "zoomMargin" складывается со значениями, которые были рассчитаны в менеджере отступов <code>map.margin.Manager</code> .

* Обязательный параметр/опция.

Пример:

```
// Устанавливает центр и коэффициент масштабирования карты.
// Если текущий zoomRange не совпадает с zoomRange для нового центра карты,
// могут быть отображены серые тайлы в случае, если область показа очень мала.
// Чтобы избежать этой проблемы, используйте опцию checkZoomRange.
map.setBounds([[60,-40], [20,60]], {
  checkZoomRange: true,
}).then(function () {
  // Действие было успешно завершено.
}, function (err) {
  // Не удалось показать заданный регион.
  // ...
}, this);
```

setCenter

```
{vow.Promise} setCenter(center[, zoom[, options]])
```

Устанавливает центр и коэффициент масштабирования карты. Центр задается в географических координатах.

Возвращает объект-promise. В случае корректного выполнения действия будет возвращен resolve без значения. В случае ошибки будет возвращен reject с описанием ошибки.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>center</code> *	—	Тип: Number[] Геокоординаты центра карты.

Параметр	Значение по умолчанию	Описание
<code>zoom</code>	—	Тип: Number Коэффициент масштабирования карты.
<code>options</code>	—	Тип: Object Опции.
<code>options.checkZoomRange</code>	false	Тип: Boolean Проверка возможности установить указанный коэффициент масштабирования. Если значение опции равно true, метод вызывается асинхронно. Производится обращение к серверу, который возвращает интервал допустимых для данного центра значений коэффициентов масштабирования. После этого выставляется указанный центр и корректный зум.
<code>options.duration</code>	0	Тип: Number Время анимации в миллисекундах.
<code>options.timingFunction</code>	'linear'	Тип: String Тайминг-функция. Аналогично значению CSS свойства transition-timing-function. Полный список значений: http://www.w3.org/TR/css3-transitions/#transition-timing-function_tag
<code>options.useMapMargin</code>	false	Тип: Boolean Нужно ли учитывать отступы, которые были рассчитаны в менеджере отступов map.margin.Manager .

* Обязательный параметр/опция.

Пример:

```
myMap.setCenter([40, 50], 3, {
  checkZoomRange: true
});
```

setGlobalPixelCenter

```
{vow.Promise} setGlobalPixelCenter(globalPixelCenter[, zoom[, options]])
```

Устанавливает центр и коэффициент масштабирования карты. Центр задается в глобальных пиксельных координатах.

Возвращает объект-promise. В случае корректного выполнения действия будет возвращен resolve без значения. В случае ошибки будет возвращен reject с описанием ошибки.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>globalPixelCenter</code> *	—	Тип: Number[] Пиксельные координаты нового центра карты.
<code>zoom</code>	—	Тип: Number Коэффициент масштабирования карты.
<code>options</code>	—	Тип: Object Опции.
<code>options.checkZoomRange</code>	false	Тип: Boolean Проверка возможности установить указанный коэффициент масштабирования. Если значение опции равно true, метод вызывается асинхронно. Производится обращение к серверу, который возвращает интервал допустимых для данного центра значений коэффициентов масштабирования. После этого выставляется указанный центр и корректный зум.
<code>options.duration</code>	0	Тип: Number Время анимации в миллисекундах.

Параметр	Значение по умолчанию	Описание
<code>options.timingFunction</code>	'linear'	Тип: String Тайминг-функция. Аналогично значению CSS свойства transition-timing-function. Полный список значений: http://www.w3.org/TR/css3-transitions/#transition-timing-function_tag
<code>options.useMapMargin</code>	false	Тип: Boolean Нужно ли учитывать отступы, которые были рассчитаны в менеджере отступов <code>map.margin.Manager</code> .

* Обязательный параметр/опция.

Пример:

```
// Сдвинем карту на 10 пикселей влево.
var position = map.getGlobalPixelCenter();
map.setGlobalPixelCenter([ position[0] - 10, position[1] ]);
```

setType

```
{vow.Promise} setType(type[, options])
```

Устанавливает тип карты.

Возвращает объект-promise. В случае корректного выполнения действия будет возвращен resolve без значения. В случае ошибки будет возвращен reject с описанием ошибки.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>type</code> *	—	Тип: String MapType Тип карты. Может быть ключом или экземпляром класса MapType . Список доступных ключей: <ul style="list-style-type: none"> 'yandex#map' - тип карты "схема"; 'yandex#satellite' - тип карты "спутник"; 'yandex#hybrid' - тип карты "гибрид".

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции карты.
<code>options.checkZoomRange</code>	false	Тип: Boolean Проверка возможности установить указанный тип карты при данном уровне масштабирования. При значении true производится обращение к серверу, который возвращает интервал допустимых для данного типа карты значений коэффициентов масштабирования. После этого выставляется указанный центр и корректный зум.

* Обязательный параметр/опция.

Пример:

```
map.setType('yandex#hybrid', {
  checkZoomRange: true
}).then(function () {
  // Тип карты был установлен с допустимым уровнем масштабирования.
}, this);
```

setZoom

```
{vow.Promise} setZoom(zoom[, options])
```

Устанавливает коэффициент масштабирования карты.

Возвращает объект-promise. В случае корректного выполнения действия будет возвращен resolve без значения. В случае ошибки будет возвращен reject с описанием ошибки.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>zoom</code> *	—	Тип: Number Коэффициент масштабирования карты.
<code>options</code>	—	Тип: Object Опции.
<code>options.checkZoomRange</code>	false	Тип: Boolean Проверка возможности установить указанный коэффициент масштабирования.

Параметр	Значение по умолчанию	Описание
options.duration	0	Тип: Number Время анимации в миллисекундах.
options.useMapMargin	false	Тип: Boolean Нужно ли учитывать отступы, которые были рассчитаны в менеджере отступов map.margin.Manager .

* Обязательный параметр/опция.

Пример:

```
myMap.setZoom(4, {duration: 1000});
```

map

map.action

map.action.Continuous

Расширяет [IMapAction](#).

Движение карты, состоящее из одного или более шагов. Предназначено для реализации сложных движений карты.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.action.Continuous()
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
end	Событие, оповещающее карту об окончании движения. Унаследовано от IMapAction .

Имя	Описание
tick	<p>Событие, оповещающее карту о следующем шаге движения. Содержит поля:</p> <ul style="list-style-type: none"> <code>globalPixelCenter</code> - новый центр карты в глобальных пикселах; <code>zoom</code> - новый коэффициент масштабирования карты; <code>duration</code> - время, которое отводится на выполнение шага; <code>timingFunction</code> - функция, описывающая вид движения. <p>Унаследовано от IMapAction.</p>

Методы

Имя	Возвращает	Описание
begin(mapActionManager)		<p>Запускает движение на исполнение картой. Этот метод автоматически вызывается менеджером движений карты. С момента вызова IMapAction.begin менеджер движений слушает события IMapAction.tick и IMapAction.end и исполняет их.</p> <p>Унаследован от IMapAction.</p>
end()		<p>Останавливает движение.</p> <p>Унаследован от IMapAction.</p>
isActive()	Boolean	<p>Проверяет, выполняется ли движение картой в данный момент.</p>
tick(tick)	map.action.Continuous	<p>Выполняет один шаг движения карты.</p>

Описание методов

isActive

```
{Boolean} isActive()
```

Проверяет, выполняется ли движение картой в данный момент.

Возвращает `true` - движение в данный момент выполняется, `false` - нет.

tick

```
{map.action.Continuous} tick(tick)
```

Выполняет один шаг движения карты.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tick *</code>	—	Тип: Object Параметры движения.
<code>tick.duration</code>	0	Тип: Number Время исполнения движения, в миллисекундах.
<code>tick.globalPixelCenter</code>	—	Тип: Number[] Новый центр карты в глобальных пикселях. Должен быть задан один из параметров: либо <code>pixelOffset</code> , либо <code>globalPixelCenter</code> .
<code>tick.pixelOffset</code>	—	Тип: Number[] Смещение в пикселях относительно предыдущего центра. Должен быть задан один из параметров: либо <code>pixelOffset</code> , либо <code>globalPixelCenter</code> .
<code>tick.timingFunction</code>	'linear'	Тип: String Тайминг-функция.
<code>tick.zoom</code>	—	Тип: Number Новый коэффициент масштабирования карты. Если не указан, масштаб карты не изменяется.

* Обязательный параметр/опция.

map.action.Manager

Расширяет [IEventEmitter](#).

Менеджер действий карты. Позволяет выполнять сложные движения на карте и гарантирует, что сложные движения не наложатся друг на друга. У каждой карты уже есть собственный менеджер действий, доступный как [Map.action](#). Не создавайте экземпляры этого класса без необходимости.

См.: [Map.action](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.action.Manager(map)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: Map Карта.

* Обязательный параметр/опция.

Пример:

```
// Создадим сложное движение: каждые 100 мс центр
// карты будет смещаться на случайную величину.

// Создание экземпляра сложного движения
var action = new ymaps.map.action.Continuous();
// Выполнение его на карте
myMap.action.execute(action);

// Вызов центра пиксельной карты и уровня масштабирования
var center = myMap.getGlobalPixelCenter();
var zoom = myMap.getZoom();
// Создание случайного сдвига каждые 100 миллисекунд
var interval = window.setInterval(function () {
    center[0] += Math.round(Math.random() * 100) - 50;
    center[1] += Math.round(Math.random() * 100) - 50;
    // Создание нового сдвига на карте
    action.tick({
        globalPixelCenter: center,
        zoom: zoom
    });
}, 100);

// Как только пользователь перемещает карту, наше движение
// перестает выполняться, и происходит событие "end".
var listener = action.events.once('end', function () {
    listener.removeAll();
    window.clearInterval(interval);
});
```

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
<code>begin</code>	Событие, возникающее, когда действие началось. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>action</code> - запущенное действие.
<code>break</code>	Событие, возникающее при досрочном прекращении шага действия (например, из-за выполнения другого действия или нового шага того же действия). Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>action</code> - действие.
<code>end</code>	Событие, возникающее при остановке действия. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>action</code> - остановленное действие.

Имя	Описание
tick	Событие, возникающее, когда очередной шаг действия начинает исполняться. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> action - выполняемое в данный момент действие; tick – описание шага действия в виде объекта с полями globalPixelCenter, zoom, duration и timingFunction.
tickcomplete	Событие, возникающее при окончании выполнения шага действия. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> action - выполняемое в данный момент действие; tick – описание шага действия в виде объекта с полями globalPixelCenter, zoom, duration и timingFunction.

Методы

Имя	Возвращает	Описание
breakTick()		Прерывает очередной шаг действия.
execute(action)		Запускает выполнение действия на карте. Если в этот момент на карте выполняется какое-то другое движение, то оно останавливается (вызывается метод end). Новое движение запускается посредством вызова метода begin .
getCurrentState()	Object	Определяет состояние карты в момент ее плавного движения.
getMap()	Map	Возвращает ссылку на карту.
setCorrection(userFunction)		Позволяет устанавливать пользовательские коррекции для сложных движений на карте. По окончании коррекции необходимо вернуть скорректированные значения.
stop()		Останавливает выполнение действия на карте.

Описание событий

begin

Событие, возникающее, когда действие началось. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - запущенное действие.

break

Событие, возникающее при досрочном прекращении шага действия (например, из-за выполнения другого действия или нового шага того же действия). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - действие.

end

Событие, возникающее при остановке действия. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - остановленное действие.

tick

Событие, возникающее, когда очередной шаг действия начинает исполняться. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - выполняемое в данный момент действие;
- tick – описание шага действия в виде объекта с полями globalPixelCenter, zoom, duration и timingFunction.

Пример:

```
// Отслеживает все перемещения карты, даже перетаскивание пользователем
// и плавное масштабирование.
myMap.action.events.add('tick', function (e) {
  var tick = e.get('tick');
  console.log('Сейчас карта переместится в точку (' +
    myMap.options.get('projection').fromGlobalPixels(tick.globalPixelCenter, tick.zoom).join(',') +
    ') в течение ' + e.get('tick').duration + ' миллисекунд');
});
```

tickcomplete

Событие, возникающее при окончании выполнения шага действия. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- action - выполняемое в данный момент действие;
- tick – описание шага действия в виде объекта с полями globalPixelCenter, zoom, duration и timingFunction.

Описание методов

breakTick

```
{ } breakTick()
```

Прерывает очередной шаг действия.

execute

```
{ } execute(action)
```

Запускает выполнение действия на карте. Если в этот момент на карте выполняется какое-то другое движение, то оно останавливается (вызывается метод `end`). Новое движение запускается посредством вызова метода `begin`.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>action</code> *	—	Тип: <code>IMapAction</code> Действие.

* Обязательный параметр/опция.

`getCurrentState`

```
{Object} getCurrentState()
```

Определяет состояние карты в момент ее плавного движения.

Возвращает объект с полями: `isTicking` — выполняется ли сейчас шаг плавного движения; `tickProgress` — какая часть текущего шага выполнена; `zoom` — масштаб карты во время текущего шага; `globalPixelCenter` — центр карты в глобальных пикселях в момент текущего шага.

Пример:

```
// Регистрирует текущий центр карты.  
// Работает даже во время плавного масштабирования или незавершенного  
// перетаскивания карты пользователем.  
window.setInterval(function () {  
    console.log(myMap.action.getCurrentState().center.join(', '));  
}, 100);
```

`getMap`

```
{Map} getMap()
```

Возвращает ссылку на карту.

`setCorrection`

```
{ } setCorrection(userFunction)
```

Позволяет устанавливать пользовательские коррекции для сложных движений на карте. По окончании коррекции необходимо вернуть скорректированные значения.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>userFunction</code> *	—	Тип: <code>Function</code> Пользовательская функция коррекции шагов.

* Обязательный параметр/опция.

Пример:

```
// Сделаем так, чтобы пользователь не мог перетащить центр карты  
// за пределы МКАД.  
var mskad = [  
    [55.785017, 37.841576],  
    [55.861979, 37.765992],  
    [55.898533, 37.635961],  
    [55.888897, 37.48861],
```

```
[55.83251, 37.395275],
[55.744789, 37.370248],
[55.660424, 37.434424],
[55.5922, 37.526366],
[55.574019, 37.683167],
[55.62913, 37.802473],
[55.712203, 37.837121]
],
mkadPolygon = new ymaps.Polygon([mkad], {}, {
  fillColor: '#FFFF00',
  opacity: .4
});
myMap.geoObjects.add(mkadPolygon);
myMap.action.setCorrection(function (tick) {
  var projection = myMap.options.get('projection'),
      tickCenter = projection.fromGlobalPixels(tick.globalPixelCenter, tick.zoom);
  // Если центр карты не входит в нашу область.
  if (!mkadPolygon.geometry.contains(tickCenter)) {
    tick.globalPixelCenter = projection.toGlobalPixels(
      mkadPolygon.geometry.getClosest(tickCenter).position,
      tick.zoom
    );
    tick.duration = 0;
  }
  return tick;
});
```

stop

```
{ } stop()
```

Останавливает выполнение действия на карте.

map.action.Single

Расширяет [IMapAction](#).

Простое движение карты. Движение выполняется сразу после передачи его в `map.action.Manager`.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.action.Single(tick)
```

Создаёт простое (один шаг) движение карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>tick *</code>	—	Тип: Object Параметры движения.
<code>tick.callback</code>	—	Тип: Function Функция, которая будет вызвана после выполнения действия. Принимает в качестве параметра ошибку или null, если действие выполнилось успешно.
<code>tick.center</code>	—	Тип: Number[] Новый центр карты в геокоординатах.

Параметр	Значение по умолчанию	Описание
<code>tick.checkZoomRange</code>	false	<p>Тип: Boolean</p> <p>Флаг, показывающий, нужно ли проверять новый коэффициент масштабирования карты. Если значение флага true, то перед выполнением действия будет запрошен интервал допустимых коэффициентов масштабирования в новой точке. Если указанный коэффициент выходит за интервал допустимых значений, он будет скорректирован. При этом будет изменено значение нового центра карты в глобальных пиксельных координатах.</p>
<code>tick.duration</code>	0	<p>Тип: Number</p> <p>Время исполнения движения, в миллисекундах.</p>
<code>tick.globalPixelCenter</code>	—	<p>Тип: Number[]</p> <p>Новый центр карты в глобальных пикселях. При одновременном задании параметров center и globalPixelCenter, параметр center игнорируется.</p>
<code>tick.timingFunction</code>	'linear'	<p>Тип: String</p> <p>Тайминг-функция.</p>
<code>tick.zoom</code>	—	<p>Тип: Number</p> <p>Новый коэффициент масштабирования карты.</p>

* Обязательный параметр/опция.

Пример:

```
var myCallback = function(err) {
    if (err) {
        throw err;
    }
},
myAction = new ymaps.map.action.Single({
    center: [0, 0],
    zoom: 4,
    duration: 1000,
    timingFunction: 'ease-in',
    checkZoomRange: true,
    callback: myCallback
});

// Действие выполнится сразу после вызова метода execute.
myMap.action.execute(myAction);
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
end	Событие, оповещающее карту об окончании движения. Унаследовано от IMapAction .
tick	Событие, оповещающее карту о следующем шаге движения. Содержит поля: <ul style="list-style-type: none">globalPixelCenter - новый центр карты в глобальных пикселах;zoom - новый коэффициент масштабирования карты;duration - время, которое отводится на выполнение шага;timingFunction - функция, описывающая вид движения. Унаследовано от IMapAction .

Методы

Имя	Возвращает	Описание
begin(mapActionManager)		Запускает движение на исполнение картой. Этот метод автоматически вызывается менеджером движений карты. С момента вызова IMapAction.begin менеджер движений слушает события IMapAction.tick и IMapAction.end и исполняет их. Унаследован от IMapAction .
end()		Останавливает движение. Унаследован от IMapAction .
isActive()	Boolean	Проверяет, выполняется ли движение картой в данный момент.

Описание методов**isActive**

```
{Boolean} isActive()
```

Проверяет, выполняется ли движение картой в данный момент.

Возвращает true - движение в данный момент выполняется, false - нет.

map.addon

map.addon.balloon

Примечание: Конструктор класса `map.addon.balloon` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Модуль, который предоставляет возможность использовать балун у карты. Добавляет карте ([Map](#)) интерфейс [IBalloonOwner](#). При подключении `package.full` (стандартный набор модулей) доступен по умолчанию. Если [Map](#) подключается отдельно, то необходимо явно указать этот модуль в загрузчике. Если [map.addon.balloon](#) подключается отдельно после создания [Map](#), то интерфейс [IBalloonOwner](#) не будет добавлен. Тогда для инициализации менеджера балуна нужно будет использовать метод `map.addon.balloon#get`.

Методы

Методы

Имя	Возвращает	Описание
get(map)	IPopupManager	Возвращает менеджер балуна карты.

Описание методов

get

```
{IPopupManager} get(map)
```

Возвращает менеджер балуна карты.

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Карта

* Обязательный параметр/опция.

Пример:

```
ymaps.map.addon.balloon.get(myMap)
```

map.addon.hint

Примечание: Конструктор класса `map.addon.hint` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Модуль, который предоставляет возможность использовать хинт у карты. Добавляет карте ([Map](#)) интерфейс [IHintOwner](#). При подключении `package.full` (стандартный набор модулей) доступен по умолчанию. Если [Map](#) подключается отдельно, то необходимо явно указать этот модуль в загрузчике. Если [map.addon.hint](#) подключается отдельно после создания [Map](#), то интерфейс [IHintOwner](#) не будет добавлен. Тогда для инициализации менеджера балуна нужно будет использовать метод `map.addon.hint#get`.

Методы

Методы

Имя	Возвращает	Описание
get(map)	IPopupManager	Возвращает менеджер хинта карты.

Описание методов

get

```
{IPopupManager} get(map)
```

Возвращает менеджер хинта карты.

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Карта

* Обязательный параметр/опция.

Пример:

```
ymaps.map.addon.hint.get(myMap)
```

map.Balloon

Расширяет [IBalloonManager](#), [IBalloonSharingManager](#).

Менеджер балуна карты. У каждой карты уже есть собственный менеджер балуна, доступный как `myMap.balloon`. В один момент на карте может быть открыт только один балун, управляемый менеджером. Не создавайте экземпляры этого класса без необходимости.

См.: [Balloon Map.balloon](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.Balloon(map)
```

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Ссылка на объект карты.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
autopanbegin	Начало автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
autopanend	Окончание автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
beforeuserclose	Событие, предшествующее событию Balloon.userclose . Позволяет отменить пользовательское действие посредством вызова метода <code>preventDefault</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
close	Заккрытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. Унаследовано от IPopupManager .
open	Открытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. Унаследовано от IPopupManager .
userclose	Заккрытие балуна пользователем. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .

Методы

Имя	Возвращает	Описание
autoPan()	vow.Promise	Перемещает карту таким образом, чтобы балун оказался видимым. Унаследован от IBalloonManager .
close([force])	vow.Promise	Закрывает инфо-объект. Унаследован от IPopupManager .

Имя	Возвращает	Описание
destroy()		Отключает менеджер инфо-объекта. Унаследован от IPopupManager .
getData()	Object null	Возвращает данные инфо-объекта или null. Унаследован от IPopupManager .
getOptions()	IOptionManager null	Возвращает менеджер опций или null. Унаследован от IPopupManager .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopupManager .
getOverlaySync()	IOverlay null	Возвращает оверлей, если тот существует. Унаследован от IPopupManager .
getPosition()	Number[] null	Возвращает координаты инфо-объекта или 'null'. Унаследован от IPopupManager .
isOpen()	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopupManager .
open([position[, data[, options]])	vow.Promise	Открывает инфо-объект в указанной позиции. Унаследован от IPopupManager .
setData(data)	vow.Promise	Задаёт инфо-объекту новые данные. Унаследован от IPopupManager .
setOptions(options)	vow.Promise	Задаёт инфо-объекту новые опции. Унаследован от IPopupManager .
setPosition(position)	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopupManager .

map.behavior

map.behavior.Manager

Расширяет [ICustomizable](#), [IEventEmitter](#), [IParentOnMap](#).

Менеджер поведений карты. Позволяет включать и отключать поведения. У каждой карты уже есть собственный менеджер поведений, доступный как `map.behaviors`. Не инстанцируйте экземпляры этого класса без необходимости

См.: [Map.behaviors](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.behavior.Manager(map[, behaviors[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: Map Карта.

Параметр	Значение по умолчанию	Описание
behaviors	—	<p>Тип: String[]</p> <p>Список поведений карты, включенных сразу при создании карты. По умолчанию — "drag", "dblClickZoom", "rightMouseButtonMagnifier" для настольных браузеров; "drag", "dblClickZoom" и "multiTouch" — для мобильных.</p> <p>Возможные значения ключей:</p> <ul style="list-style-type: none"> "default" - короткий синоним для включения/отключения поведений карты по умолчанию; "drag" - перемещение карты при нажатой левой кнопке мыши либо одиночным касанием behavior.Drag; "scrollZoom" — изменение масштаба колесом мыши behavior.ScrollZoom; "dblClickZoom" - масштабирование карты двойным щелчком кнопки мыши behavior.DblClickZoom; "multiTouch" — масштабирование карты двойным касанием (например, пальцами на сенсорном экране) behavior.MultiTouch; "rightMouseButtonMagnifier" - увеличение области, выделенной правой кнопкой мыши (только для настольных браузеров), behavior.RightMouseButtonMagnifier; "leftMouseButtonMagnifier" - увеличение области, выделенной левой кнопкой мыши либо одиночным касанием, behavior.LeftMouseButtonMagnifier; "ruler" - измерение расстояния behavior.Ruler. "routeEditor" - редактор маршрутов behavior.RouteEditor; <p>Добавлять и удалять классы поведений можно через хранилище поведений behavior.storage.</p>

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции поведения. Можно задавать следующие опции:</p> <ul style="list-style-type: none"> • опции поведения behavior.Drag с префиксом drag; • опции поведения behavior.ScrollZoom с префиксом scrollZoom; • опции поведения behavior.DblClickZoom с префиксом dblClickZoom; • опции поведения behavior.MultiTouch с префиксом multiTouch; • опции поведения magnifier.RightMouseButtonMagnifier с префиксом rightMouseButtonMagnifier; • опции поведения behavior.LeftMouseButtonMagnifier с префиксом leftMouseButtonMagnifier; • опции поведения behavior.Ruler с префиксом ruler.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none">oldMap - старая карта;newMap - новая карта. Унаследовано от IParentOnMap .
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .

Методы

Имя	Возвращает	Описание
disable(behaviors)	map.behavior.Manager	Отключает поведения на карте.
enable(behaviors)	map.behavior.Manager	Включает поведения на карте.
get(behaviorName)	IBehavior	Возвращает экземпляр поведения по ключу.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
isEnabled(behaviorName)	Boolean	Проверяет, включено ли в настоящий момент поведение.

Описание методов**disable**

```
{map.behavior.Manager} disable(behaviors)
```

Отключает поведения на карте.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
behaviors *	—	Тип: String String[] Список отключаемых поведений.

* Обязательный параметр/опция.

Пример:

```
myMap.behaviors.disable('drag');
```

enable

```
{map.behavior.Manager} enable(behaviors)
```

Включает поведения на карте.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>behaviors</code> *	—	Тип: String String[] Список подключаемых поведений.

* Обязательный параметр/опция.

Пример:

```
myMap.behaviors.enable(['ruler', 'multiTouch']);
```

get

```
{IBehavior} get(behaviorName)
```

Возвращает экземпляр поведения по ключу.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>behaviorName</code> *	—	Тип: String Имя поведения.

* Обязательный параметр/опция.

Пример:

```
myMap.behaviors.get('drag');
```

isEnabled

```
{Boolean} isEnabled(behaviorName)
```

Проверяет, включено ли в настоящий момент поведение.

Возвращает true - поведение включено, false - выключено.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>behaviorName</code> *	—	Тип: String Идентификатор поведения.

* Обязательный параметр/опция.

Пример:

```
// Если поведение "drag" отключено, то подключаем его.  
if (!(myMap.behaviors.isEnabled('drag'))) {  
    myMap.behaviors.enable('drag');  
}
```

map.Container

Расширяет [IDomEventEmitter](#).

Менеджер контейнера карты. У каждой карты уже есть свой собственный менеджер контейнера, доступный как `map.container`. Не инстанцируйте экземпляры этого класса без необходимости.

См.: [Map.container](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.Container(parentElement)
```

Параметры:

Параметр	Значение по умолчанию	Описание
parentElement *	—	Тип: String HTMLElement HTML-элемент, в котором будет создана карта или его id.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .

События

Имя	Описание
beforefullscreenexit	Событие, предшествующее событию <code>fullscreenexit</code> . Если у данного события вызвать метод Event.preventDefault , то последующее событие "fullscreenexit" будет отменено. Экземпляр класса Event .
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
fullscreenenter	<p>Карта переключена в полноэкранный режим. Экземпляр класса Event.</p>
fullscreenexit	<p>Карта вышла из полноэкранного режима. Экземпляр класса Event.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
sizechange	<p>Изменение размера контейнера карты. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>oldSize</code>: <code>Number[]</code>; <code>newSize</code>: <code>Number[]</code>; <code>oldOffset</code>: <code>Number[]</code>; <code>newOffset</code>: <code>Number[]</code>; <code>preservePixelPosition</code>: <code>Boolean</code>.
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
enterFullscreen()		Позволяет переводить карту в полноэкранный режим.
exitFullscreen()		Позволяет вывести карту из полноэкранного режима.
fitToViewport([preservePixelPosition])		Вызывается при изменении размера контейнера карты для того, чтобы карта применила новый размер.
getElement()	<code>HTMLElement</code>	Возвращает HTML-элемент карты.
getOffset()	<code>Number[]</code>	Возвращает смещение контейнера карты в пикселях относительно левого верхнего угла документа.

Имя	Возвращает	Описание
<code>getParentElement()</code>	HTMLElement	Возвращает пользовательский HTML-элемент, в котором была создана карта.
<code>getSize()</code>	Number[]	Возвращает размеры контейнера карты в пикселях.
<code>isFullscreen()</code>	Boolean	Возвращает признак находится ли карта в полноэкранном режиме.

Описание событий

beforefullscreenexit

Событие, предшествующее событию fullscreenexit. Если у данного события вызвать метод `Event.preventDefault`, то последующее событие "fullscreenexit" будет отменено. Экземпляр класса [Event](#).

fullscreenenter

Карта переключена в полноэкранный режим. Экземпляр класса [Event](#).

fullscreenexit

Карта вышла из полноэкранного режима. Экземпляр класса [Event](#).

sizechange

Изменение размера контейнера карты. Экземпляр класса [Event](#). Имена полей, доступных через метод `Event.get`:

- `oldSize`: Number[];
- `newSize`: Number[];
- `oldOffset`: Number[];
- `newOffset`: Number[];
- `preservePixelPosition`: Boolean.

Описание методов

enterFullscreen

```
{ } enterFullscreen()
```

Позволяет переводить карту в полноэкранный режим.

exitFullscreen

```
{ } exitFullscreen()
```

Позволяет вывести карту из полноэкранного режима.

fitToViewport

```
{ } fitToViewport([preservePixelPosition])
```

Вызывается при изменении размера контейнера карты для того, чтобы карта применила новый размер.

Параметры:

Параметр	Значение по умолчанию	Описание
preservePixelPosition	—	Тип: Boolean Сохранять положение центра карты.

Пример:

```
// Изменим размеры контейнера карты.  
map.container.getElement().style.width = '300px';  
// Иницилируем пересчет размеров.  
map.container.fitToViewport();
```

getElement

```
{HTMLElement} getElement()
```

Возвращает HTML-элемент карты.

getOffset

```
{Number[]} getOffset()
```

Возвращает смещение контейнера карты в пикселях относительно левого верхнего угла документа.

getParentElement

```
{HTMLElement} getParentElement()
```

Возвращает пользовательский HTML-элемент, в котором была создана карта.

getSize

```
{Number[]} getSize()
```

Возвращает размеры контейнера карты в пикселях.

isFullscreen

```
{Boolean} isFullscreen()
```

Возвращает признак находится ли карта в полноэкранном режиме.

map.Converter

Класс для преобразования глобальных пиксельных координат точки (считаются от левого верхнего угла мира) в локальные (считаются от левого верхнего угла окна) и наоборот. У каждой карты уже есть собственный конвертер, доступный как `map.converter`. Не实例ируйте экземпляры этого класса без необходимости.

См.: [Map.converter](#)

[Конструктор](#) | [Методы](#)

Конструктор

```
map.Converter(map)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map *</code>	—	Тип: Map Ссылка на карту.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
<code>globalToPage(globalPixelPoint)</code>	<code>Number[]</code>	Преобразовывает глобальные пиксельные координаты точки в локальные.
<code>pageToGlobal(pagePixelPoint)</code>	<code>Number[]</code>	Преобразовывает локальные пиксельные координаты точки в глобальные.

Описание методов

`globalToPage`

```
{Number[]} globalToPage(globalPixelPoint)
```

Преобразовывает глобальные пиксельные координаты точки в локальные.

Возвращает преобразованные координаты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>globalPixelPoint *</code>	—	Тип: <code>Number[]</code> Пиксельные координаты точки, которые необходимо преобразовать.

* Обязательный параметр/опция.

Пример:

```
// Преобразуем географические координаты в пиксели окна браузера
var projection = map.options.get('projection');
console.log(map.converter.globalToPage(
  projection.toGlobalPixels(
    // географические координаты
    [55, 37],
    map.getZoom()
  )
));
```

`pageToGlobal`

```
{Number[]} pageToGlobal(pagePixelPoint)
```

Преобразовывает локальные пиксельные координаты точки в глобальные.

Возвращает преобразованные координаты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>pagePixelPoint</code> *	—	Тип: <code>Number[]</code> Пиксельные координаты точки, которые необходимо преобразовать.

* Обязательный параметр/опция.

Пример:

```
// Преобразуем координаты курсора мыши в геокоординаты
var projection = map.options.get('projection');
$('#map').bind('click', function (e) {
    console.log(projection.fromGlobalPixels(
        map.converter.pageToGlobal([e.pageX, e.pageY]), map.getZoom()
    ).join(' '));
});
```

map.Copyrights

Менеджер размещённой на карте информации об авторских правах. У каждой карты уже есть собственный менеджер информации об авторских правах, доступный как `map.copyrights`. Не инстанцируйте экземпляры этого класса без необходимости.

См.: [Map.copyrights](#)

[Конструктор](#) | [События](#) | [Методы](#)

Конструктор

```
map.Copyrights(map)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: Map Карта.

* Обязательный параметр/опция.

Пример:

```
// Добавим статическую информацию об авторских правах на карту.
var accessor = map.copyrights.add('© Gerardus Mercator');
// ...
// Удаление информации об авторских правах.
accessor.remove();
```

События

Имя	Описание
<code>change</code>	Событие изменения размещённой на карте информации об авторских правах. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"><code>oldCopyrights</code> - старый массив информации об авторских правах;<code>newCopyrights</code> - новый массив информации об авторских правах;

Методы

Имя	Возвращает	Описание
<code>add(customCopyrights)</code>	<code>ICopyrightsAccessor</code>	Добавляет на карту статическую (не зависящую от текущего центра и масштаба карты) информацию об авторских правах.
<code>addProvider(provider)</code>	<code>map.Copyrights</code>	Добавляет нового провайдера информации об авторских правах.
<code>get([point[, zoom]])</code>	<code>vow.Promise</code>	Определяет информацию об авторских правах в указанной точке.
<code>getPromoLink()</code>	<code>String</code>	Возвращает внешнюю ссылку из блока "Открыть в Яндекс.Картах".
<code>removeProvider(provider)</code>	<code>map.Copyrights</code>	Удаляет провайдера информации об авторских правах.

Описание событий**change**

Событие изменения размещённой на карте информации об авторских правах. Экземпляр класса `Event`. Имена полей, доступных через метод `Event.get`:

- `oldCopyrights` - старый массив информации об авторских правах;
- `newCopyrights` - новый массив информации об авторских правах;

Описание методов**add**

```
{ICopyrightsAccessor} add(customCopyrights)
```

Добавляет на карту статическую (не зависящую от текущего центра и масштаба карты) информацию об авторских правах.

Возвращает объект, позволяющий управлять добавленной информацией.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>customCopyrights</code> *	—	<p>Тип: <code>String HTMLElement String[] HTMLElement[]</code></p> <p>Информация об авторских правах в виде строки, DOM-элемента либо массива строк/ DOM-элементов.</p>

* Обязательный параметр/опция.

addProvider

```
{map.Copyrights} addProvider(provider)
```

Добавляет нового провайдера информации об авторских правах.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>provider</code> *	—	Тип: <code>ICopyrightsProvider</code> Провайдер.

* Обязательный параметр/опция.

Пример:

```
// Создание динамического поставщика информации об авторских правах
// что позволит получить актуальную информацию об авторских правах для
// конкретная точка на карте с удаленного сервера.
var myProvider = {
  getCopyrights: function (center, zoom) {
    var deferred = ymaps.vow.defer();
    $.ajax('http://some.server/copyrights/?ll=' + center.join(',') + '&z=' + zoom, {
      // Сервер должен возвращать массив строк
      success: function (res) {
        deferred.resolve(res);
      }
    });
    return deferred.promise();
  }
};
// Добавление поставщика на карту; теперь, когда центр или масштаб меняются,
// менеджер информации об авторских правах на карту будет
// запросить обновленную информацию с удаленного сервера и отобразить ее
// полученные данные автоматически.
map.copyrights.addProvider(myProvider);
```

get

```
{vow.Promise} get([point[, zoom]])
```

Определяет информацию об авторских правах в указанной точке.

Возвращает Promise объект, который будет подтверждён массивом строк/DOM-элементов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>point</code>	—	Тип: <code>Number[]</code> Точка (в географических координатах), для которой требуется определить информацию об авторских правах. Если не указана, берётся текущий центр карты.

Параметр	Значение по умолчанию	Описание
<code>zoom</code>	—	Тип: Number Уровень масштабирования, для которого требуется определить информацию об авторских правах. Если не указан, берётся текущий уровень масштабирования карты.

getPromoLink

```
{String} getPromoLink()
```

Возвращает внешнюю ссылку из блока "Открыть в Яндекс.Картах".

removeProvider

```
{map.Copyrights} removeProvider(provider)
```

Удаляет провайдера информации об авторских правах.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>provider *</code>	—	Тип: ICopyrightsProvider Провайдер.

* Обязательный параметр/опция.

map.GeoObjects

Расширяет [IGeoObjectCollection](#).

Коллекция геообъектов карты. У каждой карты уже есть своя коллекция геообъектов, доступная как `map.geoObjects`. Не инстанцируйте экземпляры этого класса без необходимости.

См.: [Map.geoObjects](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.GeoObjects(map[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map *</code>	—	Тип: Map Карта.

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции коллекции геообъектов. Через опции <code>map.geoObjects</code> можно задавать настройки геообъектов, добавленных на карту:</p> <ul style="list-style-type: none"> • опции кластеров с префиксом <code>cluster</code>. • опции кластеров с префиксом <code>cluster</code>.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>

События

Имя	Описание
add	<p>Добавлен (вставлен) дочерний геообъект. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>index</code>: Integer - индекс добавленного геообъекта; • <code>child</code>: IGeoObject - ссылка на добавленный геообъект. <p>Унаследовано от IGeoObjectCollection.</p>
boundschange	<p>Изменение координат географической области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса Event.</p> <p>Унаследовано от IGeoObjectCollection.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldMap</code> - старая карта; • <code>newMap</code> - новая карта. <p>Унаследовано от IParentOnMap.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
pixelboundschange	<p>Изменение пиксельных координат области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса Event.</p> <p>Унаследовано от IGeoObjectCollection.</p>

Имя	Описание
remove	<p>Удален дочерний геообъект. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> index: Integer — индекс удаленного геообъекта; child: IGeoObject — ссылка на удаленный геообъект. <p>Унаследовано от IGeoObjectCollection.</p>
set	<p>В коллекцию записан новый дочерний геообъект. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> index: Integer - индекс геообъекта; child: IGeoObject - ссылка на новый геообъект; prevChild: IGeoObject — ссылка на предыдущее значение для данного индекса. <p>Унаследовано от IGeoObjectCollection.</p>

Методы

Имя	Возвращает	Описание
add(child[, index])	map.GeoObjects	Добавляет (вставляет) дочерний геообъект в коллекцию.
each(callback[, context])		Для каждого дочернего геообъекта, вызывает функцию-обработчик.
get(index)	IGeoObject	Возвращает дочерний геообъект с заданным индексом.
getBounds()	Number[][] null	Возвращает географические координаты области, охватывающей коллекции и ее дочерние геообъекты.
getIterator()	Iterator	Возвращает итератор по коллекции.
getLength()	Integer	Возвращает длину коллекции.
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getPixelBounds()	Number[][] null	Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и ее дочерние геообъекты.

Имя	Возвращает	Описание
<code>indexOf(object)</code>	Integer	Возвращает индекс дочернего геообъекта. Если геообъект в коллекции не найден, возвращается -1.
<code>remove(child)</code>	<code>map.GeoObjects</code>	Удаляет дочерний геообъект из коллекции.
<code>removeAll()</code>	<code>map.GeoObjects</code>	Очищает коллекцию.
<code>set(index, child)</code>	<code>map.GeoObjects</code>	Записывает в коллекцию новый дочерний геообъект.
<code>splice(index, number)</code>	<code>GeoObjectCollection</code>	Удаляет геообъекты из коллекции. Если необходимо, вставляет на их место другие. Объекты, которые будут вставлены вместо удаляемых, передаются как дополнительные параметры (после параметра <code>number</code>).

Описание методов

add

```
{map.GeoObjects} add(child[, index])
```

Добавляет (вставляет) дочерний геообъект в коллекцию.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>child *</code>	—	Тип: <code>IGeoObject</code> Дочерний геообъект.
<code>index</code>	—	Тип: Integer Индекс, в который добавляется новый объект. По умолчанию объект добавляется в конец коллекции.

* Обязательный параметр/опция.

each

```
{}
```

```
each(callback[, context])
```

Для каждого дочернего геообъекта, вызывает функцию-обработчик.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>callback *</code>	—	Тип: Function Функция-обработчик.
<code>context</code>	—	Тип: Object Контекст выполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

```
// Отображаем индекс геообъекта в коллекции как содержимое его иконки.  
myGeoObjects.events.add(["add", "remove", "set"], function () {  
    this.each(function (el, i) {  
        el.properties.set("iconContent", i);  
    })  
}, myGeoObjects);
```

get

```
{IGeoObject} get(index)
```

Возвращает дочерний геообъект с заданным индексом.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index *</code>	—	Тип: Integer Индекс.

* Обязательный параметр/опция.

getBounds

```
{Number[][]|null} getBounds()
```

Возвращает географические координаты области, охватывающей коллекции и ее дочерние геообъекты.

getIterator

```
{IIterator} getIterator()
```

Возвращает итератор по коллекции.

getLength

```
{Integer} getLength()
```

Возвращает длину коллекции.

getPixelBounds

```
{Number[][]|null} getPixelBounds()
```

Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и её дочерние геообъекты.

indexOf

```
{Integer} indexOf(object)
```

Возвращает индекс дочернего геообъекта. Если геообъект в коллекции не найден, возвращается -1.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>object *</code>	—	Тип: Object Дочерний геообъект.

* Обязательный параметр/опция.

remove

```
{map.GeoObjects} remove(child)
```

Удаляет дочерний геообъект из коллекции.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>child *</code>	—	Тип: <code>IGeoObject</code> Удаляемый геообъект.

* Обязательный параметр/опция.

Пример:

```
// При клике по геообъекту удаляем его из коллекции.  
myGeoObjects.events.add("click", function (e) {  
  if (e.get("target").getParent() == this) {  
    this.remove(e.get("target"));  
  }  
}, myGeoObjects);
```

removeAll

```
{map.GeoObjects} removeAll()
```

Очищает коллекцию.

Возвращает ссылку на себя.

set

```
{map.GeoObjects} set(index, child)
```

Записывает в коллекцию новый дочерний геообъект.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index *</code>	—	Тип: Integer Индекс.
<code>child *</code>	—	Тип: IGeoObject Дочерний геообъект.

* Обязательный параметр/опция.

splice

```
{GeoObjectCollection} splice(index, number)
```

Удаляет геообъекты из коллекции. Если необходимо, вставляет на их место другие. Объекты, которые будут вставлены вместо удаляемых, передаются как дополнительные параметры (после параметра `number`).

Возвращает коллекцию удаленных геообъектов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>index *</code>	—	Тип: Integer Индекс геообъекта, с которого начинается удаление.
<code>number *</code>	—	Тип: Integer Количество удаляемых геообъектов.

* Обязательный параметр/опция.

Пример:

```
// Удаляет второй объект.  
myGeoObjects.splice(1, 1);  
// Вставляет новый объект obj во вторую позицию.  
myGeoObjects.splice(1, 0, obj);  
// Заменяет второй объект на новый объект obj.  
myGeoObjects.splice(1, 1, obj);
```

map.Hint

Расширяет [IHintManager](#), [IHintSharingManager](#).

Менеджер хинта карты. У каждой карты уже есть собственный менеджер хинта, доступный как `myMap.hint`. В один момент на карте может быть открыт только один хинт, управляемый менеджером. Не создавайте экземпляры этого класса без необходимости.

См.: [Hint Map.hint](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.Hint(map)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: Map Ссылка на объект карты.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
<code>close</code>	Заккрытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. Унаследовано от IPopupManager .
<code>open</code>	Открытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. Унаследовано от IPopupManager .

Методы

Имя	Возвращает	Описание
<code>close([force])</code>	vow.Promise	Закрывает инфо-объект. Унаследован от IPopupManager .
<code>destroy()</code>		Отключает менеджер инфо-объекта. Унаследован от IPopupManager .
<code>getData()</code>	<code>Object null</code>	Возвращает данные инфо-объекта или <code>null</code> . Унаследован от IPopupManager .
<code>getOptions()</code>	IOptionManager <code>null</code>	Возвращает менеджер опций или <code>null</code> . Унаследован от IPopupManager .

Имя	Возвращает	Описание
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopupManager .
getOverlaySync()	IOverlay null	Возвращает оверлей, если тот существует. Унаследован от IPopupManager .
getPosition()	Number[] null	Возвращает координаты инфо-объекта или `null`. Унаследован от IPopupManager .
isOpen()	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopupManager .
open([position[, data[, options]])	vow.Promise	Открывает инфо-объект в указанной позиции. Унаследован от IPopupManager .
setData(data)	vow.Promise	Задаёт инфо-объекту новые данные. Унаследован от IPopupManager .
setOptions(options)	vow.Promise	Задаёт инфо-объекту новые опции. Унаследован от IPopupManager .
setPosition(position)	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopupManager .

map.layer

map.layer.Manager

Расширяет [ILayer](#), [IMapObjectCollection](#).

Менеджер слоев карты.

См.: [Map.layers](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.layer.Manager(map[, options])
```

Создает экземпляр класса для работы со слоями карты. У любой карты уже есть свой менеджер слоёв, доступный как `map.layers`. Не инстанцируйте экземпляры этого класса без необходимости.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map *</code>	—	Тип: Map Карта
<code>options</code>	—	Тип: <code>Object</code> Опции слоев карты. Через опции <code>map.layers</code> можно задавать настройки слоев, добавленных на карту. Опции слоев хотспотов задаются с префиксом <code>'hotspotLayer'</code> .
<code>options.trafficImageZIndex</code>	201	Тип: <code>Number</code> Z-index картиночного слоя пробок.
<code>options.trafficInfoZIndex</code>	1	Тип: <code>Number</code> Приоритет хотспотного слоя инфоточек.
<code>options.trafficJamZIndex</code>	0	Тип: <code>Number</code> Приоритет хотспотного слоя пробок.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
<code>options</code>	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
<code>add</code>	Произошло добавление дочернего объекта. Унаследовано от ICollection .
<code>brightnesschange</code>	Событие изменения яркости слоя. Унаследовано от ILayer .
<code>copyrightschange</code>	Событие изменения доступной информации об авторских правах. Унаследовано от ILayer .

Имя	Описание
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. <p>Унаследовано от IChild.</p>
remove	<p>Произошло удаление дочернего объекта.</p> <p>Унаследовано от ICollection.</p>
tileloadchange	<p>Событие изменения состояния загрузки тайла. Поля данных:</p> <ul style="list-style-type: none"> readyTileNumber-количество готовых тайлов. Тайл считается готовым, когда она загружен и отрисован. Тип: Number. totalTileNumber-общее количество видимых тайлов. Тип: Number. <p>Унаследовано от ILayer.</p>
zoomrangechange	<p>Событие изменения доступной информации о диапазоне уровней масштабирования.</p> <p>Унаследовано от ILayer.</p>

Методы

Имя	Возвращает	Описание
add(object)	ICollection	<p>Добавляет в коллекцию дочерний объект.</p> <p>Унаследован от ICollection.</p>
each(callback[, context])		<p>Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.</p>
getBrightness()	Number	<p>Необязательный метод.</p> <p>Унаследован от ILayer.</p>

Имя	Возвращает	Описание
getCopyrights(coords, zoom)	vow.Promise	<p>Необязательный метод. Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования.</p> <p>Унаследован от ILayer.</p>
getIterator()	Iterator	<p>Возвращает итератор по коллекции.</p> <p>Унаследован от ICollection.</p>
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getParent()	IParentOnMap null	<p>Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.</p> <p>Унаследован от IChildOnMap.</p>
getZoomRange(point)	vow.Promise	<p>Необязательный метод. Определяет доступный в указанной точке диапазон уровней масштабирования. Если данные есть, то возвращенный объект-promise будет подтвержден и передаст в качестве результата массив из двух чисел - минимальный и максимальный доступный в точке уровень масштабирования. Если данных нет, то promise отклоняется с указанием ошибки.</p> <p>Унаследован от ILayer.</p>
remove(object)	ICollection	<p>Удаляет из коллекции дочерний объект.</p> <p>Унаследован от ICollection.</p>
setParent(parent)	IChildOnMap	<p>Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта.</p> <p>Унаследован от IChildOnMap.</p>

Описание методов

each

```
{ } each(callback[, context])
```

Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>callback</code> *	—	Тип: Function Функция-обработчик.
<code>context</code>	—	Тип: Object Контекст исполнения функции.

* Обязательный параметр/опция.

map.margin

map.margin.Accessor

Объект, который предоставляет доступ к прямоугольной области в менеджере отступов.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.margin.Accessor(screenArea)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>screenArea</code> *	—	Тип: Object Прямоугольная область, которая задана в экранных координатах. Область описывается в виде объекта, который содержит информацию о смещении от краев карты (left, top, right, bottom) и размеры области (width, height). Значения могут быть заданы в виде процентов от ширины/высоты контейнера карты. Не инстанцируйте экземпляры этого класса без необходимости.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events		Менеджер событий.

События

Имя	Описание
change	Прямоугольная область была изменена.
remove	Аксессор был удален из менеджера отступов.

Методы

Имя	Возвращает	Описание
getArea()	Object	Возвращает прямоугольная область.
remove()	map.margin.Accessor	Удаление прямоугольной области из менеджера отступов.
setArea(screenArea)	map.margin.Accessor	Возвращает ссылку на себя.

Описание полей**events**

```
events
```

Менеджер событий.

Описание событий**change**

Прямоугольная область была изменена.

remove

Аксессор был удален из менеджера отступов.

Описание методов**getArea**

```
{Object} getArea()
```

Возвращает прямоугольная область.

remove

```
{map.margin.Accessor} remove()
```

Удаление прямоугольной области из менеджера отступов.

Возвращает ссылку на себя.

setArea

```
{map.margin.Accessor} setArea(screenArea)
```

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
screenArea *	—	Тип: Object Прямоугольная область, которая задана в экранных координатах. Область описывается в виде объекта, который содержит информацию о смещении от краев карты (left, top, right, bottom) и размеры области (width, height). Значения могут быть заданы в виде процентов от ширины/высоты контейнера карты.

* Обязательный параметр/опция.

Пример:

```
accessor.setArea({  
  top: 50,  
  left: 50,  
  width: 100,  
  height: 100  
});
```

map.margin.Manager

Расширяет [IEventEmitter](#).

Менеджер отступов карты.

Менеджер для расчета оптимальных отступов от краев контейнера карты.

Менеджер принимает на вход описание прямоугольных областей в экранных координатах, которые обозначают занятые области контейнера карты.

Отступы могут учитываться при установке текущей видимой области карты, чтобы добиться наилучшего отображения данных на карте. Данные на карте никогда не попадут под занятые области контейнера карты. `Map#setBound`.

Не инстанцируйте экземпляры этого класса без необходимости.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.margin.Manager()
```

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
change	Изменение отступов карты.

Методы

Имя	Возвращает	Описание
addArea(screenArea)	map.margin.Accessor	Добавление новой прямоугольной области.
destroy()	map.margin.Manager	Уничтожение менеджера отступов.
getMargin()	Number[]	Возвращает текущее значение отступов от краёв карты. Значения в массиве идут в следующем порядке: верхний отступ, правый отступ, нижний отступ, левый отступ.
getOffset()	Number[]	Возвращает разницу (в пикселах) между геометрическим центром карты (без учета отступов) и логическим (с учетом отступов).
setDefaultMargin(margin)		Задаёт отступы от краёв карты по умолчанию.

Описание событий

change

Изменение отступов карты.

Описание методов

addArea

```
{map.margin.Accessor} addArea(screenArea)
```

Добавление новой прямоугольной области.

Возвращает объект, который предоставляет доступ к добавленной области. Добавленную прямоугольную область можно удалить только через полученный объект.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>screenArea</code> *	—	Тип: Object Прямоугольная область, которая задана в локальных координатах. Область описывается в виде объекта, который содержит информацию о смещении от краев карты (<code>left</code> , <code>top</code> , <code>right</code> , <code>bottom</code>) и размеры области (<code>width</code> , <code>height</code>). Значения могут быть заданы в виде процентов от ширины/высоты контейнера карты.

* Обязательный параметр/опция.

Примеры:

1.

```
// Смещение от верхнего левого угла.
map.margin.addArea({
  left: 0,
  top: 0,
  width: 78,
  height: 101
});
console.log(map.margin.getMargin()); // [0, 0, 0, 78]
```

2.

```
// Смещение от правого нижнего угла.
map.margin.addArea({
  right: 50,
  bottom: 50,
  width: 26,
  height: 25
});
console.log(map.margin.getMargin()); // [0, 0, 75, 0]
```

3.

```
// Задание смещения в процентах.
map.margin.addArea({
  left: 50,
  bottom: '1%',
  width: 20,
  height: 20
});
console.log(map.margin.getMargin()); // [0, 0, 26, 0]
```

4.

```
// Задание размеров в процентах.
map.margin.addArea({
  left: 0,
  top: 50,
  width: '100%',
  height: 25
});
console.log(map.margin.getMargin()); // [75, 0, 0, 0]
```

5.

```
// Установка нескольких областей.
map.margin.addArea({
  top: 10,
  left: 10,
  width: 20,
  height: 20
});
```

```
});
map.margin.addArea({
  top: 20,
  right: 40,
  width: 100,
  height: 100
});
map.margin.addArea({
  bottom: 20,
  left: 30,
  width: 120,
  height: 30
});
console.log(map.margin.getMargin()); // [120, 0, 50, 0]
```

destroy

```
{map.margin.Manager} destroy()
```

Уничтожение менеджера отступов.

Возвращает ссылку на себя.

getMargin

```
{Number[]} getMargin()
```

Возвращает текущее значение отступов от краёв карты. Значения в массиве идут в следующем порядке: верхний отступ, правый отступ, нижний отступ, левый отступ.

getOffset

```
{Number[]} getOffset()
```

Возвращает разницу (в пикселах) между геометрическим центром карты (без учета отступов) и логическим (с учетом отступов).

setDefaultMargin

```
{ } setDefaultMargin(margin)
```

Задаёт отступы от краёв карты по умолчанию.

Параметры:

Параметр	Значение по умолчанию	Описание
margin *	—	Тип: Number Number[] Значения отступов в виде одного, двух или четырёх чисел (аналогично заданию margin в CSS).

* Обязательный параметр/опция.

map.pane

map.pane.Manager

Менеджер пейна карты. У каждой карты уже есть свой менеджер пейнов, доступный как map.panes. Не создавайте экземпляры этого класса без необходимости. Список ключей доступных по умолчанию пейнов карты и их zIndex'ы:

- 'ground': [pane.MovablePane](#) (zIndex: 100) — самый нижний пейн, предназначенный для размещения подложки карты;

- 'areas': [pane.MovablePane](#) (zIndex: 200) - панель для объектов с площадью, таких как многоугольники.
- 'shadows': [pane.MovablePane](#) (zIndex: 300) — пейн теней объектов карты, находящихся выше;
- 'places': [pane.MovablePane](#) (zIndex: 400) — пейн точечных объектов, таких как метки;
- 'events': [pane.EventsPane](#) (zIndex: 500) — пейн, предназначенный для слушания событий карты;
- 'overlaps': [pane.MovablePane](#) (zIndex: 600) — пейн для объектов, не требующих использования активных областей для реализации своей интерактивности;
- 'balloon': [pane.MovablePane](#) (zIndex: 700) — пейн балуна;
- 'outerBalloon': [pane.MovablePane](#) (zIndex: 800) — внешний пейн балуна;
- 'controls': [pane.StaticPane](#) (zIndex: 900) — пейн элементов управления карты;
- 'hint': [pane.StaticPane](#) (zIndex: 1100) — пейн всплывающей подсказки;
- 'outerHint': [pane.StaticPane](#) (zIndex: 1200) — внешний пейн всплывающей подсказки;

См.: [Map.panes](#)

[Конструктор](#) | [Методы](#)

Конструктор

```
map.pane.Manager(map)
```

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Карта.

* Обязательный параметр/опция.

Пример:

```
// Добавим водяной знак над контейнером карты.
// Для этого изменим background у контейнера событий.
map.panes.get('events').getElement().style.backgroundImage = 'my/background/image';
```

Методы

Имя	Возвращает	Описание
append(key, pane)		Добавляет новый пейн в карту. Ключ добавляемого пейна должен быть уникальным в рамках текущего набора ключей пейнов карты.
destroy()		Деструктор.
get(key)	IPane null	Возвращает пейн карты с заданным ключом, либо null, если такой пейн в данной карте отсутствует.
getLower()	String	Принимает в качестве аргументов ключи пейнов карты и возвращает ключ самого нижнего пейна из переданного набора. Если ключи не заданы, поиск выполняется по всем пейнам карты.

Имя	Возвращает	Описание
<code>getUpper()</code>	String	Принимает в качестве аргументов ключи пейнов карты и возвращает ключ самого верхнего пейна из переданного набора. Если ключи не заданы, поиск выполняется по всем пейнам карты.
<code>insertBefore(key, pane, referenceKey)</code>		Вставляет новый пейн перед другим пейном карты. Ключ добавляемого пейна должен быть уникальным в рамках текущего набора ключей пейнов карты.
<code>remove(pane)</code>		Удаляет пейн из карты.

Описание методов

append

```
{ } append(key, pane)
```

Добавляет новый пейн в карту. Ключ добавляемого пейна должен быть уникальным в рамках текущего набора ключей пейнов карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key</code> *	—	Тип: String Ключ добавляемого пейна.
<code>pane</code> *	—	Тип: <code>IPane</code> Добавляемый пейн.

* Обязательный параметр/опция.

destroy

```
{ } destroy()
```

Деструктор.

get

```
{IPane|null} get(key)
```

Возвращает пейн карты с заданным ключом, либо null, если такой пейн в данной карте отсутствует.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key *</code>	—	Тип: String Ключ пейна.

* Обязательный параметр/опция.

getLower

```
{String} getLower()
```

Принимает в качестве аргументов ключи пейнов карты и возвращает ключ самого нижнего пейна из переданного набора. Если ключи не заданы, поиск выполняется по всем пейнам карты.

Возвращает ключ самого нижнего пейна карты.

getUpper

```
{String} getUpper()
```

Принимает в качестве аргументов ключи пейнов карты и возвращает ключ самого верхнего пейна из переданного набора. Если ключи не заданы, поиск выполняется по всем пейнам карты.

Возвращает ключ самого верхнего пейна карты.

insertBefore

```
{ } insertBefore(key, pane, referenceKey)
```

Вставляет новый пейн перед другим пейном карты. Ключ добавляемого пейна должен быть уникальным в рамках текущего набора ключей пейнов карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key *</code>	—	Тип: String Ключ добавляемого пейна.
<code>pane *</code>	—	Тип: <code>IPane</code> Добавляемый пейн.
<code>referenceKey *</code>	—	Тип: String Ключ пейна, перед которым нужно добавить новый пейн.

* Обязательный параметр/опция.

remove

```
{ } remove(pane)
```

Удаляет пейн из карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>pane</code> *	—	Тип: IPane Удаляемый пейн.

* Обязательный параметр/опция.

map.ZoomRange

Расширяет [IEventEmitter](#).

Менеджер коэффициентов масштабирования карты. У каждой карты уже есть свой менеджер коэффициентов масштабирования, доступный как `map.zoomRange`. Не инстанцируйте экземпляры этого класса без необходимости.

См.: [Map.zoomRange](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
map.ZoomRange(map, constraints)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: Map Карта.
<code>constraints</code> *	—	Тип: <code>Number[]</code> Массив, содержащий минимальный и максимальный зум карты.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
<code>change</code>	Произошло изменение диапазона масштабов карты.

Методы

Имя	Возвращает	Описание
<code>get([coords])</code>	vow.Promise	Определяет доступный в указанной точке диапазон уровней масштабирования.

Имя	Возвращает	Описание
<code>getCurrent()</code>	<code>Number[]</code>	Возвращает текущее (последнее полученное) значение минимального и максимального уровня масштабирования карты.

Описание событий

change

Произошло изменение диапазона масштабов карты.

Описание методов

get

```
{vow.Promise} get([coords])
```

Определяет доступный в указанной точке диапазон уровней масштабирования.

Возвращает Promise объект, который будет подтверждён массивом из двух чисел — максимальный и минимальный уровень масштабирования в данной точке.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>coords</code>	—	Тип: <code>Number[]</code> Координаты точки. Если не указана, берётся текущий центр карты.

Пример:

```
// Найдем координаты офиса компании Яндекс и покажем его на карте
// на максимально возможном уровне масштабирования.
ymaps.geocode('Москва, улица Льва Толстого, 16').then(function (res) {
  var coords = res.geoObjects.get(0).geometry.getCoordinates();
  map.zoomRange.get(coords).then(function (range) {
    map.setCenter(coords, range[1]);
  });
});
```

getCurrent

```
{Number[]} getCurrent()
```

Возвращает текущее (последнее полученное) значение минимального и максимального уровня масштабирования карты.

MapEvent

Расширяет [Event](#).

Объект, описывающий произошедшее над картой событие. Имена полей, доступных через метод [Event.get](#):

- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;

- `pagePixels` - координаты события в глобальных пикселях от левого верхнего угла страницы (также доступно под именем `position`);
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

[Конструктор](#) | [Методы](#)

Конструктор

```
MapEvent(originalEvent[, sourceEvent])
```

Параметры:

Параметр	Значение по умолчанию	Описание
originalEvent *	—	Тип: Object Данные, связанные с событием. Должны содержать поле "map" указывающее на карту, над которой произошло событие.
sourceEvent	—	Тип: IEvent Исходное событие.

* Обязательный параметр/опция.

Пример:

```
// Откроем балун в точке клика по карте.  
map.events.add('click', function (e) {  
    map.balloon.open(e.get('coords'), 'Щелк!');  
});
```

Методы

Имя	Возвращает	Описание
allowMapEvent()		Разрешает распространение данного события на карту. Унаследован от IEvent .
callMethod(name)	Object	Вызывает указанный метод. Операция эквивалентна поиску поля через <code>get</code> и вызову с передачей в качестве контекста <code>originalEvent</code> . Все аргументы после первого передаются в качестве параметров вызываемому методу. Унаследован от Event .

Имя	Возвращает	Описание
get(name)	Object	<p>Возвращает значение поля из <code>originalEvent</code>. <code>originalEvent</code> всегда имеет следующие поля:</p> <ul style="list-style-type: none"> <code>type</code> - строковый тип события; <code>target</code> - ссылка на объект, породивший событие. <p>Унаследован от Event.</p>
getSourceEvent()	IEvent null	<p>Возвращает исходное событие.</p> <p>Унаследован от IEvent.</p>
isDefaultPrevented()	Boolean	<p>Проверяет, отменена ли реакция по умолчанию на событие в рамках системы событий API Яндекс.Карт.</p> <p>Унаследован от Event.</p>
isImmediatePropagationStopped()	Boolean	<p>Определяет, прекращено ли распространение события в системе событий API Яндекс.Карт.</p> <p>Унаследован от Event.</p>
isMapEventAllowed()	Boolean	<p>Возвращает true, если событие карты разрешено.</p> <p>Унаследован от IEvent.</p>
isPropagationStopped()	Boolean	<p>Проверяет, остановлено ли распространение события по иерархии объектов и коллекций в системе событий API Яндекс.Карт.</p> <p>Унаследован от Event.</p>
preventDefault()		<p>Отменяет реакцию по умолчанию на событие в рамках системы событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву.</p> <p>Унаследован от Event.</p>
stopImmediatePropagation()		<p>Прекращает распространение события в системе событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву.</p> <p>Унаследован от Event.</p>

Имя	Возвращает	Описание
stopPropagation()		Прекращает распространение события по иерархии объектов и коллекций в системе событий API Яндекс.Карт. Вызов этого метода никак не влияет на распространение исходного DOM-события (если оно есть) по DOM-дереву. Унаследован от Event .

mapType

mapType.storage

Статический объект.

Экземпляр класса [util.Storage](#)

Хранилище типов карты.

Методы

Пример:

```
// Добавим свой тип карты.  
  
// Допустим, у нас есть свой слой карты.  
var myLayer = function () {  
    return new ymaps.Layer('http://some.server/?%c');  
}  
// Добавляем его в хранилище слоёв.  
ymaps.layer.storage.add('my#layer', myLayer);  
// Создаём свой тип карты, состоящий из одного слоя.  
var myType = new ymaps.MapType('My map type', ['my#layer']);  
// Добавляем его в хранилище типов карты.  
ymaps.mapType.storage.add('my#mapType', myType);  
// Теперь можем установить карте созданный тип.  
map.setType('my#type');
```

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.

MapType

Тип карты.

[Конструктор](#) | [Методы](#)

Конструктор

```
MapType(name, layers)
```

Создает экземпляр типа карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>name</code> *	—	Тип: String Название типа.
<code>layers</code> *	—	Тип: Function[] String[] Массив, содержащий конструкторы слоев или ключи.

* Обязательный параметр/опция.

Пример:

```
// Создадим собственный тип карты, состоящий из спутниковых снимков MapQuest
// с наложенным поверх Яндекс.Гибридом.

// Класс тайлов MapQuest
var MQLayer = function () {
    var layer = new ymaps.Layer('http://oatile.d.mqcdn.com/naip/%z/%x/%y.jpg');
    // Копирайты
    layer.getCopyrights = function () {
        return ymaps.vow.resolve('Данный, изображения и информация о карте предоставлены MapQuest, Open Street
        Мари контрибьютерами, CC-BY-SA');
    };
    // Доступные уровни зума
    layer.getZoomRange = function () {
        return ymaps.vow.resolve([0, 18]);
    };
    return layer;
};

// Добавляем слой с ключем.
ymaps.layer.storage.add('mq#aerial', MQLayer);
// Создаем тип карты, состоящий из слова 'mq#aerial' и 'yandex#skeleton'
var myMapType = new ymaps.MapType('MQ + Ya', ['mq#aerial', 'yandex#skeleton']);
// Добавим в хранилище типов карты
ymaps.mapType.storage.add('mq_ya#hybrid', myMapType);
// Теперь мы можем выставлять этот тип для любой карты
map.setType('mq_ya#hybrid');
```

Методы

Имя	Возвращает	Описание
<code>getLayers()</code>	Function[] String[]	Возвращает список слоев данного типа карты - массив конструкторов или ключей для слоев.
<code>getName()</code>	String	Возвращает название типа карты.

Описание методов

getLayers

```
{Function[]|String[]} getLayers()
```

Возвращает список слоев данного типа карты - массив конструкторов или ключей для слоев.

getName

```
{String} getName()
```

Возвращает название типа карты.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип:

* Обязательный параметр/опция.

meta

Статический объект.

Информация об API.

[Поля](#)

Поля

Имя	Тип	Описание
<code>coordinatesOrder</code>	String	Порядок координат, используемый в API. Возможные значения: <ul style="list-style-type: none">• <code>latlong</code> — [широта, долгота];• <code>longlat</code> — [долгота, широта]. Задается GET-параметром "coordorder" при подключении API. Подробнее про параметры подключения API
<code>countryCode</code>	String	Двузначный код страны. Возвращается в формате ISO 3166-1 . Задается GET-параметром "lang" при подключении API. Подробнее про параметры подключения API
<code>languageCode</code>	String	Двузначный код языка. Возвращается в формате ISO 639-1 . Задается GET-параметром "lang" при подключении API. Подробнее про параметры подключения API
<code>mode</code>	String	Режим работы API Яндекс.Карт. Возможные значения: <ul style="list-style-type: none">• <code>release</code>;• <code>debug</code>. Задается GET-параметром "mode" при подключении API. Подробнее про параметры подключения API

Имя	Тип	Описание
<code>ns</code>	Object	Ссылка на пространство имен API Яндекс.Карт. Имеет значение независимо от значения параметра "ns" при подключении API.
<code>version</code>	String	Версия API Яндекс.Карт.

Описание полей

coordinatesOrder

```
{String} coordinatesOrder
```

Порядок координат, используемый в API. Возможные значения:

- `latlong` — [широта, долгота];
- `longlat` — [долгота, широта].

Задается GET-параметром "coordorder" при подключении API.

[Подробнее про параметры подключения API](#)

countryCode

```
{String} countryCode
```

Двузначный код страны. Возвращается в формате [ISO 3166-1](#).

Задается GET-параметром "lang" при подключении API.

[Подробнее про параметры подключения API](#)

languageCode

```
{String} languageCode
```

Двузначный код языка. Возвращается в формате [ISO 639-1](#).

Задается GET-параметром "lang" при подключении API.

[Подробнее про параметры подключения API](#)

mode

```
{String} mode
```

Режим работы API Яндекс.Карт. Возможные значения:

- `release`;
- `debug`.

Задается GET-параметром "mode" при подключении API.

[Подробнее про параметры подключения API](#)

ns

```
{Object} ns
```

Ссылка на пространство имен API Яндекс.Карт.

Имеет значение независимо от значения параметра "ns" при подключении API.

version

```
{String} version
```

Версия API Яндекс.Карт.

modules

Статический объект.

Модульная система, на основе которой работает API Яндекс.Карт.

API Яндекс.Карт состоит из большого количества взаимосвязанных модулей. Модуль - это некоторая программная единица. К примеру, класс, конкретная реализация класса, статический объект или функция. Модульная система гарантирует, что при инициализации конкретного модуля все необходимые ему для работы модули уже будут проинициализированы.

Модульная система предоставляет асинхронный доступ, так как может потребоваться произвести загрузку недостающих модулей.

В модульную систему можно добавлять свои собственные модули.

modules.define

Статическая функция.

Определение модуля в модульной системе.

Примечание: Рекомендуется создавать собственные модули в своем пространстве имен, чтобы случайно не заменить необходимые для работы API модули.

Возвращает ссылку на себя.

```
{ modules } modules.define(module[, depends, resolveCallback[, context]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>module</code> *	—	Тип: String Название модуля.
<code>depends</code>	—	Тип: String[] Массив названий необходимых модулей. Этот аргумент может быть пропущен.

Параметр	Значение по умолчанию	Описание
<code>resolveCallback *</code>	—	<p>Тип: Function</p> <p>Функция, которая определяет модуль. Первым аргументом в <code>resolveCallback</code> поступит <code>provide</code>-функция, в которую нужно будет передать модуль. Вызов <code>provide</code>-функции может быть отложен во времени. Последующие аргументы - это модули, указанные в зависимостях. Порядок модулей будет соответствовать порядку в массиве <code>depends</code>. В случае если модуль не может быть зарезолвен, то необходимо об этом уведомить модульную систему. Это можно сделать передав в <code>provide</code>-функцию второй аргумент. Вторым аргументом будет передан в <code>errorCallback</code> и в <code>promise</code> как ошибка в запрос модуля. Впоследствии модуль можно будет запросить повторно.</p>
<code>context</code>	—	<p>Тип: Object</p> <p>Контекст исполнения функции.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
// Задание собственного модуля.
// 'plugin.*' - это собственное пространство имен.
ymaps.modules.define('CustomModule', function (provide) {
    var CustomModule = function (defValue) {
        this.field = defValue;
    };
    provide(CustomModule);
});
// Запрос модулей.
ymaps.modules.require(['CustomModule'])
    .spread(
        function (CustomModule) {
            // ...
        },
        this
    );
```

2.

```
// Задание собственного асинхронного модуля.
ymaps.modules.define('CustomAsyncModule', function (provide) {
    // Для работы модуля необходимо загрузить внешний скрипт.
    $.getScript("ajax/test.js")
        .done(function( script, textStatus ) {
            function CustomAsyncModule () {
                // ...
            }
            // Вызов provide-функции после загрузки скрипта.
            provide(CustomAsyncModule);
        });
});
// Запрос модулей.
ymaps.modules.require(['CustomAsyncModule'])
    .spread(
        function (CustomAsyncModule) {
            // ...
        },
        this
    );
```

3.

```
// Создание пользовательского асинхронного модуля с учетом ошибки.
ymaps.modules.define('plugin.CustomModule', function (provide) {
    $.getScript( "ajax/test.js" )
        .done(function( script, textStatus ) {
            // Обработка успешной загрузки скрипта.
            provide({ a: 1 });
        })
        .fail(function( jqxhr, settings, exception ) {
            // Уведомление модульной системы о том, что модуль не может быть подготовлен прямо сейчас.
            provide(null, new Error('Обишка при загрузке'));
        });
});

// Запрос модулей с обработкой ошибок.
ymaps.modules.require(['plugin.CustomModule'])
    .spread(
        function (CustomModule) {
            // ...
        },
        function (error) {
            console.log(error.messsage); // "Error when loading".
            // Код обработки ошибки получения модуля. Модуль можно запросить повторно.
        },
        this
    );
```

4.

```
// Определение пользовательского модуля с зависимостями.
ymaps.modules.define('CustomLayoutModule', [
    'templateLayoutFactory',
    'layout.storage'
], function (provide, templateLayoutFactory, layoutStorage) {
    var customLayoutClass = templateLayoutFactory.createClass(
        '<div>My Layout</div>',
        {
            build: function () {
                customLayoutClass.superclass.build.call(this);
                // ...
            },
            clear: function () {
                // ...
                customLayoutClass.superclass.clear.call(this);
            }
        }
    );
    layoutStorage.add('customLayout', customLayoutClass);
    provide(customLayoutClass);
});
// Запрос пользовательского макета
ymaps.modules.require(['CustomLayoutModule'])
    .spread(
        function (CustomLayoutModule) {
            // ...
        },
        this
    );
```

modules.isDefined

Статическая функция.

Проверка доступности модуля по названию.

Возвращает true, если модуль определен, false — иначе.

```
{ Boolean } modules.isDefined(module)
```

Параметры:

Параметр	Значение по умолчанию	Описание
module *	—	Тип: String Модуль

* Обязательный параметр/опция.

Пример:

```
if (ymaps.modules.isDefined('plugin.CustomModule')) {
  ymaps.modules.require(['plugin.CustomModule'])
    .spread(
      function (CustomModule) {
        // ...
      });
}
```

modules.require

Статическая функция.

Запрос получения модулей.

Возвращает Promise объект, который подтверждается запрошенными модулями. Либо отклоняется в случае, если произошла ошибка. К примеру, один из запрошенных модулей отсутствует в модульной системе.

```
{ vow.Promise } modules.require(modules[, successCallback[, errorCallback[, context]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>modules *</code>	—	Тип: String String[] Название модуля или массив названий модулей.
<code>successCallback</code>	—	Тип: Function Функция, которая будет вызвана после получения всех модулей. В функцию как аргументы будут переданы запрошенные сущности. Порядок аргументов будет соответствовать порядку в массиве <code>modules</code> .
<code>errorCallback</code>	—	Тип: Function Функция, которая будет вызвана в случае ошибки.
<code>context</code>	—	Тип: Object Контекст исполнения функции обратного вызова.

* Обязательный параметр/опция.

Примеры:

1.

```
// Запросим модули "Map" и "Placemark" с помощью объекта Promise.
ymaps.modules.require(['Map', 'Placemark'])
  .spread(
    function (Map, Placemark) {
      var myMap = new Map("map", {
        center: [55.72, 37.64],
        zoom: 5
      });
      myMap.geoObjects.add(
        new Placemark(myMap.getCenter())
      );
    },
```

```
function (error) {  
    // Обработка ошибок.  
},  
this  
);
```

2.

```
// Запрашиваем модуль 'Map'.  
ymaps.modules.require('Map')  
    .spread(  
        function (Map) {  
            var myMap = new Map("map", {  
                center: [55.72, 37.64],  
                zoom: 5  
            });  
        },  
        this  
    );
```

Monitor

Объект, отслеживающий изменения определенных полей данных на заданном менеджере данных. Также может использоваться для отслеживания изменений опций.

[Конструктор](#) | [Методы](#)

Конструктор

```
Monitor(dataManager)
```

Параметры:

Параметр	Значение по умолчанию	Описание
dataManager *	—	Тип: IDataManager IOptionManager Менеджер данных.

* Обязательный параметр/опция.

Пример:

```
// Отслеживание изменений опции метки.  
var placemark = new ymaps.Placemark([0, 0],  
    optionMonitor = new ymaps.Monitor(placemark.options);  
optionMonitor.add("cursor", function (newValue) {  
    alert("cursor: " + newValue);  
});  
myMap.geoObjects.add(placemark);  
// Выведет строку "cursor: arrow".  
myMap.options.set({  
    geoObjectCursor: "arrow"  
});
```

Методы

Имя	Возвращает	Описание
add(name, changeCallback[, context[, params]])	Monitor	Включает наблюдение за определенным полем, либо группой полей данных.
forceChange()	Monitor	Иницирует проверку изменения значений наблюдаемых полей данных.

Имя	Возвращает	Описание
<code>get(name)</code>	Object	Возвращает текущее значение одного из наблюдаемых полей данных.
<code>remove(name)</code>	Monitor	Отключает наблюдение за определенным полем, либо группой полей данных.
<code>removeAll()</code>	Monitor	Отключает наблюдение для всех полей данных.

Описание методов

add

```
{Monitor} add(name, changeCallback[, context[, params]])
```

Включает наблюдение за определенным полем, либо группой полей данных.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>name *</code>	—	Тип: String[String[]] Имя, либо массив имен полей данных, за которыми устанавливается наблюдение.
<code>changeCallback *</code>	—	Тип: Function Обработчик изменения поля данных, либо одного из полей данных группы.
<code>context</code>	—	Тип: Object Контекст исполнения обработчика изменений данных, а также опциональных обработчиков.
<code>params</code>	—	Тип: Object Опциональные параметры.

Параметр	Значение по умолчанию	Описание
params.compareCallback	—	<p>Тип: Function</p> <p>Обработчик сравнивающий старое и новое значения поля данных. Принимает два аргумента: старое значение, новое значение. Менее приоритетен по отношению к обработчикам заданным с помощью параметра <code>compareCallbacks</code>.</p>
params.compareCallbacks	—	<p>Тип: Object</p> <p>Хэш вида {имя поля данных: ссылка на обработчик}. Данный параметр позволяет задать индивидуальные обработчики сравнения значений для различных полей данных в группе.</p>
params.defaultValue	—	<p>Тип: Object</p> <p>Значение по умолчанию, используемое в том случае, если поле данных не определено.</p>
params.defaultValues	—	<p>Тип: Object</p> <p>Хэш вида {имя поля данных: значение по умолчанию}. Данный параметр позволяет задать индивидуальные значения по умолчанию для различных полей данных в группе.</p>
params.resolveCallback	—	<p>Тип: Function</p> <p>Обработчик разрешающий значение поля данных. Принимает два аргумента: имя поля данных, ссылку на менеджер данных. Менее приоритетен по отношению к обработчикам заданным с помощью параметра <code>resolveCallbacks</code>.</p>

Параметр	Значение по умолчанию	Описание
<code>params.resolveCallbacks</code>	—	Тип: Object Хэш вида {имя поля данных: ссылка на обработчик}. Данный параметр позволяет задать индивидуальные обработчики разрешения значений для различных полей данных в группе.

* Обязательный параметр/опция.

forceChange

```
{Monitor} forceChange()
```

Иницирует проверку изменения значений наблюдаемых полей данных.

Возвращает ссылку на себя.

get

```
{Object} get(name)
```

Возвращает текущее значение одного из наблюдаемых полей данных.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>name *</code>	—	Тип: String Имя поля данных.

* Обязательный параметр/опция.

remove

```
{Monitor} remove(name)
```

Отключает наблюдение за определенным полем, либо группой полей данных.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>name *</code>	—	Тип: String String[] Имя, либо массив имен полей данных, для которых отключается наблюдение.

* Обязательный параметр/опция.

removeAll

```
{Monitor} removeAll()
```

Отключает наблюдение для всех полей данных.

Возвращает ссылку на себя.

multiRouter**multiRouter.bicycle****multiRouter.bicycle.Path**

Примечание: Конструктор класса `multiRouter.bicycle.Path` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение пути мультимаршрута на велосипеде. Единичный маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение пути мультимаршрута на велосипеде.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.bicycle.PathModel	Модель данных пути мультимаршрута.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
update	Обновление отображения пути. Экземпляр класса Event .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

model

```
{multiRouter.bicycle.PathModel} model
```

Модель данных пути мультимаршрута.

Описание событий

update

Обновление отображения пути. Экземпляр класса [Event](#).

multiRouter.bicycle.PathModel

Примечание: Конструктор класса multiRouter.bicycle.PathModel скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных сегмента пути велосипедного маршрута. Единичный маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

Поля

Создает модель данных велосипедного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

multiRouter.bicycle.Route

Примечание: Конструктор класса multiRouter.bicycle.Route скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение единичного велосипедного маршрута. Мультимаршрут может состоять из нескольких отдельных маршрутов.

Поля | События | Методы

Создает отображение единичного велосипедного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .

Имя	Тип	Описание
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
balloonclose	Закрытие балуна.
balloonopen	Открытие балуна.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
update	Обновление отображения маршрута. Экземпляр класса Event .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .

Имя	Возвращает	Описание
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание событий**balloonclose**

Закрытие балуна.

balloonopen

Открытие балуна.

update

Обновление отображения маршрута. Экземпляр класса [Event](#).

multiRouter.bicycle.RouteModel

Примечание: Конструктор класса multiRouter.bicycle.RouteModel скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных единичного велосипедного маршрута. Мультимаршрут может состоять из нескольких отдельных маршрутов.

Поля

Создает модель данных единичного велосипедного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventManager .

multiRouter.bicycle.Segment

Примечание: Конструктор класса multiRouter.bicycle.Segment скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение сегмента велосипедного маршрута. Сегментом велосипедного маршрута называется часть пути от одного маневра до другого.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение сегмента велосипедного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IParentOnMap .

Имя	Описание
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление отображения сегмента. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание событий

update

Обновление отображения сегмента. Экземпляр класса [Event](#).

multiRouter.bicycle.SegmentModel

Примечание: Конструктор класса multiRouter.bicycle.SegmentModel скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventEmitter](#).

Модель данных сегмента пути велосипедного маршрута. Сегментом велосипедного маршрута называется часть пути от одного маневра до другого.

Поля

Создает модель данных сегмента пути велосипедного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

multiRouter.driving

multiRouter.driving.Path

Примечание: Конструктор класса multiRouter.driving.Path скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение пути автомобильного мультимаршрута. Единичный маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение пути автомобильного мультимаршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.driving.PathModel	Модель данных пути мультимаршрута.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	data.Manager	Модель данных пути мультимаршрута. Доступны следующие поля: <ul style="list-style-type: none"> index: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута; type: String - идентификатор типа маршрута, принимающий значение "driving" для автомобильных маршрутов; distance: Object - объект с полями text и value, описывающий протяженность пути в метрах; duration: Object - объект с полями text и value, описывающий время движения по пути в секундах; durationInTraffic: Object - объект с полями text и value, описывающий время проезда пути в секундах с учетом пробок; coordinates: Number[][] - координаты всех точек пути; encodedCoordinates: String - строка закодированных в base64 координат всех точек пути.

Имя	Тип	Описание
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
update	Обновление отображения пути. Экземпляр класса Event .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .

Имя	Возвращает	Описание
<code>getParent()</code>	<code>IParentOnMap</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>getSegments()</code>	<code>GeoObjectCollection</code>	Возвращает дочернюю коллекцию сегментов, из которых состоит путь.
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание полей

model

```
{multiRouter.driving.PathModel} model
```

Модель данных пути мультимаршрута.

properties

```
{data.Manager} properties
```

Модель данных пути мультимаршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута;
- `type`: String - идентификатор типа маршрута, принимающий значение "driving" для автомобильных маршрутов;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность пути в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время движения по пути в секундах;
- `durationInTraffic`: Object - объект с полями `text` и `value`, описывающий время проезда пути в секундах с учетом пробок;
- `coordinates`: Number[][] - координаты всех точек пути;
- `encodedCoordinates`: String - строка закодированных в base64 координат всех точек пути.

Описание событий

update

Обновление отображения пути. Экземпляр класса [Event](#).

Описание методов

getSegments

```
{GeoObjectCollection} getSegments()
```

Возвращает дочернюю коллекцию сегментов, из которых состоит путь.

multiRouter.driving.PathModel

Примечание: Конструктор класса multiRouter.driving.PathModel скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных пути автомобильного маршрута. Единичный маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных пути автомобильного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventManager .
properties	data.Manager	Модель данных пути мультимаршрута. Доступны следующие поля: <ul style="list-style-type: none"> index: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута; type: String - идентификатор типа маршрута, принимающий значение "driving" для автомобильных маршрутов; distance: Object - объект с полями text и value, описывающий протяженность пути в метрах; duration: Object - объект с полями text и value, описывающий время движения по пути в секундах; durationInTraffic: Object - объект с полями text и value, описывающий время проезда пути в секундах с учетом пробок; coordinates: Number[][] - координаты всех точек пути; encodedCoordinates: String - строка закодированных в base64 координат всех точек пути.
route	multiRouter.driving.RouteModel	Ссылка на родительскую модель маршрута.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> segmentsChange: Boolean – признак изменений в наборе сегментов.

Методы

Имя	Возвращает	Описание
<code>destroy()</code>		Уничтожает модель.
<code>getSegments()</code>	<code>multiRouter.driving.SegmentModel[]</code>	Возвращает массив сегментов пути.
<code>getType()</code>	String	Возвращает идентификатор типа пути маршрута. Для автомобильных маршрутов возвращается строка "driving".
<code>update(pathJson)</code>		Обновляет состояние модели.

Описание полей**properties**

```
{data.Manager} properties
```

Модель данных пути мультимаршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута;
- `type`: String - идентификатор типа маршрута, принимающий значение "driving" для автомобильных маршрутов;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность пути в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время движения по пути в секундах;
- `durationInTraffic`: Object - объект с полями `text` и `value`, описывающий время проезда пути в секундах с учетом пробок;
- `coordinates`: Number[][] - координаты всех точек пути;
- `encodedCoordinates`: String - строка закодированных в base64 координат всех точек пути.

route

```
{multiRouter.driving.RouteModel} route
```

Ссылка на родительскую модель маршрута.

Описание событий**update**

Обновление модели новыми данными. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `segmentsChange`: Boolean – признак изменений в наборе сегментов.

Описание методов**destroy**

```
{ } destroy()
```

Уничтожает модель.

getSegments

```
{multiRouter.driving.SegmentModel[]} getSegments()
```

Возвращает массив сегментов пути.

getType

```
{String} getType()
```

Возвращает идентификатор типа пути маршрута. Для автомобильных маршрутов возвращается строка "driving".

update

```
{ } update(pathJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
pathJson *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

multiRouter.driving.Route

Примечание: Конструктор класса `multiRouter.driving.Route` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение единичного автомобильного маршрута. Мультимаршрут может состоять из нескольких отдельных маршрутов.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение единичного автомобильного маршрута.

Поля

Имя	Тип	Описание
balloon	IMultiRouterRouteBalloon	Балун маршрута.
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.driving.RouteModel	Модель данных единичного маршрута.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

Имя	Тип	Описание
properties	data.Manager	<p>Данные маршрута. Доступны следующие поля:</p> <ul style="list-style-type: none"> • <code>index</code>: Integer - порядковый номер маршрута в мультимаршруте; • <code>type</code>: String - идентификатор типа маршрута, принимающий значение "driving" для автомобильных маршрутов; • <code>blocked</code>: Boolean - признак того, что маршрут содержит перекрытые участки; • <code>distance</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий протяженность маршрута в метрах; • <code>duration</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий время проезда маршрута в секундах; • <code>durationInTraffic</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий время проезда пути в секундах с учетом пробок; • <code>boundedBy</code>: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.
state	IDataManager	<p>Состояние геообъекта.</p> <p>Унаследовано от IGeoObject.</p>

События

Имя	Описание
balloonclose	Закрытие балуна.
balloonopen	Открытие балуна.
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
update	Обновление отображения маршрута. Экземпляр класса Event .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getPaths()	GeoObjectCollection	Возвращает дочернюю коллекцию путей, из которых состоит маршрут.

Имя	Возвращает	Описание
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

balloon

```
{IMultiRouterRouteBalloon} balloon
```

Балун маршрута.

model

```
{multiRouter.driving.RouteModel} model
```

Модель данных единичного маршрута.

properties

```
{data.Manager} properties
```

Данные маршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер маршрута в мультимаршруте;
- `type`: String - идентификатор типа маршрута, принимающий значение "driving" для автомобильных маршрутов;
- `blocked`: Boolean - признак того, что маршрут содержит перекрытые участки;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность маршрута в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время проезда маршрута в секундах;
- `durationInTraffic`: Object - объект с полями `text` и `value`, описывающий время проезда пути в секундах с учетом пробок;
- `boundedBy`: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

Описание событий

balloonclose

Закрытие балуна.

balloonopen

Открытие балуна.

update

Обновление отображения маршрута. Экземпляр класса [Event](#).

Описание методов

getPaths

```
{GeoObjectCollection} getPaths()
```

Возвращает дочернюю коллекцию путей, из которых состоит маршрут.

multiRouter.driving.RouteModel

Примечание: Конструктор класса multiRouter.driving.RouteModel скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных единичного автомобильного маршрута. Мультимаршрут может состоять из нескольких отдельных маршрутов.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных единичного автомобильного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventManager .
multiRoute	multiRouter.MultiRouteModel	Ссылка на родительскую модель маршрута.
properties	data.Manager	Данные маршрута. Доступны следующие поля: <ul style="list-style-type: none">• index: Integer - порядковый номер маршрута в мультимаршруте;• type: String - идентификатор типа маршрута, принимающий значение "driving" для автомобильных маршрутов;• blocked: Boolean - признак того, что маршрут содержит перекрытые участки;• distance: Object - объект с полями text и value, описывающий протяженность маршрута в метрах;• duration: Object - объект с полями text и value, описывающий время проезда маршрута в секундах;• durationInTraffic: Object - объект с полями text и value, описывающий время проезда пути в секундах с учетом пробок;• boundedBy: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"><code>pathsChange</code>: Boolean - признак изменений в наборе путей.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getPaths()	multiRouter.driving.PathModel[]	Возвращает массив путей маршрута.
getType()	String	Возвращает идентификатор типа маршрутизации. Для автомобильных маршрутов возвращается строка "driving".
update(routeJson)		Обновляет состояние модели.

Описание полей

multiRoute

```
{multiRouter.MultiRouteModel} multiRoute
```

Ссылка на родительскую модель маршрута.

properties

```
{data.Manager} properties
```

Данные маршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер маршрута в мультимаршруте;
- `type`: String - идентификатор типа маршрута, принимающий значение "driving" для автомобильных маршрутов;
- `blocked`: Boolean - признак того, что маршрут содержит перекрытые участки;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность маршрута в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время проезда маршрута в секундах;
- `durationInTraffic`: Object - объект с полями `text` и `value`, описывающий время проезда пути в секундах с учетом пробок;
- `boundedBy`: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pathsChange`: Boolean - признак изменений в наборе путей.

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getPaths

```
{multiRouter.driving.PathModel[]} getPaths()
```

Возвращает массив путей маршрута.

getType

```
{String} getType()
```

Возвращает идентификатор типа маршрутизации. Для автомобильных маршрутов возвращается строка "driving".

update

```
{ } update(routeJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>routeJson</code> *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

multiRouter.driving.Segment

Примечание: Конструктор класса `multiRouter.driving.Segment` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение сегмента автомобильного маршрута. Сегментом автомобильного маршрута называется часть пути от одного манёвра до другого.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение сегмента автомобильного маршрута.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
<code>geometry</code>	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
<code>model</code>	multiRouter.driving.SegmentModel	Модель данных сегмента.

Имя	Тип	Описание
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>
properties	data.Manager	<p>Данные сегмента. Доступны следующие поля:</p> <ul style="list-style-type: none"> index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; type: String - идентификатор типа сегмента, принимающий значение "driving" для автомобильных сегментов; street: String - текстовое описание улицы, по которой проходит сегмент; action: Object - объект с полями text и value, описывающий завершающий маневр сегмента; distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах; durationInTraffic: Object - объект с полями text и value, описывающий время проезда пути в секундах с учетом пробок; text: String - текстовое описание сегмента; viaPoints: Integer[] - индексы транзитных точек, лежащих на данном сегменте; lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.
state	IDataManager	<p>Состояние геообъекта.</p> <p>Унаследовано от IGeoObject.</p>

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
update	Обновление отображения сегмента. Экземпляр класса Event .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

model

```
{multiRouter.driving.SegmentModel} model
```

Модель данных сегмента.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- `index`: `Integer` – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- `type`: `String` - идентификатор типа сегмента, принимающий значение "driving" для автомобильных сегментов;
- `street`: `String` - текстовое описание улицы, по которой проходит сегмент;
- `action`: `Object` - объект с полями `text` и `value`, описывающий завершающий маневр сегмента;
- `distance`: `Object` - объект с полями `"text"` и `"value"`, описывающий протяженность сегмента в метрах;
- `duration`: `Object` - объект с полями `"text"` и `"value"`, описывающий время проезда сегмента в секундах;
- `durationInTraffic`: `Object` - объект с полями `text` и `value`, описывающий время проезда пути в секундах с учетом пробок;
- `text`: `String` - текстовое описание сегмента;
- `viaPoints`: `Integer[]` - индексы транзитных точек, лежащих на данном сегменте;
- `lodIndex`: `Integer` - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление отображения сегмента. Экземпляр класса [Event](#).

multiRouter.driving.SegmentModel

Примечание: Конструктор класса `multiRouter.driving.SegmentModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных сегмента пути автомобильного маршрута. Сегментом автомобильного маршрута называется часть пути от одного манёвра до другого.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных сегмента пути автомобильного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventManager .
geometry	geometry.base.LineString	Геометрия сегмента.
path	multiRouter.driving.PathModel	Ссылка на родительскую модель пути.

Имя	Тип	Описание
properties	data.Manager	<p>Данные сегмента. Доступны следующие поля:</p> <ul style="list-style-type: none"> index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; type: String - идентификатор типа сегмента, принимающий значение "driving" для автомобильных сегментов; street: String - текстовое описание улицы, по которой проходит сегмент; action: Object - объект с полями text и value, описывающий завершающий маневр сегмента; distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах; durationInTraffic: Object - объект с полями text и value, описывающий время проезда пути в секундах с учетом пробок; text: String - текстовое описание сегмента; viaPoints: Integer[] - индексы транзитных точек, лежащих на данном сегменте; lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getType()	String	Возвращает идентификатор типа сегмента. Для сегментов автомобильных маршрутов возвращается строка "driving".
getViaPoints()	multiRouter.ViaPointModel[]	Возвращает массив транзитных точек сегмента.

Имя	Возвращает	Описание
<code>update(segmentJson)</code>		Обновляет состояние модели.

Описание полей

geometry

```
{geometry.base.LineString} geometry
```

Геометрия сегмента.

path

```
{multiRouter.driving.PathModel} path
```

Ссылка на родительскую модель пути.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- `index`: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- `type`: String - идентификатор типа сегмента, принимающий значение "driving" для автомобильных сегментов;
- `street`: String - текстовое описание улицы, по которой проходит сегмент;
- `action`: Object - объект с полями `text` и `value`, описывающий завершающий маневр сегмента;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность сегмента в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время проезда сегмента в секундах;
- `durationInTraffic`: Object - объект с полями `text` и `value`, описывающий время проезда пути в секундах с учетом пробок;
- `text`: String - текстовое описание сегмента;
- `viaPoints`: Integer[] - индексы транзитных точек, лежащих на данном сегменте;
- `lodIndex`: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#).

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getType

```
{String} getType()
```

Возвращает идентификатор типа сегмента. Для сегментов автомобильных маршрутов возвращается строка "driving".

getViaPoints

```
{multiRouter.ViaPointModel[]} getViaPoints()
```

Возвращает массив транзитных точек сегмента.

update

```
{ } update(segmentJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>segmentJson</code> *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

multiRouter.Editor

Расширяет [ICustomizable](#), [IEventEmitter](#).

Редактор мультимаршрута.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
multiRouter.Editor(multiRoute[, state[, options]])
```

Создает редактор мультимаршрута.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>multiRoute</code> *	—	Тип: multiRouter.MultiRoute Редактируемый мультимаршрут.
<code>state</code>	—	Тип: Object Объект, описывающий начальное состояние редактора. Список доступных полей смотрите в описании multiRouter.Editor.state .
<code>options</code>	—	Тип: Object Опции.
<code>options.drawCursor</code>	—	Тип: Object Тип курсора, который будет использоваться в режиме добавления новых путевых точек.

Параметр	Значение по умолчанию	Описание
<code>options.drawOver</code>	true	<p>Тип: Object</p> <p>Разрешает ставить точки поверх объектов карты в режиме добавления новых путевых точек.</p>
<code>options.midPointsType</code>	"way"	<p>Тип: String</p> <p>Определяет тип добавляемых точек при перетаскивании маркера, появляющегося при наведении курсора мыши на активный маршрут. Может принимать одно из следующих строковых значений:</p> <ul style="list-style-type: none"> "way" – добавлять путевые точки; "via" – добавлять транзитные точки. <p>Также смотрите описание поля <code>addMidPoints</code> менеджера состояния <code>multiRouter.Editor.state</code>.</p>

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	<code>IEventManager</code>	<p>Менеджер событий.</p> <p>Унаследовано от <code>IEventEmitter</code>.</p>
<code>options</code>	<code>IOptionManager</code>	<p>Менеджер опций.</p> <p>Унаследовано от <code>ICustomizable</code>.</p>

Имя	Тип	Описание
<code>state</code>	<code>data.Manager</code>	<p>Менеджер состояния редактора маршрута.</p> <p>Доступные поля:</p> <ul style="list-style-type: none">• <code>addWayPoints</code>: Boolean - разрешает добавление новых путевых точек при клике на карту. Значение по умолчанию: <code>false</code>.• <code>dragWayPoints</code>: Boolean - разрешает перетаскивание уже существующих путевых точек. Значение по умолчанию: <code>true</code>.• <code>removeWayPoints</code>: Boolean – разрешает удаление путевых точек по двойному клику по ним. Значение по умолчанию: <code>false</code>.• <code>dragViaPoints</code>: Boolean - разрешает перетаскивание уже существующих транзитных точек. Значение по умолчанию: <code>true</code>.• <code>removeViaPoints</code>: Boolean – разрешает удаление транзитных точек по двойному клику по ним. Значение по умолчанию: <code>true</code>.• <code>addMidPoints</code>: Boolean - разрешает добавление промежуточных транзитных или путевых точек посредством перетаскивания маркера, появляющегося при наведении курсора мыши на активный маршрут. Тип добавляемых точек задается опцией <code>midPointsType</code>. Значение по умолчанию: <code>true</code>.

События

Имя	Описание
beforemidpointadd	<p>Событие, предшествующее событию <code>midpointadd</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>coords</code>: <code>Number[]</code> - координаты добавления промежуточной точки; <code>pointType</code>: <code>String</code> – идентификатор типа добавляемой промежуточной точки; <code>insertIndex</code>: <code>Integer</code> – индекс вставки промежуточной точки во множестве опорных точек мультимаршрута. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPointType</code> - метод, позволяющий задать тип добавляемой точки. В качестве аргумента принимает строковый идентификатор типа (смотрите описание опции <code>midPointsType</code>). <code>setInsertIndex</code> - метод, позволяющий скорректировать индекс вставки промежуточной точки, который будет фактически применен. В качестве аргумента принимает новый индекс. <p>Если у данного события вызвать метод Event.preventDefault, то последующее добавление промежуточной точки маршрута, как и событие <code>"midpointadd"</code>, будет отменено.</p>
beforemidpointdrag	<p>Событие, предшествующее событию <code>midpointdrag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code>: <code>Number[]</code> - пиксельный сдвиг по осям на данном шаге; <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие <code>"midpointdrag"</code> будет отменено.</p>

Имя	Описание
beforemidpointpinshow	<p>Событие, предшествующее событию <code>midpointpinshow</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pin</code>: Placemark - ссылка на объект маркера; <code>globalPixels</code>: <code>Number[]</code> - глобальные пиксельные координаты маркера; <code>segment</code>: multiRouter.driving.Segment – ссылка на сегмент маршрута, над которым появился маркер. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "midpointpinshow" будет отменено, а маркер будет скрыт.</p>
beforeviapointdrag	<p>Событие, предшествующее событию <code>viapointdrag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code>: <code>Number[]</code> - пиксельный сдвиг по осям на данном шаге; <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "viapointdrag" будет отменено.</p>
beforeviapointdragstart	<p>Событие, предшествующее событию <code>viapointdragstart</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>domEvent</code>: DomEvent – исходное DOM-событие, если оно имеется; <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "viapointdragstart", будет отменено.</p>
beforeviapointremove	<p>Событие, предшествующее событию <code>viapointremove</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>viaPoint</code>: multiRouter.ViaPoint - ссылка на объект удаляемой транзитной точки. <p>Если у данного события вызвать метод Event.preventDefault, то удаление транзитной точки, как и последующее событие "viapointremove", будет отменено.</p>

Имя	Описание
beforewaypointadd	<p>Событие, предшествующее событию <code>waypointadd</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>coords: Number[]</code> - координаты добавленной путевой точки. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setCoords</code> - метод, позволяющий скорректировать координаты добавляемой путевой точки. В качестве аргумента принимает новые координаты в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то добавление путевой точки, как и последующее событие <code>"waypointadd"</code>, будет отменено.</p>
beforewaypointdrag	<p>Событие, предшествующее событию <code>waypointdrag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset: Number[]</code> - пиксельный сдвиг по осям на данном шаге; <code>wayPoint: multiRouter.WayPoint</code> – ссылка на объект перетаскиваемой путевой точки. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие <code>"waypointdrag"</code> будет отменено.</p>
beforewaypointdragstart	<p>Событие, предшествующее событию <code>waypointdragstart</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>domEvent: DomEvent</code> – исходное DOM-событие, если оно имеется; <code>wayPoint: multiRouter.WayPoint</code> – ссылка на объект перетаскиваемой путевой точки. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие <code>"waypointdragstart"</code>, будет отменено.</p>
beforewaypointremove	<p>Событие, предшествующее событию <code>waypointremove</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>wayPoint: multiRouter.WayPoint</code> - ссылка на объект удаляемой путевой точки. <p>Если у данного события вызвать метод Event.preventDefault, то удаление путевой точки, как и последующее событие <code>"waypointremove"</code>, будет отменено.</p>

Имя	Описание
midpointadd	<p>Добавление промежуточной точки маршрута. Тип точки определяется значением опции <code>midPointsType</code>. Экземпляр класса <code>Event</code>. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>coords</code>: <code>Number[]</code> - координаты добавления промежуточной точки; <code>pointType</code>: <code>String</code> – идентификатор типа добавляемой промежуточной точки; <code>insertIndex</code>: <code>Integer</code> – индекс вставки промежуточной точки во множестве опорных точек мультимаршрута.
midpointdrag	<p>Перетаскивание добавленной промежуточной точки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code>: <code>Number[]</code> - пиксельный сдвиг по осям на данном шаге;
midpointdragend	<p>Завершение перетаскивания добавленной промежуточной точки. Экземпляр класса Event.</p>
midpointpinshow	<p>Появление перетаскиваемого маркера при наведении на активный маршрут. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pin</code>: Placemark - ссылка на объект маркера; <code>globalPixels</code>: <code>Number[]</code> - глобальные пиксельные координаты маркера; <code>segment</code>: multiRouter.driving.Segment – ссылка на сегмент маршрута, над которым появился маркер.
optionschange	<p>Изменение в опциях объекта. Унаследовано от ICustomizable.</p>
viapointdrag	<p>Перетаскивание транзитной точки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code>: <code>Number[]</code> - пиксельный сдвиг по осям на данном шаге; <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки.
viapointdragend	<p>Завершение перетаскивания транзитной точки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки.
viapointdragstart	<p>Начало перетаскивания транзитной точки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>domEvent</code>: DomEvent – исходное DOM-событие, если оно имеется; <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки.

Имя	Описание
viapointremove	Удаление транзитной точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> viaPoint: multiRouter.ViaPoint - ссылка на объект удаленной транзитной точки.
waypointadd	Добавление путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> coords: Number[] - координаты добавленной путевой точки.
waypointdrag	Перетаскивание путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> pixelOffset: Number[] - пиксельный сдвиг по осям на данном шаге; wayPoint: multiRouter.WayPoint – ссылка на объект перетаскиваемой путевой точки.
waypointdragend	Завершение перетаскивания путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> wayPoint: multiRouter.WayPoint – ссылка на объект перетаскиваемой путевой точки.
waypointdragstart	Начало перетаскивания путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> domEvent: DomEvent – исходное DOM-событие, если оно имеется; wayPoint: multiRouter.WayPoint – ссылка на объект перетаскиваемой путевой точки.
waypointremove	Удаление путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> wayPoint: multiRouter.WayPoint - ссылка на объект удаленной путевой точки.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает редактор мультимаршрута.
getMultiRoute()	multiRouter.MultiRoute	Возвращает ссылку на редактируемый мультимаршрут.

Описание полей

state

```
{data.Manager} state
```

Менеджер состояния редактора маршрута.

Доступные поля:

- `addWayPoints`: Boolean - разрешает добавление новых путевых точек при клике на карту. Значение по умолчанию: `false`.
- `dragWayPoints`: Boolean - разрешает перетаскивание уже существующих путевых точек. Значение по умолчанию `true`.
- `removeWayPoints`: Boolean – разрешает удаление путевых точек по двойному клику по ним. Значение по умолчанию: `false`.
- `dragViaPoints`: Boolean - разрешает перетаскивание уже существующих транзитных точек. Значение по умолчанию `true`.
- `removeViaPoints`: Boolean – разрешает удаление транзитных точек по двойному клику по ним. Значение по умолчанию `true`.
- `addMidPoints`: Boolean - разрешает добавление промежуточных транзитных или путевых точек посредством перетаскивания маркера, появляющегося при наведении курсора мыши на активный маршрут. Тип добавляемых точек задается опцией `midPointsType`. Значение по умолчанию `true`.

Описание событий

`beforemidpointadd`

Событие, предшествующее событию `midpointadd`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `coords`: Number[] - координаты добавления промежуточной точки;
- `pointType`: String – идентификатор типа добавляемой промежуточной точки;
- `insertIndex`: Integer – индекс вставки промежуточной точки во множестве опорных точек мультимаршрута.

Имена методов, доступных через метод [Event.callMethod](#):

- `setPointType` - метод, позволяющий задать тип добавляемой точки. В качестве аргумента принимает строковый идентификатор типа (смотрите описание опции `midPointsType`).
- `setInsertIndex` - метод, позволяющий скорректировать индекс вставки промежуточной точки, который будет фактически применен. В качестве аргумента принимает новый индекс.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее добавление промежуточной точки маршрута, как и событие "midpointadd", будет отменено.

`beforemidpointdrag`

Событие, предшествующее событию `midpointdrag`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset`: Number[] - пиксельный сдвиг по осям на данном шаге;

Имена методов, доступных через метод [Event.callMethod](#):

- `setPixelOffset` - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "midpointdrag" будет отменено.

`beforemidpointpinshow`

Событие, предшествующее событию `midpointpinshow`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pin`: [Placemark](#) - ссылка на объект маркера;
- `globalPixels`: Number[] - глобальные пиксельные координаты маркера;
- `segment`: [multiRouter.driving.Segment](#) – ссылка на сегмент маршрута, над которым появился маркер.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "midpointpinshow" будет отменено, а маркер будет скрыт.

beforeviapointdrag

Событие, предшествующее событию `viapointdrag`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset`: `Number[]` - пиксельный сдвиг по осям на данном шаге;
- `viaPoint`: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

Имена методов, доступных через метод [Event.callMethod](#):

- `setPixelOffset` - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "viapointdrag" будет отменено.

beforeviapointdragstart

Событие, предшествующее событию `viapointdragstart`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `domEvent`: [DomEvent](#) – исходное DOM-событие, если оно имеется;
- `viaPoint`: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие "viapointdragstart", будет отменено.

beforeviapointremove

Событие, предшествующее событию `viapointremove`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `viaPoint`: [multiRouter.ViaPoint](#) - ссылка на объект удаляемой транзитной точки.

Если у данного события вызвать метод [Event.preventDefault](#), то удаление транзитной точки, как и последующее событие "viapointremove", будет отменено.

beforewaypointadd

Событие, предшествующее событию `waypointadd`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `coords`: `Number[]` - координаты добавленной путевой точки.

Имена методов, доступных через метод [Event.callMethod](#):

- `setCoords` - метод, позволяющий скорректировать координаты добавляемой путевой точки. В качестве аргумента принимает новые координаты в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то добавление путевой точки, как и последующее событие "waypointadd", будет отменено.

beforewaypointdrag

Событие, предшествующее событию `waypointdrag`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset`: `Number[]` - пиксельный сдвиг по осям на данном шаге;
- `wayPoint`: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

Имена методов, доступных через метод [Event.callMethod](#):

- `setPixelOffset` - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "waypointdrag" будет отменено.

beforewaypointdragstart

Событие, предшествующее событию waypointdragstart. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent: [DomEvent](#) – исходное DOM-событие, если оно имеется;
- wayPoint: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие "waypointdragstart", будет отменено.

beforewaypointremove

Событие, предшествующее событию waypointremove. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- wayPoint: [multiRouter.WayPoint](#) - ссылка на объект удаляемой путевой точки.

Если у данного события вызвать метод [Event.preventDefault](#), то удаление путевой точки, как и последующее событие "waypointremove", будет отменено.

midpointadd

Добавление промежуточной точки маршрута. Тип точки определяется значением опции midPointsType. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- coords: Number[] - координаты добавления промежуточной точки;
- pointType: String – идентификатор типа добавляемой промежуточной точки;
- insertIndex: Integer – индекс вставки промежуточной точки во множестве опорных точек мультимаршрута.

midpointdrag

Перетаскивание добавленной промежуточной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset: Number[] - пиксельный сдвиг по осям на данном шаге;

midpointdragend

Завершение перетаскивания добавленной промежуточной точки. Экземпляр класса [Event](#).

midpointpinshow

Появление перетаскиваемого маркера при наведении на активный маршрут. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pin: [Placemark](#) - ссылка на объект маркера;
- globalPixels: Number[] - глобальные пиксельные координаты маркера;
- segment: [multiRouter.driving.Segment](#) – ссылка на сегмент маршрута, над которым появился маркер.

viapointdrag

Перетаскивание транзитной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset: Number[] - пиксельный сдвиг по осям на данном шаге;
- viaPoint: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

viapointdragend

Завершение перетаскивания транзитной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- viaPoint: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

viapointdragstart

Начало перетаскивания транзитной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent: [DomEvent](#) – исходное DOM-событие, если оно имеется;
- viaPoint: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

viapointremove

Удаление транзитной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- viaPoint: [multiRouter.ViaPoint](#) - ссылка на объект удаленной транзитной точки.

waypointadd

Добавление путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- coords: [Number\[\]](#) - координаты добавленной путевой точки.

waypointdrag

Перетаскивание путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset: [Number\[\]](#) - пиксельный сдвиг по осям на данном шаге;
- wayPoint: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

waypointdragend

Завершение перетаскивания путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- wayPoint: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

waypointdragstart

Начало перетаскивания путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent: [DomEvent](#) – исходное DOM-событие, если оно имеется;
- wayPoint: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

waypointremove

Удаление путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- wayPoint: [multiRouter.WayPoint](#) - ссылка на объект удаленной путевой точки.

Описание методов**destroy**

```
{ } destroy()
```

Уничтожает редактор мультимаршрута.

getMultiRoute

```
{multiRouter.MultiRoute} getMultiRoute()
```

Возвращает ссылку на редактируемый мультимаршрут.

multiRouter.EditorAddon

Примечание: Конструктор класса multiRouter.EditorAddon скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [ICustomizable](#), [IEventEmitter](#).

Аддон редактора мультимаршрута.

[Поля](#) | [События](#) | [Методы](#)

Создает аддон редактора мультимаршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
state	data.Manager	Менеджер состояния редактора маршрута.

События

Имя	Описание
beforemidpointadd	<p>Событие, предшествующее событию <code>midpointadd</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>coords</code>: <code>Number[]</code> - координаты добавления промежуточной точки; <code>pointType</code>: <code>String</code> – идентификатор типа добавляемой промежуточной точки; <code>insertIndex</code>: <code>Integer</code> – индекс вставки промежуточной точки во множестве опорных точек мультимаршрута. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPointType</code> - метод, позволяющий задать тип добавляемой точки. В качестве аргумента принимает строковый идентификатор типа (смотрите описание опции <code>midPointsType</code>). <code>setInsertIndex</code> - метод, позволяющий скорректировать индекс вставки промежуточной точки, который будет фактически применен. В качестве аргумента принимает новый индекс. <p>Если у данного события вызвать метод Event.preventDefault, то последующее добавление промежуточной точки маршрута, как и событие <code>"midpointadd"</code>, будет отменено.</p>
beforemidpointdrag	<p>Событие, предшествующее событию <code>midpointdrag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code>: <code>Number[]</code> - пиксельный сдвиг по осям на данном шаге; <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие <code>"midpointdrag"</code> будет отменено.</p>

Имя	Описание
beforemidpointpinshow	<p>Событие, предшествующее событию <code>midpointpinshow</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pin</code>: Placemark - ссылка на объект маркера; <code>globalPixels</code>: <code>Number[]</code> - глобальные пиксельные координаты маркера; <code>segment</code>: multiRouter.driving.Segment – ссылка на сегмент маршрута, над которым появился маркер. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "midpointpinshow" будет отменено, а маркер будет скрыт.</p>
beforeviapointdrag	<p>Событие, предшествующее событию <code>viapointdrag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code>: <code>Number[]</code> - пиксельный сдвиг по осям на данном шаге; <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "viapointdrag" будет отменено.</p>
beforeviapointdragstart	<p>Событие, предшествующее событию <code>viapointdragstart</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>domEvent</code>: DomEvent – исходное DOM-событие, если оно имеется; <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "viapointdragstart", будет отменено.</p>
beforeviapointremove	<p>Событие, предшествующее событию <code>viapointremove</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>viaPoint</code>: multiRouter.ViaPoint - ссылка на объект удаляемой транзитной точки. <p>Если у данного события вызвать метод Event.preventDefault, то удаление транзитной точки, как и последующее событие "viapointremove", будет отменено.</p>

Имя	Описание
beforewaypointadd	<p>Событие, предшествующее событию <code>waypointadd</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>coords: Number[]</code> - координаты добавленной путевой точки. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setCoords</code> - метод, позволяющий скорректировать координаты добавляемой путевой точки. В качестве аргумента принимает новые координаты в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то добавление путевой точки, как и последующее событие <code>"waypointadd"</code>, будет отменено.</p>
beforewaypointdrag	<p>Событие, предшествующее событию <code>waypointdrag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset: Number[]</code> - пиксельный сдвиг по осям на данном шаге; <code>wayPoint: multiRouter.WayPoint</code> – ссылка на объект перетаскиваемой путевой точки. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие <code>"waypointdrag"</code> будет отменено.</p>
beforewaypointdragstart	<p>Событие, предшествующее событию <code>waypointdragstart</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>domEvent: DomEvent</code> – исходное DOM-событие, если оно имеется; <code>wayPoint: multiRouter.WayPoint</code> – ссылка на объект перетаскиваемой путевой точки. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие <code>"waypointdragstart"</code>, будет отменено.</p>
beforewaypointremove	<p>Событие, предшествующее событию <code>waypointremove</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>wayPoint: multiRouter.WayPoint</code> - ссылка на объект удаляемой путевой точки. <p>Если у данного события вызвать метод Event.preventDefault, то удаление путевой точки, как и последующее событие <code>"waypointremove"</code>, будет отменено.</p>

Имя	Описание
midpointadd	<p>Добавление промежуточной точки маршрута. Тип точки определяется значением опции <code>midPointsType</code>. Экземпляр класса <code>Event</code>. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>coords</code>: <code>Number[]</code> - координаты добавления промежуточной точки; <code>pointType</code>: <code>String</code> – идентификатор типа добавляемой промежуточной точки; <code>insertIndex</code>: <code>Integer</code> – индекс вставки промежуточной точки во множестве опорных точек мультимаршрута.
midpointdrag	<p>Перетаскивание добавленной промежуточной точки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code>: <code>Number[]</code> - пиксельный сдвиг по осям на данном шаге;
midpointdragend	<p>Завершение перетаскивания добавленной промежуточной точки. Экземпляр класса Event.</p>
midpointpinshow	<p>Появление перетаскиваемого маркера при наведении на активный маршрут. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pin</code>: Placemark - ссылка на объект маркера; <code>globalPixels</code>: <code>Number[]</code> - глобальные пиксельные координаты маркера; <code>segment</code>: multiRouter.driving.Segment – ссылка на сегмент маршрута, над которым появился маркер.
optionschange	<p>Изменение в опциях объекта. Унаследовано от ICustomizable.</p>
start	<p>Включение редактора. Экземпляр класса Event.</p>
stop	<p>Выключение редактора. Экземпляр класса Event.</p>
viapointdrag	<p>Перетаскивание транзитной точки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>pixelOffset</code>: <code>Number[]</code> - пиксельный сдвиг по осям на данном шаге; <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки.
viapointdragend	<p>Завершение перетаскивания транзитной точки. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>viaPoint</code>: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки.

Имя	Описание
viapointdragstart	Начало перетаскивания транзитной точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> domEvent: DomEvent – исходное DOM-событие, если оно имеется; viaPoint: multiRouter.ViaPoint – ссылка на объект перетаскиваемой транзитной точки.
viapointremove	Удаление транзитной точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> viaPoint: multiRouter.ViaPoint – ссылка на объект удаленной транзитной точки.
waypointadd	Добавление путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> coords: Number[] – координаты добавленной путевой точки.
waypointdrag	Перетаскивание путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> pixelOffset: Number[] – пиксельный сдвиг по осям на данном шаге; wayPoint: multiRouter.WayPoint – ссылка на объект перетаскиваемой путевой точки.
waypointdragend	Завершение перетаскивания путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> wayPoint: multiRouter.WayPoint – ссылка на объект перетаскиваемой путевой точки.
waypointdragstart	Начало перетаскивания путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> domEvent: DomEvent – исходное DOM-событие, если оно имеется; wayPoint: multiRouter.WayPoint – ссылка на объект перетаскиваемой путевой точки.
waypointremove	Удаление путевой точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> wayPoint: multiRouter.WayPoint – ссылка на объект удаленной путевой точки.

Методы

Имя	Возвращает	Описание
isActive()	Boolean	Возвращает признак того, что редактор в данный момент включен.
start(state)		Включает редактор.
stop()		Отключает редактор.

Описание полей

state

```
{data.Manager} state
```

Менеджер состояния редактора маршрута.

Описание событий

beforemidpointadd

Событие, предшествующее событию `midpointadd`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `coords`: `Number[]` - координаты добавления промежуточной точки;
- `pointType`: `String` – идентификатор типа добавляемой промежуточной точки;
- `insertIndex`: `Integer` – индекс вставки промежуточной точки во множестве опорных точек мультимаршрута.

Имена методов, доступных через метод [Event.callMethod](#):

- `setPointType` - метод, позволяющий задать тип добавляемой точки. В качестве аргумента принимает строковый идентификатор типа (смотрите описание опции `midPointsType`).
- `setInsertIndex` - метод, позволяющий скорректировать индекс вставки промежуточной точки, который будет фактически применен. В качестве аргумента принимает новый индекс.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее добавление промежуточной точки маршрута, как и событие `"midpointadd"`, будет отменено.

beforemidpointdrag

Событие, предшествующее событию `midpointdrag`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset`: `Number[]` - пиксельный сдвиг по осям на данном шаге;

Имена методов, доступных через метод [Event.callMethod](#):

- `setPixelOffset` - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие `"midpointdrag"` будет отменено.

beforemidpointpinshow

Событие, предшествующее событию `midpointpinshow`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pin`: [Placemark](#) - ссылка на объект маркера;
- `globalPixels`: `Number[]` - глобальные пиксельные координаты маркера;
- `segment`: [multiRouter.driving.Segment](#) – ссылка на сегмент маршрута, над которым появился маркер.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие `"midpointpinshow"` будет отменено, а маркер будет скрыт.

beforeviapointdrag

Событие, предшествующее событию `viapointdrag`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset`: `Number[]` - пиксельный сдвиг по осям на данном шаге;

- `viaPoint`: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

Имена методов, доступных через метод [Event.callMethod](#):

- `setPixelOffset` - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "viapointdrag" будет отменено.

beforeviapointdragstart

Событие, предшествующее событию `viapointdragstart`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `domEvent`: [DomEvent](#) – исходное DOM-событие, если оно имеется;
- `viaPoint`: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие "viapointdragstart", будет отменено.

beforeviapointremove

Событие, предшествующее событию `viapointremove`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `viaPoint`: [multiRouter.ViaPoint](#) - ссылка на объект удаляемой транзитной точки.

Если у данного события вызвать метод [Event.preventDefault](#), то удаление транзитной точки, как и последующее событие "viapointremove", будет отменено.

beforewaypointadd

Событие, предшествующее событию `waypointadd`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `coords`: `Number[]` - координаты добавленной путевой точки.

Имена методов, доступных через метод [Event.callMethod](#):

- `setCoords` - метод, позволяющий скорректировать координаты добавляемой путевой точки. В качестве аргумента принимает новые координаты в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то добавление путевой точки, как и последующее событие "waypointadd", будет отменено.

beforewaypointdrag

Событие, предшествующее событию `waypointdrag`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset`: `Number[]` - пиксельный сдвиг по осям на данном шаге;
- `wayPoint`: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

Имена методов, доступных через метод [Event.callMethod](#):

- `setPixelOffset` - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее событие "waypointdrag" будет отменено.

beforewaypointdragstart

Событие, предшествующее событию `waypointdragstart`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `domEvent`: [DomEvent](#) – исходное DOM-событие, если оно имеется;
- `wayPoint`: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

Если у данного события вызвать метод [Event.preventDefault](#), то последующее перетаскивание, как и событие `"waypointdragstart"`, будет отменено.

beforewaypointremove

Событие, предшествующее событию `waypointremove`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `wayPoint`: [multiRouter.WayPoint](#) – ссылка на объект удаляемой путевой точки.

Если у данного события вызвать метод [Event.preventDefault](#), то удаление путевой точки, как и последующее событие `"waypointremove"`, будет отменено.

midpointadd

Добавление промежуточной точки маршрута. Тип точки определяется значением опции `midPointsType`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `coords`: [Number\[\]](#) – координаты добавления промежуточной точки;
- `pointType`: [String](#) – идентификатор типа добавляемой промежуточной точки;
- `insertIndex`: [Integer](#) – индекс вставки промежуточной точки во множестве опорных точек мультимаршрута.

midpointdrag

Перетаскивание добавленной промежуточной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelOffset`: [Number\[\]](#) – пиксельный сдвиг по осям на данном шаге;

midpointdragend

Завершение перетаскивания добавленной промежуточной точки. Экземпляр класса [Event](#).

midpointpinshow

Появление перетаскиваемого маркера при наведении на активный маршрут. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pin`: [Placemark](#) – ссылка на объект маркера;
- `globalPixels`: [Number\[\]](#) – глобальные пиксельные координаты маркера;
- `segment`: [multiRouter.driving.Segment](#) – ссылка на сегмент маршрута, над которым появился маркер.

start

Включение редактора. Экземпляр класса [Event](#).

stop

Выключение редактора. Экземпляр класса [Event](#).

viapointdrag

Перетаскивание транзитной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset: Number[] - пиксельный сдвиг по осям на данном шаге;
- viaPoint: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

viapointdragend

Завершение перетаскивания транзитной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- viaPoint: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

viapointdragstart

Начало перетаскивания транзитной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent: [DomEvent](#) – исходное DOM-событие, если оно имеется;
- viaPoint: [multiRouter.ViaPoint](#) – ссылка на объект перетаскиваемой транзитной точки.

viapointremove

Удаление транзитной точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- viaPoint: [multiRouter.ViaPoint](#) - ссылка на объект удаленной транзитной точки.

waypointadd

Добавление путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- coords: Number[] - координаты добавленной путевой точки.

waypointdrag

Перетаскивание путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- pixelOffset: Number[] - пиксельный сдвиг по осям на данном шаге;
- wayPoint: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

waypointdragend

Завершение перетаскивания путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- wayPoint: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

waypointdragstart

Начало перетаскивания путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- domEvent: [DomEvent](#) – исходное DOM-событие, если оно имеется;
- wayPoint: [multiRouter.WayPoint](#) – ссылка на объект перетаскиваемой путевой точки.

waypointremove

Удаление путевой точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- wayPoint: [multiRouter.WayPoint](#) - ссылка на объект удаленной путевой точки.

Описание методов

isActive

```
{Boolean} isActive()
```

Возвращает признак того, что редактор в данный момент включен.

start

```
{ } start(state)
```

Включает редактор.

Параметры:

Параметр	Значение по умолчанию	Описание
state *	—	Тип: Object Начальное состояние редактора.

* Обязательный параметр/опция.

stop

```
{ } stop()
```

Отключает редактор.

multiRouter.masstransit

multiRouter.masstransit.Path

Примечание: Конструктор класса `multiRouter.masstransit.Path` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение пути маршрута общественного транспорта. Единичный маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение пути маршрута общественного транспорта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.masstransit.PathModel	Модель данных пути мультимаршрута.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

Имя	Тип	Описание
properties	data.Manager	<p>Модель данных пути мультимаршрута. Доступны следующие поля:</p> <ul style="list-style-type: none"> <code>index</code>: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута; <code>type</code>: String – идентификатор типа маршрута. Принимает значение "masstransit" для маршрутов общественного транспорта; <code>distance</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий протяженность пути в метрах; <code>duration</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий время движения по пути в секундах; <code>coordinates</code>: Number[][] - координаты всех точек пути; <code>encodedCoordinates</code>: String - строка закодированных в base64 координат всех точек пути.
state	IDataManager	<p>Состояние геообъекта.</p> <p>Унаследовано от IGeoObject.</p>

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>

Имя	Описание
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> clientX - координата касания X относительно видимой области браузера; clientY - координата касания Y относительно видимой области браузера; pageX - координата касания X касания относительно начала документа; pageY - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление отображения пути. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getSegmentMarkers()	GeoObjectCollection	Возвращает дочернюю коллекцию маркеров сегментов.
getSegments()	GeoObjectCollection	Возвращает дочернюю коллекцию сегментов, из которых состоит путь.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

model

```
{multiRouter.masstransit.PathModel} model
```

Модель данных пути мультимаршрута.

properties

```
{data.Manager} properties
```


Модель данных пути мультимаршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута;
- `type`: String – идентификатор типа маршрута. Принимает значение "masstransit" для маршрутов общественного транспорта;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность пути в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время движения по пути в секундах;
- `coordinates`: Number[][] - координаты всех точек пути;
- `encodedCoordinates`: String - строка закодированных в base64 координат всех точек пути.

Описание событий

update

Обновление отображения пути. Экземпляр класса [Event](#).

Описание методов

getSegmentMarkers

```
{GeoObjectCollection} getSegmentMarkers()
```

Возвращает дочернюю коллекцию маркеров сегментов.

getSegments

```
{GeoObjectCollection} getSegments()
```

Возвращает дочернюю коллекцию сегментов, из которых состоит путь.

multiRouter.masstransit.PathModel

Примечание: Конструктор класса `multiRouter.masstransit.PathModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventEmitter](#).

Модель данных пути маршрута общественного транспорта. Единичный маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных пути маршрута общественного транспорта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

Имя	Тип	Описание
properties	data.Manager	<p>Модель данных пути мультимаршрута. Доступны следующие поля:</p> <ul style="list-style-type: none"> <code>index</code>: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута; <code>type</code>: String – идентификатор типа маршрута. Принимает значение "masstransit" для маршрутов общественного транспорта; <code>distance</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий протяженность пути в метрах; <code>duration</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий время движения по пути в секундах; <code>coordinates</code>: Number[][] - координаты всех точек пути; <code>encodedCoordinates</code>: String - строка закодированных в base64 координат всех точек пути.
route	multiRouter.masstransit.RouteModel	Ссылка на родительскую модель маршрута.

События

Имя	Описание
update	<p>Обновление модели новыми данными. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>segmentsChange</code>: Boolean – признак изменений в наборе сегментов.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getSegments()	(multiRouter.masstransit.TransferSegmentModel multiRouter.masstransit.TransportSegmentModel multiRouter.masstransit.WalkSegmentModel) []	Возвращает массив сегментов маршрута.
getType()	String	Возвращает идентификатор типа маршрутизации. Для маршрутов общественного транспорта возвращается строка "masstransit".
update(path.Json)		Обновляет состояние модели.

Описание полей

properties

```
{data.Manager} properties
```

Модель данных пути мультимаршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута;
- `type`: String – идентификатор типа маршрута. Принимает значение "masstransit" для маршрутов общественного транспорта;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность пути в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время движения по пути в секундах;
- `coordinates`: Number[][] - координаты всех точек пути;
- `encodedCoordinates`: String - строка закодированных в base64 координат всех точек пути.

route

```
{multiRouter.masstransit.RouteModel} route
```

Ссылка на родительскую модель маршрута.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `segmentsChange`: Boolean – признак изменений в наборе сегментов.

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getSegments

```
{(multiRouter.masstransit.TransferSegmentModel|  
multiRouter.masstransit.TransportSegmentModel|  
multiRouter.masstransit.WalkSegmentModel)[ ]} getSegments()
```

Возвращает массив сегментов пути.

getType

```
{String} getType()
```

Возвращает идентификатор типа маршрутизации. Для маршрутов общественного транспорта возвращается строка "masstransit".

update

```
{ } update(pathJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>pathJson</code> *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

multiRouter.masstransit.Route

Примечание: Конструктор класса `multiRouter.masstransit.Route` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение единичного маршрута общественного транспорта. Мультимаршрут может состоять из нескольких отдельных маршрутов.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение единичного маршрута общественного транспорта.

Поля

Имя	Тип	Описание
balloon	IMultiRouterRouteBalloon	Балун маршрута.
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.masstransit.RouteModel	Модель данных единичного маршрута.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	data.Manager	Данные маршрута. Доступны следующие поля: <ul style="list-style-type: none"> <code>index</code>: Integer - порядковый номер маршрута в мультимаршруте; <code>type</code>: String – идентификатор типа маршрута. Принимает значение "masstransit" для маршрутов общественного транспорта; <code>distance</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий протяженность маршрута в метрах; <code>duration</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий время проезда маршрута в секундах; <code>boundedBy</code>: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
balloonclose	Закрытие балуна.
balloonopen	Открытие балуна.

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>overlay</code>: IOverlay null - ссылка на оверлей; • <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление отображения маршрута. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.</p> <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	<p>Метод предоставляет синхронный доступ к оверлею.</p> <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	<p>Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.</p> <p>Унаследован от IChildOnMap.</p>

Имя	Возвращает	Описание
<code>getPaths()</code>	GeoObjectCollection	Возвращает дочернюю коллекцию путей, из которых состоит маршрут.
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей**balloon**

```
{IMultiRouterRouteBalloon} balloon
```

Балун маршрута.

model

```
{multiRouter.masstransit.RouteModel} model
```

Модель данных единичного маршрута.

properties

```
{data.Manager} properties
```

Данные маршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер маршрута в мультимаршруте;
- `type`: String – идентификатор типа маршрута. Принимает значение "masstransit" для маршрутов общественного транспорта;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность маршрута в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время проезда маршрута в секундах;
- `boundedBy`: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

Описание событий**balloonclose**

Заккрытие балуна.

balloonopen

Открытие балуна.

update

Обновление отображения маршрута. Экземпляр класса [Event](#).

Описание методов

getPaths

```
{GeoObjectCollection} getPaths()
```

Возвращает дочернюю коллекцию путей, из которых состоит маршрут.

multiRouter.masstransit.RouteModel

Примечание: Конструктор класса multiRouter.masstransit.RouteModel скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных единичного маршрута общественного транспорта. Мультимаршрут может состоять из нескольких отдельных маршрутов.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных единичного маршрута общественного транспорта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
multiRoute	multiRouter.MultiRouteModel	Ссылка на родительскую модель маршрута.
properties	data.Manager	Данные маршрута. Доступны следующие поля: <ul style="list-style-type: none"> index: Integer - порядковый номер маршрута в мультимаршруте; type: String – идентификатор типа маршрута. Принимает значение "masstransit" для маршрутов общественного транспорта; distance: Object - объект с полями text и value, описывающий протяженность маршрута в метрах; duration: Object - объект с полями text и value, описывающий время проезда маршрута в секундах; boundedBy: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> pathsChange: Boolean - признак изменений в наборе путей.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.

Имя	Возвращает	Описание
<code>getPaths()</code>	<code>multiRouter.masstransit.PathModel[]</code>	Возвращает массив путей маршрута.
<code>getType()</code>	String	Возвращает идентификатор типа маршрутизации. Для маршрутов общественного транспорта возвращается строка "masstransit".
<code>update(routeJson)</code>		Обновляет состояние модели.

Описание полей

multiRoute

```
{multiRouter.MultiRouteModel} multiRoute
```

Ссылка на родительскую модель маршрута.

properties

```
{data.Manager} properties
```

Данные маршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер маршрута в мультимаршруте;
- `type`: String – идентификатор типа маршрута. Принимает значение "masstransit" для маршрутов общественного транспорта;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность маршрута в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время проезда маршрута в секундах;
- `boundedBy`: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pathsChange`: Boolean - признак изменений в наборе путей.

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getPaths

```
{multiRouter.masstransit.PathModel[]} getPaths()
```

Возвращает массив путей маршрута.

getType

```
{String} getType()
```

Возвращает идентификатор типа маршрутизации. Для маршрутов общественного транспорта возвращается строка "masstransit".

update

```
{ } update(routeJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>routeJson</code> *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

multiRouter.masstransit.StopModel

Примечание: Конструктор класса `multiRouter.masstransit.StopModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventEmitter](#).

Модель данных остановки транспортного сегмента маршрута общественного транспорта.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных остановки транспортного сегмента.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
<code>geometry</code>	geometry.base.Point	Геометрия остановки.
<code>properties</code>	data.Manager	Данные остановки. Доступны следующие поля: <ul style="list-style-type: none"> <code>Integer</code> – порядковый номер остановки во множестве остановок соответствующего транспортного сегмента; <code>id: String</code> – идентификатор остановки; <code>name: String</code> – название остановки; <code>lodIndex: Integer</code> – порядковый номер точки остановки во множестве всех координат соответствующего ей пути маршрута.
<code>segment</code>	multiRouter.masstransit.TransportSegmentModel	Ссылка на родительскую модель транспортного сегмента.

События

Имя	Описание
<code>update</code>	Обновление модели новыми данными. Экземпляр класса Event .

Методы

Имя	Описание
<code>update(stopJson)</code>	Обновляет состояние модели.

Описание полей**geometry**

```
{geometry.base.Point} geometry
```

Геометрия остановки.

properties

```
{data.Manager} properties
```

Данные остановки. Доступны следующие поля:

- `Integer` – порядковый номер остановки во множестве остановок соответствующего транспортного сегмента;
- `id: String` – идентификатор остановки;
- `name: String` – название остановки;
- `lodIndex: Integer` – порядковый номер точки остановки во множестве всех координат соответствующего ей пути маршрута.

segment

```
{multiRouter.masstransit.TransportSegmentModel} segment
```

Ссылка на родительскую модель транспортного сегмента.

Описание событий**update**

Обновление модели новыми данными. Экземпляр класса [Event](#).

Описание методов**update**

```
{ } update(stopJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>stopJson *</code>	—	Тип: <code>Object</code> JSON-данные.

* Обязательный параметр/опция.

multiRouter.masstransit.TransferSegment

Примечание: Конструктор класса multiRouter.masstransit.TransferSegment скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение сегмента перехода маршрута общественного транспорта. Сегментом пути маршрута общественного транспорта называется часть пути от одной пересадки до другой.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение сегмента перехода маршрута общественного транспорта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.masstransit.TransferSegmentModel	Модель данных сегмента.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	data.Manager	Данные сегмента. Доступны следующие поля: <ul style="list-style-type: none"> index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; type: String - идентификатор типа сегмента, принимающий значение "transfer" для сегментов переходов; text: String - текстовое описание сегмента; distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах; lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление отображения сегмента. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.</p> <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	<p>Метод предоставляет синхронный доступ к оверлею.</p> <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	<p>Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.</p> <p>Унаследован от IChildOnMap.</p>

Имя	Возвращает	Описание
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

model

```
{multiRouter.masstransit.TransferSegmentModel} model
```

Модель данных сегмента.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- `index`: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- `type`: String - идентификатор типа сегмента, принимающий значение "transfer" для сегментов переходов;
- `text`: String - текстовое описание сегмента;
- `distance`: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах;
- `duration`: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах;
- `lodIndex`: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление отображения сегмента. Экземпляр класса [Event](#).

multiRouter.masstransit.TransferSegmentModel

Примечание: Конструктор класса `multiRouter.masstransit.TransferSegmentModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventEmitter](#).

Модель данных сегмента перехода пути маршрута общественного транспорта. Сегментом пути маршрута общественного транспорта называется часть пути от одной пересадки до другой.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных сегмента перехода пути маршрута общественного транспорта.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
<code>geometry</code>	geometry.base.LineString	Геометрия сегмента.

Имя	Тип	Описание
<code>path</code>	<code>multiRouter.masstransit.PathModel</code>	Ссылка на родительскую модель пути.
<code>properties</code>	<code>data.Manager</code>	<p>Данные сегмента. Доступны следующие поля:</p> <ul style="list-style-type: none"> <code>index</code>: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; <code>type</code>: String - идентификатор типа сегмента, принимающий значение "transfer" для сегментов переходов; <code>text</code>: String - текстовое описание сегмента; <code>distance</code>: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; <code>duration</code>: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах; <code>lodIndex</code>: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

События

Имя	Описание
<code>update</code>	Обновление модели новыми данными. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
<code>destroy(segmentJson)</code>		Обновляет состояние модели.
<code>getType()</code>	String	Возвращает идентификатор типа сегмента. Для сегментов перехода маршрутов общественного транспорта возвращается строка "transfer".

Описание полей

geometry

```
{geometry.base.LineString} geometry
```

Геометрия сегмента.

path

```
{multiRouter.masstransit.PathModel} path
```

Ссылка на родительскую модель пути.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- type: String - идентификатор типа сегмента, принимающий значение "transfer" для сегментов переходов;
- text: String - текстовое описание сегмента;
- distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах;
- duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах;
- lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#).

Описание методов

destroy

```
{ } destroy(segment.Json)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
segment.Json *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

getType

```
{String} getType()
```

Возвращает идентификатор типа сегмента. Для сегментов перехода маршрутов общественного транспорта возвращается строка "transfer".

multiRouter.masstransit.TransportSegment

Примечание: Конструктор класса multiRouter.masstransit.TransportSegment скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение транспортного сегмента маршрута общественного транспорта. Сегментом пути маршрута общественного транспорта называется часть пути от одной пересадки до другой.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение транспортного сегмента маршрута общественного транспорта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.masstransit.TransportSegmentModel	Модель данных сегмента.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	data.Manager	Данные сегмента. Доступны следующие поля: <ul style="list-style-type: none"> index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; String - идентификатор типа сегмента, принимающий значение "transport" для транспортных сегментов; text: String - текстовое описание сегмента; transports: ITransportProperties[] - массив, описывающий конкретные транспортные средства, с помощью которых возможно перемещение по данному сегменту; stops: Object - описание остановок в формате GeoJson:FeatureCollection; distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах; lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code>.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление отображения сегмента. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.</p> <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	<p>Метод предоставляет синхронный доступ к оверлею.</p> <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	<p>Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.</p> <p>Унаследован от IChildOnMap.</p>

Имя	Возвращает	Описание
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

model

```
{multiRouter.masstransit.TransportSegmentModel} model
```

Модель данных сегмента.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- String - идентификатор типа сегмента, принимающий значение "transport" для транспортных сегментов;
- text: String - текстовое описание сегмента;
- transports: [ITransportProperties\[\]](#) - массив, описывающий конкретные транспортные средства, с помощью которых возможно перемещение по данному сегменту;
- stops: Object - описание остановок в формате [GeoJson:FeatureCollection](#);
- distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах;
- duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах;
- lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление отображения сегмента. Экземпляр класса [Event](#).

multiRouter.masstransit.TransportSegmentModel

Примечание: Конструктор класса multiRouter.masstransit.TransportSegmentModel скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventEmitter](#).

Модель данных транспортного сегмента пути маршрута общественного транспорта. Сегментом пути маршрута общественного транспорта называется часть пути от одной пересадки до другой.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных транспортного сегмента пути маршрута общественного транспорта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
geometry	geometry.base.LineString	Геометрия сегмента.
path	multiRouter.masstransit.PathModel	Ссылка на родительскую модель пути.
properties	data.Manager	Данные сегмента. Доступны следующие поля: <ul style="list-style-type: none"> index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; String - идентификатор типа сегмента, принимающий значение "transport" для транспортных сегментов; text: String - текстовое описание сегмента; transports: ITransportProperties[] - массив, описывающий конкретные транспортные средства, с помощью которых возможно перемещение по данному сегменту; stops: Object - описание остановок в формате GeoJson:FeatureCollection; distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах; lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getStops()	multiRouter.masstransit.StopModel	Возвращает массив остановок на транспортном маршруте.

Имя	Возвращает	Описание
<code>getType()</code>	String	Возвращает идентификатор типа сегмента. Для транспортных сегментов маршрутов общественного транспорта возвращается строка "transport".
<code>update(segmentJson)</code>		Обновляет состояние модели.

Описание полей

geometry

```
{geometry.base.LineString} geometry
```

Геометрия сегмента.

path

```
{multiRouter.masstransit.PathModel} path
```

Ссылка на родительскую модель пути.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- String - идентификатор типа сегмента, принимающий значение "transport" для транспортных сегментов;
- text: String - текстовое описание сегмента;
- transports: [ITransportProperties\[\]](#) - массив, описывающий конкретные транспортные средства, с помощью которых возможно перемещение по данному сегменту;
- stops: Object - описание остановок в формате [GeoJson:FeatureCollection](#);
- distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах;
- duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах;
- lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#).

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getStops

```
{multiRouter.masstransit.StopModel[]} getStops()
```

Возвращает массив остановок на транспортном маршруте.

getType

```
{String} getType()
```

Возвращает идентификатор типа сегмента. Для транспортных сегментов маршрутов общественного транспорта возвращается строка "transport".

update

```
{ } update(segmentJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
segmentJson *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

multiRouter.masstransit.WalkSegment

Примечание: Конструктор класса `multiRouter.masstransit.WalkSegment` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение пешего сегмента маршрута общественного транспорта. Сегментом пути маршрута общественного транспорта называется часть пути от одной пересадки до другой.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение пешего сегмента маршрута общественного транспорта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.masstransit.WalkSegmentModel	Модель данных сегмента.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

Имя	Тип	Описание
properties	data.Manager	<p>Данные сегмента. Доступны следующие поля:</p> <ul style="list-style-type: none"> index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; type: String - идентификатор типа сегмента, принимающий значение "walk" для пешех сегментов; text: String - текстовое описание сегмента; distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах; lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.
state	IDataManager	<p>Состояние геообъекта.</p> <p>Унаследовано от IGeoObject.</p>

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>

Имя	Описание
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство touches, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> clientX - координата касания X относительно видимой области браузера; clientY - координата касания Y относительно видимой области браузера; pageX - координата касания X касания относительно начала документа; pageY - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление отображения сегмента. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей**model**

```
{multiRouter.masstransit.WalkSegmentModel} model
```

Модель данных сегмента.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- `index`: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- `type`: String - идентификатор типа сегмента, принимающий значение "walk" для пешех сегментов;
- `text`: String - текстовое описание сегмента;
- `distance`: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах;
- `duration`: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах;

- `lodIndex`: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление отображения сегмента. Экземпляр класса [Event](#).

multiRouter.masstransit.WalkSegmentModel

Примечание: Конструктор класса `multiRouter.masstransit.WalkSegmentModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных пешего сегмента пути маршрута общественного транспорта. Сегментом пути маршрута общественного транспорта называется часть пути от одной пересадки до другой.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных пешего сегмента пути маршрута общественного транспорта.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventManager .
geometry	geometry.base.LineString	Геометрия сегмента.
path	multiRouter.masstransit.PathModel	Ссылка на родительскую модель пути.
properties	data.Manager	Данные сегмента. Доступны следующие поля: <ul style="list-style-type: none"> • <code>index</code>: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; • <code>type</code>: String - идентификатор типа сегмента, принимающий значение "walk" для пешеходных сегментов; • <code>text</code>: String - текстовое описание сегмента; • <code>distance</code>: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; • <code>duration</code>: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах; • <code>lodIndex</code>: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
<code>destroy()</code>		Уничтожает модель.
<code>getType()</code>	String	Возвращает идентификатор типа сегмента. Для пеших сегментов маршрутов общественного транспорта возвращается строка "walk".

Описание полей

geometry

```
{geometry.base.LineString} geometry
```

Геометрия сегмента.

path

```
{multiRouter.masstransit.PathModel} path
```

Ссылка на родительскую модель пути.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- type: String - идентификатор типа сегмента, принимающий значение "walk" для пеших сегментов;
- text: String - текстовое описание сегмента;
- distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах;
- duration: Object - объект с полями "text" и "value", описывающий время проезда сегмента в секундах;
- lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#).

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getType

```
{String} getType()
```

Возвращает идентификатор типа сегмента. Для пеших сегментов маршрутов общественного транспорта возвращается строка "walk".

multiRouter.MultiRoute

Расширяет [IGeoObject](#).
Мультимаршрут на карте. Позволяет отображать на карте маршрут и несколько альтернатив к нему. Предоставляет интерфейс выбора активного маршрута.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
multiRouter.MultiRoute(model[, options])
```

Создает мультимаршрут на карте.

Параметры:

Параметр	Значение по умолчанию	Описание
model *	—	Тип: multiRouter.MultiRouteModel IMultiRouteModel.Json Модель данных мультимаршрута, либо объект описания модели. В случае передачи объекта описания, модель на его основе создается автоматически.

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции мультимаршрута. Для задания опций составных частей маршрутов доступны следующие префиксы:</p> <ul style="list-style-type: none"> wayPoint – опции путевых точек; wayPointStart – опции отображения начальной путевой точки; wayPointFinish – опции отображения конечной путевой точки; viaPoint – опции транзитных точек; pin – опции точечных маркеров на маршруте; editor – опции редактора мультимаршрута (см. multiRouter.Editor). <p>Для задания опций единичных маршрутов доступны следующие префиксы:</p> <ul style="list-style-type: none"> route – опции маршрутов, в том числе неактивных; routeActive – опции активного маршрута. <p>Причем опции заданные с префиксом "routeActive" более приоритетны по отношению к опциям с префиксом "route". Для задания опций составных частей маршрутов общественного транспорта доступны следующие префиксы.</p> <ul style="list-style-type: none"> routeMarker – опции меток сегментов маршрута; routeWalkMarker – опции меток пешеходных сегментов маршрута; routeTransferMarker – опции меток сегментов переходов; routeTransportMarker – опции меток транспортных сегментов маршрута; routeWalkSegment – опции линий пешеходных сегментов маршрута; routeTransferSegment – опции линий сегментов переходов; routeTransportSegment – опции линий транспортных сегментов маршрута; routePedestrianSegment – опции линий сегментов пешеходного маршрута. <p>Также данные префиксы доступны в "routeActive*" варианте.</p>

Параметр	Значение по умолчанию	Описание
options.activeRouteAutoSelection	true	Тип: Boolean После обновления данных автоматически выбирать в качестве активного маршрут с минимальным временем проезда.
options.boundsAutoApply	false	Тип: Boolean При добавлении мультимаршрута на карту позволяет автоматически установить центр и коэффициент масштабирования так, чтобы мультимаршрут был виден целиком.
options.dragUpdateInterval	—	Тип: String Number Величина временного интервала перестроения маршрута при перетаскивании опорных точек. Может быть задана в миллисекундах, либо оптимальное значение будет рассчитываться автоматически. Значение данной опции транслируется в параметр IMultiRouteParams.requestSendInterval модели мультимаршрута.
options.preventDragUpdate	false	Тип: Boolean Позволяет отключить обновление маршрута во время перетаскивания опорных точек.
options.useMapMargin	true	Тип: Boolean Нужно ли учитывать отступы карты map.margin.Manager .
options.zoomMargin	0	Тип: Number Number[] Отступы от границ видимой области карты при изменении коэффициента масштабирования. Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы top, right, bottom, left. При включенной опции "useMapMargin" значение "zoomMargin" складывается со значениями, которые были рассчитаны в менеджере отступов map.margin.Manager .

* Обязательный параметр/опция.

Примеры:

1.

```
// Создадим мульти-маршрут и добавим его на карту.
var multiRoute = new ymaps.multiRouter.MultiRoute({
  referencePoints: ['Москва, Ленинский проспект', 'Москва, Кулаков переулок'],
}, {
  editorDrawOver: false,
  waypointDraggable: true,
  viaPointDraggable: true,
  // Зададим собственное оформление линий мультимаршрута.
  routeStrokeColor: "000088",
  routeActiveStrokeColor: "ff0000",
  pinIconFillColor: "ff0000",
  boundsAutoApply: true,
  zoomMargin: 30
});
myMap.geoObjects.add(multiRoute);
```

2.

```
// Создадим мультимаршрут и добавим его на карту.
var multiRoute = new ymaps.multiRouter.MultiRoute({
  referencePoints: ['Москва, Ленинский проспект', 'Москва, Кулаков переулок', 'Зеленоград'],
});
myMap.geoObjects.add(multiRoute);

// Once multi-route is loaded.
multiRoute.events.once('update', function () {
  // Установим первый маршрут, у которого нет перекрытых
  // участков, в качестве активного. Откроем его балун.
  var routes = multiRoute.getRoutes();
  for (var i = 0, l = routes.getLength(); i < l; i++) {
    var route = routes.get(i);
    if (!route.properties.get('blocked')) {
      multiRoute.setActiveRoute(route);
      route.balloon.open();
      break;
    }
  }
});
```

Поля

Имя	Тип	Описание
editor	multiRouter.EditorAddOn	Редактор мультимаршрута.
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.MultiRouteModel	Модель данных мультимаршрута.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
activeroutechange	Изменение активного маршрута. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>oldActiveRoute</code>: multiRouter.driving.Route multiRouter.masstransit.Route null - предыдущий активный маршрут.

Имя	Описание
balloonclose	<p>Заккрытие балуна. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalTarget: multiRouter.pedestrian.Route multiRouter.driving.Route multiRouter.masstransit.Route null - маршрут, балун которого был закрыт.
balloonopen	<p>Открытие балуна. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalTarget: multiRouter.pedestrian.Route multiRouter.driving.Route multiRouter.masstransit.Route null - маршрут, на котором был открыт балун.
boundsautoapply	<p>Событие возникает в момент установки центра и коэффициента масштабирования карты для оптимального отображения мультимаршрута. Также смотрите описание опции boundsAutoApply. Экземпляр класса Event.</p>
boundschange	<p>Изменение координат географической области, охватывающей мультимаршрут. Экземпляр класса Event.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
pixelboundschange	<p>Изменение пиксельных координат области, охватывающей мультимаршрут. Экземпляр класса Event.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление мультимаршрута. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
<code>getActiveRoute()</code>	<code>multiRouter.driving.Route</code> <code>multiRouter.transit.Route</code> <code>null</code>	Возвращает активный маршрут.
<code>getBounds()</code>	<code>Number[][]</code> <code>null</code>	Возвращает географические координаты области, охватывающей мультимаршрут.
<code>getMap()</code>	<code>Map</code>	Возвращает ссылку на карту. Унаследован от <code>IParentOnMap</code> .
<code>getOverlay()</code>	<code>vow.Promise</code>	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от <code>IGeoObject</code> .
<code>getOverlaySync()</code>	<code>IOverlay</code> <code>null</code>	Метод предоставляет синхронный доступ к оверлею. Унаследован от <code>IGeoObject</code> .
<code>getParent()</code>	<code>IParentOnMap</code> <code>null</code>	Возвращает ссылку на родительский объект или <code>null</code> , если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>getPixelBounds()</code>	<code>Number[][]</code> <code>null</code>	Возвращает глобальные пиксельные координаты области, охватывающей мультимаршрут.
<code>getRoutes()</code>	<code>GeoObjectCollection</code>	Возвращает дочернюю коллекцию единичных маршрутов мультимаршрута.
<code>getViaPoints()</code>	<code>GeoObjectCollection</code>	Возвращает дочернюю коллекцию транзитных точек мультимаршрута.
<code>getWayPoints()</code>	<code>GeoObjectCollection</code>	Возвращает дочернюю коллекцию путевых точек мультимаршрута.
<code>setActiveRoute(route)</code>		Устанавливает активный маршрут. При этом предыдущий активный маршрут деактивируется, и генерируется событие <code>multiRouter.MultiRoute.activeroutechange</code> .

Имя	Возвращает	Описание
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

editor

```
{multiRouter.EditorAddOn} editor
```

Редактор мультимаршрута.

Пример:

```
// Начало редактирования маршрута.
ymaps.route(['Москва', 'Питер'], { multiRoute: true })
  .done(function (multiRoute) {
    myMap.geoObjects.add(multiRoute);
    multiRoute.editor.start({
      addWayPoints: true,
      removeWayPoints: true
    });
    // ...
    // Завершение редактирования маршрута.
    route.editor.stop();
  });
```

model

```
{multiRouter.MultiRouteModel} model
```

Модель данных мультимаршрута.

Описание событий

activeroutechange

Изменение активного маршрута. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- oldActiveRoute: `multiRouter.driving.Route|multiRouter.masstransit.Route|null` - предыдущий активный маршрут.

balloonclose

Закрытие балуна. Имена полей, доступных через метод [Event.get](#):

- originalTarget: `multiRouter.pedestrian.Route|multiRouter.driving.Route|multiRouter.masstransit.Route|null` - маршрут, балун которого был закрыт.

balloonopen

Открытие балуна. Имена полей, доступных через метод [Event.get](#):

- originalTarget: `multiRouter.pedestrian.Route|multiRouter.driving.Route|multiRouter.masstransit.Route|null` - маршрут, на котором был открыт балун.

boundsautoapply

Событие возникает в момент установки центра и коэффициента масштабирования карты для оптимального отображения мультимаршрута. Также смотрите описание опции `boundsAutoApply`. Экземпляр класса [Event](#).

boundschange

Изменение координат географической области, охватывающей мультимаршрут. Экземпляр класса [Event](#).

pixelboundschange

Изменение пиксельных координат области, охватывающей мультимаршрут. Экземпляр класса [Event](#).

update

Обновление мультимаршрута. Экземпляр класса [Event](#).

Описание методов**getActiveRoute**

```
{multiRouter.driving.Route|multiRouter.masstransit.Route|null} getActiveRoute()
```

Возвращает активный маршрут.

getBounds

```
{Number[][]|null} getBounds()
```

Возвращает географические координаты области, охватывающей мультимаршрут.

getPixelBounds

```
{Number[][]|null} getPixelBounds()
```

Возвращает глобальные пиксельные координаты области, охватывающей мультимаршрут.

getRoutes

```
{GeoObjectCollection} getRoutes()
```

Возвращает дочернюю коллекцию единичных маршрутов мультимаршрута.

getViaPoints

```
{GeoObjectCollection} getViaPoints()
```

Возвращает дочернюю коллекцию транзитных точек мультимаршрута.

getWayPoints

```
{GeoObjectCollection} getWayPoints()
```

Возвращает дочернюю коллекцию путевых точек мультимаршрута.

setActiveRoute

```
{ } setActiveRoute(route)
```

Устанавливает активный маршрут. При этом предыдущий активный маршрут деактивируется, и генерируется событие [multiRouter.MultiRoute.activeroutechange](#).

Параметры:

Параметр	Значение по умолчанию	Описание
route *	—	Тип: multiRouter.driving.Route multiRouter.masstransit.Route null Устанавливает активный маршрут.

* Обязательный параметр/опция.

multiRouter.MultiRouteModel

Расширяет [IEventEmitter](#).

Модель данных мультимаршрута

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
multiRouter.MultiRouteModel(referencePoints[, params])
```

Создает модель данных мультимаршрута.

Параметры:

Параметр	Значение по умолчанию	Описание
referencePoints *	—	Тип: IMultiRouteReferencePoint [] Описание опорных точек мультимаршрута.
params	—	Тип: IMultiRouteParams Параметры маршрутизации.

* Обязательный параметр/опция.

Пример:

```
// Создаем модель мультимаршрутизатора.
var multiRouteModel = new ymaps.multiRouter.MultiRouteModel(['Moscow', 'Tver', 'Peterburg'], {
  avoidTrafficJams: true,
  viaIndexes: [1]
});

// Создаем представление мультимаршрутизатора на основе модели.
var multiRouteView = new ymaps.multiRouter.MultiRoute(multiRouteModel);
myMap.geoObjects.add(multiRouteView);

// Подписываемся на события модели.
multiRouteView.model.events
  .add("requestsuccess", function (event) {
    var routes = event.get("target").getRoutes();
    console.log("Found routes: " + routes.length);
    for (var i = 0, l = routes.length; i < l; i++) {
      console.log("Route length " + (i + 1) + ": " + routes[i].properties.get("distance").text);
    }
  })
  .add("requestfail", function (event) {
    console.log("Error: " + event.get("error").message);
  });
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
properties	data.Manager	Данные мультимаршрута.

События

Имя	Описание
requestcancel	Запрос за данными отменен. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>referencePoints</code>: <code>Object[]</code> - массив, описывающий множество опорных точек; <code>params</code>: <code>Object</code> – параметры маршрутизации.
requestchange	Изменились опорные данные модели мультимаршрута (опорные точки, либо параметры маршрутизации). В результате формируется новый запрос за данными к сервису маршрутизации. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>referencePoints</code>: <code>Object[]</code> - массив, описывающий множество опорных точек; <code>params</code>: <code>Object</code> - параметры маршрутизации. <code>oldReferencePoints</code>: <code>Object[]</code> – массив, описывающий предыдущее множество опорных точек; <code>oldParams</code>: <code>Object</code> - предыдущие параметры маршрутизации.
requestfail	Запрос за данными завершился с ошибкой. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>error</code>: <code>Error</code> – объект ошибки.
requestsend	Отправлен новый запрос за данными модели мультимаршрута. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>referencePoints</code>: <code>Object[]</code> - массив, описывающий множество опорных точек; <code>params</code>: <code>Object</code> – параметры маршрутизации.

Имя	Описание
requestsuccess	<p>Запрос за данными завершился успешно, модель данных обновлена. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>init</code>: Boolean – признак инициализирующего запроса; <code>rough</code>: Boolean - признак промежуточного запроса (используется для оптимизации размера ответа сервера при редактировании маршрута); <code>wayPointsChange</code>: Boolean - признак изменений в наборе путевых точек; <code>viaPointsChange</code>: Boolean - признак изменений в наборе транзитных точек; <code>routesChange</code>: Boolean - признак изменений в наборе маршрутов.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getAllPoints()	(multiRouter.WayPointModel multiRouter.ViaPointModel) []	Возвращает объединенный массив моделей транзитных и путевых точек в порядке, соответствующем порядку следования соответствующих опорных точек.
getJson()	Object	Возвращает JSON-данные модели мультимаршрута.
getParams()	IMultiRouteParams	Возвращает объект, описывающий текущие значения параметров маршрутизации.
getPoints()	(multiRouter.WayPointModel multiRouter.ViaPointModel) []	Устаревшее название метода multiRouter.MultiRouteModel.getAllPoints . Не рекомендуется к использованию.

Имя	Возвращает	Описание
getReferencePointIndexes()	Object	Возвращает объект, содержащий следующие поля: <ul style="list-style-type: none"> way: Integer[] - индексы опорных точек, соответствующие множеству путевых точек модели; via: Integer[] - индексы опорных точек, соответствующие множеству транзитных точек модели.
getReferencePoints()	IMultiRouteReferencePoint[]	Возвращает массив опорных точек мультимаршрута.
getRoutes()	multiRouter.driving.RouteModel[] и multiRouter.transit.RouteModel[]	Возвращает массив моделей дочерних маршрутов.
getViaPoints()	multiRouter.ViaPointModel[]	Возвращает массив моделей дочерних транзитных точек.
getWayPoints()	multiRouter.WayPointModel[]	Возвращает массив моделей дочерних путевых точек.
setParams(params[, extend[, clearRequests]])		Задаёт параметры маршрутизации.
setReferencePoints(referencePoints[, vialIndexes[, clearRequests]])		Задаёт опорные точки мультимаршрута.

Описание полей

properties

```
{data.Manager} properties
```

Данные мультимаршрута.

Описание событий

requestcancel

Запрос за данными отменен. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- referencePoints: Object[] - массив, описывающий множество опорных точек;
- params: Object – параметры маршрутизации.

requestchange

Изменились опорные данные модели мультимаршрута (опорные точки, либо параметры маршрутизации). В результате формируется новый запрос за данными к сервису маршрутизации. Экземпляр класса `Event`. Имена полей, доступных через метод `Event.get`:

- `referencePoints`: `Object[]` - массив, описывающий множество опорных точек;
- `params`: `Object` - параметры маршрутизации.
- `oldReferencePoints`: `Object[]` – массив, описывающий предыдущее множество опорных точек;
- `oldParams`: `Object` - предыдущие параметры маршрутизации.

requestfail

Запрос за данными завершился с ошибкой. Экземпляр класса `Event`. Имена полей, доступных через метод `Event.get`:

- `error`: `Error` – объект ошибки.

requestsend

Отправлен новый запрос за данными модели мультимаршрута. Экземпляр класса `Event`. Имена полей, доступных через метод `Event.get`:

- `referencePoints`: `Object[]` - массив, описывающий множество опорных точек;
- `params`: `Object` – параметры маршрутизации.

requestsuccess

Запрос за данными завершился успешно, модель данных обновлена. Экземпляр класса `Event`. Имена полей, доступных через метод `Event.get`:

- `init`: `Boolean` – признак инициализирующего запроса;
- `rough`: `Boolean` - признак промежуточного запроса (используется для оптимизации размера ответа сервера при редактировании маршрута);
- `wayPointsChange`: `Boolean` - признак изменений в наборе путевых точек;
- `viaPointsChange`: `Boolean` - признак изменений в наборе транзитных точек;
- `routesChange`: `Boolean` - признак изменений в наборе маршрутов.

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getAllPoints

```
{(multiRouter.WayPointModel|multiRouter.ViaPointModel)[]} getAllPoints()
```

Возвращает объединенный массив моделей транзитных и путевых точек в порядке, соответствующем порядку следования соответствующих опорных точек.

getJson

```
{Object} getJson()
```

Возвращает JSON-данные модели мультимаршрута.

getParams

```
{IMultiRouteParams} getParams()
```

Возвращает объект, описывающий текущие значения параметров маршрутизации.

getPoints

```
{(multiRouter.WayPointModel|multiRouter.ViaPointModel)[]} getPoints()
```

Устаревшее название метода `multiRouter.MultiRouteModel.getAllPoints`. Не рекомендуется к использованию.

Возвращает объединенный массив моделей транзитных и путевых точек в порядке, соответствующем порядку следования соответствующих опорных точек.

getReferencePointIndexes

```
{Object} getReferencePointIndexes()
```

Возвращает объект, содержащий следующие поля:

- `way: Integer[]` - индексы опорных точек, соответствующие множеству путевых точек модели;
- `via: Integer[]` - индексы опорных точек, соответствующие множеству транзитных точек модели.

getReferencePoints

```
{IMultiRouteReferencePoint[]} getReferencePoints()
```

Возвращает массив опорных точек мультимаршрута.

getRoutes

```
{multiRouter.driving.RouteModel[]|multiRouter.masstransit.RouteModel[]} getRoutes()
```

Возвращает массив моделей дочерних маршрутов.

getViaPoints

```
{multiRouter.ViaPointModel[]} getViaPoints()
```

Возвращает массив моделей дочерних транзитных точек.

getWayPoints

```
{multiRouter.WayPointModel[]} getWayPoints()
```

Возвращает массив моделей дочерних путевых точек.

setParams

```
{ } setParams(params[, extend[, clearRequests]])
```

Задаёт параметры маршрутизации.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>params</code> *	—	Тип: <code>IMultiRouteParams</code> Параметры маршрутизации.

Параметр	Значение по умолчанию	Описание
extend	false	Тип: Boolean Позволяет изменить лишь часть параметров, а значения не переданных оставить прежним.
clearRequests	false	Тип: Boolean Позволяет очистить очередь предыдущих запросов к серверу.

* Обязательный параметр/опция.

setReferencePoints

```
{ } setReferencePoints(referencePoints[, viaIndexes[, clearRequests]])
```

Задаёт опорные точки мультимаршрута.

Параметры:

Параметр	Значение по умолчанию	Описание
referencePoints *	—	Тип: IMultiRouteReferencePoint[] Массив опорных точек.
viaIndexes	—	Тип: Integer[] Индексы транзитных точек в массиве опорных точек.
clearRequests	false	Тип: Boolean Позволяет очистить очередь предыдущих запросов к серверу.

* Обязательный параметр/опция.

multiRouter.pedestrian

multiRouter.pedestrian.Path

Примечание: Конструктор класса `multiRouter.pedestrian.Path` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение пути пешеходного маршрута. Единичный маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение пути пешеходного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.pedestrian.PathModel	Модель данных пути мультимаршрута.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	data.Manager	Модель данных пути мультимаршрута. Доступны следующие поля: <ul style="list-style-type: none"> index: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута; type: String - идентификатор типа маршрута, принимающий значение "pedestrian" для пешеходных маршрутов; distance: Object - объект с полями text и value, описывающий протяженность пути в метрах; duration: Object - объект с полями text и value, описывающий время движения по пути в секундах; coordinates: Number[][] - координаты всех точек пути; encodedCoordinates: String - строка закодированных в base64 координат всех точек пути.
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
update	Обновление отображения пути. Экземпляр класса Event .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getSegmentMarkers()	GeoObjectCollection	Возвращает дочернюю коллекцию маркеров сегментов.
getSegments()	GeoObjectCollection	Возвращает дочернюю коллекцию сегментов, из которых состоит путь.

Имя	Возвращает	Описание
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание полей

model

```
{multiRouter.pedestrian.PathModel} model
```

Модель данных пути мультимаршрута.

properties

```
{data.Manager} properties
```

Модель данных пути мультимаршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута;
- `type`: String - идентификатор типа маршрута, принимающий значение "pedestrian" для пешеходных маршрутов;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность пути в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время движения по пути в секундах;
- `coordinates`: Number[][] - координаты всех точек пути;
- `encodedCoordinates`: String - строка закодированных в base64 координат всех точек пути.

Описание событий

update

Обновление отображения пути. Экземпляр класса [Event](#).

Описание методов

getSegmentMarkers

```
{GeoObjectCollection} getSegmentMarkers()
```

Возвращает дочернюю коллекцию маркеров сегментов.

getSegments

```
{GeoObjectCollection} getSegments()
```

Возвращает дочернюю коллекцию сегментов, из которых состоит путь.

multiRouter.pedestrian.PathModel

Примечание: Конструктор класса `multiRouter.pedestrian.PathModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventEmitter](#).

Модель данных пути пешеходного маршрута. Единичный маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

[Поля](#) | [События](#) | [Методы](#)

Создание модели данных пути пешеходного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
properties	data.Manager	Модель данных пути мультимаршрута. Доступны следующие поля: <ul style="list-style-type: none"> <code>index</code>: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута; <code>type</code>: String - идентификатор типа маршрута, принимающий значение "pedestrian" для пешеходных маршрутов; <code>distance</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий протяженность пути в метрах; <code>duration</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий время движения по пути в секундах; <code>coordinates</code>: Number[][] - координаты всех точек пути; <code>encodedCoordinates</code>: String - строка закодированных в base64 координат всех точек пути.
route	multiRouter.pedestrian.RouteModel	Ссылка на родительскую модель маршрута.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>segmentsChange</code>: Boolean – признак изменений в наборе сегментов.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getSegments()	multiRouter.pedestrian.SegmentModel	Возвращает массив сегментов пути.
getType()	String	Возвращает идентификатор типа маршрутизации. Для пешеходных маршрутов возвращается строка "pedestrian".

Имя	Возвращает	Описание
<code>update(pathJson)</code>		Обновляет состояние модели.

Описание полей

properties

```
{data.Manager} properties
```

Модель данных пути мультимаршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер пути в соответствующем маршруте мультимаршрута;
- `type`: String - идентификатор типа маршрута, принимающий значение "pedestrian" для пешеходных маршрутов;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность пути в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время движения по пути в секундах;
- `coordinates`: Number[][] - координаты всех точек пути;
- `encodedCoordinates`: String - строка закодированных в base64 координат всех точек пути.

route

```
{multiRouter.pedestrian.RouteModel} route
```

Ссылка на родительскую модель маршрута.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `segmentsChange`: Boolean – признак изменений в наборе сегментов.

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getSegments

```
{multiRouter.pedestrian.SegmentModel[]} getSegments()
```

Возвращает массив сегментов пути.

getType

```
{String} getType()
```

Возвращает идентификатор типа маршрутизации. Для пешеходных маршрутов возвращается строка "pedestrian".

update

```
{ } update(pathJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>pathJson</code> *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

`multiRouter.pedestrian.Route`

Примечание: Конструктор класса `multiRouter.pedestrian.Route` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение единичного пешеходного маршрута. Мультимаршрут может состоять из нескольких отдельных маршрутов.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение единичного пешеходного маршрута.

Поля

Имя	Тип	Описание
<code>balloon</code>	IMultiRouterRouteBalloon	Балун маршрута.
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
<code>geometry</code>	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
<code>model</code>	multiRouter.pedestrian.RouteModel	Модель данных единичного маршрута.
<code>options</code>	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
<code>properties</code>	data.Manager	Данные маршрута. Доступны следующие поля: <ul style="list-style-type: none"> <code>index</code>: Integer - порядковый номер маршрута в мультимаршруте; <code>type</code>: String - идентификатор типа маршрута, принимающий значение "pedestrian" для пешеходных маршрутов; <code>distance</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий протяженность маршрута в метрах; <code>duration</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий время движения по маршруту в секундах; <code>boundedBy</code>: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.
<code>state</code>	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
balloonclose	Закрытие балуна.
balloonopen	Открытие балуна.
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
update	Обновление отображения маршрута. Экземпляр класса Event .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .

Имя	Возвращает	Описание
<code>getParent()</code>	<code>IParentOnMap</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>getPaths()</code>	<code>GeoObjectCollection</code>	Возвращает дочернюю коллекцию путей, из которых состоит маршрут.
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание полей

balloon

```
{IMultiRouterRouteBalloon} balloon
```

Балун маршрута.

model

```
{multiRouter.pedestrian.RouteModel} model
```

Модель данных единичного маршрута.

properties

```
{data.Manager} properties
```

Данные маршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер маршрута в мультимаршруте;
- `type`: String - идентификатор типа маршрута, принимающий значение "pedestrian" для пешеходных маршрутов;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность маршрута в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время движения по маршруту в секундах;
- `boundedBy`: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

Описание событий

balloonclose

Закрытие балуна.

balloonopen

Открытие балуна.

update

Обновление отображения маршрута. Экземпляр класса [Event](#).

Описание методов**getPaths**

```
{GeoObjectCollection} getPaths()
```

Возвращает дочернюю коллекцию путей, из которых состоит маршрут.

multiRouter.pedestrian.RouteModel

Примечание: Конструктор класса `multiRouter.pedestrian.RouteModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных единичного пешеходного маршрута. Мультимаршрут может состоять из нескольких отдельных маршрутов.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных единичного пешеходного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventManager .
multiRoute	multiRouter.MultiRouteModel	Ссылка на родительскую модель маршрута.
properties	data.Manager	Данные маршрута. Доступны следующие поля: <ul style="list-style-type: none">• <code>index</code>: Integer - порядковый номер маршрута в мультимаршруте;• <code>type</code>: String - идентификатор типа маршрута, принимающий значение "pedestrian" для пешеходных маршрутов;• <code>distance</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий протяженность маршрута в метрах;• <code>duration</code>: Object - объект с полями <code>text</code> и <code>value</code>, описывающий время движения по маршруту в секундах;• <code>boundedBy</code>: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"><code>pathsChange</code>: Boolean - признак изменений в наборе путей.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getPaths()	multiRouter.pedestrian.PathModel	Возвращает массив путей маршрута.
getType()	String	Возвращает идентификатор типа маршрутизации. Для пешеходных маршрутов возвращается строка "pedestrian".
update(routeJson)		Обновляет состояние модели.

Описание полей

multiRoute

```
{multiRouter.MultiRouteModel} multiRoute
```

Ссылка на родительскую модель маршрута.

properties

```
{data.Manager} properties
```

Данные маршрута. Доступны следующие поля:

- `index`: Integer - порядковый номер маршрута в мультимаршруте;
- `type`: String - идентификатор типа маршрута, принимающий значение "pedestrian" для пешеходных маршрутов;
- `distance`: Object - объект с полями `text` и `value`, описывающий протяженность маршрута в метрах;
- `duration`: Object - объект с полями `text` и `value`, описывающий время движения по маршруту в секундах;
- `boundedBy`: Number[][] - координаты нижнего и верхнего углов прямоугольника, ограничивающего маршрут.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pathsChange`: Boolean - признак изменений в наборе путей.

Описание методов

destroy

```
{ } destroy()
```


Уничтожает модель.

getPaths

```
{multiRouter.pedestrian.PathModel[]} getPaths()
```

Возвращает массив путей маршрута.

getType

```
{String} getType()
```

Возвращает идентификатор типа маршрутизации. Для пешеходных маршрутов возвращается строка "pedestrian".

update

```
{ } update(routeJson)
```

Обновляет состояние модели.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>routeJson</code> *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

multiRouter.pedestrian.SegmentModel

Примечание: Конструктор класса `multiRouter.pedestrian.SegmentModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventEmitter](#).

Модель данных пешего сегмента пешеходного маршрута.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных пешего сегмента.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
<code>geometry</code>	geometry.base.LineString	Геометрия сегмента.
<code>path</code>	multiRouter.pedestrian.PathModel	Ссылка на родительскую модель пути.

Имя	Тип	Описание
properties	data.Manager	<p>Данные сегмента. Доступны следующие поля:</p> <ul style="list-style-type: none"> index: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута; type: String - идентификатор типа сегмента, принимающий значение "pedestrian" для пешех сегментов; text: String - текстовое описание сегмента; distance: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах; duration: Object - объект с полями text и value, описывающий время движения по сегменту в секундах; lodIndex: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

События

Имя	Описание
update	Обновление модели новыми данными. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getType()	String	Возвращает идентификатор типа сегмента. Для пешех сегментов пешеходных маршрутов возвращается строка "pedestrian".

Описание полей

geometry

```
{geometry.base.LineString} geometry
```

Геометрия сегмента.

path

```
{multiRouter.pedestrian.PathModel} path
```

Ссылка на родительскую модель пути.

properties

```
{data.Manager} properties
```

Данные сегмента. Доступны следующие поля:

- `index`: Integer – порядковый номер сегмента во множестве сегментов соответствующего пути маршрута;
- `type`: String - идентификатор типа сегмента, принимающий значение "pedestrian" для пеших сегментов;
- `text`: String - текстовое описание сегмента;
- `distance`: Object - объект с полями "text" и "value", описывающий протяженность сегмента в метрах;
- `duration`: Object - объект с полями text и value, описывающий время движения по сегменту в секундах;
- `lodIndex`: Integer - порядковый номер первой точки сегмента во множестве всех координат соответствующего ему пути маршрута.

Описание событий

update

Обновление модели новыми данными. Экземпляр класса [Event](#).

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getType

```
{String} getType()
```

Возвращает идентификатор типа сегмента. Для пеших сегментов пешеходных маршрутов возвращается строка "pedestrian".

multiRouter.pedestrian.WalkSegment

Примечание: Конструктор класса `multiRouter.pedestrian.WalkSegment` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение пешего сегмента пешеходного маршрута.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение пешего сегмента пешеходного маршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.</p> <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	<p>Метод предоставляет синхронный доступ к оверлею.</p> <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	<p>Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.</p> <p>Унаследован от IChildOnMap.</p>

Имя	Возвращает	Описание
setParent(parent)	IChildOnMap	<p>Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта.</p> <p>Унаследован от IChildOnMap.</p>

multiRouter.ViaPoint

Примечание: Конструктор класса multiRouter.ViaPoint скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение транзитной точки. Транзитные точки не подразумевают остановку. Таким образом, при прохождении через транзитную точку, сегмент пути маршрута не прерывается.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение транзитной точки.

Поля

Имя	Тип	Описание
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometry	IGeometry null	<p>Геометрия геообъекта.</p> <p>Унаследовано от IGeoObject.</p>
model	multiRouter.ViaPointModel	Модель данных транзитной точки.
options	IOptionManager	<p>Менеджер опций.</p> <p>Унаследовано от ICustomizable.</p>
properties	data.Manager	<p>Данные транзитной точки мультимаршрута. Доступны следующие поля:</p> <ul style="list-style-type: none"> index: Integer – порядковый номер точки; lodIndex: Integer - порядковый номер транзитной точки во множестве всех координат соответствующего ей пути маршрута.
state	IDataManager	<p>Состояние геообъекта.</p> <p>Унаследовано от IGeoObject.</p>

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code>.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>

Имя	Описание
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление отображения транзитной точки. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	<p>Возвращает ссылку на карту.</p> <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	<p>Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.</p> <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	<p>Метод предоставляет синхронный доступ к оверлею.</p> <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	<p>Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.</p> <p>Унаследован от IChildOnMap.</p>

Имя	Возвращает	Описание
<code>setParent(parent)</code>	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

model

```
{multiRouter.ViaPointModel} model
```

Модель данных транзитной точки.

properties

```
{data.Manager} properties
```

Данные транзитной точки мультимаршрута. Доступны следующие поля:

- `index: Integer` – порядковый номер точки;
- `lodIndex: Integer` - порядковый номер транзитной точки во множестве всех координат соответствующего ей пути маршрута.

Описание событий

update

Обновление отображения транзитной точки. Экземпляр класса [Event](#).

multiRouter.ViaPointModel

Примечание: Конструктор класса `multiRouter.ViaPointModel` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных транзитной точки мультимаршрута. Транзитные точки не подразумевают остановку. Таким образом, при прохождении через транзитную точку, сегмент пути маршрута не прерывается.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных транзитной точки мультимаршрута.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventManager .
<code>geometry</code>	geometry.base.Point	Геометрия транзитной точки мультимаршрута.
<code>multiRoute</code>	multiRouter.MultiRouteModel	Ссылка на родительскую модель маршрута.

Имя	Тип	Описание
properties	data.Manager	<p>Данные транзитной точки мультимаршрута. Доступны следующие поля:</p> <ul style="list-style-type: none"> index: Integer – порядковый номер точки; lodIndex: Integer - порядковый номер транзитной точки во множестве всех координат соответствующего ей пути маршрута.

События

Имя	Описание
referencepointchange	<p>Изменена опорная точка. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> oldReferencePoint: Object - описание предыдущей опорной точки.
update	Обновление модели новыми данными. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает модель.
getReferencePoint()	Object	Возвращает соответствующую опорную точку.
getReferencePointIndex()	Integer	Возвращает индекс соответствующей опорной точки во множестве опорных точек родительского мультимаршрута.
setReferencePoint(referencePoint)		<p>Задаёт описание соответствующей опорной точки. Опорная точка может быть задана одним из следующих способов:</p> <ul style="list-style-type: none"> строка, содержащая почтовый адрес опорной точки; массив, содержащий широту и долготу опорной точки; геометрия <code>geometry.Point</code>, описывающая опорную точку.

Имя	Возвращает	Описание
<code>update(viaPointJson)</code>		Обновляет модель новыми данными.

Описание полей

geometry

```
{geometry.base.Point} geometry
```

Геометрия транзитной точки мультимаршрута.

multiRoute

```
{multiRouter.MultiRouteModel} multiRoute
```

Ссылка на родительскую модель маршрута.

properties

```
{data.Manager} properties
```

Данные транзитной точки мультимаршрута. Доступны следующие поля:

- `index`: Integer – порядковый номер точки;
- `lodIndex`: Integer - порядковый номер транзитной точки во множестве всех координат соответствующего ей пути маршрута.

Описание событий

referencepointchange

Изменена опорная точка. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldReferencePoint`: Object - описание предыдущей опорной точки.

update

Обновление модели новыми данными. Экземпляр класса [Event](#).

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getReferencePoint

```
{Object} getReferencePoint()
```

Возвращает соответствующую опорную точку.

getReferencePointIndex

```
{Integer} getReferencePointIndex()
```

Возвращает индекс соответствующей опорной точки во множестве опорных точек родительского мультимаршрута.

setReferencePoint

```
{ } setReferencePoint(referencePoint)
```

Задаёт описание соответствующей опорной точки. Опорная точка может быть задана одним из следующих способов:

- строка, содержащая почтовый адрес опорной точки;
- массив, содержащий широту и долготу опорной точки;
- геометрия `geometry.Point`, описывающая опорную точку.

Параметры:

Параметр	Значение по умолчанию	Описание
referencePoint *	—	Тип: Object Описание опорной точки.

* Обязательный параметр/опция.

update

```
{ } update(viaPointJson)
```

Обновляет модель новыми данными.

Параметры:

Параметр	Значение по умолчанию	Описание
viaPointJson *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

multiRouter.WayPoint

Примечание: Конструктор класса `multiRouter.WayPoint` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Отображение путевой точки. Путевые точки подразумевают остановку и делят маршрут на т.н. пути.

[Поля](#) | [События](#) | [Методы](#)

Создает отображение путевой точки.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
model	multiRouter.WayPointModel	Модель данных путевой точки.

Имя	Тип	Описание
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	data.Manager	Данные транзитной точки мультимаршрута. Доступны следующие поля: <ul style="list-style-type: none"> index: Integer – порядковый номер точки; request: String - строка с координатами или адресом, которой задается опорная точка. См. IMultiRouteReferencePoint. address: String - почтовый адрес точки; description: String - описание точки; name: String - название точки; geocoderMetaData: Object-метаданные Геокодера.
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .

Имя	Описание
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство touches, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> clientX - координата касания X относительно видимой области браузера; clientY - координата касания Y относительно видимой области браузера; pageX - координата касания X касания относительно начала документа; pageY - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>
propertieschange	<p>Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. <p>Унаследовано от IGeoObject.</p>
update	<p>Обновление отображения путевой точки. Экземпляр класса Event.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей**model**

```
{multiRouter.WayPointModel} model
```

Модель данных путевой точки.

properties

```
{data.Manager} properties
```

Данные транзитной точки мультимаршрута. Доступны следующие поля:

- index: Integer – порядковый номер точки;
- request: String - строка с координатами или адресом, которой задается опорная точка. См. [IMultiRouteReferencePoint](#).
- address: String - почтовый адрес точки;
- description: String - описание точки;

- name: String - название точки;
- geocoderMetaData: Object-метаданные Геокодера.

Описание событий

update

Обновление отображения путевой точки. Экземпляр класса [Event](#).

multiRouter.WayPointModel

Примечание: Конструктор класса multiRouter.WayPointModel скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventManager](#).

Модель данных путевой точки. Путевые точки подразумевают остановку и делят маршрут на т.н. пути.

[Поля](#) | [События](#) | [Методы](#)

Создает модель данных транзитной точки мультимаршрута.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
geometry	geometry.base.Point	Геометрия транзитной точки мультимаршрута.
multiRoute	multiRouter.MultiRouteModel	Ссылка на родительскую модель маршрута.
properties	data.Manager	Данные транзитной точки мультимаршрута. Доступны следующие поля: <ul style="list-style-type: none"> • index: Integer – порядковый номер точки; • request: String - строка с координатами или адресом, которой задается опорная точка. См. IMultiRouteReferencePoint. • address: String - почтовый адрес точки; • description: String - описание точки; • name: String - название точки; • geocoderMetaData: Object-метаданные Геокодера.

События

Имя	Описание
referencepointchange	Изменена опорная точка. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • oldReferencePoint: Object - описание предыдущей опорной точки.
update	Обновление модели новыми данными. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
<code>destroy()</code>		Уничтожает модель.
<code>getReferencePoint()</code>	Object	Возвращает соответствующую опорную точку.
<code>getReferencePointIndex()</code>	Integer	Возвращает индекс соответствующей опорной точки во множестве опорных точек родительского мультимаршрута.
<code>setReferencePoint(referencePoint)</code>		<p>Задаёт описание соответствующей опорной точки. Опорная точка может быть задана одним из следующих способов:</p> <ul style="list-style-type: none"> • строка, содержащая почтовый адрес опорной точки; • массив, содержащий широту и долготу опорной точки; • геометрия <code>geometry.Point</code>, описывающая опорную точку.
<code>update(wayPointJson)</code>		Обновляет модель новыми данными.

Описание полей**geometry**

```
{geometry.base.Point} geometry
```

Геометрия транзитной точки мультимаршрута.

multiRoute

```
{multiRouter.MultiRouteModel} multiRoute
```

Ссылка на родительскую модель маршрута.

properties

```
{data.Manager} properties
```

Данные транзитной точки мультимаршрута. Доступны следующие поля:

- `index`: `Integer` – порядковый номер точки;
- `request`: `String` - строка с координатами или адресом, которой задается опорная точка. См. `IMultiRouteReferencePoint`.
- `address`: `String` - почтовый адрес точки;
- `description`: `String` - описание точки;
- `name`: `String` - название точки;
- `geocoderMetaData`: `Object`-метаданные Геокодера.

Описание событий

referencepointchange

Изменена опорная точка. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldReferencePoint`: `Object` - описание предыдущей опорной точки.

update

Обновление модели новыми данными. Экземпляр класса [Event](#).

Описание методов

destroy

```
{ } destroy()
```

Уничтожает модель.

getReferencePoint

```
{Object} getReferencePoint()
```

Возвращает соответствующую опорную точку.

getReferencePointIndex

```
{Integer} getReferencePointIndex()
```

Возвращает индекс соответствующей опорной точки во множестве опорных точек родительского мультимаршрута.

setReferencePoint

```
{ } setReferencePoint(referencePoint)
```

Задаёт описание соответствующей опорной точки. Опорная точка может быть задана одним из следующих способов:

- строка, содержащая почтовый адрес опорной точки;
- массив, содержащий широту и долготу опорной точки;
- геометрия `geometry.Point`, описывающая опорную точку.

Параметры:

Параметр	Значение по умолчанию	Описание
referencePoint *	—	Тип: <code>Object</code> Описание опорной точки.

* Обязательный параметр/опция.

update

```
{ } update(wayPointJson)
```

Обновляет модель новыми данными.

Параметры:

Параметр	Значение по умолчанию	Описание
wayPointJson *	—	Тип: Object JSON-данные.

* Обязательный параметр/опция.

ObjectManager

Расширяет [ICustomizable](#), [IEventEmitter](#), [IGeoObject](#), [IParentOnMap](#).

Менеджер объектов. Позволяет оптимально отображать, кластеризовать и управлять видимостью объектов. Обратите внимание, что у объектов, отрисованных на карте через данный менеджер, нельзя включать режимы редактирования и перетаскивания.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
ObjectManager([options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции.</p> <ul style="list-style-type: none"> Можно задавать все опции, указанные в описании Clusterer, за исключением опций hasBalloon и hasHint. Опции для кластеров задаются с префиксом cluster. Список опций указан в описании класса ClusterPlacemark; Опции для одиночных объектов задаются с префиксом geoObject. Список опций определен в классе GeoObject. Обратите внимание, менеджер не учитывает опцию 'visible'.
options.clusterize	false	<p>Тип: Boolean</p> <p>Флаг, показывающий, нужно ли кластеризовать объекты. Обратите внимание, что на данный момент кластеризация работает только для точечных объектов. При включенном режиме кластеризации все неточечные объекты будут игнорироваться.</p>
options.syncOverlayInit	false	<p>Тип: Boolean</p> <p>Флаг, разрешающий создавать оверлеи для объектов синхронно. Обратите внимание, что при синхронном создании оверлея нужно самостоятельно обеспечить загрузку нужного класса, реализующего интерфейс IOverlay. По умолчанию оверлеи создаются асинхронно, при этом класс оверлея загружается по требованию.</p>
options.viewportMargin	128	<p>Тип: Number Number[]</p> <p>Отступ для области, в которой показываются объекты. С помощью данной опции область показа объектов расширяется по отношению к видимой области карты.</p>

Пример:

```
var objectManager = new ymaps.ObjectManager({
  // Включаем кластеризацию.
  clusterize: true,
  // Опции кластеров задаются с префиксом 'cluster'.
  clusterHasBalloon: false,
  // Опции геообъектов задаются с префиксом 'geoObject'.
  geoObjectOpenBalloonOnClick: false
});

// Опции можно задавать напрямую в дочерние коллекции.
objectManager.clusters.options.set({
  preset: 'islands#grayClusterIcons',
  hintContentLayout: ymaps.templateLayoutFactory.createClass('Группа объектов')
});
objectManager.objects.options.set('preset', 'islands#grayIcon');
```

Поля

Имя	Тип	Описание
clusters	objectManager.ClusterCollection	Коллекция кластеров, сгенерированных менеджером.
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
objects	objectManager.ObjectCollection	Коллекция объектов, добавленных в слой.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
add(objects)	ObjectManager	Добавление объектов в менеджер.
getBounds()	Number[][] null	Вычисляет границы области в геокоординатах, охватывающей все объекты в составе менеджера.
getFilter()	String Function null	Возвращает заданную функцию-фильтр.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getObjectState(id)	Object	Получение информации о текущем состоянии объекта, добавленного в менеджер.
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .

Имя	Возвращает	Описание
getPixelBounds()	Number[][] null	Вычисляет границы области в глобальных пиксельных координатах, охватывающей все объекты в составе менеджера.
remove(objects)	ObjectManager	Удаление объектов из менеджера.
removeAll()	ObjectManager	Удаление всех объектов из менеджера.
setFilter(filterFunction)		Устанавливает функцию-фильтр для объектов.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

clusters

```
{objectManager.ClusterCollection} clusters
```

Коллекция кластеров, сгенерированных менеджером.

Пример:

```
objectManager.objects.events.add('click', function (e) {
  var objectId = e.get('objectId');
  objectManager.objects.balloon.open(objectId);
});
```

objects

```
{objectManager.ObjectCollection} objects
```

Коллекция объектов, добавленных в слой.

Пример:

```
objectManager.objects.events.add('click', function (e) {
  objectManager.objects.balloon.open(objectId);});
```

Описание методов

add

```
{ObjectManager} add(objects)
```

Добавление объектов в менеджер.

Возвращает ссылку на менеджер объектов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>objects *</code>	—	<p>Тип: <code>Object Object[] String</code></p> <p>Строка или объект с JSON-описанием объектов. JSON-описание объектов строится по следующему принципу (см. пример ниже). Объектом может быть сущность или коллекция сущностей. Коллекция сущностей представляет из себя объект с полями:</p> <ul style="list-style-type: none"> <code>type</code> — тип объекта. Значение поля должно быть равно <code>"FeatureCollection"</code>; <code>features</code> - массив дочерних сущностей коллекции. Дочерние объекты могут быть сущностями или вложенными коллекциями сущностей. <p>Сущность представляет собой объект с полями:</p> <ul style="list-style-type: none"> <code>id</code> – уникальный идентификатор объекта. Обязательное поле. <code>type</code> — тип объекта. Значение поля должно быть равно <code>"Feature"</code>; <code>geometry</code> — геометрия объекта. Содержит поля <code>type</code> и <code>coordinates</code>. Значение соответствует, передаваемому в конструктор GeoObject; <code>options</code> - опции геообъекта; <code>properties</code> - данные геообъекта.

* Обязательный параметр/опция.

Примеры:

1.

```
var objectManager = new ymaps.ObjectManager({ clusterize: true }),
    currentId = 0;

// Добавим единичный объект.
objectManager.add({
  type: 'Feature',
  id: currentId++,
  geometry: {
    type: 'Point',
    coordinates: [56.23, 34.79]
  },
  properties: {
    hintContent: 'Содержание всплывающей подсказки',
    balloonContent: 'Содержание балуна'
  }
});
map.geoObjects.add(objectManager);
```

2.

```
// Добавим массив точечных объектов.
var objects = [];
for (var i = 0, l = coordinates.length; i < l; i++) {
  objects.push({
    type: 'Feature',
    id: currentId++,
    geometry: {
      type: 'Point',
      coordinates: coordinates[i]
    }
  });
}
objectManager.add(objects);
map.geoObjects.add(objectManager);
```

3.

```
// Добавление коллекций объектов.
var collection = {
  type: 'FeatureCollection',
  features: [{
    type: 'Feature',
    id: currentId++,
    geometry: {
      type: 'Point',
      coordinates: [24.34, 65.24]
    }
  }, {
    type: 'Feature',
    id: currentId++,
    geometry: {
      type: 'Point',
      coordinates: [25.34, 63.24]
    }
  }
]
};
objectManager.add(collection);
map.geoObjects.add(objectManager);
```

4.

```
// Добавление не точечных объектов
var objects = [];

// Добавление круга
objects.push({
  type: 'Feature',
  id: 1,
  geometry: {
    type: 'Circle',
    coordinates: [55.755381, 37.619044],
    radius: 300
  }
});

// Добавление прямоугольника
objects.push({
  type: 'Feature',
  id: 2,
  geometry: {
    type: 'Rectangle',
```

```

        coordinates: [
            [55.764286, 37.606169],
            [55.759688, 37.620588]
        ],
    },
    options: {
        fillColor: '#FFFFFF',
        opacity: 0.8
    }
});

// Добавление полилинии
objects.push({
    type: 'Feature',
    id: 3,
    geometry: {
        type: 'LineString',
        coordinates: [
            [55.75901100, 37.6308886],
            [55.7516538, 37.6299444],
            [55.74603822, 37.6380125]
        ]
    },
    options: {
        strokeColor: "#FF0000",
        strokeWidth: 5
    }
});

// Добавление многоугольника
objects.push({
    type: 'Feature',
    id: 4,
    geometry: {
        type: 'Polygon',
        coordinates: [[
            [55.75175065, 37.6041094],
            [55.75005637, 37.6137224],
            [55.742891186, 37.6166407],
            [55.74153546, 37.60342281],
            [55.74700649, 37.59775798]
        ]]
    },
    options: {
        opacity: 0.2,
        strokeWidth: 2,
        fillColor: '#00FF00'
    }
});

objectManager.add(objects);

```

getBounds

```
{Number[][]|null} getBounds()
```

Вычисляет границы области в геокоординатах, охватывающей все объекты в составе менеджера.

Возвращает массив координат области или null, если менеджер не добавлен на карту.

getFilter

```
{String|Function|null} getFilter()
```

Возвращает заданную функцию-фильтр.

getObjectState

```
{Object} getObjectState(id)
```

Получение информации о текущем состоянии объекта, добавленного в менеджер.

Возвращает объект, содержащий следующие поля:

- **found** - признак, указывающий, существует ли объект с переданным идентификатором. Type: Boolean.
- **isShown** - признак, указывающий, находится ли объект в видимой области карты. Type: Boolean.
- **cluster** - JSON-описание кластера, в который добавлен объект. Помимо обязательных полей, содержит поле `properties.geoObjects` с массивом объектов, попавших в кластер. Поле возвращается только при включенной кластеризации;

- `isClustered` – признак, указывающий, попал ли объект в состав кластера. Поле возвращается только при включенной кластеризации; Type: Boolean.
- `isFilteredOut` - признак, указывающий, прошел ли объект фильтрацию. Если фильтр не задан или объект прошел фильтрацию, значение поля будет `false`. Type: Boolean.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>id</code> *	—	Тип: Object Идентификатор объекта, для которого нужно получить состояние.

* Обязательный параметр/опция.

Пример:

```
// Открытие балуна кластера с выбранным объектом.
// Получим данные о состоянии объекта внутри кластера.
var objectState = objectManager.getObjectState(objects[1].id);
// Проверяем, находится ли объект в видимой области карты.
if (objectState.found && objectState.isShown) {
    // Если объект попадает в кластер, открываем балун кластера с нужным выбранным объектом.
    if (objectState.isClustered) {
        objectManager.clusters.state.set('activeObject', objects[1]);
        objectManager.clusters.balloon.open(objectState.cluster.id);
    } else {
        // Если объект не попал в кластер, открываем его собственный балун.
        objectManager.objects.balloon.open(objects[1].id);
    }
}
```

getPixelBounds

```
{Number[][]|null} getPixelBounds()
```

Вычисляет границы области в глобальных пиксельных координатах, охватывающей все объекты в составе менеджера.

Возвращает массив координат области или `null`, если менеджер не добавлен на карту.

remove

```
{ObjectManager} remove(objects)
```

Удаление объектов из менеджера.

Возвращает ссылку на менеджер объектов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>objects</code> *	—	Тип: Object Object[] String Строка, объект с JSON-описанием объектов или массив идентификаторов удаляемых объектов. Формат описания объектов см. в описании метода ObjectManager.add

* Обязательный параметр/опция.

Примеры:**1.**

```
// Удаление объектов с идентификаторами 0 и 56.  
objectManager.remove([0, 56]);
```

2.

```
// Удаление массива объектов  
objectManager.remove([  
  {  
    type: 'Feature',  
    id: 23,  
    geometry: {  
      type: 'Point'  
      coordinates: [45.33, 24.33]  
    }  
  },  
  {  
    type: 'Feature',  
    id: 52,  
    geometry: {  
      type: 'Point'  
      coordinates: [45.23, 24.34]  
    }  
  }  
]);
```

3.

```
// Удалим коллекцию объектов.  
objectManager.remove({  
  type: 'FeatureCollection',  
  features: [  
    // Описание объектов в составе коллекции.  
    // ...  
  ]  
});
```

removeAll

```
{ObjectManager} removeAll()
```

Удаление всех объектов из менеджера.

Возвращает ссылку на себя.

Пример:

```
objectManager.removeAll();
```

setFilter

```
{ } setFilter(filterFunction)
```

Устанавливает функцию-фильтр для объектов.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>filterFunction</code> *	—	<p>Тип: Function String</p> <p>функция-фильтр. Получает на вход единичный объект, добавленный в ObjectManager. Если функция возвращает true, объект будет обработан. Если false - объект будет исключен из дальнейшей обработки. Также в качестве фильтра можно передавать строку. В строке-фильтре доступны следующие ключевые слова:</p> <ul style="list-style-type: none"> options – обращение к опциям объекта; properties – обращение к данным объекта; geometry – обращение к геометрии объекта; id – обращение к идентификатору объекта. <p>В качестве фильтра можно указывать выражение, возвращающее true или false.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
// Пропустить объекты, id которых меньше 100.
objectManager.setFilter('id > 100');
```

2.

```
// На карте будут отображаться только объекты с заданными типами.
objectManager.setFilter('properties.type == "кафе" || properties.type == "аптека");
```

3.

```
// Можно задавать функцию-фильтр.
objectManager.setFilter(function (object) {
    return object.properties.name != 'Тот, кого нельзя показать.';});
```

objectManager

objectManager.addon

objectManager.addon.clustersBalloon

Примечание: Конструктор класса `objectManager.addon.clustersBalloon` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Методы

Пример:

```
<script src="http://api-maps.yandex.ru/2.1/?  
lang=ru_RU&load=Map,ObjectManager,objectManager.addon.clustersBalloon" type="text/javascript"></script>
```

Методы

Имя	Возвращает	Описание
<code>get()</code>	<code>IPopupManager</code>	Возвращает менеджер балуна менеджера объектов.

Описание методов

get

```
{IPopupManager} get()
```

Возвращает менеджер балуна менеджера объектов.

Пример:

```
ymaps.objectManager.addon.clustersBalloon.get(мойMap)
```

objectManager.addon.clustersHint

Примечание: Конструктор класса `objectManager.addon.clustersHint` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Методы

Пример:

```
<script src="http://api-maps.yandex.ru/2.1/?  
lang=ru_RU&load=Map,ObjectManager,objectManager.addon.clustersHint" type="text/javascript"></script>
```

Методы

Имя	Возвращает	Описание
<code>get()</code>	<code>IPopupManager</code>	Возвращает менеджер хинта <code>objectManager</code> .

Описание методов

get

```
{IPopupManager} get()
```

Возвращает менеджер хинта [ObjectManager](#).

Пример:

```
ymaps.objectManager.addon.clustersHint.get(myMap);
```

objectManager.addon.objectsBalloon

Примечание: Конструктор класса `objectManager.addon.objectsBalloon` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Методы

Пример:

```
<script src="http://api-maps.yandex.ru/2.1/?  
lang=ru_RU&load=Map,ObjectManager,objectManager.addon.objectsBalloon" type="text/javascript"></script>
```

Методы

Имя	Возвращает	Описание
get(collection)	IPopupManager	Возвращает менеджер балуна менеджера объектов.

Описание методов

get

```
{IPopupManager} get(collection)
```

Возвращает менеджер балуна менеджера объектов.

Параметры:

Параметр	Значение по умолчанию	Описание
collection *	—	Тип: objectManager.ObjectCollection Коллекция

* Обязательный параметр/опция.

Пример:

```
ymaps.objectManager.addon.objectsBalloon.get(myMap)
```

objectManager.addon.objectsHint

Примечание: Конструктор класса `objectManager.addon.objectsHint` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Статический объект.

Методы

Пример:

```
<script src="http://api-maps.yandex.ru/2.1/?  
lang=ru_RU&load=Map,ObjectManager,objectManager.addon.objectsHint" type="text/javascript"></script>
```

Методы

Имя	Возвращает	Описание
get()	IPopupManager	Возвращает менеджер хинта ObjectManager .

Описание методов**get**

```
{IPopupManager} get()
```

Возвращает менеджер хинта [ObjectManager](#).

Пример:

```
ymaps.objectManager.addon.objectsHint.get(myMap)
```

objectManager.Balloon

Расширяет [IBalloonManager](#).

Менеджер балуна коллекций ObjectManager. Позволяет управлять балуном объекта, открывать и скрывать его. Использует внутри себя менеджер балуна карты [map.Balloon](#). Коллекции объектов в составе [ObjectManager](#) содержат экземпляр этого класса, доступный как `myObjectManager.objects.balloon` и `myObjectsLayer.clusters.balloon`. Не создавайте экземпляры этого класса без необходимости.

См.: [Balloon](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
objectManager.Balloon(collection)
```

Параметры:

Параметр	Значение по умолчанию	Описание
collection *	—	Тип: IReadOnlyCollection Коллекция объектов.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
autopanbegin	Начало автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
autopanend	Окончание автоматического перемещения центра карты, вызванного методом <code>autoPan</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
beforeuserclose	Событие, предшествующее событию Balloon.userclose . Позволяет отменить пользовательское действие посредством вызова метода <code>preventDefault</code> . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .
close	Заккрытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. Унаследовано от IPopupManager .
open	Открытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. Унаследовано от IPopupManager .
userclose	Заккрытие балуна пользователем. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект IBalloonOwner. Унаследовано от IBalloonManager .

Методы

Имя	Возвращает	Описание
autoPan()	vow.Promise	Перемещает карту таким образом, чтобы балун оказался видимым. Унаследован от IBalloonManager .
close([force])	vow.Promise	Закрывает открытый балун.
destroy()		Отключает менеджер инфо-объекта. Унаследован от IPopupManager .

Имя	Возвращает	Описание
getData()	Object null	Возвращает хэш с описанием объекта, на котором открыт балун или null, если балун не был открыт.
getOptions()	IOptionsManager null	Возвращает менеджер опций или null. Унаследован от IPopupManager .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopupManager .
getOverlaySync()	IOverlay null	Возвращает оверлей, если тот существует. Унаследован от IPopupManager .
getPosition()	Number[] null	Возвращает координаты инфо-объекта или 'null'. Унаследован от IPopupManager .
isOpen(id)	Boolean	Метод, определяющий, открыт ли балун на объекте с переданным идентификатором.
open(objectId, anchorPixelPosition)	vow.Promise	Открывает балун на объекте с переданным идентификатором.
setData(objectData)	vow.Promise	Задаёт новые данные для отображения балуна.
setOptions(options)	vow.Promise	Задаёт инфо-объекту новые опции. Унаследован от IPopupManager .
setPosition(position)	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopupManager .

Описание методов

close

```
{vow.Promise} close([force])
```

Закрывает открытый балун.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>force</code>	<code>false</code>	Тип: Boolean Мгновенное закрытие.

Пример:

```
// Закроем все балуны на слое.  
objectManager.objects.balloon.close();  
objectManager.clusters.balloon.close();
```

getData

```
{Object|null} getData()
```

Возвращает хэш с описанием объекта, на котором открыт балун или null, если балун не был открыт.

Пример:

```
var cluster = objectManager.clusters.balloon.getData();  
if (cluster) {  
    alert('Сейчас открыт балун кластера с ' + cluster.properties.geoObjects.length + ' объектами.');
```

isOpen

```
{Boolean} isOpen(id)
```

Метод, определяющий, открыт ли балун на объекте с переданным идентификатором.

Возвращает состояние балуна: открыт/закрыт.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>id *</code>	—	Тип: Object Идентификатор объекта.

* Обязательный параметр/опция.

Пример:

```
// При повторном клике на метку будем закрывать балун.  
objectManager.objects.options.set('hideIconOnBalloonOpen', false);  
objectManager.objects.events.add('click', function (e) {  
    var objectId = e.get('objectId');  
    if (objectManager.objects.balloon.isOpen(objectId)) {  
        objectManager.objects.balloon.close();  
    }  
});
```

open

```
{vow.Promise} open(objectId, anchorPixelPosition)
```

Открывает балун на объекте с переданным идентификатором.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>objectId *</code>	—	Тип: Object Идентификатор объекта, на котором надо открыть балун.
<code>anchorPixelPosition *</code>	—	Тип:

* Обязательный параметр/опция.

Пример:

```
// Загрузка данных объекта перед открытием балуна.
// Отменим автоматическое открытие балуна по клику на объекте.
objectManager.options.set('geoObjectOpenBalloonOnClick', false);
objectManager.objects.events.add('click', function (e) {
  var objectId = e.get('objectId');
  // По клику на объекте вызовем подгрузку данных для объекта.
  loadObjectData(objectId).then(function (data) {
    // Как только данные будут готовы, зададим их объекту и откроем балун.
    objectManager.objects.getById(objectId).properties = data;
    objectManager.objects.balloon.open(objectId);
  });
});
```

setData

```
{vow.Promise} setData(objectData)
```

Задаёт новые данные для отображения балуна.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>objectData *</code>	—	Тип: Object Хэш с описанием объекта, на котором открыт балун. Соответствует описанию объекта, который подается на вход ObjectManager.add.

* Обязательный параметр/опция.

Пример:

```
// Подгрузка данных после открытия балуна на объекте.
objectManager.objects.events.add('balloonopen', function (e) {
  var objectId = e.get('objectId');
  object = objectManager.objects.getById(objectId);
  if (!object.properties) {
    loadObjectData(objectId).then(function (data) {
      if (objectManager.balloon.isOpen(objectid)) {
        object.properties = data;
        objectManager.objects.balloon.setData(object);
      }
    });
  }
});
```

objectManager.ClusterCollection

Расширяет [ICustomizable](#), [IEventEmitter](#).

Коллекция кластеров, сгенерированных [ObjectManager](#). Кластеры добавляются и удаляются из коллекции автоматически в ходе работы и доступны только на чтение. Объект-кластер представляет из себя JSON-структуру, аналогичную объектам в составе слоя. Поля объекта-кластера:

- `id` - уникальный идентификатор кластера;
- `geometry` - описание геометрии кластера;
- `properties` - описание данных кластера. В поле `properties.geoObjects` хранится массив объектов, входящих в состав кластера;
- `options` - опции кластера. Необязательное поле;

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
objectManager.ClusterCollection()
```

Поля

Имя	Тип	Описание
balloon	objectManager.Balloon	Балун кластера в составе менеджера.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
hint	objectManager.Hint	Хинт объекта в составе ObjectManager . Названия полей доступны через метод Event.get : <ul style="list-style-type: none">• <code>objectId</code> – идентификатор объекта, над которым был показан хинт (всплывающая подсказка).

Имя	Тип	Описание
options	option.Manager	<p>Менеджер опций. Имена полей, доступных через метод option.Manager.get:</p> <ul style="list-style-type: none"> • hasBalloon - флаг наличия у коллекции поля .balloon. Если при клике на кластер не нужно открывать балун, рекомендуется установить эту опцию в значение <code>false</code>, чтобы избежать лишних инициализаций; • hasHint - флаг наличия у коллекции поля .hint. Если при наведении на кластер не нужно показывать всплывающую подсказку, рекомендуется установить эту опцию в значение <code>false</code>, чтобы избежать лишних инициализаций; • hideIconOnBalloonOpen - скрывать иконку при открытии балуна. Значение по умолчанию <code>true</code>. • openBalloonOnClick - опция, позволяющая запретить открытие балуна при клике на кластер. По умолчанию открытие балуна разрешено; • openHintOnHover - опция, позволяющая запретить показ всплывающей подсказки при наведении на кластер. По умолчанию показ хинтов разрешен.
overlays	objectManager.ClusterCollection	<p>Коллекция оверлеев кластеров. Все события, за исключением событий <code>add</code> и <code>remove</code>, пропагируются от коллекции оверлеев в коллекцию кластеров.</p>
state	data.Manager	<p>Состояние коллекции кластеров. Определяется следующими полями:</p> <ul style="list-style-type: none"> • activeObject – JSON-описание объекта, выбранного в балуне кластера.

События

Имя	Описание
add	<p>Добавление кластера в коллекцию. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • objectId – идентификатор добавленного объекта; • child - добавленный объект.
clusteroptionschange	<p>Изменение опций кластера через метод objectManager.ClusterCollection.setClusterOptions. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • objectId - идентификатор кластера, у которого заменили опции.

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .
remove	Удаление кластера из коллекции. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>objectId</code> – идентификатор удаленного объекта; <code>child</code> – удаленный объект.

Методы

Имя	Возвращает	Описание
each(callback, context)		
getAll()	<code>Object[]</code>	Возвращает массив объектов, содержащихся в коллекции.
getById(id)	<code>Object null</code>	Возвращает объект-кластер с указанным идентификатором или <code>null</code> , если такого кластера не существует.
getIterator()	Iterator	Возвращает итератор по коллекции.
getLength()	<code>Number</code>	Возвращает количество объектов в коллекции.
getObjectManager()	ObjectManager	Возвращает родительский слой объектов коллекции.
setClusterOptions(objectId, options)	objectManager.ObjectCollection	Возвращает ссылку на себя.

Описание полей

balloon

```
{objectManager.Balloon} balloon
```

Балун кластера в составе менеджера.

hint

```
{objectManager.Hint} hint
```

Хинт объекта в составе [ObjectManager](#). Названия полей доступны через метод [Event.get](#):

- `objectId` – идентификатор объекта, над которым был показан хинт (всплывающая подсказка).

options

```
{option.Manager} options
```

Менеджер опций. Имена полей, доступных через метод `option.Manager.get`:

- `hasBalloon` - флаг наличия у коллекции поля `.balloon`. Если при клике на кластер не нужно открывать балун, рекомендуется установить эту опцию в значение `false`, чтобы избежать лишних инициализаций;
- `hasHint` - флаг наличия у коллекции поля `.hint`. Если при наведении на кластер не нужно показывать всплывающую подсказку, рекомендуется установить эту опцию в значение `false`, чтобы избежать лишних инициализаций;
- `hideIconOnBalloonOpen` - скрывать иконку при открытии балуна. Значение по умолчанию `true`.
- `openBalloonOnClick` - опция, позволяющая запретить открытие балуна при клике на кластер. По умолчанию открытие балуна разрешено;
- `openHintOnHover` - опция, позволяющая запретить показ всплывающей подсказки при наведении на кластер. По умолчанию показ хинтов разрешен.

Пример:

```
objectManager.objects.options.set({
  preset: 'islands#greenDotIcon',
  hintContentLayout: ymaps.templateLayoutFactory.createClass('{{properties.name}}')
});
```

overlays

```
{objectManager.OverlayCollection} overlays
```

Коллекция оверлеев кластеров. Все события, за исключением событий `add` и `remove`, пропагируются от коллекции оверлеев в коллекцию кластеров.

Пример:

```
// Изменение цвета значка кластера при наведении курсора мыши.
objectManager.clusters.events.add(['mouseenter', 'mouseleave'], function (e) {
  var objectId = e.get('objectId');
  var overlay = objectManager.clusters.overlays.getById(objectId);
  if (e.get('type') == 'mouseenter') {
    setRedColor(objectId);
    overlay.events.add('mapchange', onMapChange);
  } else {
    setGreenColor(objectId);
    overlay.events.remove('mapchange', onMapChange);
  }
});

function onMapChange (e) {
  setGreenColor(objectManager.clusters.overlays.getId(e.get('target')));
}

function setGreenColor (objectId) {
  objectManager.clusters.setClusterOptions(objectId, {
    preset: 'islands#greenClusterIcons'
  });
}

function setRedColor (objectId) {
  objectManager.clusters.setClusterOptions(objectId, {
    preset: 'islands#redClusterIcons'
  });
}
```

state

```
{data.Manager} state
```

Состояние коллекции кластеров. Определяется следующими полями:

- `activeObject` – JSON-описание объекта, выбранного в балуне кластера.

Пример:

```
// Откроем балун кластера с нужным выбранным объектом.
var objectState = objectManager.getObjectState(myObjects[i]);
if (objectState.isClustered) {
  objectManager.clusters.state.set('activeObject', myObjects[i]);
  objectManager.clusters.balloon.open(objectState.cluster.id)
}
```

Описание событий

add

Добавление кластера в коллекцию. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` – идентификатор добавленного объекта;
- `child` - добавленный объект.

clusteroptionschange

Изменение опций кластера через метод [objectManager.ClusterCollection.setClusterOptions](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор кластера, у которого заменили опции.

remove

Удаление кластера из коллекции. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` – идентификатор удаленного объекта;
- `child` - удаленный объект.

Описание методов

each

```
{ } each(callback, context)
```

Параметры:

Параметр	Значение по умолчанию	Описание
callback *	—	Тип: Function Функция-обработчик, в которую передаются объекты коллекции.
context *	—	Тип: Object Контекст исполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

```
var clusterizedObjectsCounter = 0;
objectManager.clusters.each(function (cluster) {
    clusterizedObjectsCounter += cluster.properties.geoObjects.length;
});
alert('На карте показано ' + clusterizedObjectsCounter + ' кластеризованных объектов.');
```

getAll

```
{Object[] } getAll()
```

Возвращает массив объектов, содержащихся в коллекции.

Пример:

```
var clusterArray = objectManager.clusters.getAll();
```

getById

```
{Object|null} getById(id)
```

Возвращает объект-кластер с указанным идентификатором или null, если такого кластера не существует.

Параметры:

Параметр	Значение по умолчанию	Описание
id *	—	Тип: String Идентификатор кластера.

* Обязательный параметр/опция.

Пример:

```
// Будем менять цвет кластера, если в нем более 20 объектов.  
objectManager.clusters.events.add('add', function (e) {  
    var cluster = objectManager.clusters.getById(e.get('objectId')),  
        objects = cluster.properties.geoObjects;  
    if (objects.length > 20) {  
        objectManager.clusters.setClusterOptions(cluster.id, {  
            preset: 'islands#redClusterIcons'  
        });  
    }  
});
```

getIterator

```
{Iterator} getIterator()
```

Возвращает итератор по коллекции.

Пример:

```
var clusterizedObjectsCounter = 0;  
var it = objectManager.clusters.getIterator();  
var cluster;  
while ((cluster = it.getNext()) != it.STOP_ITERATION)  
    clusterizedObjectsCounter += cluster.properties.geoObjects.length;  
});  
alert('На карте показано ' + clusterizedObjectsCounter + ' кластеризованных объектов.');
```

getLength

```
{Number} getLength()
```

Возвращает количество объектов в коллекции.

Пример:

```
alert('На карте показано ' + objectManager.clusters.getLength() + ' кластеров.');
```

getManager

```
{ObjectManager} getManager()
```

Возвращает родительский слой объектов коллекции.

setClusterOptions

```
{objectManager.ObjectCollection} setClusterOptions(objectId, options)
```

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>objectId</code> *	—	Тип: String Идентификатор кластера.
<code>options</code> *	—	Тип: Object Объект с опциями кластера.

* Обязательный параметр/опция.

Пример:

```
// Будем менять цвет кластера, если в нем более 20 объектов.
objectManager.clusters.events.add('add', function (e) {
  var cluster = objectManager.clusters.getById(e.get('objectId')),
      objects = cluster.properties.geoObjects;
  if (objects.length > 20) {
    objectManager.clusters.setClusterOptions(cluster.id, {
      preset: 'islands#redClusterIcons'
    });
  }
});
```

objectManager.Hint

Расширяет [IHintManager](#).

Менеджер хинта объектного слоя. Позволяет управлять хинтом объектного слоя, открывать и скрывать его. Использует внутри себя менеджер хинтов карты [map.Hint](#). Объектные слои содержат экземпляры этого класса, доступные как `myObjectManager.objects.hint` и `myObjectManager.clusters.hint`. Не создавайте экземпляры этого класса без необходимости.

См.: [Hint](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
objectManager.Hint(collection)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>collection</code> *	—	Тип: <code>IReadOnlyCollection</code> Объектный слой.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
close	Заккрытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был закрыт инфо-объект. Унаследовано от IPopupManager .
open	Открытие инфо-объекта. Названия полей доступны через метод Event.get : <ul style="list-style-type: none"> <code>target</code> - ссылка на объект, на котором был открыт инфо-объект. Унаследовано от IPopupManager .

Методы

Имя	Возвращает	Описание
close([force])	vow.Promise	Скрывает всплывающую подсказку.
destroy()		Отключает менеджер инфо-объекта. Унаследован от IPopupManager .
getData()	<code>Object null</code>	Возвращает хэш с описанием объекта, на котором показан хинт или <code>null</code> , если хинт не был показан.
getOptions()	IOptionManager null	Возвращает менеджер опций или <code>null</code> . Унаследован от IPopupManager .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopupManager .
getOverlaySync()	IOverlay null	Возвращает оверлей, если тот существует. Унаследован от IPopupManager .
getPosition()	<code>Number[] null</code>	Возвращает координаты инфо-объекта или <code>`null`</code> . Унаследован от IPopupManager .

Имя	Возвращает	Описание
<code>isOpen(id)</code>	Boolean	Метод, определяющий, показана ли всплывающая подсказка на объекте с переданным идентификатором.
<code>open(objectId[, position])</code>	<code>vow.Promise</code>	Показывает всплывающую подсказку на объекте с переданным идентификатором.
<code>setData(objectData)</code>	<code>vow.Promise</code>	Задаёт новые данные для отображения всплывающей подсказки.
<code>setOptions(options)</code>	<code>vow.Promise</code>	Задаёт инфо-объекту новые опции. Унаследован от <code>IPopupManager</code> .
<code>setPosition(position)</code>	<code>vow.Promise</code>	Задаёт инфо-объекту новую позицию. Унаследован от <code>IPopupManager</code> .

Описание методов

`close`

```
{vow.Promise} close([force])
```

Скрывает всплывающую подсказку.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>force</code>	false	Тип: Boolean Мгновенное закрытие.

Пример:

```
// Закроем все всплывающие подсказки на слое.  
objectManager.objects.hint.close();  
objectManager.clusters.hint.close();
```

`getData`

```
{Object|null} getData()
```

Возвращает хэш с описанием объекта, на котором показан хинт или null, если хинт не был показан.

Пример:

```
var cluster = objectManager.clusters.hint.getData();  
if (cluster) {
```

```
alert('Кластер отображает подсказку.');
```

isOpen

```
{Boolean} isOpen(id)
```

Метод, определяющий, показана ли всплывающая подсказка на объекте с переданным идентификатором.

Возвращает состояние подсказки: показана/скрыта.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>id</code> *	—	Тип: Object Идентификатор объекта.

* Обязательный параметр/опция.

Пример:

```
// Скроем подсказку при клике на объекте.  
objectManager.objects.add('click', function (e) {  
  var objectId = e.get('objectId');  
  if (objectManager.objects.hint.isOpen(objectId)) {  
    objectManager.objects.hint.close();  
  }  
});
```

open

```
{vow.Promise} open(objectId[, position])
```

Показывает всплывающую подсказку на объекте с переданным идентификатором.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>objectId</code> *	—	Тип: Object Идентификатор объекта, на котором надо открыть подсказку.
<code>position</code>	—	Тип: Number[] Позиция, на которой надо показать всплывающую подсказку в глобальных пиксельных координатах. Если значение не задано, хинт покажется на геометрическом центре объекта.

* Обязательный параметр/опция.

Пример:

```
objectManager.clusters.hint.open(objectId);
```

setData

```
{vow.Promise} setData(objectData)
```

Задаёт новые данные для отображения всплывающей подсказки.

Возвращает объект-promise.

Параметры:

Параметр	Значение по умолчанию	Описание
objectData *	—	Тип: Object Хэш с описанием объекта, на котором открыт хинт. Соответствует описанию объекта, который подается на вход <code>ObjectManager.add</code> .

* Обязательный параметр/опция.

Пример:

```
objectManager.objects.hint.setData(objectManager.objects.getById(objectId));
```

objectManager.ObjectCollection

Расширяет [ICollection](#), [ICustomizable](#).

Коллекция объектов, добавленных в слой.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
objectManager.ObjectCollection()
```

Поля

Имя	Тип	Описание
balloon	objectManager.Balloon	Балун объекта в составе менеджера.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
hint	objectManager.Hint	Хинт объекта в составе ObjectManager .

Имя	Тип	Описание
options	option.Manager	<p>Менеджер опций. Имена полей, доступных через метод option.Manager.get:</p> <ul style="list-style-type: none"> • hasBalloon - флаг наличия у коллекции поля .balloon. Если при клике на объекте не нужно открывать балун, рекомендуется установить эту опцию в значение <code>false</code>, чтобы избежать лишних инициализаций; • hasHint - флаг наличия у коллекции поля .hint. Если при наведении на объект не нужно показывать всплывающую подсказку, рекомендуется установить эту опцию в значение <code>false</code>, чтобы избежать лишних инициализаций; • hideIconOnBalloonOpen - скрывать иконку при открытии балуна. Значение по умолчанию <code>true</code>. • openBalloonOnClick - показывать балун при клике на объект. Значение по умолчанию <code>true</code>.
overlays	objectManager.OverlayCollection	<p>Коллекция оверлеев одиночных объектов. Все события, за исключением событий <code>add</code> и <code>remove</code>, пропагируются от коллекции оверлеев в коллекцию объектов.</p>

События

Имя	Описание
add	<p>Добавление объекта в коллекцию. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>objectId</code> – идентификатор добавленного объекта; • <code>child</code> - добавленный объект.
objectoptionschange	<p>Изменение опций объекта через метод objectManager.ObjectCollection.setObjectOptions. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, у которого заменили опции.
optionschange	<p>Изменение в опциях объекта. Унаследовано от ICustomizable.</p>
remove	<p>Удаление объекта из коллекции. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>objectId</code> – идентификатор удаленного объекта; • <code>child</code> - удаленный объект.

Методы

Имя	Возвращает	Описание
add(data)	objectManager.ObjectCollection	Метод полностью дублирует логику ObjectManager.add .
each(callback, context)		Метод, вызывающий переданную функцию-обработчик для всех элементов коллекции.
getAll()	Object[]	Возвращает массив объектов, содержащихся в коллекции.
getById(id)	Object null	Возвращает объект или null, если объекта с переданным id не существует.
getIterator()	Iterator	Возвращает итератор по коллекции. Унаследован от ICollection .
getLength()	Number	Возвращает количество объектов в коллекции.
getObjectManager()	ObjectManager	Возвращает родительский слой объектов коллекции.
remove(data)	objectManager.ObjectCollection	Метод полностью дублирует логику ObjectManager.remove .
removeAll()	objectManager.ObjectCollection	Удаляет все элементы коллекции.
setObjectOptions(objectId, options)	objectManager.ObjectCollection	Метод, позволяющий динамически обновлять опции объекта. Метод следует использовать в случаях, когда вы хотите, чтобы новые опции мгновенно применились к отображению объекта на карте. Обратите внимание, менеджер не учитывает опцию 'visible'.

Описание полей

balloon

```
{objectManager.Balloon} balloon
```

Балун объекта в составе менеджера.

hint

```
{objectManager.Hint} hint
```

Хинт объекта в составе [ObjectManager](#).

options

```
{option.Manager} options
```

Менеджер опций. Имена полей, доступных через метод `option.Manager.get`:

- `hasBalloon` - флаг наличия у коллекции поля `.balloon`. Если при клике на объекте не нужно открывать балун, рекомендуется установить эту опцию в значение `false`, чтобы избежать лишних инициализаций;
- `hasHint` - флаг наличия у коллекции поля `.hint`. Если при наведении на объект не нужно показывать всплывающую подсказку, рекомендуется установить эту опцию в значение `false`, чтобы избежать лишних инициализаций;
- `hideIconOnBalloonOpen` - скрывать иконку при открытии балуна. Значение по умолчанию `true`.
- `openBalloonOnClick` - показывать балун при клике на объект. Значение по умолчанию `true`.

Пример:

```
// Зададим стили для одиночных объектов в составе коллекции.  
objectManager.objects.options.set({  
  preset: 'islands#redIcon',  
  hasBalloon: false,  
  zIndex: 500  
});
```

overlays

```
{objectManager.OverlayCollection} overlays
```

Коллекция оверлеев одиночных объектов. Все события, за исключением событий `add` и `remove`, пропагируются от коллекции оверлеев в коллекцию объектов.

Пример:

```
objectManager.objects.overlays.events.add('add', function (e) {  
  alert('Объект ' + e.get('objectId') + ' показан на карте.'));  
});
```

Описание событий

add

Добавление объекта в коллекцию. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` – идентификатор добавленного объекта;
- `child` - добавленный объект.

objectoptionschange

Изменение опций объекта через метод [objectManager.ObjectCollection.setObjectOptions](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, у которого заменили опции.

remove

Удаление объекта из коллекции. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` – идентификатор удаленного объекта;
- `child` - удаленный объект.

Описание методов

add

```
{objectManager.ObjectCollection} add(data)
```


Метод полностью дублирует логику [ObjectManager.add](#).

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
data *	—	Тип: Object Object[] String Объекты, добавляемые в слой.

* Обязательный параметр/опция.

Пример:

```
objectManager.objects.add({
  type: 'Feature',
  geometry: {
    type: 'Point',
    coordinates: [55.33, 36.64]
  }
});
```

each

```
{ } each(callback, context)
```

Метод, вызывающий переданную функцию-обработчик для всех элементов коллекции.

Параметры:

Параметр	Значение по умолчанию	Описание
callback *	—	Тип: Function Функция-обработчик, в которую передаются объекты коллекции.
context *	—	Тип: Object Контекст выполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

```
var singleCounter = 0,
    var clusterCounter = 0;
objectManager.objects.each(function (object) {
  var objectState = objectManager.getObjectState(object.id);
  if (objectState.isClustered) {
    clusterCounter++;
  } else {
    if (objectState.isShown) {
      singleCounter++;
    }
  }
});
alert('Количество одиночных меток на карте: ' + singleCounter);
alert('Количество кластеризированных меток: ' + clusterCounter);
```

getAll

```
{Object[] } getAll()
```

Возвращает массив объектов, содержащихся в коллекции.

getById

```
{Object|null} getById(id)
```

Возвращает объект или null, если объекта с переданным id не существует.

Параметры:

Параметр	Значение по умолчанию	Описание
id *	—	Тип: Number Идентификатор объекта.

* Обязательный параметр/опция.

Пример:

```
objectManager.objects.add('click', function (e) {  
    var objectId = e.get('objectId'),  
        object = objectManager.objects.getById(objectId);  
});
```

getLength

```
{Number} getLength()
```

Возвращает количество объектов в коллекции.

Пример:

```
alert('Количество объектов в слое: ' + objectManager.objects.getLength());
```

getManager

```
{ObjectManager} getManager()
```

Возвращает родительский слой объектов коллекции.

remove

```
{objectManager.ObjectCollection} remove(data)
```

Метод полностью дублирует логику [ObjectManager.remove](#).

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
data *	—	Тип: Object Object[] String Объекты, которые нужно удалить.

* Обязательный параметр/опция.

Пример:

```
// Удалим объекты с идентификаторами 34 и 25.
```

```
objectManager.objects.remove([34, 25]);
```

removeAll

```
{objectManager.ObjectCollection} removeAll()
```

Удаляет все элементы коллекции.

Возвращает ссылку на себя.

Пример:

```
objectManager.objects.removeAll();
```

setObjectOptions

```
{objectManager.ObjectCollection} setObjectOptions(objectId, options)
```

Метод, позволяющий динамически обновлять опции объекта. Метод следует использовать в случаях, когда вы хотите, чтобы новые опции мгновенно применились к отображению объекта на карте. Обратите внимание, менеджер не учитывает опцию 'visible'.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>objectId *</code>	—	Тип: Object Идентификатор объекта, которому нужно задать опции.
<code>options *</code>	—	Тип: Object Новые опции объекта.

* Обязательный параметр/опция.

Пример:

```
// Изменение цвета значка при наведении курсора.
objectManager.objects.events.add('mouseenter', function (e) {
    var objectId = e.get('objectId');
    var overlay = objectManager.objects.overlays.getById(objectId);
    setRedColor(objectId);
    overlay.events.add('mapchange', setGreenColor);
});

objectManager.objects.events.add('mouseleave', function (e) {
    var objectId = e.get('objectId');
    var overlay = objectManager.objects.overlays.getById(objectId);
    setGreenColor(objectId);
    overlay.events.remove('mapchange', setGreenColor);
});

function setGreenColor (objectId) {
    objectManager.objects.setObjectOptions(objectId, {
        preset: 'islands#greenIcon'
    });
}

function setRedColor (objectId) {
    objectManager.objects.setClusterOptions(objectId, {
        preset: 'islands#redIcon'
    });
}
```

objectManager.OverlayCollection

Расширяет [ICustomizable](#), [IEventEmitter](#).

Коллекция оверлеев.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
objectManager.OverlayCollection()
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
add	Добавление оверлея в коллекцию. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"><code>objectId</code> - идентификатор объекта, которому принадлежит оверлей;<code>overlay</code> - добавленный оверлей.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent . Более детальное описание доступно в domEvent.manager . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"><code>objectId</code> - идентификатор объекта, которому принадлежит оверлей;<code>overlay</code> - IOverlay - оверлей, на котором произошло событие;<code>coords</code> - географические координаты точки, в которой произошло событие;<code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;<code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;<code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;<code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей; • <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей; • <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей; • <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей; • <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей;• <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none">• <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей;• <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей; • <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Имена полей, доступных через метод <code>IMultiTouchEvent.get</code>:</p> <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей; • <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Имена полей, доступных через метод <code>IMultiTouchEvent.get</code>:</p> <ul style="list-style-type: none">• <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей;• <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Имена полей, доступных через метод <code>IMultiTouchEvent.get</code>:</p> <ul style="list-style-type: none">• <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей;• <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие;• <code>coords</code> - географические координаты точки, в которой произошло событие;• <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира;• <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы;• <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера;• <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
remove	Удаление оверлея из коллекции. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей; • <code>overlay</code> - удаленный оверлей.
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> • <code>objectId</code> - идентификатор объекта, которому принадлежит оверлей; • <code>overlay</code> - <code>IOverlay</code> - оверлей, на котором произошло событие; • <code>coords</code> - географические координаты точки, в которой произошло событие; • <code>globalPixels</code> - координаты события в глобальных пикселях от левого верхнего угла мира; • <code>pagePixels</code> - координаты события в пикселях от левого верхнего угла страницы; • <code>clientPixels</code> - координаты события в пикселях от левого верхнего угла окна браузера; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется.

Методы

Имя	Возвращает	Описание
each(callback, context)		
getAll()	<code>Object[]</code>	Возвращает массив объектов, содержащихся в коллекции.
getById(id)	<code>Object null</code>	Возвращает оверлей или <code>null</code> , если оверлея с переданным <code>id</code> не существует.
getId(overlay)	<code>Number null</code>	Возвращает идентификатор объекта или <code>null</code> , если оверлей не содержится в коллекции.
getIterator()	Iterator	Возвращает итератор по коллекции.
getLength()	<code>Number</code>	Возвращает количество объектов в коллекции.

Описание событий

add

Добавление оверлея в коллекцию. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - добавленный оверлей.

click

Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - [IOverlay](#) - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

contextmenu

Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - [IOverlay](#) - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

dblclick

Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - [IOverlay](#) - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mousedown

Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mouseenter

Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса `Event`. Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mouseleave

Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса `Event`. Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mousemove

Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса `Event`. Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

mouseup

Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса `Event`. Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;

- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

multitouchend

Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent` с информацией о прикосновениях. Имена полей, доступных через метод `IMultiTouchEvent.get`:

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

multitouchmove

Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent` с информацией о прикосновениях. Имена полей, доступных через метод `IMultiTouchEvent.get`:

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

multitouchstart

Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent` с информацией о прикосновениях. Имена полей, доступных через метод `IMultiTouchEvent.get`:

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

remove

Удаление оверлея из коллекции. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - удаленный оверлей.

wheel

Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MouseEvent](#). Более детальное описание доступно в [domEvent.manager](#). Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `objectId` - идентификатор объекта, которому принадлежит оверлей;
- `overlay` - `IOverlay` - оверлей, на котором произошло событие;
- `coords` - географические координаты точки, в которой произошло событие;
- `globalPixels` - координаты события в глобальных пикселях от левого верхнего угла мира;
- `pagePixels` - координаты события в пикселях от левого верхнего угла страницы;
- `clientPixels` - координаты события в пикселях от левого верхнего угла окна браузера;
- `domEvent` - исходное DOM-событие (в виде объекта [DomEvent](#)), если оно имеется.

Описание методов

each

```
{ } each(callback, context)
```

Параметры:

Параметр	Значение по умолчанию	Описание
callback *	—	Тип: Function Функция-обработчик, в которую передаются объекты коллекции.
context *	—	Тип: Object Контекст исполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

```
objectManager.clusters.overlays.each(function (overlay) {  
    overlay.options.set('cursor', 'help');  
});
```

getAll

```
{Object[] } getAll()
```

Возвращает массив объектов, содержащихся в коллекции.

Пример:

```
var clusterOverlayArray = objectManager.clusters.overlays.getAll();
```

getById

```
{Object|null } getById(id)
```

Возвращает оверлей или `null`, если оверлея с переданным `id` не существует.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>id *</code>	—	Тип: Number Идентификатор объекта, которому принадлежит оверлей.

* Обязательный параметр/опция.

Пример:

```
objectManager.objects.add('mouseenter', function (e) {  
    var objectId = e.get('objectId'),  
        overlay = objectManager.objects.overlays.getById(objectId);  
    overlay.options.set('zIndex', 100);  
});
```

getId

```
{Number|null} getId(overlay)
```

Возвращает идентификатор объекта или null, если оверлей не содержится в коллекции.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>overlay *</code>	—	Тип: IOverlay Оверлей.

* Обязательный параметр/опция.

Пример:

```
objectManager.objects.overlays.each(function (overlay) {  
    var objectId = objectManager.objects.overlays.getId(overlay);  
    objectManager.objects.setObjectOptions(objectId, {  
        preset: 'islands#redIcon'  
    });  
});
```

getIterator

```
{Iterator} getIterator()
```

Возвращает итератор по коллекции.

Пример:

```
var it = objectManager.objects.overlays.getIterator();  
var overlay;  
while ((overlay = it.getNext()) != it.STOP_ITERATION) {  
    overlay.options.set('zIndex', 10);  
}
```

getLength

```
{Number} getLength()
```

Возвращает количество объектов в коллекции.

Пример:

```
var objectsNumber = objectManager.objects.getLength(),
```

```
overlaysNumber = objectManager.objects.overlays.getNumber();  
alert('На данный момент на карте показано ' + overlaysNumber + ' из ' + objectsNumber.);
```

option

option.Manager

Расширяет [IOptionManager](#).

Менеджер опций. Позволяет задавать и получать значения опций по строковому ключу, а также разрешать значения опций в контексте существующей иерархии менеджеров опций.

Специальный ключ "preset" позволяет задать набор опций по умолчанию для данного менеджера. Значением опции "preset" может быть хэш вида {"имя опции": "значение опции"}, либо строковый идентификатор хэша опций в хранилище [option.presetStorage](#). Данный хэш опций может также содержать поле с именем "preset", что позволяет наследовать значения опций от других наборов.

При поиске значения в иерархии сначала проверяются собственные опции, затем опции заданные с помощью ключа "preset", после чего происходит обращение к родителю, если он есть.

Для отслеживания изменений некоторых опций можно использовать [Monitor](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
option.Manager([options[, parent[, name]])
```

Создает менеджер опций.

Параметры:

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Хэш опций.
parent	—	Тип: IOptionManager Родительский менеджер опций.
name	—	Тип: String Имя менеджера опций.

Примеры:

1.

```
// Пример построения иерархии менеджеров опций.  
var parentManager = new ymaps.option.Manager({  
  key1: '123'  
}),  
  childManager = new ymaps.option.Manager({  
  key2: '234'  
}, parentManager);  
// Выведет 123. значение берется из manager1.  
alert(childManager.get('key1'));  
// Выведет 234. значение берется из manager2.  
alert(childManager.get('key2'));  
// Перекрываем опцию.  
childManager.set('key1', '345');  
// Выведет 345, значение берется из manager2.  
alert(childManager.get('key1'));
```



```
// Выведет 123, значение берется из manager1.
alert(parentManager.get('key1'));
```

2.

```
// Пример использования опции "предустановка".
var optionManager = new ymaps.option.Manager({
  preset: 'islands#blueIcon'
});
var subOptionManager = new ymaps.option.Manager();
// Там нет данных, потому что subOptionManager пуст.
alert(subOptionManager.get('iconImageSize'));
// Связывание двух менеджеров.
subOptionManager.setParent(optionManager);
// [37, 42] - значение берется из пресета в Родительском менеджере.
alert(subOptionManager.get('iconImageSize'));
// Переопределение значения iconImageSize на уровне subOptionManager.
subOptionManager.set('iconImageSize', [10, 12]);
// [10, 12] - значение берется из subOptionManager.
alert(subOptionManager.get('iconImageSize'));
// Отмена переопределения iconImageSize.
subOptionManager.unset('iconImageSize');
// [37, 42] - значение снова берется из пресета в Родительском менеджере.
alert(subOptionManager.get('iconImageSize'));
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий объекта. Унаследовано от IFreezable .

События

Имя	Описание
change	Изменения произошли в значениях опций, либо в иерархии наследования опций. Экземпляр класса Event .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. Унаследовано от IChild .

Методы

Имя	Возвращает	Описание
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable .

Имя	Возвращает	Описание
get(key[, defaultValue])		<p>Возвращает значение заданной опции в контексте существующей иерархии наследования опций. При вызове данного метода сначала происходит поиск значения в текущем менеджере опций, а затем, если значение не определено, поиск продолжается в иерархии родительских менеджеров.</p> <p>Унаследован от IOptionManager.</p>
getAll()	Object	<p>Возвращает ссылку на внутренний хэш хранящий значения опций.</p> <p>Унаследован от IOptionManager.</p>
getName()	String	<p>Возвращает имя менеджера опций.</p> <p>Унаследован от IOptionManager.</p>
getNative(key)	Object	<p>Возвращает значение заданной опции, определенное на данном уровне иерархии опций, т.е. в данном менеджере.</p> <p>Унаследован от IOptionManager.</p>
getParent()	IOptionManager null	<p>Возвращает родительский менеджер опций.</p> <p>Унаследован от IOptionManager.</p>
isFrozen()	Boolean	<p>Возвращает true, если объект находится в "замороженном" режиме, иначе false.</p> <p>Унаследован от IFreezable.</p>
resolve(key[, name])	Object	<p>Метод, предназначенный для вызова дочерними менеджерами опций.</p> <p>Унаследован от IOptionManager.</p>

Имя	Возвращает	Описание
<code>set(key[, value])</code>	<code>option.Manager</code>	<p>Задаёт значения опций для данного менеджера. Поддерживаются две сигнатуры:</p> <ul style="list-style-type: none"> один аргумент, представляющий собой хеш вида {"имя опции": "значение опции"}; два аргумента, первый из которых - имя опции, второй - значение.
<code>setName(name)</code>		<p>Задаёт имя менеджера опций.</p> <p>Унаследован от <code>IOptionManager</code>.</p>
<code>setParent(parent)</code>	<code>IChild</code>	<p>Задаёт родительский менеджер опций.</p> <p>Унаследован от <code>IOptionManager</code>.</p>
<code>unfreeze()</code>	<code>IFreezable</code>	<p>Переводит объект в активный режим.</p> <p>Унаследован от <code>IFreezable</code>.</p>
<code>unset(keys)</code>	<code>option.Manager</code>	Стирает значения заданных опций в данном менеджере.
<code>unsetAll()</code>	<code>option.Manager</code>	Стирает значения всех опции в данном менеджере.

Описание событий

change

Изменения произошли в значениях опций, либо в иерархии наследования опций. Экземпляр класса `Event`.

Описание методов

set

```
{option.Manager} set(key[, value])
```

Задаёт значения опций для данного менеджера. Поддерживаются две сигнатуры:

- один аргумент, представляющий собой хеш вида {"имя опции": "значение опции"};
- два аргумента, первый из которых - имя опции, второй - значение.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key *</code>	—	Тип: Object String Имя опции, либо хеш вида {"имя опции": "значение опции"}.
<code>value</code>	—	Тип: Object Значение опции, если в качестве первого аргумента было передано имя.

* Обязательный параметр/опция.

Примеры:

1.

```
// Задание нескольких опций через хэш.  
myMap.options.set({  
  dblClickZoomCentering: true,  
  dblClickFloatZoom: true  
});  
// Генерируется одно событие option.Manager.event:change.
```

2.

```
// Задание опций отдельно.  
myMap.options  
  .set("dblClickZoomCentering", true)  
  .set("dblClickFloatZoom", true);  
// После каждого set будет генерироваться событие option.Manager.event:change.
```

3.

```
// Использование freeze для минимизации количества событий option.Manager.event:change.  
myMap.options.freeze();  
myMap.options.set({  
  dblClickZoomCentering: true,  
  dblClickFloatZoom: true  
});  
myMap.options.set('cursor', 'zoom');  
myMap.unfreeze();  
// Генерируется одно событие option.Manager.event:change.
```

unset

```
{option.Manager} unset(keys)
```

Стирает значения заданных опций в данном менеджере.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>keys *</code>	—	Тип: String String[] Имя опции или массив имен опций, значения которых должны быть аннулированы.

* Обязательный параметр/опция.

Пример:

```
map.options.unset(['dblClickZoomCentering', 'dblClickFloatZoom']);
```

unsetAll

```
{option.Manager} unsetAll()
```

Стирает значения всех опции в данном менеджере.

Возвращает ссылку на себя.

Пример:

```
var geoObject = new ymaps.Placemark([37, 55], {}, {preset:'islands#blueIcon'});
myMap.geoObjects.add(geoObject);
// Изменение стиля, как мы хотим.
geoObject.options.set({
  iconLayout: 'default#image',
  iconImageHref: 'http://mysite.ru/icon.png',
  iconImageSize: [16, 16]
});
// Восстановление первоначального внешнего вида
geoObject.options
  // Чтобы избежать двойной реакции объекта geo
  // чтобы опция изменилась, сначала мы вызываем "заморозку", а затем после
  // установив все значения, мы называем их "разморозкой".
  .freeze()
  .unsetAll()
  .set('preset', 'islands#blueIcon')
  .unfreeze();
```

option.presetStorage

Статический объект.







Экземпляр класса [util.Storage](#)

Хранилище предустановленных опций. Список доступных ключей:

Метки с текстом


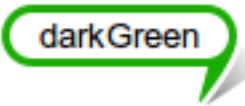









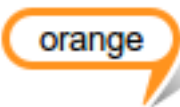

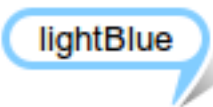


Содержимое метки задается через поле [properties.iconContent](#).

Значок	Ключ	Значок	Ключ
	'islands#blueIcon'		'islands#darkGreenIcon'
	'islands#redIcon'		'islands#violetIcon'
	'islands#darkOrangeIcon'		'islands#blackIcon'
	'islands#nightIcon'		'islands#yellowIcon'
	'islands#darkBlueIcon'		'islands#greenIcon'

Значок	Ключ	Значок	Ключ
	'islands#pinkIcon'		'islands#orangeIcon'
	'islands#grayIcon'		'islands#lightBlueIcon'
	'islands#brownIcon'		'islands#oliveIcon'

Метки с текстом (иконки тянутся под контент)

Содержимое метки задается через поле [properties.iconContent](#).

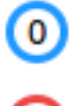











Значок	Ключ	Значок	Ключ
	'islands#blueStretchyIcon'		'islands#darkGreenStretchyIcon'
	'islands#redStretchyIcon'		'islands#violetStretchyIcon'
	'islands#darkOrangeStretchyIcon'		'islands#blackStretchyIcon'
	'islands#nightStretchyIcon'		'islands#yellowStretchyIcon'
	'islands#darkBlueStretchyIcon'		'islands#greenStretchyIcon'
	'islands#pinkStretchyIcon'		'islands#orangeStretchyIcon'
	'islands#grayStretchyIcon'		'islands#lightBlueStretchyIcon'
	'islands#brownStretchyIcon'		'islands#oliveStretchyIcon'





Метки без содержимого с точкой в центре

Значок	Ключ	Значок	Ключ
	'islands#blueDotIcon'		'islands#darkGreenDotIcon'
	'islands#redDotIcon'		'islands#violetDotIcon'
	'islands#darkOrangeDotIcon'		'islands#blackDotIcon'
	'islands#nightDotIcon'		'islands#yellowDotIcon'
	'islands#darkBlueDotIcon'		'islands#greenDotIcon'
	'islands#pinkDotIcon'		'islands#orangeDotIcon'
	'islands#grayDotIcon'		'islands#lightBlueDotIcon'
	'islands#brownDotIcon'		'islands#oliveDotIcon'

















Метки в виде кругов с текстом

Содержимое метки задается через поле [properties.iconContent](#).

Значок	Ключ	Значок	Ключ
	'islands#blueCircleIcon'		'islands#darkGreenCircleIcon'
	'islands#redCircleIcon'		'islands#violetCircleIcon'
	'islands#darkOrangeCircleIcon'		'islands#blackCircleIcon'
	'islands#nightCircleIcon'		'islands#yellowCircleIcon'
	'islands#darkBlueCircleIcon'		'islands#greenCircleIcon'
	'islands#pinkCircleIcon'		'islands#orangeCircleIcon'







Значок	Ключ	Значок	Ключ
	'islands#grayCircleIcon'		'islands#lightBlueCircleIcon'
	'islands#brownCircleIcon'		'islands#oliveCircleIcon'

Метки в виде кругов с точкой в центре

Значок	Ключ	Значок	Ключ
	'islands#blueCircleDotIcon'		'islands#darkGreenCircleDotIcon'
	'islands#redCircleDotIcon'		'islands#violetCircleDotIcon'
	'islands#darkOrangeCircleDotIcon'		'islands#blackCircleDotIcon'
	'islands#nightCircleDotIcon'		'islands#yellowCircleDotIcon'
	'islands#darkBlueCircleDotIcon'		'islands#greenCircleDotIcon'
	'islands#pinkCircleDotIcon'		'islands#orangeCircleDotIcon'
	'islands#grayCircleDotIcon'		'islands#lightBlueCircleDotIcon'
	'islands#brownCircleDotIcon'		'islands#oliveCircleDotIcon'



Метки со значком

Ключ для данного типа меток формируется по следующему правилу: 'islands#{цвет}{значок}Icon'. Например, 'islands#blueHomeIcon'. Ниже приведен список доступных значков и соответствующих для них ключей. Список доступных цветов можно посмотреть в любой другой таблице.

Значок	Ключ	Значок	Ключ
	'islands#blueAirportIcon'		'islands#blueAttentionIcon'
	'islands#blueAutoIcon'		'islands#blueBarIcon'
	'islands#blueBarberIcon'		'islands#blueBeachIcon'

Значок	Ключ	Значок	Ключ
	'islands#blueBicycleIcon'		'islands#blueBicycle2Icon'
	'islands#blueBookIcon'		'islands#blueCarWashIcon'
	'islands#blueChurchIcon'		'islands#blueCinemaIcon'
	'islands#blueCircusIcon'		'islands#blueCourtIcon'
	'islands#blueDeliveryIcon'		'islands#blueDiscountIcon'
	'islands#blueDogIcon'		'islands#blueEducationIcon'
	'islands#blueEntertainmentIcon'		'islands#blueFactoryIcon'
	'islands#blueFamilyIcon'		'islands#blueFashionIcon'
	'islands#blueFoodIcon'		'islands#blueFuelStationIcon'
	'islands#blueGardenIcon'		'islands#blueGovernmentIcon'
	'islands#blueHeartIcon'		'islands#blueHomeIcon'
	'islands#blueHotelIcon'		'islands#blueHydroIcon'
	'islands#blueInfoIcon'		'islands#blueLaundryIcon'
	'islands#blueLeisureIcon'		'islands#blueMassTransitIcon'









Значок	Ключ	Значок	Ключ
	'islands#blueMedical'		'islands#blueMoneyIcon'
	'islands#blueMountain'		'islands#blueNightClubIcon'
	'islands#blueObservation'		'islands#blueParkIcon'
	'islands#blueParking'		'islands#bluePersonIcon'
	'islands#bluePocket'		'islands#bluePoolIcon'
	'islands#bluePostIcon'		'islands#blueRailwayIcon'
	'islands#blueRapidTransit'		'islands#blueRepairShopIcon'
	'islands#blueRunIcon'		'islands#blueScienceIcon'
	'islands#blueShopping'		'islands#blueSouvenirsIcon'
	'islands#blueSportIcon'		'islands#blueStarIcon'
	'islands#blueTheater'		'islands#blueToiletIcon'
	'islands#blueUnderground'		'islands#blueVegetationIcon'
	'islands#blueVideoIcon'		'islands#blueWasteIcon'
	'islands#blueWaterPark'		'islands#blueWaterwayIcon'

Значок	Ключ	Значок	Ключ
	'islands#blueWorship'		'islands#blueZoolcon'


Метки в виде круга со значком

Ключ для данного типа меток формируется по следующему правилу: 'islands#{цвет}{значок}CircleIcon'. Например, 'islands#blueHomeCircleIcon'. Ниже приведен список доступных значков и соответствующих для них ключей. Список доступных цветов можно посмотреть в любой другой таблице.



Значок	Ключ	Значок	Ключ
	'islands#blueHomeCircleIcon'		'islands#blueScienceCircleIcon'
	'islands#blueAirportCircleIcon'		'islands#blueAttentionCircleIcon'
	'islands#blueAutoCircleIcon'		'islands#blueBarCircleIcon'
	'islands#blueBarberCircleIcon'		'islands#blueBeachCircleIcon'
	'islands#blueBicycleCircleIcon'		'islands#blueBicycle2CircleIcon'
	'islands#blueBookCircleIcon'		'islands#blueCarWashCircleIcon'
	'islands#blueChristianChurchCircleIcon'		'islands#blueCinemaCircleIcon'
	'islands#blueCircusCircleIcon'		'islands#blueCourtCircleIcon'
	'islands#blueDeliveryCircleIcon'		'islands#blueDiscountCircleIcon'
	'islands#blueDogCircleIcon'		'islands#blueEducationCircleIcon'
	'islands#blueEntertainmentCenterCircleIcon'		'islands#blueFactoryCircleIcon'
	'islands#blueFamilyCircleIcon'		'islands#blueFashionCircleIcon'
	'islands#blueFoodCircleIcon'		'islands#blueFuelStationCircleIcon'
	'islands#blueGardenCircleIcon'		'islands#blueGovernmentCircleIcon'
	'islands#blueHeartCircleIcon'		'islands#blueHomeCircleIcon'

Значок	Ключ	Значок	Ключ
	'islands#blueHotelCircleIcon'		'islands#blueHydroCircleIcon'
	'islands#blueInfoCircleIcon'		'islands#blueLaundryCircleIcon'
	'islands#blueLeisureCircleIcon'		'islands#blueMassTransitCircleIcon'
	'islands#blueMedicalCircleIcon'		'islands#blueMoneyCircleIcon'
	'islands#blueMountainCircleIcon'		'islands#blueNightClubCircleIcon'
	'islands#blueObservationCircleIcon'		'islands#blueParkCircleIcon'
	'islands#blueParkingCircleIcon'		'islands#bluePersonCircleIcon'
	'islands#bluePocketCircleIcon'		'islands#bluePoolCircleIcon'
	'islands#bluePostCircleIcon'		'islands#blueRailwayCircleIcon'
	'islands#blueRapidTransitCircleIcon'		'islands#blueRepairShopCircleIcon'
	'islands#blueRunCircleIcon'		'islands#blueScienceCircleIcon'
	'islands#blueShoppingCircleIcon'		'islands#blueSouvenirsCircleIcon'
	'islands#blueSportCircleIcon'		'islands#blueStarCircleIcon'
	'islands#blueTheaterCircleIcon'		'islands#blueToiletCircleIcon'
	'islands#blueUnderpassCircleIcon'		'islands#blueVegetationCircleIcon'
	'islands#blueVideoCircleIcon'		'islands#blueWasteCircleIcon'
	'islands#blueWaterParkCircleIcon'		'islands#blueWaterwayCircleIcon'
	'islands#blueWorshipCircleIcon'		'islands#blueZooCircleIcon'

Пиктограммы

Значок	Ключ
	'islands#geolocationIcon'

Значки кластеров

Значок	Ключ	Значок	Ключ
	'islands#blueClusterIcons'		'islands#invertedBlueClusterIcons'
	'islands#redClusterIcons'		'islands#invertedRedClusterIcons'
	'islands#darkOrangeClusterIcons'		'islands#invertedDarkOrangeClusterIcons'
	'islands#nightClusterIcons'		'islands#invertedNightClusterIcons'
	'islands#darkBlueClusterIcons'		'islands#invertedDarkBlueClusterIcons'
	'islands#pinkClusterIcons'		'islands#invertedPinkClusterIcons'
	'islands#grayClusterIcons'		'islands#invertedGrayClusterIcons'
	'islands#brownClusterIcons'		'islands#invertedBrownClusterIcons'

Значок	Ключ	Значок	Ключ
	'islands#darkGreenClusterIcons'		'islands#invertedDarkGreenClusterIcons'
	'islands#violetClusterIcons'		'islands#invertedVioletClusterIcons'
	'islands#blackClusterIcons'		'islands#invertedBlackClusterIcons'
	'islands#yellowClusterIcons'		'islands#invertedYellowClusterIcons'
	'islands#greenClusterIcons'		'islands#invertedGreenClusterIcons'
	'islands#orangeClusterIcons'		'islands#invertedOrangeClusterIcons'
	'islands#lightBlueClusterIcons'		'islands#invertedLightBlueClusterIcons'
	'islands#oliveClusterIcons'		'islands#invertedOliveClusterIcons'

Методы

Методы

Имя	Возвращает	Описание
<code>add(key, object)</code>	<code>util.Storage</code>	Добавляет объект в хранилище.

Имя	Возвращает	Описание
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.

overlay

overlay.Circle

Расширяет [IOverlay](#).

Интерактивный оверлей круга. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.Circle(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelCircleGeometry Пиксельная геометрия фигуры.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.fill	—	Тип: Boolean Наличие заливки graphics.style.color .
options.fillColor	—	Тип: String Цвет заливки.
options.fillImageHref	—	Тип: String Фоновое изображение. При включении данной опции значение <code>fillColor</code> игнорируется.

Параметр	Значение по умолчанию	Описание
options.fillMethod	'stretch'	<p>Тип: String</p> <p>Тип заливки фоном. Может принимать одно из двух значений:</p> <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - фоновое изображение повторяется без изменений размера. Аналог background-repeat в css. Можно использовать для заливки фигуры неким шаблоном.
options.fillOpacity	—	<p>Тип: Number</p> <p>Прозрачность заливки.</p>
options.interactive	true	<p>Тип: Boolean</p> <p>Опция, позволяющая отключить реакцию объекта на dom-события.</p>
options.opacity	—	<p>Тип: Number</p> <p>Общая прозрачность.</p>
options.outline	—	<p>Тип: Boolean</p> <p>Наличие обводки.</p>
options.separateContainer	—	<p>Тип: Boolean</p> <p>Производит отрисовку в отдельном слое.</p>
options.strokeColor	—	<p>Тип: String</p> <p>Цвет линии graphics.style.color.</p>
options.strokeOpacity	—	<p>Тип: Number</p> <p>Прозрачность контура.</p>
options.strokeStyle	—	<p>Тип: Number[] String</p> <p>Стиль контура(не поддерживается в режиме Canvas) graphics.style.stroke.</p>

Параметр	Значение по умолчанию	Описание
<code>options.strokeWidth</code>	—	Тип: Number Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
<code>options</code>	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
<code>click</code>	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
<code>contextmenu</code>	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
<code>datachange</code>	Изменение данных. Поля данных: <ul style="list-style-type: none"> <code>oldData</code> - старые данные; <code>newData</code> - новые данные. Унаследовано от IOverlay .
<code>dblclick</code>	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
<code>emptinesschange</code>	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .
<code>geometrychange</code>	Изменение геометрии. Поля данных: <ul style="list-style-type: none"> <code>oldGeometry</code> - старая пиксельная геометрия; <code>newGeometry</code> - новая пиксельная геометрия. Унаследовано от IOverlay .

Имя	Описание
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство touches, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> clientX - координата касания X относительно видимой области браузера; clientY - координата касания Y относительно видимой области браузера; pageX - координата касания X касания относительно начала документа; pageY - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	<code>Object</code>	<p>Возвращает объект данных оверлей.</p> <p>Унаследован от IOverlay.</p>
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>
getMap()	Map <code>null</code>	<p>Возвращает ссылку на текущую карту.</p> <p>Унаследован от IOverlay.</p>
getShape()	IShape <code>null</code>	<p>Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо <code>null</code>, если нет возможности рассчитать фигуру.</p> <p>Унаследован от IOverlay.</p>

Имя	Возвращает	Описание
<code>isEmpty()</code>	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
<code>setData(data)</code>		Задает данные оверлея. Унаследован от IOverlay .
<code>setGeometry(geometry)</code>		Задает пиксельную геометрию оверлея. Унаследован от IOverlay .
<code>setMap(map)</code>		Задает карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

overlay.hotspot

overlay.hotspot.Circle

Расширяет [IOverlay](#).

Оверлей круглой активной области. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.hotspot.Circle(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geometry *</code>	—	Тип: IPixelCircleGeometry Геометрия.
<code>data</code>	—	Тип: Object Данные.
<code>options</code>	—	Тип: Object Опции.
<code>options.fill</code>	—	Тип: Boolean Наличие заливки graphics.style.color .

Параметр	Значение по умолчанию	Описание
<code>options.fillColor</code>	—	Тип: String Цвет заливки.
<code>options.fillImageHref</code>	—	Тип: String Фоновое изображение. При включении данной опции значение <code>fillColor</code> игнорируется.
<code>options.fillMethod</code>	'stretch'	Тип: String Тип заливки фоном. Может принимать одно из двух значений: <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - фоновое изображение повторяется без изменений размера. Аналог background-repeat в css. Можно использовать для заливки фигуры неким шаблоном.
<code>options.fillOpacity</code>	—	Тип: Number Прозрачность заливки.
<code>options.interactive</code>	true	Тип: Boolean Опция, позволяющая отключить реакцию объекта на dom-события.
<code>options.opacity</code>	—	Тип: Number Общая прозрачность.
<code>options.outline</code>	—	Тип: Boolean Наличие обводки.
<code>options.separateContainer</code>	—	Тип: Boolean Производит отрисовку в отдельном слое.
<code>options.strokeColor</code>	—	Тип: String Цвет линии <code>graphics.style.color</code> .

Параметр	Значение по умолчанию	Описание
options.strokeOpacity	—	Тип: Number Прозрачность контура.
options.strokeStyle	—	Тип: Number[] String Стиль контура(не поддерживается в режиме Canvas) graphics.style.stroke .
options.strokeWidth	—	Тип: Number Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> oldData - старые данные; newData - новые данные. Унаследовано от IOverlay .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .

Имя	Описание
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	<p>Возвращает объект данных оверлея.</p> <p>Унаследован от IOverlay.</p>
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>

Имя	Возвращает	Описание
<code>getMap()</code>	<code>Map</code> null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
<code>getShape()</code>	<code>IShape</code> null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
<code>isEmpty()</code>	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
<code>setData(data)</code>		Задаёт данные оверлея. Унаследован от IOverlay .
<code>setGeometry(geometry)</code>		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
<code>setMap(map)</code>		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

overlay.hotspot.Placemark

Расширяет [IOverlay](#).

Оверлей точечной активной области. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.hotspot.Placemark(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geometry</code> *	—	Тип: IPixelPointGeometry Пиксельная геометрия фигуры.
<code>data</code>	—	Тип: Object Данные.

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.cursor</code>	—	Тип: String Курсор при наведении мыши.
<code>options.interactive</code>	true	Тип: Boolean Опция, позволяющая отключить реакцию объекта на dom-события.
<code>options.interactivityModel</code>	'default#geoObject'	Тип: String Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
<code>options.layout</code>	—	Тип: Function String Макет. (Тип: конструктор объекта с интерфейсом ILayout).
<code>options.offset</code>	[0,0]	Тип: Array Смещение в пикселях.
<code>options.pane</code>	'places'	Тип: String Контейнер, в который будет помещен макет метки.
<code>options.shadow</code>	false	Тип: Boolean Флаг наличия тени.
<code>options.shadowLayout</code>	—	Тип: Function String Макет тени. (Тип: конструктор объекта с интерфейсом ILayout).
<code>options.shadowOffset</code>	[0,0]	Тип: Array Смещение тени в пикселях.
<code>options.shadowsPane</code>	'shadows'	Тип: Array Контейнер, в который будет помещен макет тени метки.
<code>options.zIndex</code>	—	Тип: Number z-index элемента.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> <code>oldData</code> - старые данные; <code>newData</code> - новые данные. Унаследовано от IOverlay .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .
geometrychange	Изменение геометрии. Поля данных: <ul style="list-style-type: none"> <code>oldGeometry</code> - старая пиксельная геометрия; <code>newGeometry</code> - новая пиксельная геометрия. Унаследовано от IOverlay .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IOverlay .

Имя	Описание
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	<code>Object</code>	<p>Возвращает объект данных оверлей.</p> <p>Унаследован от IOverlay.</p>
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>
getMap()	Map <code>null</code>	<p>Возвращает ссылку на текущую карту.</p> <p>Унаследован от IOverlay.</p>
getShape()	IShape <code>null</code>	<p>Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо <code>null</code>, если нет возможности рассчитать фигуру.</p> <p>Унаследован от IOverlay.</p>

Имя	Возвращает	Описание
<code>isEmpty()</code>	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
<code>setData(data)</code>		Задаёт данные оверлея. Унаследован от IOverlay .
<code>setGeometry(geometry)</code>		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
<code>setMap(map)</code>		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

overlay.hotspot.Polygon

Расширяет [IOverlay](#).

Оверлей активной области в форме многоугольника. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.hotspot.Polygon(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geometry *</code>	—	Тип: IPixelPolygonGeometry Геометрия фигуры.
<code>data</code>	—	Тип: Object Данные.
<code>options</code>	—	Тип: Object Опции.
<code>options.fill</code>	—	Тип: String Наличие заливки.
<code>options.fillColor</code>	—	Тип: String Цвет заливки graphics.style.color .

Параметр	Значение по умолчанию	Описание
options.fillImageHref	—	<p>Тип: String</p> <p>Фоновое изображение. При включении данной опции значение <code>fillColor</code> игнорируется.</p>
options.fillMethod	'stretch'	<p>Тип: String</p> <p>Тип заливки фоном. Может принимать одно из двух значений:</p> <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - Фоновое изображение повторяется без изменений размера. Аналог <code>background-repeat</code> в css. Можно использовать для заливки фигуры неким шаблоном.
options.fillOpacity	—	<p>Тип: Number</p> <p>Прозрачность заливки.</p>
options.interactive	true	<p>Тип: Boolean</p> <p>Опция, позволяющая отключить реакцию объекта на dom-события.</p>
options.opacity	—	<p>Тип: Number</p> <p>Общая прозрачность.</p>
options.outline	—	<p>Тип: String</p> <p>Наличие обводки.</p>
options.separateContainer	—	<p>Тип: Boolean</p> <p>Производит отрисовку в отдельном слое.</p>
options.strokeColor	—	<p>Тип: String</p> <p>Цвет линии graphics.style.color.</p>
options.strokeOpacity	—	<p>Тип: Number</p> <p>Прозрачность контура.</p>

Параметр	Значение по умолчанию	Описание
options.strokeStyle	—	Тип: <code>Number[] String</code> Стиль контура graphics.style.stroke .
options.strokeWidth	—	Тип: <code>Number</code> Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> <code>oldData</code> - старые данные; <code>newData</code> - новые данные. Унаследовано от IOverlay .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .

Имя	Описание
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	<p>Возвращает объект данных оверлея.</p> <p>Унаследован от IOverlay.</p>
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>

Имя	Возвращает	Описание
<code>getMap()</code>	<code>Map</code> null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
<code>getShape()</code>	<code>IShape</code> null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
<code>isEmpty()</code>	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
<code>setData(data)</code>		Задаёт данные оверлея. Унаследован от IOverlay .
<code>setGeometry(geometry)</code>		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
<code>setMap(map)</code>		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

overlay.hotspot.Polyline

Расширяет [IOverlay](#).

Оверлей активной области в форме ломаной линии. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.hotspot.Polyline(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geometry *</code>	—	Тип: IPixelLineStringGeometry Пиксельная геометрия фигуры.
<code>data</code>	—	Тип: Object Данные.

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции.
options.interactive	true	Тип: Boolean Опция, позволяющая отключить реакцию объекта на dom-события.
options.opacity	—	Тип: Number Общая прозрачность.
options.separateContainer	—	Тип: Boolean Производит отрисовку в отдельном слое.
options.strokeColor	—	Тип: String Цвет линии graphics.style.color .
options.strokeOpacity	—	Тип: Number Прозрачность контура.
options.strokeStyle	—	Тип: Number[] String Стиль контура graphics.style.stroke .
options.strokeWidth	—	Тип: Number Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
datachange	<p>Изменение данных. Поля данных:</p> <ul style="list-style-type: none"> oldData - старые данные; newData - новые данные. <p>Унаследовано от IOverlay.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
emptinesschange	<p>Изменение признака пустоты оверлея. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>

Имя	Описание
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	<p>Возвращает объект данных оверлея.</p> <p>Унаследован от IOverlay.</p>
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>
getMap()	Map null	<p>Возвращает ссылку на текущую карту.</p> <p>Унаследован от IOverlay.</p>
getShape()	IShape null	<p>Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру.</p> <p>Унаследован от IOverlay.</p>
isEmpty()	Boolean	<p>Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.</p> <p>Унаследован от IOverlay.</p>
setData(data)		<p>Задаёт данные оверлея.</p> <p>Унаследован от IOverlay.</p>
setGeometry(geometry)		<p>Задаёт пиксельную геометрию оверлея.</p> <p>Унаследован от IOverlay.</p>
setMap(map)		<p>Задаёт карту, на которой должен отобразиться оверлей.</p> <p>Унаследован от IOverlay.</p>

overlay.hotspot.Rectangle

Расширяет [IOverlay](#).

Оверлей прямоугольной активной области. По умолчанию оверлеи не добавлены в package.full (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.hotspot.Rectangle(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelRectangleGeometry Пиксельная геометрия фигуры.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.borderRadius	—	Тип: Number Радиус скругления углов.
options.fill	—	Тип: String Наличие заливки.
options.fillColor	—	Тип: String Цвет заливки graphics.style.color .
options.fillImageHref	—	Тип: String Фоновое изображение. При включении данной опции значение <code>fillColor</code> игнорируется.
options.fillMethod	'stretch'	Тип: String Тип заливки фоном. Может принимать одно из двух значений: <ul style="list-style-type: none">• <code>stretch</code> - фоновое изображение растягивается по размеру оверлея.• <code>tile</code> - Фоновое изображение повторяется без изменений размера. Аналог <code>background-repeat</code> в css. Можно использовать для заливки фигуры неким шаблоном.

Параметр	Значение по умолчанию	Описание
options.fillOpacity	—	Тип: Number Прозрачность заливки.
options.interactive	true	Тип: Boolean Опция, позволяющая отключить реакцию объекта на dom-события.
options.opacity	—	Тип: Number Общая прозрачность.
options.outline	—	Тип: String Наличие обводки.
options.separateContainer	—	Тип: Boolean Производит отрисовку в отдельном слое.
options.strokeColor	—	Тип: String Цвет линии graphics.style.color .
options.strokeOpacity	—	Тип: Number Прозрачность контура.
options.strokeStyle	—	Тип: Number[] String Стиль контура(не поддерживается в режиме Canvas) graphics.style.stroke .
options.strokeWidth	—	Тип: Number Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
datachange	<p>Изменение данных. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldData</code> - старые данные; • <code>newData</code> - новые данные. <p>Унаследовано от IOverlay.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
emptinesschange	<p>Изменение признака пустоты оверлея. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldGeometry</code> - старая пиксельная геометрия; • <code>newGeometry</code> - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldMap</code> - старая карта; • <code>newMap</code> - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
shapechange	Изменение фигуры области охватывающей оверлей. Экземпляр класса Event . Унаследовано от IOverlay .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных оверлея. Унаследован от IOverlay .
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay .
getMap()	Map null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
getShape()	IShape null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
isEmpty()	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
setData(data)		Задаёт данные оверлея. Унаследован от IOverlay .
setGeometry(geometry)		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
setMap(map)		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

overlay.html

HTML-оверлей балуна. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.html.Balloon(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelPointGeometry Пиксельная геометрия фигуры.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.cursor	—	Тип: String Курсор при наведении мыши.
options.interactivityModel	"default#opaque"	Тип: String Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
options.layout	—	Тип: Function String Макет. (Тип: конструктор объекта с интерфейсом ILayout).
options.offset	[0,0]	Тип: Array Смещение в пикселях.
options.pane	"balloon"	Тип: String Контейнер, в который будет помещен макет балуна.
options.shadow	true	Тип: Boolean Флаг наличия тени.
options.shadowLayout	—	Тип: Function String Макет тени (Тип: конструктор объекта с интерфейсом ILayout).

Параметр	Значение по умолчанию	Описание
options.shadowOffset	[0,0]	Тип: Array Смещение тени в пикселях.
options.shadowsPane	"shadows"	Тип: Array Контейнер, в который будет помещен макет тени балуна.
options.zIndex	—	Тип: Number z-index элемента.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> • <code>oldData</code> - старые данные; • <code>newData</code> - новые данные. Унаследовано от IOverlay .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .

Имя	Описание
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getBalloonElement()	<code>HTMLElement</code>	Возвращает родительский элемент макета балуна.
getBalloonLayout()	vow.Promise	Возвращает Promise объект макет балуна.
getBalloonLayoutSync()	ILayout	Возвращает макет балуна.

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных оверлея. Унаследован от IOverlay .
getElement()	HTMLElement	Возвращает родительский элемент макета балуна.
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay .
getLayout()	vow.Promise	Возвращает Promise объект макет балуна.
getLayoutSync()	ILayout null	Возвращает макет балуна.
getMap()	Map null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
getMode()	String	Возвращает текущий режим балуна: "panel" — режим панели, "standard" — стандартное отображение в виде всплывающего окна.
getShadowElement()	HTMLElement	Возвращает родительский элемент макета тени балуна.
getShadowLayout()	vow.Promise	Возвращает Promise объект, который будет разрешен макетом тени балуна.
getShadowLayoutSync()	ILayout null	Возвращает макет тени балуна.
getShape()	IShape null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
isEmpty()	Boolean	Возвращает true, если макет пуст или если макет еще не был загружен, т.е. не имеет содержимого.
setData(data)		Задаёт данные оверлея. Унаследован от IOverlay .
setGeometry(geometry)		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .

Имя	Возвращает	Описание
<code>setMap(map)</code>		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

Описание методов

`getBalloonElement`

```
{HTMLElement} getBalloonElement()
```

Возвращает родительский элемент макета балуна.

`getBalloonLayout`

```
{vow.Promise} getBalloonLayout()
```

Возвращает Promise объект, который будет разрешен макетом балуна.

`getBalloonLayoutSync`

```
{ILayout} getBalloonLayoutSync()
```

Возвращает макет балуна.

`getElement`

```
{HTMLElement} getElement()
```

Возвращает родительский элемент макета балуна.

`getLayout`

```
{vow.Promise} getLayout()
```

Возвращает Promise объект, который будет разрешен макетом балуна.

`getLayoutSync`

```
{ILayout|null} getLayoutSync()
```

Возвращает макет балуна.

`getMode`

```
{String} getMode()
```

Возвращает текущий режим балуна: "panel" — режим панели, "standard" — стандартное отображение в виде всплывающего окна.

`getShadowElement`

```
{HTMLElement} getShadowElement()
```

Возвращает родительский элемент макета тени балуна.

getShadowLayout

```
{vow.Promise} getShadowLayout()
```

Возвращает Promise объект, который будет разрешен макетом тени балуна.

getShadowLayoutSync

```
{ILayout|null} getShadowLayoutSync()
```

Возвращает макет тени балуна.

isEmpty

```
{Boolean} isEmpty()
```

Возвращает true, если макет пуст или если макет еще не был загружен, т.е. не имеет содержимого.

overlay.html.Hint

Расширяет [IOverlay](#).

Простой HTML-оверлей. По умолчанию оверлеи не добавлены в package.full (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.html.Hint(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelPointGeometry Пиксельная геометрия фигуры.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.cursor	—	Тип: String Курсор при наведении мыши.
options.interactivityModel	'default#opaque'	Тип: String Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
options.layout	—	Тип: ILayout String Макет.

Параметр	Значение по умолчанию	Описание
options.pane	"outerHint"	Тип: String Контейнер, в который будет помещен оверлей.
options.zIndex	—	Тип: Number z-index элемента.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> oldData - старые данные; newData - новые данные. Унаследовано от IOverlay .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .

Имя	Описание
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	<p>Возвращает объект данных оверлея.</p> <p>Унаследован от IOverlay.</p>
getElement()	HTMLElement	<p>Возвращает родительский элемент макета иконки.</p>
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>

Имя	Возвращает	Описание
<code>getLayout()</code>	<code>vow.Promise</code>	Возвращает Promise объект, который будет разрешен макетом иконки.
<code>getLayoutSync()</code>	<code>ILayout null</code>	Возвращает макет иконки.
<code>getMap()</code>	<code>Map null</code>	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
<code>getShape()</code>	<code>IShape null</code>	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
<code>isEmpty()</code>	Boolean	Возвращает true, если макет пуст или если макет еще не был загружен, т.е. не имеет содержимого.
<code>setData(data)</code>		Задает данные оверлея. Унаследован от IOverlay .
<code>setGeometry(geometry)</code>		Задает пиксельную геометрию оверлея. Унаследован от IOverlay .
<code>setMap(map)</code>		Задает карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

Описание методов

getElement

```
{HTMLElement} getElement()
```

Возвращает родительский элемент макета иконки.

getLayout

```
{vow.Promise} getLayout()
```

Возвращает Promise объект, который будет разрешен макетом иконки.

getLayoutSync

```
{ILayout|null} getLayoutSync()
```

Возвращает макет иконки.

isEmpty

```
{Boolean} isEmpty()
```

Возвращает true, если макет пуст или если макет еще не был загружен, т.е. не имеет содержимого.

overlay.html.Placemark

Расширяет [IOverlay](#).

HTML-оверлей метки. По умолчанию оверлеи не добавлены в package.full (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.html.Placemark(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelPointGeometry Пиксельная геометрия.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.cursor	—	Тип: String Курсор при наведении мыши.
options.interactivityModel	'default#geoObject'	Тип: String Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
options.layout	—	Тип: Function String Макет. (Тип: конструктор объекта с интерфейсом ILayout).
options.offset	[0,0]	Тип: Array Смещение в пикселях.
options.pane	'places'	Тип: String Контейнер, в который будет помещен макет метки.
options.shadow	false	Тип: Boolean Флаг наличия тени.

Параметр	Значение по умолчанию	Описание
options.shadowLayout	—	Тип: Function String Макет тени. (Тип: конструктор объекта с интерфейсом ILayout).
options.shadowOffset	[0,0]	Тип: Array Смещение тени в пикселях.
options.shadowsPane	'shadows'	Тип: Array Контейнер, в который будет помещен макет тени метки.
options.zIndex	—	Тип: Number z-index элемента.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> oldData - старые данные; newData - новые данные. Унаследовано от IOverlay .

Имя	Описание
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
emptinesschange	<p>Изменение признака пустоты оверлея. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	<p>Возвращает объект данных оверлея.</p> <p>Унаследован от IOverlay.</p>
getElement()	HTMLElement	<p>Возвращает родительский элемент макета иконки.</p>
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>

Имя	Возвращает	Описание
getIconElement()	HTMLElement	Возвращает родительский элемент макета иконки.
getIconLayout()	vow.Promise	Возвращает Promise объект, который будет разрешен макетом иконки.
getIconLayoutSync()	ILayout null	Возвращает макет иконки.
getLayout()	vow.Promise	Возвращает Promise объект, который будет разрешен макетом иконки.
getLayoutSync()	ILayout null	Возвращает макет иконки.
getMap()	Map null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
getShadowElement()	HTMLElement	Возвращает родительский элемент макета тени иконки.
getShadowLayout()	vow.Promise	Возвращает Promise объект, который будет разрешен макетом тени иконки.
getShadowLayoutSync()	ILayout null	Возвращает макет тени иконки.
getShape()	IShape null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
isEmpty()	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
setData(data)		Задаёт данные оверлея. Унаследован от IOverlay .
setGeometry(geometry)		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
setMap(map)		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

Описание методов

getElement

```
{HTMLElement} getElement()
```

Возвращает родительский элемент макета иконки.

getIconElement

```
{HTMLElement} getIconElement()
```

Возвращает родительский элемент макета иконки.

getIconLayout

```
{vow.Promise} getIconLayout()
```

Возвращает Promise объект, который будет разрешен макетом иконки.

getIconLayoutSync

```
{ILayout|null} getIconLayoutSync()
```

Возвращает макет иконки.

getLayout

```
{vow.Promise} getLayout()
```

Возвращает Promise объект, который будет разрешен макетом иконки.

getLayoutSync

```
{ILayout|null} getLayoutSync()
```

Возвращает макет иконки.

getShadowElement

```
{HTMLElement} getShadowElement()
```

Возвращает родительский элемент макета тени иконки.

getShadowLayout

```
{vow.Promise} getShadowLayout()
```

Возвращает Promise объект, который будет разрешен макетом тени иконки.

getShadowLayoutSync

```
{ILayout|null} getShadowLayoutSync()
```

Возвращает макет тени иконки.

overlay.html.Rectangle

Расширяет [IOverlay](#).

HTML-оверлей прямоугольника. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.html.Rectangle(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelRectangleGeometry Пиксельная геометрия фигуры.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции оверлея.
options.cursor	—	Тип: String Курсор при наведении мыши.
options.fillColor	—	Тип: String Цвет заливки graphics.style.color . Опция стандартного макета прямоугольника.
options.fillImageHref	—	Тип: String Фоновое изображение. При включении данной опции значение <code>fillColor</code> игнорируется. Опция стандартного макета прямоугольника.

Параметр	Значение по умолчанию	Описание
options.fillMethod	'stretch'	<p>Тип: String</p> <p>Тип заливки фоном. Может принимать одно из двух значений:</p> <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - Фоновое изображение повторяется без изменений размера. Аналог background-repeat в css. Можно использовать для заливки фигуры неким шаблоном. <p>Опция стандартного макета прямоугольника.</p>
options.fillOpacity	—	<p>Тип: Number</p> <p>Прозрачность заливки. Опция стандартного макета прямоугольника.</p>
options.interactivityModel	'default#geoObject'	<p>Тип: String</p> <p>Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage.</p>
options.opacity	—	<p>Тип: Number</p> <p>Общая прозрачность.</p>
options.pane	"areas"	<p>Тип: String</p> <p>Контейнер, в который будет помещен оверлей.</p>
options.strokeColor	—	<p>Тип: String</p> <p>Цвет линии graphics.style.color. Опция стандартного макета прямоугольника.</p>
options.strokeStyle	—	<p>Тип: Number[] String</p> <p>Стиль обводки, поддерживаемый стандартным CSS-свойством <i>border-style</i>. Опция стандартного макета прямоугольника.</p>
options.strokeWidth	—	<p>Тип: Number</p> <p>Ширина линии. Опция стандартного макета прямоугольника.</p>

Параметр	Значение по умолчанию	Описание
options.zIndex	—	Тип: Number z-index элемента.
dataSet.options.borderRadius	—	Тип: Number Радиус скругления углов. Опция стандартного макета прямоугольника.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> oldData - старые данные; newData - новые данные. Унаследовано от IOverlay .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .

Имя	Описание
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	<p>Возвращает объект данных оверлея.</p> <p>Унаследован от IOverlay.</p>
getElement()	HTMLElement	<p>Возвращает родительский элемент макета прямоугольника.</p>

Имя	Возвращает	Описание
<code>getGeometry()</code>	<code>IPixelGeometry</code>	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay .
<code>getLayout()</code>	<code>vow.Promise</code>	Возвращает Promise объект вернуть макет прямоугольника.
<code>getLayoutSync()</code>	<code>ILayout null</code>	Возвращает макет прямоугольника.
<code>getMap()</code>	<code>Map null</code>	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
<code>getShape()</code>	<code>IShape null</code>	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
<code>isEmpty()</code>	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
<code>setData(data)</code>		Задает данные оверлея. Унаследован от IOverlay .
<code>setGeometry(geometry)</code>		Задает пиксельную геометрию оверлея. Унаследован от IOverlay .
<code>setMap(map)</code>		Задает карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

Описание методов

getElement

```
{HTMLElement} getElement()
```

Возвращает родительский элемент макета прямоугольника.

getLayout

```
{vow.Promise} getLayout()
```

Возвращает объект-обещание вернуть макет прямоугольника.

getLayoutSync

```
{ILayout|null} getLayoutSync()
```

Возвращает макет прямоугольника.

overlay.Pin

Расширяет [IOverlay](#).

Интерактивный оверлей круглой метки. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.Pin(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelPointGeometry Пиксельная геометрия метки.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.fill	—	Тип: Boolean Наличие заливки graphics.style.color .
options.fillColor	—	Тип: String Цвет заливки.
options.fillImageHref	—	Тип: String Фоновое изображение. При включении данной опции значение <code>fillColor</code> игнорируется.

Параметр	Значение по умолчанию	Описание
options.fillMethod	'stretch'	<p>Тип: String</p> <p>Тип заливки фоном. Может принимать одно из двух значений:</p> <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - фоновое изображение повторяется без изменений размера. Аналог background-repeat в css. Можно использовать для заливки фигуры неким шаблоном.
options.fillOpacity	—	<p>Тип: Number</p> <p>Прозрачность заливки.</p>
options.interactive	true	<p>Тип: Boolean</p> <p>Опция, позволяющая отключить реакцию объекта на dom-события.</p>
options.opacity	—	<p>Тип: Number</p> <p>Общая прозрачность.</p>
options.outline	—	<p>Тип: Boolean</p> <p>Наличие обводки.</p>
options.radius	—	<p>Тип: Number</p> <p>Радиус метки в пикселях.</p>
options.separateContainer	—	<p>Тип: Boolean</p> <p>Производит отрисовку в отдельном слое.</p>
options.strokeColor	—	<p>Тип: String</p> <p>Цвет линии graphics.style.color.</p>
options.strokeOpacity	—	<p>Тип: Number</p> <p>Прозрачность контура.</p>

Параметр	Значение по умолчанию	Описание
options.strokeStyle	—	Тип: Number[] String Стиль контура(не поддерживается в режиме Canvas) graphics.style.stroke .
options.strokeWidth	—	Тип: Number Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> oldData - старые данные; newData - новые данные. Унаследовано от IOverlay .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .

Имя	Описание
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	<p>Возвращает объект данных оверлея.</p> <p>Унаследован от IOverlay.</p>
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>

Имя	Возвращает	Описание
getMap()	Map null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
getShape()	IShape null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
isEmpty()	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
setData(data)		Задаёт данные оверлея. Унаследован от IOverlay .
setGeometry(geometry)		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
setMap(map)		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

overlay.Placemark

Расширяет [IOverlay](#).

Оверлей метки. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.Placemark(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelPointGeometry Пиксельная геометрия.
data	—	Тип: Object Данные.

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.cursor</code>	—	Тип: String Курсор при наведении мыши.
<code>options.interactive</code>	true	Тип: Boolean Опция, позволяющая отключить реакцию объекта на dom-события.
<code>options.interactivityModel</code>	'default#geoObject'	Тип: String Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
<code>options.layout</code>	—	Тип: Function String Макет. (Тип: конструктор объекта с интерфейсом ILayout).
<code>options.offset</code>	[0,0]	Тип: Array Смещение в пикселях.
<code>options.pane</code>	'places'	Тип: String Контейнер, в который будет помещен макет метки.
<code>options.shadow</code>	false	Тип: Boolean Флаг наличия тени.
<code>options.shadowLayout</code>	—	Тип: Function String Макет тени. (Тип: конструктор объекта с интерфейсом ILayout).
<code>options.shadowOffset</code>	[0,0]	Тип: Array Смещение тени в пикселях.
<code>options.shadowsPane</code>	'shadows'	Тип: Array Контейнер, в который будет помещен макет тени метки.
<code>options.zIndex</code>	—	Тип: Number z-index элемента.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
datachange	Изменение данных. Поля данных: <ul style="list-style-type: none"> <code>oldData</code> - старые данные; <code>newData</code> - новые данные. Унаследовано от IOverlay .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
emptinesschange	Изменение признака пустоты оверлея. Экземпляр класса Event . Унаследовано от IOverlay .
geometrychange	Изменение геометрии. Поля данных: <ul style="list-style-type: none"> <code>oldGeometry</code> - старая пиксельная геометрия; <code>newGeometry</code> - новая пиксельная геометрия. Унаследовано от IOverlay .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IOverlay .

Имя	Описание
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	<p>Возвращает объект данных оверлей.</p> <p>Унаследован от IOverlay.</p>
getElement()	HTMLElement	Возвращает родительский элемент макета иконки.
getGeometry()	IPixelGeometry	<p>Возвращает текущую пиксельную геометрию.</p> <p>Унаследован от IOverlay.</p>
getIconElement()	HTMLElement	Возвращает родительский элемент макета иконки.
getIconLayout()	vow.Promise	Возвращает Promise объект, который будет разрешен макетом иконки.
getIconLayoutSync()	ILayout null	Возвращает макет иконки.
getLayout()	vow.Promise	Возвращает Promise объект, который будет разрешен макетом иконки.
getLayoutSync()	ILayout null	Возвращает макет иконки.

Имя	Возвращает	Описание
<code>getMap()</code>	<code>Map</code> null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
<code>getShadowElement()</code>	<code>HTMLElement</code>	Возвращает родительский элемент макета тени иконки.
<code>getShadowLayout()</code>	<code>vow.Promise</code>	Возвращает Promise объект, который будет разрешен макетом тени иконки.
<code>getShadowLayoutSync()</code>	<code>ILayout</code> null	Возвращает макет тени иконки.
<code>getShape()</code>	<code>IShape</code> null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
<code>isEmpty()</code>	<code>Boolean</code>	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
<code>setData(data)</code>		Задаёт данные оверлея. Унаследован от IOverlay .
<code>setGeometry(geometry)</code>		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
<code>setMap(map)</code>		Задаёт карту, на которой должен отображаться оверлей. Унаследован от IOverlay .

Описание методов

getElement

```
{HTMLElement} getElement()
```

Возвращает родительский элемент макета иконки.

getIconElement

```
{HTMLElement} getIconElement()
```

Возвращает родительский элемент макета иконки.

getIconLayout

```
{vow.Promise} getIconLayout()
```

Возвращает Promise объект, который будет разрешен макетом иконки.

getIconLayoutSync

```
{ILayout|null} getIconLayoutSync()
```

Возвращает макет иконки.

getLayout

```
{vow.Promise} getLayout()
```

Возвращает Promise объект, который будет разрешен макетом иконки.

getLayoutSync

```
{ILayout|null} getLayoutSync()
```

Возвращает макет иконки.

getShadowElement

```
{HTMLElement} getShadowElement()
```

Возвращает родительский элемент макета тени иконки.

getShadowLayout

```
{vow.Promise} getShadowLayout()
```

Возвращает Promise объект, который будет разрешен макетом тени иконки.

getShadowLayoutSync

```
{ILayout|null} getShadowLayoutSync()
```

Возвращает макет тени иконки.

overlay.Polygon

Расширяет [IOverlay](#).

Оверлей прямоугольника.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.Polygon(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelPolygonGeometry Пиксельная геометрия фигуры.

Параметр	Значение по умолчанию	Описание
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.fill	—	Тип: String Наличие заливки.
options.fillColor	—	Тип: String Цвет заливки graphics.style.color .
options.fillImageHref	—	Тип: String Фоновое изображение. При включении данной опции значение <code>fillColor</code> игнорируется.
options.fillMethod	'stretch'	Тип: String Тип заливки фоном. Может принимать одно из двух значений: <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - Фоновое изображение повторяется без изменений размера. Аналог <code>background-repeat</code> в css. Можно использовать для заливки фигуры неким шаблоном.
options.fillOpacity	—	Тип: Number Прозрачность заливки.
options.interactive	true	Тип: Boolean Опция, позволяющая отключить реакцию объекта на dom-события.
options.opacity	—	Тип: Number Общая прозрачность.

Параметр	Значение по умолчанию	Описание
options.outline	—	Тип: String Наличие обводки.
options.separateContainer	—	Тип: Boolean Производит отрисовку в отдельном слое.
options.strokeColor	—	Тип: String Цвет линии graphics.style.color .
options.strokeOpacity	—	Тип: Number Прозрачность контура.
options.strokeStyle	—	Тип: Number[] String Стиль контура graphics.style.stroke .
options.strokeWidth	—	Тип: Number Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
datachange	<p>Изменение данных. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldData</code> - старые данные; • <code>newData</code> - новые данные. <p>Унаследовано от IOverlay.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
emptinesschange	<p>Изменение признака пустоты оверлея. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldGeometry</code> - старая пиксельная геометрия; • <code>newGeometry</code> - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldMap</code> - старая карта; • <code>newMap</code> - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
shapechange	<p>Изменение фигуры области охватывающей оверлей. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных оверлея. Унаследован от IOverlay .
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay .
getMap()	Map null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
getShape()	IShape null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
isEmpty()	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
setData(data)		Задаёт данные оверлея. Унаследован от IOverlay .
setGeometry(geometry)		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
setMap(map)		Задаёт карту, на которой должен отображаться оверлей. Унаследован от IOverlay .

overlay.Polyline

Расширяет [IOverlay](#).

Оверлей линии. По умолчанию оверлеи не добавлены в раскage.full (стандартного набора модулей). Для создания собственного экземпляра оверлея необходимо использовать [overlay.storage](#).

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.Polyline(geometry[], data[], options[])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelLineStringGeometry Пиксельная геометрия фигуры.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.interactive	true	Тип: Boolean Опция, позволяющая отключить реакцию объекта на dom-события.
options.opacity	—	Тип: Number Общая прозрачность.
options.separateContainer	—	Тип: Boolean Производит отрисовку в отдельном слое.
options.strokeColor	—	Тип: String Цвет линии graphics.style.color .
options.strokeOpacity	—	Тип: Number Прозрачность контура.
options.strokeStyle	—	Тип: Number[] String Стиль контура graphics.style.stroke .
options.strokeWidth	—	Тип: Number Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
datachange	<p>Изменение данных. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldData</code> - старые данные; • <code>newData</code> - новые данные. <p>Унаследовано от IOverlay.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
emptinesschange	<p>Изменение признака пустоты оверлея. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldGeometry</code> - старая пиксельная геометрия; • <code>newGeometry</code> - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> • <code>oldMap</code> - старая карта; • <code>newMap</code> - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
shapechange	Изменение фигуры области охватывающей оверлей. Экземпляр класса Event . Унаследовано от IOverlay .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных оверлея. Унаследован от IOverlay .
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay .
getMap()	Map null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
getShape()	IShape null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
isEmpty()	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
setData(data)		Задаёт данные оверлея. Унаследован от IOverlay .
setGeometry(geometry)		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
setMap(map)		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

overlay.Rectangle

Расширяет [IOverlay](#)

Справочник JavaScript API Яндекс.Карт. Версия 2.1.79

Оверлей прямоугольника. По умолчанию оверлеи не добавлены в package.full (стандартного набора

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
overlay.Rectangle(geometry[, data[, options]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: IPixelRectangleGeometry Пиксельная геометрия фигуры.
data	—	Тип: Object Данные.
options	—	Тип: Object Опции.
options.borderRadius	—	Тип: Number Радиус скругления углов.
options.fill	—	Тип: String Наличие заливки.
options.fillColor	—	Тип: String Цвет заливки graphics.style.color .
options.fillImageHref	—	Тип: String Фоновое изображение. При включении данной опции значение <code>fillColor</code> игнорируется.
options.fillMethod	'stretch'	Тип: String Тип заливки фоном. Может принимать одно из двух значений: <ul style="list-style-type: none">• <code>stretch</code> - фоновое изображение растягивается по размеру оверлея.• <code>tile</code> - Фоновое изображение повторяется без изменений размера. Аналог <code>background-repeat</code> в css. Можно использовать для заливки фигуры неким шаблоном.

Параметр	Значение по умолчанию	Описание
options.fillOpacity	—	Тип: Number Прозрачность заливки.
options.interactive	true	Тип: Boolean Опция, позволяющая отключить реакцию объекта на dom-события.
options.opacity	—	Тип: Number Общая прозрачность.
options.outline	—	Тип: String Наличие обводки.
options.separateContainer	—	Тип: Boolean Производит отрисовку в отдельном слое.
options.strokeColor	—	Тип: String Цвет линии graphics.style.color .
options.strokeOpacity	—	Тип: Number Прозрачность контура.
options.strokeStyle	—	Тип: Number[] String Стиль контура(не поддерживается в режиме Canvas) graphics.style.stroke .
options.strokeWidth	—	Тип: Number Ширина линии.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
datachange	<p>Изменение данных. Поля данных:</p> <ul style="list-style-type: none"> oldData - старые данные; newData - новые данные. <p>Унаследовано от IOverlay.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
emptinesschange	<p>Изменение признака пустоты оверлея. Экземпляр класса Event.</p> <p>Унаследовано от IOverlay.</p>
geometrychange	<p>Изменение геометрии. Поля данных:</p> <ul style="list-style-type: none"> oldGeometry - старая пиксельная геометрия; newGeometry - новая пиксельная геометрия. <p>Унаследовано от IOverlay.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IOverlay.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
shapechange	Изменение фигуры области охватывающей оверлей. Экземпляр класса Event . Унаследовано от IOverlay .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных оверлея. Унаследован от IOverlay .
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay .
getMap()	Map null	Возвращает ссылку на текущую карту. Унаследован от IOverlay .
getShape()	IShape null	Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. Унаследован от IOverlay .
isEmpty()	Boolean	Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от IOverlay .
setData(data)		Задаёт данные оверлея. Унаследован от IOverlay .
setGeometry(geometry)		Задаёт пиксельную геометрию оверлея. Унаследован от IOverlay .
setMap(map)		Задаёт карту, на которой должен отобразиться оверлей. Унаследован от IOverlay .

overlay.storage

Статический объект

Хранилище оверлеев. По умолчанию оверлеи не добавлены в `package.full` (стандартного набора модулей). При добавлении геообъекта на карту происходит загрузка необходимого оверлея. Для получения класса оверлея необходимо использовать метод [require](#) данного хранилища. По умолчанию в хранилище объявлены следующие ключи для асинхронного доступа:

- 'default#placemark' — оверлей метки-изображения [overlay.Placemark](#);
- 'default#pin' — оверлей метки в виде круга [overlay.Pin](#);
- 'default#circle' — оверлей круга [overlay.Circle](#);
- 'default#rectangle' — оверлей прямоугольника [overlay.Rectangle](#);
- 'default#polyline' — оверлей линии [overlay.Polyline](#);
- 'default#polygon' — оверлей многоугольника [overlay.Polygon](#);
- 'hotspot#placemark' — хотспотный оверлей точки [overlay.hotspot.Placemark](#);
- 'hotspot#circle' — хотспотный оверлей круга [overlay.hotspot.Circle](#);
- 'hotspot#rectangle' — хотспотный оверлей прямоугольника [overlay.hotspot.Rectangle](#);
- 'hotspot#polyline' — хотспотный оверлей линии [overlay.hotspot.Polyline](#);
- 'hotspot#polygon' — хотспотный оверлей многоугольника [overlay.hotspot.Polygon](#);
- 'html#balloon' — HTML-оверлей балуна [overlay.html.Balloon](#);
- 'html#hint' — HTML-оверлей всплывающей подсказки [overlay.html.Hint](#);
- 'html#placemark' — HTML-оверлей метки [overlay.html.Placemark](#);
- 'html#rectangle' — HTML-оверлей прямоугольника [overlay.html.Rectangle](#).

Методы

Пример:

```
ymaps.overlay.storage.require(['hotspot#circle'], function (HotspotOverlayClass) {
    // Создаем экземпляр полученного класса.
    var overlay = new HotspotOverlayClass(
        new ymaps.geometry.Circle([30, 50], 10), {}, {}
    );
});
```

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.
define(key[, depends, resolveCallback[, context]])	util.AsyncStorage	Определяет в хранилище асинхронное значение.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
isDefined(key)	Boolean	Проверка доступности ключа в хранилище.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.
require(keys[, successCallback[, errorCallback[, context]])	vow.Promise	Асинхронный запрос на получение значений из хранилища.

pane

pane.EventsPane

Расширяет [IEventPane](#).

Пейн событий.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
pane.EventsPane(map, params)
```

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Карта.
params *	—	Тип: Object Параметры.
params.css	—	Тип: Object CSS стили DOM-элемента пейна.
params.zIndex	0	Тип: Number zIndex пейна.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
overflowchange	<p>Изменение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Экземпляр IEvent.</p> <p>Унаследовано от IPane.</p>
wheel	<p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
zindexchange	<p>Изменение значения <code>zIndex</code> пейна. Экземпляр IEvent.</p> <p>Унаследовано от IPane.</p>

Методы

Имя	Возвращает	Описание
destroy()		<p>Уничтожает пейн.</p> <p>Унаследован от IPane.</p>
getElement()	<code>HTMLElement</code>	<p>Возвращает ссылку на DOM-контейнер пейна.</p> <p>Унаследован от IPane.</p>
getMap()	Map	<p>Возвращает карту, которой принадлежит пейн.</p> <p>Унаследован от IPane.</p>

Имя	Возвращает	Описание
getOverflow()	String	<p>Возвращает значение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений:</p> <ul style="list-style-type: none"> "visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым; "hidden" - область отображения содержимого пейна ограничена контейнером карты. <p>Унаследован от IPane.</p>
getZIndex()	Number	<p>Возвращает zIndex пейна.</p> <p>Унаследован от IPane.</p>

pane.MovablePane

Расширяет [IContainerPane](#).

Двигающийся пейн карты.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
pane.MovablePane(map, params)
```

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	<p>Тип: Map</p> <p>Карта.</p>
params *	—	<p>Тип: Object</p> <p>Параметры.</p>
params.css	—	<p>Тип: Object</p> <p>CSS стили DOM-элемента пейна.</p>

Параметр	Значение по умолчанию	Описание
params.margin	0	Тип: Number Дополнительный отступ от границ контейнера карты, расширяющий область отображения пейна.
params.overflow	"hidden"	Тип: String Параметр определяющий видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений: <ul style="list-style-type: none"> "visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым; "hidden" - область отображения содержимого пейна ограничена контейнером карты.
params.zIndex	0	Тип: Number zIndex пейна.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
actionbegin	Начало движения пейна. Экземпляр IEvent . Унаследовано от IContainerPane .
actionend	Окончание движения пейна. Экземпляр IEvent . Унаследовано от IContainerPane .
clientpixelschange	Изменение локальной координатной системы пейна. Данное событие означает, что объекты, рассчитывающие свою позицию внутри пейна из глобальных пиксельных координат карты, должны пересчитать ее и обновить свое положение внутри DOM-элемента пейна. Экземпляр IEvent . Унаследовано от IContainerPane .
overflowchange	Изменение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Экземпляр IEvent . Унаследовано от IPane .

Имя	Описание
viewportchange	Изменение области показа пейна. Экземпляр IEvent . Унаследовано от IContainerPane .
zindexchange	Изменение значения zIndex пейна. Экземпляр IEvent . Унаследовано от IPane .

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает пейн. Унаследован от IPane .
fromClientPixels(clientPixelPoint)	Number[]	Преобразует клиентские пиксельные координаты в глобальные. Унаследован от IPositioningContext .
getElement()	HTMLElement	Возвращает ссылку на DOM-контейнер пейна. Унаследован от IPane .
getMap()	Map	Возвращает карту, которой принадлежит пейн. Унаследован от IPane .
getOverflow()	String	Возвращает значение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений: <ul style="list-style-type: none"> "visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым; "hidden" - область отображения содержимого пейна ограничена контейнером карты. Унаследован от IPane .
getViewport()	Number[][]	Возвращает область показа пейна в клиентских координатах. Унаследован от IContainerPane .

Имя	Возвращает	Описание
getZIndex()	Number	Возвращает zIndex пейна. Унаследован от IPane .
getZoom()	Number	Возвращает текущий уровень масштабирования, в котором работает контекст позиционирования. Унаследован от IPositioningContext .
toClientPixels(globalPixelPoint)	Number[]	Преобразует глобальные пиксельные координаты в клиентские. Унаследован от IPositioningContext .

pane.StaticPane

Расширяет [IContainerPane](#).

Статичный пейн карты.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
pane.StaticPane(map, params)
```

Параметры:

Параметр	Значение по умолчанию	Описание
map *	—	Тип: Map Карта.
params *	—	Тип: Object Параметры.
params.css	—	Тип: Object CSS стили DOM-элемента пейна.
params.margin	0	Тип: Number Дополнительный отступ от границ контейнера карты, расширяющий область отображения пейна.

Параметр	Значение по умолчанию	Описание
params.overflow	"hidden"	<p>Тип: String</p> <p>Параметр определяющий видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений:</p> <ul style="list-style-type: none"> "visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым; "hidden" - область отображения содержимого пейна ограничена контейнером карты.
params.zIndex	0	<p>Тип: Number</p> <p>zIndex пейна.</p>

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	<p>Менеджер событий.</p> <p>Унаследовано от IEventEmitter.</p>

События

Имя	Описание
actionbegin	<p>Начало движения пейна. Экземпляр IEvent.</p> <p>Унаследовано от IContainerPane.</p>
actionend	<p>Окончание движения пейна. Экземпляр IEvent.</p> <p>Унаследовано от IContainerPane.</p>
clientpixelschange	<p>Изменение локальной координатной системы пейна. Данное событие означает, что объекты, рассчитывающие свою позицию внутри пейна из глобальных пиксельных координат карты, должны пересчитать ее и обновить свое положение внутри DOM-элемента пейна. Экземпляр IEvent.</p> <p>Унаследовано от IContainerPane.</p>
overflowchange	<p>Изменение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Экземпляр IEvent.</p> <p>Унаследовано от IPane.</p>
viewportchange	<p>Изменение области показа пейна. Экземпляр IEvent.</p> <p>Унаследовано от IContainerPane.</p>
zindexchange	<p>Изменение значения zIndex пейна. Экземпляр IEvent.</p> <p>Унаследовано от IPane.</p>

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает пейн. Унаследован от IPane .
fromClientPixels(clientPixelPoint)	Number[]	Преобразует клиентские пиксельные координаты в глобальные. Унаследован от IPositioningContext .
getElement()	HTMLElement	Возвращает ссылку на DOM-контейнер пейна. Унаследован от IPane .
getMap()	Map	Возвращает карту, которой принадлежит пейн. Унаследован от IPane .
getOverflow()	String	Возвращает значение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений: <ul style="list-style-type: none"> "visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым; "hidden" - область отображения содержимого пейна ограничена контейнером карты. Унаследован от IPane .
getViewport()	Number[][]	Возвращает область показа пейна в клиентских координатах. Унаследован от IContainerPane .
getZIndex()	Number	Возвращает zIndex пейна. Унаследован от IPane .

Имя	Возвращает	Описание
getZoom()	Number	Возвращает текущий уровень масштабирования, в котором работает контекст позиционирования. Унаследован от IPositioningContext .
toClientPixels(globalPixelPoint)	Number[]	Преобразует глобальные пиксельные координаты в клиентские. Унаследован от IPositioningContext .

Panorama

Примечание: Конструктор класса Panorama скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IPanorama](#).

Объект, описывающий панораму.

См.: [panorama.locate](#)

[Методы](#)

Методы

Имя	Возвращает	Описание
createPlayer(element[, options])	vow.Promise	Создает новый экземпляр класса panorama.Player и открывает в нем панораму.
getAngularBBBox()	Number[]	Возвращает сферические координаты, которые задают на панорамной сфере область, занимаемую изображением. Координаты указываются в формате [thetaTop, phiRight, thetaBottom, phiLeft] (аналогично CSS). Унаследован от IPanorama .
getConnectionArrows()	IPanoramaConnectionArrow []	Возвращает массив связей панорамы. Унаследован от IPanorama .
getConnectionMarkers()	IPanoramaConnectionMarker []	Возвращает массив маркеров-переходов на панораме. Унаследован от IPanorama .

Имя	Возвращает	Описание
getCoordSystem()	ICoordSystem	<p>Возвращает систему координат, в которой заданы позиции панорамы и всех её маркеров и связей.</p> <p>Унаследован от IPanorama.</p>
getDefaultDirection()	Number[]	<p>Возвращает направление взгляда по умолчанию. Будет использовано плеером при открытии панорамы.</p> <p>Унаследован от IPanorama.</p>
getDefaultSpan()	Number[]	<p>Возвращает размер области обзора по умолчанию. Будет использовано плеером при открытии панорамы.</p> <p>Унаследован от IPanorama.</p>
getGraph()	IPanoramaGraph null	<p>Возвращает граф панорам, связанных с текущей панорамой, для осуществления быстрых переходов.</p> <p>Унаследован от IPanorama.</p>
getLayer()	String	<p>Возвращает слой, которому принадлежит панорама.</p>
getMarkers()	IPanoramaMarker []	<p>Возвращает массив маркеров на панораме.</p> <p>Унаследован от IPanorama.</p>
getName()	String	<p>Возвращает название панорамы, отображаемое плеером в интерфейсе.</p> <p>Унаследован от IPanorama.</p>
getPosition()	Number[]	<p>Возвращает положение панорамы в указанной в опциях системе координат. Задается в формате [lon, lat, height], [lat, lon, height] или [x, y, height] в зависимости от системы и порядка координат. height – высота панорамы, заданная относительно некоторого уровня (не обязательно уровня моря).</p> <p>Унаследован от IPanorama.</p>
getTileLevels()	IPanoramaTileLevel []	<p>Возвращает массив уровней масштабирования панорамного изображения.</p> <p>Унаследован от IPanorama.</p>

Имя	Возвращает	Описание
<code>getTileSize()</code>	<code>Number[]</code>	Возвращает размер тайлов, на которые нарезано панорамное изображение. Унаследован от IPanorama .

Описание методов

`createPlayer`

```
{vow.Promise} createPlayer(element[, options])
```

Создает новый экземпляр класса [panorama.Player](#) и открывает в нем панораму.

Возвращает Promise объект, который будет разрешен экземпляром класса [panorama.Player](#) с открытой текущей панорамой.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>element *</code>	—	Тип: <code>HTMLElement String</code> Ссылка на HTML-элемент, в котором будет размещен плеер, или идентификатор этого HTML-элемента.
<code>options</code>	—	Тип: <code>Object</code> Опции.
<code>options.direction</code>	<code>'auto'</code>	Тип: <code>Number[] String</code> Направление взгляда в формате <code>[bearing, pitch]</code> , где <code>bearing</code> – азимут направления в градусах, <code>pitch</code> – угол подъема над линией горизонта в градусах. Специальное строковое значение <code>auto</code> означает, что после открытия панорамы будет применено направление, указанное в метаданных панорамы.
<code>options.span</code>	<code>'auto'</code>	Тип: <code>Number[] String</code> Угловые размеры поля обзора в формате <code>[horizontalSpan, verticalSpan]</code> , где <code>horizontalSpan</code> — горизонтальный размер поля, <code>verticalSpan</code> — вертикальный.

* Обязательный параметр/опция.

getLayer

```
{String} getLayer()
```

См.: [panorama.locate](#)

Возвращает слой, которому принадлежит панорама.

panorama**panorama.Base**

Расширяет [IPanorama](#).

[Конструктор](#) | [Методы](#)

Конструктор

```
panorama.Base()
```

Инициализирует панораму со стандартными параметрами.

Пример:

```
function Panorama () {
    ymaps.panorama.Base.call(this);
    // Проверяем, что с нашей панорамой все в порядке.
    this.validate();
}

ymaps.util.defineClass(Panorama, ymaps.panorama.Base, {
    getPosition: function () {
        // Давайте поместим нашу панораму в центр системы координат.
        return [0, 0, 0];
    },

    getCoordSystem: function () {
        return ymaps.coordSystem.cartesian;
    },

    getAngularBBox: function () {
        // Мы сделаем нашу панораму полностью сферической.
        return [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0];
    },

    getTileSize: function () {
        return [512, 512];
    },

    getTileLevels: function () {
        // Наша панорама будет иметь только одно изображение.
        return [{
            getTileUrl: function (x, y) {
                return '/' + x + '/' + y + '.jpg';
            },

            getImageSize: function () {
                return [4096, 2048];
            }
        }];
    }
});
```

Методы

Имя	Возвращает	Описание
getAngularBBox()		Обязательный к переопределению метод.
getConnectionArrows()	IPanoramaConnectionArrow[]	Возвращает пустой массив, как будто на панораме нет ни одной связи в виде стрелок.

Имя	Возвращает	Описание
getConnectionMarkers()	IPanoramaConnectionMarker[]	Возвращает пустой массив, как будто на панораме нет ни одной связи в виде маркера.
getConnections()	IPanoramaConnectionMarker[]	 Внимание: Данный метод является устаревшим. Данный метод является устаревшим . Вместо него переопределяйте метод <code>panorama.Base.getConnectionMarkers</code> .
getCoordSystem()	ICoordSystem	Возвращает географическую систему координат.
getDefaultDirection()	<code>Number[]</code>	Возвращает направление на "север" в горизонт.
getDefaultSpan()	<code>Number[]</code>	Возвращает область обзора 130 на 80 градусов в радианах.
getGraph()	<code>null</code>	Возвращает <code>null</code> , как будто на панораме нет быстрых переходов на другие панорамы (по графу).
getMarkers()	IPanoramaMarker[]	Возвращает пустой массив, как будто в панораме нет ни одного маркера.
getName()	<code>String</code>	Возвращает пустую строку.
getPosition()		Обязательный к переопределению метод.
getThoroughfares()	IPanoramaConnectionArrow[]	 Внимание: Данный метод является устаревшим. Данный метод является устаревшим . Вместо него переопределяйте метод <code>panorama.Base.getConnectionArrows</code> .
getTileLevels()		Обязательный к переопределению метод.
getTileSize()		Обязательный к переопределению метод.

Имя	Возвращает	Описание
<code>validate()</code>		<p>Проверяет консистентность и корректность данных, возвращаемых методами объекта панорамы. Условия, которые проверяет этот метод:</p> <ul style="list-style-type: none">• позиции всех объектов имеют три компоненты (включая высоту);• размер тайла должен быть степенью двойки (например, 128, 256, 512 пикселей);• панорама является полным кругом (то есть ширина угловой области должна быть 2π);• каждый уровень масштабирования панорамного изображения содержит целое число тайлов по горизонтали (по вертикали это не обязательно). <p>Если вызов этого метода генерирует ошибку для объекта панорамы, стабильная работа плеера с такой панорамой не гарантируется.</p>

Описание методов

`getAngularBBox`

```
{ } getAngularBBox()
```

Обязательный к переопределению метод.

`getConnectionArrows`

```
{ IPanoramaConnectionArrow[] } getConnectionArrows()
```

Возвращает пустой массив, как будто на панораме нет ни одной связи (в виде стрелок).

getConnectionMarkers

```
{IPanoramaConnectionMarker[]} getConnectionMarkers()
```

Возвращает пустой массив, как будто на панораме нет ни одной связи (в виде маркера).

getConnections

```
{IPanoramaConnectionMarker[]} getConnections()
```

Данный метод является **устаревшим**. Вместо него переопределяйте метод `panorama.Base.getConnectionMarkers`.

Данный метод является устаревшим.

Возвращает пустой массив, как будто в панораме нет ни одной связи.

getCoordSystem

```
{ICoordSystem} getCoordSystem()
```

Возвращает географическую систему координат.

getDefaultDirection

```
{Number[]} getDefaultDirection()
```

Возвращает направление на "север" в горизонт.

getDefaultSpan

```
{Number[]} getDefaultSpan()
```

Возвращает область обзора 130 на 80 градусов в радианах.

getGraph

```
{null} getGraph()
```

Возвращает `null`, как будто на панораме нет ни одного быстрого перехода (по графу) на другие панорамы.

getMarkers

```
{IPanoramaMarker[]} getMarkers()
```

Возвращает пустой массив, как будто в панораме нет ни одного маркера.

getName

```
{String} getName()
```

Возвращает пустую строку.

getPosition

```
{ } getPosition()
```

Обязательный к переопределению метод.

getThoroughfares

```
{IPanoramaConnectionArrow[]} getThoroughfares()
```

Данный метод является **устаревшим**. Вместо него переопределяйте метод `panorama.Base.getConnectionArrows`.

Данный метод является устаревшим.

Возвращает пустой массив, как будто в панораме нет ни одного перехода.

getTileLevels

```
{ } getTileLevels()
```

Обязательный к переопределению метод.

getTileSize

```
{ } getTileSize()
```

Обязательный к переопределению метод.

validate

```
{ } validate()
```

Проверяет консистентность и корректность данных, возвращаемых методами объекта панорамы. Условия, которые проверяет этот метод:

- позиции всех объектов имеют три компоненты (включая высоту);
- размер тайла должен быть степенью двойки (например, 128, 256, 512 пикселей);
- панорама является полным кругом (то есть ширина угловой области должна быть 2π);
- каждый уровень масштабирования панорамного изображения содержит целое число тайлов по горизонтали (по вертикали это не обязательно).

Если вызов этого метода генерирует ошибку для объекта панорамы, стабильная работа плеера с такой панорамой не гарантируется.

panorama.Base.createPanorama

Статическая функция.

Возвращает экземпляр панорамы.

```
{ IPanorama } panorama.Base.createPanorama(params)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>params</code> *	—	Тип: Object Параметры панорамы.
<code>params.angularBBox</code> *	—	Тип: Number[] Угловой

Параметр	Значение по умолчанию	Описание
<code>params.coordSystem</code>	—	Тип: ICoordSystem Координатная система, в которой задана позиция панорамы. По умолчанию будет использована coordSystem.geo .
<code>params.name</code>	"	Тип: String Название панорамы.
<code>params.position *</code>	—	Тип: Number[] Позиция панорамы.
<code>params.tileLevels *</code>	—	Тип: IPanoramaTileLevel[] Массив уровней детализации панорамного изображения.
<code>params.tileSize *</code>	—	Тип: Number[] Размер тайлов панорамного изображения.

* Обязательный параметр/опция.

Пример:

```
var player = new ymaps.panorama.Player(
    'player',
    ymaps.panorama.Base.createPanorama({
        coordSystem: ymaps.coordSystem.cartesian,
        // Давайте поместим нашу панораму в центр системы координат.
        position: [0, 0],
        name: 'My panorama',
        // Мы сделаем нашу панораму полностью сферической.
        angularBBBox: [0.5 * Math.PI, 2 * Math.PI, -0.5 * Math.PI, 0],
        tileSize: [512, 512],
        tileLevels: [{
            getTileUrl: function (x, y) {
                return '/' + x + '/' + y + '.jpg';
            },
            getImageSize: function () {
                return [4096, 2048];
            }
        }]
    })
);
```

panorama.Base.getMarkerPositionFromDirection

Статическая функция.

Рассчитывает координаты маркера на основе двух величин: по направлению взгляда на маркер и расстоянию до него. Координаты вычисляются в той системе координат, в которой задана панорама.

Возвращает положение панорамы в системе координат, заданной через опцию `coordSystem`. Задается в формате `[lon, lat, height]`, `[lat, lon, height]` или `[x, y, height]` в зависимости от системы и порядка координат. `height` — высота маркера в метрах, заданная относительно того же уровня, что и высота панорамы.

```
{ Number[] } panorama.Base.getMarkerPositionFromDirection(panorama, direction, distance)
```

Параметры:

Параметр	Значение по умолчанию	Описание
panorama *	—	Тип: IPanorama Объект, описывающий панораму.
direction *	—	Тип: Number[] Направление взгляда на маркер в формате <code>[bearing, pitch]</code> , где: <ul style="list-style-type: none"> <code>bearing</code> — направление на маркер в горизонтальной плоскости, для географической системы координат это азимут в радианах, для декартовой — угол в радианах, отсчитываемый от оси X против часовой стрелки; <code>pitch</code> — угол подъема направления на маркер в радианах.
distance *	—	Тип: Number Расстояние до маркера.

* Обязательный параметр/опция.

panorama.createPlayer

Статическая функция.

Осуществляет поиск панорамы в окрестностях заданной точки. Если найдена хоть одна панорама, создает плеер с этой панорамой.

Возвращает Promise объект, который будет разрешен экземпляром класса [panorama.Player](#) или отклонен с описанием ошибки.

```
{ vow.Promise } panorama.createPlayer(element, point[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
element *	—	Тип: HTMLElement String Ссылка на HTML-элемент, в котором будет размещен плеер, или идентификатор этого HTML-элемента.
point *	—	Тип: Number[] Точка, около которой будет произведен поиск панорамы.
options	—	Тип: Object Опции.

Параметр	Значение по умолчанию	Описание
<code>options.direction</code>	'auto'	<p>Тип: <code>Number[] String</code></p> <p>Направление взгляда в формате <code>[bearing, pitch]</code>, где <code>bearing</code> – азимут направления в градусах, <code>pitch</code> – угол подъема над линией горизонта в градусах. Специальное строковое значение <code>auto</code> означает, что после открытия панорамы будет применено направление, указанное в метаданных панорамы.</p>
<code>options.layer</code>	'yandex#panorama'	<p>Тип: <code>String</code></p> <p>Слой, в котором будет произведен поиск панорамы. Доступны два слоя:</p> <ul style="list-style-type: none"> 'yandex#panorama' — панорамы на земле; 'yandex#airPanorama' – воздушные панорамы.
<code>options.span</code>	'auto'	<p>Тип: <code>Number[] String</code></p> <p>Угловые размеры поля обзора в формате <code>[horizontalSpan, verticalSpan]</code>, где <code>horizontalSpan</code> – горизонтальный размер поля, <code>verticalSpan</code> – вертикальный.</p>

* Обязательный параметр/опция.

panorama.isSupported

Статическая функция.

Проверяет, поддерживает ли панорамный плеер платформу.

Возвращает `true`, если браузер поддерживается плеером и `false` в ином случае.

```
{ Boolean } panorama.isSupported()
```

panorama.locate

Статическая функция.

Осуществляет поиск панорамы в заданных точке и слое. Результатом запроса является массив найденных панорам, представленных в виде объектов [Panorama](#).

Возвращает `Promise` объект, который будет разрешен массивом найденных панорам (если не найдено ни одной панорамы, то массив будет пуст) или отклонен с описанием ошибки.

```
{ vow.Promise } panorama.locate(point[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>point</code> *	—	Тип: <code>Number[]</code> Точка, около которой будет произведен поиск панорамы.
<code>options</code>	—	Тип: <code>Object</code> Опции.
<code>options.layer</code>	'yandex#panorama'	Тип: <code>String</code> Слой, в котором будет произведен поиск панорамы. Доступны два слоя: <ul style="list-style-type: none"> 'yandex#panorama' — панорамы на земле; 'yandex#airPanorama' – воздушные панорамы.

* Обязательный параметр/опция.

panorama.Manager

Примечание: Конструктор класса `panorama.Manager` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IEventEmitter](#).

Менеджер привязанного к карте плеера панорам.

[Поля](#) | [События](#) | [Методы](#)

Поля

Имя	Тип	Описание
<code>events</code>	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
<code>closeplayer</code>	Плеер панорам закрыт. Экземпляр класса Event .
<code>disablelookup</code>	Режим поиска панорам выключен. Экземпляр класса Event .
<code>enablelookup</code>	Режим поиска панорам включен. Экземпляр класса Event .
<code>locate</code>	Начат поиск панорамы для точки. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>point</code> – точка, в окрестностях которой будет произведен поиск панорам; <code>options</code> – опции поиска панорам.

Имя	Описание
locatefail	Поиск панорамы завершен с ошибкой. Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> point – точка, в окрестностях которой будет произведен поиск панорам; options – опции поиска панорам. error — ошибка, произошедшая при поиске;
locatesuccess	Поиск панорамы завершен успешно. Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> point – точка, в окрестностях которой будет произведен поиск панорам; options – опции поиска панорам. result — список найденных панорам.
openplayer	Плеер панорам открыт. Экземпляр класса Event .

Методы

Имя	Возвращает	Описание
closePlayer()		Скрывает плеер панорам.
disableLookup()		Выключает режим поиска панорам на карте.
enableLookup()		Включает режим поиска панорам на карте.
getPlayer()	panorama.Player	Возвращает текущий плеер панорам или null, если плеер закрыт.
isLookupEnabled()	Boolean	Проверяет включен ли режим поиска панорам на карте.
openPlayer(panorama[, locateOptions[, options]])	vow.Promise	Открывает плеер панорам.

Описание событий

closeplayer

Плеер панорам закрыт. Экземпляр класса [Event](#).

disablelookup

Режим поиска панорам выключен. Экземпляр класса [Event](#).

enablelookup

Режим поиска панорам включен. Экземпляр класса [Event](#).

locate

Начат поиск панорамы для точки. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `point` – точка, в окрестностях которой будет произведен поиск панорам;
- `options` – опции поиска панорам.

locatefail

Поиск панорамы завершен с ошибкой. Имена полей, доступных через метод [Event.get](#):

- `point` – точка, в окрестностях которой будет произведен поиск панорам;
- `options` – опции поиска панорам.
- `error` — ошибка, произошедшая при поиске;

locatesuccess

Поиск панорамы завершен успешно. Имена полей, доступных через метод [Event.get](#):

- `point` – точка, в окрестностях которой будет произведен поиск панорам;
- `options` – опции поиска панорам.
- `result` — список найденных панорам.

openplayer

Плеер панорам открыт. Экземпляр класса [Event](#).

Описание методов**closePlayer**

```
{ } closePlayer()
```

Скрывает плеер панорам.

disableLookup

```
{ } disableLookup()
```

Выключает режим поиска панорам на карте.

enableLookup

```
{ } enableLookup()
```

Включает режим поиска панорам на карте.

getPlayer

```
{panorama.Player} getPlayer()
```

Возвращает текущий плеер панорам или `null`, если плеер закрыт.

isLookupEnabled

```
{Boolean} isLookupEnabled()
```

Проверяет включен ли режим поиска панорам на карте.

openPlayer

```
{vow.Promise} openPlayer(panorama[, locateOptions[, options]])
```

Открывает плеер панорам.

Возвращает Promise, который будет отклонен ошибкой, если панораму не удалось открыть или если открытие было отменено вызовом closePlayer.

Параметры:

Параметр	Значение по умолчанию	Описание
panorama *	—	Тип: IPanorama Number[] Панорама или координаты панорамы.
locateOptions	—	Тип: Object Опции функции panorama.locate .
options	—	Тип: Object
options.direction	'auto'	Тип: Number[] Направление взгляда на панораме.

* Обязательный параметр/опция.

panorama.Player

Расширяет [IEventEmitter](#).

Класс для создания и управления плеером панорам.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
panorama.Player(element, panorama[, options])
```

Создает экземпляр плеера панорам.

Параметры:

Параметр	Значение по умолчанию	Описание
element *	—	Тип: HTMLElement string Ссылка на HTML-элемент, в котором будет размещен плеер, или идентификатор этого HTML-элемента.
panorama *	—	Тип: IPanorama Панорама, которая будет открыта в созданном плеере.

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.autoFitToViewport</code>	"always"	Тип: String Опция, позволяющая отключить автоматическое слежение за размером контейнера плеера. По умолчанию плеер всегда следит за размером своего контейнера, перестраивая изображение, если они изменились. Доступные значения: <ul style="list-style-type: none">• <code>none</code> – не отслеживать изменения размеров контейнера.• <code>ifNull</code> – как только контейнер получит CSS значение 'display', отличное от «none», плеер автоматически подстроит свои размеры под это значение. После отслеживание будет остановлено.• <code>always</code> – всегда отслеживать изменения размеров контейнера.

Параметр	Значение по умолчанию	Описание
<code>options.controls</code>	—	<p>Тип: <code>String[]</code></p> <p>Набор контролов плеера. Доступные контролы:</p> <ul style="list-style-type: none"> <code>closeControl</code> – кнопка для закрытия плеера. <code>fullscreenControl</code> – кнопка перевода плеера в полноэкранный режим. <code>panoramaName</code> – название панорамы (автоматически скрывается в плеере маленького размера). <code>zoomControl</code> – кнопки для управления масштабом панорамы. <p>Набор контролов по умолчанию включает в себя все вышеперечисленные.</p>
<code>options.direction</code>	<code>'auto'</code>	<p>Тип: <code>Number[] String</code></p> <p>Направление взгляда в формате <code>[bearing, pitch]</code>, где <code>bearing</code> – азимут направления в градусах, <code>pitch</code> – угол подъема над линией горизонта в градусах. Специальное строковое значение <code>auto</code> означает, что после открытия панорамы будет применено направление, указанное в метаданных панорамы.</p>
<code>options.hotkeysEnabled</code>	<code>false</code>	<p>Тип: <code>Boolean</code></p> <p>Опция, включающая управление плеером с клавиатуры. Обратите внимание, что при включении плеер начинает перехватывать нажатия некоторых клавиш (например, курсорных), отменяя при это реакцию браузера по умолчанию, что может мешать пользователю взаимодействовать с вашей страницей. Поэтому по умолчанию управление с клавиатуры выключено.</p>

Параметр	Значение по умолчанию	Описание
options.scrollZoomBehavior	true	Тип: Boolean Опция, позволяющая отключить масштабирование панорамы колесом мыши. По умолчанию включено, и плеер перехватывает события колеса мыши.
options.span	'auto'	Тип: Number[] String Угловые размеры поля обзора в формате [horizontalSpan, verticalSpan], где horizontalSpan — горизонтальный размер поля, verticalSpan — вертикальный.
options.suppressMapOpenBlock	false	Тип: Boolean Нужно ли скрывать предложение открыть текущую панораму в Яндекс.Картах, максимально сохранив всю имеющуюся информацию о ней. true — скрывать, false — не скрывать. Ссылка на Яндекс.Карты отображается в левом верхнем углу плеера.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
destroy	Плеер закрыт пользователем или уничтожен с помощью метода <code>panorama.Player.destroy</code> .
directionchange	Сменилось направление взгляда.
error	В процессе работы плеера возникла ошибка. При этом пользователю будет показан соответствующий экран.
fullscreenenter	Плеер перешел в полноэкранный режим.
fullscreenexit	Плеер вышел из полноэкранного режима.
markercollapse	Пользователь кликнул по раскрытому маркеру. Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>marker</code> – маркер, который был свернут.

Имя	Описание
markerexpand	Пользователь кликнул по свернутому маркеру. Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>marker</code> – маркер, который был развернут.
markermouseenter	На маркер наведен указатель мыши. Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>marker</code> – маркер, на который был наведен указатель мыши.
markermouseleave	Пользователь сместил указатель мыши за пределы маркера. Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>marker</code> – маркер, с которого был убран указатель мыши.
panoramachange	Сменилась открытая панорама (например, в результате вызова функции <code>panorama.Player.setPanorama</code> или пользовательского действия).
spanchange	Сменился размер поля обзора.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает плеер.
fitToViewport()		Проверяет размер контейра плеера и, если он изменился с последней проверки, перестраивает изображение.
getDirection()	<code>Number[]</code>	Возвращает текущее направление обзора в формате <code>[bearing, pitch]</code> , где <code>bearing</code> — азимут направления в градусах, <code>pitch</code> — угол подъема над линией горизонта в градусах.
getPanorama()	IPanorama	Возвращает панораму, которая открыта в плеере на данный момент.
getSpan()	<code>Number[]</code>	Возвращает текущие угловые размеры поля обзора в формате <code>[horizontalSpan, verticalSpan]</code> , где <code>horizontalSpan</code> — горизонтальный размер поля в градусах, <code>verticalSpan</code> — вертикальный размер в градусах.
lookAt(point)	panorama.Player	Поворачивает обзор так, чтобы в центре поля зрения оказалась переданная точка.

Имя	Возвращает	Описание
<code>moveTo(point[, options])</code>	<code>vow.Promise</code>	Ищет панораму по заданным параметрам и открывает ее.
<code>setDirection(direction)</code>	<code>panorama.Player</code>	Устанавливает новое направление обзора.
<code>setPanorama(panorama)</code>	<code>panorama.Player</code>	Открывает в плеере переданную панораму.
<code>setSpan(span)</code>	<code>panorama.Player</code>	Устанавливает новые размеры поля обзора.

Описание событий

destroy

Плеер закрыт пользователем или уничтожен с помощью метода `panorama.Player.destroy`.

directionchange

Сменилось направление взгляда.

error

В процессе работы плеера возникла ошибка. При этом пользователю будет показан соответствующий экран.

fullscreenenter

Плеер перешел в полноэкранный режим.

fullscreenexit

Плеер вышел из полноэкранного режима.

markercollapse

Пользователь кликнул по раскрытому маркеру. Имена полей, доступных через метод [Event.get](#):

- `marker` – маркер, который был свернут.

markerexpand

Пользователь кликнул по свернутому маркеру. Имена полей, доступных через метод [Event.get](#):

- `marker` – маркер, который был развернут.

markermouseenter

На маркер наведен указатель мыши. Имена полей, доступных через метод [Event.get](#):

- `marker` – маркер, на который был наведен указатель мыши.

markermouseleave

Пользователь сместил указатель мыши за пределы маркера. Имена полей, доступных через метод [Event.get](#):

- `marker` – маркер, с которого был убран указатель мыши.

panoramachange

Сменилась открытая панорама (например, в результате вызова функции `panorama.Player.setPanorama` или пользовательского действия).

spanchange

Сменился размер поля обзора.

Описание методов**destroy**

```
{ } destroy()
```

Уничтожает плеер.

fitToViewport

```
{ } fitToViewport()
```

Проверяет размер контейра плеера и, если он изменился с последней проверки, перестраивает изображение.

getDirection

```
{Number[]} getDirection()
```

Возвращает текущее направление обзора в формате `[bearing, pitch]`, где `bearing` — азимут направления в градусах, `pitch` — угол подъема над линией горизонта в градусах.

getPanorama

```
{IPanorama} getPanorama()
```

Возвращает открытую панораму в плеере.

getSpan

```
{Number[]} getSpan()
```

Возвращает текущие угловые размеры поля обзора в формате `[horizontalSpan, verticalSpan]`, где `horizontalSpan` — горизонтальный размер поля в градусах, `verticalSpan` — вертикальный размер в градусах.

lookAt

```
{panorama.Player} lookAt(point)
```

Поворачивает обзор так, чтобы в центре поля зрения оказалась переданная точка.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>point *</code>	—	<p>Тип: <code>Number[]</code></p> <p>Точка, на которую будет повернут обзор. Может быть представлена в виде массива двух или трех координат. Первые две координаты интерпретируются как географические координаты точки. Если переданы три координаты, то третья интерпретируется как высота точки относительно панорамы в метрах.</p>

* Обязательный параметр/опция.

moveTo

```
{vow.Promise} moveTo(point[, options])
```

Ищет панораму по заданным параметрам и открывает ее.

Возвращает объект-обещание, который будет разрешен, если панорама найдена и успешно открыта в плеере, или отклонен с описанием ошибки в ином случае.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>point *</code>	—	<p>Тип: <code>Number[]</code></p> <p>Точка, около которой будет произведен поиск панорамы.</p>
<code>options</code>	—	<p>Тип: <code>Object</code></p> <p>Опции.</p>
<code>options.direction</code>	'auto'	<p>Тип: <code>Number[] String</code></p> <p>Направление взгляда в формате <code>[bearing, pitch]</code>, где <code>bearing</code> – азимут направления в градусах, <code>pitch</code> – угол подъема над линией горизонта в градусах. Специальное строковое значение <code>auto</code> означает, что после открытия панорамы будет применено направление, указанное в метаданных панорамы.</p>

Параметр	Значение по умолчанию	Описание
<code>options.layer</code>	'yandex#panorama'	Тип: String Слой, в котором будет произведен поиск панорамы. Доступны два слоя: <ul style="list-style-type: none"> 'yandex#panorama' – панорамы на земле; 'yandex#airPanorama' – воздушные панорамы.
<code>options.span</code>	'auto'	Тип: Number[] String Угловые размеры поля обзора в формате [horizontalSpan, verticalSpan], где horizontalSpan – горизонтальный размер поля, verticalSpan – вертикальный.

* Обязательный параметр/опция.

setDirection

```
{panorama.Player} setDirection(direction)
```

Устанавливает новое направление обзора.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>direction</code> *	—	Тип: Number[] String Направление взгляда в формате [bearing, pitch], где bearing – азимут направления в градусах, pitch – угол подъема над линией горизонта в градусах. Специальное строковое значение auto означает, что после открытия панорамы будет применено направление, указанное в метаданных панорамы.

* Обязательный параметр/опция.

setPanorama

```
{panorama.Player} setPanorama(panorama)
```

Открывает в плеере переданную панораму.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>panorama</code> *	—	Тип: <code>IPanorama</code> Панорама.

* Обязательный параметр/опция.

setSpan

```
{panorama.Player} setSpan(span)
```

Устанавливает новые размеры поля обзора.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>span</code> *	—	Тип: <code>Number[] String</code> Угловые размеры поля обзора в формате <code>[horizontalSpan, verticalSpan]</code> , где <code>horizontalSpan</code> — горизонтальный размер поля, <code>verticalSpan</code> — вертикальный.

* Обязательный параметр/опция.

Placemark

Расширяет `GeoObject`.

Метка. Представляет собой геообъект с геометрией `geometry.Point`.

См.: `GeoObject` `geometry.Point`

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Placemark(geometry[, properties[, options]])
```

Создает экземпляр метки.

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: <code>Number[] Object IPointGeometry</code> Координаты метки или хэш с описанием геометрии, или ссылка на объект точечной геометрии.

Параметр	Значение по умолчанию	Описание
properties	—	<p>Тип: Object IDataManager</p> <p>Данные метки. Могут задаваться как экземпляр класса, реализующего интерфейс IDataManager, либо в виде хэша. При значениях опций по умолчанию геообъектом трактуются следующие поля данных:</p> <ul style="list-style-type: none">• <code>iconContent</code> — содержимое иконки геообъекта;• <code>iconCaption</code> — подпись иконки геообъекта;• <code>hintContent</code> — содержимое всплывающей подсказки геообъекта;• <code>balloonContent</code> — содержимое балуна геообъекта;• <code>balloonContentHeader</code> — содержимое заголовка балуна геообъекта;• <code>balloonContentBody</code> — содержимое основной части балуна геообъекта;• <code>balloonContentFooter</code> — содержимое нижней части балуна геообъекта. <p>Поле <code>balloonContent</code> является кратким обозначением для поля <code>balloonContentBody</code>, но при одновременном задании <code>balloonContentBody</code> более приоритетен. Также вы можете дополнить данные метки своими собственными полями и использовать их везде, где это возможно. Например в макете метки или макете балуна.</p>

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	<p>Тип: Object</p> <p>Опции метки.</p> <p>Для изменения стиля и цвета иконки используются следующие опции:</p> <ul style="list-style-type: none"> <code>preset</code> — ключ предустановленных стилей метки. Список доступных ключей содержится в описании option.presetStorage. Опцию <code>preset</code> можно задавать совместно с опцией <code>iconColor</code>, только если она принимает одно из следующих значений: <ul style="list-style-type: none"> <code>'islands#icon'</code> <code>'islands#dotIcon'</code> <code>'islands#circleIcon'</code> <code>'islands#circleDotIcon'</code> <code>iconColor</code> — цвет иконки метки. Можно задавать в любом формате, допустимом в CSS (например, по названию или в формате RGB). Эта опция применяется для стандартных иконок в браузерах, поддерживающих SVG. <p>Примечание: В IE8 данная опция не работает.</p> <p>Если требуется задать собственный макет иконки, то необходимо указать следующие опции:</p> <ul style="list-style-type: none"> <code>iconLayout</code> — макет иконки. Тип: конструктор объекта с интерфейсом <code>ILayout</code> или его ключ в хранилище. Список доступных макетов: <ul style="list-style-type: none"> <code>'default#image'</code> — пользовательское изображение иконки; <code>'default#imageWithContent'</code> — пользовательское изображение иконки с содержимым; дополнительные опции классов <code>"layout.Image"</code> и <code>"layout.ImageWithContent"</code> с префиксом <code>icon</code>. <p>Примечание: Для задания макета тени иконки используется тот же набор опций, но с префиксом <code>iconShadow</code>. Например, <code>iconShadowLayout</code>.</p> <p>Также через класс <code>Placemark</code> можно задать опции отдельных объектов:</p> <ul style="list-style-type: none"> опции бабуна метки с префиксом <code>balloon</code>; Опции всплывающей подсказки метки с префиксом <code>hint</code>.

Параметр	Значение по умолчанию	Описание
<code>options.cursor</code>	"pointer"	Тип: String Вид курсора над меткой.
<code>options.draggable</code>	false	Тип: Boolean Определяет возможность перетаскивания метки.
<code>options.hasBalloon</code>	true	Тип: Boolean Определяет наличие поля balloon у метки.
<code>options.hasHint</code>	true	Тип: Boolean Определяет наличие поля hint у метки.
<code>options.hideIconOnBalloonOpen</code>	true	Тип: Boolean Скрывать метку при открытии балуна.
<code>options.iconOffset</code>	—	Тип: Number[] Пиксельное смещение иконки относительно заданной позиции.
<code>options.iconShape</code>	—	Тип: <code>IGeometryJson</code> null Фигура активной области метки. Задается в виде JSON-описания пиксельной геометрии иконки. Эту опцию нужно использовать при создании своих HTML макетов. Координаты геометрии фигуры отсчитываются от точки привязки.
<code>options.interactiveZIndex</code>	true	Тип: Boolean Включает режим автоматического изменения z-index метки в зависимости от ее состояния.
<code>options.interactivityModel</code>	"default#geoObject"	Тип: String Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
<code>options.openBalloonOnClick</code>	true	Тип: Boolean Определяет, показывать ли балун при щелчке на метке.

Параметр	Значение по умолчанию	Описание
<code>options.openEmptyBalloon</code>	false	<p>Тип: Boolean</p> <p>Определяет, показывать ли пустой балун при щелчке на метке.</p>
<code>options.openEmptyHint</code>	false	<p>Тип: Boolean</p> <p>Определяет, показывать ли пустой хинт при наведении указателя мыши на метку.</p>
<code>options.openHintOnHover</code>	true	<p>Тип: Boolean</p> <p>Определяет, показывать ли хинт при наведении указателя мыши на метку.</p>
<code>options.pane</code>	"places"	<p>Тип: String</p> <p>Ключ пейна, в который помещается оверлей метки.</p>
<code>options.pointOverlay</code>	"default#placemark"	<p>Тип: String Function</p> <p>Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра:</p> <ul style="list-style-type: none"> <code>geometry</code>: IPixelPointGeometry — непосредственно пиксельная геометрия; <code>data</code>: IDataManager или Object - данные оверлея; <code>options</code>: Object - опции оверлея. <p>Функция возвращает объект vow.Promise.</p>
<code>options.syncOverlayInit</code>	false	<p>Тип: Boolean</p> <p>Включает синхронное добавление оверлея на карту. По умолчанию добавление оверлея осуществляется асинхронно, что позволяет предотвратить зависания браузера при добавлении на карту большого числа геообъектов. Однако, асинхронное добавление не позволяет получать доступ к оверлею сразу после добавления метки на карту.</p>

Параметр	Значение по умолчанию	Описание
options.useMapMarginInDragging	true	Тип: Boolean При перетаскивании объекта к краю карты происходит автоматическое изменение центра карты. Нужно ли учитывать отступы карты при автоматическом смещении центра карты map.margin.Manager .
options.visible	true	Тип: Boolean Определяет видимость метки.
options.zIndex	—	Тип: Number z-index иконки метки в обычном состоянии. Наименее приоритетный.
options.zIndexActive	—	Тип: Number z-index иконки метки с открытым балуном. Наиболее приоритетный.
options.zIndexDrag	—	Тип: Number z-index иконки метки при перетаскивании.
options.zIndexHover	—	Тип: Number z-index иконки метки при наведении на него указателя мыши.

* Обязательный параметр/опция.

Примеры:

1.

```
// Создадим метку.
var placemark = new ymaps.Placemark([55.75, 37.61], {
  balloonContent: '',
  iconContent: "Азербайджан"}
, {
  preset: "islands#yellowStretchyIcon",
  // Отключаем кнопку закрытия балуна.
  balloonCloseButton: false,
  // Балун будем открывать и закрывать кликом по иконке метки.
  hideIconOnBalloonOpen: false
});
geoMap.geoObjects.add(placemark);
```

2.

```
var placemark = new ymaps.Placemark([55.75, 37.61], {}, {
  // Задаем стиль метки (метка в виде круга).
  preset: "islands#circleDotIcon",
  // Задаем цвет метки (в формате RGB).
  iconColor: '#ff0000'
});
geoMap.geoObjects.add(placemark);
```

Поля

Имя	Тип	Описание
balloon	geoObject.Balloon	Балун геообъекта. Унаследовано от GeoObject .
editor	geometryEditor.Point	Редактор геометрии "Точка".
events	event.Manager	Менеджер событий. Унаследовано от GeoObject .
geometry	geometry.Point	Геометрия типа "Точка".
hint	geoObject.Hint	Хинт геообъекта. Унаследовано от GeoObject .
options	option.Manager	Менеджер опций геообъекта. Унаследовано от GeoObject .
properties	data.Manager	Менеджер данных геообъекта. Унаследовано от GeoObject .
state	data.Manager	Состояние геообъекта. Определяется следующими полями: <ul style="list-style-type: none"> • <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. • <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; • <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается. Унаследовано от GeoObject .

События

Имя	Описание
balloonclose	Закрытие балуна. Экземпляр класса Event . Унаследовано от GeoObject .
balloonopen	Открытие балуна на геообъекте. Экземпляр класса Event . Унаследовано от GeoObject .

Имя	Описание
beforedrag	<p>Событие, предшествующее событию drag. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p> <p>Унаследовано от GeoObject.</p>
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p> <p>Унаследовано от GeoObject.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent - оригинальное событие редактора геометрии. <p>Унаследовано от GeoObject.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Закрытие хинта. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>

Имя	Описание
hintopen	Открытие хинта на геообъекте. Экземпляр класса Event . Унаследовано от GeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mousemove	Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent . Унаследовано от IDomEventEmitter .

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. <p>Унаследован от IChildOnMap.</p>
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. <p>Унаследован от IChildOnMap.</p>

Описание полей

editor

```
{geometryEditor.Point} editor
```

Редактор геометрии "Точка".

geometry

```
{geometry.Point} geometry
```

Геометрия типа "Точка".

Polygon

Расширяет [GeoObject](#).

Многоугольник. Представляет собой геообъект с геометрией [geometry.Polygon](#).

См.: [GeoObject](#) [geometry.Polygon](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Polygon(geometry[, properties[, options]])
```

Создает экземпляр многоугольника.

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	Тип: <code>Number[] Object IPolygonGeometry</code> Координаты вершин ломаных, определяющих внешнюю и внутренние границы многоугольника, хэш-объект с параметрами геометрии или ссылка на объект геометрии. Внутренняя граница может отсутствовать.

Параметр	Значение по умолчанию	Описание
properties	—	<p>Тип: Object IDataManager</p> <p>Данные многоугольника. Могут задаваться как экземпляр класса, реализующего интерфейс IDataManager, либо в виде хэша. При значениях опций по умолчанию геообъектом трактуются следующие поля данных:</p> <ul style="list-style-type: none">• <code>hintContent</code> - содержимое всплывающей подсказки многоугольника;• <code>balloonContent</code> - содержимое балуна многоугольника;• <code>balloonContentHeader</code> - содержимое заголовка балуна многоугольника;• <code>balloonContentBody</code> - содержимое основной части балуна многоугольника;• <code>balloonContentFooter</code> - содержимое нижней части балуна многоугольника. <p>Поле <code>balloonContent</code> является кратким обозначением для поля <code>balloonContentBody</code>, но при одновременном задании <code>balloonContentBody</code> более приоритетен. Также вы можете дополнить данные многоугольника своими собственными полями и использовать их везде, где это возможно. Например в макете балуна многоугольника.</p>

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции многоугольника. Используя этот параметр можно задавать как опции самого многоугольника, так и опции его составных частей:</p> <ul style="list-style-type: none"> • опции балуна многоугольника с префиксом <code>balloon</code>; • Опции всплывающей подсказки многоугольника с префиксом <code>hint</code>. • Опции редактора геометрии многоугольника с префиксом <code>editor</code>. Смотрите описание класса geometryEditor.Polygon. • Опции геометрии могут задаваться без префикса. Смотрите описание класса IGeometry геометрии geometry.Polygon.
options.cursor	"pointer"	<p>Тип: String</p> <p>Вид курсора над многоугольником.</p>
options.draggable	false	<p>Тип: Boolean</p> <p>Определяет возможность перетаскивания многоугольника.</p>
options.fill	true	<p>Тип: Boolean</p> <p>Наличие заливки фигуры.</p>
options.fillColor	"0066ff99"	<p>Тип: String</p> <p>Цвет заливки.</p>
options.fillImageHref	—	<p>Тип: String</p> <p>Фоновое изображение. При включении данной опции в режиме заливки stretch значение опции <code>fillColor</code> игнорируется.</p>

Параметр	Значение по умолчанию	Описание
options.fillMethod	'stretch'	<p>Тип: String</p> <p>Тип заливки фоном. Может принимать одно из двух значений:</p> <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - Фоновое изображение повторяется без изменений размера. Аналог background-repeat в css. Можно использовать для заливки фигуры неким шаблоном.
options.fillOpacity	1	<p>Тип: Number</p> <p>Прозрачность заливки.</p>
options.hasBalloon	true	<p>Тип: Boolean</p> <p>Определяет наличие поля balloon у многоугольника.</p>
options.hasHint	true	<p>Тип: Boolean</p> <p>Определяет наличие поля hint у многоугольника.</p>
options.interactiveZIndex	false	<p>Тип: Boolean</p> <p>Включает режим автоматического изменения z-index полигона в зависимости от его состояния.</p>
options.interactivityModel	"default#geoObject"	<p>Тип: String</p> <p>Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage.</p>
options.opacity	1	<p>Тип: Number</p> <p>Прозрачность.</p>
options.openBalloonOnClick	true	<p>Тип: Boolean</p> <p>Определяет, показывать ли балун при щелчке на многоугольнике.</p>

Параметр	Значение по умолчанию	Описание
options.openEmptyBalloon	false	Тип: Boolean Определяет показывать ли пустой балун при щелчке на многоугольнике.
options.openEmptyHint	false	Тип: Boolean Определяет, показывать ли хинт при наведении указателя мыши на многоугольник.
options.openHintOnHover	true	Тип: Boolean Определяет, показывать ли хинт при наведении указателя мыши на многоугольник.
options.outline	true	Тип: Boolean Наличие обводки фигуры.
options.pane	"areas"	Тип: String Ключ пейна, в который помещается оверлей многоугольника.
options.polygonOverlay	"default#polygon"	Тип: String Function Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра: <ul style="list-style-type: none"> • geometry: IPixelPolygonGeometry - непосредственно пиксельная геометрия; • data: IDataManager или Object - данные оверлея; • options: Object - опции оверлея. Функция возвращает объект vow.Promise .
options.strokeColor	"0066ffff"	Тип: String String[] Цвет линии или обводки. Можно задать несколько значений для множественной обводки.

Параметр	Значение по умолчанию	Описание
options.strokeOpacity	1	<p>Тип: Number Number[]</p> <p>Прозрачность линии или обводки. Можно задать несколько значений для множественной обводки.</p>
options.strokeStyle	—	<p>Тип: String Object String[] Object[]</p> <p>Стиль линии или обводки. Доступные стили перечислены в объекте graphics.style.stroke.</p>
options.strokeWidth	1	<p>Тип: Number Number[]</p> <p>Толщина линии или обводки. Можно задать несколько значений для множественной обводки.</p>
options.syncOverlayInit	false	<p>Тип: Boolean</p> <p>Включает синхронное добавление оверлея на карту. По умолчанию добавление оверлея осуществляется асинхронно, что позволяет предотвратить зависания браузера при добавлении на карту большого числа геообъектов. Однако, асинхронное добавление не позволяет получать доступ к оверлею сразу после добавления многоугольника на карту.</p>
options.useMapMarginInDragging	true	<p>Тип: Boolean</p> <p>При перетаскивании объекта к краю карты происходит автоматическое изменение центра карты. Нужно ли учитывать отступы карты при автоматическом смещении центра карты map.margin.Manager.</p>
options.visible	true	<p>Тип: Boolean</p> <p>Определяет видимость многоугольника.</p>
options.zIndex	—	<p>Тип: Number</p> <p>z-index многоугольника в обычном состоянии. Наименее приоритетный.</p>
options.zIndexActive	—	<p>Тип: Number</p> <p>z-index многоугольника с открытым балуном. Наиболее приоритетный.</p>

Параметр	Значение по умолчанию	Описание
options.zIndexDrag	—	Тип: Number z-index многоугольника при перетаскивании.
options.zIndexHover	—	Тип: Number z-index многоугольника при наведении на него указателя мыши.

* Обязательный параметр/опция.

Пример:

```
var polygon = new ymaps.Polygon([
  // Координаты внешнего контура.
  [[-80, 60], [-90, 50], [-60, 40], [-80, 60]],
  // Координаты внутреннего контура.
  [[-90, 80], [-90, 30], [-20, 40], [-90, 80]]
], {
  hintContent: "Многоугольник"
}, {
  fillColor: '#6699ff',
  // Делаем полигон прозрачным для событий карты.
  interactivityModel: 'default#transparent',
  strokeWidth: 8,
  opacity: 0.5
});
myMap.geoObjects.add(polygon);
myMap.setBounds(polygon.geometry.getBounds());
```

Поля

Имя	Тип	Описание
balloon	geoObject.Balloon	Балун геообъекта. Унаследовано от GeoObject .
editor	geometryEditor.Polygon	Редактор геометрии "Многоугольник".
events	event.Manager	Менеджер событий. Унаследовано от GeoObject .
geometry	geometry.Polygon	Геометрия типа "Многоугольник".
hint	geoObject.Hint	Хинт геообъекта. Унаследовано от GeoObject .
options	option.Manager	Менеджер опций геообъекта. Унаследовано от GeoObject .
properties	data.Manager	Менеджер данных геообъекта. Унаследовано от GeoObject .

Имя	Тип	Описание
state	data.Manager	<p>Состояние геообъекта. Определяется следующими полями:</p> <ul style="list-style-type: none"> • <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. • <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; • <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается. <p>Унаследовано от GeoObject.</p>

События

Имя	Описание
balloonclose	<p>Закрытие балуна. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
balloonopen	<p>Открытие балуна на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
beforedrag	<p>Событие, предшествующее событию <code>drag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> • <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p> <p>Унаследовано от GeoObject.</p>

Имя	Описание
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p> <p>Унаследовано от GeoObject.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>

Имя	Описание
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent - оригинальное событие редактора геометрии. <p>Унаследовано от GeoObject.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Закрытие хинта. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
hintopen	<p>Открытие хинта на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .

Имя	Возвращает	Описание
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей

editor

```
{geometryEditor.Polygon} editor
```

Редактор геометрии "Многоугольник".

geometry

```
{geometry.Polygon} geometry
```

Геометрия типа "Многоугольник".

Polyline

Расширяет [GeoObject](#).

Ломаная линия. Представляет собой геообъект с геометрией [geometry.LineString](#).

См.: [GeoObject](#) [geometry.LineString](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Polyline(geometry[, properties[, options]])
```

Создает экземпляр ломаной.

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	<p>Тип: Number[] Object ILineStringGeometry</p> <p>Координаты вершин, хэш-объект с параметрами геометрии или ссылка на объект геометрии ломаной линии.</p>
properties	—	<p>Тип: Object IDataManager</p> <p>Данные ломаной. Могут задаваться как экземпляр класса, реализующего интерфейс IDataManager, либо в виде хэша. При значениях опций по умолчанию геообъектом трактуются следующие поля данных:</p> <ul style="list-style-type: none"> • <code>hintContent</code> - содержимое всплывающей подсказки ломаной; • <code>balloonContent</code> - содержимое балуна ломаной; • <code>balloonContentHeader</code> - содержимое заголовка балуна ломаной; • <code>balloonContentBody</code> - содержимое основной части балуна ломаной; • <code>balloonContentFooter</code> - содержимое нижней части балуна ломаной. <p>Поле <code>balloonContent</code> является кратким обозначением для поля <code>balloonContentBody</code>, но при одновременном задании <code>balloonContentBody</code> более приоритетен. Также вы можете дополнить данные ломаной своими собственными полями и использовать их, например, в макете хинта.</p>

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции ломаной. Используя этот параметр можно задавать как опции самой ломаной, так и опции ее составных частей:</p> <ul style="list-style-type: none"> • опции балуна линии с префиксом balloon; • Опции всплывающей подсказки линии с префиксом hint. • Опции редактора геометрии линии с префиксом editor. Смотрите описание класса geometryEditor.LineString. • Опции геометрии могут задаваться без префикса. Смотрите описание класса IGeometry геометрии geometry.LineString.
options.cursor	"pointer"	<p>Тип: String</p> <p>Вид курсора над ломаной.</p>
options.draggable	false	<p>Тип: Boolean</p> <p>Определяет возможность перетаскивания ломаной.</p>
options.hasBalloon	true	<p>Тип: Boolean</p> <p>Определяет наличие поля balloon у ломаной.</p>
options.hasHint	true	<p>Тип: Boolean</p> <p>Определяет наличие поля hint у ломаной.</p>
options.interactiveZIndex	false	<p>Тип: Boolean</p> <p>Включает режим автоматического изменения z-index ломаной в зависимости от ее состояния.</p>

Параметр	Значение по умолчанию	Описание
options.interactivityModel	"default#geoObject"	<p>Тип: String</p> <p>Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage.</p>
options.lineStringOverlay	"default#polyline"	<p>Тип: String Function</p> <p>Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра:</p> <ul style="list-style-type: none"> • geometry:IPixelLineStringGeometry - непосредственно пиксельная геометрия; • data: IDataManager или Object - данные оверлея; • options: Object - опции оверлея. <p>Функция возвращает объект vow.Promise.</p>
options.opacity	1	<p>Тип: Number</p> <p>Прозрачность.</p>
options.openBalloonOnClick	true	<p>Тип: Boolean</p> <p>Определяет, показывать ли балун при щелчке на ломаной.</p>
options.openEmptyBalloon	false	<p>Тип: Boolean</p> <p>Определяет, показывать ли пустой балун при щелчке на ломаной.</p>
options.openEmptyHint	false	<p>Тип: Boolean</p> <p>Определяет, показывать ли хинт при наведении указателя мыши на ломаную.</p>
options.openHintOnHover	true	<p>Тип: Boolean</p> <p>Определяет, показывать ли хинт при наведении указателя мыши на ломаную.</p>

Параметр	Значение по умолчанию	Описание
options.pane	"areas"	Тип: IPane String Ключ пейна, в который помещается оверлей ломаной.
options.strokeColor	"0066ffff"	Тип: String String[] Цвет линии или обводки. Можно задать несколько значений для множественной обводки.
options.strokeOpacity	1	Тип: Number Number[] Прозрачность линии или обводки. Можно задать несколько значений для множественной обводки.
options.strokeStyle	—	Тип: String Object String[] Object[] Стиль линии или обводки. Доступные стили перечислены в объекте graphics.style.stroke .
options.strokeWidth	1	Тип: Number Number[] Толщина линии или обводки. Можно задать несколько значений для множественной обводки.
options.syncOverlayInit	false	Тип: Boolean Включает синхронное добавление оверлея на карту. По умолчанию добавление оверлея осуществляется асинхронно, что позволяет предотвратить зависания браузера при добавлении на карту большого числа ломаных. Однако, асинхронное добавление не позволяет получить доступ к оверлею сразу после добавления ломаной на карту.
options.useMapMarginInDragging	true	Тип: Boolean При перетаскивании объекта к краю карты происходит автоматическое изменение центра карты. Нужно ли учитывать отступы карты при автоматическом смещении центра карты map.margin.Manager .
options.visible	true	Тип: Boolean Определяет видимость ломаной.

Параметр	Значение по умолчанию	Описание
options.zIndex	—	Тип: Number z-index ломаной в обычном состоянии. Наименее приоритетный.
options.zIndexActive	—	Тип: Number z-index ломаной с открытым балуном. Наиболее приоритетный.
options.zIndexDrag	—	Тип: Number z-index ломаной при перетаскивании.
options.zIndexHover	—	Тип: Number z-index ломаной при наведении на него указателя мыши.

* Обязательный параметр/опция.

Пример:

```
// Создадим ломаную.
var polyline = new ymaps.Polyline([
  [-80, 60], [-90, 50], [-60, 40], [-80, 60]
], {
  hintContent: "Ломаная"
}, {
  draggable: true,
  strokeColor: '#000000',
  strokeWidth: 4,
  // Первой цифрой задаем длину штриха. Второй – длину разрыва.
  strokeStyle: '1 5'
});
// Добавляем линию на карту.
myMap.geoObjects.add(polyline);
// Устанавливаем карте границы линии.
myMap.setBounds(polyline.geometry.getBounds());
```

Поля

Имя	Тип	Описание
balloon	geoObject.Balloon	Балун геообъекта. Унаследовано от GeoObject .
editor	geometryEditor.LineString	Редактор геометрии "Ломаная линия".
events	event.Manager	Менеджер событий. Унаследовано от GeoObject .
geometry	geometry.LineString	Геометрия типа "Ломаная линия".
hint	geoObject.Hint	Хинт геообъекта. Унаследовано от GeoObject .
options	option.Manager	Менеджер опций геообъекта. Унаследовано от GeoObject .
properties	data.Manager	Менеджер данных геообъекта. Унаследовано от GeoObject .

Имя	Тип	Описание
state	data.Manager	<p>Состояние геообъекта. Определяется следующими полями:</p> <ul style="list-style-type: none"> • <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. • <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; • <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается. <p>Унаследовано от GeoObject.</p>

События

Имя	Описание
balloonclose	<p>Заккрытие балуна. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
balloonopen	<p>Открытие балуна на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
beforedrag	<p>Событие, предшествующее событию <code>drag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> • <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p> <p>Унаследовано от GeoObject.</p>

Имя	Описание
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p> <p>Унаследовано от GeoObject.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>

Имя	Описание
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent - оригинальное событие редактора геометрии. <p>Унаследовано от GeoObject.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Закрытие хинта. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
hintopen	<p>Открытие хинта на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .

Имя	Возвращает	Описание
<code>getParent()</code>	<code>IParentOnMap</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание полей

editor

```
{geometryEditor.LineString} editor
```

Редактор геометрии "Ломаная линия".

geometry

```
{geometry.LineString} geometry
```

Геометрия типа "Ломаная линия".

Рорир

Расширяет `IPopup`.

Класс для создания инфо-объекта.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Рорир(map[, options])
```

Инфо-объект.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>map</code> *	—	Тип: <code>Map</code> Ссылка на карту.

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции.
options.closeTimeout	700	Тип: Number Задержка закрытия (в мс).
options.interactivityModel	—	Тип: String Ключ модели интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage .
options.openTimeout	150	Тип: Number Задержка открытия (в мс).
options.pane	—	Тип: IPane String Пейн инфо-объекта. Список доступных по умолчанию пейнов смотрите в map.pane.Manager .
options.projection	—	Тип: IProjection Проекция координат в глобальные пиксели.
options.zIndex	—	Тип: String z-index инфо-объекта.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

События

Имя	Описание
close	Закрытие инфо-объекта. Унаследовано от IPopup .
open	Открытие инфо-объекта. Унаследовано от IPopup .

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .

Методы

Имя	Возвращает	Описание
close([force])	vow.Promise	Закрывает инфо-объект. Унаследован от IPopup .
getData()		Возвращает данные инфо-объекта. Унаследован от IPopup .
getOverlay()	vow.Promise	Возвращает объект-обещание вернуть оверлей. Унаследован от IPopup .
getOverlaySync()	IOverlay	Возвращает оверлей, если тот существует. Унаследован от IPopup .
getPosition()		Возвращает координаты инфо-объекта. Унаследован от IPopup .
isOpen()	Boolean	Возвращает состояние инфо-объекта: открыт/закрыт. Унаследован от IPopup .
open([position[, data]])	vow.Promise	Открывает инфо-объект в указанной позиции. Если инфо-объект уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией IProjection , которая содержится в опциях. Унаследован от IPopup .
setData(data)	vow.Promise	Задаёт инфо-объекту новые данные. Унаследован от IPopup .
setPosition(position)	vow.Promise	Задаёт инфо-объекту новую позицию. Унаследован от IPopup .

projection

projection.Cartesian

Расширяет [IProjection](#).

Декартова проекция прямоугольной области. Учитывает параметр coordorder загрузки API — при значении 'latlong' в массиве координат точки на первом месте должен стоять y, а на втором x.

[Конструктор](#) | [Методы](#)**Конструктор**

```
projection.Cartesian(bounds[, cycled[, scale]])
```

Создает проекцию прямоугольной координатной области в пиксели. Размер области в пикселях всегда NxN, где $N = 256 * 2^{\text{zoom}}$.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>bounds</code> *	—	Тип: <code>Number[]</code> Массив из двух точек - координат левого нижнего и правого верхнего углов прямоугольной координатной области.
<code>cycled</code>	<code>[false, false]</code>	Тип: <code>Boolean[]</code> Массив из признаков зацикленности карты по x и y.
<code>scale</code>	1	Тип: <code>Number Number[]</code> Масштаб одного деления на оси. Может быть числом или парой чисел для каждой из осей.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
<code>fromGlobalPixels(globalPixelPoint, zoom)</code>	<code>Number[]</code>	Преобразует пиксельные координаты на указанном уровне масштабирования в координаты проекции. Унаследован от IProjection .
<code>getCoordSystem()</code>	<code>ICoordSystem</code>	Возвращает координатную систему, используемую проекцией. Унаследован от IProjection .
<code>isCycled()</code>	<code>Boolean[]</code>	Показатель зацикленности проекции. Унаследован от IProjection .
<code>toGlobalPixels(coordPoint, zoom)</code>	<code>Number[]</code>	Преобразует координаты проекции в глобальные пиксельные на указанном уровне масштабирования. Унаследован от IProjection .

projection.sphericalMercator

Статический объект.

Экземпляр класса [IProjection](#)

Меркаторовская проекция на сферу. Используется многими картографическими сервисами, в частности, OpenStreetMap.

[Методы](#)

Пример:

```
// Создадим карту в сферической проекции Меркатора.
var map = new ymaps.Map('YMapsID', {
  center: [55, 37],
  zoom: 6
}, {
  projection: ymaps.projection.sphericalMercator
});
map.layers.add(new ymaps.Layer('http://tile.openstreetmap.org/%z/%x/%y.png'));
```

Методы

Имя	Возвращает	Описание
fromGlobalPixels(globalPixelPoint, zoom)	Number[]	Преобразует пиксельные координаты на указанном уровне масштабирования в координаты проекции.
getCoordSystem()	ICoordSystem	Возвращает координатную систему, используемую проекцией.
isCycled()	Boolean[]	Показатель зацикленности проекции.
toGlobalPixels(coordPoint, zoom)	Number[]	Преобразует координаты проекции в глобальные пиксельные на указанном уровне масштабирования.

projection.wgs84Mercator

Статический объект.

Экземпляр класса [IProjection](#)

Меркаторовская проекция на референсный эллипсоид wgs84. Используется Яндекс.Картами по умолчанию.

[Методы](#)

Методы

Имя	Возвращает	Описание
fromGlobalPixels(globalPixelPoint, zoom)	Number[]	Преобразует пиксельные координаты на указанном уровне масштабирования в координаты проекции.
getCoordSystem()	ICoordSystem	Возвращает координатную систему, используемую проекцией.
isCycled()	Boolean[]	Показатель зацикленности проекции.

Имя	Возвращает	Описание
<code>toGlobalPixels(coordPoint, zoom)</code>	<code>Number[]</code>	Преобразует координаты проекции в глобальные пиксельные на указанном уровне масштабирования.

ready

Статическая функция.

Выполняет переданную функцию, когда API и DOM готовы к использованию.

Возвращает Promise объект, который подтверждается пространством имен API, либо отклоняется, если при загрузке произошла ошибка.

```
{ vow.Promise } ready([successCallback[, errorCallback[, context]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>successCallback</code>	—	<p>Тип: <code>Function Object</code></p> <p>Функция, которая будет вызвана при успешной загрузке и инициализации API и DOM, или объект с параметрами, если используется расширенный синтаксис.</p> <p>Доступный список параметров:</p> <ul style="list-style-type: none"> <code>require</code> - массив дополнительных модулей, которые должны быть загружены вместе с API; <code>successCallback</code> — функция, которая будет вызвана при успешной загрузке API; <code>errorCallback</code> — функция, которая будет вызвана в случае ошибки; <code>context</code> — контекст исполнения функций. <p>Все параметры являются необязательными.</p> <p>В <code>successCallback</code> будет передано пространство имен API.</p>

Параметр	Значение по умолчанию	Описание
errorCallback	—	Тип: Function Функция, которая будет вызвана если произошла ошибка при инициализации. В функцию будет передана ошибка.
context	—	Тип: Object Контекст исполнения функции.

Примеры:

1.

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <script src="http://api-maps.yandex.ru/2.1/?apikey=your API key&lang=ru_RU" type="text/javascript"></script>
  <script type="text/javascript">
    ymaps.ready(function () {
      var map = new ymaps.Map('map', {
        center: [55.7, 37.6],
        zoom: 10
      });
      // ...
    });
  </script>
</head>
<body>
  <div id="map" style="width: 500px; height: 500px;"></div>
</body>
</html>
```

2.

```
// Пример использования расширенного синтаксиса функции ready.
ymaps.ready({
  // successCallback будет вызван, когда загрузятся API и модуль "myModule1".
  require: ['myModule1'],
  successCallback: function (ym) {
    var map = new ymaps.Map('map', {
      center: [55.7, 37.6],
      zoom: 10
    });
    var obj = new ymaps.myModule1();
    // ...
  })
})
```

Rectangle

Расширяет [GeoObject](#).

Прямоугольник. Представляет собой геообъект с геометрией [geometry.Rectangle](#).

См.: [GeoObject geometry.Rectangle](#)

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
Rectangle(geometry[, properties[, options]])
```

Создает экземпляр прямоугольника.

Параметры:

Параметр	Значение по умолчанию	Описание
geometry *	—	<p>Тип: <code>Number[] Object IRectangleGeometry</code></p> <p>Координаты двух противоположных углов, хэш-объект с параметрами геометрии или ссылка на объект геометрии прямоугольника.</p>
properties	—	<p>Тип: <code>Object IDataManager</code></p> <p>Данные прямоугольника. Могут задаваться как экземпляр класса, реализующего интерфейс <code>IDataManager</code>, либо в виде хэша. При значениях опций по умолчанию геообъектом трактуются следующие поля данных:</p> <ul style="list-style-type: none"> • <code>hintContent</code> - содержимое всплывающей подсказки прямоугольника; • <code>balloonContent</code> - содержимое балуна прямоугольника; • <code>balloonContentHeader</code> - содержимое заголовка балуна прямоугольника; • <code>balloonContentBody</code> - содержимое основной части балуна прямоугольника; • <code>balloonContentFooter</code> - содержимое нижней части балуна прямоугольника. <p>Поле <code>balloonContent</code> является кратким обозначением для поля <code>balloonContentBody</code>, но при одновременном задании <code>balloonContentBody</code> более приоритетен. Также вы можете дополнить данные прямоугольника своими собственными полями и использовать их, например, в макете балуна.</p>

Параметр	Значение по умолчанию	Описание
options	—	<p>Тип: Object</p> <p>Опции прямоугольника. Используя этот параметр можно задавать как опции самого прямоугольника, так и опции его составных частей:</p> <ul style="list-style-type: none"> • опции балуна прямоугольника с префиксом <code>balloon</code>; • Опции всплывающей подсказки прямоугольника с префиксом <code>hint</code>. • Опции геометрии могут задаваться без префикса. Смотрите описание класса IGeometry геометрии geometry.Rectangle.
options.cursor	"pointer"	<p>Тип: String</p> <p>Вид курсора над прямоугольником.</p>
options.draggable	false	<p>Тип: Boolean</p> <p>Определяет возможность перетаскивания прямоугольника.</p>
options.fill	true	<p>Тип: Boolean</p> <p>Наличие заливки фигуры.</p>
options.fillColor	"0066ff99"	<p>Тип: String</p> <p>Цвет заливки.</p>
options.fillImageHref	—	<p>Тип: String</p> <p>Фоновое изображение. При включении данной опции в режиме заливки stretch значение опции <code>fillColor</code> игнорируется.</p>

Параметр	Значение по умолчанию	Описание
options.fillMethod	'stretch'	<p>Тип: String</p> <p>Тип заливки фоном. Может принимать одно из двух значений:</p> <ul style="list-style-type: none"> stretch - фоновое изображение растягивается по размеру оверлея. tile - Фоновое изображение повторяется без изменений размера. Аналог background-repeat в css. Можно использовать для заливки фигуры неким шаблоном.
options.fillOpacity	1	<p>Тип: Number</p> <p>Прозрачность заливки.</p>
options.hasBalloon	true	<p>Тип: Boolean</p> <p>Определяет наличие поля balloon у прямоугольника.</p>
options.hasHint	true	<p>Тип: Boolean</p> <p>Определяет наличие поля hint у прямоугольника.</p>
options.interactiveZIndex	false	<p>Тип: Boolean</p> <p>Включает режим автоматического изменения z-index прямоугольника в зависимости от его состояния.</p>
options.interactivityModel	"default#geoObject"	<p>Тип: String</p> <p>Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage.</p>
options.opacity	1	<p>Тип: Number</p> <p>Прозрачность.</p>
options.openBalloonOnClick	true	<p>Тип: Boolean</p> <p>Определяет, показывать ли балун при щелчке на прямоугольнике.</p>

Параметр	Значение по умолчанию	Описание
options.openEmptyBalloon	false	Тип: Boolean Определяет, показывать ли пустой балун при щелчке на прямоугольнике.
options.openEmptyHint	false	Тип: Boolean Определяет, показывать ли хинт при наведении указателя мыши на прямоугольник.
options.openHintOnHover	true	Тип: Boolean Определяет, показывать ли хинт при наведении указателя мыши на прямоугольник.
options.outline	true	Тип: Boolean Наличие обводки фигуры.
options.pane	"places"	Тип: String Ключ пейна, в который помещается оверлей прямоугольника.
options.rectangleOverlay	"default#rectangle"	Тип: String Function Ключ-идентификатор из хранилища overlay.storage или класс оверлея. Функция-генератор принимает три параметра: <ul style="list-style-type: none"> geometry: IPixelCircleGeometry - непосредственно пиксельная геометрия; data: IDataManager или Object - данные оверлея; options: Object - опции оверлея. Функция возвращает объект vow.Promise .
options.strokeColor	"0066ffff"	Тип: String String[] Цвет линии или обводки. Можно задать несколько значений для множественной обводки.

Параметр	Значение по умолчанию	Описание
<code>options.strokeOpacity</code>	1	<p>Тип: Number Number[]</p> <p>Прозрачность линии или обводки. Можно задать несколько значений для множественной обводки.</p>
<code>options.strokeStyle</code>	—	<p>Тип: String Object String[] Object[]</p> <p>Стиль линии или обводки. Доступные стили перечислены в объекте graphics.style.stroke.</p>
<code>options.strokeWidth</code>	1	<p>Тип: Number Number[]</p> <p>Толщина линии или обводки. Можно задать несколько значений для множественной обводки.</p>
<code>options.syncOverlayInit</code>	false	<p>Тип: Boolean</p> <p>Включает синхронное добавление оверлея на карту. По умолчанию добавление оверлея осуществляется асинхронно, что позволяет предотвратить зависания браузера при добавлении на карту большого числа геообъектов. Однако, асинхронное добавление не позволяет получать доступ к оверлею сразу после добавления прямоугольника на карту.</p>
<code>options.useMapMarginInDragging</code>	true	<p>Тип: Boolean</p> <p>При перетаскивании объекта к краю карты происходит автоматическое изменение центра карты. Нужно ли учитывать отступы карты при автоматическом смещении центра карты map.margin.Manager.</p>
<code>options.visible</code>	true	<p>Тип: Boolean</p> <p>Определяет видимость прямоугольника.</p>
<code>options.zIndex</code>	—	<p>Тип: Number</p> <p>z-index прямоугольника в обычном состоянии. Наименее приоритетный.</p>
<code>options.zIndexActive</code>	—	<p>Тип: Number</p> <p>z-index прямоугольника с открытым балуном. Наиболее приоритетный.</p>

Параметр	Значение по умолчанию	Описание
options.zIndexDrag	—	Тип: Number z-index прямоугольника при перетаскивании.
options.zIndexHover	—	Тип: Number z-index прямоугольника при наведении на него указателя мыши.

* Обязательный параметр/опция.

Пример:

```
// Создание геодезической окружности радиусом 1000 километров.
var circle = new ymaps.Circle([[50, 50], 1000000], {}, {
  draggable: true
});
// Добавление круга на карту.
myMap.geoObjects.add(circle);

// Создание прямоугольника на основе границ круга.
var rectangle = new ymaps.Rectangle(circle.geometry.getBounds(), {}, {
  fill: false,
  coordRendering: "boundsPath",
  strokeWidth: 4
});
// Добавление прямоугольника на карту.
myMap.geoObjects.add(rectangle);

// Обновление координат прямоугольника при изменении геометрии окружности.
circle.geometry.events.add("change", function (event) {
  this.geometry.setCoordinates(event.get("target").getBounds());
}, rectangle);
```

Поля

Имя	Тип	Описание
balloon	geoObject.Balloon	Балун геообъекта. Унаследовано от GeoObject .
editor	null	Редактор геометрии "Прямоугольник" на данный момент не реализован.
events	event.Manager	Менеджер событий. Унаследовано от GeoObject .
geometry	geometry.Rectangle	Геометрия типа "Прямоугольник".
hint	geoObject.Hint	Хинт геообъекта. Унаследовано от GeoObject .
options	option.Manager	Менеджер опций геообъекта. Унаследовано от GeoObject .
properties	data.Manager	Менеджер данных геообъекта. Унаследовано от GeoObject .

Имя	Тип	Описание
state	data.Manager	<p>Состояние геообъекта. Определяется следующими полями:</p> <ul style="list-style-type: none"> • <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. • <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; • <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается. <p>Унаследовано от GeoObject.</p>

События

Имя	Описание
balloonclose	<p>Заккрытие балуна. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
balloonopen	<p>Открытие балуна на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
beforedrag	<p>Событие, предшествующее событию <code>drag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> • <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p> <p>Унаследовано от GeoObject.</p>

Имя	Описание
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p> <p>Унаследовано от GeoObject.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>

Имя	Описание
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent - оригинальное событие редактора геометрии. <p>Унаследовано от GeoObject.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Закрытие хинта. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
hintopen	<p>Открытие хинта на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .

Имя	Возвращает	Описание
<code>getParent()</code>	<code>IParentOnMap</code> null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от <code>IChildOnMap</code> .
<code>setParent(parent)</code>	<code>IChildOnMap</code>	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от <code>IChildOnMap</code> .

Описание полей

editor

```
{null} editor
```

Редактор геометрии "Прямоугольник" на данный момент не реализован.

geometry

```
{geometry.Rectangle} geometry
```

Геометрия типа "Прямоугольник".

regions

regions.load



Внимание: Данная функция является устаревшей. Используйте `borders.load`.

Статическая функция.

Предоставляет доступ к геометрии различных регионов и стран мира.

Возвращает объект-promise.

```
{ vow.Promise } regions.load(region[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>region *</code>	—	<p>Тип: String</p> <p>ISO_3166-1 код страны (RU, UA, BY, KZ) для загрузки административно-территориального деления, либо '001' для загрузки геометрии границ стран мира.</p>
<code>options</code>	—	<p>Тип: Object</p> <p>Опции отображения.</p>
<code>options.disputedBorders</code>	—	<p>Тип: String</p> <p>Двухбуквенный код государства, официальная позиция которого будет использоваться для определения административного подчинения спорных территорий. Допустимые значения: 'RU', 'UA', 'BY', 'KZ'. По умолчанию совпадает с кодом страны, который указан при загрузке API. Неподдерживаемые коды стран приводятся к RU. Для региона '001' (границы стран мира) поддерживается код 'UN' — мировые границы согласно позиции ООН.</p>
<code>options.lang</code>	—	<p>Тип: String</p> <p>Язык (ru, uk, en, be).</p>
<code>options.quality</code>	1	<p>Тип: Number</p> <p>Уровень качества. Доступные значения:</p> <ul style="list-style-type: none"> 0 - минимальное качество; 1 - стандартное качество; 2 - повышенное качество; 3 - высокое качество. <p>Уровень качества влияет на точность представления кривых и объем файла данных.</p>

* Обязательный параметр/опция.

Пример:

```
ymaps.regions.load('RU', {
  lang: 'en'
}).then(function (result) {
  geoMap.geoObjects.add(result.geoObjects);
});
```

RemoteObjectManager

Расширяет [ICustomizable](#), [IEventEmitter](#), [IGeoObject](#), [IParentOnMap](#).

Менеджер объектов, осуществляющий их оптимальную подгрузку с сервера. Менеджер отправляет запросы за данными по указанному url в JSONP-формате. Описание формата соответствует формату добавляемых в `ObjectManager` объектов (см. [ObjectManager.add](#)). Также поддерживаются объекты типа 'Cluster', содержащие поля:

- `type` — тип объекта, для кластеров всегда 'Cluster';
- `id` - уникальный идентификатор кластера;
- `geometry` - геометрия кластера в JSON-формате;
- `features` - массив объектов в составе кластера. Необязательное поле;
- `bbox` — массив координат, описывающих прямоугольную область, содержащую все объекты в составе кластера;
- `number` - количество объектов в кластере;
- `properties` - данные кластера.

Данный модуль рассчитан на загрузку и отображение данных, предварительно обработанных на сервере. В частности, рекомендуется использовать модуль для отображения результатов серверной кластеризации. Данные перезапрашиваются заново при изменении коэффициента масштабирования карты. Модуль не осуществляет кластеризацию объектов или фильтрацию по области видимости на клиенте. Чтобы кластеризовать объекты после их загрузки на сторону клиента, используйте [LoadingObjectManager](#). Обратите внимание, что у объектов, отрисованных на карте через данный менеджер, нельзя включать режимы редактирования и перетаскивания.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
RemoteObjectManager(urlTemplate[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
urlTemplate *	—	Тип: String шаблон URL данных. Поддерживаются специальные конструкции по аналогии с Layer . Также поддерживаются подстановки: <ul style="list-style-type: none">• <code>%b</code> заменяется на массив географических координат, описывающих прямоугольную область, для которой требуется загрузить данные.• <code>%t</code> заменяется на массив номеров тайлов, описывающих прямоугольную область, для которой требуется загрузить данные.

Параметр	Значение по умолчанию	Описание
options	—	Тип: Object Опции. <ul style="list-style-type: none">• Можно задавать все опции, указанные в описании Clusterer, за исключением опций hasBalloon и hasHint.• Опции для кластеров задаются с префиксом cluster. Список опций указан в описании класса ClusterPlacemark;• Опции для одиночных объектов задаются с префиксом geoObject. Список опций определен в классе GeoObject. Обратите внимание, менеджер не учитывает опцию 'visible'.
options.loadTileSize	256	Тип: Number Размер тайла для загрузки данных.
options.paddingParamName	'callback'	Тип: Boolean Имя GET-параметра, который содержит значение jsonp-колбека.

Параметр	Значение по умолчанию	Описание
<code>options.paddingTemplate</code>	null	<p>Тип: String</p> <p>Шаблон для jsonp-колбека. Поддерживает те же подстановки, что и <code>urlTemplate</code>. Все символы, не являющиеся буквой или цифрой, будут заменены на <code>'_'</code>. Если параметр не задан, то имя jsonp-колбека будет сгенерировано автоматически. Примеры преобразований при <code>tileNumber=[3, 1], zoom=9</code>:</p> <ul style="list-style-type: none"> <code>'myCallback=%x' => 'myCallback_3'</code> <code>'%c' => 'x_3_y_1_z_9'</code> <code>'callback2_%c' => 'callback2_x_3_y_1_z_9'</code> <code>'callback%test' => 'callback_test'</code> <code>'callback_%b' => 'callback_85_0841__180_0000_85_0841_180_0000'</code> <p>Обратите внимание, что если не использовать в значении опции подстановки, то это может привести к ошибке. Все запросы будут обращаться к одной callback-функции.</p>
<code>options.splitRequests</code>	false	<p>Тип: Boolean</p> <p>Разделять запросы за данными на запросы за одиночными тайлами. По умолчанию запросы делаются за данными для прямоугольной области, содержащей несколько тайлов.</p>
<code>options.syncOverlayInit</code>	false	<p>Тип: Boolean</p> <p>Флаг, разрешающий создавать оверлеи для объектов синхронно. Обратите внимание, что при синхронном создании оверлея нужно самостоятельно обеспечить загрузку нужного класса, реализующего интерфейс <code>IOverlay</code>. По умолчанию оверлеи создаются асинхронно, при этом класс оверлея загружается по требованию.</p>

* Обязательный параметр/опция.

Примеры:

1.

```

var objectManager = new ymaps.RemoteObjectManager('http://myServer.com/tile?bbox=%b', {
  // Опции кластеров задаются с префиксом 'cluster'.
  clusterHasBalloon: false,
  // Опции геообъектов задаются с префиксом 'geoObject'.
  geoObjectOpenBalloonOnClick: false
});

// Опции можно задавать напрямую в дочерние коллекции.
objectManager.clusters.options.set({
  preset: 'islands#grayClusterIcons',
  hintContentLayout: ymaps.templateLayoutFactory.createClass('Группа объектов')
});
objectManager.objects.options.set('preset', 'islands#grayIcon');

```

2.

Пример ответа для RemoteObjectManager.

```

jsonp_callback({
  // Ответ содержит поля error и data. Если возникает ошибка, то поле "error"
  // содержит код ошибки или описание.
  error: null,
  data: {
    type: 'FeatureCollection',
    features: [
      {
        type: 'Feature',
        geometry: {
          type: 'Point',
          coordinates: [55, 35]
        },
        id: 23,
        properties: {
          balloonContent: 'Placemark balloon content',
          iconContent: 'Placemark content'
        },
        options: {
          preset: 'islands#yellowIcon'
        }
      },
      {
        type: 'Cluster',
        id: 24,
        bbox: [[35, 46], [46, 57]],
        number: 34,
        // Массив, описывающий 34 объекта в кластере.
        // Необязательное поле;
        // Если этот параметр опущен, то при щелчке по кластеру открывается пустой балун.
        features: [{
          type: 'Feature',
          id: 512,
          properties: {
            balloonContent: 'Placemark balloon content',
            clusterCaption: 'Placemark title in the cluster balloon'
          },
          // ...
        }],
        geometry: {
          type: 'Point',
          coordinates: [40.5, 51]
        },
        properties: {
          iconContent: 34
        }
      }
    ]
  }
});

```

Поля

Имя	Тип	Описание
clusters	objectManager.ClusterCollection	Коллекция кластеров, сгенерированных менеджером.
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
objects	objectManager.ObjectCollection	Коллекция объектов, добавленных в слой.

Имя	Тип	Описание
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dataloaderror	Во время загрузки данных произошла ошибка. Экземпляр класса Event .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getBounds()	Number[][] null	Вычисляет границы области в геокоординатах, охватывающей все загруженные объекты в составе менеджера.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getObjectState(id)	Object	Получение информации о текущем состоянии объекта, добавленного в менеджер.

Имя	Возвращает	Описание
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getPixelBounds()	Number[][] null	Вычисляет границы области в глобальных пиксельных координатах, охватывающей все загруженные объекты в составе менеджера.
getTileUrl()	String null	Возвращает URL тайла с данными.
getUrlTemplate()	String	Возвращает URL шаблона данных.
reloadData()		Метод, удаляющий все загруженные ранее данные и отправляющий запрос за новыми данными.
setFilter(filterFunction)		Устанавливает функцию-фильтр для объектов. Фильтрует как одиночные объекты, так и кластеры.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .
setUrlTemplate(urlTemplate)		

Описание полей

clusters

```
{objectManager.ClusterCollection} clusters
```

Коллекция кластеров, сгенерированных менеджером.

Пример:

```
objectManager.objects.events.add('click', function (e) {  
    var objectId = e.get('objectId');  
    objectManager.objects.balloon.open(objectId);  
});
```

objects

```
{objectManager.ObjectCollection} objects
```

Коллекция объектов, добавленных в слой.

Пример:

```
objectManager.objects.events.add('click', function (e) {    var objectId = e.get('objectId');  
    objectManager.objects.balloon.open(objectId);});
```

Описание событий

dataloadererror

Во время загрузки данных произошла ошибка. Экземпляр класса [Event](#).

Описание методов

getBounds

```
{Number[][]|null} getBounds()
```

Вычисляет границы области в геокоординатах, охватывающей все загруженные объекты в составе менеджера.

Возвращает массив координат области или null, если менеджер не добавлен на карту.

getObjectState

```
{Object} getObjectState(id)
```

Получение информации о текущем состоянии объекта, добавленного в менеджер.

Возвращает объект, содержащий следующие поля:

- found - признак, указывающий, существует ли объект с переданным идентификатором в загруженных данных. Type: Boolean.
- isShown - признак, указывающий, находится ли объект в видимой области карты. Type: Boolean.
- isFilteredOut - признак, указывающий, прошел ли объект фильтрацию. Если фильтр не задан или объект прошел фильтрацию, значение поля будет false. Type: Boolean.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>id *</code>	—	Тип: Object Идентификатор объекта, для которого нужно получить состояние.

* Обязательный параметр/опция.

Пример:

```
remoteObjectManager.setFilter('properties.type == "shop"');  
// ...  
if (!remoteObjectManager.getObjectState(7).isFilteredOut) {  
    remoteObjectManager.objects.balloon.open(7);  
}
```

getPixelBounds

```
{Number[][]|null} getPixelBounds()
```

Вычисляет границы области в глобальных пиксельных координатах, охватывающей все загруженные объекты в составе менеджера.

Возвращает массив координат области или null, если менеджер не добавлен на карту.

getTileUrl

```
{String|null} getTileUrl()
```

Возвращает URL тайла с данными.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>parameters *</code>	—	Тип:

* Обязательный параметр/опция.

Пример:

```
var objectManager = new ymaps.RemoteObjectManager('http://myServer.com/tile?bbox=%b');  
objectManager.getTileUrl = function (parameters) {  
    var boundingBox = parameters.boundingBox.join('~');  
    return this.getUrlTemplate().replace(/%b/g, boundingBox);  
};
```

getUrlTemplate

```
{String} getUrlTemplate()
```

Возвращает URL шаблона данных.

reloadData

```
{ } reloadData()
```

Метод, удаляющий все загруженные ранее данные и отправляющий запрос за новыми данными.

setFilter

```
{ } setFilter(filterFunction)
```

Устанавливает функцию-фильтр для объектов. Фильтрует как одиночные объекты, так и кластеры.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>filterFunction</code> *	—	<p>Тип: Function String</p> <p>функция-фильтр. Получает на вход единственный объект, добавленный в ObjectManager. Если функция возвращает true, объект будет обработан. Если false - объект будет исключен из дальнейшей обработки. Также в качестве фильтра можно передавать строку. В строке-фильтре доступны следующие ключевые слова:</p> <ul style="list-style-type: none">• options – обращение к опциям объекта;• properties – обращение к данным объекта;• geometry – обращение к геометрии объекта;• id – обращение к идентификатору объекта. <p>В качестве фильтра можно указывать выражение, возвращающее true или false.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
// Выбрать кластеры с id > 100.  
objectManager.setFilter('object.type == "Cluster" && id > 100');
```

2.

```
// На карте будут отображаться только объекты с заданными типами.  
objectManager.setFilter('properties.type == "кафе" || properties.type == "аптека");
```

3.

```
// Можно задавать функцию-фильтр.  
objectManager.setFilter(function (object) {  
    return object.properties.name !== 'Тот, кого нельзя показать.';});
```

setUrlTemplate

```
{ } setUrlTemplate(urlTemplate)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>urlTemplate</code> *	—	Тип: String шаблон URL данных.

* Обязательный параметр/опция.

route

Статическая функция.

Прокладывает маршрут через заданные точки.

Возвращает Promise объект, который подтверждается объектом маршрута `router.Route`, либо объектом мультимаршрута `multiRouter.MultiRoute`, в зависимости от значения параметра `multiRoute`. Promise отклоняется при возникновении ошибки.

```
{ vow.Promise } route(points[, params])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>points</code> *	—	Тип: Object[] Массив точек, через которые должен проходить маршрут. Каждая точка может быть задана строкой, содержащей адрес, массивом координат, и JSON-объектом со следующими полями: <ul style="list-style-type: none">• <code>type</code>: String - тип точки. Значение <code>'wayPoint'</code> задает путевую точку маршрута. Значение <code>'viaPoint'</code> задает транзитную точку, т.е. точку, через которую нужно проехать без остановки.• <code>point</code>: Number[] String - массив координат точки, либо ее адрес в виде строки.
<code>params</code>	—	Тип: Object Параметры маршрутизации.

Параметр	Значение по умолчанию	Описание
params.avoidTrafficJams	false	<p>Тип: Boolean</p> <p>Позволяет прокладывать мультимаршрут с учетом информации о пробках. При использовании опции учитывайте, что объезд пробок не всегда возможен.</p>
params.boundedBy	—	<p>Тип: Number[][]</p> <p>Область на карте, где предположительно находятся искомые объекты. Используется, если точки маршрута заданы почтовым адресом, а не координатами.</p>
params.mapStateAutoApply	false	<p>Тип: Boolean</p> <p>Флаг, позволяющий автоматически установить центр и коэффициент масштабирования карты так, чтобы построенный маршрут был виден целиком.</p>
params.multiRoute	false	<p>Тип: Boolean</p> <p>Позволяет строить мультимаршруты.</p>
params.reverseGeocoding	false	<p>Тип: Boolean</p> <p>Использовать ли обратное геокодирование для точек маршрута, заданных координатами.</p>
params.routingMode	"auto"	<p>Тип: String</p> <p>Тип маршрутизации. Может принимать одно из трех строковых значений:</p> <ul style="list-style-type: none"> "auto" — автомобильная маршрутизация; "masstransit" — маршрутизация с использованием общественного транспорта. Доступна только для мультимаршрутов (опция multiRoute должна быть выставлена в true); "pedestrian" — пешеходная маршрутизация. Доступна только для мультимаршрутов (опция multiRoute должна быть выставлена в true);

Параметр	Значение по умолчанию	Описание
params.searchCoordOrder	—	Тип: String Определяет, каким образом нужно интерпретировать координаты в запросе. Используется, если точки маршрута заданы координатами, а не почтовым адресом.
params.strictBounds	false	Тип: Boolean Искать только внутри области, заданной опцией <code>boundedBy</code> .
params.useMapMargin	true	Тип: Boolean Нужно ли учитывать отступы карты map.margin.Manager .
params.viaIndexes	[]	Тип: Integer[] Индексы транзитных точек мультимаршрута.
params.zoomMargin	0	Тип: Number Number[] Отступы от границ видимой области карты при изменении коэффициента масштабирования. Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы <code>top</code> , <code>right</code> , <code>bottom</code> , <code>left</code> . При включенном параметре <code>"useMapMargin"</code> значение <code>"zoomMargin"</code> складывается со значениями, которые были рассчитаны в менеджере отступов map.margin.Manager .

* Обязательный параметр/опция.

Примеры:

1.

```
// Строим маршрут из Королева в Красногорск через Химки и Мытищи,
// где Мытищи - транзитная точка. Установим координаты для Красногорска.
ymaps.route([
  'Королев',
  { type: 'viaPoint', point: 'Мытищи' },
  'Химки',
  { type: 'wayPoint', point: [55.811511, 37.312518] }
], {
  mapStateAutoApply: true
}).then(function (route) {
  route.getPaths().options.set({
    // балун показывает только информацию о времени в пути с трафиком
    balloonContentLayout: ymaps.templateLayoutFactory.createClass('{{ properties.humanJamsTime }}'),
    // вы можете настроить внешний вид маршрута
    strokeColor: '0000ffff',
    opacity: 0.9
  });
  // добавляем маршрут на карту
  map.geoObjects.add(route);
```

```
});
```

2.

```
// Построим мультимаршрут и добавим его на карту с применением автомасштабирования.  
ymaps.route(['Южное Бутово', 'Москва, метро Парк Культуры'], {  
  multiRoute: true  
}).done(function (route) {  
  route.options.set("mapStateAutoApply", true);  
  myMap.geoObjects.add(route);  
}, function (err) {  
  throw err;  
}, this);
```

router

router.addon

router.addon.editor
router.addon.editor.get

Статическая функция.

Возвращает редактор маршрутов.

```
{ router.Route } router.addon.editor.get()
```

Пример:

```
ymaps.router.addon.editor.get(myMap)
```

router.Editor

Примечание: Конструктор класса `router.Editor` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [ICustomizable](#), [IEventEmitter](#).

Редактор маршрутов. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [События](#) | [Методы](#)

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .

Имя	Тип	Описание
<code>state</code>	<code>IDataManager</code>	<p>Менеджер состояния редактора маршрута.</p> <p>Поля данных, доступные посредством методов <code>get</code> и <code>set</code>:</p> <ul style="list-style-type: none"> <code>routeloading</code>: Boolean - признак того, что в данный момент идет загрузка данных из сервиса маршрутизации. <code>waypointsdrag</code>: Boolean - признак того, что в данный момент происходит перетаскивание путевой точки. <code>viapointsdrag</code>: Boolean - признак того, что в данный момент происходит перетаскивание транзитной точки.

События

Имя	Описание
<code>optionschange</code>	<p>Изменение в опциях объекта.</p> <p>Унаследовано от <code>ICustomizable</code>.</p>
<code>routeupdate</code>	Обновление маршрута. Через значение флага <code>e.get('rough')</code> можно определить событие бросилось после завершения редактирования или во время. Если вы хотите в своем приложении обновлять информацию, связанную с маршрутом, то нужно делать проверку <code>e.get('rough') == false</code> , чтобы избежать частой обработки этого события.
<code>start</code>	Включение редактора.
<code>stop</code>	Выключение редактора.
<code>viapointadd</code>	Добавление транзитной точки. Добавляемую транзитную точку можно получить через <code>e.get('viaPoint')</code>
<code>viapointdragend</code>	Завершение перетаскивания транзитной точки. Путевую точку можно получить через <code>e.get('viaPoint')</code>
<code>viapointdragstart</code>	Начало перетаскивания транзитной точки. Путевую точку можно получить через <code>e.get('viaPoint')</code>
<code>viapointremove</code>	Удаление транзитной точки. Удаляемую транзитную точку можно получить через <code>e.get('viaPoint')</code>
<code>waypointadd</code>	Добавление путевой точки. Добавляемую путевую точку можно получить через <code>e.get('wayPoint')</code>
<code>waypointdragend</code>	Завершение перетаскивания путевой точки. Путевую точку можно получить через <code>e.get('wayPoint')</code>
<code>waypointdragstart</code>	Начало перетаскивания путевой точки. Путевую точку можно получить через <code>e.get('wayPoint')</code>
<code>waypointremove</code>	Удаление путевой точки. Удаляемую путевую точку можно получить через <code>e.get('wayPoint')</code>

Методы

Имя	Описание
<code>start([options])</code>	Включает редактор маршрута.
<code>stop()</code>	Выключает редактор маршрута.

Описание полей

state

```
{IDataManager} state
```

Менеджер состояния редактора маршрута.

Поля данных, доступные посредством методов `get` и `set`:

- `routeloading`: Boolean - признак того, что в данный момент идет загрузка данных из сервиса маршрутизации.
- `waypointsdrag`: Boolean - признак того, что в данный момент происходит перетаскивание путевой точки.
- `viapointsdrag`: Boolean - признак того, что в данный момент происходит перетаскивание транзитной точки.

Описание событий

routeupdate

Обновление маршрута. Через значение флага `e.get('rough')` можно определить событие бросилось после завершения редактирования или во время. Если вы хотите в своем приложении обновлять информацию, связанную с маршрутом, то нужно делать проверку `e.get('rough') == false`, чтобы избежать частой обработки этого события.

start

Включение редактора.

stop

Выключение редактора.

viapointadd

Добавление транзитной точки. Добавляемую транзитную точку можно получить через `e.get('viaPoint')`

viapointdragend

Завершение перетаскивания транзитной точки. Путевую точку можно получить через `e.get('viaPoint')`

viapointdragstart

Начало перетаскивания транзитной точки. Путевую точку можно получить через `e.get('viaPoint')`

viapointremove

Удаление транзитной точки. Удаляемую транзитную точку можно получить через `e.get('viaPoint')`

waypointadd

Добавление путевой точки. Добавляемую путевую точку можно получить через `e.get('wayPoint')`

waypointdragend

Завершение перетаскивания путевой точки. Путевую точку можно получить через `e.get('wayPoint')`

waypointdragstart

Начало перетаскивания путевой точки. Путевую точку можно получить через `e.get('wayPoint')`

waypointremove

Удаление путевой точки. Удаляемую путевую точку можно получить через `e.get('wayPoint')`

Описание методов**start**

```
{ } start([options])
```

Включает редактор маршрута.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	Тип: Object Опции.
<code>options.addViaPoints</code>	true	Тип: Boolean true - добавление транзитных точек разрешено, false - запрещено.
<code>options.addWayPoints</code>	false	Тип: Boolean true - добавление путевых точек при клике на карте разрешено, false - запрещено.
<code>options.editViaPoints</code>	true	Тип: Boolean true - редактирование (перемещение) транзитных точек разрешено, false - запрещено.
<code>options.editWayPoints</code>	true	Тип: Boolean true - редактирование (перемещение) путевых точек разрешено, false - запрещено.

Параметр	Значение по умолчанию	Описание
options.removeViaPoints	true	Тип: Boolean true - удаление транзитных точек двойным кликом разрешено, false - запрещено.
options.removeWayPoints	false	Тип: Boolean true - удаление путевых точек двойным кликом разрешено, false - запрещено.

stop

```
{ } stop()
```

Выключает редактор маршрута.

router.Path

Примечание: Конструктор класса `router.Path` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [GeoObject](#).

Объект, описывающий часть (путь) маршрута. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию. Маршрут может содержать несколько путей, каждый путь соединяет две путевые точки.

См.: [route](#)

[Поля](#) | [События](#) | [Методы](#)

Поля

Имя	Тип	Описание
balloon	geoObject.Balloon	Балун геообъекта. Унаследовано от GeoObject .
editor	IGeometryEditor	Редактор геометрии геообъекта. Унаследовано от GeoObject .
events	event.Manager	Менеджер событий. Унаследовано от GeoObject .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от GeoObject .
hint	geoObject.Hint	Хинт геообъекта. Унаследовано от GeoObject .
options	option.Manager	Менеджер опций геообъекта. Унаследовано от GeoObject .
properties	data.Manager	Менеджер данных геообъекта. Унаследовано от GeoObject .

Имя	Тип	Описание
state	data.Manager	<p>Состояние геообъекта. Определяется следующими полями:</p> <ul style="list-style-type: none"> • <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. • <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; • <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается. <p>Унаследовано от GeoObject.</p>

События

Имя	Описание
balloonclose	<p>Заккрытие балуна. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
balloonopen	<p>Открытие балуна на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
beforedrag	<p>Событие, предшествующее событию <code>drag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> • <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p> <p>Унаследовано от GeoObject.</p>

Имя	Описание
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p> <p>Унаследовано от GeoObject.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>

Имя	Описание
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code> - оригинальное событие редактора геометрии. <p>Унаследовано от GeoObject.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Закрытие хинта. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
hintopen	<p>Открытие хинта на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getHumanJamsTime()	String	Возвращает строковое представление времени проезда пути с единицами измерения с учетом пробок.
getHumanLength()	String	Возвращает строковое представление длины пути с единицами измерения.
getHumanTime()	String	Возвращает строковое представление времени проезда пути с единицами измерения.
getJamsTime()	Integer	Возвращает время проезда пути в секундах с учетом пробок.
getLength()	Number	Возвращает длину пути в метрах.

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getSegments()	router.Segment[]	Возвращает сегменты пути.
getTime()	Integer	Возвращает время проезда пути в секундах.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание методов

getHumanJamsTime

```
{String} getHumanJamsTime()
```

Возвращает строковое представление времени проезда пути с единицами измерения с учетом пробок.

getHumanLength

```
{String} getHumanLength()
```

Возвращает строковое представление длины пути с единицами измерения.

getHumanTime

```
{String} getHumanTime()
```

Возвращает строковое представление времени проезда пути с единицами измерения.

getJamsTime

```
{Integer} getJamsTime()
```

Возвращает время проезда пути в секундах с учетом пробок.

getLength

```
{Number} getLength()
```

Возвращает длину пути в метрах.

getSegments

```
{router.Segment[]} getSegments()
```

Возвращает сегменты пути.

getTime

```
{Integer} getTime()
```

Возвращает время проезда пути в секундах.

router.Route

Примечание: Конструктор класса `router.Route` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [IGeoObject](#).

Объект, описывающий проложенный маршрут. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

См.: [route](#)

[Поля](#) | [События](#) | [Методы](#)

Поля

Имя	Тип	Описание
editor	router.Editor	Редактор маршрутов.
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от IGeoObject .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
properties	IDataManager	Данные геообъекта. Унаследовано от IGeoObject .

Имя	Тип	Описание
state	IDataManager	Состояние геообъекта. Унаследовано от IGeoObject .

События

Имя	Описание
boundsapply	Событие выставления границ маршрута карте при заданной опции <code>options.mapStateAutoApply</code> .
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие геометрии. Унаследовано от IGeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> <code>oldMap</code> - старая карта; <code>newMap</code> - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Имя	Описание
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> • <code>clientX</code> - координата касания X относительно видимой области браузера; • <code>clientY</code> - координата касания Y относительно видимой области браузера; • <code>pageX</code> - координата касания X касания относительно начала документа; • <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
update	Событие обновления маршрута при включенном редакторе маршрутов.
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getHumanJamsTime()	String	Возвращает строковое представление времени проезда пути с единицами измерения с учетом пробок.
getHumanLength()	String	Возвращает строковое представление длины пути с единицами измерения.
getHumanTime()	String	Возвращает строковое представление времени проезда пути с единицами измерения.
getJamsTime()	Integer	Возвращает время проезда маршрута в секундах с учетом пробок.
getLength()	Number	Возвращает длину маршрута в метрах.

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getPaths()	GeoObjectCollection	Возвращает коллекцию путей, из которых состоит маршрут.
getTime()	Integer	Возвращает время проезда маршрута в секундах.
getViaPoints()	GeoObjectCollection	Возвращает коллекцию транзитных точек маршрута.
getWayPoints()	GeoObjectCollection	Возвращает коллекцию путевых точек маршрута.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание полей**editor**

```
{router.Editor} editor
```

Редактор маршрутов.

Пример:

```
// Включение редактора маршрута.
ymaps.route(['Москва', 'Санкт-Петербург'], function (route) {
    route.editor.start();
    // ...
    // Завершение редактирования маршрута.
    route.editor.stop();
});
```

Описание событий**boundsapply**

Событие выставления границ маршрута карте при заданной опции `options.mapStateAutoApply`.

update

Событие обновления маршрута при включенном редакторе маршрутов.

Описание методов**getHumanJamsTime**

```
{String} getHumanJamsTime()
```

Возвращает строковое представление времени проезда пути с единицами измерения с учетом пробок.

getHumanLength

```
{String} getHumanLength()
```

Возвращает строковое представление длины пути с единицами измерения.

getHumanTime

```
{String} getHumanTime()
```

Возвращает строковое представление времени проезда пути с единицами измерения.

getJamsTime

```
{Integer} getJamsTime()
```

Возвращает время проезда маршрута в секундах с учетом пробок.

getLength

```
{Number} getLength()
```

Возвращает длину маршрута в метрах.

getPaths

```
{GeoObjectCollection} getPaths()
```

Возвращает коллекцию путей, из которых состоит маршрут.

getTime

```
{Integer} getTime()
```

Возвращает время проезда маршрута в секундах.

getViaPoints

```
{GeoObjectCollection} getViaPoints()
```

Возвращает коллекцию транзитных точек маршрута.

getWayPoints

```
{GeoObjectCollection} getWayPoints()
```

Возвращает коллекцию путевых точек маршрута.

router.Segment

Примечание: Конструктор класса `router.Segment` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Объект, описывающий сегмент маршрута. Сегмент - часть маршрута от одного манёвра до другого. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Методы](#)

Методы

Имя	Возвращает	Описание
getAction()	String	<p>Возвращает направление поворота маршрута в конце сегмента. Возможные значения:</p> <ul style="list-style-type: none"> • left - налево; • slight left — левее; • hard left - резкий поворот налево; • right - направо; • slight right — правее; • hard right - резкий поворот направо; • none - прямо; • back - разворот; • enter roundabout - въезд на перекресток с круговым движением; • leave roundabout [N] - съезд с перекрестка с круговым движением. Число N - номер поворота с круга. Данное число может отсутствовать. Например, "leave roundabout" или "leave roundabout 2"; • merge - въезд (например, на шоссе или магистраль). Обозначает слияние с транспортным потоком; • board ferry - паромная переправа.
getAngle()	Number	Определяет угол поворота маршрута в конце отрезка.
getCoordinates()	Number[][]	Возвращает координаты ломаной, описывающей геометрию сегмента.
getHumanAction()	String	Возвращает направление поворота в виде локализованной человекопонятной строки.

Имя	Возвращает	Описание
getHumanJamsTime()	String	Возвращает строковое представление времени проезда сегмента с единицами измерения с учетом пробок.
getHumanLength()	String	Возвращает строковое представление длины сегмента с единицами измерения.
getHumanTime()	String	Возвращает строковое представление времени проезда сегмента с единицами измерения.
getIndex()	Integer	Возвращает индекс данного сегмента в массиве всех сегментов пути.
getJamsTime()	Integer	Возвращает время проезда сегмента в секундах с учетом пробок.
getLength()	Number	Возвращает длину маршрута в метрах.
getPolylineEndIndex()	Integer	Возвращает индекс точки в геометрии пути, на которой заканчивается сегмент.
getPolylineStartIndex()	Integer	Возвращает индекс точки в геометрии пути, с которой начинается сегмент.
getStreet()	String	Возвращает название улицы, по которой проходит сегмент маршрута.
getTime()	Integer	Возвращает время проезда сегмента в секундах.

Описание методов

getAction

```
{String} getAction()
```

Возвращает направление поворота маршрута в конце сегмента. Возможные значения:

- left - налево;
- slight left — левее;
- hard left - резкий поворот налево;
- right - направо;
- slight right — правее;
- hard right - резкий поворот направо;
- none - прямо;
- back - разворот;
- enter roundabout - въезд на перекресток с круговым движением;

- `leave roundabout [N]` - съезд с перекрестка с круговым движением. Число N - номер поворота с круга. Данное число может отсутствовать. Например, `"leave roundabout"` или `"leave roundabout 2"`;
- `merge` - въезд (например, на шоссе или магистраль). Обозначает слияние с транспортным потоком;
- `board ferry` - паромная переправа.

getAngle

```
{Number} getAngle()
```

Определяет угол поворота маршрута в конце отрезка.

Возвращает угол поворота (в градусах).

getCoordinates

```
{Number[][]} getCoordinates()
```

Возвращает координаты ломаной, описывающей геометрию сегмента.

getHumanAction

```
{String} getHumanAction()
```

Возвращает направление поворота в виде локализованной человекопонятной строки.

getHumanJamsTime

```
{String} getHumanJamsTime()
```

Возвращает строковое представление времени проезда сегмента с единицами измерения с учетом пробок.

getHumanLength

```
{String} getHumanLength()
```

Возвращает строковое представление длины сегмента с единицами измерения.

getHumanTime

```
{String} getHumanTime()
```

Возвращает строковое представление времени проезда сегмента с единицами измерения.

getIndex

```
{Integer} getIndex()
```

Возвращает индекс данного сегмента в массиве всех сегментов пути.

getJamsTime

```
{Integer} getJamsTime()
```

Возвращает время проезда сегмента в секундах с учетом пробок.

getLength

```
{Number} getLength()
```

Возвращает длину маршрута в метрах.

getPolylineEndIndex

```
{Integer} getPolylineEndIndex()
```

Возвращает индекс точки в геометрии пути, на которой заканчивается сегмент.

getPolylineStartIndex

```
{Integer} getPolylineStartIndex()
```

Возвращает индекс точки в геометрии пути, с которой начинается сегмент.

getStreet

```
{String} getStreet()
```

Возвращает название улицы, по которой проходит сегмент маршрута.

getTime

```
{Integer} getTime()
```

Возвращает время проезда сегмента в секундах.

router.ViaPoint

Примечание: Конструктор класса `router.ViaPoint` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Расширяет [GeoObject](#).

Объект, описывающий транзитную точку маршрута. Конструктор недоступен в `package.full` (стандартный набор модулей). Данный модуль загружается по требованию.

[Поля](#) | [События](#) | [Методы](#)

Поля

Имя	Тип	Описание
balloon	geoObject.Balloon	Балун геообъекта. Унаследовано от GeoObject .
editor	IGeometryEditor	Редактор геометрии геообъекта. Унаследовано от GeoObject .
events	event.Manager	Менеджер событий. Унаследовано от GeoObject .
geometry	IGeometry null	Геометрия геообъекта. Унаследовано от GeoObject .
hint	geoObject.Hint	Хинт геообъекта. Унаследовано от GeoObject .
options	option.Manager	Менеджер опций геообъекта. Унаследовано от GeoObject .
properties	data.Manager	Менеджер данных геообъекта. Унаследовано от GeoObject .

Имя	Тип	Описание
state	data.Manager	<p>Состояние геообъекта. Определяется следующими полями:</p> <ul style="list-style-type: none"> • <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. • <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; • <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается. <p>Унаследовано от GeoObject.</p>

События

Имя	Описание
balloonclose	<p>Заккрытие балуна. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
balloonopen	<p>Открытие балуна на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
beforedrag	<p>Событие, предшествующее событию <code>drag</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> • <code>setPixelOffset</code> - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p> <p>Унаследовано от GeoObject.</p>

Имя	Описание
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p> <p>Унаследовано от GeoObject.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>

Имя	Описание
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent - оригинальное событие редактора геометрии. <p>Унаследовано от GeoObject.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Закрытие хинта. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
hintopen	<p>Открытие хинта на геообъекте. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>
mapchange	<p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. <p>Унаследовано от IParentOnMap.</p>
mousedown	<p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseenter	<p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseleave	<p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
mousemove	<p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
mouseup	<p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MouseEvent. Более детальное описание доступно в domEvent.manager.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchend	<p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent.</p> <p>Унаследовано от IDomEventEmitter.</p>
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> overlay: IOverlay null - ссылка на оверлей; oldOverlay: IOverlay null - предыдущий оверлей геообъекта. Унаследовано от IGeoObject .
parentchange	Сменился родительский объект. Поля данных: <ul style="list-style-type: none"> oldParent - старый родитель; newParent - новый родитель. Унаследовано от IChild .
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> originalEvent: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEvent.manager . Унаследовано от IDomEventEmitter .

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap .
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. Унаследован от IGeoObject .
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. Унаследован от IGeoObject .

Имя	Возвращает	Описание
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. Унаследован от IChildOnMap .
getPathIndex()	Integer	Возвращает индекс пути, в котором расположена точка.
getSegmentIndex()	Integer	Возвращает индекс сегмента пути, в котором расположена точка.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. Унаследован от IChildOnMap .

Описание методов

getPathIndex

```
{Integer} getPathIndex()
```

Возвращает индекс пути, в котором расположена точка.

getSegmentIndex

```
{Integer} getSegmentIndex()
```

Возвращает индекс сегмента пути, в котором расположена точка.

router.WayPoint

Расширяет [GeoObject](#).

Объект, описывающий путевую точку маршрута.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
router.WayPoint(feature[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>feature</code> *	—	Тип: Object Свойства и геометрия.
<code>options</code>	—	Тип: Object Опции.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
<code>balloon</code>	<code>geoObject.Balloon</code>	Балун геообъекта. Унаследовано от <code>GeoObject</code> .
<code>editor</code>	<code>IGeometryEditor</code>	Редактор геометрии геообъекта. Унаследовано от <code>GeoObject</code> .
<code>events</code>	<code>event.Manager</code>	Менеджер событий. Унаследовано от <code>GeoObject</code> .
<code>geometry</code>	<code>IGeometry</code> null	Геометрия геообъекта. Унаследовано от <code>GeoObject</code> .
<code>hint</code>	<code>geoObject.Hint</code>	Хинт геообъекта. Унаследовано от <code>GeoObject</code> .
<code>options</code>	<code>option.Manager</code>	Менеджер опций геообъекта. Унаследовано от <code>GeoObject</code> .
<code>properties</code>	<code>data.Manager</code>	Менеджер данных путевой точки. Если путевая точка была задана адресом, то в поле <code>GeocoderMetaData</code> будут содержаться метаданные геокодера. См. geocode .
<code>state</code>	<code>data.Manager</code>	Состояние геообъекта. Определяется следующими полями: <ul style="list-style-type: none"> <code>active</code>: Boolean - признак того, что на геообъекте открыт балун. <code>hover</code>: Boolean - признак того, что в данный момент на геообъект наведен указатель мыши; <code>drag</code>: Boolean - признак того, что в данный момент геообъект перетаскивается. Унаследовано от <code>GeoObject</code> .

События

Имя	Описание
<code>balloonclose</code>	Закрытие балуна. Экземпляр класса <code>Event</code> . Унаследовано от <code>GeoObject</code> .
<code>balloonopen</code>	Открытие балуна на геообъекте. Экземпляр класса <code>Event</code> . Унаследовано от <code>GeoObject</code> .

Имя	Описание
beforedrag	<p>Событие, предшествующее событию drag. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; pixelOffset - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Имена методов, доступных через метод Event.callMethod:</p> <ul style="list-style-type: none"> setPixelOffset - метод, позволяющий скорректировать значение пиксельного сдвига, который будет фактически применен. В качестве аргумента принимает новый пиксельный сдвиг в виде массива из двух чисел. <p>Если у данного события вызвать метод Event.preventDefault, то последующее событие "drag" будет отменено.</p> <p>Унаследовано от GeoObject.</p>
beforedragstart	<p>Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> position - координаты относительно документа. Массива вида [pageX, pageY]; domEvent - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие "dragstart", будет отменено.</p> <p>Унаследовано от GeoObject.</p>
click	<p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
contextmenu	<p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>
dblclick	<p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent. Более детальное описание доступно в domEventManager.</p> <p>Унаследовано от IDomEventEmitter.</p>

Имя	Описание
drag	<p>Перетаскивание геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>pixelOffset</code> - массив из двух чисел, описывающий пиксельный сдвиг на данном шаге; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragend	<p>Завершение перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
dragstart	<p>Начало перетаскивания геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; • <code>domEvent</code> - исходное DOM-событие (в виде объекта DomEvent), если оно имеется. <p>Унаследовано от GeoObject.</p>
editorstatechange	<p>Изменение состояния редактора геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>originalEvent</code> - оригинальное событие редактора геометрии. <p>Унаследовано от GeoObject.</p>
geometrychange	<p>Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> • <code>originalEvent</code>: IEvent - оригинальное событие геометрии. <p>Унаследовано от IGeoObject.</p>
hintclose	<p>Закрытие хинта. Экземпляр класса Event.</p> <p>Унаследовано от GeoObject.</p>

Имя	Описание
hintopen	Открытие хинта на геообъекте. Экземпляр класса Event . Унаследовано от GeoObject .
mapchange	Сменилась карта. Поля данных: <ul style="list-style-type: none"> oldMap - старая карта; newMap - новая карта. Унаследовано от IParentOnMap .
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Унаследовано от IDomEventEmitter .
mouseenter	Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Унаследовано от IDomEventEmitter .
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Унаследовано от IDomEventEmitter .
mousemove	Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Унаследовано от IDomEventEmitter .
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . Унаследовано от IDomEventEmitter .
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent . Унаследовано от IDomEventEmitter .

Имя	Описание
multitouchmove	<p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
multitouchstart	<p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <code>clientX</code> - координата касания X относительно видимой области браузера; <code>clientY</code> - координата касания Y относительно видимой области браузера; <code>pageX</code> - координата касания X касания относительно начала документа; <code>pageY</code> - координата касания Y касания относительно начала документа. <p>Унаследовано от IDomEventEmitter.</p>
optionschange	<p>Изменение в опциях объекта.</p> <p>Унаследовано от ICustomizable.</p>
overlaychange	<p>Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>overlay</code>: IOverlay null - ссылка на оверлей; <code>oldOverlay</code>: IOverlay null - предыдущий оверлей геообъекта. <p>Унаследовано от IGeoObject.</p>
parentchange	<p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <code>oldParent</code> - старый родитель; <code>newParent</code> - новый родитель. <p>Унаследовано от IChild.</p>

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>originalEvent</code>: IEvent - оригинальное событие менеджера данных. Унаследовано от IGeoObject .
wheel	Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса MapEvent . Более детальное описание доступно в domEventManager . <p>Унаследовано от IDomEventEmitter.</p>

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. <p>Унаследован от IParentOnMap.</p>
getOverlay()	vow.Promise	Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки. <p>Унаследован от IGeoObject.</p>
getOverlaySync()	IOverlay null	Метод предоставляет синхронный доступ к оверлею. <p>Унаследован от IGeoObject.</p>
getParent()	IParentOnMap null	Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен. <p>Унаследован от IChildOnMap.</p>
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. <p>Унаследован от IChildOnMap.</p>

Описание полей

properties

```
{data.Manager} properties
```

Менеджер данных путевой точки. Если путевая точка была задана адресом, то в поле GeocoderMetaData будут содержаться метаданные геокодера. См. [geocode](#).

shape

shape.Circle

Расширяет [IShape](#).

Пиксельная фигура "Круг".

[Конструктор](#) | [Методы](#)

Конструктор

```
shape.Circle(pixelGeometry[, params])
```

Создает фигуру.

Параметры:

Параметр	Значение по умолчанию	Описание
pixelGeometry *	—	Тип: IPixelCircleGeometry Пиксельная геометрия фигуры.
params	—	Тип: Object Параметры фигуры.
params.fill	true	Тип: Boolean Флаг наличия заливки.
params.outline	true	Тип: Boolean Флаг наличия линии обводки.
params.strokeWidth	0	Тип: Number Толщина линии обводки в пикселях.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет лежит ли переданная точка внутри фигуры. Унаследован от IShape .

Имя	Возвращает	Описание
equals(shape)	Boolean	Возвращает true, если переданная фигура эквивалентна данной. Унаследован от IShape .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей фигуру. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IShape .
getGeometry()	IPixelGeometry	Возвращает пиксельную геометрию фигуры. Унаследован от IShape .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IShape .
scale(factor)	IShape	Создает отмасштабированную копию фигуры. Унаследован от IShape .
shift(offset)	IShape	Создает копию фигуры сдвинутую на заданную величину. Унаследован от IShape .

shape.LineString

Расширяет [IShape](#).

Пиксельная фигура "Ломаная линия".

[Конструктор](#) | [Методы](#)

Конструктор

```
shape.LineString(pixelGeometry[, params])
```

Создает фигуру.

Параметры:

Параметр	Значение по умолчанию	Описание
pixelGeometry *	—	Тип: IPixelLineStringGeometry Пиксельная геометрия фигуры.

Параметр	Значение по умолчанию	Описание
<code>params</code>	—	Тип: Object Параметры фигуры.
<code>params.strokeWidth</code>	1	Тип: Number Толщина линии в пикселях.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
<code>contains(position)</code>	Boolean	Проверяет лежит ли переданная точка внутри фигуры. Унаследован от IShape .
<code>equals(shape)</code>	Boolean	Возвращает true, если переданная фигура эквивалентна данной. Унаследован от IShape .
<code>getBounds()</code>	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей фигуру. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IShape .
<code>getGeometry()</code>	IPixelGeometry	Возвращает пиксельную геометрию фигуры. Унаследован от IShape .
<code>getType()</code>	String	Возвращает идентификатор типа геометрии. Унаследован от IShape .
<code>scale(factor)</code>	IShape	Создает отмасштабированную копию фигуры. Унаследован от IShape .
<code>shift(offset)</code>	IShape	Создает копию фигуры сдвинутую на заданную величину. Унаследован от IShape .

shape.MultiGeometry

Расширяет [IShape](#).

Пиксельная фигура "Мультиполигон"

[Конструктор](#) | [Методы](#)

Конструктор

```
shape.MultiGeometry(pixelGeometry[, params])
```

Создает фигуру.

Параметры:

Параметр	Значение по умолчанию	Описание
pixelGeometry *	—	Тип: IPixelMultiGeometry Пиксельная геометрия фигуры.
params	—	Тип: Object Параметры фигуры.
params.fill	true	Тип: Boolean Флаг наличия заливки.
params.outline	true	Тип: Boolean Флаг наличия линии обводки.
params.strokeWidth	0	Тип: Number Толщина линии обводки в пикселях.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет лежит ли переданная точка внутри фигуры. Унаследован от IShape .
equals(shape)	Boolean	Возвращает true, если переданная фигура эквивалентна данной. Унаследован от IShape .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей фигуру. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IShape .

Имя	Возвращает	Описание
getGeometry()	IPixelGeometry	Возвращает пиксельную геометрию фигуры. Унаследован от IShape .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IShape .
scale(factor)	IShape	Создает отмасштабированную копию фигуры. Унаследован от IShape .
shift(offset)	IShape	Создает копию фигуры сдвинутую на заданную величину. Унаследован от IShape .

shape.MultiPolygon

Расширяет [IShape](#).

Пиксельная фигура "Мультиполигон".

[Конструктор](#) | [Методы](#)

Конструктор

```
shape.MultiPolygon(pixelGeometry[, params])
```

Создает фигуру.

Параметры:

Параметр	Значение по умолчанию	Описание
pixelGeometry *	—	Тип: IPixelMultiPolygonGeometry Пиксельная геометрия фигуры.
params	—	Тип: Object Параметры фигуры.
params.fill	true	Тип: Boolean Флаг наличия заливки.
params.outline	true	Тип: Boolean Флаг наличия линии обводки.
params.strokeWidth	0	Тип: Number Толщина линии обводки в пикселях.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет лежит ли переданная точка внутри фигуры. Унаследован от IShape .
equals(shape)	Boolean	Возвращает true, если переданная фигура эквивалентна данной. Унаследован от IShape .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей фигуру. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IShape .
getGeometry()	IPixelGeometry	Возвращает пиксельную геометрию фигуры. Унаследован от IShape .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IShape .
scale(factor)	IShape	Создает отмасштабированную копию фигуры. Унаследован от IShape .
shift(offset)	IShape	Создает копию фигуры сдвинутую на заданную величину. Унаследован от IShape .

shape.Polygon

Расширяет [IShape](#).

Пиксельная фигура "Полигон".

[Конструктор](#) | [Методы](#)

Конструктор

```
shape.Polygon(pixelGeometry[, params])
```

Создает фигуру.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>pixelGeometry</code> *	—	Тип: IPixelPolygonGeometry Пиксельная геометрия фигуры.
<code>params</code>	—	Тип: Object Параметры фигуры.
<code>params.fill</code>	true	Тип: Boolean Флаг наличия заливки.
<code>params.outline</code>	true	Тип: Boolean Флаг наличия линии обводки.
<code>params.strokeWidth</code>	0	Тип: Number Толщина линии обводки в пикселях.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
<code>contains(position)</code>	Boolean	Проверяет лежит ли переданная точка внутри фигуры. Унаследован от IShape .
<code>equals(shape)</code>	Boolean	Возвращает true, если переданная фигура эквивалентна данной. Унаследован от IShape .
<code>getBounds()</code>	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей фигуру. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IShape .
<code>getGeometry()</code>	IPixelGeometry	Возвращает пиксельную геометрию фигуры. Унаследован от IShape .
<code>getType()</code>	String	Возвращает идентификатор типа геометрии. Унаследован от IShape .

Имя	Возвращает	Описание
scale(factor)	IShape	Создает отмасштабированную копию фигуры. Унаследован от IShape .
shift(offset)	IShape	Создает копию фигуры сдвинутую на заданную величину. Унаследован от IShape .

shape.Rectangle

Расширяет [IShape](#).

Пиксельная фигура "Прямоугольник".

[Конструктор](#) | [Методы](#)

Конструктор

```
shape.Rectangle(pixelGeometry[, params])
```

Создает фигуру.

Параметры:

Параметр	Значение по умолчанию	Описание
pixelGeometry *	—	Тип: IPixelRectangleGeometry Пиксельная геометрия фигуры.
params	—	Тип: Object Параметры фигуры.
params.fill	true	Тип: Boolean Флаг наличия заливки.
params.outline	true	Тип: Boolean Флаг наличия линии обводки.
params.strokeWidth	0	Тип: Number Толщина линии обводки в пикселях.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет лежит ли переданная точка внутри фигуры. Унаследован от IShape .

Имя	Возвращает	Описание
equals(shape)	Boolean	Возвращает true, если переданная фигура эквивалентна данной. Унаследован от IShape .
getBounds()	Number[][] null	Возвращает координаты двух противоположных углов области, охватывающей фигуру. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. Унаследован от IShape .
getGeometry()	IPixelGeometry	Возвращает пиксельную геометрию фигуры. Унаследован от IShape .
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IShape .
scale(factor)	IShape	Создает отмасштабированную копию фигуры. Унаследован от IShape .
shift(offset)	IShape	Создает копию фигуры сдвинутую на заданную величину. Унаследован от IShape .

shape.storage

Статический объект.

Экземпляр класса [util.Storage](#)

Хранилище геометрий хотспотных шейпов.

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.

suggest

Статическая функция.

Обрабатывает запросы за поисковыми подсказками. Возвращает объект Promise, который будет либо отклонён с ошибкой, либо подтверждён массивом объектов вида { displayName: "Мытищи, Московская область", value: "Россия, Московская область, Мытищи " }. Поле displayName отвечает за представление топонима в удобном для чтения пользователем виде, поле value — значение, которое необходимо подставить в поле ввода после выбора пользователем данной подсказки.

Возвращает объект Promise.

```
{ vow.Promise } suggest(request[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>request *</code>	—	Тип: String Строка запроса.
<code>options</code>	—	Тип: Object Опции.
<code>options.boundedBy</code>	—	Тип: Number[][] Прямоугольная область на карте, где предположительно находится искомый объект. Должна быть задана как массив, например [[30, 40], [50, 50]].
<code>options.provider</code>	'yandex#map'	Тип: <code>ISuggestProvider</code> String Провайдер поисковых подсказок. Можно воспользоваться встроенным провайдером подсказок для объектов карты 'yandex#map' или задать собственный.
<code>options.results</code>	—	Тип: Number Максимальное количество возвращаемых результатов.

* Обязательный параметр/опция.

Пример:

```
ymaps.suggest('мыт').then(function (items) {  
  // items - массив поисковых подсказок.  
});
```

SuggestView

Расширяет `ICustomizable`, `IEventEmitter`.

Создаёт выпадающую панель с поисковыми подсказками и прикрепляет к HTML-элементу `<input type="text">`.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
SuggestView(element[, options])
```

Параметры:

Параметр	Значение по умолчанию	Описание
element *	—	Тип: HTMLElement string HTML-элемент или его id.
options	—	Тип: Object Опции.
options.boundedBy	—	Тип: Number[][] Прямоугольная область на карте, где предположительно находится искомый объект. Должна быть задана как массив, например <code>[[30, 40], [50, 50]]</code> .
options.container	—	Тип: HTMLElement HTML-элемент для размещения верстки панели подсказок. Если не задан, панель подсказок добавляется в родителя HTML-элемента <code>input</code> , для которого создаётся.

Параметр	Значение по умолчанию	Описание
<code>options.layout</code>	<code>'islands#suggestView'</code>	<p>Тип: String ISuggestViewLayout</p> <p>Макет панели.</p> <p>В конструктор макета передается объект, содержащий поля:</p> <ul style="list-style-type: none"> <code>suggestView</code> — ссылка панель с поисковыми подсказками; <code>options</code> — менеджер опций элемента управления (см. <code>suggestView.options</code>); <code>state</code> — менеджер состояния элемента управления (см. <code>suggestView.state</code>). <p>Макет меняет свой внешний вид на основе состояния и опций элемента панели подсказок. Элемент управления, в свою очередь, реагирует на интерфейсные события макета и меняет значения полей <code>suggestView.state</code> в зависимости от полученных команд.</p>
<code>options.offset</code>	—	<p>Тип: <code>Number[]</code></p> <p>Отступы панели подсказок от её положения по умолчанию (по умолчанию панель подсказок прикрепляется к нижней границе элемента <code>input</code> и имеет одинаковую с ним ширину). Задаётся в виде смещений по горизонтали и вертикали относительно левого нижнего угла элемента <code>input</code>.</p>
<code>options.provider</code>	<code>"yandex#map"</code>	<p>Тип: String ISuggestProvider</p> <p>Провайдер поисковых подсказок. Может быть задан объектом, реализующим интерфейс ISuggestProvider, либо стандартным значением <code>"yandex#map"</code>.</p>
<code>options.results</code>	5	<p>Тип: <code>Number</code></p> <p>Максимальное количество показываемых подсказок.</p>

Параметр	Значение по умолчанию	Описание
<code>options.width</code>	—	Тип: Number Ширина панели подсказок. По умолчанию совпадает с шириной HTML-элемента <code>input</code> , к которому панель привязана.
<code>options.zIndex</code>	40000	Тип: Number z-index для dom-элемента панели подсказок.

* Обязательный параметр/опция.

Примеры:

1.

```
<input type="text" id="suggest"/>
<script src="//api-maps.yandex.ru/2.1/?lang=ru_RU&load=SuggestView&onload=onLoad"></script>
<script>
function onLoad (ymaps) {
    var suggestView = new ymaps.SuggestView('suggest');
}
</script>
```

2.

```
<input type="text" id="suggest"/>
<script src="//api-maps.yandex.ru/2.1/?lang=ru_RU&load=SuggestView&onload=onLoad"></script>
<script>
function onLoad (ymaps) {
    var suggestView = new ymaps.SuggestView('suggest', {results: 1, offset: [20, 30]});
}
</script>
```

Поля

Имя	Тип	Описание
<code>events</code>	<code>IEventManager</code>	Менеджер событий. Унаследовано от <code>IEventEmitter</code> .
<code>options</code>	<code>IOptionManager</code>	Менеджер опций. Унаследовано от <code>ICustomizable</code> .

Имя	Тип	Описание
<code>state</code>	<code>data.Manager</code>	<p>Состояние панели поисковых подсказок. Имена полей, доступных через метод <code>data.Manager.get</code>:</p> <ul style="list-style-type: none"><code>request</code>: String — текущий активный запрос;<code>items</code>: Object[] — массив поисковых подсказок (объектов с полями <code>value</code> и <code>displayName</code>);<code>activeIndex</code>: Number null — индекс текущей активной подсказки, выделенная пользователем путём наведения мыши или с клавиатуры, или null, если ни одна подсказка не активна;<code>panelClosed</code>: Boolean — признак, была ли панель закрыта пользователем путём нажатия клавиши ESC или выбора одной из подсказок.

События

Имя	Описание
<code>optionschange</code>	<p>Изменение в опциях объекта.</p> <p>Унаследовано от <code>ICustomizable</code>.</p>
<code>select</code>	<p>Пользователь выбрал одну из поисковых подсказок. Имена полей, доступных через метод <code>Event.get</code>:</p> <ul style="list-style-type: none"><code>item</code> — поисковая подсказка, объект с полями <code>displayName</code> и <code>value</code>.

Методы

Имя	Описание
<code>destroy()</code>	Уничтожает панель подсказок.

Описание полей

`state`

```
{data.Manager} state
```

Состояние панели поисковых подсказок. Имена полей, доступных через метод `data.Manager.get`:

- `request`: String — текущий активный запрос;
- `items`: Object[] — массив поисковых подсказок (объектов с полями `value` и `displayName`);
- `activeIndex`: Number|null — индекс текущей активной подсказки, выделенная пользователем путём наведения мыши или с клавиатуры, или null, если ни одна подсказка не активна;
- `panelClosed`: Boolean — признак, была ли панель закрыта пользователем путём нажатия клавиши ESC или выбора одной из подсказок.

Описание событий

`select`

Пользователь выбрал одну из поисковых подсказок. Имена полей, доступных через метод `Event.get`:

- `item` — поисковая подсказка, объект с полями `displayName` и `value`.

Описание методов

`destroy`

```
{ } destroy()
```

Уничтожает панель подсказок.

template

template.filtersStorage

Статический объект.

Экземпляр класса [util.Storage](#)

Хранилище фильтров для шаблонов. Фильтры из хранилища можно использовать во всех макетах, созданных с помощью [templateLayoutFactory](#). По умолчанию в хранилище объявлены следующие ключи для фильтров:

- `default` — позволяет устанавливать значения по умолчанию. Например, так: `{{ properties.header|default:"Название" }}`

Методы

Примеры:

1.

```
// Напишем простой фильтр, который будет преобразовывать
// дату в формате «dd.mm.yyyy» в формат «dd month yyyy».
// Для этого создадим функцию-фильтр, которая будет возвращать новое значение.

// При вызове фильтра в фильтр передаются следующие аргументы:
// менеджер данных шаблона данные.Менеджер, значение и значение, установленное для фильтра.
var dateFilter = function (data, dateString, filterValue) {
    var months = [
        'январь', 'февраль', 'март', 'апрель',
        'мая', 'июнь', 'июль', 'август',
        'сентябрь', 'октябрь', 'ноябрь', 'декабрь'
    ];
    var date = dateString.split('.');

    date[1] = months[parseInt(date[1], 10)];
    return date.join(' ');
};

ymaps.template.filtersStorage.add('date', dateFilter);

// Тогда мы можем использовать его в таких конструкциях, как
// {{ "21.10.2014"|date }} и значение будет равно "21 октября 2014 года".
```

2.

```
// Напишем фильтр, который будет находить и заменять подстроки в тексте.
// Формат значений для замены в фильтре «subString_newSubString».

// При вызове фильтра в функцию будут переданы следующие аргументы:
// менеджер данных шаблона data.Manager, текст и значение, заданное фильтру.
var replaceFilter = function (data, text, replace) {
    replace = replace.trim();
    // Убираем кавычки.
    replace = replace.slice(1, replace.length - 1);

    // Будем находить в тексте часть, указанную до «_» и заменять её на то, что указано после.
    var values = replace.split('_'),
        from = new RegExp(values[0], 'g'),
        to = values[1];

    return text.replace(from, to);
};

// Теперь можно использовать в шаблонах конструкции вида
// {{ значение будет "text replaced test replace".
```

3.

```
// В данном примере в макет добавляются значение опции "colorClass" и значение свойства "header".
// Если для свойства "header" не задано значение, вставляется строка "Title".
var LayoutClass = ymaps.templateLayoutFactory.createClass(
  '<h1 class="{{ options.colorClass }}">' +
    '{{ properties.header|default:"Title" }}' +
  '</h1>'
);
```

Методы

Имя	Возвращает	Описание
add(key, object)	util.Storage	Добавляет объект в хранилище.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.
remove(key)	util.Storage	Удаляет пару "ключ: значение" из хранилища.

Template

Шаблонизатор. В API Яндекс.Карт поддерживается базовый синтаксис языков шаблонов Twig/Django Templates. Поддерживаются следующие операции:

- Подстановка значения — **{{ field_name }}**.
- Если запрашиваемое поле данных отсутствует или имеет пустое значение, то можно предусмотреть стандартное значение — **{{ field_name|default:default_value }}**. Значением по умолчанию может быть число, строка (записывается в кавычках) или другое поле данных.
- По умолчанию значение обрабатывается функцией escape для предотвращения XSS уязвимости. Для отмены поведения необходимо добавить фильтр "raw" filter - **{{ field_name|default:default_value|raw }}**.
- С помощью [template.filtersStorage](#) можно создавать свои фильтры и использовать их аналогично описанным выше.
- Для вставки подмакета необходимо использовать конструкцию **{% include field_name_or_key %}**. Шаблонизатор, встретив такую конструкцию, попытается использовать значение в поле как ключ вложенного макета.
- Условие записывается в виде:

{% if condition %} ... {% else %} ... {% endif %}, блоки **else** и **elseif** можно опустить.

Внутри блоков **if**, **else** и **elseif** можно использовать любые конструкции языка шаблонов.
- Для выполнения итерации по массиву или по объекту нужно использовать конструкцию **for**.

{% for value in array_or_hash %} ... {% endfor %}. Внутри блока **for** можно использовать любые конструкции языка шаблонов.
- Для получения номера итерации в массиве или имени поля в хеше нужно использовать конструкцию:

{% for значение в массив_или_хеш %} ... {% endfor %}.
- Вы можете добавить дополнительную проверку на опасные атрибуты `onload` и `onerror` с помощью комментария `sanitize` в шаблоне. При наличии этого комментария шаблонизатор покажет ошибку если один из этих атрибутов будет в шаблоне, в противном случае в будет warning в консоли.

[Конструктор](#) | [Методы](#)

Конструктор

Template(text)

Параметры:

Параметр	Значение по умолчанию	Описание
text *	—	Тип: String Строка-шаблон.

* Обязательный параметр/опция.

Примеры:

1.

```
// Получим имя пользователя из менеджера данных data.Manager.  
// Если имя не задано, то в результате получится строка «Незарегистрированный пользователь».  
var data = new ymaps.data.Manager({  
  user: {  
    name: "Виктор",  
    age: 25  
  },  
  home: [55.72725771214265, 37.640390506634006]  
}),  
var template = new ymaps.Template('{{ user.name |default: "Unregistered user"}}');  
var result = template.build(data);  
  
var result = template.build(data);  
console.log(result.text); // Выведет в консоль «Виктор».
```

2.

```
// Предположим, что у нас есть  
3 группы пользователей, и нам нужно напечатать индивидуальное приветствие для каждой группы.  
var data = new ymaps.data.Manager({  
  groups: {  
    administrator: {  
      id: 1,  
      name: "administrator"  
    },  
    moderator: {  
      id: 2,  
      name: "moderator"  
    },  
    user: {  
      id: 3,  
      name: "user"  
    }  
  },  
  userGroupId: 2  
});  
var template = new ymaps.Template('Hi, \\  
{% if (userGroupId == 1) %}{{ groups.administrator.name }}\  
{% elseif (userGroupId == 2) %}{{ groups.moderator.name }}\  
{% elseif (userGroupId == 3) %}{{ groups.user.name }}\  
{% else %}guest{% endif %}!'  
);  
  
var result = template.build(data);  
console.log(result.text); // Выведет «Привет, модератор!» в консоль.
```

3.

```
// С использованием комментария sanitize  
var template = new ymaps.Template('<!--sanitize--><div>Привет!</div>');
```

Методы

Имя	Возвращает	Описание
<code>build(data)</code>	Object	<p>Возвращает объект со следующими полями:</p> <ul style="list-style-type: none"><code>{String} text</code> — результат выполнения шаблонизации;<code>{Object[]} renderedValues</code> — массив с использованными данными из менеджера.

Описание методов

build

```
{Object} build(data)
```

Возвращает объект, содержащий следующие поля:

- `{String} text` — результат выполнения шаблонизации;
- `{Object[]} renderedValues` — массив с использованными данными из менеджера.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>data *</code>	—	<p>Тип: <code>IDataManager</code></p> <p>Менеджер данных.</p>

* Обязательный параметр/опция.

Пример:

```
// Получим адрес дома из имеющихся координат и выведем по шаблону
// всех его жильцов в формате: «имя: возраст».
var data = new ymaps.data.Manager({
  users: [
    {name: "Виталий", age: 40},
    {name: "Георгий", age: 20}
  ],
  home: {
    coords: [55.736652, 37.620589],
    address: null
  }
}),
var template = new ymaps.Template('{{home.address}}: <ul>{% for user in users %}<li>{{user.name}}: {{user.age}}</li>{% endfor %}</ul>');

// Сделаем обратное геокодирование с помощью geocode.
ymaps.geocode(data.get('home.coords')).then(function (res) {
  var address = res.geoObjects.get(0).properties.get('name');
  // Установим в менеджер полученный адрес.
  data.set('home.address', address);

  // Соберём шаблон из имеющихся данных.
  var result = template.build(data);
  // Выведем результат в консоль.
  console.log(result.text);
});
```


templateLayoutFactory

Статический объект.

Фабрика для создания класса макета из текстового шаблона. Позволяет создавать классы, реализующие интерфейс [ILayout](#), используя язык шаблонов. В API Яндекс.Карт поддерживается базовый синтаксис языков шаблонов Twig/Django Templates. Более подробно о синтаксисе можно прочитать в описании [Template](#).

См.: [layout.templateBased.Base](#)

Методы

Примеры:

1.

```
// В данном примере в макет добавляются значение опции "colorClass" и значение свойства "header".
// Если для свойства "header" не задано значение, вставляется строка "Title".
var LayoutClass = ymaps.templateLayoutFactory.createClass(
  '<h1 class="{ { options.colorClass } }">' +
    '{ { properties.header|default:"Title" } }' +
  '</h1>'
);
```

2.

```
// В зависимости от значения опции "width" подключается один из макетов.
var LayoutClass = ymaps.templateLayoutFactory.createClass(
  '{ % if options.width > 200 % }' +
  // Необходимый макет будет найден в опциях.
  '{ % include options.wideLayout % }' +
  '{ % else % }' +
  // Записываем ключ.
  '{ % include "cluster#balloonCarousel" % }' +
  '{ % endif % }'
);
```

3.

```
// Вывод в макет балуна массива имен.
var CustomLayoutClass = ymaps.templateLayoutFactory.createClass(
  '<ul>' +
  '{ % for name in properties.names % }' +
  // Переменная name доступна только в блоке for ... endfor.
  '<li>{ { name } }</li>' +
  '{ % endfor % }' +
  '</ul>'
);

var placemark = new ymaps.Placemark([54.83, 37.11], {
  names: ['Logan', 'Sofia', 'Mason', 'Layla']
}, {
  balloonContentLayout: CustomLayoutClass
});
```

4.

```
// Получение имени полей.
var CustomLayoutClass = ymaps.templateLayoutFactory.createClass(
  '<ul>' +
  '{ % for key, value in properties.hash % }' +
  '<li>{ { key } } { { value } }</li>' +
  '{ % endfor % }' +
  '</ul>'
);

var placemark = new ymaps.Placemark([54.83, 37.11], {
  hash: { key1: "value1", key2: "value2", key3: "value3" }
}, {
  balloonContentLayout: CustomLayoutClass
});
```

Методы

Имя	Статический	Возвращает	Описание
<code>templateLayoutFactory.createClass(template[, overrides[, staticMethods]])</code>		Function	Возвращает конструктор макета. Созданный класс будет унаследован от класса layout.templateBased.Base с переопределением списка методов, указанных в <code>overrides</code> .

Описание методов**createClass**

```
{Function} <static> templateLayoutFactory.createClass(template[, overrides[, staticMethods]])
```

Возвращает конструктор макета. Созданный класс будет унаследован от класса [layout.templateBased.Base](#) с переопределением списка методов, указанных в `overrides`.

Параметры:

Параметр	Значение по умолчанию	Описание
template *	—	Тип: String Шаблон html-содержимого макетов.
overrides	—	Тип: Object Переопределения родительских методов. Можно переопределять/дополнять методы <code>build</code> , <code>clear</code> и <code>rebuild</code> .
staticMethods	—	Тип: Object Установка статических методов класса.

* Обязательный параметр/опция.

traffic**traffic.provider****traffic.provider.Actual**

Расширяет [ITrafficProvider](#).

Провайдер актуальных пробок. Доступен в хранилище провайдеров под ключом 'traffic#actual'.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
traffic.provider.Actual([options[, state]])
```

Создает провайдер актуальных пробок.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	<p>Тип: Object</p> <p>Опции провайдера. Опции для слоев провайдера задаются через опции глобальной коллекции слоев Map.layers.</p> <ul style="list-style-type: none">Опции для картиночного слоя Layer задаются с префиксом 'trafficImage';Опции для картиночного слоя hotspot.Layer задаются с префиксом 'trafficJam';Опции для слоя инфоточек задаются с префиксом 'trafficInfo'. Слой инфоточек является экземпляром класса hotspot.Layer.
<code>options.autoUpdate</code>	true	<p>Тип: Boolean</p> <p>Флаг, включающий автоматическое обновление данных о пробках. Автоматическое обновление происходит только в том случае, если на карте каждые 4 минуты возникает событие <code>mousemove</code>. Если событие не возникает, пробки перестают обновляться до появления нового события.</p>
<code>state</code>	—	<p>Тип: Object</p> <p>Состояние провайдера.</p>
<code>state.infoLayerShown</code>	false	<p>Тип: Boolean</p> <p>Флаг, включающий отображение слоя дорожных событий.</p>

Пример:

```
// Создадим провайдер пробок "сейчас" с включенным слоем дорожных событий
// и разместим его на карте.
var actualProvider = new ymaps.traffic.provider.Actual({}, {infoLayerShown: true});
actualProvider.setMap(myMap);

// Запретим показ балунов по клику для слоя инфоточек.
```

```

myMap.layers.options.set({
    // Название опции сформировано из опции хотспотного слоя
    // 'openBalloonOnClick' путем добавления префикса 'trafficInfo'.
    trafficInfoOpenBalloonOnClick: false
});

// ...
// Удаление провайдера с карты.
actualProvider.setMap(null);

```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
state	data.Manager	Состояние провайдера. Имена полей, доступных через метод <code>data.Manager.get</code> : <ul style="list-style-type: none"> <code>isInitd</code> - признак, готов ли провайдер предоставлять данные; <code>infoLayerShown</code> - признак, показан ли слой дорожных событий; <code>timestamp</code> - текущее время во временной зоне UTC+0 в секундах; <code>localtime</code> - локальное время, для которого сервер в текущий момент отдает данные, в формате 'ЧЧ:ММ'; <code>level</code> - уровень загруженности дорог в баллах - от 0 до 10. <code>isotime</code> - строка, содержащая текущую дату в формате "YYYY-MM-DDThh:mm:ss±hhmm".

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .

Методы

Имя	Возвращает	Описание
getMap()	Map null	Возвращает ссылку на карту. Унаследован от ITrafficProvider .
setMap(Reference)		
update()		Шлёт запрос на обновление пробок.

Описание полей

state

```
{data.Manager} state
```

Состояние провайдера. Имена полей, доступных через метод `data.Manager.get`:

- `isInited` - признак, готов ли провайдер предоставлять данные;
- `infoLayerShown` - признак, показан ли слой дорожных событий;
- `timestamp` - текущее время во временной зоне UTC+0 в секундах;
- `localtime` - локальное время, для которого сервер в текущий момент отдает данные, в формате 'ЧЧ:ММ';
- `level` - уровень загруженности дорог в баллах - от 0 до 10.
- `isotime` - строка, содержащая текущую дату в формате "YYYY-MM-DDThh:mm:ss±hhmm".

Пример:

```
var actualProvider = new ymaps.traffic.provider.Actual();
actualProvider.setMap(myMap);
actualProvider.state.events.add('change', function () {
  if (actualProvider.state.get('isInited')) {
    alert('Провайдер готов предоставлять данные.');
```

Описание методов

update

```
{ } update()
```

Шлёт запрос на обновление пробок.

Пример:

```
var trafficControl = new ymaps.control.TrafficControl({shown: true});
map.controls.add(trafficControl);
function updateProvider () {
  trafficControl.getProvider('traffic#actual').update();
}
// Будем слать запрос на обновление данных каждые 4 минуты.
window.setInterval(updateProvider, 4 * 60 * 1000);
```

traffic.provider.Archive

Расширяет [ITrafficProvider](#).

Провайдер архива пробок. Позволяет показывать состояние пробок, типичное для данного региона в конкретный день недели и конкретное время суток.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
traffic.provider.Archive([options[, state]])
```

Создает экземпляр провайдера архива пробок.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>options</code>	—	<p>Тип: Object</p> <p>Опции провайдера. Опции для слоев провайдера задаются через опции глобальной коллекции слоев <code>Map.layers</code>.</p> <ul style="list-style-type: none"> Опции для карточного слоя <code>Layer</code> задаются с префиксом <code>'trafficImage'</code>; Опции для карточного слоя <code>hotspot.Layer</code> задаются с префиксом <code>'trafficJam'</code>;
<code>options.showCurrentTimeFirst</code>	<code>true</code>	<p>Тип: Boolean</p> <p>При первом показе архивных данных выставить время, близкое к текущему.</p>
<code>state</code>	—	<p>Тип: Object</p> <p>Состояние провайдера.</p>
<code>state.timestamp</code>	—	<p>Тип: Number</p> <p>Время, для которого показываются пробки "обычно". Это время, прошедшее с 00:00 понедельника до требуемого времени в секундах. Должно быть кратно $60 * 15 = 900$, так как данные на сервере доступны для моментов времени с разницей в 15 минут. Время задается для нулевого часового пояса (UTC+0).</p>

Пример:

```
// Создадим провайдер пробок "обычно" и зададим ему timestamp для 17:47 среды
// в нулевом часовом поясе. // Обратите внимание, что местное время будет зависеть
// от расположения центра карты.
// К примеру, 17:47 в нулевом часовом поясе будет соответствовать 21:47 в Москве.

// Вычислим значение параметра timestamp для требуемого времени.
var timestamp = 2 * 24 * 60 * 60 + // дважды по 24 часа — это время за понедельник и вторник.
  17 * 60 * 60 + // 17 часов прошло с 00:00 среды.
  45 * 60, // поскольку время должно быть кратно 15 минутам, берем не 47 минут, а 45.
archiveProvider = new ymaps.traffic.provider.Archive({
  // Запретим при первом открытии выставлять время, близкое к текущему.
  showCurrentTimeFirst: false
}), {
  // Зададим начальное время самостоятельно.
  timestamp: timestamp
});
archiveProvider.setMap(map);

// Запретим показ всплывающих подсказок для слоя пробок.
myMap.layers.options.set({
  // Название опции сформировано из опции слоя активных областей
  // 'openHintOnHover' путем добавления префикса 'trafficJam'.
  trafficJamOpenHintOnHover: false
});

// ...
```

```
// Удаляем провайдера с карты.
archiveProvider.setMap(null);
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable .
state	data.Manager	Состояние провайдера. Имена полей, доступных через метод <code>data.Manager.get</code> : <ul style="list-style-type: none"> <code>isInit</code> - признак, готов ли провайдер предоставлять данные; <code>timeZone</code> - смещение по времени для текущего временного пояса относительно UTC+0. Измеряется в секундах; <code>dst</code> - признак перехода на зимнее/летнее время. При <code>dst='dst'</code> время летнее; <code>timestamp</code> - текущее время во временной зоне UTC+0 в секундах; <code>localtime</code> - локальное время, которое возвращает в ответе сервер; <code>level</code> - уровень загруженности дорог в баллах - от 0 до 10.

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable .

Методы

Имя	Возвращает	Описание
getMap()	Map null	Возвращает ссылку на карту. Унаследован от ITrafficProvider .
getTime()	Object null	Возвращает день недели, часы и минуты состояния провайдера с учетом часового пояса и перехода на зимнее/летнее время. То есть то время, которое пользователь видит в элементе управления пробками.
setMap(Reference)		

Имя	Возвращает	Описание
<code>setTime(time[, callback])</code>		<p>Позволяет задавать время для архивного провайдера в минутах, часах и днях недели. Выставляет локальное время только после того, как провайдер инициализирует поля <code>timeZone</code> и <code>dst</code>.</p> <ul style="list-style-type: none"><code>timeZone</code> - поле, которое показывает, в каком часовом поясе на данный момент находится центр карты. При перемещении центра карты из одного часового пояса в другой локальное время может изменяться.<code>dst</code> - признак перехода на летнее/зимнее время. При <code>dst='dst'</code> время летнее. <p>Постоянной составляющей времени для провайдера пробок "обычно" служит поле <code>timestamp</code>, которое отражает текущее время в нулевом часовом поясе (UTC+0). При переходе из одного часового пояса в другой <code>timestamp</code> не изменяется. Значения полей <code>"timestamp"</code>, <code>"dst"</code> и <code>"timeZone"</code> можно получить через поле traffic.provider.Archive.state.</p>

Описание полей

state

```
{data.Manager} state
```

Состояние провайдера. Имена полей, доступных через метод `data.Manager.get`:

- `isInited` - признак, готов ли провайдер предоставлять данные;

- `timeZone` - смещение по времени для текущего временного пояса относительно UTC+0. Измеряется в секундах;
- `dst` - признак перехода на зимнее/летнее время. При `dst='dst'` время летнее;
- `timestamp` - текущее время во временной зоне UTC+0 в секундах;
- `localtime` - локальное время, которое возвращает в ответе сервер;
- `level` - уровень загруженности дорог в баллах - от 0 до 10.

Пример:

```
var archiveProvider = new ymaps.traffic.provider.Archive();
archiveProvider.setMap(myMap);
archiveProvider.state.events.add('change', function () {
    if (archiveProvider.state.get('isInited')) {
        alert('Provider is ready to provide data.');
```

Описание методов**getTime**

```
{Object|null} getTime()
```

Возвращает день недели, часы и минуты состояния провайдера с учетом часового пояса и перехода на зимнее/летнее время. То есть то время, которое пользователь видит в элементе управления пробками.

Возвращает объект с полями:

- `dayOfWeek` — сокращенное название дня недели ('mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun');
- `hours` — часы;
- `minutes` - минуты.

Если центр карты находится в точке, для которой не удастся опеределить часовой пояс, функция возвращает `null` - если мы не знаем, в каком часовом поясе находимся, мы не можем узнать местное время.

setTime

```
{ } setTime(time[, callback])
```

Позволяет задавать время для архивного провайдера в минутах, часах и днях недели. Выставляет локальное время только после того, как провайдер инициализирует поля `timeZone` и `dst`.

- `timeZone` - поле, которое показывает, в каком часовом поясе на данный момент находится центр карты. При перемещении центра карты из одного часового пояса в другой локальное время может изменяться.
- `dst` - признак перехода на летнее/зимнее время. При `dst='dst'` время летнее.

Постоянной составляющей времени для провайдера пробок "обычно" служит поле `timestamp`, которое отражает текущее время в нулевом часовом поясе (UTC+0). При переходе из одного часового пояса в другой `timestamp` не изменяется. Значения полей "timestamp", "dst" и "timeZone" можно получить через поле [traffic.provider.Archive.state](#).

Параметры:

Параметр	Значение по умолчанию	Описание
<code>time</code> *	—	Тип: Object Объект с задаваемыми параметрами.

Параметр	Значение по умолчанию	Описание
<code>time.dayOfWeek</code>	—	Тип: String Сокращенное название дня недели: 'mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun'.
<code>time.hours</code>	—	Тип: Number Часы.
<code>time.minutes</code>	—	Тип: Number Минуты.
<code>callback</code>	—	Тип: Function Функция, которая вызывается после того, как было выставлено время. Принимает на вход хэш с выставленными данными.

* Обязательный параметр/опция.

Пример:

```
// Создадим элемент управления, который сразу показывает на карте
// провайдер пробок "обычно".
var trafficControl = new ymaps.control.TrafficControl({
  shown: true,
  providerKey: 'traffic#archive'
});
map.controls.add(trafficControl);
// Как только провайдер получит данные о текущей временной зоне,
// будет выставлено локальное время.
trafficControl.getProvider('traffic#archive').setTime({
  dayOfWeek: 'fri',
  hours: 9,
  minutes: 15
}, function (time) {
  alert('Локальное время ' + time.hours + ':' + time.minutes + ' set!');});
```

traffic.provider.storage

Статический объект.

Экземпляр класса [util.Storage](#)

Хранилище провайдеров.

Методы

Методы

Имя	Возвращает	Описание
<code>add(key, object)</code>	util.Storage	Добавляет объект в хранилище.
<code>get(key)</code>	Object	Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка.

Имя	Возвращает	Описание
<code>remove(key)</code>	<code>util.Storage</code>	Удаляет пару "ключ: значение" из хранилища.

util

util.AsyncStorage

Расширяет `util.Storage`.

Хранилище, которое предоставляет асинхронный доступ к значениям по ключам.

[Конструктор](#) | [Методы](#)

Конструктор

```
util.AsyncStorage()
```

Методы

Имя	Возвращает	Описание
<code>add(key, object)</code>	<code>util.Storage</code>	Добавляет объект в хранилище. Унаследован от <code>util.Storage</code> .
<code>define(key[, depends, resolveCallback[, context]])</code>	<code>util.AsyncStorage</code>	Определяет в хранилище асинхронное значение.
<code>get(key)</code>	<code>Object</code>	Возвращает объект, хранящийся под указанным ключом, или исходный <code>key</code> , в случае, если это не строка. Унаследован от <code>util.Storage</code> .
<code>isDefined(key)</code>	<code>Boolean</code>	Проверка доступности ключа в хранилище.
<code>remove(key)</code>	<code>util.Storage</code>	Удаляет пару "ключ: значение" из хранилища. Унаследован от <code>util.Storage</code> .
<code>require(keys[, successCallback[, errorCallback[, context]])</code>	<code>vow.Promise</code>	Асинхронный запрос на получение значений из хранилища.

Описание методов

define

```
{util.AsyncStorage} define(key[, depends, resolveCallback[, context]])
```

Определяет в хранилище асинхронное значение.

Возвращает ссылку на себя.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key *</code>	—	<p>Тип: String</p> <p>Ключ, по которому можно будет произвести асинхронный вызов.</p>
<code>depends</code>	—	<p>Тип: String[]</p> <p>Массив ключей значений из текущего хранилища, которые должны быть проинициализированы перед текущим. Этот аргумент может быть пропущен.</p>
<code>resolveCallback *</code>	—	<p>Тип: Function</p> <p>Функция, которая определяет значение, доступное по ключу. Первым аргументом в <code>resolveCallback</code> поступит <code>provide</code>-функция, в которую нужно будет передать значение. Вызов <code>provide</code>-функции может быть отложен во времени. Последующие аргументы — это значения из хранилища, указанные в зависимостях. Порядок модулей будет соответствовать порядку в массиве <code>depends</code>.</p>
<code>context</code>	—	<p>Тип: Object</p> <p>Контекст исполнения функции.</p>

* Обязательный параметр/опция.

Примеры:

1.

```
asyncStorage
  .define('red', function (provide) {
    provide('#FF0000');
  });
```

2.

```
asyncStorage
  .define('green', function (provide) {
    // Вызов provide-функции может быть асинхронным.
    setTimeout(function () {
      provide('#008000');
    }, 400);
  });
```

3.

```

asyncStorage
  .define('yellow', function (provide) {
    provide('#FFFF00');
  })
  // Для определения значения ключа 'violet' необходимо значение ключа 'yellow'.
  .define('violet', ['yellow'], function (provide, yellow) {
    console.log(yellow); // #FFFF00
    setTimeout(function () {
      provide('#9B30FF');
    }, 400);
  });

```

4.

```

var asyncStorage = new ymaps.util.AsyncStorage();
asyncStorage
  .define('red', function (provide) {
    provide('#FF0000');
  })
  .define('green', function (provide) {
    setTimeout(function () {
      provide('#008000');
    }, 400);
  })
  .define('yellow', function (provide) {
    provide('#FFFF00');
  })
  .define('violet', ['yellow'], function (provide, yellow) {
    setTimeout(function () {
      provide('#9B30FF');
    }, 400);
  });
// Запрашиваем
asyncStorage.require(['red', 'green', 'violet'])
  .spread(function (red, green, violet) {
    // Выводит #FF0000, #008000, #9B30FF.
    console.log(red, green, violet);
    // После первого асинхронного доступа значения можно получить через синхронный интерфейс.
    // Выводит #FF0000 #008000.
    console.log(asyncStorage.get('red'), asyncStorage.get('green'), asyncStorage.get('violet'));
    // Значение ключа 'yellow' теперь тоже содержится в хранилище,
    // так как было необходимо для определения 'violet'.

    // Выводит #FFFF00.
    console.log(asyncStorage.get('yellow'));
  });

```

isDefined

```
{Boolean} isDefined(key)
```

Проверка доступности ключа в хранилище.

Возвращает true, если ключ определен, false — иначе.

Параметры:

Параметр	Значение по умолчанию	Описание
key *	—	Тип: String Ключ значения.

* Обязательный параметр/опция.

Пример:

```

if (asyncStorage.isDefined('red')) {
  asyncStorage.require('red')
    .spread(function (red) {
      // ...
    });
}

```

require

```
{vow.Promise} require(keys[, successCallback[, errorCallback[, context]])
```

Асинхронный запрос на получение значений из хранилища.

Возвращает Promise объект, который предоставляет асинхронный доступ к значению.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>keys</code> *	—	Тип: String String[] Ключ или массив ключей.
<code>successCallback</code>	—	Тип: Function Функция обратного вызова, которая будет вызвана после получения всех значений. В функцию как аргументы будут переданы значения из хранилища. Порядок аргументов будет соответствовать порядку в массиве keys.
<code>errorCallback</code>	—	Тип: Function Функция обратного вызова, которая будет вызвана в случае ошибки. В функцию будет передан объект ошибки.
<code>context</code>	—	Тип: Object Контекст исполнения функции обратного вызова.

* Обязательный параметр/опция.

Примеры:**1.**

```
asyncStorage.require(['green'])  
  .spread(function (green) {  
    // ...  
  });
```

2.

```
var asyncStorage = new ymaps.util.AsyncStorage();  
asyncStorage  
  .define('red', function (provide) {  
    provide('#FF0000');  
  })  
  .define('green', function (provide) {  
    setTimeout(function () {  
      provide('#008000');  
    }, 400);  
  });  
// Запрос  
asyncStorage.require(['red', 'green'])  
  .spread(function (red, green) {  
    // Выводит #FF0000 #008000.
```

```
}); console.log(red, green);
```

util.augment

Статическая функция.

Базовая функция, реализующая наследование в JavaScript. Реализует наследование прототипа без исполнения конструктора родителя. К дочернему классу добавляется поле 'superclass', указывающее на прототип родительского класса, и поле 'constructor', которое указывает на конструктор класса. Через поле 'constructor' объекта 'superclass' можно обратиться к конструктору родительского класса.

Возвращает прототип дочернего класса.

```
{ Object } util.augment(ChildClass, ParentClass, override)
```

Параметры:

Параметр	Значение по умолчанию	Описание
ChildClass *	—	Тип: Function Дочерний класс.
ParentClass *	—	Тип: Function Родительский класс.
override *	—	Тип: Object Набор дополнительных полей и функций, которые будут приписаны к прототипу дочернего класса.

* Обязательный параметр/опция.

Пример:

```
// Родительский класс.
var ParentClass = function (param1, param2) {
    this.param1 = param1;
    this.param2 = param2;
};

ParentClass.prototype = {
    foo: function () {
        alert('Parent!');
    }
};

// Дочерний класс.
var ChildClass = function (param1, param2, param3) {
    // Вызываем конструктор родителя.
    ChildClass.superclass.constructor.call(this, param1, param2);
    this._param3 = param3;
}

// Наследуем ChildClass from ParentClass.
maps.util.augment(ChildClass, ParentClass, {
    // Переопределяем у наследника метод "foo".
    foo: function () {
        // Вызываем метод родительского класса.
        ChildClass.superclass.foo.call(this);
        alert('Child!');
    }
});
```

util.bind



Внимание: Данная функция является устаревшей. Используйте нативный метод bind().

Статическая функция.

Связывает переданную функцию с переданным контекстом исполнения.

Возвращает копию переданной функции с заданным контекстом исполнения `this`.

```
{ Function } util.bind(callback, context)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>callback *</code>	—	Тип: Function Функция.
<code>context *</code>	—	Тип: Object Контекст исполнения.

* Обязательный параметр/опция.

Пример:

```
var myObject = {  
    name: 'test!'  
};  
ymaps.geocode.load('Moscow')  
    .then(ymaps.util.bind(function (res) {  
        alert(this.name); // test!  
    }, myObject));
```

util.bounds

Статический объект.

Набор статических методов для работы с прямоугольными областями, представленными в виде двух противоположных точек в координатной системе проекции.

Методы

Методы

Имя	Возвращает	Описание
<code>areIntersecting(bounds1, bounds2[, projection])</code>	Boolean	Определяет пересекаются ли две прямоугольные области.
<code>containsBounds(outer, inner[, projection])</code>	Boolean	Определяет содержат ли прямоугольная область другую прямоугольную область полностью.
<code>containsPoint(bounds, point[, projection])</code>	Boolean	Определяет содержат ли прямоугольная область точку.
<code>fromBounds(sourceBounds[, projection])</code>	Number[][]	Вычисляет прямоугольную область, в которую попадают все переданные.
<code>fromGlobalPixelBounds(pixelBounds, zoom[, projection])</code>	Number[][]	Производит перевод прямоугольной области из пиксельных координат в геокоординаты с учетом масштаба.

Имя	Возвращает	Описание
<code>fromPoints(points[, projection])</code>	<code>Number[][]</code>	Вычисляет минимальную прямоугольную область, в которую попадают все переданные точки.
<code>getCenter(bounds[, projection])</code>	<code>Number[]</code>	Вычисляет центр прямоугольной области в координатной системе проекции.
<code>getCenterAndZoom(bounds, containerSize[, projection[, params]])</code>	<code>Object</code>	Вычисляет центр и уровень масштабирования, которые необходимо установить карте для того, чтобы полностью отобразить переданную область.
<code>getIntersections(bounds1, bounds2[, projection])</code>	<code>Number[][][]</code>	Возвращает все пересечения двух прямоугольных областей. В случае, если данные передаются в геокоординатах, то пересечений может быть больше одного. Если области не пересекаются, то возвращается пустой массив.
<code>getSize(bounds[, projection])</code>	<code>Number[]</code>	Вычисляет размеры прямоугольной области в координатной системе проекции.
<code>toGlobalPixelBounds(geoBounds, zoom[, projection])</code>	<code>Number[][]</code>	Производит перевод границ из геокоординат в глобальные пиксели с учетом масштаба.

Описание методов

areIntersecting

```
{Boolean} areIntersecting(bounds1, bounds2[, projection])
```

Определяет пересекаются ли две прямоугольные области.

Возвращает признак пересечения.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>bounds1 *</code>	—	Тип: <code>Number[][]</code> Первая область.
<code>bounds2 *</code>	—	Тип: <code>Number[][]</code> Вторая область.

Параметр	Значение по умолчанию	Описание
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.

* Обязательный параметр/опция.

containsBounds

```
{Boolean} containsBounds(outer, inner[, projection])
```

Определяет содержат ли прямоугольная область другую прямоугольную область полностью.

Возвращает признак попадания.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>outer *</code>	—	Тип: <code>Number[][]</code> Внешняя область
<code>inner *</code>	—	Тип: <code>Number[][]</code> Проверяемая область.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.

* Обязательный параметр/опция.

containsPoint

```
{Boolean} containsPoint(bounds, point[, projection])
```

Определяет содержат ли прямоугольная область точку.

Возвращает признак попадания.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>bounds *</code>	—	Тип: <code>Number[][]</code> Внешняя область
<code>point *</code>	—	Тип: <code>Number[]</code> Проверяемая точка.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.

* Обязательный параметр/опция.

fromBounds

```
{Number[][][]} fromBounds(sourceBounds[, projection])
```

Вычисляет прямоугольную область, в которую попадают все переданные.

Возвращает вычисленная область.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>sourceBounds</code> *	—	Тип: <code>Number[][]</code> Массив прямоугольных областей
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.

* Обязательный параметр/опция.

fromGlobalPixelBounds

```
{Number[][][]} fromGlobalPixelBounds(pixelBounds, zoom[, projection])
```

Производит перевод прямоугольной области из пиксельных координат в геокоординаты с учетом масштаба.

Возвращает вычисленные границы в геокоординатах.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>pixelBounds</code> *	—	Тип: <code>Number[][]</code> Исходные границы.
<code>zoom</code> *	—	Тип: <code>Number</code> Масштаб.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция, которая будет использоваться для вычисления геокоординат.

* Обязательный параметр/опция.

fromPoints

```
{Number[][][]} fromPoints(points[, projection])
```

Вычисляет минимальную прямоугольную область, в которую попадают все переданные точки.

Возвращает вычисленная область.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>points *</code>	—	Тип: <code>Number[][]</code> Массив точек.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.

* Обязательный параметр/опция.

getCenter

```
{Number[]} getCenter(bounds[, projection])
```

Вычисляет центр прямоугольной области в координатной системе проекции.

Возвращает точку-центр в координатной системе входящих данных.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>bounds *</code>	—	Тип: <code>Number[][]</code> Область.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.

* Обязательный параметр/опция.

getCenterAndZoom

```
{Object} getCenterAndZoom(bounds, containerSize[, projection[, params]])
```

Вычисляет центр и уровень масштабирования, которые необходимо установить карте для того, чтобы полностью отобразить переданную область.

Возвращает объект с полями `center` (`Number[]`) и `zoom` (`Number`).

Параметры:

Параметр	Значение по умолчанию	Описание
<code>bounds *</code>	—	Тип: <code>Number[][]</code> Область, заданная в географических координатах. Первая точка содержит минимальные значения широты и долготы, вторая точка - максимальные.

Параметр	Значение по умолчанию	Описание
<code>containerSize</code> *	—	Тип: <code>Number[]</code> Размер контейнера карты.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.
<code>params</code>	—	Тип: <code>Boolean Object</code> Параметры или значение опции <code>preciseZoom</code> .
<code>params.inscribe</code>	<code>true</code>	Тип: <code>Boolean</code> <code>true</code> - вписывать область в карту, <code>false</code> — вписывать карту в область.
<code>params.margin</code>	<code>0</code>	Тип: <code>Number Number[]</code> Отступы от границ видимой области карты. Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы <code>top</code> , <code>right</code> , <code>bottom</code> , <code>left</code> .
<code>params.preciseZoom</code>	<code>false</code>	Тип: <code>Boolean</code> При значении <code>false</code> уровень масштабирования будет округлен в меньшую сторону.

* Обязательный параметр/опция.

getIntersections

```
{Number[][][]} getIntersections(bounds1, bounds2[, projection])
```

Возвращает все пересечения двух прямоугольных областей. В случае, если данные передаются в геокоординатах, то пересечений может быть больше одного. Если области не пересекаются, то возвращается пустой массив.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>bounds1</code> *	—	Тип: <code>Number[][]</code> Первая область.

Параметр	Значение по умолчанию	Описание
<code>bounds2 *</code>	—	Тип: <code>Number[][]</code> Вторая область.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.

* Обязательный параметр/опция.

getSize

```
{Number[]} getSize(bounds[, projection])
```

Вычисляет размеры прямоугольной области в координатной системе проекции.

Возвращает размер области.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>bounds *</code>	—	Тип: <code>Number[][]</code> Область.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция.

* Обязательный параметр/опция.

toGlobalPixelBounds

```
{Number[][]} toGlobalPixelBounds(geoBounds, zoom[, projection])
```

Производит перевод границ из геокоординат в глобальные пиксели с учетом масштаба.

Возвращает вычисленные пиксельные границы.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>geoBounds *</code>	—	Тип: <code>Number[][]</code> Исходные границы.
<code>zoom *</code>	—	Тип: <code>Number</code> Масштаб.
<code>projection</code>	<code>projection.wgs84Mercator</code>	Тип: <code>IProjection</code> Проекция, в координатной системе которой заданы геокоординаты.

* Обязательный параметр/опция.

util.cursor

util.cursor.Accessor

Объект, предоставляющий доступ к добавленному на карту курсору.

[Конструктор](#) | [Методы](#)

Конструктор

```
util.cursor.Accessor(key)
```

Параметры:

Параметр	Значение по умолчанию	Описание
key *	—	Тип: String Ключ, под которым хранится курсор в хранилище курсоров.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
getKey()	String	Возвращает текущий ключ для доступа к курсору в хранилище курсоров.
remove()		Удаляет курсор с карты.
setKey()		Задаёт новый ключ для доступа к курсору. По этому ключу должен находиться соответствующий курсор в хранилище курсоров. В случае, если данный курсор активен, то он сразу сменится.

Описание методов

getKey

```
{String} getKey()
```

Возвращает текущий ключ для доступа к курсору в хранилище курсоров.

remove

```
{ } remove()
```

Удаляет курсор с карты.

setKey

```
{ } setKey()
```

Задаёт новый ключ для доступа к курсору. По этому ключу должен находиться соответствующий курсор в хранилище курсоров. В случае, если данный курсор активен, то он сразу сменится.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key *</code>	—	Тип: String

* Обязательный параметр/опция.

util.cursor.Manager

Менеджер курсоров.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
util.cursor.Manager(element)
```

Менеджер курсоров над DOM-элементом. Используется прямое присваивание через `style.cursor`.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>element *</code>	—	Тип: HTMLElement DOM-элемент, для которого выполняется установка курсоров.

* Обязательный параметр/опция.

Пример:

```
// Поставим курсор "стрелка" над картой.
var cursor = map.cursors.push('arrow');
setTimeout(function () {
    // Через 5 секунд установим курсору другой ключ.
    cursor.setKey('grabbing');
    setTimeout(function () {
        // Еще через 5 секунд удалим этот курсор с карты.
        cursor.remove();
    }, 5000);
}, 5000);
```

Поля

Имя	Тип	Описание
<code>events</code>	<code>event.Manager</code>	Менеджер событий менеджера курсоров.

События

Имя	Описание
<code>change</code>	Изменение курсора.

Методы

Имя	Возвращает	Описание
<code>push(key)</code>	<code>util.cursor.Accessor</code>	Устанавливает новый курсор и дописывает его в стек курсоров объекта.

Описание полей**events**

```
{event.Manager} events
```

Менеджер событий менеджера курсоров.

Описание событий**change**

Изменение курсора.

Описание методов**push**

```
{util.cursor.Accessor} push(key)
```

Устанавливает новый курсор и дописывает его в стек курсоров объекта.

Возвращает объект, предоставляющий доступ к добавленному на карту курсору.

Параметры:

Параметр	Значение по умолчанию	Описание
<code>key</code> *	—	Тип: String Курсор. Допустимые значения: <ul style="list-style-type: none">• "arrow" - стрелка;• "crosshair" - перекрестье;• "grab" - "рука";• "grabbing" - сжатая "рука"• "help" - стрелка со знаком вопроса;• "zoom" - лупа;• "move" - курсор в виде четырех стрелок;• "pointer" - "палец";• "inherit" - унаследовать курсор от родителя.

* Обязательный параметр/опция.

util.defineClass

Статическая функция.

Базовая функция, реализующая объявление классов в API Яндекс.Карт. При помощи этой функции можно объявить новый класс, указать у этого класса набор методов и произвести наследование от другого класса.

К дочернему классу приписывается поле `superclass`, указывающее на прототип родительского класса. Через поле `'constructor'` объекта `'superclass'` можно обратиться к конструктору родительского класса.

Возвращает класс.

```
{ Object } util.defineClass(constructor[, parentClass[, override]])
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>constructor</code> *	—	Тип: Function Конструктор класса.

Параметр	Значение по умолчанию	Описание
<code>parentClass</code>	—	Тип: Function Родительский класс, от которого необходимо произвести наследование. Этот аргумент может быть пропущен.
<code>override</code>	—	Тип: Object Набор дополнительных полей и функций, которые будут приписаны к прототипу класса. Источников может быть несколько (функция может иметь произвольное число параметров), данные копируются справа налево (последний аргумент имеет наивысший приоритет при копировании).

* Обязательный параметр/опция.

Примеры:

1.

```
// Объявление класса с методами.
var MyClass = ymaps.util.defineClass(function () {
  this.field = 'fieldValue';
}, {
  doSomethingAwesome: function () {
    return 'methodResult';
  },
  stop: function () {
    //...
  }
});
var object = new MyClass();
console.log(object.field); // 'fieldValue'
console.log(object.doSomethingAwesome()); // 'methodResult'
```

2.

```
// Реализуем собственный класс кнопки на основе наследования от базового класса кнопки.
// Нажатие на кнопку переключает тип тайлов карты.
var CustomControl = ymaps.util.defineClass(function () {
  // Задаем жесткий набор опций без возможности установки извне.
  var data = {
    data: { content: 'Change the map type' },
    options: {
      selectOnClick: false,
      maxWidth: 150
    }
  };
  CustomControl.superclass.constructor.call(this, data);
}, ymaps.control.Button, {
  // Определим список методов класса.

  // Переопределяем методы кнопки enable и disable.
  enable: function () {
    // Нужно обязательно вызывать методы базового класса,
    // чтобы не нарушить логику работы кнопки.
    CustomControl.superclass.enable.call(this);
    // Включение и отключение поведения кнопок.
    this.events.add('click', this.switchType, this);
  },

  disable: function () {
    this.events.remove('click', this.switchType, this);
    CustomControl.superclass.disable.call(this);
  },

  // Создаем наши собственные методы.
  switchType: function () {
    var map = this.getMap();
    if (map) {
      if (map.getType() == 'yandex#map') {
        this.setSatelliteMapType();
      }
    }
  }
});
```

```

        } else {
            this.setSchemeMapType();
        }
    },
    setSchemeMapType: function () {
        var map = this.getMap();
        if (map) {
            map.setType('yandex#map');
        }
    },
    setSatelliteMapType: function () {
        var map = this.getMap();
        if (map) {
            map.setType('yandex#satellite');
        }
    }
});
// Создание экземпляра нового класса и добавление его на карту.
var typeSwitcherButton = new CustomControl();
// В этом примере код создания карты был опущен.
myMap.controls.add(typeSwitcherButton);
// Вызов метода экземпляра класса.
typeSwitcherButton.setSatelliteMapType();

```

util.Dragger

Расширяет [IEventEmitter](#).

Специальный инструмент, который предоставляет механизм для реализации перетаскивания элементов на странице. При использовании следует обратить внимание, что во время работы драггера в системе событий API Яндекс.Карт не будут поступать события `mousemove` и `mouseup`.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

Конструктор

```
util.Dragger([params])
```

Параметры:

Параметр	Значение по умолчанию	Описание
params	—	Тип: Object Параметры драггера.
params.autoStartElement	—	Тип: HTMLElement IDomEventEmitter DOM-элемент или реализация интерфейса IDomEventEmitter, нажатие на который запустит драггер.
params.byRightButton	false	Тип: Boolean Для начала работы драггера при помощи параметра 'autoStartElement' используется правая кнопка мыши. На устройствах без поддержки мыши перетаскивание правой кнопкой не будет работать.
params.tremor	3	Тип: Number Минимальное смещение в пикселях от стартовой точки, необходимое для запуска драггера.

Пример:

```
// Пример перетаскивания DOM-элемента на карту.
// Метка создается там, где освобождается элемент DOM.
var myMap = new ymaps.Map('map', {center: [35.76, 37.67], zoom: 5});
// Элемент DOM должен иметь свойство CSS position: absolute.
var element = document.getElementById('someId');
var dragger = new ymaps.util.Dragger({
  autoStartElement: element
});
var draggerEventsGroup = dragger.events.group();

draggerEventsGroup
  .add('start', function (event) {
    var pos = event.get('position');
    positionElement(pos[0], pos[1]);
    console.log('start');
  })
  .add('move', function (event) {
    var pos = event.get('position');
    positionElement(pos[0], pos[1]);
    console.log('move');
  })
  .add('stop', function (event) {
    draggerEventsGroup.removeAll();
    element.parentElement.removeChild(element);
    // Получение географических координат в точке, где остановился тягач.
    var placemarkPosition = myMap.options.get('projection').fromGlobalPixels(
      myMap.converter.pageToGlobal(event.get('position')),
      myMap.getZoom()
    );
    myMap.geoObjects.add(
      new ymaps.Placemark(placemarkPosition)
    );
    console.log('stop');
  });

function positionElement (x, y) {
  element.style.left = x + 'px';
  element.style.top = y + 'px';
}
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter .

События

Имя	Описание
cancel	Отмена работы драггера. Это событие происходит если драггер прекратил свою работу без отправки события <code>util.Dragger.start</code> . К примеру, если было прекращено нажатие без смещения. Экземпляр класса Event . Имена полей, доступных через метод Event.get :
move	Изменение положения. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; <code>delta</code> - разница между положениями текущего и предыдущего события драггера.
start	Начало работы драггера. Это событие происходит не в момент нажатия, а при первом изменении положения после нажатия. Экземпляр класса Event . Имена полей, доступных через метод Event.get : <ul style="list-style-type: none"> <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>;

Имя	Описание
stop	<p>Завершение работы драггера. Это событие не может произойти без события <code>util.Dragger.start</code>. Экземпляр класса Event. Имена полей, доступных через метод Event.get:</p> <ul style="list-style-type: none"> <code>position</code> - координаты относительно документа. Массива вида <code>[pageX, pageY]</code>; <code>delta</code> - разница между положениями текущего и предыдущего события драггера.

Методы

Имя	Возвращает	Описание
destroy()		Останавливает работу драггера и удаляет прослушивание события "mousedown" у элемента <code>autoStartElement</code> .
isDragging()	Boolean	Возвращает работает ли сейчас драггер.
start(event)		Запускает драггер. Этот метод автоматически вызывается по событию "mousedown" элемента <code>autoStartElement</code> , если он задан. Данный метод можно использовать для инициализации по требованию. К примеру, при получении ответа от сервера. Для корректной работы при инициализации по требованию необходимо чтобы передаваемое событие поддерживало поле 'position' в методе <code>get</code> .
stop()		Останавливает драггер. Этот метод можно использовать для досрочной остановки работы драггера.

Описание событий

cancel

Отмена работы драггера. Это событие происходит если драггер прекратил свою работу без отправки события `util.Dragger.start`. К примеру, если было прекращено нажатие без смещения. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

move

Изменение положения. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- position - координаты относительно документа. Массива вида [pageX, pageY];
- delta - разница между положениями текущего и предыдущего события драггера.

start

Начало работы драггера. Это событие происходит не в момент нажатия, а при первом изменении положения после нажатия. Экземпляр класса Event. Имена полей, доступных через метод [Event.get](#):

- position - координаты относительно документа. Массива вида [pageX, pageY];

stop

Завершение работы драггера. Это событие не может произойти без события util.Dragger.start. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- position - координаты относительно документа. Массива вида [pageX, pageY];
- delta - разница между положениями текущего и предыдущего события драггера.

Описание методов**destroy**

```
{ } destroy()
```

Останавливает работу драггера и удаляет прослушивание события "mousedown" у элемента autoStartElement.

isDragging

```
{Boolean} isDragging()
```

Возвращает работает ли сейчас драггер.

start

```
{ } start(event)
```

Запускает драггер. Этот метод автоматически вызывается по событию "mousedown" элемента autoStartElement, если он задан. Данный метод можно использовать для инициализации по требованию. К примеру, при получении ответа от сервера. Для корректной работы при инициализации по требованию необходимо чтобы передаваемое событие поддерживало поле 'position' в методе get.

Параметры:

Параметр	Значение по умолчанию	Описание
event *	—	Тип: IDomEvent Инициализирующее событие.

* Обязательный параметр/опция.

stop

```
{ } stop()
```

Останавливает драггер. Этот метод можно использовать для досрочной остановки работы драггера.

util.extend

Статическая функция.

Функция, копирующая свойства из одного или нескольких JavaScript-объектов в другой JavaScript-объект.

```
util.extend(target, ...source)
```

Параметры:

Параметр	Значение по умолчанию	Описание
<code>target</code> *	—	Тип: Object Целевой JavaScript-объект. Будет модифицирован в результате работы функции.
<code>source</code> *	—	Тип: Object JavaScript-объект - источник. Все его свойства будут скопированы. Источников может быть несколько (функция может иметь произвольное число параметров), данные копируются справа налево (последний аргумент имеет наивысший приоритет при копировании).

* Обязательный параметр/опция.

Пример:

```
var options = ymaps.util.extend({
  prop1: 'a',
  prop2: 'b'
}, {
  prop2: 'c',
  prop3: 'd'
}, {
  prop3: 'e'
});
// Мы получаем результат: {
//   prop1: 'a',
//   prop2: 'c',
//   prop3: 'e'
// }
```

util.hd

Статический объект.

Позволяет работать с HD экранами различных устройств.

Методы

Методы

Имя	Возвращает	Описание
<code>getPixelRatio()</code>	Number	Возвращает отношение виртуальных пикселей на экране к физическим.

Имя	Возвращает	Описание
selectRatio(hash)	Number	Возвращает результат выборки — пиксельное соотношение, либо 1, если передана, например, строка.
selectValue(hash)	Object	Возвращает из переданного хэша объект равный или наиболее близкий к ключу соотношения пикселей. В случае, если передан не хэш, а, например, строка или функция, вернется она же.

Описание методов

getPixelRatio

```
{Number} getPixelRatio()
```

Возвращает отношение виртуальных пикселей на экране к физическим.

selectRatio

```
{Number} selectRatio(hash)
```

Возвращает результат выборки — пиксельное соотношение, либо 1, если передана, например, строка.

Параметры:

Параметр	Значение по умолчанию	Описание
hash *	—	Тип: IRatioMap Объект типа IRatioMap для экранов с различным пиксельным соотношением.

* Обязательный параметр/опция.

selectValue

```
{Object} selectValue(hash)
```

Возвращает из переданного хэша объект равный или наиболее близкий к ключу соотношения пикселей. В случае, если передан не хэш, а, например, строка или функция, вернется она же.

Параметры:

Параметр	Значение по умолчанию	Описание
hash *	—	Тип: Object IRatioMap Объект типа IRatioMap для экранов с различным пиксельным соотношением.

* Обязательный параметр/опция.

Пример:

```
// Нам нужно добавить хорошо выглядящую на HD-экранах картинку на балун.
var balloonImages = {
  // Обычная картинка.
  "1": "100x100.png",
  // Картинка для Retina-экранов.
  "2": "200x200.png"
};
var img = ymaps.util.hd.selectValue(balloonImages);
// Добавляем точку на карту.
myMap.geoObjects.add(new ymaps.GeoObject({
  geometry: "Point",
  coordinates: [55, 37],
  properties: {
    balloonContent:
      'first</code> *  | —                     | Тип: Number[]<br><br>Массив для сравнения.             |
| <code>second</code> * | —                     | Тип: Number[]<br><br>Массив, который будем сравнивать. |
| <code>diff</code>     | —                     | Тип: Number<br><br>Точность сравнения.                 |

\* Обязательный параметр/опция.

**util.math.cycleRestrict**

Статическая функция.

Приводит цифровое значение к заданному диапазону. Считается, что диапазон значений замкнут в кольцо. Если значение выходит за одну из границ диапазона, то излишек отсчитывается по кругу, со стороны другой границы.

**Возвращает** ограниченное значение.

```
{ Number } util.math.cycleRestrict(value, min, max)
```

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                                |
|----------------------|-----------------------|-----------------------------------------|
| <code>value</code> * | —                     | Тип: Number<br>Ограничиваемое значение. |
| <code>min</code> *   | —                     | Тип: Number<br>Минимальный предел.      |
| <code>max</code> *   | —                     | Тип: Number<br>Максимальный предел.     |

\* Обязательный параметр/опция.

#### Пример:

```
// Возвращает 110
ymaps.util.math.cycleRestrict(-250, -180, 180);
// Возвращает 60
ymaps.util.math.cycleRestrict(-300, -180, 180);
// Возвращает -170
ymaps.util.math.cycleRestrict(190, -180, 180);
```

#### **util.math.restrict**

Статическая функция.

Ограничивает цифровое значение на входе заданными минимальным и максимальным пределами.

**Возвращает** ограниченное значение.

```
{ Number } util.math.restrict(value, min, max)
```

#### Параметры:

| Параметр             | Значение по умолчанию | Описание                                |
|----------------------|-----------------------|-----------------------------------------|
| <code>value</code> * | —                     | Тип: Number<br>Ограничиваемое значение. |
| <code>min</code> *   | —                     | Тип: Number<br>Минимальный предел.      |
| <code>max</code> *   | —                     | Тип: Number<br>Максимальный предел.     |

\* Обязательный параметр/опция.

#### Пример:

```
// Приведенный ниже код возвращает "-180".
ymaps.util.math.restrict(-250, -180, 180);
```

#### **util.pixelBounds**

Статический объект.

[Методы](#)

## Методы

| Имя                                            | Возвращает      | Описание                                                                             |
|------------------------------------------------|-----------------|--------------------------------------------------------------------------------------|
| <code>areIntersecting(bounds1, bounds2)</code> | Boolean         | Определяет пересекаются ли две прямоугольные области.                                |
| <code>containsBounds(outer, inner)</code>      | Boolean         | Определяет содержат ли прямоугольная область другую прямоугольную область полностью. |
| <code>containsPoint(bounds, point)</code>      | Boolean         | Определяет содержат ли прямоугольная область точку.                                  |
| <code>fromBounds(sourceBounds)</code>          | Number[][]      | Вычисляет прямоугольную область, в которую попадают все переданные.                  |
| <code>fromPoints(points)</code>                | Number[][]      | Вычисляет прямоугольную область, в которую попадают переданные точки.                |
| <code>getCenter(bounds)</code>                 | Number[]        | Вычисляет центр прямоугольной области.                                               |
| <code>getIntersection(bounds1, bounds2)</code> | Number[][] null | Вычисляет пересечение двух прямоугольных областей.                                   |
| <code>getSize(bounds)</code>                   | Number[]        | Вычисляет размеры прямоугольной области.                                             |

## Описание методов

**areIntersecting**

```
{Boolean} areIntersecting(bounds1, bounds2)
```

Определяет пересекаются ли две прямоугольные области.

**Возвращает** признак пересечения.

**Параметры:**

| Параметр               | Значение по умолчанию | Описание                           |
|------------------------|-----------------------|------------------------------------|
| <code>bounds1</code> * | —                     | Тип: Number[][]<br>Первая область. |
| <code>bounds2</code> * | —                     | Тип: Number[][]<br>Вторая область. |

\* Обязательный параметр/опция.

**containsBounds**

```
{Boolean} containsBounds(outer, inner)
```

Определяет содержат ли прямоугольная область другую прямоугольную область полностью.

**Возвращает** признак попадания.

**Параметры:**

| Параметр | Значение по умолчанию | Описание                                  |
|----------|-----------------------|-------------------------------------------|
| outer *  | —                     | Тип: Number[]<br><br>Внешняя область      |
| inner *  | —                     | Тип: Number[]<br><br>Проверяемая область. |

\* Обязательный параметр/опция.

**containsPoint**

```
{Boolean} containsPoint(bounds, point)
```

Определяет содержат ли прямоугольная область точку.

**Возвращает** признак попадания.

**Параметры:**

| Параметр | Значение по умолчанию | Описание                                |
|----------|-----------------------|-----------------------------------------|
| bounds * | —                     | Тип: Number[]<br><br>Внешняя область    |
| point *  | —                     | Тип: Number[]<br><br>Проверяемая точка. |

\* Обязательный параметр/опция.

**fromBounds**

```
{Number[][]} fromBounds(sourceBounds)
```

Вычисляет прямоугольную область, в которую попадают все переданные.

**Возвращает** вычисленная область.

**Параметры:**

| Параметр       | Значение по умолчанию | Описание                                             |
|----------------|-----------------------|------------------------------------------------------|
| sourceBounds * | —                     | Тип: Number[][]<br><br>Массив прямоугольных областей |

\* Обязательный параметр/опция.

### fromPoints

```
{Number[][]} fromPoints(points)
```

Вычисляет прямоугольную область, в которую попадают переданные точки.

**Возвращает** вычисленная область.

#### Параметры:

| Параметр        | Значение по умолчанию | Описание                         |
|-----------------|-----------------------|----------------------------------|
| <b>points</b> * | —                     | Тип: Number[][]<br>Массив точек. |

\* Обязательный параметр/опция.

### getCenter

```
{Number[]} getCenter(bounds)
```

Вычисляет центр прямоугольной области.

**Возвращает** точку-центр в координатной системе входящих данных.

#### Параметры:

| Параметр        | Значение по умолчанию | Описание                    |
|-----------------|-----------------------|-----------------------------|
| <b>bounds</b> * | —                     | Тип: Number[][]<br>Область. |

\* Обязательный параметр/опция.

### getIntersection

```
{Number[][]|null} getIntersection(bounds1, bounds2)
```

Вычисляет пересечение двух прямоугольных областей.

**Возвращает** прямоугольную область, которая образуется пересечением переданных, либо Null, если области не пересекаются.

#### Параметры:

| Параметр         | Значение по умолчанию | Описание                           |
|------------------|-----------------------|------------------------------------|
| <b>bounds1</b> * | —                     | Тип: Number[][]<br>Первая область. |
| <b>bounds2</b> * | —                     | Тип: Number[][]<br>Вторая область. |

\* Обязательный параметр/опция.

## getSize

```
{Number[]} getSize(bounds)
```

Вычисляет размеры прямоугольной области.

**Возвращает** размер области.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                     |
|-----------------------|-----------------------|----------------------------------------------|
| <code>bounds</code> * | —                     | Тип: <code>Number[][]</code><br><br>Область. |

\* Обязательный параметр/опция.

## util.requireCenterAndZoom

Статическая функция.

Вычисляет оптимальный центр и уровень масштабирования карты для отображения переданной области на указанном типе карты. Уровень масштаба будет в рамках доступного диапазона масштабов.

**Возвращает** Promise объект, который будет подтвержден объектом с полями `center` и `zoom` или будет отклонен с ошибкой.

```
{ vow.Promise } util.requireCenterAndZoom(mapType, bounds, containerSize[, params])
```

**Параметры:**

| Параметр                     | Значение по умолчанию | Описание                                                                                                                                                                                                                       |
|------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>mapType</code> *       | —                     | Тип: <code>String MapType map.ZoomRange</code><br><br>Тип карты. Строка-ключ из <code>mapType.storage</code> , либо экземпляр класса <code>MapType</code> . Или менеджер коэффициентов масштабирования карты конкретной карты. |
| <code>bounds</code> *        | —                     | Тип: <code>Number[][]</code><br><br>Область, заданная в географических координатах. Первая точка содержит минимальные значения широты и долготы, вторая точка - максимальные.                                                  |
| <code>containerSize</code> * | —                     | Тип: <code>Number[]</code><br><br>Размер контейнера карты.                                                                                                                                                                     |
| <code>params</code>          | —                     | Тип: <code>Object</code><br><br>Параметры.                                                                                                                                                                                     |

| Параметр                           | Значение по умолчанию | Описание                                                                                                                                                                                                                                                                               |
|------------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">params.inscribe</a>    | true                  | Тип: Boolean<br><br>true - вписывать область в карту, false — вписывать карту в область.                                                                                                                                                                                               |
| <a href="#">params.margin</a>      | 0                     | Тип: Number Number[]<br><br>Отступы от границ видимой области карты. Если задано одно число - оно применяется ко всем сторонам. Если задано два - то это горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы top, right, bottom, left. |
| <a href="#">params.preciseZoom</a> | false                 | Тип: Boolean<br><br>При значении false уровень масштабирования будет округлен в меньшую сторону.                                                                                                                                                                                       |

\* Обязательный параметр/опция.

#### Пример:

```
// Находим оптимальный центр и уровень масштабирования карты.
ymaps.util.requireCenterAndZoom(
 myMap.getType(),
 [[50.531219, 31.278264], [50.966841, 31.964909]],
 myMap.container.getSize()
).then(function (result) {
 // Устанавливаем карте оптимальный центр и зум.
 myMap.setCenter(result.center, result.zoom);
});
```

## util.Storage

Хранилище объектов по ключу.

[Конструктор](#) | [Методы](#)

### Конструктор

```
util.Storage()
```

### Методы

| Имя                              | Возвращает                   | Описание                                                                                            |
|----------------------------------|------------------------------|-----------------------------------------------------------------------------------------------------|
| <a href="#">add(key, object)</a> | <a href="#">util.Storage</a> | Добавляет объект в хранилище.                                                                       |
| <a href="#">get(key)</a>         | Object                       | Возвращает объект, хранящийся под указанным ключом, или исходный key, в случае, если это не строка. |
| <a href="#">remove(key)</a>      | <a href="#">util.Storage</a> | Удаляет пару "ключ: значение" из хранилища.                                                         |



**Описание методов****add**

```
{util.Storage} add(key, object)
```

Добавляет объект в хранилище.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                        |
|-----------------------|-----------------------|---------------------------------|
| <code>key *</code>    | —                     | Тип: String<br>Ключ.            |
| <code>object *</code> | —                     | Тип: Object<br>Хранимый объект. |

\* Обязательный параметр/опция.

**get**

```
{Object} get(key)
```

**Возвращает** объект, хранящийся под указанным ключом или исходный key, в случае, если это не строка.

**Параметры:**

| Параметр           | Значение по умолчанию | Описание                    |
|--------------------|-----------------------|-----------------------------|
| <code>key *</code> | —                     | Тип: String Object<br>Ключ. |

\* Обязательный параметр/опция.

**remove**

```
{util.Storage} remove(key)
```

Удаляет пару "ключ: значение" из хранилища.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр           | Значение по умолчанию | Описание             |
|--------------------|-----------------------|----------------------|
| <code>key *</code> | —                     | Тип: String<br>Ключ. |

\* Обязательный параметр/опция.

## vow

Статический объект.

Содержит методы для создания и обработки объектов-promise.

### Примечание:

Данный класс является частью библиотеки [Vow](#).

Ниже описаны только некоторые методы. Полный список методов доступен здесь: <http://dfilatov.github.io/vow/>.

Copyright (c) 2012-2013 Filatov Dmitry ([dfilatov@yandex-team.ru](mailto:dfilatov@yandex-team.ru)). Dual licensed under the [MIT](#) and [GPL](#) licenses.

### Методы

#### Методы

| Имя                            | Возвращает                   | Описание                                                                                                                         |
|--------------------------------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">all(iterable)</a>  | <a href="#">vow.Promise</a>  | Возвращает объект-promise, который будет разрешен или отклонён только когда будут разрешены или отклонены все указанные объекты. |
| <a href="#">defer()</a>        | <a href="#">vow.Deferred</a> | Создаёт новый deferred-объект. Эквивалент для `new ymaps.vow.Deferred()`.                                                        |
| <a href="#">reject(reason)</a> | <a href="#">vow.Promise</a>  | Возвращает объект-promise, отклонённый с указанной причиной.                                                                     |
| <a href="#">resolve(value)</a> | <a href="#">vow.Promise</a>  | Возвращает Promise объект, разрешённый указанным значением.                                                                      |

## Описание методов

### all

```
{vow.Promise} all(iterable)
```

**Возвращает** объект-promise, который будет разрешен или отклонён только когда будут разрешены или отклонены все указанные объекты.

### Параметры:

| Параметр                   | Значение по умолчанию | Описание                                                           |
|----------------------------|-----------------------|--------------------------------------------------------------------|
| <a href="#">iterable</a> * | —                     | Тип: Object Object[]<br><br>Набор объектов-promise и/или значений. |

\* Обязательный параметр/опция.

**Примеры:****1.**

```
var deferred1 = ymaps.vow.defer(),
 deferred2 = ymaps.vow.defer();

ymaps.vow.all([deferred1.promise(), deferred2.promise(), 3])
 .then(function(value) {
 // value => [1, 2, 3]
 });

deferred1.resolve(1);
deferred2.resolve(2);
```

**2.**

```
var deferred1 = ymaps.vow.defer(),
 deferred2 = ymaps.vow.defer();

ymaps.vow.all({ p1 : deferred1.promise(), p2 : deferred2.promise(), p3 : 3 })
 .then(function(value) {
 // value => { p1 : 1, p2 : 2, p3 : 3 }
 });

deferred1.resolve(1);
deferred2.resolve(2);
```

**defer**

```
{vow.Deferred} defer()
```

Создаёт новый deferred-объект. Эквивалент для `new ymaps.vow.Deferred()`.

См.: [vow.Deferred](#)

**Возвращает** объект-deferred.

**reject**

```
{vow.Promise} reject(reason)
```

**Возвращает** объект-promise, отклонённый с указанной причиной.

**Параметры:**

| Параметр                 | Значение по умолчанию | Описание                           |
|--------------------------|-----------------------|------------------------------------|
| <a href="#">reason</a> * | —                     | Тип: Object<br>Причина отклонения. |

\* Обязательный параметр/опция.

**resolve**

```
{vow.Promise} resolve(value)
```

**Возвращает** Promise объект, разрешённый указанным значением.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                 |
|-------------------------|-----------------------|--------------------------|
| <a href="#">value</a> * | —                     | Тип: Object<br>Значение. |

\* Обязательный параметр/опция.

## vow.Deferred

Класс, описывающий объекты-deferred.

**Примечание:** Данный класс является частью библиотеки [Vow](#). Ниже описаны только некоторые методы. Полный список методов доступен здесь: <http://dfilatov.github.io/vow/>. Copyright (c) 2012-2013 Filatov Dmitry ([dfilatov@yandex-team.ru](mailto:dfilatov@yandex-team.ru)). Dual licensed under the [MIT](#) and [GPL](#) licenses.

**Примечание:** Не является отдельным модулем: доступен после подключения модуля [vow](#).

[Конструктор](#) | [Методы](#)

### Конструктор

```
vow.Deferred()
```

Создаёт объект-deferred.

**Пример:**

```
function someAsyncMethod () {
 var deferred = new ymaps.vow.Deferred();
 // или:
 // `var deferred = ymaps.vow.defer();`

 doSomeAsyncStuff(function (err, value) {
 if (err) {
 deferred.reject(err);
 return;
 }

 deferred.resolve(value);
 });

 return deferred.promise();
}

someAsyncMethod().then(function (value) {
 console.log('The method result: ' + value);
}, function (err) {
 console.log('Error: ' + err);
});
```

### Методы

| Имя                            | Возвращает                  | Описание                                                 |
|--------------------------------|-----------------------------|----------------------------------------------------------|
| <a href="#">promise()</a>      | <a href="#">vow.Promise</a> | Возвращает связанный объект-promise.                     |
| <a href="#">reject(reason)</a> |                             | Отклоняет связанный объект-promise с указанной причиной. |
| <a href="#">resolve(value)</a> |                             | Разрешает связанный объект-promise указанным значением.  |

### Описание методов

#### promise

```
{vow.Promise} promise()
```

**Возвращает** связанный объект-promise.

**reject**

```
{ } reject(reason)
```

Отклоняет связанный объект-promise с указанной причиной.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                           |
|-----------------------|-----------------------|------------------------------------|
| <code>reason</code> * | —                     | Тип: Object<br>Причина отклонения. |

\* Обязательный параметр/опция.

**resolve**

```
{ } resolve(value)
```

Разрешает связанный объект-promise указанным значением.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                 |
|----------------------|-----------------------|--------------------------|
| <code>value</code> * | —                     | Тип: Object<br>Значение. |

\* Обязательный параметр/опция.

**vow.Promise**

Класс, описывающий объекты-promise.

[Спецификация Promise/A+](#).

**Примечание:** Данный класс является частью библиотеки [Vow](#). Ниже описаны только некоторые методы. Полный список методов доступен здесь: <http://dfilatov.github.io/vow/>. Copyright (c) 2012-2013 Filatov Dmitry ([dfilatov@yandex-team.ru](mailto:dfilatov@yandex-team.ru)). Dual licensed under the [MIT](#) and [GPL](#) licenses.

**Примечание:** Не является отдельным модулем: доступен после подключения модуля [vow](#).

[Конструктор](#) | [Методы](#)

**Конструктор**

```
vow.Promise([resolver])
```

Создаёт объект-promise.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                                                                                                                                                        |
|-----------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>resolver</code> | —                     | <p>Тип: Function</p> <p>Функция, которая принимает в качестве параметров методы <code>resolve</code> и <code>reject</code> для установки состояния и значения создаваемому объекту-promise.</p> |

**Пример:**

```
function someAsyncMethod () {
 return new ymaps.vow.Promise(function (resolve, reject) {
 doSomeAsyncStuff(function (err, value) {
 if (err) {
 reject(err);
 return;
 }
 resolve(value);
 });
 });
}

someAsyncMethod().then(function (value) {
 console.log('Результат метода: ' + value);
}, function (err) {
 console.log('Error: ' + err);
});
```

**Методы**

| Имя                                                                 | Возвращает               | Описание                                                                                                                                                                                                                                                |
|---------------------------------------------------------------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>done([onFulfilled[, onRejected[, onProgress[, ctx]]]])</code> |                          | Аналог метода <code>vow.Promise.then</code> , завершающий цепочку промисов. Кидает исключение в случае отклонения объекта-promise.                                                                                                                      |
| <code>spread([onFulfilled[, onRejected[, ctx]]])</code>             | <code>vow.Promise</code> | Аналог метода <code>vow.Promise.then</code> , вызывающий функции обратного вызова с набором аргументов, соответствующим массиву, которым будет разрешён/отклонён объект-promise. Обычно используется в сочетании с методами типа <code>vow.all</code> . |
| <code>then([onFulfilled[, onRejected[, onProgress[, ctx]]]])</code> | <code>vow.Promise</code> | Задаёт функцию-обработчик для объекта-promise.                                                                                                                                                                                                          |
| <code>valueOf()</code>                                              | Object                   | Возвращает значение для разрешённого объекта-promise или причину отклонения для отклонённого.                                                                                                                                                           |

## Описание методов

### done

```
{ } done([onFulfilled[, onRejected[, onProgress[, ctx]]]])
```

Аналог метода [vow.Promise.then](#), завершающий цепочку промисов. Кидает исключение в случае отклонения объекта-promise.

#### Параметры:

| Параметр                    | Значение по умолчанию | Описание                                                                                                          |
|-----------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------|
| <a href="#">onFulfilled</a> | —                     | Тип: Function<br><br>Функция обратного вызова, которая будет вызвана в случае если объект-promise будет разрешён. |
| <a href="#">onRejected</a>  | —                     | Тип: Function<br><br>Функция обратного вызова, которая будет вызвана в случае если объект-promise будет отклонён. |
| <a href="#">onProgress</a>  | —                     | Тип: Function<br><br>Функция обратного вызова, которая будет вызвана при "оповещении" объекта-promise.            |
| <a href="#">ctx</a>         | —                     | Тип: Object<br><br>Контекст выполнения функций обратного вызова.                                                  |

#### Пример:

```
var deferred = ymaps.vow.defer();
deferred.reject(Error('Internal error'));
deferred.promise().done(); // Будет брошено исключение.
```

### spread

```
{vow.Promise} spread([onFulfilled[, onRejected[, ctx]])
```

Аналог метода [vow.Promise.then](#), вызывающий функции обратного вызова с набором аргументов, соответствующим массиву, которым будет разрешён/отклонён объект-promise. Обычно используется в сочетании с методами типа [vow.all](#).

**Возвращает** новый объект-promise.

#### Параметры:

| Параметр                 | Значение по умолчанию | Описание                                                                                                          |
|--------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>onFulfilled</code> | —                     | Тип: Function<br><br>Функция обратного вызова, которая будет вызвана в случае если объект-promise будет разрешён. |
| <code>onRejected</code>  | —                     | Тип: Function<br><br>Функция обратного вызова, которая будет вызвана в случае если объект-promise будет отклонён. |
| <code>ctx</code>         | —                     | Тип: Object<br><br>Контекст выполнения функций обратного вызова.                                                  |

**Пример:**

```
var deferred1 = ymaps.vow.defer();
var deferred2 = ymaps.vow.defer();

ymaps.vow.all([deferred1.promise(), deferred2.promise()]).spread(function(arg1, arg2) {
 // arg1 => 1, arg2 => 'два'
});

deferred1.resolve(1);
deferred2.resolve('два');
```

**then**

```
{vow.Promise} then([onFulfilled[, onRejected[, onProgress[, ctx]]]])
```

Задаёт функцию-обработчик для объекта-promise.

**Возвращает** новый объект-promise. См. [Спецификацию Promise/A+](#).

**Параметры:**

| Параметр                 | Значение по умолчанию | Описание                                                                                                          |
|--------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>onFulfilled</code> | —                     | Тип: Function<br><br>Функция обратного вызова, которая будет вызвана в случае если объект-promise будет разрешён. |
| <code>onRejected</code>  | —                     | Тип: Function<br><br>Функция обратного вызова, которая будет вызвана в случае если объект-promise будет отклонён. |



| Параметр                   | Значение по умолчанию | Описание                                                                                               |
|----------------------------|-----------------------|--------------------------------------------------------------------------------------------------------|
| <a href="#">onProgress</a> | —                     | Тип: Function<br><br>Функция обратного вызова, которая будет вызвана при "оповещении" объекта-promise. |
| <a href="#">ctx</a>        | —                     | Тип: Object<br><br>Контекст выполнения функций обратного вызова.                                       |

### valueOf

```
{Object} valueOf()
```

Возвращает значение для разрешённого объекта-promise или причину отклонения для отклонённого.

## Интерфейсы

### IBalloon

Расширяет [IPopup](#).

Интерфейс балуна.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

#### Конструктор

```
IBalloon()
```

#### Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

#### События

| Имя                           | Описание                                                                       |
|-------------------------------|--------------------------------------------------------------------------------|
| <a href="#">close</a>         | Закрытие инфо-объекта.<br>Унаследовано от <a href="#">IPopup</a> .             |
| <a href="#">open</a>          | Открытие инфо-объекта.<br>Унаследовано от <a href="#">IPopup</a> .             |
| <a href="#">optionschange</a> | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> . |

## Методы

| Имя                                   | Возвращает               | Описание                                                                                                                                                                                                                                                       |
|---------------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>autoPan()</code>                | <code>vow.Promise</code> | Перемещает карту таким образом, чтобы балун оказался видимым.                                                                                                                                                                                                  |
| <code>close([force])</code>           | <code>vow.Promise</code> | Закрывает инфо-объект.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                                              |
| <code>getData()</code>                |                          | Возвращает данные инфо-объекта.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                                     |
| <code>getOverlay()</code>             | <code>vow.Promise</code> | Возвращает объект-обещание вернуть оверлей.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                         |
| <code>getOverlaySync()</code>         | <code>IOverlay</code>    | Возвращает оверлей, если тот существует.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                            |
| <code>getPosition()</code>            |                          | Возвращает координаты инфо-объекта.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                                 |
| <code>isOpen()</code>                 | <code>Boolean</code>     | Возвращает состояние инфо-объекта: открыт/закрыт.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                   |
| <code>open([position[, data]])</code> | <code>vow.Promise</code> | Открывает инфо-объект в указанной позиции. Если инфо-объект уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией <a href="#">IProjection</a> , которая содержится в опциях.<br>Унаследован от <a href="#">IPopup</a> . |
| <code>setData(data)</code>            | <code>vow.Promise</code> | Задаёт инфо-объекту новые данные.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                                   |
| <code>setPosition(position)</code>    | <code>vow.Promise</code> | Задаёт инфо-объекту новую позицию.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                                  |

## Описание методов

**autoPan**

```
{vow.Promise} autoPan()
```

Перемещает карту таким образом, чтобы балун оказался видимым.

**Возвращает** объект-promise.

## IBalloonLayout

Расширяет [ILayout](#).

Интерфейс макета балуна.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IBalloonLayout()
```

### Поля

| Имя                    | Тип                           | Описание                                                                |
|------------------------|-------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

### События

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>           | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">contextmenu</a>     | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">dblclick</a>        | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <a href="#">emptinesschange</a> | Изменение признака пустоты макета. Экземпляр класса <a href="#">Event</a> .<br>Унаследовано от <a href="#">ILayout</a> .                                                                                                                                                                                                     |
| <a href="#">mousedown</a>       | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mouseenter</a>      | Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                      |
| <a href="#">mouseleave</a>      | Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mousemove</a>           | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">mouseup</a>             | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">multitouchend</a>       | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchmove</a>      | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a>     | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">parentelementchange</a> | <p>Изменение родительского элемента. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">shapechange</a>         | <p>Изменение фигуры области охватывающей макет. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <a href="#">userclose</a>           | <p>Нажата кнопка закрытия. Пользователь закрывает балун. Реализует интерфейс <a href="#">IEvent</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| Имя                   | Описание                                                                                                                                                                                                                                                                                           |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">wheel</a> | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEventManager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |

## Методы

| Имя                                      | Возвращает                   | Описание                                                                                                                                                                                                                                                                            |
|------------------------------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>                |                              | <p>Деструктор. Вызывается при завершении работы с макетом.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                                                                                                                       |
| <a href="#">getData()</a>                | Object                       | <p>Возвращает объект данных макета.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                                                                                                                                              |
| <a href="#">getParentElement()</a>       | HTMLElement                  | <p>Возвращает родительский HTML-элемент.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                                                                                                                                         |
| <a href="#">getShape()</a>               | <a href="#">IShape</a>  null | <p>Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.</p> <p>Унаследован от <a href="#">ILayout</a>.</p> |
| <a href="#">isEmpty()</a>                | Boolean                      | <p>Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                        |
| <a href="#">setData(data)</a>            |                              | <p>Задаёт данные макета.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                         |
| <a href="#">setParentElement(parent)</a> |                              | <p>Добавляет макет в DOM-дерево.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                 |

## Описание событий

### userclose

Нажата кнопка закрытия. Пользователь закрывает балун. Реализует интерфейс [IEvent](#).

## IBalloonManager

Расширяет [IPopupManager](#).

Интерфейс менеджера балуна.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
IBalloonManager()
```

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## События

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">autopanbegin</a>    | Начало автоматического перемещения центра карты, вызванного методом <a href="#">autoPan</a> . Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>target</code> - ссылка на объект <a href="#">IBalloonOwner</a>.</li> </ul>                                                                            |
| <a href="#">autopanend</a>      | Окончание автоматического перемещения центра карты, вызванного методом <a href="#">autoPan</a> . Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>target</code> - ссылка на объект <a href="#">IBalloonOwner</a>.</li> </ul>                                                                         |
| <a href="#">beforeuserclose</a> | Событие, предшествующее событию <a href="#">Balloon.userclose</a> . Позволяет отменить пользовательское действие посредством вызова метода <code>preventDefault</code> . Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>target</code> - ссылка на объект <a href="#">IBalloonOwner</a>.</li> </ul> |
| <a href="#">close</a>           | Заккрытие инфо-объекта. Названия полей доступны через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>target</code> - ссылка на объект, на котором был закрыт инфо-объект.</li> </ul> Унаследовано от <a href="#">IPopupManager</a> .                                                                                                                                     |
| <a href="#">open</a>            | Открытие инфо-объекта. Названия полей доступны через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>target</code> - ссылка на объект, на котором был открыт инфо-объект.</li> </ul> Унаследовано от <a href="#">IPopupManager</a> .                                                                                                                                      |
| <a href="#">userclose</a>       | Заккрытие балуна пользователем. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>target</code> - ссылка на объект <a href="#">IBalloonOwner</a>.</li> </ul>                                                                                                                                          |

## Методы

| Имя                            | Возвращает                  | Описание                                                                 |
|--------------------------------|-----------------------------|--------------------------------------------------------------------------|
| <a href="#">autoPan()</a>      | <a href="#">vow.Promise</a> | Перемещает карту таким образом, чтобы балун оказался видимым.            |
| <a href="#">close([force])</a> | <a href="#">vow.Promise</a> | Закрывает инфо-объект.<br>Унаследован от <a href="#">IPopupManager</a> . |

| Имя                                                | Возвращает                           | Описание                                                                                            |
|----------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>                          |                                      | Отключает менеджер инфо-объекта.<br>Унаследован от <a href="#">IPopupManager</a> .                  |
| <a href="#">getData()</a>                          | Object null                          | Возвращает данные инфо-объекта или null.<br>Унаследован от <a href="#">IPopupManager</a> .          |
| <a href="#">getOptions()</a>                       | <a href="#">IOptionManager</a>  null | Возвращает менеджер опций или null.<br>Унаследован от <a href="#">IPopupManager</a> .               |
| <a href="#">getOverlay()</a>                       | <a href="#">vow.Promise</a>          | Возвращает объект-обещание вернуть оверлей.<br>Унаследован от <a href="#">IPopupManager</a> .       |
| <a href="#">getOverlaySync()</a>                   | <a href="#">IOverlay</a>  null       | Возвращает оверлей, если тот существует.<br>Унаследован от <a href="#">IPopupManager</a> .          |
| <a href="#">getPosition()</a>                      | Number[] null                        | Возвращает координаты инфо-объекта или 'null'.<br>Унаследован от <a href="#">IPopupManager</a> .    |
| <a href="#">isOpen()</a>                           | Boolean                              | Возвращает состояние инфо-объекта: открыт/закрыт.<br>Унаследован от <a href="#">IPopupManager</a> . |
| <a href="#">open([position[, data[, options]])</a> | <a href="#">vow.Promise</a>          | Открывает инфо-объект в указанной позиции.<br>Унаследован от <a href="#">IPopupManager</a> .        |
| <a href="#">setData(data)</a>                      | <a href="#">vow.Promise</a>          | Задаёт инфо-объекту новые данные.<br>Унаследован от <a href="#">IPopupManager</a> .                 |
| <a href="#">setOptions(options)</a>                | <a href="#">vow.Promise</a>          | Задаёт инфо-объекту новые опции.<br>Унаследован от <a href="#">IPopupManager</a> .                  |
| <a href="#">setPosition(position)</a>              | <a href="#">vow.Promise</a>          | Задаёт инфо-объекту новую позицию.<br>Унаследован от <a href="#">IPopupManager</a> .                |

## Описание событий

### autopanbegin

Начало автоматического перемещения центра карты, вызванного методом `autoPan`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `target` - ссылка на объект [IBalloonOwner](#).

### autopanend

Окончание автоматического перемещения центра карты, вызванного методом `autoPan`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `target` - ссылка на объект [IBalloonOwner](#).

### beforeuserclose

Событие, предшествующее событию [Balloon.userclose](#). Позволяет отменить пользовательское действие посредством вызова метода `preventDefault`. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `target` - ссылка на объект [IBalloonOwner](#).

### userclose

Заккрытие балуна пользователем. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `target` - ссылка на объект [IBalloonOwner](#).

## Описание методов

### autoPan

```
{vow.Promise} autoPan()
```

Перемещает карту таким образом, чтобы балун оказался видимым.

**Возвращает** объект-promise.

## IBalloonOwner

Объект, имеющий балун, доступный через свойство "balloon".

[Конструктор](#) | [Поля](#) | [События](#)

### Конструктор

```
IBalloonOwner()
```

### Поля

| Имя                     | Тип                             | Описание       |
|-------------------------|---------------------------------|----------------|
| <a href="#">balloon</a> | <a href="#">IBalloonManager</a> | Балун объекта. |

### События

| Имя                          | Описание          |
|------------------------------|-------------------|
| <a href="#">balloonclose</a> | Заккрытие балуна. |
| <a href="#">balloonopen</a>  | Открытие балуна.  |



**Описание полей****balloon**

```
{IBalloonManager} balloon
```

Балун объекта.

**Описание событий****balloonclose**

Закрытие балуна.

**balloonopen**

Открытие балуна.

**IBaseCircleGeometry**

Расширяет [IBaseGeometry](#), [ICircleGeometryAccess](#).

Интерфейс геометрии "Круг".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
IBaseCircleGeometry()
```

**Поля**

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

**События**

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> - старые координаты центра;</li><li><code>newCoordinates</code> - новые координаты центра;</li><li><code>oldRadius</code> - старый радиус;</li><li><code>newRadius</code> - новый радиус.</li></ul> Унаследовано от <a href="#">ICircleGeometryAccess</a> . |

## Методы

| Имя                                         | Возвращает                            | Описание                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a>          | Boolean                               | <p>Проверяет, лежит ли переданная точка внутри круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                                                                         |
| <a href="#">freeze()</a>                    | <a href="#">IFreezable</a>            | <p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p>                                                                                                                                                                                                              |
| <a href="#">getBounds()</a>                 | Number[][] null                       | <p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.</p> <p>Унаследован от <a href="#">IBaseGeometry</a>.</p> |
| <a href="#">getClosest(anchorPosition)</a>  | Object                                | <p>Ищет на окружности точку, ближайшую к anchorPosition.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                                                                      |
| <a href="#">getCoordinates()</a>            | Number[][] null                       | <p>Возвращает координаты центра круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                                                                                        |
| <a href="#">getRadius()</a>                 | Number                                | <p>Возвращает радиус круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                                                                                                   |
| <a href="#">getType()</a>                   | String                                | <p>Возвращает строку "Circle".</p>                                                                                                                                                                                                                                                                             |
| <a href="#">isFrozen()</a>                  | Boolean                               | <p>Возвращает true, если объект находится в "замороженном" режиме, иначе false.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p>                                                                                                                                                                          |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">ICircleGeometryAccess</a> | <p>Задаёт координаты центра круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                                                                                            |
| <a href="#">setRadius(radius)</a>           | <a href="#">ICircleGeometryAccess</a> | <p>Задаёт радиус круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                                                                                                       |

| Имя                        | Возвращает                 | Описание                                                                          |
|----------------------------|----------------------------|-----------------------------------------------------------------------------------|
| <a href="#">unfreeze()</a> | <a href="#">IFreezable</a> | Переводит объект в активный режим.<br>Унаследован от <a href="#">IFreezable</a> . |

#### Описание методов

##### getType

```
{String} getType()
```

Возвращает строку "Circle".

## IBaseGeometry

Расширяет [IEventEmitter](#).

Интерфейс базовой геометрии.

[Конструктор](#) | [Поля](#) | [Методы](#)

#### Конструктор

```
IBaseGeometry()
```

#### Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

#### Методы

| Имя                         | Возвращает      | Описание                                                                                                                                                                                                                                           |
|-----------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getBounds()</a> | Number[][] null | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. |
| <a href="#">getType()</a>   | String          | Возвращает идентификатор типа геометрии.                                                                                                                                                                                                           |

#### Описание методов

##### getBounds

```
{Number[][]|null} getBounds()
```

**Возвращает** координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.

### getType

```
{String} getType()
```

**Возвращает** идентификатор типа геометрии.

## IBaseLinearRingGeometry

Расширяет [IBaseGeometry](#), [ILinearRingGeometryAccess](#).

Интерфейс геометрии "Замкнутый контур".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IBaseLinearRingGeometry()
```

### Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

### События

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> — старые координаты;</li><li><code>newCoordinates</code> - новые координаты;</li><li><code>oldFillRule</code> - старый алгоритм заливки;</li><li><code>newFillRule</code> - новые алгоритм заливки.</li></ul> Унаследовано от <a href="#">ILinearRingGeometryAccess</a> . |

### Методы

| Имя                                | Возвращает                 | Описание                                                                                                               |
|------------------------------------|----------------------------|------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a> | Boolean                    | Проверяет, лежит ли переданная точка внутри контура.<br><br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> . |
| <a href="#">freeze()</a>           | <a href="#">IFreezable</a> | Переводит объект в "замороженный" режим.<br><br>Унаследован от <a href="#">IFreezable</a> .                            |

| Имя                                        | Возвращает                                | Описание                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">get(index)</a>                 | Number[]                                  | Возвращает координаты точки с заданным индексом.<br><br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                       |
| <a href="#">getBounds()</a>                | Number[][] null                           | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getChildGeometry(index)</a>    | <a href="#">IPointGeometryAccess</a>      | Создает и возвращает объект <a href="#">IPointGeometryAccess</a> для заданной вершины ломаной линии.<br><br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                   |
| <a href="#">getClosest(anchorPosition)</a> | Object                                    | Ищет на контуре точку, ближайшую к anchorPosition.<br><br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                     |
| <a href="#">getCoordinates()</a>           | Number[][]                                | Возвращает массив координат геометрии.<br><br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                                 |
| <a href="#">getFillRule()</a>              | String                                    | Возвращает идентификатор алгоритма заливки.<br><br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                            |
| <a href="#">getLength()</a>                | Integer                                   | Возвращает количество точек в геометрии.<br><br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                               |
| <a href="#">getType()</a>                  | String                                    | Возвращает строку "LinearRing".                                                                                                                                                                                                                                                                          |
| <a href="#">insert(index, coordinates)</a> | <a href="#">ILinearRingGeometryAccess</a> | Добавляет новую точку с заданным индексом.<br><br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                             |

| Имя                                         | Возвращает                                | Описание                                                                                                                                                                                                                                                                              |
|---------------------------------------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">isFrozen()</a>                  | Boolean                                   | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                           |
| <a href="#">remove(index)</a>               | Number[]                                  | Удаляет точку с заданным индексом.<br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                      |
| <a href="#">set(index, coordinates)</a>     | <a href="#">ILinearRingGeometryAccess</a> | Задаёт координаты точки с заданным индексом.<br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                            |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">ILinearRingGeometryAccess</a> | Задаёт массив координат геометрии.<br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                      |
| <a href="#">setFillRule(fillRule)</a>       | <a href="#">ILinearRingGeometryAccess</a> | Задаёт алгоритм заливки контура.<br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> .                                                                                                                                                                                        |
| <a href="#">splice(index, number)</a>       | Number[][]                                | Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра number.<br>Унаследован от <a href="#">ILinearRingGeometryAccess</a> . |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>                | Переводит объект в активный режим.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                                                     |

## Описание методов

### getType

```
{String} getType()
```

Возвращает строку "LinearRing".

## IBaseLineStringGeometry

Расширяет [IBaseGeometry](#), [ILineStringGeometryAccess](#).

Интерфейс геометрии "Ломаная линия".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

`IBaseLineStringGeometry()`

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## События

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                          |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> — старые координаты;</li><li><code>newCoordinates</code> - новые координаты;</li></ul> Унаследовано от <a href="#">ILineStringGeometryAccess</a> . |

## Методы

| Имя                         | Возвращает                                    | Описание                                                                                                                                                                                                                                                                                             |
|-----------------------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">freeze()</a>    | <a href="#">IFreezable</a>                    | Переводит объект в "замороженный" режим.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                                                              |
| <a href="#">get(index)</a>  | <code>Number[]</code>                         | Возвращает координаты точки с заданным индексом.<br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                                                                                                                                                       |
| <a href="#">getBounds()</a> | <code>Number[][]</code> или <code>null</code> | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br>Унаследован от <a href="#">IBaseGeometry</a> . |

| Имя                                         | Возвращает                                | Описание                                                                                                                                                               |
|---------------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getChildGeometry(index)</a>     | <a href="#">IPointGeometryAccess</a>      | Создает и возвращает объект <a href="#">IPointGeometryAccess</a> для заданной вершины ломаной линии.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> . |
| <a href="#">getClosest(anchorPosition)</a>  | Object                                    | Ищет на ломаной линии точку, ближайшую к <code>anchorPosition</code> .<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                               |
| <a href="#">getCoordinates()</a>            | Number[][]                                | Возвращает массив координат геометрии.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                               |
| <a href="#">getLength()</a>                 | Integer                                   | Возвращает количество точек в геометрии.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                             |
| <a href="#">getType()</a>                   | String                                    | Возвращает строку "LineString".                                                                                                                                        |
| <a href="#">insert(index, coordinates)</a>  | <a href="#">ILineStringGeometryAccess</a> | Добавляет новую точку с заданным индексом.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                           |
| <a href="#">isFrozen()</a>                  | Boolean                                   | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br><br>Унаследован от <a href="#">IFreezable</a> .                                        |
| <a href="#">remove(index)</a>               | Number[]                                  | Удаляет точку с заданным индексом.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                                   |
| <a href="#">set(index, coordinates)</a>     | <a href="#">ILineStringGeometryAccess</a> | Задаёт координаты точки с заданным индексом.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                         |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">ILineStringGeometryAccess</a> | Задаёт массив координат геометрии.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                                   |



| Имя                                   | Возвращает                 | Описание                                                                                                                                                                                                                                                                                  |
|---------------------------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">splice(index, number)</a> | Number[][]                 | Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра number.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> . |
| <a href="#">unfreeze()</a>            | <a href="#">IFreezable</a> | Переводит объект в активный режим.<br><br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                                                     |

### Описание методов

#### getType

```
{String} getType()
```

Возвращает строку "LineString".

## IBasePointGeometry

Расширяет [IBaseGeometry](#), [IPointGeometryAccess](#).

Интерфейс геометрии "Точка".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IBasePointGeometry()
```

### Поля

| Имя                    | Тип                           | Описание                                                                 |
|------------------------|-------------------------------|--------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br><br>Унаследовано от <a href="#">IEventEmitter</a> . |

**События**

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | <p>Изменение координат. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>oldCoordinates</code> — старые координаты;</li> <li><code>newCoordinates</code> - новые координаты;</li> </ul> <p>Унаследовано от <a href="#">IPointGeometryAccess</a>.</p> |

**Методы**

| Имя                                         | Возвращает                           | Описание                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getBounds()</a>                 | <code>Number[][] null</code>         | <p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.</p> <p>Унаследован от <a href="#">IBaseGeometry</a>.</p> |
| <a href="#">getCoordinates()</a>            | <code>Number[][] null</code>         | <p>Возвращает координаты точки.</p> <p>Унаследован от <a href="#">IPointGeometryAccess</a>.</p>                                                                                                                                                                                                                |
| <a href="#">getType()</a>                   | <code>String</code>                  | Возвращает строку "Point".                                                                                                                                                                                                                                                                                     |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IPointGeometryAccess</a> | <p>Задаёт координаты точки.</p> <p>Унаследован от <a href="#">IPointGeometryAccess</a>.</p>                                                                                                                                                                                                                    |

**Описание методов****getType**

```
{String} getType()
```

Возвращает строку "Point".

**IBasePolygonGeometry**

Расширяет [IBaseGeometry](#), [IPolygonGeometryAccess](#).

Интерфейс геометрии "Многоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

`IBasePolygonGeometry()`

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## События

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> — старые координаты;</li><li><code>newCoordinates</code> - новые координаты;</li><li><code>oldFillRule</code> - старый алгоритм заливки;</li><li><code>newFillRule</code> - новые алгоритм заливки.</li></ul> Унаследовано от <a href="#">IPolygonGeometryAccess</a> . |

## Методы

| Имя                                | Возвращает                 | Описание                                                                                                                   |
|------------------------------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a> | Boolean                    | Проверяет, лежит ли переданная точка внутри многоугольника.<br><br>Унаследован от <a href="#">IPolygonGeometryAccess</a> . |
| <a href="#">freeze()</a>           | <a href="#">IFreezable</a> | Переводит объект в "замороженный" режим.<br><br>Унаследован от <a href="#">IFreezable</a> .                                |
| <a href="#">get(index)</a>         | Number[][]                 | Возвращает координаты контура с заданным индексом.<br><br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .          |

| Имя                                        | Возвращает                                | Описание                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getBounds()</a>                | Number[][] null                           | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getChildGeometry(index)</a>    | <a href="#">ILinearRingGeometryAccess</a> | Создает и возвращает объект <a href="#">ILinearRingGeometryAccess</a> для заданного контура.<br><br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                              |
| <a href="#">getClosest(anchorPosition)</a> | Object                                    | Ищет на контуре многоугольника точку, ближайшую к anchorPosition.<br><br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                         |
| <a href="#">getCoordinates()</a>           | Number[][][]                              | Возвращает массив координат геометрии.<br><br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                                    |
| <a href="#">getFillRule()</a>              | String                                    | Возвращает идентификатор алгоритма заливки.<br><br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                               |
| <a href="#">getLength()</a>                | Integer                                   | Возвращает количество контуров в геометрии.<br><br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                               |
| <a href="#">getType()</a>                  | String                                    | Возвращает строку "Polygon".                                                                                                                                                                                                                                                                             |
| <a href="#">insert(index, path)</a>        | <a href="#">IPolygonGeometryAccess</a>    | Добавляет новый контур с заданным индексом.<br><br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                               |

| Имя                                         | Возвращает                                | Описание                                                                                                                                                                                                                                                                                    |
|---------------------------------------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">isFrozen()</a>                  | Boolean                                   | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                 |
| <a href="#">remove(index)</a>               | <a href="#">ILinearRingGeometryAccess</a> | Удаляет контур с заданным индексом.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                              |
| <a href="#">set(index, path)</a>            | <a href="#">IPolygonGeometryAccess</a>    | Задаёт координаты контура с заданным индексом.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                   |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IPolygonGeometryAccess</a>    | Задаёт массив координат геометрии.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                               |
| <a href="#">setFillRule(fillRule)</a>       | <a href="#">IPolygonGeometryAccess</a>    | Задаёт алгоритм заливки многоугольника.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                          |
| <a href="#">splice(index, number)</a>       | <a href="#">ILinearRingGeometryAccess</a> | Удаляет определенное число контуров начиная с заданного индекса. При этом на место удаляемых контуров могут быть добавлены новые. Координаты новых контуров можно передавать дополнительными аргументами после параметра number.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> . |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>                | Переводит объект в активный режим.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                                                           |

## Описание методов

### getType

```
{String} getType()
```

Возвращает строку "Polygon".

## IBaseRectangleGeometry

Расширяет [IBaseGeometry](#), [IRectangleGeometryAccess](#).

Интерфейс геометрии "Прямоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
IBaseRectangleGeometry()
```

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## События

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                                           |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат углов. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> - старые координаты углов;</li><li><code>newCoordinates</code> - новые координаты углов.</li></ul> Унаследовано от <a href="#">IRectangleGeometryAccess</a> . |

## Методы

| Имя                                | Возвращает                 | Описание                                                                                                                                                                                                                                                                                                 |
|------------------------------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a> | Boolean                    | Проверяет, лежит ли переданная точка внутри прямоугольника.<br><br>Унаследован от <a href="#">IRectangleGeometryAccess</a> .                                                                                                                                                                             |
| <a href="#">freeze()</a>           | <a href="#">IFreezable</a> | Переводит объект в "замороженный" режим.<br><br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                                                              |
| <a href="#">getBounds()</a>        | Number[][] null            | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |

| Имя                                         | Возвращает                               | Описание                                                                                                                           |
|---------------------------------------------|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getClosest(anchorPosition)</a>  | Object                                   | Ищет на контуре прямоугольника точку, ближайшую к anchorPosition.<br><br>Унаследован от <a href="#">IRectangleGeometryAccess</a> . |
| <a href="#">getCoordinates()</a>            | Number[][]                               | Возвращает координаты двух противоположных углов прямоугольника.<br><br>Унаследован от <a href="#">IRectangleGeometryAccess</a> .  |
| <a href="#">getType()</a>                   | String                                   | Возвращает строку "Rectangle".                                                                                                     |
| <a href="#">isFrozen()</a>                  | Boolean                                  | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br><br>Унаследован от <a href="#">IFreezable</a> .    |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IRectangleGeometryAccess</a> | Задаёт координаты двух противоположных углов прямоугольника.<br><br>Унаследован от <a href="#">IRectangleGeometryAccess</a> .      |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>               | Переводит объект в активный режим.<br><br>Унаследован от <a href="#">IFreezable</a> .                                              |

## Описание методов

### getType

```
{String} getType()
```

Возвращает строку "Rectangle".

## IBehavior

Расширяет [IChildOnMap](#), [ICustomizable](#).

Поведение карты. Добавляет к карте реакцию на определённые пользовательские действия (например, таскание, масштабирование правой кнопкой мыши, многосенсорное касание). Поведение по умолчанию выключено и включается вызовом метода enable.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IBehavior([options])
```

Параметры:

| Параметр                | Значение по умолчанию | Описание                        |
|-------------------------|-----------------------|---------------------------------|
| <a href="#">options</a> | —                     | Тип: Object<br>Опции поведения. |

## Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

## События

| Имя                           | Описание                                                                                                                                                                                                                             |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">disable</a>       | Выключение поведения.                                                                                                                                                                                                                |
| <a href="#">enable</a>        | Включение поведения.                                                                                                                                                                                                                 |
| <a href="#">optionschange</a> | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                                       |
| <a href="#">parentchange</a>  | Сменился родительский объект.<br>Поля данных: <ul style="list-style-type: none"> <li><code>oldParent</code> - старый родитель;</li> <li><code>newParent</code> - новый родитель.</li> </ul> Унаследовано от <a href="#">IChild</a> . |

## Методы

| Имя                         | Возвращает                         | Описание                                                                                                                                        |
|-----------------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">disable()</a>   |                                    | Выключает поведение.                                                                                                                            |
| <a href="#">enable()</a>    |                                    | Включает поведение.                                                                                                                             |
| <a href="#">getParent()</a> | <a href="#">IParentOnMap</a>  null | Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.<br>Унаследован от <a href="#">IChildOnMap</a> . |
| <a href="#">isEnabled()</a> | Boolean                            | Проверяет, включено ли поведение.                                                                                                               |



| Имя                               | Возвращает                  | Описание                                                                                                                                                                                         |
|-----------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">setParent(parent)</a> | <a href="#">IChildOnMap</a> | Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта.<br><br>Унаследован от <a href="#">IChildOnMap</a> . |

#### Описание событий

##### disable

Выключение поведения.

##### enable

Включение поведения.

#### Описание методов

##### disable

```
{ } disable()
```

Выключает поведение.

##### enable

```
{ } enable()
```

Включает поведение.

##### isEnabled

```
{Boolean} isEnabled()
```

Проверяет, включено ли поведение.

**Возвращает** true - поведение включено, false - выключено.

## ICanvasTile

Расширяет [ITile](#).

Интерфейс тайлов, умеющих отображаться на 2d-контексте объекта canvas.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

#### Конструктор

```
ICanvasTile(url)
```

**Параметры:**

| Параметр           | Значение по умолчанию | Описание                  |
|--------------------|-----------------------|---------------------------|
| <code>url</code> * | —                     | Тип: String<br>URL тайла. |

\* Обязательный параметр/опция.

### Поля

| Имя                 | Тип                        | Описание                                                          |
|---------------------|----------------------------|-------------------------------------------------------------------|
| <code>events</code> | <code>IEventManager</code> | Менеджер событий.<br>Унаследовано от <code>IEventEmitter</code> . |

### События

| Имя                | Описание                                                          |
|--------------------|-------------------------------------------------------------------|
| <code>ready</code> | Событие готовности тайла.<br>Унаследовано от <code>ITile</code> . |

### Методы

| Имя                                                           | Возвращает | Описание                                                           |
|---------------------------------------------------------------|------------|--------------------------------------------------------------------|
| <code>destroy()</code>                                        |            | Уничтожает тайл.<br>Унаследован от <code>ITile</code> .            |
| <code>isReady()</code>                                        | Boolean    | Проверяет готовность тайла.<br>Унаследован от <code>ITile</code> . |
| <code>renderAt(context, canvasSize, bounds[, animate])</code> |            | Отрисовывает картиночный тайл на 2d-контексте объекта canvas.      |

### Описание методов

#### renderAt

```
{ } renderAt(context, canvasSize, bounds[, animate])
```

Отрисовывает картиночный тайл на 2d-контексте объекта canvas.

#### Параметры:

| Параметр                  | Значение по умолчанию | Описание                                       |
|---------------------------|-----------------------|------------------------------------------------|
| <code>context</code> *    | —                     | Тип: Object<br>2d-контекст объекта canvas.     |
| <code>canvasSize</code> * | —                     | Тип: Number[]<br>размеры HTML-элемента canvas. |

| Параметр                 | Значение по умолчанию | Описание                                                                                  |
|--------------------------|-----------------------|-------------------------------------------------------------------------------------------|
| <a href="#">bounds</a> * | —                     | Тип: Number[][]<br><br>Область в клиентских координатах, в которой нужно отрисовать тайл. |
| <a href="#">animate</a>  | false                 | Тип: Boolean<br><br>true - анимировать рендеринг, false - нет.                            |

\* Обязательный параметр/опция.

## IChild

Расширяет [IEventManager](#).

Интерфейс объекта, имеющего родителя.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IChild()
```

### Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventManager</a> . |

### События

| Имя                          | Описание                                                                                                                                                       |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">parentchange</a> | Сменился родительский объект.<br>Поля данных: <ul style="list-style-type: none"><li>oldParent - старый родитель;</li><li>newParent - новый родитель.</li></ul> |

### Методы

| Имя                               | Возвращает             | Описание                                  |
|-----------------------------------|------------------------|-------------------------------------------|
| <a href="#">getParent()</a>       | Object                 | Возвращает ссылку на родительский объект. |
| <a href="#">setParent(parent)</a> | <a href="#">IChild</a> | Устанавливает родительский объект.        |

## Описание событий

### parentchange

Сменился родительский объект.

Поля данных:

- `oldParent` - старый родитель;
- `newParent` - новый родитель.

## Описание методов

### getParent

```
{Object} getParent()
```

Возвращает ссылку на родительский объект.

### setParent

```
{IChild} setParent(parent)
```

Устанавливает родительский объект.

Возвращает ссылку на себя.

Параметры:

| Параметр              | Значение по умолчанию | Описание                            |
|-----------------------|-----------------------|-------------------------------------|
| <code>parent</code> * | —                     | Тип: Object<br>Родительский объект. |

\* Обязательный параметр/опция.

## IChildOnMap

Расширяет [IChild](#).

Интерфейс объекта, родитель которого относится к определенной карте.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IChildOnMap()
```

### Поля

| Имя                 | Тип                           | Описание                                                             |
|---------------------|-------------------------------|----------------------------------------------------------------------|
| <code>events</code> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

**События**

| Имя                          | Описание                                                                                                                                                                                                |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">parentchange</a> | Сменился родительский объект.<br>Поля данных: <ul style="list-style-type: none"><li>oldParent - старый родитель;</li><li>newParent - новый родитель.</li></ul> Унаследовано от <a href="#">IChild</a> . |

**Методы**

| Имя                               | Возвращает                         | Описание                                                                                                                                     |
|-----------------------------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getParent()</a>       | <a href="#">IParentOnMap</a>  null | Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.                                              |
| <a href="#">setParent(parent)</a> | <a href="#">IChildOnMap</a>        | Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. |

**Описание методов****getParent**

```
{IParentOnMap|null} getParent()
```

**Возвращает** ссылку на родительский объект или null, если родительский элемент не был установлен.

**setParent**

```
{IChildOnMap} setParent(parent)
```

Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                 | Значение по умолчанию | Описание                                                        |
|--------------------------|-----------------------|-----------------------------------------------------------------|
| <a href="#">parent</a> * | —                     | Тип: <a href="#">IParentOnMap</a>  null<br>Родительский объект. |

\* Обязательный параметр/опция.

**ICircleGeometry**

Расширяет [ICircleGeometryAccess](#), [IGeometry](#).

Интерфейс геометрии "Круг".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
ICircleGeometry()
```

## Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

## События

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a>              | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> - старые координаты центра;</li><li><code>newCoordinates</code> - новые координаты центра;</li><li><code>oldRadius</code> - старый радиус;</li><li><code>newRadius</code> - новый радиус.</li></ul> Унаследовано от <a href="#">ICircleGeometryAccess</a> . |
| <a href="#">mapchange</a>           | Сменилась карта. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldMap</code> - старая карта;</li><li><code>newMap</code> - новая карта.</li></ul> Унаследовано от <a href="#">IGeometry</a> .                                                                                                                                                        |
| <a href="#">optionschange</a>       | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                                                                                                                                                                                                                                                             |
| <a href="#">pixelgeometrychange</a> | Изменилась пиксельная геометрия. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>pixelGeometry</code> - новая пиксельная геометрия <a href="#">IPixelGeometry</a>.</li></ul> Унаследовано от <a href="#">IGeometry</a> .                                                                                                                               |

## Методы

| Имя                                         | Возвращает                            | Описание                                                                                                                                                                                                                                             |
|---------------------------------------------|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a>          | Boolean                               | <p>Проверяет, лежит ли переданная точка внутри круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                               |
| <a href="#">freeze()</a>                    | <a href="#">IFreezable</a>            | <p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p>                                                                                                                                                    |
| <a href="#">getBounds()</a>                 | Number[][] null                       | <p>Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.</p> <p>Унаследован от <a href="#">IGeometry</a>.</p> |
| <a href="#">getClosest(anchorPosition)</a>  | Object                                | <p>Ищет на окружности точку, ближайшую к anchorPosition.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                            |
| <a href="#">getCoordinates()</a>            | Number[] null                         | <p>Возвращает координаты центра круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                              |
| <a href="#">getMap()</a>                    | <a href="#">Map</a>  null             | <p>Возвращает текущую карту.</p> <p>Унаследован от <a href="#">IGeometry</a>.</p>                                                                                                                                                                    |
| <a href="#">getPixelGeometry([options])</a> | <a href="#">IPixelGeometry</a>        | <p>Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.</p> <p>Унаследован от <a href="#">IGeometry</a>.</p>                                                                                              |
| <a href="#">getRadius()</a>                 | Number                                | <p>Возвращает радиус круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                                         |
| <a href="#">getType()</a>                   | String                                | <p>Возвращает строку "Circle".</p>                                                                                                                                                                                                                   |
| <a href="#">isFrozen()</a>                  | Boolean                               | <p>Возвращает true, если объект находится в "замороженном" режиме, иначе false.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p>                                                                                                                |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">ICircleGeometryAccess</a> | <p>Задаёт координаты центра круга.</p> <p>Унаследован от <a href="#">ICircleGeometryAccess</a>.</p>                                                                                                                                                  |

| Имя                               | Возвращает                            | Описание                                                                          |
|-----------------------------------|---------------------------------------|-----------------------------------------------------------------------------------|
| <a href="#">setMap(map)</a>       |                                       | Задаёт карту.<br>Унаследован от <a href="#">IGeometry</a> .                       |
| <a href="#">setRadius(radius)</a> | <a href="#">ICircleGeometryAccess</a> | Задаёт радиус круга.<br>Унаследован от <a href="#">ICircleGeometryAccess</a> .    |
| <a href="#">unfreeze()</a>        | <a href="#">IFreezable</a>            | Переводит объект в активный режим.<br>Унаследован от <a href="#">IFreezable</a> . |

### Описание методов

#### getType

```
{String} getType()
```

Возвращает строку "Circle".

## ICircleGeometryAccess

Расширяет [IFreezable](#).

Интерфейс доступа к геометрии "Круг".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
ICircleGeometryAccess()
```

### Поля

| Имя                    | Тип                           | Описание                                                                  |
|------------------------|-------------------------------|---------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий объекта.<br>Унаследовано от <a href="#">IFreezable</a> . |

### События

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> - старые координаты центра;</li><li><code>newCoordinates</code> - новые координаты центра;</li><li><code>oldRadius</code> - старый радиус;</li><li><code>newRadius</code> - новый радиус.</li></ul> |



**Методы**

| Имя                                         | Возвращает                            | Описание                                                                                                                    |
|---------------------------------------------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a>          | Boolean                               | Проверяет, лежит ли переданная точка внутри круга.                                                                          |
| <a href="#">freeze()</a>                    | <a href="#">IFreezable</a>            | Переводит объект в "замороженный" режим.<br>Унаследован от <a href="#">IFreezable</a> .                                     |
| <a href="#">getClosest(anchorPosition)</a>  | Object                                | Ищет на окружности точку, ближайшую к anchorPosition.                                                                       |
| <a href="#">getCoordinates()</a>            | Number[] null                         | Возвращает координаты центра круга.                                                                                         |
| <a href="#">getRadius()</a>                 | Number                                | Возвращает радиус круга.                                                                                                    |
| <a href="#">isFrozen()</a>                  | Boolean                               | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br>Унаследован от <a href="#">IFreezable</a> . |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">ICircleGeometryAccess</a> | Задаёт координаты центра круга.                                                                                             |
| <a href="#">setRadius(radius)</a>           | <a href="#">ICircleGeometryAccess</a> | Задаёт радиус круга.                                                                                                        |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>            | Переводит объект в активный режим.<br>Унаследован от <a href="#">IFreezable</a> .                                           |

**Описание событий****change**

Изменение координат. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- oldCoordinates - старые координаты центра;
- newCoordinates - новые координаты центра;
- oldRadius - старый радиус;
- newRadius - новый радиус.

**Описание методов****contains**

```
{Boolean} contains(position)
```

Проверяет, лежит ли переданная точка внутри круга.

**Возвращает** признак принадлежности точки кругу.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                            |
|-------------------------|-----------------------|-----------------------------------------------------|
| <code>position *</code> | —                     | Тип: <code>Number[]</code><br><br>Координаты точки. |

\* Обязательный параметр/опция.

#### Пример:

```
var myCircle = new ymaps.geometry.base.Circle([0, 0], 10);
myCircle.contains([0, 10]); // => true
```

### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на окружности точку, ближайшую к `anchorPosition`.

**Возвращает** объект со следующими полями:

- `position` - точка на окружности, ближайшая к `anchorPosition`;
- `distance` - расстояние от `anchorPosition` до `position`;

#### Параметры:

| Параметр                      | Значение по умолчанию | Описание                                                                                                      |
|-------------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------|
| <code>anchorPosition *</code> | —                     | Тип: <code>Number[]</code><br><br>Координаты точки, для которой рассчитывается ближайшая точка на окружности. |

\* Обязательный параметр/опция.

#### Пример:

```
var myCircle = new ymaps.geometry.base.Circle([0, 0], 10);
myCircle.getClosest([0, 15]).position; // => [0, 10]
```

### getCoordinates

```
{Number[]|null} getCoordinates()
```

**Возвращает** координаты центра круга.

### getRadius

```
{Number} getRadius()
```

**Возвращает** радиус круга.

### setCoordinates

```
{ICircleGeometryAccess} setCoordinates(coordinates)
```

Задаёт координаты центра круга.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                                                    |
|----------------------------|-----------------------|-------------------------------------------------------------|
| <code>coordinates</code> * | —                     | Тип: <code>Number[] null</code><br>Координаты центра круга. |

\* Обязательный параметр/опция.

**setRadius**

```
{ICircleGeometryAccess} setRadius(radius)
```

Задаёт радиус круга.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                  |
|-----------------------|-----------------------|-------------------------------------------|
| <code>radius</code> * | —                     | Тип: <code>Number</code><br>Радиус круга. |

\* Обязательный параметр/опция.

**ICollection**

Расширяет [IEventEmitter](#).

Интерфейс коллекции.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
ICollection()
```

**Поля**

| Имя                 | Тип                           | Описание                                                             |
|---------------------|-------------------------------|----------------------------------------------------------------------|
| <code>events</code> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

**События**

| Имя                 | Описание                                |
|---------------------|-----------------------------------------|
| <code>add</code>    | Произошло добавление дочернего объекта. |
| <code>remove</code> | Произошло удаление дочернего объекта.   |

**Методы**

| Имя                      | Возвращает                  | Описание                               |
|--------------------------|-----------------------------|----------------------------------------|
| <code>add(object)</code> | <a href="#">ICollection</a> | Добавляет в коллекцию дочерний объект. |

| Имя                         | Возвращает               | Описание                              |
|-----------------------------|--------------------------|---------------------------------------|
| <code>getIterator()</code>  | <code>Iterator</code>    | Возвращает итератор по коллекции.     |
| <code>remove(object)</code> | <code>ICollection</code> | Удаляет из коллекции дочерний объект. |

**Описание событий****add**

Произошло добавление дочернего объекта.

**remove**

Произошло удаление дочернего объекта.

**Описание методов****add**

```
{ICollection} add(object)
```

Добавляет в коллекцию дочерний объект.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                           |
|-----------------------|-----------------------|------------------------------------|
| <code>object *</code> | —                     | Тип: Object<br>Добавляемый объект. |

\* Обязательный параметр/опция.

**getIterator**

```
{Iterator} getIterator()
```

**Возвращает** итератор по коллекции.

**remove**

```
{ICollection} remove(object)
```

Удаляет из коллекции дочерний объект.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                         |
|-----------------------|-----------------------|----------------------------------|
| <code>object *</code> | —                     | Тип: Object<br>Удаляемый объект. |

\* Обязательный параметр/опция.

## IContainerPane

Расширяет [IPane](#), [IPositioningContext](#).

Интерфейс пейна карты, предназначенного для размещения отображений объектов, расположенных на поверхности карты. Предоставляет возможность перевода глобальных пикселей карты в собственную локальную координатную систему, и т.о. позволяет объекту позиционироваться внутри себя. Также информирует об изменении состояния карты и т.о. позволяет объекту обрабатывать эти изменения.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IContainerPane()
```

### Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

### События

| Имя                                | Описание                                                                                                                                                                                                                                                                               |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">actionbegin</a>        | Начало движения пейна. Экземпляр <a href="#">IEvent</a> .                                                                                                                                                                                                                              |
| <a href="#">actionend</a>          | Окончание движения пейна. Экземпляр <a href="#">IEvent</a> .                                                                                                                                                                                                                           |
| <a href="#">clientpixelschange</a> | Изменение локальной координатной системы пейна. Данное событие означает, что объекты, рассчитывающие свою позицию внутри пейна из глобальных пиксельных координат карты, должны пересчитать ее и обновить свое положение внутри DOM-элемента пейна. Экземпляр <a href="#">IEvent</a> . |
| <a href="#">overflowchange</a>     | Изменение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Экземпляр <a href="#">IEvent</a> .<br>Унаследовано от <a href="#">IPane</a> .                                                                                        |
| <a href="#">viewportchange</a>     | Изменение области показа пейна. Экземпляр <a href="#">IEvent</a> .                                                                                                                                                                                                                     |
| <a href="#">zindexchange</a>       | Изменение значения zIndex пейна. Экземпляр <a href="#">IEvent</a> .<br>Унаследовано от <a href="#">IPane</a> .                                                                                                                                                                         |

### Методы

| Имя                                                | Возвращает  | Описание                                                                                                           |
|----------------------------------------------------|-------------|--------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>                          |             | Уничтожает пейн.<br>Унаследован от <a href="#">IPane</a> .                                                         |
| <a href="#">fromClientPixels(clientPixelPoint)</a> | Number[]    | Преобразует клиентские пиксельные координаты в глобальные.<br>Унаследован от <a href="#">IPositioningContext</a> . |
| <a href="#">getElement()</a>                       | HTMLElement | Возвращает ссылку на DOM-контейнер пейна.<br>Унаследован от <a href="#">IPane</a> .                                |

| Имя                                              | Возвращает          | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getMap()</a>                         | <a href="#">Map</a> | Возвращает карту, которой принадлежит пейн.<br>Унаследован от <a href="#">IPane</a> .                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">getOverflow()</a>                    | String              | Возвращает значение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений: <ul style="list-style-type: none"> <li>"visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым;</li> <li>"hidden" - область отображения содержимого пейна ограничена контейнером карты.</li> </ul> Унаследован от <a href="#">IPane</a> . |
| <a href="#">getViewport()</a>                    | Number[][]          | Возвращает область показа пейна в клиентских координатах.                                                                                                                                                                                                                                                                                                                                                                                                                |
| <a href="#">getZIndex()</a>                      | Number              | Возвращает zIndex пейна.<br>Унаследован от <a href="#">IPane</a> .                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">getZoom()</a>                        | Number              | Возвращает текущий уровень масштабирования, в котором работает контекст позиционирования.<br>Унаследован от <a href="#">IPositioningContext</a> .                                                                                                                                                                                                                                                                                                                        |
| <a href="#">toClientPixels(globalPixelPoint)</a> | Number[]            | Преобразует глобальные пиксельные координаты в клиентские.<br>Унаследован от <a href="#">IPositioningContext</a> .                                                                                                                                                                                                                                                                                                                                                       |

## Описание событий

### actionbegin

Начало движения пейна. Экземпляр [IEvent](#).

**actionend**

Окончание движения пейна. Экземпляр [IEvent](#).

**clientpixelschange**

Изменение локальной координатной системы пейна. Данное событие означает, что объекты, рассчитывающие свою позицию внутри пейна из глобальных пиксельных координат карты, должны пересчитать ее и обновить свое положение внутри DOM-элемента пейна. Экземпляр [IEvent](#).

**viewportchange**

Изменение области показа пейна. Экземпляр [IEvent](#).

**Описание методов****getViewport**

```
{Number[][]} getViewport()
```

Возвращает область показа пейна в клиентских координатах.

**IControl**

Расширяет [IChildOnMap](#).

Элемент управления.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
IControl([options])
```

Интерфейс элемента управления.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                      |
|-------------------------|-----------------------|-----------------------------------------------|
| <a href="#">options</a> | —                     | Тип: Object<br><br>Опции элемента управления. |

**Поля**

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.                                                      |

**События**

| Имя                          | Описание                                                                                                                                                                                                |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">parentchange</a> | Сменился родительский объект.<br>Поля данных: <ul style="list-style-type: none"><li>oldParent - старый родитель;</li><li>newParent - новый родитель.</li></ul> Унаследовано от <a href="#">IChild</a> . |

**Методы**

| Имя                               | Возвращает                           | Описание                                                                                                                                     |
|-----------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getParent()</a>       | <a href="#">IControlParent</a>  null | Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.                                              |
| <a href="#">setParent(parent)</a> | <a href="#">IChildOnMap</a>          | Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта. |

**Описание полей****options**

```
{IOptionManager} options
```

Менеджер опций.

**Описание методов****getParent**

```
{IControlParent|null} getParent()
```

**Возвращает** ссылку на родительский объект или null, если родительский элемент не был установлен.

**setParent**

```
{IChildOnMap} setParent(parent)
```

Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта.

**Возвращает** ссылку на себя.

**Параметры:**



| Параметр              | Значение по умолчанию | Описание                                                       |
|-----------------------|-----------------------|----------------------------------------------------------------|
| <code>parent</code> * | —                     | Тип: <code>IControlParent</code>  null<br>Родительский объект. |

\* Обязательный параметр/опция.

## IControlParent

Расширяет `IParentOnMap`.

Интерфейс родительского объекта для элемента управления.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IControlParent()
```

### Поля

| Имя                | Тип                       | Описание            |
|--------------------|---------------------------|---------------------|
| <code>state</code> | <code>IDataManager</code> | Менеджер состояний. |

### События

| Имя                    | Описание                                                                                                                                                                                                 |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>mapchange</code> | Сменилась карта. Поля данных: <ul style="list-style-type: none"><li><code>oldMap</code> - старая карта;</li><li><code>newMap</code> - новая карта.</li></ul> Унаследовано от <code>IParentOnMap</code> . |

### Методы

| Имя                                 | Возвращает               | Описание                                                                                                           |
|-------------------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------|
| <code>getChildElement(child)</code> | <code>vow.Promise</code> | Возвращает объект-обещание, который подтверждается HTML-элементом, в котором должен разместиться дочерний элемент. |
| <code>getMap()</code>               | <code>Map</code>         | Возвращает ссылку на карту.<br>Унаследован от <code>IParentOnMap</code> .                                          |

## Описание полей

### state

```
{IDataManager} state
```

Менеджер состояний.

## Описание методов

### getChildElement

```
{vow.Promise} getChildElement(child)
```

**Возвращает** объект-обещание, который подтверждается HTML-элементом, в который должен разместиться дочерний элемент.

#### Параметры:

| Параметр             | Значение по умолчанию | Описание                                                 |
|----------------------|-----------------------|----------------------------------------------------------|
| <code>child *</code> | —                     | Тип: <a href="#">IControl</a><br><br>Дочерний геообъект. |

\* Обязательный параметр/опция.

## ICoordSystem

Интерфейс координатной системы карты. Этот интерфейс необходимо реализовать в случае использования нестандартных (например, цилиндрических) координат. Для решения задач поиска траектории движения по земной поверхности, используйте [coordSystem.geo](#), по декартовой плоскости — [coordSystem.cartesian](#).

**См.:** [coordSystem.geo](#) [coordSystem.cartesian](#)

[Конструктор](#) | [Методы](#)

### Конструктор

```
ICoordSystem()
```

### Методы

| Имя                                         | Возвращает | Описание                                                                                               |
|---------------------------------------------|------------|--------------------------------------------------------------------------------------------------------|
| <a href="#">getDistance(point1, point2)</a> | Number     | Возвращает кратчайшее (вдоль геодезической линии) расстояние между двумя заданными точками (в метрах). |

| Имя                                                                        | Возвращает | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------------------------------------------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>solveDirectProblem(startPoint, direction, distance)</code>           | Object     | <p>Решает так называемую первую (прямую) <b>геодезическую задачу</b>: где мы окажемся, если выйдем из указанной точки в указанном направлении и пройдем, не сворачивая, указанное расстояние. Решением прямой геодезической задачи являются следующие данные:</p> <ul style="list-style-type: none"> <li>• конечная точка,</li> <li>• конечное направление,</li> <li>• функция пути,</li> <li>• функция, позволяющая для произвольного момента времени указать, в какой точке мы будем находиться и в каком направлении двигаться.</li> </ul> |
| <code>solveInverseProblem(startPoint, endPoint[, reverseDirection])</code> | Object     | <p>Решает так называемую вторую (обратную) <b>геодезическую задачу</b>: построить кратчайший маршрут между двумя точками на картографируемой поверхности и определить расстояние и направление движения. Обратите внимание: на карте земной поверхности кратчайшие маршруты изображаются кривыми линиями. Для геообъектов в API можно включить режим отображения кратчайших расстояний между точками с помощью опции <code>geodesic</code>.</p>                                                                                               |

## Описание методов

### getDistance

```
{Number} getDistance(point1, point2)
```

**Возвращает** кратчайшее (вдоль геодезической линии) расстояние между двумя заданными точками (в метрах).

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                       |
|-----------------------|-----------------------|--------------------------------|
| <code>point1 *</code> | —                     | Тип: Number[]<br>Первая точка. |
| <code>point2 *</code> | —                     | Тип: Number[]<br>Вторая точка. |

\* Обязательный параметр/опция.

#### Пример:

```
// Расчет расстояния между Москвой и Нью-Йорком.
// Координаты Москвы
ymaps.geocode('Москва').then(function (res) {
 var moscowCoords = res.geoObjects.get(0).geometry.getCoordinates();
 // Координаты Нью-Йорка
 ymaps.geocode('Нью-Йорк').then(function (res) {
 var newYorkCoords = res.geoObjects.get(0).geometry.getCoordinates();
 // Расстояние
 alert(ymaps.formatter.distance(
 ymaps.coordSystem.geo.getDistance(moscowCoords, newYorkCoords)
));
 });
});
```

#### `solveDirectProblem`

```
{Object} solveDirectProblem(startPoint, direction, distance)
```

Решает так называемую первую (прямую) **геодезическую задачу**: где мы окажемся, если выйдем из указанной точки в указанном направлении и пройдем, не сворачивая, указанное расстояние. Решением прямой геодезической задачи являются следующие данные:

- конечная точка,
- конечное направление,
- функция пути,
- функция, позволяющая для произвольного момента времени указать, в какой точке мы будем находиться и в каком направлении двигаться.

**Возвращает** объект, содержащий следующие поля:

- `startPoint` - начальная точка в геокоординатах,
- `startDirection` - начальное направление движения,
- `endPoint` - конечная точка в геокоординатах,
- `endDirection` - конечное направление движения,
- `distance` - расстояние в метрах,
- `pathFunction` — функция, принимающая число от 0 до 1 (доля пройденного пути) и возвращающая структуру с полями `point` и `direction`.

#### Параметры:

| Параметр                  | Значение по умолчанию | Описание                            |
|---------------------------|-----------------------|-------------------------------------|
| <code>startPoint *</code> | —                     | Тип: Number[]<br>Точка отправления. |

| Параметр                 | Значение по умолчанию | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>direction *</code> | —                     | <p>Тип: <code>Number[]</code></p> <p>Направление. Задаётся в виде вектора (приращение координат) <code>[dlat, dlon]</code> либо <code>[dlon, dlat]</code>, в зависимости от параметра <code>coordorder</code>. Для того чтобы получить из заданного таким образом направления азимут (угол между направлением движения и направлением на север), необходимо вычислить арктангенс величины <code>dlon/dlat</code> (в JavaScript — стандартная функция <code>Math.atan2(dlon, dlat)</code>). Для того чтобы по известному азимуту "a" вычислить направление движения, нужно положить <code>dlat = cos(a)</code>, <code>dlon = sin(a)</code>.</p> |
| <code>distance *</code>  | —                     | <p>Тип: <code>Number</code></p> <p>Пройденное расстояние в метрах.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

\* Обязательный параметр/опция.

#### Пример:

```
// Допустим, мы вылетели из аэропорта Домодедово на северо-восток и
// пролетели 200 километров по прямой. // Изобразим наш путь с помощью меток на карте.

// Найдём с помощью геокодирования координаты начальной точки.
ymaps.geocode('Domodedovo airport').then(function (res) {
 var startPoint = res.geoObjects.get(0).geometry.getCoordinates();
 // Движемся на северо-восток, азимут 45 градусов
 // или pi/4 радиан.
 var azimuth = Math.PI / 4;
 // Направление движения.
 var direction = [Math.cos(azimuth), Math.sin(azimuth)],
 // Путевая функция.
 var path = ymaps.coordSystem.geo
 .solveDirectProblem(startPoint, direction, 2e5).pathFunction;

 // Изобразим путь на карте с помощью меток,
 // поставленных через каждые 10 км.
 for (var i = 0; i <= 20; i++) {
 map.geoObjects.add(new ymaps.Placemark(path(i/20).point));
 }
});
```

#### `solveInverseProblem`

```
{Object} solveInverseProblem(startPoint, endPoint[, reverseDirection])
```

Решает так называемую вторую (обратную) **геодезическую задачу**: построить кратчайший маршрут между двумя точками на картографируемой поверхности и определить расстояние и направление движения. Обратите внимание: на карте земной поверхности кратчайшие маршруты изображаются кривыми линиями. Для геообъектов в API можно включить режим отображения кратчайших расстояний между точками с помощью опции `geodesic`.

**Возвращает** объект, содержащий следующие поля:

- `startPoint` - начальная точка в геокоординатах,

- `startDirection` - начальное направление движения,
- `endPoint` - конечная точка в геокоординатах,
- `endDirection` - конечное направление движения,
- `distance` - расстояние в метрах,
- `pathFunction` - функция пути, принимающая число от 0 до 1 (доля пройденного пути) и возвращающая структуру с полями `point` и `direction`.

**Параметры:**

| Параметр                      | Значение по умолчанию | Описание                                                                                                          |
|-------------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>startPoint</code> *     | —                     | Тип: Number[]<br><br>Точка отправления.                                                                           |
| <code>endPoint</code> *       | —                     | Тип: Number[]<br><br>Точка прибытия.                                                                              |
| <code>reverseDirection</code> | false                 | Тип: Boolean<br><br>Направление движения.<br>false - выбрать кратчайшую дугу, true - выбрать обратную кратчайшей. |

\* Обязательный параметр/опция.

**Пример:**

```
// Строительство кратчайшего маршрута из Калининграда во Владивосток.
// Найдём координаты Калининграда
ymaps.geocode('Калининград').then(function (res) {
 var startPoint = res.geoObjects.get(0).geometry.getCoordinates();
 // Найдём координаты Владивостока.
 ymaps.geocode('Владивосток').then(function (res) {
 var endPoint = res.geoObjects.get(0).geometry.getCoordinates(),
 // Найдём функцию пути между двумя точками.
 path = ymaps.coordSystem.geo
 .solveInverseProblem(startPoint, endPoint).pathFunction;
 // Изобразим путь 20-ю точками.
 for (var i = 0; i <= 20; i++) {
 // Находим промежуточную точку.
 var position = path(i/20).point;
 // Добавим в промежуточную точку метку.
 map.geoObjects.add(new ymaps.Placemark(position, {
 // Покажем в контенте метки пройденное расстояние.
 iconContent: ymaps.formatter.distance(
 ymaps.coordSystem.geo.getDistance(startPoint, position)
), {
 preset: 'islands#redStretchyIcon'
 }
));
 }
 });
});
```

**ICopyrightsAccessor**

Расширяет [ICopyrightsProvider](#).

Интерфейс объекта, предоставляющего доступ к пользовательской информации об авторских правах, добавленной на карту с помощью метода `map.Copyrights.add`.

**См.:** `map.Copyrights.add`

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
ICopyrightsAccessor()
```

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## События

| Имя                              | Описание                                                                                                   |
|----------------------------------|------------------------------------------------------------------------------------------------------------|
| <a href="#">copyrightschange</a> | Событие изменения информации об авторских правах.<br>Унаследовано от <a href="#">ICopyrightsProvider</a> . |

## Методы

| Имя                                         | Возвращает                  | Описание                                                                                                                                                            |
|---------------------------------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getCopyrights(coords, zoom)</a> | <a href="#">vow.Promise</a> | Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования.<br><br>Унаследован от <a href="#">ICopyrightsProvider</a> . |
| <a href="#">remove()</a>                    |                             | Удаляет добавленную через данный объект информацию об авторских правах.<br><br>Унаследован от <a href="#">ICopyrightsProvider</a> .                                 |
| <a href="#">setCopyrights(copyrights)</a>   |                             | Задаёт новое значение для добавленной через данный объект информации об авторских правах.<br><br>Унаследован от <a href="#">ICopyrightsProvider</a> .               |

## ICopyrightsProvider

Расширяет [IEventEmitter](#).

Провайдер информации об авторских правах.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
ICopyrightsProvider()
```

**Поля**

| Имя                 | Тип                        | Описание                                                          |
|---------------------|----------------------------|-------------------------------------------------------------------|
| <code>events</code> | <code>IEventManager</code> | Менеджер событий.<br>Унаследовано от <code>IEventEmitter</code> . |

**События**

| Имя                           | Описание                                          |
|-------------------------------|---------------------------------------------------|
| <code>copyrightschange</code> | Событие изменения информации об авторских правах. |

**Методы**

| Имя                                      | Возвращает               | Описание                                                                                                |
|------------------------------------------|--------------------------|---------------------------------------------------------------------------------------------------------|
| <code>getCopyrights(coords, zoom)</code> | <code>vow.Promise</code> | Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. |
| <code>remove()</code>                    |                          | Удаляет добавленную через данный объект информацию об авторских правах.                                 |
| <code>setCopyrights(copyrights)</code>   |                          | Задаёт новое значение для добавленной через данный объект информации об авторских правах.               |

**Описание событий****`copyrightschange`**

Событие изменения информации об авторских правах.

**Описание методов****`getCopyrights`**

```
{vow.Promise} getCopyrights(coords, zoom)
```

Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования.

**Возвращает** Promise, который будет подтверждён массивом строк или DOM-элементов с информацией об авторских правах.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                                                                    |
|-----------------------|-----------------------|-------------------------------------------------------------------------------------------------------------|
| <code>coords</code> * | —                     | Тип: <code>Number[]</code><br><br>Точка на карте, для которой запрашивается информация об авторских правах. |



| Параметр               | Значение по умолчанию | Описание                                                                                               |
|------------------------|-----------------------|--------------------------------------------------------------------------------------------------------|
| <a href="#">zoom</a> * | —                     | Тип: Number<br><br>Уровень масштабирования, для которого запрашивается информация об авторских правах. |

\* Обязательный параметр/опция.

### remove

```
{ } remove()
```

Удаляет добавленную через данный объект информацию об авторских правах.

### setCopyrights

```
{ } setCopyrights(copyrights)
```

Задаёт новое значение для добавленной через данный объект информации об авторских правах.

#### Параметры:

| Параметр                     | Значение по умолчанию | Описание                                                                              |
|------------------------------|-----------------------|---------------------------------------------------------------------------------------|
| <a href="#">copyrights</a> * | —                     | Тип: String HTMLElement String[] HTMLElement[]<br><br>Информация об авторских правах. |

\* Обязательный параметр/опция.

## ICustomizable

Расширяет [IEventEmitter](#).

Интерфейс объекта, который можно настраивать с помощью глобального реестра опций.

[Конструктор](#) | [Поля](#) | [События](#)

### Конструктор

```
ICustomizable()
```

### Поля

| Имя                     | Тип                            | Описание                                                                 |
|-------------------------|--------------------------------|--------------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br><br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.                                                          |

**События**

| Имя                           | Описание                    |
|-------------------------------|-----------------------------|
| <a href="#">optionschange</a> | Изменение в опциях объекта. |

**Описание полей****options**

```
{IOptionManager} options
```

Менеджер опций.

**Описание событий****optionschange**

Изменение в опциях объекта.

**IDataManager**

Расширяет [IEventEmitter](#).

Интерфейс менеджера произвольных данных.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
IDataManager()
```

**Поля**

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

**События**

| Имя                    | Описание          |
|------------------------|-------------------|
| <a href="#">change</a> | Изменение данных. |

**Методы**

| Имя                                     | Возвращает | Описание                         |
|-----------------------------------------|------------|----------------------------------|
| <a href="#">get(path, defaultValue)</a> | Object     | Возвращает значение поля данных. |

**Описание событий****change**

Изменение данных.

## Описание методов

### get

```
{Object} get(path, defaultValue)
```

Возвращает значение поля данных.

#### Параметры:

| Параметр                    | Значение по умолчанию | Описание                                                                                                              |
|-----------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>path</code> *         | —                     | Тип: String<br><br>Строка с именем свойства;<br>может содержать '!'.<br><br>Тип: Object<br><br>Значение по умолчанию. |
| <code>defaultValue</code> * | —                     |                                                                                                                       |

\* Обязательный параметр/опция.

#### Пример:

```
dataManager.get("icon.shadow.size", [10, 10]);
```

## IDomEvent

Расширяет [IEvent](#).

Объект DOM-события в системе API Яндекс.Карт. Оборачивает оригинальное DOM-событие браузера с целью нормализации имен полей данных. Т.е. посредством методов `get` и `callMethod` можно получать доступ к полям и методам оригинального DOM-события. При этом автоматически выполняется приведение к стандарту для тех полей и методов, реализация которых различна в разных браузерах. Так, `event.callMethod('stopPropagation')` остановит распространение DOM-события во всех браузерах, включая Internet Explorer.

[Конструктор](#) | [Методы](#)

#### Конструктор

```
IDomEvent(originalEvent)
```

#### Параметры:

| Параметр                     | Значение по умолчанию | Описание                                 |
|------------------------------|-----------------------|------------------------------------------|
| <code>originalEvent</code> * | —                     | Тип: Object<br><br>Исходное DOM-событие. |

\* Обязательный параметр/опция.

## Методы

| Имя                                             | Возвращает | Описание                                                                                                                                                                            |
|-------------------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">allowMapEvent()</a>                 |            | Разрешает распространение данного события на карту.<br>Унаследован от <a href="#">IEvent</a> .                                                                                      |
| <a href="#">callMethod(name)</a>                |            | Вызывает указанный метод у исходного события. Второй и последующие аргументы передаются методу при вызове.<br>Унаследован от <a href="#">IEvent</a> .                               |
| <a href="#">get(name)</a>                       | Object     | Возвращает заданное свойство события. С помощью этого метода можно получать доступ как к свойствам исходного события, так и к дополнительным свойствам, предоставляемым API карт.   |
| <a href="#">getSourceEvent()</a>                | IDomEvent  | Возвращает исходное DOM-событие.                                                                                                                                                    |
| <a href="#">isDefaultPrevented()</a>            | Boolean    | Возвращает true, если реакция по умолчанию на событие отменена.<br>Унаследован от <a href="#">IEvent</a> .                                                                          |
| <a href="#">isImmediatePropagationStopped()</a> | Boolean    | Возвращает true, если обработка события прервана.<br>Унаследован от <a href="#">IEvent</a> .                                                                                        |
| <a href="#">isMapEventAllowed()</a>             | Boolean    | Возвращает true, если событие карты разрешено.<br>Унаследован от <a href="#">IEvent</a> .                                                                                           |
| <a href="#">isPropagationStopped()</a>          | Boolean    | Возвращает true, если распространение события прервано,<br>Унаследован от <a href="#">IEvent</a> .                                                                                  |
| <a href="#">preventDefault()</a>                |            | Отменяет реакцию по умолчанию на событие в рамках системы событий API Яндекс.Карт. Вызов этого метода не влияет на обработку браузером действия по умолчанию исходного DOM-события. |

| Имя                                     | Возвращает | Описание                                                                                                                                                                                                                                 |
|-----------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>stopImmediatePropagation()</code> |            | Прекращает обработку события в системе событий API Яндекс.Карт. Т.е. после вызова данного метода ни один обработчик данного события не будет вызван. Вызов этого метода не влияет на обработку исходного DOM-события на уровне браузера. |
| <code>stopPropagation()</code>          |            | Прекращает распространение DOM-события в системе событий API Яндекс.Карт. Вызов этого метода не влияет на распространение исходного DOM-события по DOM-дереву.                                                                           |

### Описание методов

#### get

```
{Object} get(name)
```

Возвращает заданное свойство события. С помощью этого метода можно получать доступ как к свойствам исходного события, так и к дополнительным свойствам, предоставляемым API карт.

**Возвращает** значение свойства.

#### Параметры:

| Параметр                            | Значение по умолчанию | Описание                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code><a href="#">name</a> *</code> | —                     | Тип: String<br><br>Имя свойства.<br>Поддерживаются дополнительные свойства: <ul style="list-style-type: none"><li>'propagatedData' - данные события, которые сохраняются при его пропагировании (распространении по DOM-дереву);</li><li>'position' - необязательное поле, которое содержит координаты события относительно документа.</li></ul> |

\* Обязательный параметр/опция.

### getSourceEvent

```
{IDomEvent} getSourceEvent()
```

Возвращает исходное DOM-событие.

### preventDefault

```
{ } preventDefault()
```

Отменяет реакцию по умолчанию на событие в рамках системы событий API Яндекс.Карт. Вызов этого метода не влияет на обработку браузером действия по умолчанию исходного DOM-события.

### stopImmediatePropagation

```
{ } stopImmediatePropagation()
```

Прекращает обработку события в системе событий API Яндекс.Карт. Т.е. после вызова данного метода ни один обработчик данного события не будет вызван. Вызов этого метода не влияет на обработку исходного DOM-события на уровне браузера.

### stopPropagation

```
{ } stopPropagation()
```

Прекращает распространение DOM-события в системе событий API Яндекс.Карт. Вызов этого метода не влияет на распространение исходного DOM-события по DOM-дереву.

## IDomEventEmitter

Расширяет [IEventEmitter](#).

Интерфейс объекта, генерирующего "DOM-события". Данный интерфейс декларирует перечень событий, используемых для отражения интерактивных действий пользователя с различными объектами в среде API Яндекс Карт. Объекты, реализующие данный интерфейс, не обязаны фактически генерировать все события декларируемые интерфейсом.

[Конструктор](#) | [Поля](#) | [События](#)

### Конструктор

```
IDomEventEmitter()
```

### Поля

| Имя                    | Тип                           | Описание          |
|------------------------|-------------------------------|-------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий. |

### События

| Имя                   | Описание                                                                                                                                                                                                                                                               |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a> | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> . |

| Имя                            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contextmenu</a>    | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">dblclick</a>       | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <a href="#">mousedown</a>      | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">mouseenter</a>     | Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="#">mouseleave</a>     | Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">mousemove</a>      | Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">mouseup</a>        | Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchend</a>  | Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <a href="#">multitouchmove</a> | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> |

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchstart</a> | Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code> , которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля: <ul style="list-style-type: none"><li><code>clientX</code> - координата касания X относительно видимой области браузера;</li><li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li><li><code>pageX</code> - координата касания X касания относительно начала документа;</li><li><code>pageY</code> - координата касания Y касания относительно начала документа.</li></ul> |
| <a href="#">wheel</a>           | Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

### Описание полей

#### events

```
{EventManager} events
```

Менеджер событий.

### Описание событий

#### click

Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).

#### contextmenu

Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).

#### dblclick

Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).

#### mousedown

Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).

#### mouseenter

Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).



**mouseleave**

Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).

**mousemove**

Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).

**mouseup**

Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).

**multitouchend**

Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent`.

**multitouchmove**

Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent` с информацией о прикосновениях. Определяет свойство `touches`, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:

- `clientX` - координата касания X относительно видимой области браузера;
- `clientY` - координата касания Y относительно видимой области браузера;
- `pageX` - координата касания X касания относительно начала документа;
- `pageY` - координата касания Y касания относительно начала документа.

**multitouchstart**

Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса `IMultiTouchEvent` с информацией о прикосновениях. Определяет свойство `touches`, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:

- `clientX` - координата касания X относительно видимой области браузера;
- `clientY` - координата касания Y относительно видимой области браузера;
- `pageX` - координата касания X касания относительно начала документа;
- `pageY` - координата касания Y касания относительно начала документа.

**wheel**

Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса [MapEvent](#). Более детальное описание доступно в [domEvent.manager](#).

**IDomTile**

Расширяет [ITile](#).

Интерфейс тайлов, представляющих из себя DOM-объект.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
IDomTile(url)
```

## Параметры:

| Параметр              | Значение по умолчанию | Описание                  |
|-----------------------|-----------------------|---------------------------|
| <a href="#">url</a> * | —                     | Тип: String<br>URL тайла. |

\* Обязательный параметр/опция.

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## События

| Имя                   | Описание                                                             |
|-----------------------|----------------------------------------------------------------------|
| <a href="#">ready</a> | Событие готовности тайла.<br>Унаследовано от <a href="#">ITile</a> . |

## Методы

| Имя                                                      | Возвращает | Описание                                                              |
|----------------------------------------------------------|------------|-----------------------------------------------------------------------|
| <a href="#">destroy()</a>                                |            | Уничтожает тайл.<br>Унаследован от <a href="#">ITile</a> .            |
| <a href="#">isReady()</a>                                | Boolean    | Проверяет готовность тайла.<br>Унаследован от <a href="#">ITile</a> . |
| <a href="#">renderAt(context, clientBounds, animate)</a> |            | Добавляет тайл в родительский HTML-элемент.                           |

## Описание методов

### renderAt

```
{ } renderAt(context, clientBounds, animate)
```

Добавляет тайл в родительский HTML-элемент.

## Параметры:

| Параметр                  | Значение по умолчанию | Описание                                      |
|---------------------------|-----------------------|-----------------------------------------------|
| <a href="#">context</a> * | —                     | Тип: HTMLElement<br>Родительский HTML-элемент |

| Параметр                    | Значение по умолчанию | Описание                                                                             |
|-----------------------------|-----------------------|--------------------------------------------------------------------------------------|
| <code>clientBounds</code> * | —                     | Тип: Number[][]<br><br>Область в клиентских координатах, которую должен занять тайл. |
| <code>animate</code> *      | —                     | Тип: Boolean<br><br>true - анимировать рендеринг, false - нет.                       |

\* Обязательный параметр/опция.

## IEvent

Событие, бросаемое менеджером событий IEventManager.

[Конструктор](#) | [Методы](#)

### Конструктор

```
IEvent()
```

### Методы

| Имя                                          | Возвращает  | Описание                                                                                                                                                                          |
|----------------------------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>allowMapEvent()</code>                 |             | Разрешает распространение данного события на карту.                                                                                                                               |
| <code>callMethod(name)</code>                |             | Вызывает указанный метод у исходного события. Второй и последующие аргументы передаются методу при вызове.                                                                        |
| <code>get(name)</code>                       | Object      | Возвращает свойство события по ключу. С помощью этого метода можно получать доступ как к свойствам исходного события, так и к дополнительным свойствам, предоставляемым API карт. |
| <code>getSourceEvent()</code>                | IEvent null | Возвращает исходное событие.                                                                                                                                                      |
| <code>isDefaultPrevented()</code>            | Boolean     | Возвращает true, если реакция по умолчанию на событие отменена.                                                                                                                   |
| <code>isImmediatePropagationStopped()</code> | Boolean     | Возвращает true, если обработка события прервана.                                                                                                                                 |
| <code>isMapEventAllowed()</code>             | Boolean     | Возвращает true, если событие карты разрешено.                                                                                                                                    |

| Имя                                     | Возвращает | Описание                                                                                                                                             |
|-----------------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>isPropagationStopped()</code>     | Boolean    | Возвращает true, если распространение события прервано,                                                                                              |
| <code>preventDefault()</code>           |            | Отменяет реакцию по умолчанию на событие в рамках системы событий API Яндекс.Карт.                                                                   |
| <code>stopImmediatePropagation()</code> |            | Прекращает обработку события в системе событий API Яндекс.Карт. Т.е. после вызова данного метода ни один обработчик данного события не будет вызван. |
| <code>stopPropagation()</code>          |            | Прекращает распространение события в системе событий API Яндекс.Карт.                                                                                |

### Описание методов

#### allowMapEvent

```
{ } allowMapEvent()
```

Разрешает распространение данного события на карту.

#### callMethod

```
{ } callMethod(name)
```

Вызывает указанный метод у исходного события. Второй и последующие аргументы передаются методу при вызове.

#### Параметры:

| Параметр      | Значение по умолчанию | Описание                   |
|---------------|-----------------------|----------------------------|
| <i>name</i> * | —                     | Тип: String<br>Имя метода. |

\* Обязательный параметр/опция.

#### get

```
{Object} get(name)
```

Возвращает свойство события по ключу. С помощью этого метода можно получать доступ как к свойствам исходного события, так и к дополнительным свойствам, предоставляемым API карт.

**Возвращает** значение свойства.

#### Параметры:

| Параметр            | Значение по умолчанию | Описание                     |
|---------------------|-----------------------|------------------------------|
| <code>name *</code> | —                     | Тип: String<br>Имя свойства. |

\* Обязательный параметр/опция.

### **getSourceEvent**

```
{IEvent|null} getSourceEvent()
```

**Возвращает** исходное событие.

### **isDefaultPrevented**

```
{Boolean} isDefaultPrevented()
```

**Возвращает** true, если реакция по умолчанию на событие отменена.

### **isImmediatePropagationStopped**

```
{Boolean} isImmediatePropagationStopped()
```

**Возвращает** true, если обработка события прервана.

### **isMapEventAllowed**

```
{Boolean} isMapEventAllowed()
```

**Возвращает** true, если событие карты разрешено.

### **isPropagationStopped**

```
{Boolean} isPropagationStopped()
```

**Возвращает** true, если распространение события прервано.

### **preventDefault**

```
{ } preventDefault()
```

Отменяет реакцию по умолчанию на событие в рамках системы событий API Яндекс.Карт.

### **stopImmediatePropagation**

```
{ } stopImmediatePropagation()
```

Прекращает обработку события в системе событий API Яндекс.Карт. Т.е. после вызова данного метода ни один обработчик данного события не будет вызван.

### **stopPropagation**

```
{ } stopPropagation()
```

Прекращает распространение события в системе событий API Яндекс.Карт.

## IEventController

Интерфейс контроллера событий. Позволяет контролировать подписки и отписки от событий на определённом менеджере событий.

[Конструктор](#) | [Методы](#)

### Конструктор

```
IEventController()
```

### Методы

| Имя                                            | Описание                                                                                                                                                      |
|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">onStartListening(events, type)</a> | Вызывается при первой подписке на указанный тип событий через указанный менеджер событий. Этот метод является опциональным.                                   |
| <a href="#">onStopListening(events, type)</a>  | Вызывается, когда определённый тип событий перестаёт слушаться на указанном менеджере событий (удалена последняя подписка). Этот метод является опциональным. |

### Описание методов

#### onStartListening

```
{ } onStartListening(events, type)
```

Вызывается при первой подписке на указанный тип событий через указанный менеджер событий. Этот метод является опциональным.

#### Параметры:

| Параметр                 | Значение по умолчанию | Описание                                                |
|--------------------------|-----------------------|---------------------------------------------------------|
| <a href="#">events</a> * | —                     | Тип: <a href="#">IEventManager</a><br>Менеджер событий. |
| <a href="#">type</a> *   | —                     | Тип: String<br>Тип события.                             |

\* Обязательный параметр/опция.

#### onStopListening

```
{ } onStopListening(events, type)
```

Вызывается, когда определённый тип событий перестаёт слушаться на указанном менеджере событий (удалена последняя подписка). Этот метод является опциональным.

#### Параметры:

| Параметр              | Значение по умолчанию | Описание                                                |
|-----------------------|-----------------------|---------------------------------------------------------|
| <code>events</code> * | —                     | Тип: <a href="#">IEventManager</a><br>Менеджер событий. |
| <code>type</code> *   | —                     | Тип: String<br>Тип события.                             |

\* Обязательный параметр/опция.

## IEventManager

Интерфейс объекта, у которого можно слушать события.

[Конструктор](#) | [Поля](#)

### Конструктор

```
IEventManager()
```

### Поля

| Имя                 | Тип                           | Описание          |
|---------------------|-------------------------------|-------------------|
| <code>events</code> | <a href="#">IEventManager</a> | Менеджер событий. |

### Описание полей

#### events

```
{IEventManager} events
```

Менеджер событий.

## IEventGroup

Группа слушателей событий.

[Конструктор](#) | [Методы](#)

### Конструктор

```
IEventGroup()
```

### Методы

| Имя                                                         | Возвращает                  | Описание                                   |
|-------------------------------------------------------------|-----------------------------|--------------------------------------------|
| <code>add(types, callback[, context[, priority]])</code>    | <a href="#">IEventGroup</a> | Добавляет слушатель события.               |
| <code>remove(types, callback[, context[, priority]])</code> | <a href="#">IEventGroup</a> | Удаляет слушатель события из группы.       |
| <code>removeAll()</code>                                    | <a href="#">IEventGroup</a> | Удаляет всех слушателей событий из группы. |

**Описание методов****add**

```
{IEventGroup} add(types, callback[, context[, priority]])
```

Добавляет слушатель события.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                                                                           |
|-------------------------|-----------------------|----------------------------------------------------------------------------------------------------|
| <code>types</code> *    | —                     | Тип: String String[]<br><br>Тип события или массив типов.                                          |
| <code>callback</code> * | —                     | Тип: Function<br><br>Функция-обработчик. В качестве параметра в функцию передается объект события. |
| <code>context</code>    | —                     | Тип: Object<br><br>Контекст выполнения функции-обработчика.                                        |
| <code>priority</code>   | 0                     | Тип: Integer<br><br>Приоритет подписки.                                                            |

\* Обязательный параметр/опция.

**remove**

```
{IEventGroup} remove(types, callback[, context[, priority]])
```

Удаляет слушатель события из группы.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                                                                           |
|-------------------------|-----------------------|----------------------------------------------------------------------------------------------------|
| <code>types</code> *    | —                     | Тип: String String[]<br><br>Тип или массив типов событий.                                          |
| <code>callback</code> * | —                     | Тип: Function<br><br>Функция-обработчик. В качестве параметра в функцию передается объект события. |



| Параметр              | Значение по умолчанию | Описание                                                    |
|-----------------------|-----------------------|-------------------------------------------------------------|
| <code>context</code>  | —                     | Тип: Object<br><br>Контекст выполнения функции-обработчика. |
| <code>priority</code> | 0                     | Тип: Integer<br><br>Приоритет подписки.                     |

\* Обязательный параметр/опция.

### **removeAll**

```
{IEventGroup} removeAll()
```

Удаляет всех слушателей событий из группы.

**Возвращает** ссылку на себя.

## **IEventManager**

Расширяет [IEventTrigger](#).

Менеджер событий. При помощи менеджера событий можно подписываться на события и отписываться от них, а также инициировать сами события.

[Конструктор](#) | [Методы](#)

### **Конструктор**

```
IEventManager()
```

### **Методы**

| Имя                                                         | Возвращает                          | Описание                                                                           |
|-------------------------------------------------------------|-------------------------------------|------------------------------------------------------------------------------------|
| <code>add(types, callback[, context[, priority]])</code>    | <a href="#">IEventManager</a>       | Добавляет новую подписку.                                                          |
| <code>fire(type[, event])</code>                            | <a href="#">IEventManager</a>       | Иницирует событие.                                                                 |
| <code>getParent()</code>                                    | <a href="#">IEventManager</a>  null | Возвращает ссылку на родительский менеджер событий.                                |
| <code>group()</code>                                        | <a href="#">IEventGroup</a>         | Возвращает группу слушателей событий, ассоциированную с данным менеджером событий. |
| <code>once(types, callback[, context[, priority]])</code>   | <a href="#">IEventManager</a>       | Добавляет слушателя, который вызовет функцию-обработчик только 1 раз.              |
| <code>remove(types, callback[, context[, priority]])</code> | <a href="#">IEventManager</a>       | Удаляет существующую подписку.                                                     |

| Имя                            | Возвращает | Описание                                     |
|--------------------------------|------------|----------------------------------------------|
| <code>setParent(parent)</code> |            | Устанавливает родительский менеджер событий. |

## Описание методов

### add

```
{EventManager} add(types, callback[, context[, priority]])
```

Добавляет новую подписку.

**Возвращает** ссылку на себя.

#### Параметры:

| Параметр                | Значение по умолчанию | Описание                                                                                                                                                                                                          |
|-------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>types *</code>    | —                     | Тип: String String[]<br><br>Тип события или массив типов.                                                                                                                                                         |
| <code>callback *</code> | —                     | Тип: Function<br><br>Функция-обработчик события. В качестве параметра в функцию передается объект, описывающий событие. Может быть либо произвольным объектом, либо реализовывать интерфейс <code>IEvent</code> . |
| <code>context</code>    | —                     | Тип: Object<br><br>Контекст исполнения обработчика.                                                                                                                                                               |
| <code>priority</code>   | 0                     | Тип: Integer<br><br>Приоритет подписки.                                                                                                                                                                           |

\* Обязательный параметр/опция.

### fire

```
{EventManager} fire(type[, event])
```

Иницирует событие.

**Возвращает** ссылку на себя.

#### Параметры:

| Параметр            | Значение по умолчанию | Описание                                                                                                                                                                                                             |
|---------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>type *</code> | —                     | Тип: String<br><br>Тип события.                                                                                                                                                                                      |
| <code>event</code>  | —                     | Тип: Object  <a href="#">Event</a><br><br>Событие. Если передан хэш с данными, то для него будет вызван метод <code>createEventObject</code> , и дальнейшие действия будут производиться с новым созданным событием. |

\* Обязательный параметр/опция.

### `getParent`

```
{EventManager|null} getParent()
```

**Возвращает** ссылку на родительский менеджер событий.

### `group`

```
{EventGroup} group()
```

**Возвращает** группу слушателей событий, ассоциированную с данным менеджером событий.

### `once`

```
{EventManager} once(types, callback[, context[, priority]])
```

Добавляет слушателя, который вызовет функцию-обработчик только 1 раз.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                                                                                                                                                                    |
|-------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>types *</code>    | —                     | Тип: String String[]<br><br>Тип события или массив типов.                                                                                                                                   |
| <code>callback *</code> | —                     | Тип: Function<br><br>Функция-обработчик события. В качестве параметра в функцию передается объект, описывающий событие. Этот объект должен реализовывать интерфейс <a href="#">IEvent</a> . |

| Параметр              | Значение по умолчанию | Описание                                            |
|-----------------------|-----------------------|-----------------------------------------------------|
| <code>context</code>  | —                     | Тип: Object<br><br>Контекст исполнения обработчика. |
| <code>priority</code> | 0                     | Тип: Integer<br><br>Приоритет подписки.             |

\* Обязательный параметр/опция.

### remove

```
{EventManager} remove(types, callback[, context[, priority]])
```

Удаляет существующую подписку.

**Возвращает** ссылку на себя.

#### Параметры:

| Параметр                | Значение по умолчанию | Описание                                                                                                                                                                                    |
|-------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>types</code> *    | —                     | Тип: String String[]<br><br>Тип события или массив типов.                                                                                                                                   |
| <code>callback</code> * | —                     | Тип: Function<br><br>Функция-обработчик события. В качестве параметра в функцию передается объект, описывающий событие. Этот объект должен реализовывать интерфейс <a href="#">IEvent</a> . |
| <code>context</code>    | —                     | Тип: Object<br><br>Контекст исполнения обработчика.                                                                                                                                         |
| <code>priority</code>   | 0                     | Тип: Integer<br><br>Приоритет подписки.                                                                                                                                                     |

\* Обязательный параметр/опция.

### setParent

```
{ } setParent(parent)
```

Устанавливает родительский менеджер событий.

#### Параметры:

| Параметр                 | Значение по умолчанию | Описание                                                                       |
|--------------------------|-----------------------|--------------------------------------------------------------------------------|
| <a href="#">parent</a> * | —                     | Тип: <a href="#">IEventManager</a>  null<br><br>Родительский менеджер событий. |

\* Обязательный параметр/опция.

## IEventPane

Расширяет [IDomEventEmitter](#), [IPane](#).

Интерфейс пейна событий карты. Позволяет отслеживать на карте DOM-события, но, в отличие от [IContainerPane](#), не позволяет размещать отображения объектов карты внутри своего DOM-элемента.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IEventPane()
```

### Поля

| Имя                    | Тип                           | Описание                                                                 |
|------------------------|-------------------------------|--------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br><br>Унаследовано от <a href="#">IEventEmitter</a> . |

### События

| Имя                         | Описание                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>       | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">contextmenu</a> | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">dblclick</a>    | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <a href="#">mousedown</a>   | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mouseenter</a>  | Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> .                      |

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mouseleave</a>      | <p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="#">mousemove</a>       | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="#">mouseup</a>         | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <a href="#">multitouchend</a>   | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="#">multitouchmove</a>  | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li>• <code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li>• <code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li>• <code>pageX</code> - координата касания X касания относительно начала документа;</li> <li>• <code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a> | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li>• <code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li>• <code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li>• <code>pageX</code> - координата касания X касания относительно начала документа;</li> <li>• <code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">overflowchange</a>  | <p>Изменение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Экземпляр <a href="#">IEvent</a>.</p> <p>Унаследовано от <a href="#">IPane</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

| Имя                          | Описание                                                                                                                                                                                                                                                                                 |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">wheel</a>        | Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">zindexchange</a> | Изменение значения zIndex пейна. Экземпляр <a href="#">IEvent</a> .<br>Унаследовано от <a href="#">IPane</a> .                                                                                                                                                                           |

## Методы

| Имя                           | Возвращает          | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>     |                     | Уничтожает пейн.<br>Унаследован от <a href="#">IPane</a> .                                                                                                                                                                                                                                                                                                                                                                                                               |
| <a href="#">getElement()</a>  | HTMLElement         | Возвращает ссылку на DOM-контейнер пейна.<br>Унаследован от <a href="#">IPane</a> .                                                                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">getMap()</a>      | <a href="#">Map</a> | Возвращает карту, которой принадлежит пейн.<br>Унаследован от <a href="#">IPane</a> .                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">getOverflow()</a> | String              | Возвращает значение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений: <ul style="list-style-type: none"> <li>"visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым;</li> <li>"hidden" - область отображения содержимого пейна ограничена контейнером карты.</li> </ul> Унаследован от <a href="#">IPane</a> . |
| <a href="#">getZIndex()</a>   | Number              | Возвращает zIndex пейна.<br>Унаследован от <a href="#">IPane</a> .                                                                                                                                                                                                                                                                                                                                                                                                       |

## IEventTrigger

Объект иницирующий события.

[Конструктор](#) | [Методы](#)

## Конструктор

```
IEventTrigger()
```

## Методы

| Имя                                    | Возвращает                 | Описание           |
|----------------------------------------|----------------------------|--------------------|
| <code>fire(type[, eventObject])</code> | <code>IEventTrigger</code> | Иницирует событие. |

## Описание методов

### fire

```
{IEventTrigger} fire(type[, eventObject])
```

Иницирует событие.

**Возвращает** ссылку на себя.

### Параметры:

| Параметр                 | Значение по умолчанию | Описание                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>type *</code>      | —                     | Тип: String<br><br>Тип события.                                                                                                                                                                                                                                                                                                                                         |
| <code>eventObject</code> | —                     | Тип: Object  <a href="#">IEvent</a><br><br>Объект, описывающий событие. Может быть либо произвольным объектом, либо реализовывать интерфейс <a href="#">IEvent</a> . Во втором случае после вызова каждого обработчика будет проверяться значение метода <code>isImmediatePropagationStopped()</code> - если оно истинно, оповещение о событии немедленно прекращается. |

\* Обязательный параметр/опция.

## IEventWorkflowController

Расширяет [IEventController](#).

Интерфейс контроллера событий, позволяющего влиять на распространение событий по дереву.

[Конструктор](#) | [Методы](#)

## Конструктор

```
IEventWorkflowController()
```



**Методы**

| Имя                                                     | Описание                                                                                                                                                                                                           |
|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>onAfterEventFiring(events, type[, event])</code>  | Функция, вызываемая после окончания обработки события менеджером событий. Этот метод является опциональным.                                                                                                        |
| <code>onBeforeEventFiring(events, type[, event])</code> | Функция, вызываемая до начала обработки события менеджером событий. Этот метод является опциональным.                                                                                                              |
| <code>onStartListening(events, type)</code>             | Вызывается при первой подписке на указанный тип событий через указанный менеджер событий. Этот метод является опциональным.<br>Унаследован от <a href="#">IEventController</a> .                                   |
| <code>onStopListening(events, type)</code>              | Вызывается, когда определённый тип событий перестаёт слушаться на указанном менеджере событий (удалена последняя подписка). Этот метод является опциональным.<br>Унаследован от <a href="#">IEventController</a> . |

**Описание методов****onAfterEventFiring**

```
{ } onAfterEventFiring(events, type[, event])
```

Функция, вызываемая после окончания обработки события менеджером событий. Этот метод является опциональным.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                |
|-----------------------|-----------------------|---------------------------------------------------------|
| <code>events</code> * | —                     | Тип: <a href="#">IEventManager</a><br>Менеджер событий. |
| <code>type</code> *   | —                     | Тип: String<br>Тип события.                             |
| <code>event</code>    | —                     | Тип: <a href="#">IEvent</a><br>Событие.                 |

\* Обязательный параметр/опция.

**onBeforeEventFiring**

```
{ } onBeforeEventFiring(events, type[, event])
```

Функция, вызываемая до начала обработки события менеджером событий. Этот метод является опциональным.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                |
|-----------------------|-----------------------|---------------------------------------------------------|
| <code>events</code> * | —                     | Тип: <a href="#">IEventManager</a><br>Менеджер событий. |
| <code>type</code> *   | —                     | Тип: String<br>Тип события.                             |
| <code>event</code>    | —                     | Тип: <a href="#">IEvent</a><br>Событие.                 |

\* Обязательный параметр/опция.

## IExpandableControlLayout

Расширяет [ILayout](#).

Интерфейс макета, который может находится в свернутом и развернутом состоянии.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IExpandableControlLayout()
```

### Поля

| Имя                 | Тип                           | Описание                                                                |
|---------------------|-------------------------------|-------------------------------------------------------------------------|
| <code>events</code> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

### События

| Имя                          | Описание                                                                                                                                                                                                                                                                                                                     |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>click</code>           | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <code>collapse</code>        | Событие, инициирующее сворачивание объекта.                                                                                                                                                                                                                                                                                  |
| <code>contextmenu</code>     | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <code>dblclick</code>        | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <code>emptinesschange</code> | Изменение признака пустоты макета. Экземпляр класса <a href="#">Event</a> .<br>Унаследовано от <a href="#">ILayout</a> .                                                                                                                                                                                                     |

| Имя                            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">expand</a>         | Событие, инициирующее разворачивание объекта.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <a href="#">mousedown</a>      | <p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEventManager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">mouseenter</a>     | <p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEventManager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <a href="#">mouseleave</a>     | <p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEventManager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">mousemove</a>      | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEventManager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">mouseup</a>        | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEventManager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">multitouchend</a>  | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchmove</a> | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchstart</a>     | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">parentelementchange</a> | <p>Изменение родительского элемента. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <a href="#">shapechange</a>         | <p>Изменение фигуры области охватывающей макет. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">wheel</a>               | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Методы

| Имя                                | Возвращает                   | Описание                                                                                                                                                                                                                                                                            |
|------------------------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>          |                              | <p>Деструктор. Вызывается при завершении работы с макетом.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                                                                                                                       |
| <a href="#">getData()</a>          | Object                       | <p>Возвращает объект данных макета.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                                                                                                                                              |
| <a href="#">getParentElement()</a> | HTMLElement                  | <p>Возвращает родительский HTML-элемент.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                                                                                                                                                                                         |
| <a href="#">getShape()</a>         | <a href="#">IShape</a>  null | <p>Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.</p> <p>Унаследован от <a href="#">ILayout</a>.</p> |

| Имя                                      | Возвращает | Описание                                                                                                                                                                                           |
|------------------------------------------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">isEmpty()</a>                | Boolean    | Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.<br>Унаследован от <a href="#">ILayout</a> . |
| <a href="#">setData(data)</a>            |            | Задаёт данные макета.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                  |
| <a href="#">setParentElement(parent)</a> |            | Добавляет макет в DOM-дерево.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                          |

### Описание событий

#### collapse

Событие, инициирующее сворачивание объекта.

#### expand

Событие, инициирующее разворачивание объекта.

## IFreezable

Интерфейс объекта с отключаемым событием изменения состояния. Объект, реализующий [IFreezable](#), может функционировать в одном из следующих режимов:

- 1. Активный. В этом режиме каждое изменение внутреннего состояния объекта сопровождается генерацией события [IFreezable.change](#).
- 2. Замороженный. В этом режиме изменения состояния объекта не сопровождаются событием [IFreezable.change](#), но если изменения имели место, событие [IFreezable.change](#) будет сгенерировано один раз при переходе в активный режим.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IFreezable()
```

### Поля

| Имя                    | Тип                           | Описание                  |
|------------------------|-------------------------------|---------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий объекта. |

### События

| Имя                    | Описание                                 |
|------------------------|------------------------------------------|
| <a href="#">change</a> | Изменение внутреннего состояния объекта. |

**Методы**

| Имя                        | Возвращает                 | Описание                                                                     |
|----------------------------|----------------------------|------------------------------------------------------------------------------|
| <a href="#">freeze()</a>   | <a href="#">IFreezable</a> | Переводит объект в "замороженный" режим.                                     |
| <a href="#">isFrozen()</a> | Boolean                    | Возвращает true, если объект находится в "замороженном" режиме, иначе false. |
| <a href="#">unfreeze()</a> | <a href="#">IFreezable</a> | Переводит объект в активный режим.                                           |

**Описание полей****events**

```
{EventManager} events
```

Менеджер событий объекта.

**Описание событий****change**

Изменение внутреннего состояния объекта.

**Описание методов****freeze**

```
{IFreezable} freeze()
```

Переводит объект в "замороженный" режим.

**Возвращает** ссылку на себя.

**isFrozen**

```
{Boolean} isFrozen()
```

**Возвращает** true, если объект находится в "замороженном" режиме, иначе false.

**unfreeze**

```
{IFreezable} unfreeze()
```

Переводит объект в активный режим.

**Возвращает** ссылку на себя.

**IGeocodeProvider**

Интерфейс провайдера геокодера.

[Конструктор](#) | [Методы](#)

**Конструктор**

```
IGeocodeProvider()
```

## Методы

| Имя                                      | Возвращает               | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>geocode(request[, options])</code> | <code>vow.Promise</code> | Отправляет запрос на геокодирование. Добавить функцию-обработчик для обработки результата геокодирования можно через возвращаемый объект-promise. Объект, поступающий на вход функции-обработчика может содержать поля только следующего вида: <code>geoObjects</code> , <code>layers</code> , <code>mapState</code> , <code>metaData</code> .                                                                                                                                                                                                                 |
| <code>suggest(request[, options])</code> | <code>vow.Promise</code> | Отправляет запрос за поисковыми подсказками. Возвращает объект <code>Promise</code> , который будет либо отклонён с ошибкой, либо подтверждён массивом объектов вида <code>{ displayName: "Мытищи, Московская область", value: "Россия, Московская область, Мытищи " }</code> . Поле <code>displayName</code> отвечает за представление топонима в удобном для чтения пользователем виде, поле <code>value</code> — значение, которое необходимо подставить в поле ввода после выбора пользователем данной подсказки.<br><br>Этот метод является опциональным. |

## Описание методов

## geocode

```
{vow.Promise} geocode(request[, options])
```

Отправляет запрос на геокодирование. Добавить функцию-обработчик для обработки результата геокодирования можно через возвращаемый объект-promise. Объект, поступающий на вход функции-обработчика может содержать поля только следующего вида: `geoObjects`, `layers`, `mapState`, `metaData`.

**Возвращает** объект-promise.

## Параметры:

| Параметр               | Значение по умолчанию | Описание                                        |
|------------------------|-----------------------|-------------------------------------------------|
| <code>request</code> * | —                     | Тип: <code>String</code><br><br>Строка запроса. |

| Параметр                          | Значение по умолчанию | Описание                                                                                                                                                                          |
|-----------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>options</code>              | —                     | Тип: Object<br><br>Опции.                                                                                                                                                         |
| <code>options.boundedBy</code>    | —                     | Тип: Number[][]<br><br>Прямоугольная область на карте, где предположительно находится искомый объект. Должна быть задана как массив, например <code>[[30, 40], [50, 50]]</code> . |
| <code>options.results</code>      | —                     | Тип: Number<br><br>Максимальное количество возвращаемых результатов.                                                                                                              |
| <code>options.skip</code>         | —                     | Тип: Number<br><br>Число результатов, которое необходимо пропустить.                                                                                                              |
| <code>options.strictBounds</code> | —                     | Тип: Boolean<br><br>Искать только внутри области, заданной опцией <code>boundedBy</code> .                                                                                        |

\* Обязательный параметр/опция.

## suggest

```
{vow.Promise} suggest(request[, options])
```

Отправляет запрос за поисковыми подсказками. Возвращает объект Promise, который будет либо отклонён с ошибкой, либо подтверждён массивом объектов вида `{ displayName: "Мытищи, Московская область", value: "Россия, Московская область, Мытищи " }`. Поле `displayName` отвечает за представление топонима в удобном для чтения пользователем виде, поле `value` — значение, которое необходимо подставить в поле ввода после выбора пользователем данной подсказки.

Этот метод является опциональным.

**Возвращает** объект Promise.

**Параметры:**

| Параметр               | Значение по умолчанию | Описание                           |
|------------------------|-----------------------|------------------------------------|
| <code>request</code> * | —                     | Тип: String<br><br>Строка запроса. |
| <code>options</code>   | —                     | Тип: Object<br><br>Опции.          |



| Параметр                          | Значение по умолчанию | Описание                                                                                                                                                                                       |
|-----------------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>options.boundedBy</code>    | —                     | Тип: <code>Number[][]</code><br><br>Прямоугольная область на карте, где предположительно находится искомый объект. Должна быть задана как массив, например <code>[[30, 40], [50, 50]]</code> . |
| <code>options.results</code>      | —                     | Тип: <code>Number</code><br><br>Максимальное количество возвращаемых результатов.                                                                                                              |
| <code>options.strictBounds</code> | —                     | Тип: <code>Boolean</code><br><br>Искать только внутри области, заданной опцией <code>boundedBy</code> .                                                                                        |

\* Обязательный параметр/опция.

## IGeometry

Расширяет [IBaseGeometry](#), [ICustomizable](#).

Интерфейс геометрии.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IGeometry()
```

### Поля

| Имя                  | Тип                            | Описание                                                             |
|----------------------|--------------------------------|----------------------------------------------------------------------|
| <code>events</code>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <code>options</code> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

### События

| Имя                    | Описание                                                                                                                                                                                                                                                   |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>mapchange</code> | Сменилась карта. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>oldMap</code> - старая карта;</li> <li><code>newMap</code> - новая карта.</li> </ul> |

| Имя                                 | Описание                                                                                                                                                                                                                                                                           |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">optionschange</a>       | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                                                                                     |
| <a href="#">pixelgeometrychange</a> | Изменилась пиксельная геометрия. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>pixelGeometry</code> - новая пиксельная геометрия <a href="#">IPixelGeometry</a>.</li> </ul> |

## Методы

| Имя                                         | Возвращает                     | Описание                                                                                                                                                                                     |
|---------------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getBounds()</a>                 | <code>Number[][] null</code>   | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно. |
| <a href="#">getMap()</a>                    | <code>Map null</code>          | Возвращает текущую карту.                                                                                                                                                                    |
| <a href="#">getPixelGeometry([options])</a> | <a href="#">IPixelGeometry</a> | Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.                                                                                              |
| <a href="#">getType()</a>                   | <code>String</code>            | Возвращает идентификатор типа геометрии.<br><br>Унаследован от <a href="#">IBaseGeometry</a> .                                                                                               |
| <a href="#">setMap(map)</a>                 |                                | Задаёт карту.                                                                                                                                                                                |

## Описание событий

### mapchange

Сменилась карта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldMap` - старая карта;
- `newMap` - новая карта.

### pixelgeometrychange

Изменилась пиксельная геометрия. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `pixelGeometry` - новая пиксельная геометрия [IPixelGeometry](#).

## Описание методов

### getBounds

```
{Number[][]|null} getBounds()
```

**Возвращает** координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.

**Пример:**

```
// Задаем центр и масштаб карты таким образом, чтобы геометрия целиком отобразилась в видимой области.
map.setBounds(myGeometry.getBounds());
```

**getMap**

```
{Map|null} getMap()
```

Возвращает текущую карту.

**getPixelGeometry**

```
{IPixelGeometry} getPixelGeometry([options])
```

Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                                                                               |
|-------------------------|-----------------------|--------------------------------------------------------------------------------------------------------|
| <a href="#">options</a> | —                     | Тип: Object<br><br>Хэш опций, позволяющий перекрыть часть текущих опций геометрии для данного расчета. |

**Пример:**

```
// Получаем пиксельное представление геометрии с учетом геодезии, но без оптимизации удаления невидимых точек.
myGeometry.getPixelGeometry({
 geodesic: true,
 simplification: false
}).getCoordinates();
```

**setMap**

```
{ } setMap(map)
```

Задает карту.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                               |
|-----------------------|-----------------------|--------------------------------------------------------|
| <a href="#">map</a> * | —                     | Тип: <a href="#">Map</a>  null<br><br>Ссылка на карту. |

\* Обязательный параметр/опция.

**IGeometryEditor**

Расширяет [ICustomizable](#), [IEventEmitter](#).

Интерфейс редактора геометрии.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
IGeometryEditor()
```

**Поля**

| Имя                      | Тип                            | Описание                                                             |
|--------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>   | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">geometry</a> | <a href="#">IGeometry</a>      | Редактируемая геометрия.                                             |
| <a href="#">options</a>  | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |
| <a href="#">state</a>    | <a href="#">IDataManager</a>   | Состояние редактора геометрии.                                       |

**События**

| Имя                           | Описание                                                                          |
|-------------------------------|-----------------------------------------------------------------------------------|
| <a href="#">optionschange</a> | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .    |
| <a href="#">statechange</a>   | Изменение состояния редактора геометрии. Экземпляр класса <a href="#">Event</a> . |

**Методы**

| Имя                            | Описание                        |
|--------------------------------|---------------------------------|
| <a href="#">startEditing()</a> | Включает режим редактирования.  |
| <a href="#">stopEditing()</a>  | Отключает режим редактирования. |

**Описание полей****geometry**

```
{IGeometry} geometry
```

Редактируемая геометрия.

**state**

```
{IDataManager} state
```

Состояние редактора геометрии.

**Описание событий****statechange**

Изменение состояния редактора геометрии. Экземпляр класса [Event](#).

**Описание методов****startEditing**

```
{ } startEditing()
```

Включает режим редактирования.

## stopEditing

```
{ } stopEditing()
```

Отключает режим редактирования.

## IGeometryEditorChildModel

Расширяет [IGeometryEditorModel](#).

Интерфейс дочерней модели данных.

[Конструктор](#) | [Поля](#) | [Методы](#)

### Конструктор

```
IGeometryEditorChildModel(geometry, editor, pixels, parent)
```

### Параметры:

| Параметр                   | Значение по умолчанию | Описание                                                                                                                                                                                                                                                    |
|----------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">geometry</a> * | —                     | Тип: <a href="#">IBaseGeometry</a><br><br>Редактируемая дочерняя геометрия. Интерфейс <a href="#">IBaseGeometry</a> не подразумевает наличие события <code>pixelgeometrychange</code> , поэтому пиксельные данные получаются от родительской модели данных. |
| <a href="#">editor</a> *   | —                     | Тип: <a href="#">IGeometryEditor</a><br><br>Ссылка на редактор геометрии.                                                                                                                                                                                   |
| <a href="#">pixels</a> *   | —                     | Тип: <code>Number[]</code><br><br>Пиксельные данные модели.                                                                                                                                                                                                 |
| <a href="#">parent</a> *   | —                     | Тип: <a href="#">IGeometryEditorModel</a><br><br>Родительская модель данных.                                                                                                                                                                                |

\* Обязательный параметр/опция.

### Поля

| Имя                      | Тип                             | Описание                                                             |
|--------------------------|---------------------------------|----------------------------------------------------------------------|
| <a href="#">editor</a>   | <a href="#">IGeometryEditor</a> | Редактор геометрии.                                                  |
| <a href="#">events</a>   | <a href="#">IEventManager</a>   | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">geometry</a> | <a href="#">IBaseGeometry</a>   | Геометрия модели.                                                    |

**Методы**

| Имя                            | Возвращает                           | Описание                                                                                      |
|--------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------|
| <code>destroy()</code>         |                                      | Деструктор.<br>Унаследован от <a href="#">IGeometryEditorModel</a> .                          |
| <code>getParent()</code>       | <a href="#">IGeometryEditorModel</a> | Возвращает родительскую модель данных.                                                        |
| <code>getPixels()</code>       | Number[]                             | Возвращает пиксельные данные модели.<br>Унаследован от <a href="#">IGeometryEditorModel</a> . |
| <code>setPixels(pixels)</code> |                                      | Задаёт пиксельные данные модели.                                                              |

**Описание полей****editor**

```
{IGeometryEditor} editor
```

Редактор геометрии.

**geometry**

```
{IBaseGeometry} geometry
```

Геометрия модели.

**Описание методов****getParent**

```
{IGeometryEditorModel} getParent()
```

Возвращает родительскую модель данных.

**setPixels**

```
{ } setPixels(pixels)
```

Задаёт пиксельные данные модели.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                |
|-----------------------|-----------------------|-----------------------------------------|
| <code>pixels *</code> | —                     | Тип: Number[]<br><br>Пиксельные данные. |

\* Обязательный параметр/опция.

**IGeometryEditorModel**

Расширяет [IEventEmitter](#).

Интерфейс модели данных редактора геометрии.

[Конструктор](#) | [Поля](#) | [Методы](#)

### Конструктор

```
IGeometryEditorModel(geometry, editor)
```

### Параметры:

| Параметр                   | Значение по умолчанию | Описание                                                              |
|----------------------------|-----------------------|-----------------------------------------------------------------------|
| <a href="#">geometry</a> * | —                     | Тип: <a href="#">IBaseGeometry</a><br>Редактируемая геометрия.        |
| <a href="#">editor</a> *   | —                     | Тип: <a href="#">IGeometryEditor</a><br>Ссылка на редактор геометрии. |

\* Обязательный параметр/опция.

### Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

### Методы

| Имя                         | Возвращает | Описание                             |
|-----------------------------|------------|--------------------------------------|
| <a href="#">destroy()</a>   |            | Деструктор.                          |
| <a href="#">getPixels()</a> | Number[]   | Возвращает пиксельные данные модели. |

### Описание методов

#### destroy

```
{ } destroy()
```

Деструктор.

#### getPixels

```
{Number[]} getPixels()
```

Возвращает пиксельные данные модели.

## IGeometryEditorRootModel

Расширяет [IGeometryEditorModel](#).

Интерфейс корневой модели данных.

[Конструктор](#) | [Поля](#) | [Методы](#)

## Конструктор

```
IGeometryEditorRootModel(geometry, editor)
```

## Параметры:

| Параметр                   | Значение по умолчанию | Описание                                                              |
|----------------------------|-----------------------|-----------------------------------------------------------------------|
| <a href="#">geometry</a> * | —                     | Тип: <a href="#">IGeometry</a><br>Редактируемая геометрия.            |
| <a href="#">editor</a> *   | —                     | Тип: <a href="#">IGeometryEditor</a><br>Ссылка на редактор геометрии. |

\* Обязательный параметр/опция.

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## Методы

| Имя                         | Возвращает | Описание                                                                                      |
|-----------------------------|------------|-----------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>   |            | Деструктор.<br>Унаследован от <a href="#">IGeometryEditorModel</a> .                          |
| <a href="#">getPixels()</a> | Number[]   | Возвращает пиксельные данные модели.<br>Унаследован от <a href="#">IGeometryEditorModel</a> . |

## IGeometryJson

Интерфейс объекта, описывающего JSON-представление геометрии.

[Конструктор](#) | [Поля](#)

## Конструктор

```
IGeometryJson()
```

## Поля

| Имя                  | Тип    | Описание                      |
|----------------------|--------|-------------------------------|
| <a href="#">type</a> | String | Идентификатор типа геометрии. |



**Описание полей****type**`{String} type`

Идентификатор типа геометрии.

**IGeoObject**

Расширяет [IChildOnMap](#), [ICustomizable](#), [IDomEventEmitter](#), [IParentOnMap](#).

Интерфейс геообъекта.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**`IGeoObject()`**Поля**

| Имя                        | Тип                             | Описание                                                                |
|----------------------------|---------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a>     | <a href="#">IEventManager</a>   | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">geometry</a>   | <a href="#">IGeometry</a>  null | Геометрия геообъекта.                                                   |
| <a href="#">options</a>    | <a href="#">IOptionManager</a>  | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .      |
| <a href="#">properties</a> | <a href="#">IDataManager</a>    | Данные геообъекта.                                                      |
| <a href="#">state</a>      | <a href="#">IDataManager</a>    | Состояние геообъекта.                                                   |

**События**

| Имя                            | Описание                                                                                                                                                                                                                                                                                                                     |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>          | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">contextmenu</a>    | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">dblclick</a>       | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <a href="#">geometrychange</a> | Изменение геометрии геообъекта. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> :<br><ul style="list-style-type: none"> <li><code>originalEvent</code>: <a href="#">IEvent</a> - оригинальное событие геометрии.</li> </ul>                                            |

| Имя                            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mapchange</a>      | <p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> <li>oldMap - старая карта;</li> <li>newMap - новая карта.</li> </ul> <p>Унаследовано от <a href="#">IParentOnMap</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <a href="#">mousedown</a>      | <p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <a href="#">mouseenter</a>     | <p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">mouseleave</a>     | <p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <a href="#">mousemove</a>      | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <a href="#">mouseup</a>        | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <a href="#">multitouchend</a>  | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <a href="#">multitouchmove</a> | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li>clientX - координата касания X относительно видимой области браузера;</li> <li>clientY - координата касания Y относительно видимой области браузера;</li> <li>pageX - координата касания X касания относительно начала документа;</li> <li>pageY - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |

| Имя                              | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchstart</a>  | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">optionschange</a>    | <p>Изменение в опциях объекта.</p> <p>Унаследовано от <a href="#">ICustomizable</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <a href="#">overlaychange</a>    | <p>Изменение оверлея геообъекта. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>overlay</code>: <a href="#">IOverlay</a> null - ссылка на оверлей;</li> <li><code>oldOverlay</code>: <a href="#">IOverlay</a> null - предыдущий оверлей геообъекта.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <a href="#">parentchange</a>     | <p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <li><code>oldParent</code> - старый родитель;</li> <li><code>newParent</code> - новый родитель.</li> </ul> <p>Унаследовано от <a href="#">IChild</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <a href="#">propertieschange</a> | <p>Изменение данных геообъекта. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>originalEvent</code>: <a href="#">IEvent</a> - оригинальное событие менеджера данных.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">wheel</a>            | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

## Методы

| Имя                      | Возвращает          | Описание                                                                               |
|--------------------------|---------------------|----------------------------------------------------------------------------------------|
| <a href="#">getMap()</a> | <a href="#">Map</a> | <p>Возвращает ссылку на карту.</p> <p>Унаследован от <a href="#">IParentOnMap</a>.</p> |

| Имя                               | Возвращает                         | Описание                                                                                                                                                                                         |
|-----------------------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getOverlay()</a>      | <a href="#">vow.Promise</a>        | Возвращает объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.                                     |
| <a href="#">getOverlaySync()</a>  | <a href="#">IOverlay</a>  null     | Метод предоставляет синхронный доступ к оверлею.                                                                                                                                                 |
| <a href="#">getParent()</a>       | <a href="#">IParentOnMap</a>  null | Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.<br><br>Унаследован от <a href="#">IChildOnMap</a> .                                              |
| <a href="#">setParent(parent)</a> | <a href="#">IChildOnMap</a>        | Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта.<br><br>Унаследован от <a href="#">IChildOnMap</a> . |

**Описание полей****geometry**

```
{IGeometry|null} geometry
```

Геометрия геообъекта.

**properties**

```
{IDataManager} properties
```

Данные геообъекта.

**state**

```
{IDataManager} state
```

Состояние геообъекта.

**Описание событий****geometrychange**

Изменение геометрии геообъекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- originalEvent: [IEvent](#) - оригинальное событие геометрии.

### overlaychange

Изменение оверлея геообъекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- overlay: [IOverlay](#)|null - ссылка на оверлей;
- oldOverlay: [IOverlay](#)|null - предыдущий оверлей геообъекта.

### propertieschange

Изменение данных геообъекта. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- originalEvent: [IEvent](#) - оригинальное событие менеджера данных.

### Описание методов

#### getOverlay

```
{vow.Promise} getOverlay()
```

**Возвращает** объект-обещание, который подтверждается объектом оверлея в момент его фактического создания, либо отклоняется с передачей соответствующей ошибки.

#### getOverlaySync

```
{IOverlay|null} getOverlaySync()
```

Метод предоставляет синхронный доступ к оверлею.

**Возвращает** ссылку на оверлей, либо null, если оверлей отсутствует.

## IGeoObjectCollection

Расширяет [ICustomizable](#), [IEventEmitter](#), [IParentOnMap](#).

Интерфейс коллекции геообъектов.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IGeoObjectCollection()
```

### Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

## События

| Имя                               | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">add</a>               | <p>Добавлен (вставлен) дочерний геообъект. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>index</code>: <code>Integer</code> - индекс добавленного геообъекта;</li> <li><code>child</code>: <a href="#">IGeoObject</a> - ссылка на добавленный геообъект.</li> </ul>                                                                                                     |
| <a href="#">boundschange</a>      | <p>Изменение координат географической области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса <a href="#">Event</a>.</p>                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">mapchange</a>         | <p>Сменилась карта. Поля данных:</p> <ul style="list-style-type: none"> <li><code>oldMap</code> - старая карта;</li> <li><code>newMap</code> - новая карта.</li> </ul> <p>Унаследовано от <a href="#">IParentOnMap</a>.</p>                                                                                                                                                                                                                                                      |
| <a href="#">optionschange</a>     | <p>Изменение в опциях объекта.</p> <p>Унаследовано от <a href="#">ICustomizable</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                         |
| <a href="#">pixelboundschange</a> | <p>Изменение пиксельных координат области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса <a href="#">Event</a>.</p>                                                                                                                                                                                                                                                                                                                                          |
| <a href="#">remove</a>            | <p>Удален дочерний геообъект. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>index</code>: <code>Integer</code> — индекс удаленного геообъекта;</li> <li><code>child</code>: <a href="#">IGeoObject</a> — ссылка на удаленный геообъект.</li> </ul>                                                                                                                      |
| <a href="#">set</a>               | <p>В коллекцию записан новый дочерний геообъект. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>index</code>: <code>Integer</code> - индекс геообъекта;</li> <li><code>child</code>: <a href="#">IGeoObject</a> - ссылка на новый геообъект;</li> <li><code>prevChild</code>: <a href="#">IGeoObject</a> — ссылка на предыдущее значение для данного индекса.</li> </ul> |

## Методы

| Имя                                       | Возвращает                           | Описание                                                       |
|-------------------------------------------|--------------------------------------|----------------------------------------------------------------|
| <a href="#">add(child[, index])</a>       | <a href="#">IGeoObjectCollection</a> | Добавляет (вставляет) дочерний геообъект в коллекцию.          |
| <a href="#">each(callback[, context])</a> |                                      | Для каждого дочернего геообъекта, вызывает функцию-обработчик. |

| Имя                                   | Возвращает                           | Описание                                                                                                                                                                                             |
|---------------------------------------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">get(index)</a>            | <a href="#">IGeoObject</a>           | Возвращает дочерний геообъект с заданным индексом.                                                                                                                                                   |
| <a href="#">getBounds()</a>           | <code>Number[][]</code>  null        | Возвращает географические координаты области, охватывающей коллекции и ее дочерние геообъекты.                                                                                                       |
| <a href="#">getIterator()</a>         | <a href="#">IIterator</a>            | Возвращает итератор по коллекции.                                                                                                                                                                    |
| <a href="#">getLength()</a>           | <code>Integer</code>                 | Возвращает длину коллекции.                                                                                                                                                                          |
| <a href="#">getMap()</a>              | <a href="#">Map</a>                  | Возвращает ссылку на карту.<br><br>Унаследован от <a href="#">IParentOnMap</a> .                                                                                                                     |
| <a href="#">getPixelBounds()</a>      | <code>Number[][]</code>  null        | Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и ее дочерние геообъекты.                                                                                                |
| <a href="#">indexOf(object)</a>       | <code>Integer</code>                 | Возвращает индекс дочернего геообъекта. Если геообъект в коллекции не найден, возвращается -1.                                                                                                       |
| <a href="#">remove(child)</a>         | <a href="#">IGeoObjectCollection</a> | Удаляет дочерний геообъект из коллекции.                                                                                                                                                             |
| <a href="#">removeAll()</a>           | <a href="#">IGeoObjectCollection</a> | Очищает коллекцию.                                                                                                                                                                                   |
| <a href="#">set(index, child)</a>     | <a href="#">IGeoObjectCollection</a> | Записывает в коллекцию новый дочерний геообъект.                                                                                                                                                     |
| <a href="#">splice(index, number)</a> | <a href="#">IGeoObjectCollection</a> | Удаляет геообъекты из коллекции. Если необходимо, вставляет на их место другие. Объекты, которые будут вставлены вместо удаляемых, передаются как дополнительные параметры (после параметра number). |

## Описание событий

### add

Добавлен (вставлен) дочерний геообъект. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `index`: `Integer` - индекс добавленного геообъекта;
- `child`: [IGeoObject](#) - ссылка на добавленный геообъект.

**boundschange**

Изменение координат географической области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса [Event](#).

**pixelboundschange**

Изменение пиксельных координат области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса [Event](#).

**remove**

Удален дочерний геообъект. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `index`: `Integer` — индекс удаленного геообъекта;
- `child`: [IGeoObject](#) — ссылка на удаленный геообъект.

**set**

В коллекцию записан новый дочерний геообъект. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `index`: `Integer` - индекс геообъекта;
- `child`: [IGeoObject](#) - ссылка на новый геообъект;
- `prevChild`: [IGeoObject](#) — ссылка на предыдущее значение для данного индекса.

**Описание методов****add**

```
{IGeoObjectCollection} add(child[, index])
```

Добавляет (вставляет) дочерний геообъект в коллекцию.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                                                                                                                           |
|----------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <code>child</code> * | —                     | Тип: <a href="#">IGeoObject</a><br><br>Дочерний геообъект.                                                                         |
| <code>index</code>   | —                     | Тип: <code>Integer</code><br><br>Индекс, в который добавляется новый объект.<br>По умолчанию объект добавляется в конец коллекции. |

\* Обязательный параметр/опция.

**each**

```
{ } each(callback[, context])
```

Для каждого дочернего геообъекта, вызывает функцию-обработчик.

**Параметры:**



| Параметр                | Значение по умолчанию | Описание                                                    |
|-------------------------|-----------------------|-------------------------------------------------------------|
| <code>callback *</code> | —                     | Тип: Function<br><br>Функция-обработчик.                    |
| <code>context</code>    | —                     | Тип: Object<br><br>Контекст выполнения функции-обработчика. |

\* Обязательный параметр/опция.

### get

```
{IGeoObject} get(index)
```

**Возвращает** дочерний геообъект с заданным индексом.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                    |
|----------------------|-----------------------|-----------------------------|
| <code>index *</code> | —                     | Тип: Integer<br><br>Индекс. |

\* Обязательный параметр/опция.

### getBounds

```
{Number[][]|null} getBounds()
```

**Возвращает** географические координаты области, охватывающей коллекции и ее дочерние геообъекты.

### getIterator

```
{Iterator} getIterator()
```

**Возвращает** итератор по коллекции.

### getLength

```
{Integer} getLength()
```

**Возвращает** длину коллекции.

### getPixelBounds

```
{Number[][]|null} getPixelBounds()
```

**Возвращает** глобальные пиксельные координаты области, охватывающей коллекцию и её дочерние геообъекты.

### indexOf

```
{Integer} indexOf(object)
```

**Возвращает** индекс дочернего геообъекта. Если геообъект в коллекции не найден, возвращается -1.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                            |
|-----------------------|-----------------------|-----------------------------------------------------|
| <code>object *</code> | —                     | Тип: <code>Object</code><br><br>Дочерний геообъект. |

\* Обязательный параметр/опция.

**remove**

```
{IGeoObjectCollection} remove(child)
```

Удаляет дочерний геообъект из коллекции.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                                                 |
|----------------------|-----------------------|----------------------------------------------------------|
| <code>child *</code> | —                     | Тип: <code>IGeoObject</code><br><br>Удаляемый геообъект. |

\* Обязательный параметр/опция.

**removeAll**

```
{IGeoObjectCollection} removeAll()
```

Очищает коллекцию.

**Возвращает** ссылку на себя.

**set**

```
{IGeoObjectCollection} set(index, child)
```

Записывает в коллекцию новый дочерний геообъект.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                                                |
|----------------------|-----------------------|---------------------------------------------------------|
| <code>index *</code> | —                     | Тип: <code>Integer</code><br><br>Индекс.                |
| <code>child *</code> | —                     | Тип: <code>IGeoObject</code><br><br>Дочерний геообъект. |

\* Обязательный параметр/опция.

**splice**

```
{IGeoObjectCollection} splice(index, number)
```

Удаляет геообъекты из коллекции. Если необходимо, вставляет на их место другие. Объекты, которые будут вставлены вместо удаляемых, передаются как дополнительные параметры (после параметра `number`).

**Возвращает** коллекцию удаленных геообъектов.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                               |
|-----------------------|-----------------------|------------------------------------------------------------------------|
| <code>index *</code>  | —                     | Тип: Integer<br><br>Индекс геообъекта, с которого начинается удаление. |
| <code>number *</code> | —                     | Тип: Integer<br><br>Количество удаляемых геообъектов.                  |

\* Обязательный параметр/опция.

## IGeoObjectPopupData

Статический объект.

Объект с данными, передаваемый геообъектом своему инфо-объекту (например, балуну или хинту).

**Поля**

**Поля**

| Имя                     | Тип                       | Описание                        |
|-------------------------|---------------------------|---------------------------------|
| <code>geometry</code>   | <code>IGeometry</code>    | Ссылка на геометрию геообъекта. |
| <code>geoObject</code>  | <code>IGeoObject</code>   | Ссылка на геообъект.            |
| <code>properties</code> | <code>IDataManager</code> | Ссылка на свойства геообъекта.  |
| <code>userData</code>   | <code>Object</code>       | Пользовательские данные.        |

**Описание полей**

### `geometry`

```
{IGeometry} geometry
```

Ссылка на геометрию геообъекта.

### `geoObject`

```
{IGeoObject} geoObject
```

Ссылка на геообъект.

### `properties`

```
{IDataManager} properties
```

Ссылка на свойства геообъекта.

**userData**`{Object} userData`

Пользовательские данные.

**IGeoObjectSequence**

Расширяет [ICustomizable](#), [IEventEmitter](#), [IParentOnMap](#).

Интерфейс неизменяемой коллекции геообъектов.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**`IGeoObjectSequence()`**Поля**

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

**События**

| Имя                               | Описание                                                                                                                                                                                                       |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">boundschange</a>      | Изменение координат географической области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса <a href="#">Event</a> .                                                                          |
| <a href="#">mapchange</a>         | Сменилась карта. Поля данных: <ul style="list-style-type: none"> <li><code>oldMap</code> - старая карта;</li> <li><code>newMap</code> - новая карта.</li> </ul> Унаследовано от <a href="#">IParentOnMap</a> . |
| <a href="#">optionschange</a>     | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                 |
| <a href="#">pixelboundschange</a> | Изменение пиксельных координат области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса <a href="#">Event</a> .                                                                              |

**Методы**

| Имя                                       | Возвращает                 | Описание                                                       |
|-------------------------------------------|----------------------------|----------------------------------------------------------------|
| <a href="#">each(callback[, context])</a> |                            | Для каждого дочернего геообъекта, вызывает функцию-обработчик. |
| <a href="#">get(index)</a>                | <a href="#">IGeoObject</a> | Возвращает дочерний геообъект с заданным индексом.             |

| Имя                              | Возвращает               | Описание                                                                                              |
|----------------------------------|--------------------------|-------------------------------------------------------------------------------------------------------|
| <a href="#">getBounds()</a>      | Number[][] null          | Возвращает географические координаты области, охватывающей коллекции и ее дочерние геообъекты.        |
| <a href="#">getIterator()</a>    | <a href="#">Iterator</a> | Возвращает итератор по дочерним геообъектам коллекции.                                                |
| <a href="#">getLength()</a>      | Integer                  | Возвращает длину коллекции.                                                                           |
| <a href="#">getMap()</a>         | <a href="#">Map</a>      | Возвращает ссылку на карту.<br>Унаследован от <a href="#">IParentOnMap</a> .                          |
| <a href="#">getPixelBounds()</a> | Number[][] null          | Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и ее дочерние геообъекты. |
| <a href="#">indexOf(object)</a>  | Integer                  | Возвращает индекс дочернего геообъекта. Если геообъект в коллекции не найден, возвращается -1.        |

## Описание событий

### boundschange

Изменение координат географической области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса [Event](#).

### pixelboundschange

Изменение пиксельных координат области, охватывающей коллекцию и ее дочерние геообъекты. Экземпляр класса [Event](#).

## Описание методов

### each

```
{ } each(callback[, context])
```

Для каждого дочернего геообъекта, вызывает функцию-обработчик.

#### Параметры:

| Параметр                   | Значение по умолчанию | Описание                             |
|----------------------------|-----------------------|--------------------------------------|
| <a href="#">callback</a> * | —                     | Тип: Function<br>Функция-обработчик. |

| Параметр             | Значение по умолчанию | Описание                                                    |
|----------------------|-----------------------|-------------------------------------------------------------|
| <code>context</code> | —                     | Тип: Object<br><br>Контекст выполнения функции-обработчика. |

\* Обязательный параметр/опция.

### get

```
{IGeoObject} get(index)
```

**Возвращает** дочерний геообъект с заданным индексом.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                    |
|----------------------|-----------------------|-----------------------------|
| <code>index *</code> | —                     | Тип: Integer<br><br>Индекс. |

\* Обязательный параметр/опция.

### getBounds

```
{Number[][]|null} getBounds()
```

**Возвращает** географические координаты области, охватывающей коллекции и ее дочерние геообъекты.

### getIterator

```
{IIterator} getIterator()
```

**Возвращает** итератор по дочерним геообъектам коллекции.

### getLength

```
{Integer} getLength()
```

**Возвращает** длину коллекции.

### getPixelBounds

```
{Number[][]|null} getPixelBounds()
```

**Возвращает** глобальные пиксельные координаты области, охватывающей коллекцию и её дочерние геообъекты.

### indexOf

```
{Integer} indexOf(object)
```

**Возвращает** индекс дочернего геообъекта. Если геообъект в коллекции не найден, возвращается -1.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                               |
|-----------------------|-----------------------|--------------------------------------------------------|
| <code>object</code> * | —                     | Тип: <a href="#">IGeoObject</a><br>Дочерний геообъект. |

\* Обязательный параметр/опция.

## IGroupControlLayout

Расширяет [ILayout](#).

Интерфейс макета группового элемента управления.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IGroupControlLayout(data)
```

### Параметры:

| Параметр            | Значение по умолчанию | Описание                                   |
|---------------------|-----------------------|--------------------------------------------|
| <code>data</code> * | —                     | Тип: <code>Object</code><br>Данные макета. |

\* Обязательный параметр/опция.

### Поля

| Имя                 | Тип                           | Описание                                                                |
|---------------------|-------------------------------|-------------------------------------------------------------------------|
| <code>events</code> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

### События

| Имя                               | Описание                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>childcontainerchange</code> | Изменение контейнера дочерних элементов.<br>Поля данных: <ul style="list-style-type: none"> <li><code>newChildContainerElement</code> - новый элемент для дочерних объектов;</li> <li><code>oldChildContainerElement</code> - старый элемент для дочерних объектов.</li> </ul>                                               |
| <code>click</code>                | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <code>contextmenu</code>          | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">dblclick</a>        | <p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">emptinesschange</a> | <p>Изменение признака пустоты макета. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">mousedown</a>       | <p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">mouseenter</a>      | <p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <a href="#">mouseleave</a>      | <p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">mousemove</a>       | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">mouseup</a>         | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <a href="#">multitouchend</a>   | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchmove</a>  | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |



| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchstart</a>     | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">parentelementchange</a> | <p>Изменение родительского элемента. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <a href="#">shapechange</a>         | <p>Изменение фигуры области охватывающей макет. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">wheel</a>               | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Методы

| Имя                                        | Возвращает  | Описание                                                                                                      |
|--------------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>                  |             | <p>Деструктор. Вызывается при завершении работы с макетом.</p> <p>Унаследован от <a href="#">ILayout</a>.</p> |
| <a href="#">getChildContainerElement()</a> | Object      | <p>Возвращает ссылку на DOM-элемент, в который должны быть добавлены дочерние элементы.</p>                   |
| <a href="#">getData()</a>                  | Object      | <p>Возвращает объект данных макета.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                        |
| <a href="#">getParentElement()</a>         | HTMLElement | <p>Возвращает родительский HTML-элемент.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                   |

| Имя                                      | Возвращает                   | Описание                                                                                                                                                                                                                                                                  |
|------------------------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getShape()</a>               | <a href="#">IShape</a>  null | Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.<br>Унаследован от <a href="#">ILayout</a> . |
| <a href="#">isEmpty()</a>                | Boolean                      | Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.<br>Унаследован от <a href="#">ILayout</a> .                                                                        |
| <a href="#">setData(data)</a>            |                              | Задаёт данные макета.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                         |
| <a href="#">setParentElement(parent)</a> |                              | Добавляет макет в DOM-дерево.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                 |

## Описание событий

### childcontainerchange

Изменение контейнера дочерних элементов.

Поля данных:

- `newChildContainerElement` - новый элемент для дочерних объектов;
- `oldChildContainerElement` - старый элемент для дочерних объектов.

## Описание методов

### getChildContainerElement

```
{Object} getChildContainerElement()
```

**Возвращает** ссылку на DOM-элемент, в который должны быть добавлены дочерние элементы.

## IHint

Расширяет [IPopup](#).

Интерфейс хинта.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IHint()
```

## Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

## События

| Имя                           | Описание                                                                       |
|-------------------------------|--------------------------------------------------------------------------------|
| <a href="#">close</a>         | Заккрытие инфо-объекта.<br>Унаследовано от <a href="#">IPopup</a> .            |
| <a href="#">open</a>          | Открытие инфо-объекта.<br>Унаследовано от <a href="#">IPopup</a> .             |
| <a href="#">optionschange</a> | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> . |

## Методы

| Имя                                      | Возвращает                  | Описание                                                                                                                                                                                                                                                       |
|------------------------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">close([force])</a>           | <a href="#">vow.Promise</a> | Закрывает инфо-объект.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                                              |
| <a href="#">getData()</a>                |                             | Возвращает данные инфо-объекта.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                                     |
| <a href="#">getOverlay()</a>             | <a href="#">vow.Promise</a> | Возвращает объект-обещание вернуть оверлей.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                         |
| <a href="#">getOverlaySync()</a>         | <a href="#">IOverlay</a>    | Возвращает оверлей, если тот существует.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                            |
| <a href="#">getPosition()</a>            |                             | Возвращает координаты инфо-объекта.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                                 |
| <a href="#">isOpen()</a>                 | Boolean                     | Возвращает состояние инфо-объекта: открыт/закрыт.<br>Унаследован от <a href="#">IPopup</a> .                                                                                                                                                                   |
| <a href="#">open([position[, data]])</a> | <a href="#">vow.Promise</a> | Открывает инфо-объект в указанной позиции. Если инфо-объект уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией <a href="#">IProjection</a> , которая содержится в опциях.<br>Унаследован от <a href="#">IPopup</a> . |

| Имя                                   | Возвращает                  | Описание                                                                      |
|---------------------------------------|-----------------------------|-------------------------------------------------------------------------------|
| <a href="#">setData(data)</a>         | <a href="#">vow.Promise</a> | Задаёт инфо-объекту новые данные.<br>Унаследован от <a href="#">IPopup</a> .  |
| <a href="#">setPosition(position)</a> | <a href="#">vow.Promise</a> | Задаёт инфо-объекту новую позицию.<br>Унаследован от <a href="#">IPopup</a> . |

## IHintManager

Расширяет [IPopupManager](#).

Интерфейс менеджера хинта.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IHintManager()
```

### Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

### События

| Имя                   | Описание                                                                                                                                                                                                                                                           |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">close</a> | Заккрытие инфо-объекта. Названия полей доступны через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>target</code> - ссылка на объект, на котором был закрыт инфо-объект.</li></ul> Унаследовано от <a href="#">IPopupManager</a> . |
| <a href="#">open</a>  | Открытие инфо-объекта. Названия полей доступны через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>target</code> - ссылка на объект, на котором был открыт инфо-объект.</li></ul> Унаследовано от <a href="#">IPopupManager</a> .  |

### Методы

| Имя                            | Возвращает                  | Описание                                                                           |
|--------------------------------|-----------------------------|------------------------------------------------------------------------------------|
| <a href="#">close([force])</a> | <a href="#">vow.Promise</a> | Закрывает инфо-объект.<br>Унаследован от <a href="#">IPopupManager</a> .           |
| <a href="#">destroy()</a>      |                             | Отключает менеджер инфо-объекта.<br>Унаследован от <a href="#">IPopupManager</a> . |

| Имя                                                | Возвращает                           | Описание                                                                                            |
|----------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------|
| <a href="#">getData()</a>                          | Object null                          | Возвращает данные инфо-объекта или null.<br>Унаследован от <a href="#">IPopupManager</a> .          |
| <a href="#">getOptions()</a>                       | <a href="#">IOptionManager</a>  null | Возвращает менеджер опций или null.<br>Унаследован от <a href="#">IPopupManager</a> .               |
| <a href="#">getOverlay()</a>                       | <a href="#">vow.Promise</a>          | Возвращает объект-обещание вернуть оверлей.<br>Унаследован от <a href="#">IPopupManager</a> .       |
| <a href="#">getOverlaySync()</a>                   | <a href="#">IOverlay</a>  null       | Возвращает оверлей, если тот существует.<br>Унаследован от <a href="#">IPopupManager</a> .          |
| <a href="#">getPosition()</a>                      | Number[] null                        | Возвращает координаты инфо-объекта или `null`.<br>Унаследован от <a href="#">IPopupManager</a> .    |
| <a href="#">isOpen()</a>                           | Boolean                              | Возвращает состояние инфо-объекта: открыт/закрыт.<br>Унаследован от <a href="#">IPopupManager</a> . |
| <a href="#">open([position[, data[, options]])</a> | <a href="#">vow.Promise</a>          | Открывает инфо-объект в указанной позиции.<br>Унаследован от <a href="#">IPopupManager</a> .        |
| <a href="#">setData(data)</a>                      | <a href="#">vow.Promise</a>          | Задаёт инфо-объекту новые данные.<br>Унаследован от <a href="#">IPopupManager</a> .                 |
| <a href="#">setOptions(options)</a>                | <a href="#">vow.Promise</a>          | Задаёт инфо-объекту новые опции.<br>Унаследован от <a href="#">IPopupManager</a> .                  |
| <a href="#">setPosition(position)</a>              | <a href="#">vow.Promise</a>          | Задаёт инфо-объекту новую позицию.<br>Унаследован от <a href="#">IPopupManager</a> .                |

## IHintOwner

Объект, имеющий хинт, доступный через свойство "hint".

[Конструктор](#) | [Поля](#) | [События](#)

## Конструктор

```
IHintOwner()
```

## Поля

| Имя                  | Тип                          | Описание      |
|----------------------|------------------------------|---------------|
| <a href="#">hint</a> | <a href="#">IHintManager</a> | Хинт объекта. |

## События

| Имя                       | Описание        |
|---------------------------|-----------------|
| <a href="#">hintclose</a> | Скрытие хинта.  |
| <a href="#">hintopen</a>  | Открытие хинта. |

## Описание полей

### hint

```
{IHintManager} hint
```

Хинт объекта.

## Описание событий

### hintclose

Скрытие хинта.

### hintopen

Открытие хинта.

## IHotspot

Расширяет [IDomEventEmitter](#).

Интерфейс фигуры, определяющей геометрию активной области.

**Примечание:** Фигура должна задаваться в глобальных пиксельных координатах.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
IHotspot()
```

## Поля

| Имя                    | Тип                           | Описание                                                                |
|------------------------|-------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

## События

| Имя                           | Описание                                                                                                                                                                                                                                                                                                                             |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>         | <p>Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">contextmenu</a>   | <p>Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>              |
| <a href="#">dblclick</a>      | <p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>     |
| <a href="#">mousedown</a>     | <p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                 |
| <a href="#">mouseenter</a>    | <p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                      |
| <a href="#">mouseleave</a>    | <p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                 |
| <a href="#">mousemove</a>     | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                 |
| <a href="#">mouseup</a>       | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>              |
| <a href="#">multitouchend</a> | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                 |

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchmove</a>  | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a> | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">wheel</a>           | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## Методы

| Имя                               | Возвращает             | Описание                                         |
|-----------------------------------|------------------------|--------------------------------------------------|
| <a href="#">getShape()</a>        | <a href="#">IShape</a> | Возвращает фигуру активной области.              |
| <a href="#">getZIndex()</a>       | Number                 | Возвращает <code>zIndex</code> активной области. |
| <a href="#">setShape(shape)</a>   |                        | Задаёт фигуру активной области.                  |
| <a href="#">setZIndex(zIndex)</a> |                        | Задаёт <code>z-index</code> активной области.    |



## Описание методов

### getShape

```
{IShape} getShape()
```

Возвращает фигуру активной области.

### getZIndex

```
{Number} getZIndex()
```

Возвращает zIndex активной области.

### setShape

```
{ } setShape(shape)
```

Задаёт фигуру активной области.

#### Параметры:

| Параметр | Значение по умолчанию | Описание                   |
|----------|-----------------------|----------------------------|
| shape *  | —                     | Тип: IShape<br><br>Фигура. |

\* Обязательный параметр/опция.

### setZIndex

```
{ } setZIndex(zIndex)
```

Задаёт z-index активной области.

#### Параметры:

| Параметр | Значение по умолчанию | Описание                                                              |
|----------|-----------------------|-----------------------------------------------------------------------|
| zIndex * | —                     | Тип: Number<br><br>z-index, который будет задан для активной области. |

\* Обязательный параметр/опция.

## IHotspotLayerObject

Расширяет [ICustomizable](#), [IDomEventEmitter](#).

Интерфейс объекта слоя активных областей.

**Примечание:** Фигура, описывающая активную область, должна задаваться в глобальных пиксельных координатах.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

IHotspotLayerObject()

**Поля**

| Имя                     | Тип                            | Описание                                                                |
|-------------------------|--------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .      |

**События**

| Имя                         | Описание                                                                                                                                                                                                                                                                                                                     |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>       | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">contextmenu</a> | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">dblclick</a>    | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <a href="#">mousedown</a>   | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mouseenter</a>  | Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                      |
| <a href="#">mouseleave</a>  | Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mousemove</a>   | Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mouseup</a>         | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEventManager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <a href="#">multitouchend</a>   | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="#">multitouchmove</a>  | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li>• <code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li>• <code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li>• <code>pageX</code> - координата касания X касания относительно начала документа;</li> <li>• <code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a> | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li>• <code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li>• <code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li>• <code>pageX</code> - координата касания X касания относительно начала документа;</li> <li>• <code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">optionschange</a>   | <p>Изменение в опциях объекта.</p> <p>Унаследовано от <a href="#">ICustomizable</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">shapechange</a>     | <p>Изменение фигуры, описывающей активную область. Экземпляр класса <a href="#">Event</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">wheel</a>           | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEventManager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

**Методы**

| Имя                                       | Возвращает               | Описание                                        |
|-------------------------------------------|--------------------------|-------------------------------------------------|
| <a href="#">getGeometry()</a>             | Object                   | Возвращает реальную геометрию объекта.          |
| <a href="#">getHotspot()</a>              | <a href="#">IHotspot</a> | Возвращает объект описывающий активную область. |
| <a href="#">getId()</a>                   | Number                   | Возвращает идентификатор объекта.               |
| <a href="#">getProperties()</a>           | Object                   | Возвращает данные объекта.                      |
| <a href="#">setGeometry(geometry)</a>     |                          | Задаёт реальную геометрию объекта.              |
| <a href="#">setId(id)</a>                 |                          | Задаёт идентификатор объекта.                   |
| <a href="#">setProperties(properties)</a> |                          | Задаёт данные объекта.                          |

**Описание событий****shapechange**

Изменение фигуры, описывающей активную область. Экземпляр класса [Event](#).

**Описание методов****getGeometry**

```
{Object} getGeometry()
```

**Возвращает** реальную геометрию объекта.

**getHotspot**

```
{IHotspot} getHotspot()
```

**Возвращает** объект, описывающий активную область.

**getId**

```
{Number} getId()
```

**Возвращает** идентификатор объекта.

**getProperties**

```
{Object} getProperties()
```

**Возвращает** данные объекта.

**setGeometry**

```
{ } setGeometry(geometry)
```

**Задаёт** реальную геометрию объекта.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                                       |
|----------------------------|-----------------------|------------------------------------------------|
| <a href="#">geometry</a> * | —                     | Тип: Object<br><br>Реальная геометрия объекта. |

\* Обязательный параметр/опция.

**setId**

```
{ } setId(id)
```

Задаёт идентификатор объекта.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                                  |
|----------------------|-----------------------|-------------------------------------------|
| <a href="#">id</a> * | —                     | Тип: Number<br><br>Идентификатор объекта. |

\* Обязательный параметр/опция.

**setProperties**

```
{ } setProperties(properties)
```

Задаёт данные объекта.

**Параметры:**

| Параметр                     | Значение по умолчанию | Описание                           |
|------------------------------|-----------------------|------------------------------------|
| <a href="#">properties</a> * | —                     | Тип: Object<br><br>Данные объекта. |

\* Обязательный параметр/опция.

**IHotspotObjectSource**

Расширяет [ICustomizable](#).

Источник объектов для слоёв активных областей.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
IHotspotObjectSource()
```

**Поля**

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

**События**

| Имя                           | Описание                                                                       |
|-------------------------------|--------------------------------------------------------------------------------|
| <a href="#">optionschange</a> | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> . |

**Методы**

| Имя                                                               | Описание                                                                                                                                                                        |
|-------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">cancelLastRequest()</a>                               | Отменяет последний запрос за данными.                                                                                                                                           |
| <a href="#">requestObjects(layer, tileNumber, zoom, callback)</a> | Строит массив объектов <a href="#">IHotspotLayerObject</a> , соответствующих определенному слою, тайлу и коэффициенту масштабирования карты, и передает его в функцию callback. |

**Описание методов****cancelLastRequest**

```
{ } cancelLastRequest()
```

Отменяет последний запрос за данными.

**requestObjects**

```
{ } requestObjects(layer, tileNumber, zoom, callback)
```

Строит массив объектов [IHotspotLayerObject](#), соответствующих определенному слою, тайлу и коэффициенту масштабирования карты, и передает его в функцию callback.

**Параметры:**

| Параметр                     | Значение по умолчанию | Описание                                                          |
|------------------------------|-----------------------|-------------------------------------------------------------------|
| <a href="#">layer</a> *      | —                     | Тип: <a href="#">hotspot.Layer</a><br><br>Слой активных областей. |
| <a href="#">tileNumber</a> * | —                     | Тип: <a href="#">Number[]</a><br><br>Тайловые координаты.         |
| <a href="#">zoom</a> *       | —                     | Тип: <a href="#">Number</a><br><br>Уровень масштабирования.       |

| Параметр                | Значение по умолчанию | Описание                             |
|-------------------------|-----------------------|--------------------------------------|
| <code>callback *</code> | —                     | Тип: Function<br>Функция-обработчик. |

\* Обязательный параметр/опция.

## IHotspotShape

Расширяет [ICustomizable](#), [IDomEventEmitter](#).

Интерфейс фигуры, определяющей геометрию активной области.

**Примечание:** Не поддерживается начиная с версии 2.1.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IHotspotShape()
```

### Поля

| Имя                  | Тип                            | Описание                                                                |
|----------------------|--------------------------------|-------------------------------------------------------------------------|
| <code>events</code>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <code>options</code> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .      |

### События

| Имя                          | Описание                                                                                                                                                                                                                                                                                                                     |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>click</code>           | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <code>containerchange</code> | Изменение контейнера. Экземпляр класса <a href="#">Event</a> .                                                                                                                                                                                                                                                               |
| <code>contextmenu</code>     | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <code>dblclick</code>        | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <code>mousedown</code>       | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |

| Имя                            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mouseenter</a>     | <p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">mouseleave</a>     | <p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">mousemove</a>      | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">mouseup</a>        | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">multitouchend</a>  | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchmove</a> | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |



| Имя                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchstart</a> | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">optionschange</a>   | <p>Изменение в опциях объекта.</p> <p>Унаследовано от <a href="#">ICustomizable</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <a href="#">shapechange</a>     | <p>Изменение фигуры, описывающей активную область. Экземпляр класса <a href="#">Event</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">wheel</a>           | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Методы

| Имя                                       | Возвращает                     | Описание                                         |
|-------------------------------------------|--------------------------------|--------------------------------------------------|
| <a href="#">getContainer()</a>            | <code>IHotspotContainer</code> | Возвращает контейнер активных областей.          |
| <a href="#">getGeometry()</a>             | <code>Object</code>            | Возвращает реальную геометрию фигуры.            |
| <a href="#">getId()</a>                   | <code>Number</code>            | Возвращает идентификатор объекта.                |
| <a href="#">getProperties()</a>           | <code>Object</code>            | Возвращает данные объекта.                       |
| <a href="#">getShape()</a>                | <a href="#">IShape</a>         | Возвращает фигуру, описывающую активную область. |
| <a href="#">setGeometry(geometry)</a>     |                                | Задаёт реальную геометрию фигуры.                |
| <a href="#">setId(id)</a>                 |                                | Задаёт идентификатор объекта.                    |
| <a href="#">setProperties(properties)</a> |                                | Задаёт данные объекта.                           |
| <a href="#">setShape(shape)</a>           |                                |                                                  |

**Описание событий****containerchange**

Изменение контейнера. Экземпляр класса [Event](#).

**shapechange**

Изменение фигуры, описывающей активную область. Экземпляр класса [Event](#).

**Описание методов****getContainer**

```
{IHotspotContainer} getContainer()
```

**Возвращает** контейнер активных областей.

**getGeometry**

```
{Object} getGeometry()
```

**Возвращает** реальную геометрию фигуры.

**getId**

```
{Number} getId()
```

**Возвращает** идентификатор объекта.

**getProperties**

```
{Object} getProperties()
```

**Возвращает** данные объекта.

**getShape**

```
{IShape} getShape()
```

**Возвращает** фигуру, описывающую активную область.

**setGeometry**

```
{ } setGeometry(geometry)
```

Задаёт реальную геометрию фигуры.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                                      |
|----------------------------|-----------------------|-----------------------------------------------|
| <a href="#">geometry</a> * | —                     | Тип: Object<br><br>Реальная геометрия фигуры. |

\* Обязательный параметр/опция.

**setId**

```
{ } setId(id)
```

Задаёт идентификатор объекта.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                                  |
|----------------------|-----------------------|-------------------------------------------|
| <a href="#">id</a> * | —                     | Тип: Number<br><br>Идентификатор объекта. |

\* Обязательный параметр/опция.

**setProperties**

```
{ } setProperties(properties)
```

Задаёт данные объекта.

**Параметры:**

| Параметр                     | Значение по умолчанию | Описание                           |
|------------------------------|-----------------------|------------------------------------|
| <a href="#">properties</a> * | —                     | Тип: Object<br><br>Данные объекта. |

\* Обязательный параметр/опция.

**setShape**

```
{ } setShape(shape)
```

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                                                 |
|-------------------------|-----------------------|--------------------------------------------------------------------------|
| <a href="#">shape</a> * | —                     | Тип: <a href="#">IShape</a><br><br>Фигура, описывающая активную область. |

\* Обязательный параметр/опция.

**Iterator**

Интерфейс итератора. Итератор позволяет перебрать все составляющие элементы той или иной коллекции.

[Конструктор](#) | [Поля](#) | [Методы](#)

**Конструктор**

```
IIterator()
```

**Поля**

| Имя                            | Тип    | Описание                                                                              |
|--------------------------------|--------|---------------------------------------------------------------------------------------|
| <a href="#">STOP_ITERATION</a> | Object | Объект, возвращаемый методом <a href="#">Iiterator.getNext</a> по окончании перебора. |

**Методы**

| Имя                       | Возвращает                                       | Описание                                                                                                                  |
|---------------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getNext()</a> | Object  <a href="#">Iiterator.STOP_ITERATION</a> | Возвращает ссылку на следующий элемент в коллекции или <a href="#">Iiterator.#STOP_ITERATION</a> , если перебор завершен. |

**Описание полей****STOP\_ITERATION**

```
{Object} STOP_ITERATION
```

Объект, возвращаемый методом [Iiterator.getNext](#) по окончании перебора.

**Описание методов****getNext**

```
{Object|Iiterator.STOP_ITERATION} getNext()
```

**Возвращает** ссылку на следующий элемент коллекции или объект [Iiterator.#STOP\\_ITERATION](#), если перебор завершен.

**ILayer**

Расширяет [IChildOnMap](#), [ICustomizable](#), [IEventEmitter](#).

Интерфейс слоя карты.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
ILayer()
```

**Поля**

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

**События**

| Имя                              | Описание                        |
|----------------------------------|---------------------------------|
| <a href="#">brightnesschange</a> | Событие изменения яркости слоя. |

| Имя                              | Описание                                                                                                                                                                                                                                                                                          |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">copyrightschange</a> | Событие изменения доступной информации об авторских правах.                                                                                                                                                                                                                                       |
| <a href="#">optionschange</a>    | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                                                                                                    |
| <a href="#">parentchange</a>     | Сменился родительский объект.<br>Поля данных: <ul style="list-style-type: none"> <li>oldParent - старый родитель;</li> <li>newParent - новый родитель.</li> </ul> Унаследовано от <a href="#">IChild</a> .                                                                                        |
| <a href="#">tileloadchange</a>   | Событие изменения состояния загрузки тайла. Поля данных: <ul style="list-style-type: none"> <li>readyTileNumber-количество готовых тайлов. Тайл считается готовым, когда она загружен и отрисован. Тип: Number.</li> <li>totalTileNumber-общее количество видимых тайлов. Тип: Number.</li> </ul> |
| <a href="#">zoomrangechange</a>  | Событие изменения доступной информации о диапазоне уровней масштабирования.                                                                                                                                                                                                                       |

## Методы

| Имя                                         | Возвращает                         | Описание                                                                                                                                            |
|---------------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getBrightness()</a>             | Number                             | Необязательный метод.                                                                                                                               |
| <a href="#">getCopyrights(coords, zoom)</a> | <a href="#">vow.Promise</a>        | Необязательный метод.<br>Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования.                    |
| <a href="#">getParent()</a>                 | <a href="#">IParentOnMap</a>  null | Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.<br><br>Унаследован от <a href="#">IChildOnMap</a> . |

| Имя                              | Возвращает               | Описание                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>getZoomRange(point)</code> | <code>vow.Promise</code> | Необязательный метод. Определяет доступный в указанной точке диапазон уровней масштабирования. Если данные есть, то возвращенный объект- <code>promise</code> будет подтвержден и передаст в качестве результата массив из двух чисел - минимальный и максимальный доступный в точке уровень масштабирования. Если данных нет, то <code>promise</code> отклоняется с указанием ошибки. |
| <code>setParent(parent)</code>   | <code>IChildOnMap</code> | Устанавливает родительский объект. Если передать значение <code>null</code> , то элемент управления будет только удален из текущего родительского объекта.<br><br>Унаследован от <code>IChildOnMap</code> .                                                                                                                                                                            |

## Описание событий

### brightnesschange

Событие изменения яркости слоя.

### copyrightschange

Событие изменения доступной информации об авторских правах.

### tileloadchange

Событие изменения состояния загрузки тайла. Поля данных:

- `readyTileNumber`-количество готовых тайлов. Тайл считается готовым, когда она загружен и отрисован. Тип: `Number`.
- `totalTileNumber`-общее количество видимых тайлов. Тип: `Number`.

### zoomrangechange

Событие изменения доступной информации о диапазоне уровней масштабирования.

## Описание методов

### getBrightness

```
{Number} getBrightness()
```

Необязательный метод.

**Возвращает** значение яркости от 0 до 1 (0 — нулевая яркость, 1 — максимальная яркость). В зависимости от суммарной яркости слоев, добавленных на карту, выбирается цвет логотипа и копирайтов на карте.

## getCopyrights

```
{vow.Promise} getCopyrights(coords, zoom)
```

Необязательный метод. Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования.

**Возвращает** Promise, который будет подтверждён массивом строк или DOM-элементов с информацией об авторских правах.

### Параметры:

| Параметр              | Значение по умолчанию | Описание                                                                                               |
|-----------------------|-----------------------|--------------------------------------------------------------------------------------------------------|
| <code>coords</code> * | —                     | Тип: Number[]<br><br>Точка на карте, для которой запрашивается информация об авторских правах.         |
| <code>zoom</code> *   | —                     | Тип: Number<br><br>Уровень масштабирования, для которого запрашивается информация об авторских правах. |

\* Обязательный параметр/опция.

## getZoomRange

```
{vow.Promise} getZoomRange(point)
```

Необязательный метод. Определяет доступный в указанной точке диапазон уровней масштабирования. Если данные есть, то возвращенный объект-promise будет подтвержден и передаст в качестве результата массив из двух чисел - минимальный и максимальный доступный в точке уровень масштабирования. Если данных нет, то promise отклоняется с указанием ошибки.

**Возвращает** объект-promise.

### Параметры:

| Параметр             | Значение по умолчанию | Описание                   |
|----------------------|-----------------------|----------------------------|
| <code>point</code> * | —                     | Тип: Number[]<br><br>Point |

\* Обязательный параметр/опция.

## ILayout

Расширяет [IDomEventEmitter](#).

Интерфейс макета.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
ILayout(data)
```

**Параметры:**

| Параметр               | Значение по умолчанию | Описание                      |
|------------------------|-----------------------|-------------------------------|
| <a href="#">data</a> * | —                     | Тип: Object<br>Данные макета. |

\* Обязательный параметр/опция.

**Поля**

| Имя                    | Тип                           | Описание                                                                |
|------------------------|-------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

**События**

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>           | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">contextmenu</a>     | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">dblclick</a>        | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <a href="#">emptinesschange</a> | Изменение признака пустоты макета. Экземпляр класса <a href="#">Event</a> .                                                                                                                                                                                                                                                  |
| <a href="#">mousedown</a>       | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mouseenter</a>      | Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                      |
| <a href="#">mouseleave</a>      | Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |



| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mousemove</a>           | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">mouseup</a>             | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">multitouchend</a>       | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchmove</a>      | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a>     | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">parentelementchange</a> | <p>Изменение родительского элемента. Экземпляр класса <a href="#">Event</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">shapechange</a>         | <p>Изменение фигуры области охватывающей макет. Экземпляр класса <a href="#">Event</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <a href="#">wheel</a>               | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

**Методы**

| Имя                                      | Возвращает  | Описание                                                                                                                                                                                                                      |
|------------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>                |             | Деструктор. Вызывается при завершении работы с макетом.                                                                                                                                                                       |
| <a href="#">getData()</a>                | Object      | Возвращает объект данных макета.                                                                                                                                                                                              |
| <a href="#">getParentElement()</a>       | HTMLElement | Возвращает родительский HTML-элемент.                                                                                                                                                                                         |
| <a href="#">getShape()</a>               | IShape null | Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета. |
| <a href="#">isEmpty()</a>                | Boolean     | Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.                                                                        |
| <a href="#">setData(data)</a>            |             | Задаёт данные макета.                                                                                                                                                                                                         |
| <a href="#">setParentElement(parent)</a> |             | Добавляет макет в DOM-дерево.                                                                                                                                                                                                 |

**Описание событий****emptinesschange**

Изменение признака пустоты макета. Экземпляр класса [Event](#).

**parentelementchange**

Изменение родительского элемента. Экземпляр класса [Event](#).

**shapechange**

Изменение фигуры области охватывающей макет. Экземпляр класса [Event](#).

**Описание методов****destroy**

```
{ } destroy()
```

Деструктор. Вызывается при завершении работы с макетом.

**getData**

```
{Object} getData()
```

**Возвращает** объект данных макета.

### getParentElement

```
{HTMLElement} getParentElement()
```

**Возвращает** родительский HTML-элемент.

### getShape

```
{IShape|null} getShape()
```

**Возвращает** фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.

### isEmpty

```
{Boolean} isEmpty()
```

**Возвращает** true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.

### setData

```
{ } setData(data)
```

Задаёт данные макета.

#### Параметры:

| Параметр            | Значение по умолчанию | Описание                      |
|---------------------|-----------------------|-------------------------------|
| <code>data</code> * | —                     | Тип: Object<br>Данные макета. |

\* Обязательный параметр/опция.

### setParentElement

```
{ } setParentElement(parent)
```

Добавляет макет в DOM-дерево.

#### Параметры:

| Параметр              | Значение по умолчанию | Описание                                                                                                                                                           |
|-----------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>parent</code> * | —                     | Тип: HTMLElement null<br>Родительский html-элемент. Родительский элемент должен быть добавлен в DOM-дерево. Если передан null, то элемент удаляется из DOM-дерева. |

\* Обязательный параметр/опция.

## ILinearRingGeometryAccess

Расширяет [IFreezable](#).

Интерфейс доступа к геометрии "Замкнутый контур".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
ILinearRingGeometryAccess()
```

### Поля

| Имя                    | Тип                           | Описание                                                                  |
|------------------------|-------------------------------|---------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий объекта.<br>Унаследовано от <a href="#">IFreezable</a> . |

### События

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> — старые координаты;</li><li><code>newCoordinates</code> - новые координаты;</li><li><code>oldFillRule</code> - старый алгоритм заливки;</li><li><code>newFillRule</code> - новые алгоритм заливки.</li></ul> |

### Методы

| Имя                                        | Возвращает                           | Описание                                                                                             |
|--------------------------------------------|--------------------------------------|------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a>         | Boolean                              | Проверяет, лежит ли переданная точка внутри контура.                                                 |
| <a href="#">freeze()</a>                   | <a href="#">IFreezable</a>           | Переводит объект в "замороженный" режим.<br>Унаследован от <a href="#">IFreezable</a> .              |
| <a href="#">get(index)</a>                 | Number[]                             | Возвращает координаты точки с заданным индексом.                                                     |
| <a href="#">getChildGeometry(index)</a>    | <a href="#">IPointGeometryAccess</a> | Создает и возвращает объект <a href="#">IPointGeometryAccess</a> для заданной вершины ломаной линии. |
| <a href="#">getClosest(anchorPosition)</a> | Object                               | Ищет на контуре точку, ближайшую к <code>anchorPosition</code> .                                     |
| <a href="#">getCoordinates()</a>           | Number[][]                           | Возвращает массив координат геометрии.                                                               |

| Имя                                         | Возвращает                                | Описание                                                                                                                                                                                                                |
|---------------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getFillRule()</a>               | String                                    | Возвращает идентификатор алгоритма заливки.                                                                                                                                                                             |
| <a href="#">getLength()</a>                 | Integer                                   | Возвращает количество точек в геометрии.                                                                                                                                                                                |
| <a href="#">insert(index, coordinates)</a>  | <a href="#">ILinearRingGeometryAccess</a> | Добавляет новую точку с заданным индексом.                                                                                                                                                                              |
| <a href="#">isFrozen()</a>                  | Boolean                                   | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br><br>Унаследован от <a href="#">IFreezable</a> .                                                                                         |
| <a href="#">remove(index)</a>               | Number[]                                  | Удаляет точку с заданным индексом.                                                                                                                                                                                      |
| <a href="#">set(index, coordinates)</a>     | <a href="#">ILinearRingGeometryAccess</a> | Задаёт координаты точки с заданным индексом.                                                                                                                                                                            |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">ILinearRingGeometryAccess</a> | Задаёт массив координат геометрии.                                                                                                                                                                                      |
| <a href="#">setFillRule(fillRule)</a>       | <a href="#">ILinearRingGeometryAccess</a> | Задаёт алгоритм заливки контура.                                                                                                                                                                                        |
| <a href="#">splice(index, number)</a>       | Number[][]                                | Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра number. |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>                | Переводит объект в активный режим.<br><br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                   |

## Описание событий

### change

Изменение координат. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldCoordinates` — старые координаты;
- `newCoordinates` - новые координаты;
- `oldFillRule` - старый алгоритм заливки;
- `newFillRule` - новый алгоритм заливки.

## Описание методов

### contains

```
{Boolean} contains(position)
```

Проверяет, лежит ли переданная точка внутри контура.

**Возвращает** признак принадлежности точки контуру.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                        |
|-------------------------|-----------------------|-------------------------------------------------|
| <code>position *</code> | —                     | Тип: <code>Number[]</code><br>Координаты точки. |

\* Обязательный параметр/опция.

**get**

```
{Number[]} get(index)
```

**Возвращает** координаты точки с заданным индексом.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                                   |
|----------------------|-----------------------|--------------------------------------------|
| <code>index *</code> | —                     | Тип: <code>Integer</code><br>Индекс точки. |

\* Обязательный параметр/опция.

**getChildGeometry**

```
{IPointGeometryAccess} getChildGeometry(index)
```

Создает и возвращает объект `IPointGeometryAccess` для заданной вершины ломаной линии.

**Возвращает** объект геометрии "Точка", соответствующий заданной вершине.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                                             |
|----------------------|-----------------------|------------------------------------------------------|
| <code>index *</code> | —                     | Тип: <code>Integer</code><br>Индекс вершины контура. |

\* Обязательный параметр/опция.

**getClosest**

```
{Object} getClosest(anchorPosition)
```

Ищет на контуре точку, ближайшую к `anchorPosition`.

**Возвращает** объект со следующими полями:

- `position` - точка на контуре ближайшая к `anchorPosition`;
- `distance` - расстояние от `anchorPosition` до `position`;
- `closestPointIndex` - индекс вершины, ближайшей к `position`;
- `nextPointIndex` - индекс вершины, следующей за `position`;
- `prevPointIndex` - индекс вершины, предшествующей `position`.

Поля `nextPointIndex`, `prevPointIndex` могут отсутствовать, если `position` совпадает с одной из вершин контура.

**Параметры:**

| Параметр                      | Значение по умолчанию | Описание                                                                                      |
|-------------------------------|-----------------------|-----------------------------------------------------------------------------------------------|
| <code>anchorPosition</code> * | —                     | Тип: Number[]<br><br>Координаты точки, для которой рассчитывается ближайшая точка на контуре. |

\* Обязательный параметр/опция.

**getCoordinates**

```
{Number[][]} getCoordinates()
```

**Возвращает** массив координат геометрии.

**getFillRule**

```
{String} getFillRule()
```

**Возвращает** идентификатор алгоритма заливки.

**getLength**

```
{Integer} getLength()
```

**Возвращает** количество точек в геометрии.

**insert**

```
{ILinearRingGeometryAccess} insert(index, coordinates)
```

Добавляет новую точку с заданным индексом.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                               |
|----------------------------|-----------------------|----------------------------------------|
| <code>index</code> *       | —                     | Тип: Integer<br><br>Индекс точки.      |
| <code>coordinates</code> * | —                     | Тип: Number[]<br><br>Координаты точки. |

\* Обязательный параметр/опция.

**remove**

```
{Number[]} remove(index)
```

Удаляет точку с заданным индексом.

**Возвращает** координаты удаленной точки.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                      |
|----------------------|-----------------------|-------------------------------|
| <code>index *</code> | —                     | Тип: Integer<br>Индекс точки. |

\* Обязательный параметр/опция.

**set**

```
{ILinearRingGeometryAccess} set(index, coordinates)
```

Задаёт координаты точки с заданным индексом.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                           |
|----------------------------|-----------------------|------------------------------------|
| <code>index *</code>       | —                     | Тип: Integer<br>Индекс точки.      |
| <code>coordinates *</code> | —                     | Тип: Number[]<br>Координаты точки. |

\* Обязательный параметр/опция.

**setCoordinates**

```
{ILinearRingGeometryAccess} setCoordinates(coordinates)
```

Задаёт массив координат геометрии.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                                 |
|----------------------------|-----------------------|------------------------------------------|
| <code>coordinates *</code> | —                     | Тип: Number[][]<br>Координаты геометрии. |

\* Обязательный параметр/опция.

**setFillRule**

```
{ILinearRingGeometryAccess} setFillRule(fillRule)
```

Задаёт алгоритм заливки контура.

**Возвращает** ссылку на себя.

**Параметры:**



| Параметр                | Значение по умолчанию | Описание                                            |
|-------------------------|-----------------------|-----------------------------------------------------|
| <code>fillRule</code> * | —                     | Тип: String<br><br>Идентификатор алгоритма заливки. |

\* Обязательный параметр/опция.

## splice

```
{Number[][]} splice(index, number)
```

Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра `number`.

**Возвращает** массив координат удаленных точек.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                                                 |
|-----------------------|-----------------------|------------------------------------------------------------------------------------------|
| <code>index</code> *  | —                     | Тип: Integer<br><br>Индекс, начиная с которого производится удаление и добавление точек. |
| <code>number</code> * | —                     | Тип: Integer<br><br>Количество удаляемых точек.                                          |

\* Обязательный параметр/опция.

## ILineStringGeometry

Расширяет [IGeometry](#), [ILineStringGeometryAccess](#).

Интерфейс геометрии "Ломаная линия".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
ILineStringGeometry()
```

### Поля

| Имя                  | Тип                            | Описание                                                             |
|----------------------|--------------------------------|----------------------------------------------------------------------|
| <code>events</code>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <code>options</code> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

## События

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a>              | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>oldCoordinates</code> — старые координаты;</li> <li><code>newCoordinates</code> - новые координаты;</li> </ul> Унаследовано от <a href="#">ILineStringGeometryAccess</a> . |
| <a href="#">mapchange</a>           | Сменилась карта. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>oldMap</code> - старая карта;</li> <li><code>newMap</code> - новая карта.</li> </ul> Унаследовано от <a href="#">IGeometry</a> .                                               |
| <a href="#">optionschange</a>       | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                                                                                                                                                       |
| <a href="#">pixelgeometrychange</a> | Изменилась пиксельная геометрия. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>pixelGeometry</code> - новая пиксельная геометрия <a href="#">IPixelGeometry</a>.</li> </ul> Унаследовано от <a href="#">IGeometry</a> .                       |

## Методы

| Имя                         | Возвращает                                    | Описание                                                                                                                                                                                                                                   |
|-----------------------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">freeze()</a>    | <a href="#">IFreezable</a>                    | Переводит объект в "замороженный" режим.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                    |
| <a href="#">get(index)</a>  | <code>Number[]</code>                         | Возвращает координаты точки с заданным индексом.<br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                                                                                             |
| <a href="#">getBounds()</a> | <code>Number[][]</code> или <code>null</code> | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.<br>Унаследован от <a href="#">IGeometry</a> . |

| Имя                                         | Возвращает                                | Описание                                                                                                                                                               |
|---------------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getChildGeometry(index)</a>     | <a href="#">IPointGeometryAccess</a>      | Создает и возвращает объект <a href="#">IPointGeometryAccess</a> для заданной вершины ломаной линии.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> . |
| <a href="#">getClosest(anchorPosition)</a>  | Object                                    | Ищет на ломаной линии точку, ближайшую к anchorPosition.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                             |
| <a href="#">getCoordinates()</a>            | Number[][]                                | Возвращает массив координат геометрии.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                               |
| <a href="#">getLength()</a>                 | Integer                                   | Возвращает количество точек в геометрии.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                             |
| <a href="#">getMap()</a>                    | <a href="#">Map</a>  null                 | Возвращает текущую карту.<br><br>Унаследован от <a href="#">IGeometry</a> .                                                                                            |
| <a href="#">getPixelGeometry([options])</a> | <a href="#">IPixelGeometry</a>            | Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.<br><br>Унаследован от <a href="#">IGeometry</a> .                      |
| <a href="#">getType()</a>                   | String                                    | Возвращает строку "LineString".                                                                                                                                        |
| <a href="#">insert(index, coordinates)</a>  | <a href="#">ILineStringGeometryAccess</a> | Добавляет новую точку с заданным индексом.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                           |
| <a href="#">isFrozen()</a>                  | Boolean                                   | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br><br>Унаследован от <a href="#">IFreezable</a> .                                        |
| <a href="#">remove(index)</a>               | Number[]                                  | Удаляет точку с заданным индексом.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                                   |
| <a href="#">set(index, coordinates)</a>     | <a href="#">ILineStringGeometryAccess</a> | Задаёт координаты точки с заданным индексом.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                         |

| Имя                                         | Возвращает                                | Описание                                                                                                                                                                                                                                                                                  |
|---------------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">ILineStringGeometryAccess</a> | Задаёт массив координат геометрии.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> .                                                                                                                                                                                      |
| <a href="#">setMap(map)</a>                 |                                           | Задаёт карту.<br><br>Унаследован от <a href="#">IGeometry</a> .                                                                                                                                                                                                                           |
| <a href="#">splice(index, number)</a>       | Number[][]                                | Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра number.<br><br>Унаследован от <a href="#">ILineStringGeometryAccess</a> . |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>                | Переводит объект в активный режим.<br><br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                                                     |

#### Описание методов

##### getType

```
{String} getType()
```

Возвращает строку "LineString".

## ILineStringGeometryAccess

Расширяет [IFreezable](#).

Интерфейс доступа к геометрии "Ломаная линия".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

#### Конструктор

```
ILineStringGeometryAccess()
```

#### Поля

| Имя                    | Тип                           | Описание                                                                      |
|------------------------|-------------------------------|-------------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий объекта.<br><br>Унаследовано от <a href="#">IFreezable</a> . |

## События

| Имя                    | Описание                                                                                                                                                                                                                                                                                      |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | <p>Изменение координат. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>oldCoordinates</code> — старые координаты;</li> <li><code>newCoordinates</code> - новые координаты;</li> </ul> |

## Методы

| Имя                                         | Возвращает                                | Описание                                                                                                                                                        |
|---------------------------------------------|-------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">freeze()</a>                    | <a href="#">IFreezable</a>                | <p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p>                                                               |
| <a href="#">get(index)</a>                  | <code>Number[]</code>                     | Возвращает координаты точки с заданным индексом.                                                                                                                |
| <a href="#">getChildGeometry(index)</a>     | <a href="#">IPointGeometryAccess</a>      | Создает и возвращает объект <a href="#">IPointGeometryAccess</a> для заданной вершины ломаной линии.                                                            |
| <a href="#">getClosest(anchorPosition)</a>  | <code>Object</code>                       | Ищет на ломаной линии точку, ближайшую к <code>anchorPosition</code> .                                                                                          |
| <a href="#">getCoordinates()</a>            | <code>Number[][]</code>                   | Возвращает массив координат геометрии.                                                                                                                          |
| <a href="#">getLength()</a>                 | <code>Integer</code>                      | Возвращает количество точек в геометрии.                                                                                                                        |
| <a href="#">insert(index, coordinates)</a>  | <a href="#">ILineStringGeometryAccess</a> | Добавляет новую точку с заданным индексом.                                                                                                                      |
| <a href="#">isFrozen()</a>                  | <code>Boolean</code>                      | <p>Возвращает <code>true</code>, если объект находится в "замороженном" режиме, иначе <code>false</code>.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p> |
| <a href="#">remove(index)</a>               | <code>Number[]</code>                     | Удаляет точку с заданным индексом.                                                                                                                              |
| <a href="#">set(index, coordinates)</a>     | <a href="#">ILineStringGeometryAccess</a> | Задаёт координаты точки с заданным индексом.                                                                                                                    |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">ILineStringGeometryAccess</a> | Задаёт массив координат геометрии.                                                                                                                              |

| Имя                                | Возвращает              | Описание                                                                                                                                                                                                                              |
|------------------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>splice(index, number)</code> | <code>Number[][]</code> | Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра <code>number</code> . |
| <code>unfreeze()</code>            | <code>IFreezable</code> | Переводит объект в активный режим.<br>Унаследован от <code>IFreezable</code> .                                                                                                                                                        |

### Описание событий

#### change

Изменение координат. Экземпляр класса `Event`. Имена полей, доступных через метод `Event.get`:

- `oldCoordinates` — старые координаты;
- `newCoordinates` - новые координаты;

### Описание методов

#### get

```
{Number[]} get(index)
```

**Возвращает** координаты точки с заданным индексом.

#### Параметры:

| Параметр             | Значение по умолчанию | Описание                                       |
|----------------------|-----------------------|------------------------------------------------|
| <code>index</code> * | —                     | Тип: <code>Integer</code><br><br>Индекс точки. |

\* Обязательный параметр/опция.

#### Пример:

```
// Отмечаем начало линии меткой:
map.geoObjects.add(
 new ymaps.Placemark(polyline.geometry.get(0), { iconContent: 'A' })
);
```

#### getChildGeometry

```
{IPointGeometryAccess} getChildGeometry(index)
```

Создает и возвращает объект `IPointGeometryAccess` для заданной вершины ломаной линии.

**Возвращает** объект геометрии "Точка", соответствующий заданной вершине.

#### Параметры:

| Параметр             | Значение по умолчанию | Описание                        |
|----------------------|-----------------------|---------------------------------|
| <code>index *</code> | —                     | Тип: Integer<br>Индекс вершины. |

\* Обязательный параметр/опция.

### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на ломаной линии точку, ближайшую к `anchorPosition`.

**Возвращает** объект со следующими полями:

- `position` - точка на ломаной ближайшая к `anchorPosition`;
- `distance` - расстояние от `anchorPosition` до `position`;
- `closestPointIndex` - индекс вершины, ближайшей к `position`;
- `nextPointIndex` - индекс вершины, следующей за `position`;
- `prevPointIndex` - индекс вершины, предшествующей `position`.

Поля `nextPointIndex`, `prevPointIndex` могут отсутствовать, если `position` совпадает с одной из вершин ломаной линии.

**Параметры:**

| Параметр                      | Значение по умолчанию | Описание                                                                                  |
|-------------------------------|-----------------------|-------------------------------------------------------------------------------------------|
| <code>anchorPosition *</code> | —                     | Тип: Number[]<br>Координаты точки, для которой рассчитывается ближайшая точка на ломаной. |

\* Обязательный параметр/опция.

**Пример:**

```
// Удаляем из линии точки по клику на ней:
myPolyline.events.add('click', function (e) {
 myPolyline.geometry.remove(
 myPolyline.geometry.getClosest(e.get('coords')).closestPointIndex
);
});
```

### getCoordinates

```
{Number[][]} getCoordinates()
```

**Возвращает** массив координат геометрии.

### getLength

```
{Integer} getLength()
```

**Возвращает** количество точек в геометрии.

### insert

```
{ILineStyleGeometryAccess} insert(index, coordinates)
```

Добавляет новую точку с заданным индексом.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                           |
|----------------------------|-----------------------|------------------------------------|
| <code>index *</code>       | —                     | Тип: Integer<br>Индекс точки.      |
| <code>coordinates *</code> | —                     | Тип: Number[]<br>Координаты точки. |

\* Обязательный параметр/опция.

**Пример:**

```
// Добавляем новую точку в конце линии по клику на карте.
myMap.events.add('click', function (e) {
 myLineString.insert(myLineString.getLength(), e.get('coords'));
});
```

**remove**

```
{Number[]} remove(index)
```

Удаляет точку с заданным индексом.

**Возвращает** координаты удаленной точки.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                      |
|----------------------|-----------------------|-------------------------------|
| <code>index *</code> | —                     | Тип: Integer<br>Индекс точки. |

\* Обязательный параметр/опция.

**set**

```
{ILineStringGeometryAccess} set(index, coordinates)
```

Задаёт координаты точки с заданным индексом.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                           |
|----------------------------|-----------------------|------------------------------------|
| <code>index *</code>       | —                     | Тип: Integer<br>Индекс точки.      |
| <code>coordinates *</code> | —                     | Тип: Number[]<br>Координаты точки. |



\* Обязательный параметр/опция.

### setCoordinates

```
{ILineStyleGeometryAccess} setCoordinates(coordinates)
```

Задаёт массив координат геометрии.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                                     |
|----------------------------|-----------------------|----------------------------------------------|
| <code>coordinates</code> * | —                     | Тип: Number[][]<br><br>Координаты геометрии. |

\* Обязательный параметр/опция.

### splice

```
{Number[][]} splice(index, number)
```

Удаляет определенное число точек начиная с заданного индекса. При этом на место удаляемых точек могут быть добавлены новые. Координаты новых точек можно передавать дополнительными аргументами после параметра number.

**Возвращает** массив координат удаленных точек.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                                                 |
|-----------------------|-----------------------|------------------------------------------------------------------------------------------|
| <code>index</code> *  | —                     | Тип: Integer<br><br>Индекс, начиная с которого производится удаление и добавление точек. |
| <code>number</code> * | —                     | Тип: Integer<br><br>Количество удаляемых точек.                                          |

\* Обязательный параметр/опция.

**Пример:**

```
// Добавляем новую точку в начало линии по клику на карте.
myMap.events.add('click', function (e) {
 myLineStyle.splice(0, 0, myLineStyle.getLength(), e.get('coords'));
});
```

## IMapAction

Расширяет [IEventEmitter](#).

Интерфейс объекта, управляющего движением карты.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
IMapAction()
```

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## События

| Имя                  | Описание                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">end</a>  | Событие, оповещающее карту об окончании движения.                                                                                                                                                                                                                                                                                                                                                                         |
| <a href="#">tick</a> | Событие, оповещающее карту о следующем шаге движения. Содержит поля: <ul style="list-style-type: none"><li><code>globalPixelCenter</code> - новый центр карты в глобальных пикселах;</li><li><code>zoom</code> - новый коэффициент масштабирования карты;</li><li><code>duration</code> - время, которое отводится на выполнение шага;</li><li><code>timingFunction</code> - функция, описывающая вид движения.</li></ul> |

## Методы

| Имя                                     | Описание                                                                                                                                                                                                                                                                     |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">begin(mapActionManager)</a> | Запускает движение на исполнение картой. Этот метод автоматически вызывается менеджером движений карты. С момента вызова <a href="#">IMapAction.begin</a> менеджер движений слушает события <a href="#">IMapAction.tick</a> и <a href="#">IMapAction.end</a> и исполняет их. |
| <a href="#">end()</a>                   | Останавливает движение.                                                                                                                                                                                                                                                      |

## Описание событий

### **end**

Событие, оповещающее карту об окончании движения.

### **tick**

Событие, оповещающее карту о следующем шаге движения. Содержит поля:

- `globalPixelCenter` - новый центр карты в глобальных пикселах;
- `zoom` - новый коэффициент масштабирования карты;
- `duration` - время, которое отводится на выполнение шага;
- `timingFunction` - функция, описывающая вид движения.

## Описание методов

### begin

```
{ } begin(mapActionManager)
```

Запускает движение на исполнение картой. Этот метод автоматически вызывается менеджером движений карты. С момента вызова [IMapAction.begin](#) менеджер движений слушает события [IMapAction.tick](#) и [IMapAction.end](#) и исполняет их.

#### Параметры:

| Параметр                           | Значение по умолчанию | Описание                                                                                                 |
|------------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------|
| <a href="#">mapActionManager</a> * | —                     | Тип: <a href="#">map.action.Manager</a><br><br>Менеджер действий карты, на которой исполняется движение. |

\* Обязательный параметр/опция.

### end

```
{ } end()
```

Останавливает движение.

## IMapObjectCollection

Расширяет [ICollection](#), [ICustomizable](#), [IParentOnMap](#).

Коллекция объектов на карте.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IMapObjectCollection()
```

### Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

### События

| Имя                 | Описание                                                                                 |
|---------------------|------------------------------------------------------------------------------------------|
| <a href="#">add</a> | Произошло добавление дочернего объекта.<br>Унаследовано от <a href="#">ICollection</a> . |

| Имя                           | Описание                                                                                                                                                                                                       |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mapchange</a>     | Сменилась карта. Поля данных: <ul style="list-style-type: none"> <li><code>oldMap</code> - старая карта;</li> <li><code>newMap</code> - новая карта.</li> </ul> Унаследовано от <a href="#">IParentOnMap</a> . |
| <a href="#">optionschange</a> | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                 |
| <a href="#">remove</a>        | Произошло удаление дочернего объекта.<br>Унаследовано от <a href="#">ICollection</a> .                                                                                                                         |

## Методы

| Имя                            | Возвращает                  | Описание                                                                               |
|--------------------------------|-----------------------------|----------------------------------------------------------------------------------------|
| <a href="#">add(object)</a>    | <a href="#">ICollection</a> | Добавляет в коллекцию дочерний объект.<br>Унаследован от <a href="#">ICollection</a> . |
| <a href="#">getIterator()</a>  | <a href="#">Iterator</a>    | Возвращает итератор по коллекции.<br>Унаследован от <a href="#">ICollection</a> .      |
| <a href="#">getMap()</a>       | <a href="#">Map</a>         | Возвращает ссылку на карту.<br>Унаследован от <a href="#">IParentOnMap</a> .           |
| <a href="#">remove(object)</a> | <a href="#">ICollection</a> | Удаляет из коллекции дочерний объект.<br>Унаследован от <a href="#">ICollection</a> .  |

## IMapState

Объект-описание состояния карты. Объекты с таким интерфейсом возвращаются сервисом загрузки YMapsML.

См.: [geoXml.load](#)

[Конструктор](#) | [Методы](#)

### Конструктор

```
IMapState()
```

### Методы

| Имя                             | Возвращает                  | Описание                                |
|---------------------------------|-----------------------------|-----------------------------------------|
| <a href="#">applyToMap(map)</a> | <a href="#">vow.Promise</a> | Применяет состояние к переданной карте. |

## Описание методов

### applyToMap

```
{vow.Promise} applyToMap(map)
```

Применяет состояние к переданной карте.

**Возвращает** объект-promise.

**Параметры:**

| Параметр | Значение по умолчанию | Описание                               |
|----------|-----------------------|----------------------------------------|
| map *    | —                     | Тип: <a href="#">Map</a><br><br>Карта. |

\* Обязательный параметр/опция.

## IMultiRouteModelJson

Интерфейс объекта-описания модели мультимаршрута.

[Конструктор](#) | [Поля](#)

### Конструктор

```
IMultiRouteModelJson()
```

### Поля

| Имя                             | Тип                                         | Описание                                                                                                                 |
|---------------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <a href="#">params</a>          | <a href="#">IMultiRouteParams</a>           | Параметры маршрутизации.                                                                                                 |
| <a href="#">referencePoints</a> | <a href="#">IMultiRouteReferencePoint[]</a> | Описание опорных точек мультимаршрута. Также смотрите описание параметра <a href="#">IMultiRouteParams.vialIndexes</a> . |

### Описание полей

#### params

```
{IMultiRouteParams} params
```

Параметры маршрутизации.

#### referencePoints

```
{IMultiRouteReferencePoint[]} referencePoints
```

Описание опорных точек мультимаршрута. Также смотрите описание параметра [IMultiRouteParams.vialIndexes](#).

## IMultiRouteParams

Интерфейс объекта, описывающего формат задания параметров модели мультимаршрута.

[Конструктор](#) | [Поля](#)

**Конструктор**

IMultiRouteParams()

**Поля**

| Имя                                 | Тип             | Описание                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">avoidTrafficJams</a>    | Boolean         | Позволяет прокладывать мультимаршрут с учетом информации о текущих пробках на дорогах. Значение по умолчанию: false.                                                                                                                                                                                                                                                     |
| <a href="#">boundedBy</a>           | Number[][] null | Позволяет задать область на карте, где предположительно находятся искомые объекты. Используется, если точки маршрута заданы почтовым адресом, а не координатами. Значение по умолчанию null.                                                                                                                                                                             |
| <a href="#">requestSendInterval</a> | String Number   | Величина временного интервала между запросами к сервису маршрутизации. Может быть задана в миллисекундах, либо оптимальное значение будет рассчитываться автоматически. Значение по умолчанию "auto".                                                                                                                                                                    |
| <a href="#">results</a>             | Integer         | Максимально допустимое число маршрутов, возвращаемых мультимаршрутизатором. Значение по умолчанию 3.                                                                                                                                                                                                                                                                     |
| <a href="#">reverseGeocoding</a>    | Boolean         | Использовать ли обратное геокодирование для точек маршрута, заданных координатами.                                                                                                                                                                                                                                                                                       |
| <a href="#">routingMode</a>         | String          | Тип маршрутизации. Может принимать одно из двух строковых значений: <ul style="list-style-type: none"> <li>"auto" — автомобильная маршрутизация;</li> <li>"masstransit" - маршрутизация с использованием общественного транспорта.</li> <li>"pedestrian" — пешеходная маршрутизация.</li> <li>"bicycle" - велосипедный маршрут.</li> </ul> Значение по умолчанию "auto". |
| <a href="#">searchCoordOrder</a>    | String          | Определяет способ трактовки описаний опорных точек, заданных в виде массивов координат или геометрий. Может принимать одно из двух значений: "longlat", либо "latlong". По умолчанию используется текущее значение параметра <a href="#">coordorder</a> , который задается при подключении API.                                                                          |
| <a href="#">strictBounds</a>        | Boolean         | Искать объекты только внутри области, заданной параметром <a href="#">boundedBy</a> . Используется, если точки маршрута заданы почтовым адресом, а не координатами. Значение по умолчанию: false.                                                                                                                                                                        |
| <a href="#">vialIndexes</a>         | Integer[]       | Индексы транзитных точек во множестве опорных точек мультимаршрута. По умолчанию массив пуст.                                                                                                                                                                                                                                                                            |

**Описание полей****avoidTrafficJams**

{Boolean} avoidTrafficJams

Позволяет прокладывать мультимаршрут с учетом информации о текущих пробках на дорогах. Значение по умолчанию: false.

**boundedBy**

```
{Number[][]|null} boundedBy
```

Позволяет задать область на карте, где предположительно находятся искомые объекты. Используется, если точки маршрута заданы почтовым адресом, а не координатами. Значение по умолчанию null.

**requestSendInterval**

```
{String|Number} requestSendInterval
```

Величина временного интервала между запросами к сервису маршрутизации. Может быть задана в миллисекундах, либо оптимальное значение будет рассчитываться автоматически. Значение по умолчанию "auto".

**results**

```
{Integer} results
```

Максимально допустимое число маршрутов, возвращаемых мультимаршрутизатором. Значение по умолчанию 3.

**reverseGeocoding**

```
{Boolean} reverseGeocoding
```

Использовать ли обратное геокодирование для точек маршрута, заданных координатами.

**routingMode**

```
{String} routingMode
```

Тип маршрутизации. Может принимать одно из двух строковых значений:

- "auto" — автомобильная маршрутизация;
- "masstransit" - маршрутизация с использованием общественного транспорта.
- "pedestrian" — пешеходная маршрутизация.
- "bicycle" - велосипедный маршрут.

Значение по умолчанию "auto".

**searchCoordOrder**

```
{String} searchCoordOrder
```

Определяет способ трактовки описаний опорных точек, заданных в виде массивов координат или геометрий. Может принимать одно из двух значений: "longlat", либо "latlong". По умолчанию используется текущее значение параметра [coordorder](#), который задается при подключении API.

**strictBounds**

```
{Boolean} strictBounds
```

Искать объекты только внутри области, заданной параметром boundedBy. Используется, если точки маршрута заданы почтовым адресом, а не координатами. Значение по умолчанию: false.

**viaIndexes**

```
{Integer[]} viaIndexes
```

Индексы транзитных точек во множестве опорных точек мультимаршрута. По умолчанию массив пуст.

## IMultiRouteReferencePoint

Интерфейс объекта, описывающего формат задания опорной точки мультимаршрута.

[Конструктор](#)

### Конструктор

```
IMultiRouteReferencePoint()
```

Опорная точка может быть задана одним из следующих способов:

- строка, содержащая почтовый адрес опорной точки;
- массив, содержащий широту и долготу опорной точки;
- геометрия `geometry.Point`, описывающая опорную точку.

Опорные точки мультимаршрута могут подразделяться на путевые и транзитные. Путевые точки подразумевают остановку и делят маршрут на т. н. пути. Транзитные точки остановку не подразумевают. Таким образом, при прохождении через транзитную точку, сегмент пути маршрута не прерывается.

## IMultiRouterRouteBalloon

**Примечание:** Конструктор класса `IMultiRouterRouteBalloon` скрыт, так как данный класс не предназначен для самостоятельной инициализации.

Интерфейс объекта-описания модели мультимаршрута.

[События](#) | [Методы](#)

### События

| Имя                   | Описание         |
|-----------------------|------------------|
| <a href="#">close</a> | Закрытие балуна. |
| <a href="#">open</a>  | Открытие балуна. |

### Методы

| Имя                              | Возвращает                  | Описание                                                                                                        |
|----------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------|
| <a href="#">close()</a>          | <a href="#">vow.Promise</a> | Закрывает балун маршрута.                                                                                       |
| <a href="#">isOpen()</a>         | Boolean                     | Возвращает состояние инфо-объекта: открыт/закрыт.                                                               |
| <a href="#">open([position])</a> | <a href="#">vow.Promise</a> | Открывает балун в точке маршрута, ближайшей к заданной точке. Также устанавливает маршрут в качестве активного. |

### Описание событий

#### close

Закрытие балуна.



**open**

Открытие балуна.

**Описание методов****close**

```
{vow.Promise} close()
```

Закрывает балун маршрута.

**Возвращает** объект-promise.

**isOpen**

```
{Boolean} isOpen()
```

**Возвращает** состояние инфо-объекта: открыт/закрыт.

**open**

```
{vow.Promise} open([position])
```

Открывает балун в точке маршрута, ближайшей к заданной точке. Также устанавливает маршрут в качестве активного.

**Возвращает** объект-promise.

**Параметры:**

| Параметр                 | Значение по умолчанию | Описание                                                                                                        |
|--------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------|
| <a href="#">position</a> | —                     | Тип: Number[]<br><br>Точка, где необходимо расположить балун. По умолчанию балун откроется в середине маршрута. |

**IOptionManager**

Расширяет [IChild](#), [IEventEmitter](#), [IFreezable](#).

Интерфейс менеджера опций. Менеджер опций позволяет задавать значения опций, строить иерархию наследования опций, а также разрешать значения опций в контексте существующей иерархии наследования.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
IOptionManager()
```

**Поля**

| Имя                    | Тип                           | Описание                                                                      |
|------------------------|-------------------------------|-------------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий объекта.<br><br>Унаследовано от <a href="#">IFreezable</a> . |

## События

| Имя                          | Описание                                                                                                                                                                                                                                              |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a>       | Изменение опций.                                                                                                                                                                                                                                      |
| <a href="#">parentchange</a> | <p>Сменился родительский объект.</p> <p>Поля данных:</p> <ul style="list-style-type: none"> <li><code>oldParent</code> - старый родитель;</li> <li><code>newParent</code> - новый родитель.</li> </ul> <p>Унаследовано от <a href="#">IChild</a>.</p> |

## Методы

| Имя                                      | Возвращает                           | Описание                                                                                                                                                                                                                                                                     |
|------------------------------------------|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">freeze()</a>                 | <a href="#">IFreezable</a>           | <p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p>                                                                                                                                                                            |
| <a href="#">get(key[, defaultValue])</a> |                                      | Возвращает значение заданной опции в контексте существующей иерархии наследования опций. При вызове данного метода сначала происходит поиск значения в текущем менеджере опций, а затем, если значение не определено, поиск продолжается в иерархии родительских менеджеров. |
| <a href="#">getAll()</a>                 | Object                               | Возвращает ссылку на внутренний хэш хранящий значения опций.                                                                                                                                                                                                                 |
| <a href="#">getName()</a>                | String                               | Возвращает имя менеджера опций.                                                                                                                                                                                                                                              |
| <a href="#">getNative(key)</a>           | Object                               | Возвращает значение заданной опции, определенное на данном уровне иерархии опций, т.е. в данном менеджере.                                                                                                                                                                   |
| <a href="#">getParent()</a>              | <a href="#">IOptionManager</a>  null | Возвращает родительский менеджер опций.                                                                                                                                                                                                                                      |
| <a href="#">isFrozen()</a>               | Boolean                              | <p>Возвращает true, если объект находится в "замороженном" режиме, иначе false.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p>                                                                                                                                        |
| <a href="#">resolve(key[, name])</a>     | Object                               | Метод, предназначенный для вызова дочерними менеджерами опций.                                                                                                                                                                                                               |

| Имя                            | Возвращает              | Описание                                                                       |
|--------------------------------|-------------------------|--------------------------------------------------------------------------------|
| <code>setName(name)</code>     |                         | Задаёт имя менеджера опций.                                                    |
| <code>setParent(parent)</code> | <code>IChild</code>     | Задаёт родительский менеджер опций.                                            |
| <code>unfreeze()</code>        | <code>IFreezable</code> | Переводит объект в активный режим.<br>Унаследован от <code>IFreezable</code> . |

### Описание событий

#### change

Изменение опций.

### Описание методов

#### get

```
{ } get(key[, defaultValue])
```

**Возвращает** значение заданной опции в контексте существующей иерархии наследования опций. При вызове данного метода сначала происходит поиск значения в текущем менеджере опций, а затем, если значение не определено, поиск продолжается в иерархии родительских менеджеров.

#### Параметры:

| Параметр                  | Значение по умолчанию | Описание                              |
|---------------------------|-----------------------|---------------------------------------|
| <code>key</code> *        | —                     | Тип: String<br>Имя опции.             |
| <code>defaultValue</code> | —                     | Тип: Object<br>Значение по умолчанию. |

\* Обязательный параметр/опция.

#### getAll

```
{Object} getAll()
```

**Возвращает** ссылку на внутренний хэш хранящий значения опций.

#### getName

```
{String} getName()
```

**Возвращает** имя менеджера опций.

#### getNative

```
{Object} getNative(key)
```

**Возвращает** значение заданной опции, определенное на данном уровне иерархии опций, т.е. в данном менеджере.

**Параметры:**

| Параметр           | Значение по умолчанию | Описание                  |
|--------------------|-----------------------|---------------------------|
| <code>key *</code> | —                     | Тип: String<br>Имя опции. |

\* Обязательный параметр/опция.

**getParent**

```
{IOptionManager|null} getParent()
```

**Возвращает** родительский менеджер опций.

**resolve**

```
{Object} resolve(key[, name])
```

Метод, предназначенный для вызова дочерними менеджерами опций.

**Возвращает** значение опции в родительском контексте.

**Параметры:**

| Параметр           | Значение по умолчанию | Описание                                      |
|--------------------|-----------------------|-----------------------------------------------|
| <code>key *</code> | —                     | Тип: String<br>Имя опции.                     |
| <code>name</code>  | —                     | Тип: String<br>Имя дочернего менеджера опций. |

\* Обязательный параметр/опция.

**setName**

```
{ } setName(name)
```

Задаёт имя менеджера опций.

**Параметры:**

| Параметр            | Значение по умолчанию | Описание                            |
|---------------------|-----------------------|-------------------------------------|
| <code>name *</code> | —                     | Тип: String<br>Имя менеджера опций. |

\* Обязательный параметр/опция.

**setParent**

```
{IChild} setParent(parent)
```

Задаёт родительский менеджер опций.

**Возвращает** ссылку на себя.

#### Параметры:

| Параметр                 | Значение по умолчанию | Описание                                                                     |
|--------------------------|-----------------------|------------------------------------------------------------------------------|
| <a href="#">parent</a> * | —                     | Тип: <a href="#">IEventManager</a>  null<br><br>Родительский менеджер опций. |

\* Обязательный параметр/опция.

## IOverlay

Расширяет [ICustomizable](#), [IDomEventEmitter](#).

Интерфейс оверлея.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IOverlay()
```

### Поля

| Имя                     | Тип                           | Описание                                                                |
|-------------------------|-------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IEventManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .      |

### События

| Имя                         | Описание                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>       | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">contextmenu</a> | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">datachange</a>  | Изменение данных. Поля данных: <ul style="list-style-type: none"><li>• <code>oldData</code> - старые данные;</li><li>• <code>newData</code> - новые данные.</li></ul>                                                                                                                                                            |
| <a href="#">dblclick</a>    | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |

| Имя                             | Описание                                                                                                                                                                                                                                                                                                        |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">emptinesschange</a> | Изменение признака пустоты оверлея. Экземпляр класса <a href="#">Event</a> .                                                                                                                                                                                                                                    |
| <a href="#">geometrychange</a>  | Изменение геометрии. Поля данных: <ul style="list-style-type: none"> <li>oldGeometry - старая пиксельная геометрия;</li> <li>newGeometry - новая пиксельная геометрия.</li> </ul>                                                                                                                               |
| <a href="#">mapchange</a>       | Сменилась карта. Поля данных: <ul style="list-style-type: none"> <li>oldMap - старая карта;</li> <li>newMap - новая карта.</li> </ul>                                                                                                                                                                           |
| <a href="#">mousedown</a>       | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .    |
| <a href="#">mouseenter</a>      | Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .         |
| <a href="#">mouseleave</a>      | Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .    |
| <a href="#">mousemove</a>       | Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .    |
| <a href="#">mouseup</a>         | Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">multitouchend</a>   | Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                                     |

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchmove</a>  | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a> | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">optionschange</a>   | <p>Изменение в опциях объекта.</p> <p>Унаследовано от <a href="#">ICustomizable</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <a href="#">shapechange</a>     | <p>Изменение фигуры области охватывающей оверлей. Экземпляр класса <a href="#">Event</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">wheel</a>           | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## Методы

| Имя                           | Возвращает                              | Описание                                 |
|-------------------------------|-----------------------------------------|------------------------------------------|
| <a href="#">getData()</a>     | <code>Object</code>                     | Возвращает объект данных оверлея.        |
| <a href="#">getGeometry()</a> | <a href="#">IPixelGeometry</a>          | Возвращает текущую пиксельную геометрию. |
| <a href="#">getMap()</a>      | <a href="#">Map</a>   <code>null</code> | Возвращает ссылку на текущую карту.      |

| Имя                                   | Возвращает                   | Описание                                                                                                                                                   |
|---------------------------------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getShape()</a>            | <a href="#">IShape</a>  null | Возвращает фигуру, описывающую форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру. |
| <a href="#">isEmpty()</a>             | Boolean                      | Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.     |
| <a href="#">setData(data)</a>         |                              | Задаёт данные оверлея.                                                                                                                                     |
| <a href="#">setGeometry(geometry)</a> |                              | Задаёт пиксельную геометрию оверлея.                                                                                                                       |
| <a href="#">setMap(map)</a>           |                              | Задаёт карту, на которой должен отобразиться оверлей.                                                                                                      |

## Описание событий

### datachange

Изменение данных. Поля данных:

- `oldData` - старые данные;
- `newData` - новые данные.

### emptinesschange

Изменение признака пустоты оверлея. Экземпляр класса [Event](#).

### geometrychange

Изменение геометрии. Поля данных:

- `oldGeometry` - старая пиксельная геометрия;
- `newGeometry` - новая пиксельная геометрия.

### mapchange

Сменилась карта. Поля данных:

- `oldMap` - старая карта;
- `newMap` - новая карта.

### shapechange

Изменение фигуры области охватывающей оверлей. Экземпляр класса [Event](#).



## Описание методов

### getData

```
{Object} getData()
```

**Возвращает** объект данных оверлея.

### getGeometry

```
{IPixelGeometry} getGeometry()
```

**Возвращает** текущую пиксельную геометрию.

### getMap

```
{Map|null} getMap()
```

**Возвращает** ссылку на текущую карту.

### getShape

```
{IShape|null} getShape()
```

**Возвращает** фигуру, которая описывает форму области, охватывающей оверлей в глобальных пиксельных координатах, либо null, если нет возможности рассчитать фигуру.

### isEmpty

```
{Boolean} isEmpty()
```

**Возвращает** true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.

### setData

```
{ } setData(data)
```

Задаёт данные оверлея.

#### Параметры:

| Параметр      | Значение по умолчанию | Описание                       |
|---------------|-----------------------|--------------------------------|
| <i>data</i> * | —                     | Тип: Object<br>Данные оверлея. |

\* Обязательный параметр/опция.

### setGeometry

```
{ } setGeometry(geometry)
```

Задаёт пиксельную геометрию оверлея.

#### Параметры:

| Параметр                   | Значение по умолчанию | Описание                                                                                  |
|----------------------------|-----------------------|-------------------------------------------------------------------------------------------|
| <a href="#">geometry</a> * | —                     | Тип: <a href="#">IPixelGeometry</a><br><br>Геометрия в глобальных пиксельных координатах. |

\* Обязательный параметр/опция.

## setMap

```
{ } setMap(map)
```

Задаёт карту, на которой должен отображаться оверлей.

### Параметры:

| Параметр              | Значение по умолчанию | Описание                                               |
|-----------------------|-----------------------|--------------------------------------------------------|
| <a href="#">map</a> * | —                     | Тип: <a href="#">Map</a>  null<br><br>Ссылка на карту. |

\* Обязательный параметр/опция.

## IPane

Расширяет [IEventEmitter](#).

Интерфейс пейна карты. Пейн карты представляет собой объект, размещающий внутри контейнера карты свой DOM-элемент с определенным zIndex. Пейн может служить контейнером для размещения отображений различных элементов карты, таких как тайлы, оверлеи, статические элементы управления картой и т.д. Также пейн может выполнять роль непрозрачного для DOM-событий экрана, на котором могут слушаться события мыши.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IPane()
```

### Поля

| Имя                    | Тип                           | Описание                                                                 |
|------------------------|-------------------------------|--------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br><br>Унаследовано от <a href="#">IEventEmitter</a> . |

### События

| Имя                            | Описание                                                                                                                                             |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">overflowchange</a> | Изменение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Экземпляр <a href="#">IEvent</a> . |
| <a href="#">zindexchange</a>   | Изменение значения zIndex пейна. Экземпляр <a href="#">IEvent</a> .                                                                                  |

**Методы**

| Имя                           | Возвращает          | Описание                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>     |                     | Уничтожает пейн.                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">getElement()</a>  | HTMLElement         | Возвращает ссылку на DOM-контейнер пейна.                                                                                                                                                                                                                                                                                                                                                                                          |
| <a href="#">getMap()</a>      | <a href="#">Map</a> | Возвращает карту, которой принадлежит пейн.                                                                                                                                                                                                                                                                                                                                                                                        |
| <a href="#">getOverflow()</a> | String              | Возвращает значение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений: <ul style="list-style-type: none"><li>• "visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым;</li><li>• "hidden" - область отображения содержимого пейна ограничена контейнером карты.</li></ul> |
| <a href="#">getZIndex()</a>   | Number              | Возвращает zIndex пейна.                                                                                                                                                                                                                                                                                                                                                                                                           |

**Описание событий****overflowchange**

Изменение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Экземпляр [IEvent](#).

**zindexchange**

Изменение значения zIndex пейна. Экземпляр [IEvent](#).

**Описание методов****destroy**

```
{ } destroy()
```

Уничтожает пейн.

**getElement**

```
{HTMLElement} getElement()
```

**Возвращает** ссылку на DOM-контейнер пейна.

### getMap

```
{Map} getMap()
```

**Возвращает** карту, которой принадлежит пейн.

### getOverflow

```
{String} getOverflow()
```

**Возвращает** значение параметра "overflow", определяющего видимость содержимого пейна при выходе за пределы контейнера карты. Данный параметр может принимать одно из следующих строковых значений:

- "visible" - при выходе за пределы контейнера карты содержимое пейна остается видимым;
- "hidden" - область отображения содержимого пейна ограничена контейнером карты.

### getZIndex

```
{Number} getZIndex()
```

**Возвращает** zIndex пейна.

## IPanorama

Интерфейс для описания панорамы.

[Конструктор](#) | [Методы](#)

### Конструктор

```
IPanorama()
```

### Методы

| Имя                                    | Возвращает                                   | Описание                                                                                                                                                                                              |
|----------------------------------------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getAngularBBox()</a>       | Number[]                                     | Возвращает сферические координаты, которые задают на панорамной сфере область, занимаемую изображением. Координаты указываются в формате [thetaTop, phiRight, thetaBottom, phiLeft] (аналогично CSS). |
| <a href="#">getConnectionArrows()</a>  | <a href="#">IPanoramaConnectionArrow</a> []  | Возвращает массив связей панорамы.                                                                                                                                                                    |
| <a href="#">getConnectionMarkers()</a> | <a href="#">IPanoramaConnectionMarker</a> [] | Возвращает массив маркеров-переходов на панораме.                                                                                                                                                     |
| <a href="#">getCoordSystem()</a>       | <a href="#">ICoordSystem</a>                 | Возвращает систему координат, в которой заданы позиции панорамы и всех её маркеров и связей.                                                                                                          |

| Имя                                   | Возвращает                            | Описание                                                                                                                                                                                                                                                                                         |
|---------------------------------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getDefaultDirection()</a> | Number[]                              | Возвращает направление взгляда по умолчанию. Будет использовано плеером при открытии панорамы.                                                                                                                                                                                                   |
| <a href="#">getDefaultSpan()</a>      | Number[]                              | Возвращает размер области обзора по умолчанию. Будет использовано плеером при открытии панорамы.                                                                                                                                                                                                 |
| <a href="#">getGraph()</a>            | <a href="#">IPanoramaGraph</a>  null  | Возвращает граф панорам, связанных с текущей панорамой, для осуществления быстрых переходов.                                                                                                                                                                                                     |
| <a href="#">getMarkers()</a>          | <a href="#">IPanoramaMarker</a> []    | Возвращает массив маркеров на панораме.                                                                                                                                                                                                                                                          |
| <a href="#">getName()</a>             | String                                | Возвращает название панорамы, отображаемое плеером в интерфейсе.                                                                                                                                                                                                                                 |
| <a href="#">getPosition()</a>         | Number[]                              | Возвращает положение панорамы в указанной в опциях системе координат. Задается в формате [lon, lat, height], [lat, lon, height] или [x, y, height] в зависимости от системы и порядка координат. height – высота панорамы, заданная относительно некоторого уровня (не обязательно уровня моря). |
| <a href="#">getTileLevels()</a>       | <a href="#">IPanoramaTileLevel</a> [] | Возвращает массив уровней масштабирования панорамного изображения.                                                                                                                                                                                                                               |
| <a href="#">getTileSize()</a>         | Number[]                              | Возвращает размер тайлов, на которые нарезано панорамное изображение.                                                                                                                                                                                                                            |

## Описание методов

### getAngularBBox

```
{Number[]} getAngularBBox()
```

**Возвращает** сферические координаты, которые задают на панорамной сфере область, занимаемую изображением. Координаты указываются в формате [thetaTop, phiRight, thetaBottom, phiLeft] (аналогично CSS).

### getConnectionArrows

```
{IPanoramaConnectionArrow[]} getConnectionArrows()
```

**Возвращает** массив связей панорамы.

#### getConnectionMarkers

```
{IPanoramaConnectionMarker[]} getConnectionMarkers()
```

**Возвращает** массив маркеров-переходов на панораме.

#### getCoordSystem

```
{ICoordSystem} getCoordSystem()
```

**Возвращает** систему координат, в которой заданы позиции панорамы и всех её маркеров и связей.

#### getDefaultDirection

```
{Number[]} getDefaultDirection()
```

**Возвращает** направление взгляда по умолчанию. Будет использовано плеером при открытии панорамы.

#### getDefaultSpan

```
{Number[]} getDefaultSpan()
```

**Возвращает** размер области обзора по умолчанию. Будет использовано плеером при открытии панорамы.

#### getGraph

```
{IPanoramaGraph|null} getGraph()
```

**Возвращает** граф панорам, связанных с текущей панорамой, для осуществления быстрых переходов.

#### getMarkers

```
{IPanoramaMarker[]} getMarkers()
```

**Возвращает** массив маркеров на панораме.

#### getName

```
{String} getName()
```

**Возвращает** название панорамы, отображаемое плеером в интерфейсе.

#### getPosition

```
{Number[]} getPosition()
```

**Возвращает** положение панорамы в указанной в опциях системе координат. Задается в формате [lon, lat, height], [lat, lon, height] или [x, y, height] в зависимости от системы и порядка координат. height – высота панорамы, заданная относительно некоторого уровня (не обязательно уровня моря).

#### getTileLevels

```
{IPanoramaTileLevel[]} getTileLevels()
```

**Возвращает** массив уровней масштабирования панорамного изображения.

### getTileSize

```
{Number[]} getTileSize()
```

**Возвращает** размер тайлов, на которые нарезано панорамное изображение.

## IPanoramaConnection

Интерфейс, описывающий сущность на панораме, связывающую ее с другой панорамой.

[Конструктор](#) | [Методы](#)

### Конструктор

```
IPanoramaConnection()
```

### Методы

| Имя                                    | Возвращает                  | Описание                                                                                               |
|----------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------|
| <a href="#">getConnectedPanorama()</a> | <a href="#">vow.Promise</a> | Возвращает Promise объект, который будет разрешен объектом связанной панорамы, или отклонен с ошибкой. |

### Описание методов

#### getConnectedPanorama

```
{vow.Promise} getConnectedPanorama()
```

**Возвращает** Promise объект, который будет разрешен объектом связанной панорамы, или отклонен с ошибкой.

## IPanoramaConnectionArrow

Расширяет [IPanoramaConnection](#).

Интерфейс, описывающий связь, которая отображается на панораме в виде интерактивной стрелки.

[Конструктор](#) | [Поля](#) | [Методы](#)

### Конструктор

```
IPanoramaConnectionArrow()
```

### Поля

| Имя                        | Тип                          | Описание                                |
|----------------------------|------------------------------|-----------------------------------------|
| <a href="#">properties</a> | <a href="#">data.Manager</a> | Дополнительные свойства связи панорамы. |

**Методы**

| Имя                                    | Возвращает                  | Описание                                                                                                                                                           |
|----------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getConnectedPanorama()</a> | <a href="#">vow.Promise</a> | Возвращает Promise объект, который будет разрешен объектом связанной панорамы, или отклонен с ошибкой.<br><br>Унаследован от <a href="#">IPanoramaConnection</a> . |
| <a href="#">getDirection()</a>         | Number[]                    | Возвращает направление на панораму, на которую указывает связь из текущей панорамы.                                                                                |
| <a href="#">getPanorama()</a>          | <a href="#">IPanorama</a>   | Возвращает панораму, которой принадлежит связь.                                                                                                                    |

**Описание полей****properties**

```
{data.Manager} properties
```

Дополнительные свойства связи панорамы.

**Описание методов****getDirection**

```
{Number[]} getDirection()
```

**Возвращает** направление на панораму, на которую указывает связь из текущей панорамы.

**getPanorama**

```
{IPanorama} getPanorama()
```

**Возвращает** панораму, которой принадлежит связь.

**IPanoramaConnectionMarker**

Расширяет [IPanoramaConnection](#), [IPanoramaMarker](#).

Интерфейс, описывающий связи панорамы.

[Конструктор](#) | [Поля](#) | [Методы](#)

**Конструктор**

```
IPanoramaConnectionMarker()
```

**Поля**

| Имя                        | Тип                          | Описание                                                                                  |
|----------------------------|------------------------------|-------------------------------------------------------------------------------------------|
| <a href="#">properties</a> | <a href="#">data.Manager</a> | Дополнительные свойства маркера.<br><br>Унаследовано от <a href="#">IPanoramaMarker</a> . |



**Методы**

| Имя                                    | Возвращает                  | Описание                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getConnectedPanorama()</a> | <a href="#">vow.Promise</a> | Возвращает Promise объект, который будет разрешен объектом связанной панорамы, или отклонен с ошибкой.<br><br>Унаследован от <a href="#">IPanoramaConnection</a> .                                                                                                                                                                                                                     |
| <a href="#">getIconSet()</a>           | <a href="#">vow.Promise</a> | Возвращает Promise, который будет разрешен объектом, реализующим интерфейс <a href="#">IPanoramaMarkerIconSet</a> и содержащим изображения всех состояний маркера.<br><br>Унаследован от <a href="#">IPanoramaMarker</a> .                                                                                                                                                             |
| <a href="#">getPanorama()</a>          | <a href="#">IPanorama</a>   | Возвращает панораму, к которой принадлежит маркер.<br><br>Унаследован от <a href="#">IPanoramaMarker</a> .                                                                                                                                                                                                                                                                             |
| <a href="#">getPosition()</a>          | <a href="#">Number[]</a>    | Возвращает положение маркера в системе координат панорамы, которой принадлежит содержащий узел граф. Задается в формате [lon, lat, height], [lat, lon, height] или [x, y, height] в зависимости от системы и порядка координат. height – высота маркера в метрах, заданная относительно того же уровня, что и высота панорамы.<br><br>Унаследован от <a href="#">IPanoramaMarker</a> . |

**IPanoramaGraph**

Интерфейс, описывающий граф панорам.

[Конструктор](#) | [Методы](#)

**Конструктор**

```
IPanoramaGraph()
```

**Методы**

| Имя                           | Возвращает                           | Описание                                       |
|-------------------------------|--------------------------------------|------------------------------------------------|
| <a href="#">getEdges()</a>    | <a href="#">IPanoramaGraphEdge[]</a> | Возвращает массив ребер графа.                 |
| <a href="#">getNodes()</a>    | <a href="#">IPanoramaGraphNode[]</a> | Возвращает массив узлов графа.                 |
| <a href="#">getPanorama()</a> | <a href="#">IPanorama</a>            | Возвращает панораму, которой принадлежит граф. |

**Описание методов****getEdges**

```
{IPanoramaGraphEdge[]} getEdges()
```

**Возвращает** массив ребер графа.

**getNodes**

```
{IPanoramaGraphNode[]} getNodes()
```

**Возвращает** массив узлов графа.

**getPanorama**

```
{IPanorama} getPanorama()
```

**Возвращает** панораму, которой принадлежит граф.

**IPanoramaGraphEdge**

Интерфейс, описывающий ребро графа панорам.

[Конструктор](#) | [Методы](#)

**Конструктор**

```
IPanoramaGraphEdge()
```

**Методы**

| Имя                           | Возвращает                           | Описание                                                 |
|-------------------------------|--------------------------------------|----------------------------------------------------------|
| <a href="#">getEndNodes()</a> | <a href="#">IPanoramaGraphNode[]</a> | Возвращает массив из двух узлов графа, связанных ребром. |

**Описание методов****getEndNodes**

```
{IPanoramaGraphNode[]} getEndNodes()
```

**Возвращает** массив из двух узлов графа, связанных ребром.

## IPanoramaGraphNode

Расширяет [IPanoramaConnection](#).

Интерфейс, описывающий узел графа панорам.

[Конструктор](#) | [Методы](#)

### Конструктор

```
IPanoramaGraphNode()
```

### Методы

| Имя                                    | Возвращает                  | Описание                                                                                                                                                           |
|----------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getConnectedPanorama()</a> | <a href="#">vow.Promise</a> | Возвращает Promise объект, который будет разрешен объектом связанной панорамы, или отклонен с ошибкой.<br><br>Унаследован от <a href="#">IPanoramaConnection</a> . |

## IPanoramaMarker

Интерфейс, описывающий маркеры на панораме.

[Конструктор](#) | [Поля](#) | [Методы](#)

### Конструктор

```
IPanoramaMarker()
```

### Поля

| Имя                        | Тип                          | Описание                         |
|----------------------------|------------------------------|----------------------------------|
| <a href="#">properties</a> | <a href="#">data.Manager</a> | Дополнительные свойства маркера. |

### Методы

| Имя                           | Возвращает                  | Описание                                                                                                                                                           |
|-------------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getIconSet()</a>  | <a href="#">vow.Promise</a> | Возвращает Promise, который будет разрешен объектом, реализующим интерфейс <a href="#">IPanoramaMarkerIconSet</a> и содержащим изображения всех состояний маркера. |
| <a href="#">getPanorama()</a> | <a href="#">IPanorama</a>   | Возвращает панораму, к которой принадлежит маркер.                                                                                                                 |

| Имя                        | Возвращает            | Описание                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>getPosition()</code> | <code>Number[]</code> | Возвращает положение маркера в системе координат панорамы, которой принадлежит содержащий узел граф. Задается в формате <code>[lon, lat, height]</code> , <code>[lat, lon, height]</code> или <code>[x, y, height]</code> в зависимости от системы и порядка координат. <code>height</code> – высота маркера в метрах, заданная относительно того же уровня, что и высота панорамы. |

### Описание полей

#### properties

```
{data.Manager} properties
```

Дополнительные свойства маркера.

### Описание методов

#### getIconSet

```
{vow.Promise} getIconSet()
```

**Возвращает** `Promise`, который будет разрешен объектом, реализующим интерфейс `IPanoramaMarkerIconSet` и содержащим изображения всех состояний маркера.

#### getPanorama

```
{IPanorama} getPanorama()
```

**Возвращает** панораму, к которой принадлежит маркер.

#### getPosition

```
{Number[]} getPosition()
```

**Возвращает** положение маркера в системе координат панорамы, которой принадлежит содержащий узел граф. Задается в формате `[lon, lat, height]`, `[lat, lon, height]` или `[x, y, height]` в зависимости от системы и порядка координат. `height` – высота маркера в метрах, заданная относительно того же уровня, что и высота панорамы.

## IPanoramaMarkerIcon

Интерфейс, описывающий иконку маркера.

[Конструктор](#) | [Поля](#)

### Конструктор

```
IPanoramaMarkerIcon()
```

**Поля**

| Имя                    | Тип                                    | Описание                                                                                                |
|------------------------|----------------------------------------|---------------------------------------------------------------------------------------------------------|
| <a href="#">image</a>  | HTMLCanvasElement,<br>HTMLImageElement | Изображение иконки.                                                                                     |
| <a href="#">offset</a> | Number[]                               | Смещение верхнего левого угла иконки в пикселах относительно проекции точки, к которой привязан маркер. |

**Описание полей****image**

```
{HTMLCanvasElement|HTMLImageElement} image
```

Изображение иконки.

**offset**

```
{Number[]} offset
```

Смещение верхнего левого угла иконки в пикселах относительно проекции точки, к которой привязан маркер.

**IPanoramaMarkerIconSet**

Интерфейс, описывающий набор иконок маркера.

[Конструктор](#) | [Поля](#)

**Конструктор**

```
IPanoramaMarkerIconSet()
```

**Поля**

| Имя                             | Тип                                         | Описание                                                                                                                                        |
|---------------------------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">default</a>         | <a href="#">IPanoramaMarkerIcon</a>         | Иконка состояния по умолчанию.                                                                                                                  |
| <a href="#">expanded</a>        | <a href="#">IPanoramaMarkerIcon</a><br>null | Иконка для "активного" состояния маркера. Маркер переходит в это состояние при клике по нему; при повторном клике маркер становится неактивным. |
| <a href="#">expandedHovered</a> | <a href="#">IPanoramaMarkerIcon</a><br>null | Иконка для "активного подсвеченного" состояния. Маркер переходит в это состояние, когда он активен и над ним находится курсор.                  |
| <a href="#">hovered</a>         | <a href="#">IPanoramaMarkerIcon</a><br>null | Иконка для "подсвеченного" состояния маркера. В него маркер переходит, когда на него наводится курсор мыши.                                     |

**Описание полей****default**

```
{IPanoramaMarkerIcon} default
```

Иконка состояния по умолчанию.

**expanded**

```
{IPanoramaMarkerIcon|null} expanded
```

Иконка для "активного" состояния маркера. Маркер переходит в это состояние при клике по нему; при повторном клике маркер становится неактивным.

**expandedHovered**

```
{IPanoramaMarkerIcon|null} expandedHovered
```

Иконка для "активного подсвеченного" состояния. Маркер переходит в это состояние, когда он активен и над ним находится курсор.

**hovered**

```
{IPanoramaMarkerIcon|null} hovered
```

Иконка для "подсвеченного" состояния маркера. В него маркер переходит, когда на него наводится курсор мыши.

**IPanoramaTileLevel**

Интерфейс для описания уровней масштабирования панорамного изображения.

[Конструктор](#) | [Методы](#)

**Конструктор**

```
IPanoramaTileLevel()
```

**Методы**

| Имя                              | Возвращает | Описание                                                                                                                                                                                                                                                          |
|----------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getImageSize()</a>   | Number[]   | Возвращает размер всего изображения в пикселах.                                                                                                                                                                                                                   |
| <a href="#">getTileUrl(x, y)</a> | String     | Вычисляет URL тайла по его положению в панорамном изображении. Положение задается парой индексов: горизонтальным и вертикальным. Горизонтальный индекс отсчитывается слева с нуля и растет направо. Вертикальный отсчитывается с верха изображения и растет вниз. |

**Описание методов****getImageSize**

```
{Number[]} getImageSize()
```

**Возвращает** размер всего изображения в пикселах.

## getTileUrl

```
{String} getTileUrl(x, y)
```

Вычисляет URL тайла по его положению в панорамном изображении. Положение задается парой индексов: горизонтальным и вертикальным. Горизонтальный индекс отсчитывается слева с нуля и растет направо. Вертикальный отсчитывается с верха изображения и растет вниз.

**Возвращает** URL тайла с переданными индексами.

### Параметры:

| Параметр            | Значение по умолчанию | Описание                                        |
|---------------------|-----------------------|-------------------------------------------------|
| <a href="#">x</a> * | —                     | Тип: Number<br><br>Горизонтальный индекс тайла. |
| <a href="#">y</a> * | —                     | Тип: Number<br><br>Вертикальный индекс тайла.   |

\* Обязательный параметр/опция.

## IParentOnMap

Интерфейс родительского объекта, относящегося к определенному объекту карты.

[Конструктор](#) | [События](#) | [Методы](#)

### Конструктор

```
IParentOnMap()
```

### События

| Имя                       | Описание                                                                                                                           |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mapchange</a> | Сменилась карта. Поля данных: <ul style="list-style-type: none"><li>oldMap - старая карта;</li><li>newMap - новая карта.</li></ul> |

### Методы

| Имя                      | Возвращает          | Описание                    |
|--------------------------|---------------------|-----------------------------|
| <a href="#">getMap()</a> | <a href="#">Map</a> | Возвращает ссылку на карту. |

### Описание событий

#### mapchange

Сменилась карта. Поля данных:

- oldMap - старая карта;
- newMap - новая карта.

## Описание методов

### getMap

```
{Map} getMap()
```

Возвращает ссылку на карту.

## IPixelCircleGeometry

Расширяет [IPixelGeometry](#).

Интерфейс пиксельной геометрии "Круг".

[Конструктор](#) | [Поля](#) | [Методы](#)

### Конструктор

```
IPixelCircleGeometry()
```

### Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

### Методы

| Имя                              | Возвращает      | Описание                                                                                                                                                                                                                                                                                                 |
|----------------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">equals(geometry)</a> | Boolean         | Возвращает true, если переданная геометрия эквивалентна данной.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                   |
| <a href="#">getBounds()</a>      | Number[][] null | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getCoordinates()</a> | Number[]        | Возвращает координаты центра круга.                                                                                                                                                                                                                                                                      |
| <a href="#">getMetaData()</a>    | Object          | Возвращает метаданные пиксельной геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |
| <a href="#">getRadius()</a>      | Number          | Возвращает радиус круга.                                                                                                                                                                                                                                                                                 |



| Имя                           | Возвращает                     | Описание                                                                                                       |
|-------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------|
| <a href="#">getType()</a>     | String                         | Возвращает идентификатор типа геометрии.<br><br>Унаследован от <a href="#">IBaseGeometry</a> .                 |
| <a href="#">scale(factor)</a> | <a href="#">IPixelGeometry</a> | Создает отмасштабированную копию геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .             |
| <a href="#">shift(offset)</a> | <a href="#">IPixelGeometry</a> | Создает копию геометрии сдвинутую на заданную величину.<br><br>Унаследован от <a href="#">IPixelGeometry</a> . |

### Описание методов

#### getCoordinates

```
{Number[]} getCoordinates()
```

**Возвращает** координаты центра круга.

#### getRadius

```
{Number} getRadius()
```

**Возвращает** радиус круга.

## IPixelGeometry

Расширяет [IBaseGeometry](#).

Интерфейс пиксельной геометрии.

[Конструктор](#) | [Поля](#) | [Методы](#)

### Конструктор

```
IPixelGeometry()
```

### Поля

| Имя                    | Тип                           | Описание                                                                 |
|------------------------|-------------------------------|--------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br><br>Унаследовано от <a href="#">IEventEmitter</a> . |

### Методы

| Имя                              | Возвращает | Описание                                                        |
|----------------------------------|------------|-----------------------------------------------------------------|
| <a href="#">equals(geometry)</a> | Boolean    | Возвращает true, если переданная геометрия эквивалентна данной. |

| Имя                           | Возвращает                     | Описание                                                                                                                                                                                                                                                                                                 |
|-------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getBounds()</a>   | Number[][] null                | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getMetaData()</a> | Object                         | Возвращает метаданные пиксельной геометрии.                                                                                                                                                                                                                                                              |
| <a href="#">getType()</a>     | String                         | Возвращает идентификатор типа геометрии.<br><br>Унаследован от <a href="#">IBaseGeometry</a> .                                                                                                                                                                                                           |
| <a href="#">scale(factor)</a> | <a href="#">IPixelGeometry</a> | Создает отмасштабированную копию геометрии.                                                                                                                                                                                                                                                              |
| <a href="#">shift(offset)</a> | <a href="#">IPixelGeometry</a> | Создает копию геометрии сдвинутую на заданную величину.                                                                                                                                                                                                                                                  |

## Описание методов

### equals

```
{Boolean} equals(geometry)
```

Возвращает true, если переданная геометрия эквивалентна данной.

Параметры:

| Параметр                   | Значение по умолчанию | Описание                                                                                |
|----------------------------|-----------------------|-----------------------------------------------------------------------------------------|
| <a href="#">geometry</a> * | —                     | Тип: <a href="#">IPixelGeometry</a><br><br>Геометрия, с которой производится сравнение. |

\* Обязательный параметр/опция.

### getMetaData

```
{Object} getMetaData()
```

Возвращает метаданные пиксельной геометрии.

## scale

```
{IPixelGeometry} scale(factor)
```

Создает отмасштабированную копию геометрии.

**Возвращает** отмасштабированную копию геометрии.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                               |
|-----------------------|-----------------------|----------------------------------------|
| <code>factor</code> * | —                     | Тип: Number<br>Масштабный коэффициент. |

\* Обязательный параметр/опция.

**Пример:**

```
// Уменьшаем геометрию в два раза.
var smallCopy = myPixelGeometry.scale(0.5);
```

## shift

```
{IPixelGeometry} shift(offset)
```

Создает копию геометрии сдвинутую на заданную величину.

**Возвращает** сдвинутую копию геометрии.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                  |
|-----------------------|-----------------------|-------------------------------------------|
| <code>offset</code> * | —                     | Тип: Number[]<br>Величина сдвига по осям. |

\* Обязательный параметр/опция.

**Пример:**

```
// Сдвигаем все координаты геометрии на 200 пикселей влево.
var shifted = myPixelGeometry.shift([-200, 0]);
```

## IPixelLineStringGeometry

Расширяет [IPixelGeometry](#).

Интерфейс пиксельной геометрии "Ломаная линия".

[Конструктор](#) | [Поля](#) | [Методы](#)

### Конструктор

```
IPixelLineStringGeometry()
```

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## Методы

| Имя                                        | Возвращает                     | Описание                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">equals(geometry)</a>           | Boolean                        | Возвращает true, если переданная геометрия эквивалентна данной.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                   |
| <a href="#">getBounds()</a>                | Number[][] null                | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getClosest(anchorPosition)</a> | Object                         | Ищет на ломаной линии точку, ближайшую к anchorPosition.                                                                                                                                                                                                                                                 |
| <a href="#">getCoordinates()</a>           | Number[][]                     | Возвращает координаты линии.                                                                                                                                                                                                                                                                             |
| <a href="#">getLength()</a>                | Integer                        | Возвращает количество точек в геометрии.                                                                                                                                                                                                                                                                 |
| <a href="#">getMetaData()</a>              | Object                         | Возвращает метаданные пиксельной геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |
| <a href="#">getType()</a>                  | String                         | Возвращает идентификатор типа геометрии.<br><br>Унаследован от <a href="#">IBaseGeometry</a> .                                                                                                                                                                                                           |
| <a href="#">scale(factor)</a>              | <a href="#">IPixelGeometry</a> | Создает отмасштабированную копию геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |

| Имя                           | Возвращает                     | Описание                                                                                                       |
|-------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------|
| <a href="#">shift(offset)</a> | <a href="#">IPixelGeometry</a> | Создает копию геометрии сдвинутую на заданную величину.<br><br>Унаследован от <a href="#">IPixelGeometry</a> . |

## Описание методов

### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на ломаной линии точку, ближайшую к anchorPosition.

**Возвращает** объект со следующими полями:

- position - точка на ломаной ближайшая к anchorPosition;
- distance - расстояние от anchorPosition до position;
- closestPointIndex - индекс вершины, ближайшей к position;
- nextPointIndex - индекс вершины, следующей за position;
- prevPointIndex - индекс вершины, предшествующей position.

Поля nextPointIndex, prevPointIndex могут отсутствовать, если position совпадает с одной из вершин ломаной линии.

### Параметры:

| Параметр                         | Значение по умолчанию | Описание                                                                                      |
|----------------------------------|-----------------------|-----------------------------------------------------------------------------------------------|
| <a href="#">anchorPosition</a> * | —                     | Тип: Number[]<br><br>Координаты точки, для которой рассчитывается ближайшая точка на ломаной. |

\* Обязательный параметр/опция.

### getCoordinates

```
{Number[][]} getCoordinates()
```

**Возвращает** координаты линии.

### getLength

```
{Integer} getLength()
```

**Возвращает** количество точек в геометрии.

## IPixelMultiLineGeometry

Расширяет [IPixelGeometry](#).

Интерфейс пиксельной геометрии "Мультилиния".

[Конструктор](#) | [Поля](#) | [Методы](#)

## Конструктор

```
IPixelMultiLineGeometry()
```

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## Методы

| Имя                                        | Возвращает                     | Описание                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">equals(geometry)</a>           | Boolean                        | Возвращает true, если переданная геометрия эквивалентна данной.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                   |
| <a href="#">getBounds()</a>                | Number[][] null                | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getClosest(anchorPosition)</a> | Object                         | Ищет на контуре точку, ближайшую к anchorPosition.                                                                                                                                                                                                                                                       |
| <a href="#">getCoordinates()</a>           | Number[][][]                   | Возвращает координаты мультилиний.                                                                                                                                                                                                                                                                       |
| <a href="#">getLength()</a>                | Integer                        | Возвращает количество линий в мультилинии.                                                                                                                                                                                                                                                               |
| <a href="#">getMetaData()</a>              | Object                         | Возвращает метаданные пиксельной геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |
| <a href="#">getType()</a>                  | String                         | Возвращает идентификатор типа геометрии.<br><br>Унаследован от <a href="#">IBaseGeometry</a> .                                                                                                                                                                                                           |
| <a href="#">scale(factor)</a>              | <a href="#">IPixelGeometry</a> | Создает отмасштабированную копию геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |

| Имя                           | Возвращает                     | Описание                                                                                                       |
|-------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------|
| <a href="#">shift(offset)</a> | <a href="#">IPixelGeometry</a> | Создает копию геометрии сдвинутую на заданную величину.<br><br>Унаследован от <a href="#">IPixelGeometry</a> . |

## Описание методов

### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на контуре точку, ближайшую к anchorPosition.

**Возвращает** объект со следующими полями:

- position - точка на контуре мультимногоугольника ближайшая к anchorPosition;
- distance - расстояние от anchorPosition до position;
- closestPointIndex - индекс вершины мультимногоугольника, ближайшей к position;
- nextPointIndex - индекс вершины мультимногоугольника, следующей за position;
- prevPointIndex - индекс вершины мультимногоугольника, предшествующей position;
- pathIndex - индекс контура мультимногоугольника, к которому относится найденная точка.

Поля nextPointIndex, prevPointIndex могут отсутствовать, если position совпадает с одной из вершин мультимногоугольника.

### Параметры:

| Параметр                         | Значение по умолчанию | Описание                                                                                      |
|----------------------------------|-----------------------|-----------------------------------------------------------------------------------------------|
| <a href="#">anchorPosition</a> * | —                     | Тип: Number[]<br><br>Координаты точки, для которой рассчитывается ближайшая точка на контуре. |

\* Обязательный параметр/опция.

### getCoordinates

```
{Number[][][]} getCoordinates()
```

**Возвращает** координаты мультилинии.

### getLength

```
{Integer} getLength()
```

**Возвращает** число линий в мультилинии.

## IPixelMultiPolygonGeometry

Расширяет [IPixelGeometry](#).

Интерфейс пиксельной геометрии "Мультимногоугольник".

[Конструктор](#) | [Поля](#) | [Методы](#)

## Конструктор

```
IPixelMultiPolygonGeometry()
```

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## Методы

| Имя                                        | Возвращает      | Описание                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a>         | Boolean         | Проверяет лежит ли переданная точка внутри мультимногоугольника.                                                                                                                                                                                                                                         |
| <a href="#">equals(geometry)</a>           | Boolean         | Возвращает true, если переданная геометрия эквивалентна данной.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                   |
| <a href="#">getBounds()</a>                | Number[][] null | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getClosest(anchorPosition)</a> | Object          | Ищет на контуре мультимногоугольника точку, ближайшую к anchorPosition.                                                                                                                                                                                                                                  |
| <a href="#">getCoordinates()</a>           | Number[][][]    | Возвращает координаты мультимногоугольника.                                                                                                                                                                                                                                                              |



| Имя                        | Возвращает | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>getFillRule()</code> | String     | <p>Возвращает строковый идентификатор, определяющий алгоритм заливки мультимногоугольника. Идентификатор может иметь одно из двух значений:</p> <ul style="list-style-type: none"><li>• <code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.</li><li>• <code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.</li></ul> |

| Имя                        | Возвращает                     | Описание                                                                                                   |
|----------------------------|--------------------------------|------------------------------------------------------------------------------------------------------------|
| <code>getLength()</code>   | Integer                        | Возвращает количество многоугольников в мультимногоугольнике.                                              |
| <code>getMetaData()</code> | Object                         | Возвращает метаданные пиксельной геометрии.<br>Унаследован от <a href="#">IPixelGeometry</a> .             |
| <code>getType()</code>     | String                         | Возвращает идентификатор типа геометрии.<br>Унаследован от <a href="#">IBaseGeometry</a> .                 |
| <code>scale(factor)</code> | <a href="#">IPixelGeometry</a> | Создает отмасштабированную копию геометрии.<br>Унаследован от <a href="#">IPixelGeometry</a> .             |
| <code>shift(offset)</code> | <a href="#">IPixelGeometry</a> | Создает копию геометрии сдвинутую на заданную величину.<br>Унаследован от <a href="#">IPixelGeometry</a> . |

## Описание методов

### contains

```
{Boolean} contains(position)
```

Проверяет лежит ли переданная точка внутри мультимногоугольника.

**Возвращает** признак принадлежности точки многоугольнику.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                           |
|----------------------------|-----------------------|------------------------------------|
| <a href="#">position</a> * | —                     | Тип: Number[]<br>Координаты точки. |

\* Обязательный параметр/опция.

### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на контуре мультимногоугольника точку, ближайшую к `anchorPosition`.

**Возвращает** объект со следующими полями:

- `position` - точка на контуре мультимногоугольника ближайшая к `anchorPosition`;
- `distance` - расстояние от `anchorPosition` до `position`;
- `closestPointIndex` - индекс вершины мультимногоугольника, ближайшей к `position`;
- `nextPointIndex` - индекс вершины мультимногоугольника, следующей за `position`;
- `prevPointIndex` - индекс вершины мультимногоугольника, предшествующей `position`;
- `pathIndex` - индекс контура мультимногоугольника, к которому относится найденная точка.

Поля `nextPointIndex`, `prevPointIndex` могут отсутствовать, если `position` совпадает с одной из вершин мультимногоугольника.

#### Параметры:

| Параметр                      | Значение по умолчанию | Описание                                                                                                                  |
|-------------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------|
| <code>anchorPosition</code> * | —                     | Тип: <code>Number[]</code><br><br>Координаты точки, для которой рассчитывается ближайшая точка на контуре многоугольника. |

\* Обязательный параметр/опция.

#### `getCoordinates`

```
{Number[][][][]} getCoordinates()
```

**Возвращает** координаты мультимногоугольника.

#### `getFillRule`

```
{String} getFillRule()
```

**Возвращает** строковый идентификатор, определяющий алгоритм заливки мультимногоугольника. Идентификатор может иметь одно из двух значений:

- `evenOdd` - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.
- `nonZero` - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.

#### `getLength`

```
{Integer} getLength()
```

**Возвращает** количество многоугольников в мультимногоугольнике.

## IPixelPointGeometry

Расширяет [IPixelGeometry](#).

Интерфейс пиксельной геометрии "Точка".

[Конструктор](#) | [Поля](#) | [Методы](#)

#### Конструктор

```
IPixelPointGeometry()
```

**Поля**

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

**Методы**

| Имя                              | Возвращает                     | Описание                                                                                                                                                                                                                                                                                                 |
|----------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">equals(geometry)</a> | Boolean                        | Возвращает true, если переданная геометрия эквивалентна данной.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                   |
| <a href="#">getBounds()</a>      | Number[][] null                | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getCoordinates()</a> | Number[]                       | Возвращает координаты точки.                                                                                                                                                                                                                                                                             |
| <a href="#">getMetaData()</a>    | Object                         | Возвращает метаданные пиксельной геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |
| <a href="#">getType()</a>        | String                         | Возвращает идентификатор типа геометрии.<br><br>Унаследован от <a href="#">IBaseGeometry</a> .                                                                                                                                                                                                           |
| <a href="#">scale(factor)</a>    | <a href="#">IPixelGeometry</a> | Создает отмасштабированную копию геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |
| <a href="#">shift(offset)</a>    | <a href="#">IPixelGeometry</a> | Создает копию геометрии сдвинутую на заданную величину.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                           |

## Описание методов

### getCoordinates

```
{Number[]} getCoordinates()
```

Возвращает координаты точки.

## IPixelPolygonGeometry

Расширяет [IPixelGeometry](#).

Интерфейс пиксельной геометрии "Многоугольник".

[Конструктор](#) | [Поля](#) | [Методы](#)

### Конструктор

```
IPixelPolygonGeometry()
```

### Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

### Методы

| Имя                                        | Возвращает      | Описание                                                                                                                                                                                                                                                                                             |
|--------------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a>         | Boolean         | Проверяет, лежит ли переданная точка внутри многоугольника.                                                                                                                                                                                                                                          |
| <a href="#">equals(geometry)</a>           | Boolean         | Возвращает true, если переданная геометрия эквивалентна данной.<br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                   |
| <a href="#">getBounds()</a>                | Number[][] null | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getClosest(anchorPosition)</a> | Object          | Ищет на контуре многоугольника точку, ближайшую к anchorPosition.                                                                                                                                                                                                                                    |

| Имя                              | Возвращает | Описание                              |
|----------------------------------|------------|---------------------------------------|
| <a href="#">getCoordinates()</a> | Number[][] | Возвращает координаты многоугольника. |

| Имя                           | Возвращает | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getFillRule()</a> | String     | <p>Возвращает строковый идентификатор, определяющий алгоритм заливки многоугольника. Идентификатор может принимать одно из двух значений:</p> <ul style="list-style-type: none"><li>• <code>evenOdd</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.</li><li>• <code>nonZero</code> - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.</li></ul> |

| Имя                        | Возвращает                     | Описание                                                                                                   |
|----------------------------|--------------------------------|------------------------------------------------------------------------------------------------------------|
| <code>getLength()</code>   | Integer                        | Возвращает количество контуров в многоугольнике.                                                           |
| <code>getMetaData()</code> | Object                         | Возвращает метаданные пиксельной геометрии.<br>Унаследован от <a href="#">IPixelGeometry</a> .             |
| <code>getType()</code>     | String                         | Возвращает идентификатор типа геометрии.<br>Унаследован от <a href="#">IBaseGeometry</a> .                 |
| <code>scale(factor)</code> | <a href="#">IPixelGeometry</a> | Создает отмасштабированную копию геометрии.<br>Унаследован от <a href="#">IPixelGeometry</a> .             |
| <code>shift(offset)</code> | <a href="#">IPixelGeometry</a> | Создает копию геометрии сдвинутую на заданную величину.<br>Унаследован от <a href="#">IPixelGeometry</a> . |

## Описание методов

### contains

```
{Boolean} contains(position)
```

Проверяет, лежит ли переданная точка внутри многоугольника.

**Возвращает** признак принадлежности точки многоугольнику.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                           |
|----------------------------|-----------------------|------------------------------------|
| <a href="#">position</a> * | —                     | Тип: Number[]<br>Координаты точки. |

\* Обязательный параметр/опция.

### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на контуре многоугольника точку, ближайшую к `anchorPosition`.

**Возвращает** объект со следующими полями:

- `position` - точка на контуре многоугольника ближайшая к `anchorPosition`;
- `distance` - расстояние от `anchorPosition` до `position`;
- `closestPointIndex` - индекс вершины многоугольника, ближайшей к `position`;
- `nextPointIndex` - индекс вершины многоугольника, следующей за `position`;
- `prevPointIndex` - индекс вершины многоугольника, предшествующей `position`;
- `pathIndex` - индекс контура многоугольника, к которому относится найденная точка.



Поля `nextPointIndex`, `prevPointIndex` могут отсутствовать, если `position` совпадает с одной из вершин многоугольника.

#### Параметры:

| Параметр                      | Значение по умолчанию | Описание                                                                                                                  |
|-------------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------|
| <code>anchorPosition</code> * | —                     | Тип: <code>Number[]</code><br><br>Координаты точки, для которой рассчитывается ближайшая точка на контуре многоугольника. |

\* Обязательный параметр/опция.

#### `getCoordinates`

```
{Number[][][]} getCoordinates()
```

**Возвращает** координаты многоугольника.

#### `getFillRule`

```
{String} getFillRule()
```

**Возвращает** строковый идентификатор, определяющий алгоритм заливки многоугольника. Идентификатор может принимать одно из двух значений:

- `evenOdd` - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и подсчета количества сегментов контура в пределах заданной фигуры, которые пересекает этот луч. Если это число нечетное, точка находится внутри; если четное, точка находится снаружи.
- `nonZero` - алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки до бесконечности в любом направлении и проверки точек, в которых сегмент фигуры пересекает этот луч. Начиная с нуля, добавляется единица каждый раз, когда сегмент пересекает луч слева направо, и вычитается единица каждый раз, когда сегмент пересекает луч справа налево. Если после подсчета пересечений результат равен нулю, точка находится снаружи контура. В противном случае она находится внутри.

#### `getLength`

```
{Integer} getLength()
```

**Возвращает** количество контуров в многоугольнике.

## IPixelRectangleGeometry

Расширяет [IPixelGeometry](#).

Интерфейс пиксельной геометрии "Прямоугольник".

[Конструктор](#) | [Поля](#) | [Методы](#)

#### Конструктор

```
IPixelRectangleGeometry()
```

## Поля

| Имя                    | Тип                           | Описание                                                             |
|------------------------|-------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

## Методы

| Имя                                        | Возвращает                     | Описание                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">equals(geometry)</a>           | Boolean                        | Возвращает true, если переданная геометрия эквивалентна данной.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                   |
| <a href="#">getBounds()</a>                | Number[][] null                | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.<br><br>Унаследован от <a href="#">IBaseGeometry</a> . |
| <a href="#">getClosest(anchorPosition)</a> | Object                         | Ищет на прямоугольнике точку, ближайшую к anchorPosition.                                                                                                                                                                                                                                                |
| <a href="#">getCoordinates()</a>           | Number[][]                     | Возвращает координаты двух противоположных углов прямоугольника.                                                                                                                                                                                                                                         |
| <a href="#">getMetaData()</a>              | Object                         | Возвращает метаданные пиксельной геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |
| <a href="#">getType()</a>                  | String                         | Возвращает идентификатор типа геометрии.<br><br>Унаследован от <a href="#">IBaseGeometry</a> .                                                                                                                                                                                                           |
| <a href="#">scale(factor)</a>              | <a href="#">IPixelGeometry</a> | Создает отмасштабированную копию геометрии.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                                       |
| <a href="#">shift(offset)</a>              | <a href="#">IPixelGeometry</a> | Создает копию геометрии сдвинутую на заданную величину.<br><br>Унаследован от <a href="#">IPixelGeometry</a> .                                                                                                                                                                                           |

## Описание методов

### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на прямоугольнике точку, ближайшую к anchorPosition.

**Возвращает** объект со следующими полями:

- position - точка на прямоугольнике, ближайшая к anchorPosition;
- distance - расстояние от anchorPosition до position;
- closestPointIndex - индекс вершины прямоугольника, ближайшей к position;
- nextPointIndex - индекс вершины прямоугольника, следующей за position;
- prevPointIndex - индекс вершины прямоугольника, предшествующей position;
- pathIndex - индекс контура прямоугольника, к которому относится найденная точка.

Поля nextPointIndex, prevPointIndex могут отсутствовать, если position совпадает с одной из вершин прямоугольника.

### Параметры:

| Параметр                         | Значение по умолчанию | Описание                                                                                             |
|----------------------------------|-----------------------|------------------------------------------------------------------------------------------------------|
| <a href="#">anchorPosition</a> * | —                     | Тип: Number[]<br><br>Координаты точки, для которой рассчитывается ближайшая точка на прямоугольнике. |

\* Обязательный параметр/опция.

### getCoordinates

```
{Number[][]} getCoordinates()
```

**Возвращает** координаты двух противоположных углов прямоугольника.

## IPointGeometry

Расширяет [IGeometry](#), [IPointGeometryAccess](#).

Интерфейс геометрии "Точка".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IPointGeometry()
```

### Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

## События

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a>              | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>oldCoordinates</code> — старые координаты;</li> <li><code>newCoordinates</code> - новые координаты;</li> </ul> Унаследовано от <a href="#">IPointGeometryAccess</a> . |
| <a href="#">mapchange</a>           | Сменилась карта. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>oldMap</code> - старая карта;</li> <li><code>newMap</code> - новая карта.</li> </ul> Унаследовано от <a href="#">IGeometry</a> .                                          |
| <a href="#">optionschange</a>       | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                                                                                                                                                  |
| <a href="#">pixelgeometrychange</a> | Изменилась пиксельная геометрия. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>pixelGeometry</code> - новая пиксельная геометрия <a href="#">IPixelGeometry</a>.</li> </ul> Унаследовано от <a href="#">IGeometry</a> .                  |

## Методы

| Имя                              | Возвращает                    | Описание                                                                                                                                                                                                                                   |
|----------------------------------|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getBounds()</a>      | <code>Number[][]</code>  null | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.<br>Унаследован от <a href="#">IGeometry</a> . |
| <a href="#">getCoordinates()</a> | <code>Number[][]</code>  null | Возвращает координаты точки.<br>Унаследован от <a href="#">IPointGeometryAccess</a> .                                                                                                                                                      |
| <a href="#">getMap()</a>         | <a href="#">Map</a>  null     | Возвращает текущую карту.<br>Унаследован от <a href="#">IGeometry</a> .                                                                                                                                                                    |

| Имя                                         | Возвращает                           | Описание                                                                                                                                      |
|---------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getPixelGeometry([options])</a> | <a href="#">IPixelGeometry</a>       | Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.<br>Унаследован от <a href="#">IGeometry</a> . |
| <a href="#">getType()</a>                   | String                               | Возвращает строку "Point".                                                                                                                    |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IPointGeometryAccess</a> | Задает координаты точки.<br>Унаследован от <a href="#">IPointGeometryAccess</a> .                                                             |
| <a href="#">setMap(map)</a>                 |                                      | Задает карту.<br>Унаследован от <a href="#">IGeometry</a> .                                                                                   |

### Описание методов

#### getType

```
{String} getType()
```

Возвращает строку "Point".

## IPointGeometryAccess

Интерфейс доступа к геометрии "Точка".

[Конструктор](#) | [События](#) | [Методы](#)

### Конструктор

```
IPointGeometryAccess()
```

### События

| Имя                    | Описание                                                                                                                                                                                                                                                    |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li>oldCoordinates — старые координаты;</li><li>newCoordinates - новые координаты;</li></ul> |

### Методы

| Имя                                         | Возвращает                           | Описание                     |
|---------------------------------------------|--------------------------------------|------------------------------|
| <a href="#">getCoordinates()</a>            | Number[] null                        | Возвращает координаты точки. |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IPointGeometryAccess</a> | Задает координаты точки.     |

## Описание событий

### change

Изменение координат. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldCoordinates` — старые координаты;
- `newCoordinates` - новые координаты;

## Описание методов

### getCoordinates

```
{Number[]|null} getCoordinates()
```

**Возвращает** координаты точки.

### setCoordinates

```
{IPointGeometryAccess} setCoordinates(coordinates)
```

Задаёт координаты точки.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                                             |
|----------------------------|-----------------------|------------------------------------------------------|
| <code>coordinates</code> * | —                     | Тип: <code>Number[] null</code><br>Координаты точки. |

\* Обязательный параметр/опция.

## IPolygonGeometry

Расширяет [IGeometry](#), [IPolygonGeometryAccess](#).

Интерфейс геометрии "Многоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IPolygonGeometry()
```

### Поля

| Имя                  | Тип                            | Описание                                                             |
|----------------------|--------------------------------|----------------------------------------------------------------------|
| <code>events</code>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <code>options</code> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

## События

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a>              | <p>Изменение координат. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li>oldCoordinates — старые координаты;</li> <li>newCoordinates - новые координаты;</li> <li>oldFillRule - старый алгоритм заливки;</li> <li>newFillRule - новые алгоритм заливки.</li> </ul> <p>Унаследовано от <a href="#">IPolygonGeometryAccess</a>.</p> |
| <a href="#">mapchange</a>           | <p>Сменилась карта. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li>oldMap - старая карта;</li> <li>newMap - новая карта.</li> </ul> <p>Унаследовано от <a href="#">IGeometry</a>.</p>                                                                                                                                           |
| <a href="#">optionschange</a>       | <p>Изменение в опциях объекта.</p> <p>Унаследовано от <a href="#">ICustomizable</a>.</p>                                                                                                                                                                                                                                                                                                                                          |
| <a href="#">pixelgeometrychange</a> | <p>Изменилась пиксельная геометрия. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li>pixelGeometry - новая пиксельная геометрия <a href="#">IPixelGeometry</a>.</li> </ul> <p>Унаследовано от <a href="#">IGeometry</a>.</p>                                                                                                      |

## Методы

| Имя                                | Возвращает                 | Описание                                                                                                                         |
|------------------------------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a> | Boolean                    | <p>Проверяет, лежит ли переданная точка внутри многоугольника.</p> <p>Унаследован от <a href="#">IPolygonGeometryAccess</a>.</p> |
| <a href="#">freeze()</a>           | <a href="#">IFreezable</a> | <p>Переводит объект в "замороженный" режим.</p> <p>Унаследован от <a href="#">IFreezable</a>.</p>                                |
| <a href="#">get(index)</a>         | Number[][]                 | <p>Возвращает координаты контура с заданным индексом.</p> <p>Унаследован от <a href="#">IPolygonGeometryAccess</a>.</p>          |

| Имя                                         | Возвращает                                | Описание                                                                                                                                                                                                                                   |
|---------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getBounds()</a>                 | Number[][] null                           | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.<br>Унаследован от <a href="#">IGeometry</a> . |
| <a href="#">getChildGeometry(index)</a>     | <a href="#">ILinearRingGeometryAccess</a> | Создает и возвращает объект <a href="#">ILinearRingGeometryAccess</a> для заданного контура.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                    |
| <a href="#">getClosest(anchorPosition)</a>  | Object                                    | Ищет на контуре многоугольника точку, ближайшую к anchorPosition.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                               |
| <a href="#">getCoordinates()</a>            | Number[][][]                              | Возвращает массив координат геометрии.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                          |
| <a href="#">getFillRule()</a>               | String                                    | Возвращает идентификатор алгоритма заливки.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                     |
| <a href="#">getLength()</a>                 | Integer                                   | Возвращает количество контуров в геометрии.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                     |
| <a href="#">getMap()</a>                    | <a href="#">Map</a>  null                 | Возвращает текущую карту.<br>Унаследован от <a href="#">IGeometry</a> .                                                                                                                                                                    |
| <a href="#">getPixelGeometry([options])</a> | <a href="#">IPixelGeometry</a>            | Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.<br>Унаследован от <a href="#">IGeometry</a> .                                                                                              |
| <a href="#">getType()</a>                   | String                                    | Возвращает строку "Polygon".                                                                                                                                                                                                               |
| <a href="#">insert(index, path)</a>         | <a href="#">IPolygonGeometryAccess</a>    | Добавляет новый контур с заданным индексом.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                     |



| Имя                                         | Возвращает                                | Описание                                                                                                                                                                                                                                                                                    |
|---------------------------------------------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">isFrozen()</a>                  | Boolean                                   | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                 |
| <a href="#">remove(index)</a>               | <a href="#">ILinearRingGeometryAccess</a> | Удаляет контур с заданным индексом.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                              |
| <a href="#">set(index, path)</a>            | <a href="#">IPolygonGeometryAccess</a>    | Задаёт координаты контура с заданным индексом.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                   |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IPolygonGeometryAccess</a>    | Задаёт массив координат геометрии.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                               |
| <a href="#">setFillRule(fillRule)</a>       | <a href="#">IPolygonGeometryAccess</a>    | Задаёт алгоритм заливки многоугольника.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> .                                                                                                                                                                                          |
| <a href="#">setMap(map)</a>                 |                                           | Задаёт карту.<br>Унаследован от <a href="#">IGeometry</a> .                                                                                                                                                                                                                                 |
| <a href="#">splice(index, number)</a>       | <a href="#">ILinearRingGeometryAccess</a> | Удаляет определенное число контуров начиная с заданного индекса. При этом на место удаляемых контуров могут быть добавлены новые. Координаты новых контуров можно передавать дополнительными аргументами после параметра number.<br>Унаследован от <a href="#">IPolygonGeometryAccess</a> . |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>                | Переводит объект в активный режим.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                                                                           |

**Описание методов****getType**

```
{String} getType()
```

Возвращает строку "Polygon".

**IPolygonGeometryAccess**

Расширяет [IFreezable](#).

Интерфейс доступа к геометрии "Многоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
IPolygonGeometryAccess()
```

## Поля

| Имя                    | Тип                           | Описание                                                                  |
|------------------------|-------------------------------|---------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий объекта.<br>Унаследовано от <a href="#">IFreezable</a> . |

## События

| Имя                    | Описание                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> — старые координаты;</li><li><code>newCoordinates</code> - новые координаты;</li><li><code>oldFillRule</code> - старый алгоритм заливки;</li><li><code>newFillRule</code> - новые алгоритм заливки.</li></ul> |

## Методы

| Имя                                        | Возвращает                                | Описание                                                                                     |
|--------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a>         | Boolean                                   | Проверяет, лежит ли переданная точка внутри многоугольника.                                  |
| <a href="#">freeze()</a>                   | <a href="#">IFreezable</a>                | Переводит объект в "замороженный" режим.<br>Унаследован от <a href="#">IFreezable</a> .      |
| <a href="#">get(index)</a>                 | Number[][]                                | Возвращает координаты контура с заданным индексом.                                           |
| <a href="#">getChildGeometry(index)</a>    | <a href="#">ILinearRingGeometryAccess</a> | Создает и возвращает объект <a href="#">ILinearRingGeometryAccess</a> для заданного контура. |
| <a href="#">getClosest(anchorPosition)</a> | Object                                    | Ищет на контуре многоугольника точку, ближайшую к <code>anchorPosition</code> .              |
| <a href="#">getCoordinates()</a>           | Number[][][]                              | Возвращает массив координат геометрии.                                                       |
| <a href="#">getFillRule()</a>              | String                                    | Возвращает идентификатор алгоритма заливки.                                                  |

| Имя                                         | Возвращает                                | Описание                                                                                                                                                                                                                         |
|---------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getLength()</a>                 | Integer                                   | Возвращает количество контуров в геометрии.                                                                                                                                                                                      |
| <a href="#">insert(index, path)</a>         | <a href="#">IPolygonGeometryAccess</a>    | Добавляет новый контур с заданным индексом.                                                                                                                                                                                      |
| <a href="#">isFrozen()</a>                  | Boolean                                   | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                      |
| <a href="#">remove(index)</a>               | <a href="#">ILinearRingGeometryAccess</a> | Удаляет контур с заданным индексом.                                                                                                                                                                                              |
| <a href="#">set(index, path)</a>            | <a href="#">IPolygonGeometryAccess</a>    | Задаёт координаты контура с заданным индексом.                                                                                                                                                                                   |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IPolygonGeometryAccess</a>    | Задаёт массив координат геометрии.                                                                                                                                                                                               |
| <a href="#">setFillRule(fillRule)</a>       | <a href="#">IPolygonGeometryAccess</a>    | Задаёт алгоритм заливки многоугольника.                                                                                                                                                                                          |
| <a href="#">splice(index, number)</a>       | <a href="#">ILinearRingGeometryAccess</a> | Удаляет определенное число контуров начиная с заданного индекса. При этом на место удаляемых контуров могут быть добавлены новые. Координаты новых контуров можно передавать дополнительными аргументами после параметра number. |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>                | Переводит объект в активный режим.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                |

## Описание событий

### change

Изменение координат. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldCoordinates` — старые координаты;
- `newCoordinates` - новые координаты;
- `oldFillRule` - старый алгоритм заливки;
- `newFillRule` - новые алгоритм заливки.

## Описание методов

### contains

```
{Boolean} contains(position)
```

Проверяет, лежит ли переданная точка внутри многоугольника.

**Возвращает** признак принадлежности точки многоугольнику.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                            |
|-------------------------|-----------------------|-----------------------------------------------------|
| <code>position *</code> | —                     | Тип: <code>Number[]</code><br><br>Координаты точки. |

\* Обязательный параметр/опция.

#### Пример:

```
var myPolygon = new ymaps.geometry.Polygon([
 [[69, 45], [68, 55], [86, 56]]
]);

// Метод работает только с корректно заданной картой.
myPolygon.options.setParent(myMap.options);
myPolygon.geometry.setMap(myMap);

// Проверка, входит ли точка клика в полигон, с заданной выше геометрией.
myMap.events.add('click', function (e) {
 alert(myPolygon.geometry.contains(e.get('coords')) ? 'Ранил!' : 'Мимо!');});
```

#### get

```
{Number[][]} get(index)
```

Возвращает координаты контура с заданным индексом.

#### Параметры:

| Параметр             | Значение по умолчанию | Описание                                         |
|----------------------|-----------------------|--------------------------------------------------|
| <code>index *</code> | —                     | Тип: <code>Integer</code><br><br>Индекс контура. |

\* Обязательный параметр/опция.

#### getChildGeometry

```
{ILinearRingGeometryAccess} getChildGeometry(index)
```

Создает и возвращает объект [ILinearRingGeometryAccess](#) для заданного контура.

Возвращает объект геометрии, соответствующий заданному контуру.

#### Параметры:

| Параметр             | Значение по умолчанию | Описание                                         |
|----------------------|-----------------------|--------------------------------------------------|
| <code>index *</code> | —                     | Тип: <code>Integer</code><br><br>Индекс контура. |

\* Обязательный параметр/опция.

#### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на контуре многоугольника точку, ближайшую к `anchorPosition`.

Возвращает объект со следующими полями:

- `position` - точка на контуре многоугольника ближайшая к `anchorPosition`;

- distance - расстояние от anchorPosition до position;
- closestPointIndex - индекс вершины многоугольника, ближайшей к position;
- nextPointIndex - индекс вершины многоугольника, следующей за position;
- prevPointIndex - индекс вершины многоугольника, предшествующей position;
- pathIndex - индекс контура многоугольника, к которому относится найденная точка.

Поля nextPointIndex, prevPointIndex могут отсутствовать, если position совпадает с одной из вершин многоугольника.

#### Параметры:

| Параметр                      | Значение по умолчанию | Описание                                                                                                     |
|-------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------------|
| <code>anchorPosition</code> * | —                     | Тип: Number[]<br><br>Координаты точки, для которой рассчитывается ближайшая точка на контуре многоугольника. |

\* Обязательный параметр/опция.

#### Пример:

```
var myPolygon = new ymaps.Polygon([
 [[69, 45], [68, 55], [86, 56], [87, 43]],
 [[60, 48], [62, 60], [80, 62], [80, 48]]
]);
var myPoint = new ymaps.Placemark([74, 52]);
// Создаем кратчайшую нормаль от контура полигона до точки.
var normalVec = new ymaps.Polyline([
 myPolygon.geometry.getClosest(myPoint.geometry.getCoordinates()).position,
 myPoint.geometry.getCoordinates()
]);

myMap.geoObjects
 .add(myPolygon)
 .add(myPoint)
 .add(normalVec);
```

#### getCoordinates

```
{Number[][][]} getCoordinates()
```

Возвращает массив координат геометрии.

#### getFillRule

```
{String} getFillRule()
```

Возвращает идентификатор алгоритма заливки.

#### Пример:

```
var myPolygon = new ymaps.Polygon([
 [[69, 45], [68, 55], [86, 56], [87, 43]],
 [[60, 48], [62, 60], [80, 62], [80, 48]]
]),
middlePoint = [74, 52];

myMap.geoObjects.add(myPolygon);

// Проверка точки, приходящейся на пересечение двух контуров,
// по умолчанию (значение fillRule == 'evenOdd') образует вырез (дырку).
alert(myPolygon.geometry.contains(middlePoint)); // => false

// После задания значения 'nonZero', все пересечения также "закрашиваются",
// поэтому теперь точка входит в полигон.
myPolygon.geometry.setFillRule('nonZero');
alert(myPolygon.geometry.contains(middlePoint)); // => true
```

**getLength**

```
{Integer} getLength()
```

**Возвращает** количество контуров в геометрии.

**insert**

```
{IPolygonGeometryAccess} insert(index, path)
```

Добавляет новый контур с заданным индексом.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                               |
|----------------------|-----------------------|----------------------------------------|
| <code>index *</code> | —                     | Тип: Integer<br>Индекс контура.        |
| <code>path *</code>  | —                     | Тип: Number[][]<br>Координаты контура. |

\* Обязательный параметр/опция.

**remove**

```
{ILinearRingGeometryAccess} remove(index)
```

Удаляет контур с заданным индексом.

**Возвращает** удаленный контур.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                        |
|----------------------|-----------------------|---------------------------------|
| <code>index *</code> | —                     | Тип: Integer<br>Индекс контура. |

\* Обязательный параметр/опция.

**set**

```
{IPolygonGeometryAccess} set(index, path)
```

Задаёт координаты контура с заданным индексом.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр             | Значение по умолчанию | Описание                        |
|----------------------|-----------------------|---------------------------------|
| <code>index *</code> | —                     | Тип: Integer<br>Индекс контура. |

| Параметр            | Значение по умолчанию | Описание                                                |
|---------------------|-----------------------|---------------------------------------------------------|
| <code>path</code> * | —                     | Тип: <code>Number[][]</code><br><br>Координаты контура. |

\* Обязательный параметр/опция.

### setCoordinates

```
{IPolygonGeometryAccess} setCoordinates(coordinates)
```

Задаёт массив координат геометрии.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                   | Значение по умолчанию | Описание                                                    |
|----------------------------|-----------------------|-------------------------------------------------------------|
| <code>coordinates</code> * | —                     | Тип: <code>Number[][][]</code><br><br>Координаты геометрии. |

\* Обязательный параметр/опция.

### setFillRule

```
{IPolygonGeometryAccess} setFillRule(fillRule)
```

Задаёт алгоритм заливки многоугольника.

**Возвращает** ссылку на себя.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                                         |
|-------------------------|-----------------------|------------------------------------------------------------------|
| <code>fillRule</code> * | —                     | Тип: <code>String</code><br><br>Идентификатор алгоритма заливки. |

\* Обязательный параметр/опция.

### splice

```
{ILinearRingGeometryAccess[]} splice(index, number)
```

Удаляет определенное число контуров начиная с заданного индекса. При этом на место удаляемых контуров могут быть добавлены новые. Координаты новых контуров можно передавать дополнительными аргументами после параметра `number`.

**Возвращает** удаленные контуры.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                                                    |
|-----------------------|-----------------------|---------------------------------------------------------------------------------------------|
| <code>index *</code>  | —                     | Тип: Integer<br><br>Индекс, начиная с которого производится удаление и добавление контуров. |
| <code>number *</code> | —                     | Тип: Integer<br><br>Количество удаляемых контуров.                                          |

\* Обязательный параметр/опция.

## IPopup

Расширяет [ICustomizable](#), [IEventEmitter](#).

Интерфейс инфо-объекта.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IPopup()
```

### Поля

| Имя                  | Тип                            | Описание                                                             |
|----------------------|--------------------------------|----------------------------------------------------------------------|
| <code>events</code>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <code>options</code> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

### События

| Имя                        | Описание                                                                       |
|----------------------------|--------------------------------------------------------------------------------|
| <code>close</code>         | Заккрытие инфо-объекта.                                                        |
| <code>open</code>          | Открытие инфо-объекта.                                                         |
| <code>optionschange</code> | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> . |

### Методы

| Имя                         | Возвращает                  | Описание                                    |
|-----------------------------|-----------------------------|---------------------------------------------|
| <code>close([force])</code> | <a href="#">vow.Promise</a> | Закрывает инфо-объект.                      |
| <code>getData()</code>      |                             | Возвращает данные инфо-объекта.             |
| <code>getOverlay()</code>   | <a href="#">vow.Promise</a> | Возвращает объект-обещание вернуть оверлей. |



| Имя                                   | Возвращает               | Описание                                                                                                                                                                                                         |
|---------------------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>getOverlaySync()</code>         | <code>IOverlay</code>    | Возвращает оверлей, если тот существует.                                                                                                                                                                         |
| <code>getPosition()</code>            |                          | Возвращает координаты инфо-объекта.                                                                                                                                                                              |
| <code>isOpen()</code>                 | Boolean                  | Возвращает состояние инфо-объекта: открыт/закрыт.                                                                                                                                                                |
| <code>open([position[, data]])</code> | <code>vow.Promise</code> | Открывает инфо-объект в указанной позиции. Если инфо-объект уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией <code>IProjection</code> , которая содержится в опциях. |
| <code>setData(data)</code>            | <code>vow.Promise</code> | Задаёт инфо-объекту новые данные.                                                                                                                                                                                |
| <code>setPosition(position)</code>    | <code>vow.Promise</code> | Задаёт инфо-объекту новую позицию.                                                                                                                                                                               |

### Описание событий

#### close

Заккрытие инфо-объекта.

#### open

Открытие инфо-объекта.

### Описание методов

#### close

```
{vow.Promise} close([force])
```

Закрывает инфо-объект.

**Возвращает** объект-promise.

**Параметры:**

| Параметр           | Значение по умолчанию | Описание                                 |
|--------------------|-----------------------|------------------------------------------|
| <code>force</code> | false                 | Тип: Boolean<br><br>Мгновенное закрытие. |

#### getData

```
{ } getData()
```

**Возвращает** данные инфо-объекта.

**getOverlay**

```
{vow.Promise} getOverlay()
```

**Возвращает** объект-обещание вернуть оверлей.

**getOverlaySync**

```
{IOverlay} getOverlaySync()
```

**Возвращает** оверлей, если тот существует.

**getPosition**

```
{ } getPosition()
```

**Возвращает** координаты инфо-объекта.

**isOpen**

```
{Boolean} isOpen()
```

**Возвращает** состояние инфо-объекта: открыт/закрыт.

**open**

```
{vow.Promise} open([position[, data]])
```

Открывает инфо-объект в указанной позиции. Если инфо-объект уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией [IProjection](#), которая содержится в опциях.

**Возвращает** объект-promise.

**Параметры:**

| Параметр                 | Значение по умолчанию | Описание                                                      |
|--------------------------|-----------------------|---------------------------------------------------------------|
| <a href="#">position</a> | —                     | Тип: Number[]<br><br>Точка, где необходимо расположить балун. |
| <a href="#">data</a>     | —                     | Тип: Object String HTMLElement<br><br>Данные оверлея.         |

**setData**

```
{vow.Promise} setData(data)
```

Задаёт инфо-объекту новые данные.

**Возвращает** объект-promise.

**Параметры:**

| Параметр               | Значение по умолчанию | Описание                                                   |
|------------------------|-----------------------|------------------------------------------------------------|
| <a href="#">data</a> * | —                     | Тип: Object String HTMLElement<br><br>Данные инфо-объекта. |

\* Обязательный параметр/опция.

### setPosition

```
{vow.Promise} setPosition(position)
```

Задаёт инфо-объекту новую позицию.

Возвращает объект-promise.

Параметры:

| Параметр                | Значение по умолчанию | Описание                                  |
|-------------------------|-----------------------|-------------------------------------------|
| <code>position</code> * | —                     | Тип: Number[]<br>Координаты инфо-объекта. |

\* Обязательный параметр/опция.

## IPopupManager

Расширяет [IEventEmitter](#).

Интерфейс менеджера инфо-объекта.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IPopupManager()
```

### Поля

| Имя                 | Тип                           | Описание                                                             |
|---------------------|-------------------------------|----------------------------------------------------------------------|
| <code>events</code> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

### События

| Имя                | Описание                                                                                                                                                                                                          |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>close</code> | Закрытие инфо-объекта. Названия полей доступны через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>target</code> - ссылка на объект, на котором был закрыт инфо-объект.</li></ul> |
| <code>open</code>  | Открытие инфо-объекта. Названия полей доступны через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>target</code> - ссылка на объект, на котором был открыт инфо-объект.</li></ul> |

### Методы

| Имя                         | Возвращает                  | Описание                                 |
|-----------------------------|-----------------------------|------------------------------------------|
| <code>close([force])</code> | <a href="#">vow.Promise</a> | Закрывает инфо-объект.                   |
| <code>destroy()</code>      |                             | Отключает менеджер инфо-объекта.         |
| <code>getData()</code>      | Object null                 | Возвращает данные инфо-объекта или null. |

| Имя                                             | Возвращает                         | Описание                                          |
|-------------------------------------------------|------------------------------------|---------------------------------------------------|
| <code>getOptions()</code>                       | <code>IOptionsManager</code>  null | Возвращает менеджер опций или null.               |
| <code>getOverlay()</code>                       | <code>vow.Promise</code>           | Возвращает объект-обещание вернуть оверлей.       |
| <code>getOverlaySync()</code>                   | <code>IOverlay</code>  null        | Возвращает оверлей, если тот существует.          |
| <code>getPosition()</code>                      | <code>Number[]</code>  null        | Возвращает координаты инфо-объекта или 'null'.    |
| <code>isOpen()</code>                           | <code>Boolean</code>               | Возвращает состояние инфо-объекта: открыт/закрыт. |
| <code>open([position[, data[, options]])</code> | <code>vow.Promise</code>           | Открывает инфо-объект в указанной позиции.        |
| <code>setData(data)</code>                      | <code>vow.Promise</code>           | Задаёт инфо-объекту новые данные.                 |
| <code>setOptions(options)</code>                | <code>vow.Promise</code>           | Задаёт инфо-объекту новые опции.                  |
| <code>setPosition(position)</code>              | <code>vow.Promise</code>           | Задаёт инфо-объекту новую позицию.                |

### Описание событий

#### close

Заккрытие инфо-объекта. Названия полей доступны через метод `Event.get`:

- `target` - ссылка на объект, на котором был закрыт инфо-объект.

#### open

Открытие инфо-объекта. Названия полей доступны через метод `Event.get`:

- `target` - ссылка на объект, на котором был открыт инфо-объект.

### Описание методов

#### close

```
{vow.Promise} close([force])
```

Закрывает инфо-объект.

**Возвращает** объект-promise.

**Параметры:**

| Параметр           | Значение по умолчанию | Описание                                 |
|--------------------|-----------------------|------------------------------------------|
| <code>force</code> | <code>false</code>    | Тип: Boolean<br><br>Мгновенное закрытие. |

**destroy**

```
{ } destroy()
```

Отключает менеджер инфо-объекта.

**getData**

```
{Object|null} getData()
```

**Возвращает** координаты инфо-объекта или null.

**getOptions**

```
{IOptionManager|null} getOptions()
```

**Возвращает** менеджер опций или `null`.

**getOverlay**

```
{vow.Promise} getOverlay()
```

**Возвращает** объект-обещание вернуть оверлей.

**getOverlaySync**

```
{IOverlay|null} getOverlaySync()
```

**Возвращает** оверлей, если тот существует.

**getPosition**

```
{Number[]|null} getPosition()
```

**Возвращает** координаты инфо-объекта или `null`.

**isOpen**

```
{Boolean} isOpen()
```

**Возвращает** состояние инфо-объекта: открыт/закрыт.

**open**

```
{vow.Promise} open([position[, data[, options]])
```

Открывает инфо-объект в указанной позиции.

**Возвращает** объект-promise.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                          |
|-----------------------|-----------------------|---------------------------------------------------|
| <code>position</code> | —                     | Тип: Number[]<br><br>Координаты открытия.         |
| <code>data</code>     | —                     | Тип: Object string <br>HTMLElement<br><br>Данные. |

| Параметр             | Значение по умолчанию | Описание              |
|----------------------|-----------------------|-----------------------|
| <code>options</code> | —                     | Тип: Object<br>Опции. |

### setData

```
{vow.Promise} setData(data)
```

Задаёт инфо-объекту новые данные.

**Возвращает** объект-promise.

**Параметры:**

| Параметр            | Значение по умолчанию | Описание                                                   |
|---------------------|-----------------------|------------------------------------------------------------|
| <code>data</code> * | —                     | Тип: Object String <br>HTMLElement<br>Данные инфо-объекта. |

\* Обязательный параметр/опция.

### setOptions

```
{vow.Promise} setOptions(options)
```

Задаёт инфо-объекту новые опции.

**Возвращает** объект-promise.

**Параметры:**

| Параметр               | Значение по умолчанию | Описание                           |
|------------------------|-----------------------|------------------------------------|
| <code>options</code> * | —                     | Тип: Object<br>Опции инфо-объекта. |

\* Обязательный параметр/опция.

### setPosition

```
{vow.Promise} setPosition(position)
```

Задаёт инфо-объекту новую позицию.

**Возвращает** объект-promise.

**Параметры:**

| Параметр                | Значение по умолчанию | Описание                                  |
|-------------------------|-----------------------|-------------------------------------------|
| <code>position</code> * | —                     | Тип: Number[]<br>Координаты инфо-объекта. |

\* Обязательный параметр/опция.

## IPositioningContext

Интерфейс контекста позиционирования - объекта, позволяющего позиционировать внутри себя объект, заданный глобальными пиксельными координатами.

[Конструктор](#) | [Методы](#)

### Конструктор

```
IPositioningContext()
```

### Методы

| Имя                                                | Возвращает | Описание                                                                                  |
|----------------------------------------------------|------------|-------------------------------------------------------------------------------------------|
| <a href="#">fromClientPixels(clientPixelPoint)</a> | Number[]   | Преобразует клиентские пиксельные координаты в глобальные.                                |
| <a href="#">getZoom()</a>                          | Number     | Возвращает текущий уровень масштабирования, в котором работает контекст позиционирования. |
| <a href="#">toClientPixels(globalPixelPoint)</a>   | Number[]   | Преобразует глобальные пиксельные координаты в клиентские.                                |

### Описание методов

#### fromClientPixels

```
{Number[]} fromClientPixels(clientPixelPoint)
```

Преобразует клиентские пиксельные координаты в глобальные.

**Возвращает** глобальные пиксельные координаты.

**Параметры:**

| Параметр                           | Значение по умолчанию | Описание                                               |
|------------------------------------|-----------------------|--------------------------------------------------------|
| <a href="#">clientPixelPoint</a> * | —                     | Тип: Number[]<br><br>Клиентские пиксельные координаты. |

\* Обязательный параметр/опция.

#### getZoom

```
{Number} getZoom()
```

**Возвращает** текущий уровень масштабирования, в котором работает контекст позиционирования.

#### toClientPixels

```
{Number[]} toClientPixels(globalPixelPoint)
```

Преобразует глобальные пиксельные координаты в клиентские.

**Возвращает** клиентские пиксельные координаты.

**Параметры:**

| Параметр                           | Значение по умолчанию | Описание                                               |
|------------------------------------|-----------------------|--------------------------------------------------------|
| <a href="#">globalPixelPoint</a> * | —                     | Тип: Number[]<br><br>Глобальные пиксельные координаты. |

\* Обязательный параметр/опция.

**IProjection**

Проекция. Описывает, каким образом реальная карта проецируется на бесконечную пиксельную плоскость. Один "мир" должен иметь размеры 256x256 пикселей на нулевом масштабе, при этом левый верхний угол "мира" имеет координаты (0, 0), координатные оси направлены вправо и вниз. "Миры" могут быть склеенными по любой оси (либо по обеим осям сразу).

[Конструктор](#) | [Методы](#)

**Конструктор**

```
IProjection()
```

**Методы**

| Имя                                                      | Возвращает                   | Описание                                                                                     |
|----------------------------------------------------------|------------------------------|----------------------------------------------------------------------------------------------|
| <a href="#">fromGlobalPixels(globalPixelPoint, zoom)</a> | Number[]                     | Преобразует пиксельные координаты на указанном уровне масштабирования в координаты проекции. |
| <a href="#">getCoordSystem()</a>                         | <a href="#">ICoordSystem</a> | Возвращает координатную систему, используемую проекцией.                                     |
| <a href="#">isCycled()</a>                               | Boolean[]                    | Показатель за цикленности проекции.                                                          |
| <a href="#">toGlobalPixels(coordPoint, zoom)</a>         | Number[]                     | Преобразует координаты проекции в глобальные пиксельные на указанном уровне масштабирования. |

**Описание методов****fromGlobalPixels**

```
{Number[]} fromGlobalPixels(globalPixelPoint, zoom)
```

Преобразует пиксельные координаты на указанном уровне масштабирования в координаты проекции.

**Возвращает** точку в координатах проекции.

**Параметры:**



| Параметр                        | Значение по умолчанию | Описание                                             |
|---------------------------------|-----------------------|------------------------------------------------------|
| <code>globalPixelPoint</code> * | —                     | Тип: Number[]<br><br>Точка в пиксельных координатах. |
| <code>zoom</code> *             | —                     | Тип: Number<br><br>Уровень масштабирования.          |

\* Обязательный параметр/опция.

### **getCoordSystem**

```
{ICoordSystem} getCoordSystem()
```

**Возвращает** координатную систему, используемую проекцией.

### **isCycled**

```
{Boolean[]} isCycled()
```

Показатель заикленности проекции.

**Возвращает** пару флагов, показывающих, склеена ли карта по пиксельным осям (x/y).

### **toGlobalPixels**

```
{Number[]} toGlobalPixels(coordPoint, zoom)
```

Преобразует координаты проекции в глобальные пиксельные на указанном уровне масштабирования.

**Возвращает** пару пиксельных координат.

**Параметры:**

| Параметр                  | Значение по умолчанию | Описание                                           |
|---------------------------|-----------------------|----------------------------------------------------|
| <code>coordPoint</code> * | —                     | Тип: Number[]<br><br>Точка в координатах проекции. |
| <code>zoom</code> *       | —                     | Тип: Number<br><br>Уровень масштабирования.        |

\* Обязательный параметр/опция.

## **IPromiseProvider**

Объект, выполняющий "обещание".

[Конструктор](#) | [Методы](#)

### **Конструктор**

```
IPromiseProvider()
```

**Методы**

| Имя                                    | Возвращает                       | Описание                                            |
|----------------------------------------|----------------------------------|-----------------------------------------------------|
| <code>then(onResolve, onReject)</code> | <a href="#">IPromiseProvider</a> | Возвращает ссылку на себя или новый Promise объект. |

**Описание методов****then**

```
{IPromiseProvider} then(onResolve, onReject)
```

**Возвращает** ссылку на себя или новый Promise объект.

**Параметры:**

| Параметр                 | Значение по умолчанию | Описание                                                                                                    |
|--------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------|
| <code>onResolve *</code> | —                     | Тип: Function<br><br>Функция-обработчик, которая вызывается, если обещание выполнено.                       |
| <code>onReject *</code>  | —                     | Тип: Function<br><br>Функция-обработчик, которая вызывается, если обещание не выполнено (произошла ошибка). |

\* Обязательный параметр/опция.

**IRatioMap**

Интерфейс объекта, содержащего соотношение произвольных данных к `devicePixelRatio`. Используется в случаях, когда требуется поддержка экранов с отношением виртуальных пикселей к физическим более единицы. В качестве ключей в объекте должны использоваться строки, состоящие из целого или дробного числа, обозначающего коэффициент плотности пикселей.

[Конструктор](#)**Конструктор**

```
IRatioMap()
```

**Пример:**

```
// Допустим, нам нужно правильно отобразить изображение.
var images = {
 // Для обычных экранов мы будем отображать нормальное изображение.
 "1": "100x100.png",
 // Для HTC Desire, Samsung Galaxy S II и других устройств с devicePixelRatio = 1.5.
 "1.5": "150x150.png",
 // Для Apple устройств с ретина-экраном, а также, например, Sony Xperia S, HTC One X и других.
 // with the ratio = 2.
 "2": "200x100.png"
}
```

**IRectangleGeometry**

Расширяет [IGeometry](#), [IRectangleGeometryAccess](#).

Интерфейс геометрии "Прямоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

## Конструктор

```
IRectangleGeometry()
```

## Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

## События

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a>              | Изменение координат углов. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldCoordinates</code> - старые координаты углов;</li><li><code>newCoordinates</code> - новые координаты углов.</li></ul> Унаследовано от <a href="#">IRectangleGeometryAccess</a> . |
| <a href="#">mapchange</a>           | Сменилась карта. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>oldMap</code> - старая карта;</li><li><code>newMap</code> - новая карта.</li></ul> Унаследовано от <a href="#">IGeometry</a> .                                                                |
| <a href="#">optionschange</a>       | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> .                                                                                                                                                                                                                                                                                     |
| <a href="#">pixelgeometrychange</a> | Изменилась пиксельная геометрия. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li><code>pixelGeometry</code> - новая пиксельная геометрия <a href="#">IPixelGeometry</a>.</li></ul> Унаследовано от <a href="#">IGeometry</a> .                                       |

## Методы

| Имя                                | Возвращает | Описание                                                                                                                 |
|------------------------------------|------------|--------------------------------------------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a> | Boolean    | Проверяет, лежит ли переданная точка внутри прямоугольника.<br>Унаследован от <a href="#">IRectangleGeometryAccess</a> . |

| Имя                                         | Возвращает                               | Описание                                                                                                                                                                                                                                   |
|---------------------------------------------|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">freeze()</a>                    | <a href="#">IFreezable</a>               | Переводит объект в "замороженный" режим.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                    |
| <a href="#">getBounds()</a>                 | Number[][] null                          | Возвращает координаты двух противоположных углов области, охватывающей геометрию. Первый элемент массива - юго-западный угол области; второй элемент - северо-восточный угол соответственно.<br>Унаследован от <a href="#">IGeometry</a> . |
| <a href="#">getClosest(anchorPosition)</a>  | Object                                   | Ищет на контуре прямоугольника точку, ближайшую к anchorPosition.<br>Унаследован от <a href="#">IRectangleGeometryAccess</a> .                                                                                                             |
| <a href="#">getCoordinates()</a>            | Number[][]                               | Возвращает координаты двух противоположных углов прямоугольника.<br>Унаследован от <a href="#">IRectangleGeometryAccess</a> .                                                                                                              |
| <a href="#">getMap()</a>                    | <a href="#">Map</a>  null                | Возвращает текущую карту.<br>Унаследован от <a href="#">IGeometry</a> .                                                                                                                                                                    |
| <a href="#">getPixelGeometry([options])</a> | <a href="#">IPixelGeometry</a>           | Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.<br>Унаследован от <a href="#">IGeometry</a> .                                                                                              |
| <a href="#">getType()</a>                   | String                                   | Возвращает строку "Rectangle".                                                                                                                                                                                                             |
| <a href="#">isFrozen()</a>                  | Boolean                                  | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IRectangleGeometryAccess</a> | Задает координаты двух противоположных углов прямоугольника.<br>Унаследован от <a href="#">IRectangleGeometryAccess</a> .                                                                                                                  |
| <a href="#">setMap(map)</a>                 |                                          | Задает карту.<br>Унаследован от <a href="#">IGeometry</a> .                                                                                                                                                                                |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>               | Переводит объект в активный режим.<br>Унаследован от <a href="#">IFreezable</a> .                                                                                                                                                          |

## Описание методов

### getType

```
{String} getType()
```

Возвращает строку "Rectangle".

## IRectangleGeometryAccess

Расширяет [IFreezable](#).

Интерфейс доступа к геометрии "Прямоугольник".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IRectangleGeometryAccess()
```

### Поля

| Имя                    | Тип                           | Описание                                                                  |
|------------------------|-------------------------------|---------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий объекта.<br>Унаследовано от <a href="#">IFreezable</a> . |

### События

| Имя                    | Описание                                                                                                                                                                                                                                                                      |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">change</a> | Изменение координат углов. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"><li>oldCoordinates - старые координаты углов;</li><li>newCoordinates - новые координаты углов.</li></ul> |

### Методы

| Имя                                        | Возвращает                 | Описание                                                                                |
|--------------------------------------------|----------------------------|-----------------------------------------------------------------------------------------|
| <a href="#">contains(position)</a>         | Boolean                    | Проверяет, лежит ли переданная точка внутри прямоугольника.                             |
| <a href="#">freeze()</a>                   | <a href="#">IFreezable</a> | Переводит объект в "замороженный" режим.<br>Унаследован от <a href="#">IFreezable</a> . |
| <a href="#">getClosest(anchorPosition)</a> | Object                     | Ищет на контуре прямоугольника точку, ближайшую к anchorPosition.                       |
| <a href="#">getCoordinates()</a>           | Number[][]                 | Возвращает координаты двух противоположных углов прямоугольника.                        |

| Имя                                         | Возвращает                               | Описание                                                                                                                        |
|---------------------------------------------|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">isFrozen()</a>                  | Boolean                                  | Возвращает true, если объект находится в "замороженном" режиме, иначе false.<br><br>Унаследован от <a href="#">IFreezable</a> . |
| <a href="#">setCoordinates(coordinates)</a> | <a href="#">IRectangleGeometryAccess</a> | Задаёт координаты двух противоположных углов прямоугольника.                                                                    |
| <a href="#">unfreeze()</a>                  | <a href="#">IFreezable</a>               | Переводит объект в активный режим.<br><br>Унаследован от <a href="#">IFreezable</a> .                                           |

### Описание событий

#### change

Изменение координат углов. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `oldCoordinates` - старые координаты углов;
- `newCoordinates` - новые координаты углов.

### Описание методов

#### contains

```
{Boolean} contains(position)
```

Проверяет, лежит ли переданная точка внутри прямоугольника.

**Возвращает** признак принадлежности точки прямоугольнику.

#### Параметры:

| Параметр                   | Значение по умолчанию | Описание                                            |
|----------------------------|-----------------------|-----------------------------------------------------|
| <a href="#">position</a> * | —                     | Тип: <code>Number[]</code><br><br>Координаты точки. |

\* Обязательный параметр/опция.

#### getClosest

```
{Object} getClosest(anchorPosition)
```

Ищет на контуре прямоугольника точку, ближайшую к `anchorPosition`.

**Возвращает** объект со следующими полями:

- `position` - точка на контуре прямоугольника, ближайшая к `anchorPosition`;
- `distance` - расстояние от `anchorPosition` до `position`;

#### Параметры:

| Параметр                      | Значение по умолчанию | Описание                                                                                                          |
|-------------------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------|
| <code>anchorPosition</code> * | —                     | Тип: <code>Number[]</code><br><br>Координаты точки, для которой рассчитывается ближайшая точка на прямоугольнике. |

\* Обязательный параметр/опция.

### **getCoordinates**

```
{Number[][]} getCoordinates()
```

**Возвращает** координаты двух противоположных углов прямоугольника.

### **setCoordinates**

```
{IRectangleGeometryAccess} setCoordinates(coordinates)
```

Задаёт координаты двух противоположных углов прямоугольника.

**Возвращает** ссылку на себя.

#### **Параметры:**

| Параметр                   | Значение по умолчанию | Описание                                              |
|----------------------------|-----------------------|-------------------------------------------------------|
| <code>coordinates</code> * | —                     | Тип: <code>Number[][]</code><br><br>Координаты углов. |

\* Обязательный параметр/опция.

## **IRoutePanel**

Расширяет [IEventEmitter](#).

Интерфейс панели маршрута.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### **Конструктор**

```
IRoutePanel()
```

### **Поля**

| Имя                 | Тип                        | Описание                                                             |
|---------------------|----------------------------|----------------------------------------------------------------------|
| <code>events</code> | <code>IEventManager</code> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

| Имя                     | Тип                            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">options</a> | <a href="#">IOptionManager</a> | <p>Менеджер опций. Названия опций:</p> <ul style="list-style-type: none"> <li><code>allowSwitch</code>: Boolean = true – показать на панели кнопку, позволяющую менять местами начальную и конечную точки маршрута;</li> <li><code>reverseGeocoding</code>: Boolean = true – включить ли обратное геокодирование при построение маршрута;</li> <li><code>adjustMapMargin</code>: Boolean = false – будет ли панель маршрута регистрировать свои размеры в <a href="#">map.margin.Manager</a>.</li> <li><code>types</code>: Object = { auto: true, masstransit: true, pedestrian: true, bicycle: true, taxi: false } – типы маршрутизации, которые будут доступны для выбора пользователю. При изменении опции возможных типов маршрута, текущий выбранный тип автоматически меняется. Например, если был выбран пешеходный маршрут, а после этого вебмастер разрешил только автомобильные и на общественном транспорте, то автоматически выберется автомобильный. Доступные типы маршрутизации отображаются в панели только в том случае, если задано два или более типа. Обратите внимание если вы используете политику безопасности контента: для использования типа "такси" нужно обновить правила CSP на <a href="#">новейшую версию</a>.</li> </ul> |
| <a href="#">state</a>   | <a href="#">IDataManager</a>   | <p>Менеджер состояний. Названия состояний:</p> <ul style="list-style-type: none"> <li><code>type</code>: String – тип маршрутизации <a href="#">IMultiRouteParams.routingMode</a>;</li> <li><code>fromEnabled</code>: Boolean – включить возможность задавать пункт отправления в поле ввода;</li> <li><code>from</code>: String – адрес или координаты пункта отправления;</li> <li><code>toEnabled</code>: Boolean – включить возможность задавать пункт назначения в поле ввода;</li> <li><code>to</code>: String – адрес или координаты пункта назначения.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |


## События

| Имя                     | Описание                                                                  |
|-------------------------|---------------------------------------------------------------------------|
| <a href="#">disable</a> | Зависимости панели не загружены.                                          |
| <a href="#">enable</a>  | Панель маршрутизации и ее зависимости загружены и готовы к использованию. |

## Методы

| Имя                      | Возвращает | Описание                          |
|--------------------------|------------|-----------------------------------|
| <a href="#">enable()</a> |            | Загружает все зависимости панели. |



| Имя                          | Возвращает                                              | Описание                                                                                                                                                                                                                       |
|------------------------------|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>geolocate(name)</code> | <code>vow.Promise</code>                                | Задать местоположение пользователя в качестве начальной или конечной точки маршрута.                                                                                                                                           |
| <code>getRoute()</code>      | <code>multiRouter.MultiRoute</code>                     |  <b>Внимание:</b><br>Данный метод является устаревшим. См. <a href="#">IRoutePanel.getRouteAsync</a> .<br><br>Возвращает построенный маршрут. |
| <code>getRouteAsync()</code> | <code>vow.Promise.&lt;multiRouter.MultiRoute&gt;</code> | Возвращает <code>vow.Promise</code> , который будет разрешен построенным маршрутом. Promise отклоняется при возникновении ошибки.                                                                                              |
| <code>isEnabled()</code>     | <code>Boolean</code>                                    | Возвращает <code>true</code> , если панель полностью загружена, <code>false</code> – иначе.                                                                                                                                    |
| <code>switchPoints()</code>  |                                                         | Меняет местами начальную и конечную точки маршрута. Изменения автоматически отобразятся в полях ввода на панели маршрута.                                                                                                      |

### Описание полей

#### options

```
{IOptionManager} options
```

Менеджер опций. Названия опций:

- `allowSwitch`: `Boolean` = `true` – показать на панели кнопку, позволяющую менять местами начальную и конечную точки маршрута;
- `reverseGeocoding`: `Boolean` = `true` – включить ли обратное геокодирование при построение маршрута;
- `adjustMapMargin`: `Boolean` = `false` – будет ли панель маршрута регистрировать свои размеры в [map.margin.Manager](#).
- `types`: `Object` = { `auto`: `true`, `masstransit`: `true`, `pedestrian`: `true`, `bicycle`: `true`, `taxi`: `false` } – типы маршрутизации, которые будут доступны для выбора пользователю. При изменении опции возможных типов маршрута, текущий выбранный тип автоматически меняется. Например, если был выбран пешеходный маршрут, а после этого вебмастер разрешил только автомобильные и на общественном транспорте, то автоматически выберется автомобильный. Доступные типы маршрутизации отображаются в панели только в том случае, если задано два или более типа. Обратите внимание если вы используете политику безопасности контента: для использования типа "такси" нужно обновить правила CSP на [новейшую версию](#).

#### state

```
{IDataManager} state
```

Менеджер состояний. Названия состояний:

- `type`: `String` – тип маршрутизации [IMultiRouteParams.routingMode](#);

- `fromEnabled`: Boolean – включить возможность задавать пункт отправления в поле ввода;
- `from`: String – адрес или координаты пункта отправления;
- `toEnabled`: Boolean – включить возможность задавать пункт назначения в поле ввода;
- `to`: String – адрес или координаты пункта назначения.

### Описание событий

#### **disable**

Зависимости панели не загружены.

#### **enable**

Панель маршрутизации и ее зависимости загружены и готовы к использованию.

### Описание методов

#### **enable**

```
{ } enable()
```

Загружает все зависимости панели.

#### **geolocate**

```
{vow.Promise} geolocate(name)
```

Задать местоположение пользователя в качестве начальной или конечной точки маршрута.

Возвращает объект-promise. см. [geolocation](#).

#### Параметры:

| Параметр            | Значение по умолчанию | Описание                                                                                                                                                        |
|---------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>name</code> * | —                     | Тип: String<br><br>В качестве какой точки маршрута (начальной или конечной) нужно использовать местоположение пользователя. Принимает значение 'from' или 'to'. |

\* Обязательный параметр/опция.

#### **getRoute**

```
{multiRouter.MultiRoute} getRoute()
```

Данный метод является устаревшим. See [IRoutePanel.getRouteAsync](#).

Возвращает построенный маршрут.

#### **getRouteAsync**

```
{vow.Promise.<multiRouter.MultiRoute>} getRouteAsync()
```

Возвращает [vow.Promise](#), который:

- будет **разрешен** построенным маршрутом (объектом [multiRouter.MultiRoute](#)),

- либо **отклонен** с ошибкой.

## isEnabled

```
{Boolean} isEnabled()
```

**Возвращает** true, если панель полностью загружена, false – иначе.

## switchPoints

```
{ } switchPoints()
```

Меняет местами начальную и конечную точки маршрута. Изменения автоматически отобразятся в полях ввода на панели маршрута.

## ISearchControlLayout

Расширяет [IExpandableControlLayout](#).

Интерфейс макета элемента управления "Поиск по карте".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
ISearchControlLayout()
```

### Поля

| Имя                    | Тип                           | Описание                                                                |
|------------------------|-------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

### События

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>           | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">collapse</a>        | Событие, инициирующее сворачивание объекта.<br>Унаследовано от <a href="#">IExpandableControlLayout</a> .                                                                                                                                                                                                                    |
| <a href="#">contextmenu</a>     | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">dblclick</a>        | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <a href="#">emptinesschange</a> | Изменение признака пустоты макета. Экземпляр класса <a href="#">Event</a> .<br>Унаследовано от <a href="#">ILayout</a> .                                                                                                                                                                                                     |

| Имя                            | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">expand</a>         | <p>Событие, инициирующее разворачивание объекта.</p> <p>Унаследовано от <a href="#">IExpandableControlLayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <a href="#">loadmore</a>       | <p>Событие, инициирующее дополнительную подгрузку результатов поиска. Экземпляр класса <a href="#">Event</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">mousedown</a>      | <p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">mouseenter</a>     | <p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <a href="#">mouseleave</a>     | <p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">mousemove</a>      | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">mouseup</a>        | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <a href="#">multitouchend</a>  | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchmove</a> | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchstart</a>     | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">parentelementchange</a> | <p>Изменение родительского элемента. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <a href="#">resultselect</a>        | <p>Событие, инициирующее показ результата поиска на карте. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>index</code> - номер объекта в ответе сервера, который нужно показать на карте.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <a href="#">search</a>              | <p>Событие, инициирующее поиск объектов. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>request</code> – строка, содержащая запрос.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <a href="#">shapechange</a>         | <p>Изменение фигуры области охватывающей макет. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">wheel</a>               | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Методы

| Имя                                | Возвращает  | Описание                                                                                                      |
|------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>          |             | <p>Деструктор. Вызывается при завершении работы с макетом.</p> <p>Унаследован от <a href="#">ILayout</a>.</p> |
| <a href="#">getData()</a>          | Object      | <p>Возвращает объект данных макета.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                        |
| <a href="#">getParentElement()</a> | HTMLElement | <p>Возвращает родительский HTML-элемент.</p> <p>Унаследован от <a href="#">ILayout</a>.</p>                   |

| Имя                                      | Возвращает                   | Описание                                                                                                                                                                                                                                                                  |
|------------------------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getShape()</a>               | <a href="#">IShape</a>  null | Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.<br>Унаследован от <a href="#">ILayout</a> . |
| <a href="#">isEmpty()</a>                | Boolean                      | Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.<br>Унаследован от <a href="#">ILayout</a> .                                                                        |
| <a href="#">setData(data)</a>            |                              | Задаёт данные макета.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                         |
| <a href="#">setParentElement(parent)</a> |                              | Добавляет макет в DOM-дерево.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                 |

## Описание событий

### loadmore

Событие, инициирующее дополнительную подгрузку результатов поиска. Экземпляр класса [Event](#).

### resultselect

Событие, инициирующее показ результата поиска на карте. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `index` - номер объекта в ответе сервера, который нужно показать на карте.

### search

Событие, инициирующее поиск объектов. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `request` – строка, содержащая запрос.

## ISearchProvider

Интерфейс провайдера поиска.

[Конструктор](#) | [Методы](#)

### Конструктор

```
ISearchProvider()
```

## Методы

| Имя                                      | Возвращает               | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>search(request[, options])</code>  | <code>vow.Promise</code> | Отправляет запрос на поиск геообъектов. Добавить функцию-обработчик для обработки результата геокодирования можно через возвращаемый объект-promise. Объект, поступающий на вход функции-обработчика может содержать поля только следующего вида: <code>geoObjects</code> , <code>layers</code> , <code>mapState</code> , <code>metaData</code> .                                                                                                                                                                                                              |
| <code>suggest(request[, options])</code> | <code>vow.Promise</code> | Отправляет запрос за поисковыми подсказками. Возвращает объект <code>Promise</code> , который будет либо отклонён с ошибкой, либо подтверждён массивом объектов вида <code>{ displayName: "Мытищи, Московская область", value: "Россия, Московская область, Мытищи " }</code> . Поле <code>displayName</code> отвечает за представление топонима в удобном для чтения пользователем виде, поле <code>value</code> — значение, которое необходимо подставить в поле ввода после выбора пользователем данной подсказки.<br><br>Этот метод является опциональным. |

## Описание методов

## search

```
{vow.Promise} search(request[, options])
```

Отправляет запрос на поиск геообъектов. Добавить функцию-обработчик для обработки результата геокодирования можно через возвращаемый объект-promise. Объект, поступающий на вход функции-обработчика может содержать поля только следующего вида: `geoObjects`, `layers`, `mapState`, `metaData`.

**Возвращает** объект-promise.

## Параметры:

| Параметр               | Значение по умолчанию | Описание                                        |
|------------------------|-----------------------|-------------------------------------------------|
| <code>request</code> * | —                     | Тип: <code>String</code><br><br>Строка запроса. |

| Параметр                       | Значение по умолчанию | Описание                                                                                                                                                                          |
|--------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>options</code>           | —                     | Тип: Object<br><br>Опции.                                                                                                                                                         |
| <code>options.boundedBy</code> | —                     | Тип: Number[][]<br><br>Прямоугольная область на карте, где предположительно находится искомый объект. Должна быть задана как массив, например <code>[[30, 40], [50, 50]]</code> . |
| <code>options.results</code>   | —                     | Тип: Number<br><br>Максимальное количество возвращаемых результатов.                                                                                                              |
| <code>options.skip</code>      | —                     | Тип: Number<br><br>Число результатов, которое необходимо пропустить.                                                                                                              |

\* Обязательный параметр/опция.

## suggest

```
{vow.Promise} suggest(request[, options])
```

Отправляет запрос за поисковыми подсказками. Возвращает объект Promise, который будет либо отклонён с ошибкой, либо подтверждён массивом объектов вида `{ displayName: "Мытищи, Московская область", value: "Россия, Московская область, Мытищи " }`. Поле `displayName` отвечает за представление топонима в удобном для чтения пользователем виде, поле `value` — значение, которое необходимо подставить в поле ввода после выбора пользователем данной подсказки.

Этот метод является опциональным.

**Возвращает** объект Promise.

### Параметры:

| Параметр               | Значение по умолчанию | Описание                           |
|------------------------|-----------------------|------------------------------------|
| <code>request</code> * | —                     | Тип: String<br><br>Строка запроса. |
| <code>options</code>   | —                     | Тип: Object<br><br>Опции.          |



| Параметр                          | Значение по умолчанию | Описание                                                                                                                                                                                       |
|-----------------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>options.boundedBy</code>    | —                     | Тип: <code>Number[][]</code><br><br>Прямоугольная область на карте, где предположительно находится искомый объект. Должна быть задана как массив, например <code>[[30, 40], [50, 50]]</code> . |
| <code>options.results</code>      | —                     | Тип: <code>Number</code><br><br>Максимальное количество возвращаемых результатов.                                                                                                              |
| <code>options.strictBounds</code> | —                     | Тип: <code>Boolean</code><br><br>Искать только внутри области, заданной опцией <code>boundedBy</code> .                                                                                        |

\* Обязательный параметр/опция.

## ISelectableControl

Расширяет [IControl](#).

Интерфейс переключаемого и выбираемого элемента управления.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
ISelectableControl()
```

### Поля

| Имя                  | Тип                            | Описание                                                             |
|----------------------|--------------------------------|----------------------------------------------------------------------|
| <code>events</code>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <code>options</code> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">IControl</a> .        |

### События

| Имя                   | Описание                             |
|-----------------------|--------------------------------------|
| <code>deselect</code> | Элемент управления не выбран.        |
| <code>disable</code>  | Элемент управления стал недоступным. |
| <code>enable</code>   | Элемент управления стал доступным.   |

| Имя                          | Описание                                                                                                                                                                                                                             |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">parentchange</a> | Сменился родительский объект.<br>Поля данных: <ul style="list-style-type: none"> <li><code>oldParent</code> - старый родитель;</li> <li><code>newParent</code> - новый родитель.</li> </ul> Унаследовано от <a href="#">IChild</a> . |
| <a href="#">select</a>       | Элемент управления выбран.                                                                                                                                                                                                           |

## Методы

| Имя                               | Возвращает                              | Описание                                                                                                                                                                                  |
|-----------------------------------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">deselect()</a>        |                                         | Отменяет выбор элемента управления (выключает).                                                                                                                                           |
| <a href="#">disable()</a>         |                                         | Делает элемент управления недоступным (запрещены пользовательские действия).                                                                                                              |
| <a href="#">enable()</a>          |                                         | Делает элемент управления доступным (разрешены пользовательские действия).                                                                                                                |
| <a href="#">getParent()</a>       | <a href="#">IControlParent</a> или null | Возвращает ссылку на родительский объект или null, если родительский элемент не был установлен.<br>Унаследован от <a href="#">IControl</a> .                                              |
| <a href="#">isEnabled()</a>       | Boolean                                 | Возвращает true - элемент управления доступен, false - недоступен.                                                                                                                        |
| <a href="#">isSelected()</a>      | Boolean                                 | Возвращает true - элемент управления выбран, false - не выбран.                                                                                                                           |
| <a href="#">select()</a>          |                                         | Выбирает (включает) элемент управления.                                                                                                                                                   |
| <a href="#">setParent(parent)</a> | <a href="#">IChildOnMap</a>             | Устанавливает родительский объект. Если передать значение null, то элемент управления будет только удален из текущего родительского объекта.<br>Унаследован от <a href="#">IControl</a> . |

## Описание событий

### deselect

Элемент управления не выбран.

**disable**

Элемент управления стал недоступным.

**enable**

Элемент управления стал доступным.

**select**

Элемент управления выбран.

**Описание методов****deselect**

```
{ } deselect()
```

Отменяет выбор элемента управления (выключает).

**disable**

```
{ } disable()
```

Делает элемент управления недоступным (запрещены пользовательские действия).

**enable**

```
{ } enable()
```

Делает элемент управления доступным (разрешены пользовательские действия).

**isEnabled**

```
{Boolean} isEnabled()
```

**Возвращает** true - элемент управления доступен, false - недоступен.

**isSelected**

```
{Boolean} isSelected()
```

**Возвращает** true - элемент управления выбран, false - не выбран.

**select**

```
{ } select()
```

Выбирает (включает) элемент управления.

**ISelectableControlLayout**

Расширяет [ILayout](#).

Интерфейс макета кнопки.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

**Конструктор**

```
ISelectableControlLayout()
```

## Поля

| Имя                    | Тип                           | Описание                                                                |
|------------------------|-------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

## События

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>           | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">contextmenu</a>     | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">dblclick</a>        | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <a href="#">deselect</a>        | Событие, инициирующее выход элемента из состояния "выбран".                                                                                                                                                                                                                                                                  |
| <a href="#">emptinesschange</a> | Изменение признака пустоты макета. Экземпляр класса <a href="#">Event</a> .<br>Унаследовано от <a href="#">ILayout</a> .                                                                                                                                                                                                     |
| <a href="#">mousedown</a>       | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mouseenter</a>      | Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                      |
| <a href="#">mouseleave</a>      | Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mousemove</a>       | Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mouseup</a>         | Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchend</a>       | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchmove</a>      | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a>     | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">parentelementchange</a> | <p>Изменение родительского элемента. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">select</a>              | <p>Событие, инициирующее переход элемента в состояние "выбран".</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">shapechange</a>         | <p>Изменение фигуры области охватывающей макет. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <a href="#">wheel</a>               | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

**Методы**

| Имя                                      | Возвращает                   | Описание                                                                                                                                                                                                                                                                  |
|------------------------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>                |                              | Деструктор. Вызывается при завершении работы с макетом.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                       |
| <a href="#">getData()</a>                | Object                       | Возвращает объект данных макета.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                              |
| <a href="#">getParentElement()</a>       | HTMLElement                  | Возвращает родительский HTML-элемент.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                         |
| <a href="#">getShape()</a>               | <a href="#">IShape</a>  null | Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.<br>Унаследован от <a href="#">ILayout</a> . |
| <a href="#">isEmpty()</a>                | Boolean                      | Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.<br>Унаследован от <a href="#">ILayout</a> .                                                                        |
| <a href="#">setData(data)</a>            |                              | Задаёт данные макета.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                         |
| <a href="#">setParentElement(parent)</a> |                              | Добавляет макет в DOM-дерево.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                 |

**Описание событий****deselect**

Событие, инициирующее выход элемента из состояния "выбран".

**select**

Событие, инициирующее переход элемента в состояние "выбран".

**IShape**

Интерфейс фигуры. Фигура представляет совокупность пиксельной геометрии, а также математических и логических параметров ее отображения (таких как наличие контура и его ширина, наличие заливки).

[Конструктор](#) | [Методы](#)

## Конструктор

```
IShape()
```

## Методы

| Имя                             | Возвращает      | Описание                                                                                                                                                                                                                                        |
|---------------------------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>contains(position)</code> | Boolean         | Проверяет лежит ли переданная точка внутри фигуры.                                                                                                                                                                                              |
| <code>equals(shape)</code>      | Boolean         | Возвращает true, если переданная фигура эквивалентна данной.                                                                                                                                                                                    |
| <code>getBounds()</code>        | Number[][] null | Возвращает координаты двух противоположных углов области, охватывающей фигуру. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат. |
| <code>getGeometry()</code>      | IPixelGeometry  | Возвращает пиксельную геометрию фигуры.                                                                                                                                                                                                         |
| <code>getType()</code>          | String          | Возвращает идентификатор типа геометрии.                                                                                                                                                                                                        |
| <code>scale(factor)</code>      | IShape          | Создает отмасштабированную копию фигуры.                                                                                                                                                                                                        |
| <code>shift(offset)</code>      | IShape          | Создает копию фигуры сдвинутую на заданную величину.                                                                                                                                                                                            |

## Описание методов

### contains

```
{Boolean} contains(position)
```

Проверяет лежит ли переданная точка внутри фигуры.

**Возвращает** true, если переданная точка лежит внутри фигуры.

### Параметры:

| Параметр                | Значение по умолчанию | Описание                           |
|-------------------------|-----------------------|------------------------------------|
| <code>position *</code> | —                     | Тип: Number[]<br>Координаты точки. |

\* Обязательный параметр/опция.

### equals

```
{Boolean} equals(shape)
```

**Возвращает** true, если переданная фигура эквивалентна данной.

**Параметры:**

| Параметр | Значение по умолчанию | Описание                                                                     |
|----------|-----------------------|------------------------------------------------------------------------------|
| shape *  | —                     | Тип: <a href="#">IShape</a><br><br>Фигура, с которой производится сравнение. |

\* Обязательный параметр/опция.

### getBounds

```
{Number[][]|null} getBounds()
```

**Возвращает** координаты двух противоположных углов области, охватывающей фигуру. Первый элемент массива - угол с наименьшими значениями координат относительно остальных точек области; второй элемент - угол с наибольшими значениями координат.

### getGeometry

```
{IPixelGeometry} getGeometry()
```

**Возвращает** пиксельную геометрию фигуры.

### getType

```
{String} getType()
```

**Возвращает** идентификатор типа геометрии.

### scale

```
{IShape} scale(factor)
```

Создает отмасштабированную копию фигуры.

**Возвращает** отмасштабированную копию фигуры.

**Параметры:**

| Параметр | Значение по умолчанию | Описание                                   |
|----------|-----------------------|--------------------------------------------|
| factor * | —                     | Тип: Number<br><br>Масштабный коэффициент. |

\* Обязательный параметр/опция.

### shift

```
{IShape} shift(offset)
```

Создает копию фигуры сдвинутую на заданную величину.



**Возвращает** сдвинутую копию фигуры.

**Параметры:**

| Параметр              | Значение по умолчанию | Описание                                                   |
|-----------------------|-----------------------|------------------------------------------------------------|
| <code>offset</code> * | —                     | Тип: <code>Number[]</code><br><br>Величина сдвига по осям. |

\* Обязательный параметр/опция.

## ISuggestProvider

Интерфейс провайдера поисковых подсказок.

[Конструктор](#) | [Методы](#)

### Конструктор

```
ISuggestProvider()
```

### Методы

| Имя                                      | Возвращает               | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>suggest(request[, options])</code> | <code>vow.Promise</code> | Отправляет запрос за поисковыми подсказками. Возвращает объект <code>Promise</code> , который будет либо отклонён с ошибкой, либо подтверждён массивом объектов вида <code>{ displayName: "Мытищи, Московская область", value: "Россия, Московская область, Мытищи ", hl: [[0,5]] }</code> . Поле <code>displayName</code> отвечает за представление топонима в удобном для чтения пользователем виде. Поле <code>value</code> – значение, которое необходимо подставить в поле ввода после выбора пользователем данной подсказки. Поле <code>hl</code> – массив диапазонов подсветки, необходимых, чтобы показать, какая именно часть результата совпала с запросом. Диапазон подсветки - массив из двух чисел: индексы начального и конечного символов диапазона. |

### Описание методов

#### `suggest`

```
{vow.Promise} suggest(request[, options])
```

Отправляет запрос за поисковыми подсказками. Возвращает объект Promise, который будет либо отклонён с ошибкой, либо подтверждён массивом объектов вида { displayName: "Мытищи, Московская область", value: "Россия, Московская область, Мытищи ", hl: [[0,5]] }. Поле displayName отвечает за представление топонима в удобном для чтения пользователем виде. Поле value – значение, которое необходимо подставить в поле ввода после выбора пользователем данной подсказки. Поле hl – массив диапазонов подсветки, необходимых, чтобы показать, какая именно часть результата совпала с запросом. Диапазон подсветки - массив из двух чисел: индексы начального и конечного символов диапазона.

**Возвращает** объект Promise.

**Параметры:**

| Параметр                       | Значение по умолчанию | Описание                                                                                                                                                            |
|--------------------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>request *</code>         | —                     | Тип: String<br><br>Строка запроса.                                                                                                                                  |
| <code>options</code>           | —                     | Тип: Object<br><br>Опции.                                                                                                                                           |
| <code>options.boundedBy</code> | —                     | Тип: Number[][]<br><br>Прямоугольная область на карте, где предположительно находится искомый объект. Должна быть задана как массив, например [[30, 40], [50, 50]]. |
| <code>options.results</code>   | —                     | Тип: Number<br><br>Максимальное количество возвращаемых результатов.                                                                                                |

\* Обязательный параметр/опция.

## ISuggestViewLayout

Интерфейс макета панели поисковых подсказок.

[Конструктор](#) | [События](#)

### Конструктор

```
ISuggestViewLayout()
```

### События

| Имя                 | Описание                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>hover</code>  | Событие, возникающее при выделении одной из подсказок в списке наведением мыши на результат или с помощью клавиатуры. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>index</code>: Number null – индекс элемента в списке поисковых подсказок.</li> </ul> |
| <code>select</code> | Событие, возникающее при выборе пользователем одной из подсказок. Экземпляр класса <a href="#">Event</a> . Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>item</code>: Object – объект с полями <code>displayName</code> и <code>value</code>.</li> </ul>                                          |

## Описание событий

### hover

Событие, возникающее при выделении одной из подсказок в списке наведением мыши на результат или с помощью клавиатуры. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `index`: `Number|null` – индекс элемента в списке поисковых подсказок.

### select

Событие, возникающее при выборе пользователем одной из подсказок. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `item`: `Object` – объект с полями `displayName` и `value`.

## ITile

Расширяет [IEventEmitter](#).

Интерфейс тайла.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
ITile(url)
```

### Параметры:

| Параметр           | Значение по умолчанию | Описание                               |
|--------------------|-----------------------|----------------------------------------|
| <code>url</code> * | —                     | Тип: <code>String</code><br>URL тайла. |

\* Обязательный параметр/опция.

### Поля

| Имя                 | Тип                           | Описание                                                             |
|---------------------|-------------------------------|----------------------------------------------------------------------|
| <code>events</code> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |

### События

| Имя                | Описание                  |
|--------------------|---------------------------|
| <code>ready</code> | Событие готовности тайла. |

### Методы

| Имя                    | Возвращает           | Описание                    |
|------------------------|----------------------|-----------------------------|
| <code>destroy()</code> |                      | Уничтожает тайл.            |
| <code>isReady()</code> | <code>Boolean</code> | Проверяет готовность тайла. |

## Описание событий

### ready

Событие готовности тайла.

## Описание методов

### destroy

```
{ } destroy()
```

Уничтожает тайл.

### isReady

```
{Boolean} isReady()
```

Проверяет готовность тайла.

**Возвращает** true - тайл готов, false - тайл не готов.

## ITrafficControlLayout

Расширяет [IExpandableControlLayout](#).

Интерфейс макета элемента управления "Панель управления пробками".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
ITrafficControlLayout()
```

### Поля

| Имя                    | Тип                           | Описание                                                                |
|------------------------|-------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

### События

| Имя                         | Описание                                                                                                                                                                                                                                                                                                                     |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>       | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">collapse</a>    | Событие, инициирующее сворачивание объекта.<br>Унаследовано от <a href="#">IExpandableControlLayout</a> .                                                                                                                                                                                                                    |
| <a href="#">contextmenu</a> | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                         |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">dblclick</a>        | <p>Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">emptinesschange</a> | <p>Изменение признака пустоты макета. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                |
| <a href="#">expand</a>          | <p>Событие, инициирующее разворачивание объекта.</p> <p>Унаследовано от <a href="#">IExpandableControlLayout</a>.</p>                                                                                                                                                                                                            |
| <a href="#">hide</a>            | <p>Событие, инициирующее удаление пробок с карты. Скрыть пробки.</p>                                                                                                                                                                                                                                                             |
| <a href="#">mousedown</a>       | <p>Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>             |
| <a href="#">mouseenter</a>      | <p>Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                  |
| <a href="#">mouseleave</a>      | <p>Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>             |
| <a href="#">mousemove</a>       | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>             |
| <a href="#">mouseup</a>         | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>          |
| <a href="#">multitouchend</a>   | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                             |

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">multitouchmove</a>      | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a>     | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <code>IMultiTouchEvent</code> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">parentelementchange</a> | <p>Изменение родительского элемента. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <a href="#">providerkeychange</a>   | <p>Событие, инициирующее смену ключа провайдера. Экземпляр класса <a href="#">Event</a>. Имена полей, доступных через метод <a href="#">Event.get</a>:</p> <ul style="list-style-type: none"> <li><code>newProviderKey</code> - новое значение ключа провайдера;</li> <li><code>oldProviderKey</code> – старое значение ключа.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <a href="#">shapechange</a>         | <p>Изменение фигуры области охватывающей макет. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <a href="#">show</a>                | <p>Событие, инициирующее показ пробок на карте.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">wheel</a>               | <p>Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## Методы

| Имя                                      | Возвращает                   | Описание                                                                                                                                                                                                                                                                  |
|------------------------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>                |                              | Деструктор. Вызывается при завершении работы с макетом.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                       |
| <a href="#">getData()</a>                | Object                       | Возвращает объект данных макета.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                              |
| <a href="#">getParentElement()</a>       | HTMLElement                  | Возвращает родительский HTML-элемент.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                         |
| <a href="#">getShape()</a>               | <a href="#">IShape</a>  null | Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.<br>Унаследован от <a href="#">ILayout</a> . |
| <a href="#">isEmpty()</a>                | Boolean                      | Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.<br>Унаследован от <a href="#">ILayout</a> .                                                                        |
| <a href="#">setData(data)</a>            |                              | Задаёт данные макета.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                         |
| <a href="#">setParentElement(parent)</a> |                              | Добавляет макет в DOM-дерево.<br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                 |

## Описание событий

**hide**

Событие, инициирующее удаление пробок с карты. Скрыть пробки.

**providerkeychange**

Событие, инициирующее смену ключа провайдера. Экземпляр класса [Event](#). Имена полей, доступных через метод [Event.get](#):

- `newProviderKey` - новое значение ключа провайдера;
- `oldProviderKey` – старое значение ключа.

**show**

Событие, инициирующее показ пробок на карте.

## ITrafficProvider

Расширяет [ICustomizable](#), [IEventEmitter](#).

Интерфейс провайдера данных о пробках.

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
ITrafficProvider()
```

### Поля

| Имя                     | Тип                            | Описание                                                             |
|-------------------------|--------------------------------|----------------------------------------------------------------------|
| <a href="#">events</a>  | <a href="#">IEventManager</a>  | Менеджер событий.<br>Унаследовано от <a href="#">IEventEmitter</a> . |
| <a href="#">options</a> | <a href="#">IOptionManager</a> | Менеджер опций.<br>Унаследовано от <a href="#">ICustomizable</a> .   |

### События

| Имя                           | Описание                                                                       |
|-------------------------------|--------------------------------------------------------------------------------|
| <a href="#">optionschange</a> | Изменение в опциях объекта.<br>Унаследовано от <a href="#">ICustomizable</a> . |

### Методы

| Имя                                               | Возвращает                | Описание                    |
|---------------------------------------------------|---------------------------|-----------------------------|
| <a href="#">getMap()</a>                          | <a href="#">Map</a>  null | Возвращает ссылку на карту. |
| <a href="#">setMap(<a href="#">Reference</a>)</a> |                           |                             |

### Описание методов

#### getMap

```
{Map|null} getMap()
```

**Возвращает** ссылку на карту.

#### setMap

```
{Reference} setMap(Reference)
```

### Параметры:

| Параметр                    | Значение по умолчанию | Описание                                    |
|-----------------------------|-----------------------|---------------------------------------------|
| <a href="#">Reference</a> * | —                     | Тип: <a href="#">Map</a>  null<br>на карту. |

\* Обязательный параметр/опция.



## ITransportProperties

Данный класс не имеет конструктора и предназначен для описания объекта данных транспортного средства в транспортном сегменте маршрута общественного транспорта.

[Конструктор](#) | [Поля](#)

### Конструктор

```
ITransportProperties()
```

### Поля

| Имя                  | Тип    | Описание                                                                                                                                                                                                                                                                                                                                |
|----------------------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">id</a>   | String | Идентификатор транспортного средства.                                                                                                                                                                                                                                                                                                   |
| <a href="#">name</a> | String | Название маршрута транспортного средства.                                                                                                                                                                                                                                                                                               |
| <a href="#">type</a> | String | Идентификатор вида транспортного средства. Может принимать одно из следующих строковых значений: <ul style="list-style-type: none"><li>"bus" – автобус;</li><li>"trolleybus" – троллейбус;</li><li>"tramway" – трамвай;</li><li>"minibus" – маршрутное такси;</li><li>"underground" – метро;</li><li>"suburban" – электричка.</li></ul> |

### Описание полей

#### id

```
{String} id
```

Идентификатор транспортного средства.

#### name

```
{String} name
```

Название маршрута транспортного средства.

#### type

```
{String} type
```

Идентификатор вида транспортного средства. Может принимать одно из следующих строковых значений:

- "bus" – автобус;
- "trolleybus" – троллейбус;
- "tramway" – трамвай;
- "minibus" – маршрутное такси;
- "underground" – метро;
- "suburban" – электричка.

## IZoomControlLayout

Расширяет [ILayout](#).

Интерфейс макета элемента управления "Ползунок масштаба".

[Конструктор](#) | [Поля](#) | [События](#) | [Методы](#)

### Конструктор

```
IZoomControlLayout()
```

### Поля

| Имя                    | Тип                           | Описание                                                                |
|------------------------|-------------------------------|-------------------------------------------------------------------------|
| <a href="#">events</a> | <a href="#">IEventManager</a> | Менеджер событий.<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |

### События

| Имя                             | Описание                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">click</a>           | Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> . |
| <a href="#">contextmenu</a>     | Вызов контекстного меню на элементе. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .              |
| <a href="#">dblclick</a>        | Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .     |
| <a href="#">emptinesschange</a> | Изменение признака пустоты макета. Экземпляр класса <a href="#">Event</a> .<br>Унаследовано от <a href="#">ILayout</a> .                                                                                                                                                                                                     |
| <a href="#">mousedown</a>       | Нажатие кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |
| <a href="#">mouseenter</a>      | Наведение курсора на объект. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                      |
| <a href="#">mouseleave</a>      | Вывод курсора за пределы объекта. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEvent.manager</a> .<br>Унаследовано от <a href="#">IDomEventEmitter</a> .                 |

| Имя                                 | Описание                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">mousemove</a>           | <p>Перемещение курсора над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <a href="#">mouseup</a>             | <p>Отпускание кнопки мыши над объектом. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MouseEvent</a>. Более детальное описание доступно в <a href="#">domEvent.manager</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <a href="#">multitouchend</a>       | <p>Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a>.</p> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <a href="#">multitouchmove</a>      | <p>Повторяющееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p> |
| <a href="#">multitouchstart</a>     | <p>Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса <a href="#">IMultiTouchEvent</a> с информацией о прикосновениях. Определяет свойство <code>touches</code>, которое содержит список прикосновений. Каждое прикосновение описано объектом, которое содержит поля:</p> <ul style="list-style-type: none"> <li><code>clientX</code> - координата касания X относительно видимой области браузера;</li> <li><code>clientY</code> - координата касания Y относительно видимой области браузера;</li> <li><code>pageX</code> - координата касания X касания относительно начала документа;</li> <li><code>pageY</code> - координата касания Y касания относительно начала документа.</li> </ul> <p>Унаследовано от <a href="#">IDomEventEmitter</a>.</p>                   |
| <a href="#">parentelementchange</a> | <p>Изменение родительского элемента. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <a href="#">shapechange</a>         | <p>Изменение фигуры области охватывающей макет. Экземпляр класса <a href="#">Event</a>.</p> <p>Унаследовано от <a href="#">ILayout</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

| Имя                        | Описание                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">wheel</a>      | Скролл мышкой. При использовании следует иметь в виду, что при сенсорном вводе события мыши будут эмулироваться. Экземпляр класса <a href="#">MapEvent</a> . Более детальное описание доступно в <a href="#">domEventManager</a> .<br><br>Унаследовано от <a href="#">IDomEventEmitter</a> .                                                                                                           |
| <a href="#">zoomchange</a> | Событие, инициирующее смену коэффициента масштабирования карты. Изменить коэффициент масштабирования карты. Экземпляр класса Event. Имена полей, доступных через метод <a href="#">Event.get</a> : <ul style="list-style-type: none"> <li><code>newZoom</code> - новое значение коэффициента масштабирования;</li> <li><code>oldZoom</code> - старое значение коэффициента масштабирования.</li> </ul> |

## Методы

| Имя                                      | Возвращает                   | Описание                                                                                                                                                                                                                                                                      |
|------------------------------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">destroy()</a>                |                              | Деструктор. Вызывается при завершении работы с макетом.<br><br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                       |
| <a href="#">getData()</a>                | Object                       | Возвращает объект данных макета.<br><br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                              |
| <a href="#">getParentElement()</a>       | HTMLElement                  | Возвращает родительский HTML-элемент.<br><br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                         |
| <a href="#">getShape()</a>               | <a href="#">IShape</a>  null | Возвращает фигуру, описывающую форму области, охватывающей макет, либо null, если нет возможности рассчитать данную фигуру. Координаты геометрии фигуры должны отсчитываться от точки привязки родительского элемента макета.<br><br>Унаследован от <a href="#">ILayout</a> . |
| <a href="#">isEmpty()</a>                | Boolean                      | Возвращает true, если макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.<br><br>Унаследован от <a href="#">ILayout</a> .                                                                        |
| <a href="#">setData(data)</a>            |                              | Задаёт данные макета.<br><br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                         |
| <a href="#">setParentElement(parent)</a> |                              | Добавляет макет в DOM-дерево.<br><br>Унаследован от <a href="#">ILayout</a> .                                                                                                                                                                                                 |

**Описание событий****zoomchange**

Событие, инициирующее смену коэффициента масштабирования карты. Изменить коэффициент масштабирования карты. Экземпляр класса Event. Имена полей, доступных через метод [Event.get](#):

- newZoom - новое значение коэффициента масштабирования;
- oldZoom - старое значение коэффициента масштабирования.