Справочник API Яндекс.Карт

Версия 2.0.31

2.07.2013

Яндекс

Справочник API Яндекс.Карт. Версия 2.0.31. Версия 2.0
Дата сборки документа: 2.07.2013.
Этот документ является составной частью технической документации Яндекса.
Сайт справки к сервисам Яндекса: http://help.yandex.ru
© 2008—2013 ООО «ЯНДЕКС». Все права защищены.

Предупреждение об исключительных правах

Яндексу (а также указанному им правообладателю) принадлежат исключительные права на все результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации, используемые при разработке, поддержке и эксплуатации сервиса Справочник API Яндекс.Карт. К таким результатам могут относиться, но не ограничиваясь указанными, программы для ЭВМ, базы данных, изображения, тексты, другие произведения, а также изобретения, полезные модели, товарные знаки, знаки обслуживания, коммерческие обозначения и фирменные наименования. Эти права охраняются в соответствии с Гражданским кодексом РФ и международным правом.

Вы можете использовать сервис Справочник API Яндекс.Карт или его составные части только в рамках полномочий, предоставленных вам Пользовательским соглашением сервиса Справочник API Яндекс.Карт или специального соглашения.

Нарушение требований по защите исключительных прав правообладателя влечет за собой дисциплинарную, гражданско-правовую, административную или уголовную ответственность в соответствии с российским законодательством.

Контактная информация

ООО «ЯНДЕКС»
http://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

О справочнике ... 10
Пакеты ... 10
Balloon ... 20

behavior ... 27
- behavior.DblClickZoom ... 27
- behavior.Drag ... 28
- behavior.LeftMouseButtonMagnifier ... 30
- behavior.MultiTouch ... 32
- behavior.RightMouseButtonMagnifier ... 33
- behavior.RouteEditor ... 34
- behavior.Ruler ... 36
- behavior.ScrollZoom ... 38
- behavior.storage ... 40

Circle ... 41

cluster ... 46
- cluster.Balloon ... 46

Cluster ... 48
Clusterer ... 60

collection ... 67
- collection.Item ... 67

Collection ... 71

control ... 77
- control.Button ... 77
- control.Group ... 82
- control.ListBox ... 88
- control.ListBoxItem ... 96
- control.ListBoxSeparator ... 99
- control.MapTools ... 101
- control.MiniMap ... 105
- control.RadioGroup ... 110
- control.RollupButton ... 114
- control.RouteEditor ... 120
- control.ScaleLine ... 122
- control.SearchControl ... 124
- control.SmallZoomControl ... 134
- control.storage ... 138
- control.ToolBar ... 138
- control.ToolBarSeparator ... 142
- control.TrafficControl ... 144
- control.TypeSelector ... 150
- control.ZoomControl ... 157

coordSystem ... 160
- coordSystem.cartesian ... 160
- coordSystem.geo ... 160

data ... 162
- data.Manager ... 162

DomEvent ... 165

domEvent ... 166
- domEvent.manager ... 166
- domEvent.MultiPointer ... 169
- domEvent.MultiTouch ... 169
- domEvent.Pointer ... 169
- domEvent.Touch ... 170

Event ... 171

event ... 174
- event.Group ... 174
- event.Manager ... 175
- event.PriorityGroup ... 177
- event.PriorityManager ... 179

formatter ... 184
geocode ... 185
geolocation ... 188

geometry ... 189

- **geometry.base** ... 189
  - geometry.base.Circle ... 189
  - **geometry.base.LinearRing** ... 191
    - geometry.base.LinearRing.fromEncodedCoordinates ... 194
    - geometry.base.LinearRing.toEncodedCoordinates ... 194
  - **geometry.base.LineString** ... 195
    - geometry.base.LineString.fromEncodedCoordinates ... 197
    - geometry.base.LineString.toEncodedCoordinates ... 197
  - geometry.base.Point ... 198
  - **geometry.base.Polygon** ... 199
    - geometry.base.Polygon.fromEncodedCoordinates ... 202
    - geometry.base.Polygon.toEncodedCoordinates ... 203
  - geometry.base.Rectangle ... 204
- geometry.Circle ... 206
- **geometry.LineString** ... 208
  - geometry.LineString.fromEncodedCoordinates ... 213
  - geometry.LineString.getPixelRadius ... 213
  - geometry.LineString.toEncodedCoordinates ... 213
- **geometry.pixel** ... 214
  - geometry.pixel.Circle ... 214
  - geometry.pixel.LineString ... 215
  - geometry.pixel.MultiPolygon ... 216
  - geometry.pixel.Point ... 219
  - geometry.pixel.Polygon ... 220
  - geometry.pixel.Rectangle ... 223
- geometry.Point ... 225
- **geometry.Polygon** ... 227
  - geometry.Polygon.fromEncodedCoordinates ... 232
  - geometry.Polygon.toEncodedCoordinates ... 233
- geometry.Rectangle ... 234

geometryEditor ... 239
- geometryEditor.LineString ... 239
- **geometryEditor.model** ... 247
  - geometryEditor.model.ChildLinearRing ... 247
  - geometryEditor.model.ChildLineString ... 249
  - geometryEditor.model.ChildVertex ... 251
  - geometryEditor.model.Edge ... 254
  - geometryEditor.model.EdgeGeometry ... 256
  - geometryEditor.model.RootLineString ... 257
  - geometryEditor.model.RootPolygon ... 259
- geometryEditor.Point ... 261

geometryEditor.Polygon ... 263

geometryEditor.view ... 272
geometryEditor.view.Edge ... 272
geometryEditor.view.MultiPath ... 273
geometryEditor.view.Path ... 274
geometryEditor.view.Vertex ... 276

GeoObject ... 277

geoObject ... 291
geoObject.Balloon ... 291
geoObject.Hint ... 293

geoObject.overlayFactory ... 296
geoObject.overlayFactory.hotspot ... 296
geoObject.overlayFactory.interactive ... 297
geoObject.overlayFactory.interactiveGraphics ... 298
geoObject.overlayFactory.staticGraphics ... 298
geoObject.overlayFactory.storage ... 299

geoObject.OverlayFactory ... 300

GeoObjectArray ... 301
GeoObjectCollection ... 308
geoQuery ... 314
GeoQueryResult ... 316

geoXml ... 354
geoXml.load ... 354

getZoomRange ... 356

graphics ... 357

graphics.style ... 357
graphics.style.color ... 357
graphics.style.stroke ... 357

Hint ... 360

hotspot ... 366
hotspot.Layer ... 366

hotspot.layer ... 370
hotspot.layer.Balloon ... 370
hotspot.layer.Hint ... 372

hotspot.ObjectSource ... 374
hotspot.Shape ... 381

interactivityModel ... 385
interactivityModel.storage ... 385

layer ... 386
layer.storage ... 386

layer.tile ... 386
layer.tile.CanvasTile ... 386
layer.tile.DomTile ... 388

layer.tileContainer ... 390
layer.tileContainer.CanvasContainer ... 390
layer.tileContainer.DomContainer ... 392

Layer ... 395
LayerCollection ... 401

layout ... 407

layout.Image ... 407
layout.ImageWithContent ... 409
layout.storage ... 412

layout.templateBased ... 413
layout.templateBased.Base ... 413

load ... 417

map ... 418

map.action ... 418
map.action.Continuous ... 418
map.action.Manager ... 420
map.action.Single ... 424

map.Balloon ... 428

map.behavior ... 430
map.behavior.Manager ... 430

map.Container ... 435

map.control ... 436
map.control.Manager ... 436

map.Converter ... 440
map.Copyrights ... 441
map.GeoObjects ... 444
map.Hint ... 447

map.layer ... 449
map.layer.Manager ... 449

map.pane ... 453
map.pane.Manager ... 453

map.ZoomRange ... 455

Map ... 457
MapEvent ... 480

mapType ... 483
mapType.storage ... 483

MapType ... 484
Monitor ... 485

option ... 489
option.Manager ... 489
option.presetStorage ... 493

overlay ... 502

overlay.hotspot ... 502
overlay.hotspot.Base ... 502
overlay.hotspot.Circle ... 504
overlay.hotspot.Placemark ... 507
overlay.hotspot.Polygon ... 509
overlay.hotspot.Polyline ... 511
overlay.hotspot.Rectangle ... 514

overlay.html ... 517
overlay.html.Balloon ... 517
overlay.html.Label ... 521
overlay.html.Placemark ... 524

overlay.html.rectangle ... 528

overlay.html.rectangle.Layout ... 528
overlay.html.Rectangle ... 532
overlay.interactiveGraphics ... 536
overlay.interactiveGraphics.Circle ... 536
overlay.interactiveGraphics.Placemark ... 538
overlay.interactiveGraphics.Polygon ... 540
overlay.interactiveGraphics.Polyline ... 543
overlay.interactiveGraphics.Rectangle ... 545
overlay.staticGraphics ... 549
overlay.staticGraphics.Circle ... 549
overlay.staticGraphics.Placemark ... 550
overlay.staticGraphics.Polygon ... 551
overlay.staticGraphics.Polyline ... 552
overlay.staticGraphics.Rectangle ... 553
Placemark ... 555
Polygon ... 559
Polyline ... 564
projection ... 569
projection.Cartesian ... 569
projection.sphericalMercator ... 570
projection.wgs84Mercator ... 570
ready ... 572
Rectangle ... 572
regions ... 578
regions.load ... 578
route ... 579
router ... 581
router.Editor ... 581
router.Path ... 583
router.Route ... 589
router.Segment ... 594
router.ViaPoint ... 598
templateLayoutFactory ... 603
theme ... 606
theme.twirl ... 606
theme.twirl.balloon ... 606
theme.twirl.balloon.Layout ... 606
theme.twirl.geoObject ... 610
theme.twirl.geoObject.layout ... 610
theme.twirl.geoObject.layout.StretchyIcon ... 610
traffic ... 613
traffic.provider ... 613
traffic.provider.Actual ... 613
traffic.provider.Archive ... 615
traffic.provider.Forecast ... 618
traffic.provider.storage ... 621
util ... 622
util.augment ... 622
util.bind ... 622
util.bounds ... 623

util.cursor ... 623
util.cursor.Accessor ... 623
util.cursor.Manager ... 624

util.Dragger ... 627
util.extend ... 630

util.math ... 631
util.math.areEqual ... 631
util.math.cycleRestrict ... 632
util.math.restrict ... 632

util.Promise ... 633
util.Storage ... 634

Интерфейсы ... 636
IBalloonLayout ... 636
IBaseCircleGeometry ... 639
IBaseGeometry ... 640
IBaseLinearRingGeometry ... 641
IBaseLineStringGeometry ... 643
IBasePointGeometry ... 645
IBasePolygonGeometry ... 646
IBaseRectangleGeometry ... 649
IBehavior ... 650
ICanvasTile ... 652
IChild ... 654
IChildOnMap ... 655
ICircleGeometry ... 656
ICircleGeometryAccess ... 658
ICollection ... 661
IControl ... 663
IControlParent ... 664
ICoordSystem ... 665
ICopyrightsAccessor ... 668
ICopyrightsProvider ... 669
ICustomizable ... 671
IDataManager ... 672
IDomEvent ... 673
IDomEventEmitter ... 674
IDomTile ... 677
IEvent ... 678
IEventController ... 679
IEventEmitter ... 680
IEventGroup ... 680
IEventManager ... 682
IEventWorkflowController ... 684
IExpandableControlLayout ... 685
IFreezable ... 688
IGeocodeProvider ... 690
IGeometry ... 691
IGeometryEditor ... 693
IGeometryEditorChildModel ... 694
IGeometryEditorModel ... 696
IGeometryEditorRootModel ... 697
IGeoObject ... 698
IGroupControlLayout ... 702
IHotspotObjectSource ... 705
IHotspotShape ... 706
IHotspotShapeGeometry ... 710
IIterator ... 710
ILayer ... 711
ILayout ... 714

ILinearRingGeometryAccess ... 718
ILineStringGeometry ... 723
ILineStringGeometryAccess ... 725
IMapAction ... 730
IMapObjectCollection ... 731
IMapState ... 732
IOptionManager ... 733
IOverlay ... 736
IOverlayFactory ... 739
IPane ... 740
IParentOnMap ... 742
IPixelCircleGeometry ... 743
IPixelGeometry ... 744
IPixelLineStringGeometry ... 746
IPixelPointGeometry ... 747
IPixelPolygonGeometry ... 748
IPixelRectangleGeometry ... 751
IPointGeometry ... 753
IPointGeometryAccess ... 755
IPolygonGeometry ... 756
IPolygonGeometryAccess ... 759
IPositioningContext ... 764
IProjection ... 765
IPromiseProvider ... 767
IRectangleGeometry ... 768
IRectangleGeometryAccess ... 769
ISearchControlLayout ... 772
ISelectableControl ... 775
ISelectableControlLayout ... 777
ITrafficControlLayout ... 780
ITrafficProvider ... 783
IZoomControlLayout ... 784

О справочнике

Справочник содержит описание программного интерфейса **JavaScript API** версии 2.0.31.

Целевой аудиторией справочника являются разработчики сайтов, которые хотят использовать интерактивные Яндекс.Карты на своих веб-страницах.

Справочник содержит описания открытых классов и методов API и рассчитан на разработчиков, знакомых с JavaScript и сервисом Яндекс.Карты. Классы перечислены в алфавитном порядке.

Пакеты

package.full

Полная сборка API Яндекс.Карт.

Компоненты:

- Balloon
- Circle
- Clusterer
- Collection
- Event
- GeoObject
- GeoObjectArray
- GeoObjectCollection
- GeoQueryResult
- Hint
- Layer
- LayerCollection
- Map
- MapEvent
- MapType
- Monitor
- Placemark
- Polygon
- Polyline
- Rectangle
- behavior.DblClickZoom
- behavior.Drag
- behavior.LeftMouseButtonMagnifier
- behavior.MultiTouch
- behavior.RightMouseButtonMagnifier
- behavior.RouteEditor
- behavior.Ruler
- behavior.ScrollZoom
- behavior.storage
- collection.Item
- control.Button
- control.Group
- control.ListBox
- control.ListBoxItem
- control.ListBoxSeparator
- control.MapTools
- control.MiniMap
- control.RadioGroup
- control.RollupButton
- control.RouteEditor
- control.ScaleLine
- control.SearchControl
- control.SmallZoomControl
- control.ToolBar
- control.ToolBarSeparator
- control.TrafficControl
- control.TypeSelector

package.standard

Стандартная сборка API Яндекс.Карт.

Компоненты:

- Balloon
- Collection
- Event
- GeoObjectArray
- GeoObjectCollection
- Hint
- Layer
- LayerCollection
- Map
- MapEvent
- MapType
- Monitor
- Placemark
- behavior.DblClickZoom
- behavior.Drag
- behavior.LeftMouseButtonMagnifier
- behavior.MultiTouch
- behavior.RightMouseButtonMagnifier
- behavior.Ruler
- behavior.ScrollZoom
- behavior.storage
- collection.Item
- control.Button
- control.Group
- control.ListBox
- control.ListBoxItem
- control.ListBoxSeparator
- control.MapTools
- control.MiniMap
- control.RadioGroup
- control.RollupButton
- control.ScaleLine
- control.SearchControl
- control.SmallZoomControl
- control.ToolBar
- control.ToolBarSeparator
- control.TypeSelector
- control.ZoomControl
- coordSystem.cartesian
- coordSystem.geo
- data.Manager
- domEvent.manager
- event.Manager
- event.PriorityManager
- formatter
- geocode
- geolocation

package.map

Базовая функциональность — карта, поведения карты, слои.

Компоненты:

- Collection
- Event
- Layer
- LayerCollection
- Map
- MapEvent
- MapType
- Monitor
- behavior.DblClickZoom
- behavior.Drag
- behavior.LeftMouseButtonMagnifier
- behavior.MultiTouch
- behavior.RightMouseButtonMagnifier
- behavior.ScrollZoom
- behavior.storage
- collection.Item
- coordSystem.cartesian
- coordSystem.geo
- data.Manager
- domEvent.manager
- event.Manager
- event.PriorityManager
- formatter
- geolocation
- getZoomRange
- interactivityModel.storage
- layer.storage
- map.action.Continuous
- map.action.Single
- mapType.storage
- option.Manager
- option.presetStorage
- projection.Cartesian
- projection.sphericalMercator
- projection.wgs84Mercator
- util.Dragger
- util.Promise
- util.Storage
- util.augment
- util.bind
- util.bounds
- util.extend
- util.math.areEqual
- util.math.cycleRestrict
- util.math.restrict

package.controls

Элементы управления карты (кроме SearchControl и TrafficControl).
Компоненты:

- behavior.Drag
- behavior.LeftMouseButtonMagnifier
- behavior.Ruler
- control.Button
- control.Group
- control.ListBox
- control.ListBoxItem
- control.ListBoxSeparator
- control.MapTools
- control.MiniMap
- control.RadioGroup
- control.RollupButton
- control.ScaleLine
- control.SmallZoomControl
- control.ToolBar
- control.ToolBarSeparator
- control.TypeSelector
- control.ZoomControl

package.search

Инструменты поиска по карте — провайдеры геокодирования и SearchControl.
Компоненты:

- Balloon
- Hint
- Placemark
- control.SearchControl
- geocode
- geometry.Point
- layout.Image
- layout.ImageWithContent
- layout.storage
- layout.templateBased.Base
- templateLayoutFactory

package.geoObjects

Геообъекты (точки, ломаные, многоугольники, круги, прямоугольники) на карте.

Компоненты:

- Balloon
- Circle
- GeoObject
- GeoObjectArray
- GeoObjectCollection
- Hint
- Placemark
- Polygon
- Polyline
- Rectangle
- geoObject.overlayFactory.hotspot
- geoObject.overlayFactory.interactive
- geoObject.overlayFactory.interactiveGraphics
- geoObject.overlayFactory.staticGraphics
- geoObject.overlayFactory.storage
- geometry.Circle
- geometry.LineString
- geometry.Point
- geometry.Polygon
- geometry.Rectangle
- layout.Image
- layout.ImageWithContent
- layout.storage
- layout.templateBased.Base
- templateLayoutFactory

package.clusters

Кластеризация.
Компоненты:

- Balloon
- Clusterer
- Hint
- Placemark
- geometry.Point
- layout.Image
- layout.ImageWithContent
- layout.storage
- layout.templateBased.Base
- templateLayoutFactory

package.traffic

Пробки.

Компоненты:

- control.TrafficControl
- geoObject.overlayFactory.interactive
- geometry.Point
- traffic.provider.Actual
- traffic.provider.Archive
- traffic.provider.Forecast

package.route

Сервис маршрутизации.
Компоненты:

- Balloon
- Hint
- behavior.RouteEditor
- control.RouteEditor
- geoObject.overlayFactory.interactive
- layout.storage
- layout.templateBased.Base
- route
- templateLayoutFactory

package.geoXml

Инструменты для работы с YMapsML, KML и GPX.
Компоненты:

- Balloon
- Hint
- geoObject.overlayFactory.interactive
- geoXml.load
- layout.storage
- layout.templateBased.Base
- templateLayoutFactory

package.editor

Редактор геообъектов.

Компоненты:

- Balloon
- Circle
- GeoObject
- GeoObjectArray
- GeoObjectCollection
- Hint
- Placemark
- Polygon
- Polyline
- Rectangle
- geoObject.overlayFactory.hotspot
- geoObject.overlayFactory.interactive
- geoObject.overlayFactory.interactiveGraphics
- geoObject.overlayFactory.staticGraphics
- geoObject.overlayFactory.storage
- geometry.Circle
- geometry.LineString
- geometry.Point
- geometry.Polygon
- geometry.Rectangle
- geometryEditor.LineString
- geometryEditor.Point
- geometryEditor.Polygon
- layout.Image
- layout.ImageWithContent
- layout.storage
- layout.templateBased.Base
- templateLayoutFactory

package.overlays

Оверлеи (HTML, графические, оверлеи активных областей).

Компоненты:

- geometry.pixel.Circle
- geometry.pixel.LineString
- geometry.pixel.Point
- geometry.pixel.Polygon
- geometry.pixel.Rectangle
- overlay.hotspot.Circle
- overlay.hotspot.Placemark
- overlay.hotspot.Polygon
- overlay.hotspot.Polyline
- overlay.hotspot.Rectangle
- overlay.html.Balloon
- overlay.html.Label
- overlay.html.Placemark
- overlay.html.Rectangle
- overlay.interactiveGraphics.Circle
- overlay.interactiveGraphics.Placemark
- overlay.interactiveGraphics.Polygon
- overlay.interactiveGraphics.Polyline
- overlay.interactiveGraphics.Rectangle
- overlay.staticGraphics.Circle
- overlay.staticGraphics.Placemark
- overlay.staticGraphics.Polygon
- overlay.staticGraphics.Polyline
- overlay.staticGraphics.Rectangle

Balloon

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geoXml, package.geocode, package.route, package.search, package.standard.

Расширяет **IDomEventEmitter**.

Балун — всплывающее облако, которое может отображать любое HTML-содержимое. Балуны возвращаются из методов open менеджеров балуна (карты, геообъектов, слоёв активных областей и т.д.). Не создавайте их самостоятельно, если в этом нет необходимости.
Конструктор | Поля | События | Методы

Конструктор

```
Balloon(map, data, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
map		**Map**	✔	Ссылка на объект карты.
data		Object		Данные.
options		Object		Опции.

Параметр	Свойства параметра	Тип	*	Описание
	autoPan	Boolean		Сдвигать карту, чтобы отобразить открывшийся балун. Значение по умолчанию: true
	autoPanDuration	Number		Длительность перемещения к точке балуна (в мс). Значение по умолчанию: 500
	autoPanMargin	Number		Отступ от угла viewport-а карты при действии autoPan.
	closeButton	Boolean		Флаг наличия кнопки закрытия. Значение по умолчанию: true
	closeButtonLayout	Function|String		Макет кнопки закрытия. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета).
	contentBodyLayout	Function|String		Макет для основной части содержимого балуна. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета).
	contentFooterLayout	Function|String		Макет для нижней части содержимого балуна. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета).
	contentHeaderLayout	Function|String		Макет для заголовка содержимого балуна. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета).
	contentLayout	Function|String		Макет для содержимого балуна. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета).
	interactivityModel	String		Ключ модели интерактивности. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**.

Параметр	Свойства параметра	Тип	*	Описание
	layout	Function|String		Внешний макет балуна. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета).
	maxHeight	Number		Максимальная высота, в пикселах.
	maxWidth	Number		Максимальная ширина, в пикселах.
	minHeight	Number		Минимальная высота, в пикселах.
	minWidth	Number		Минимальная ширина, в пикселах.
	offset	Number[]		Дополнительное смещение позиции относительно точки привязки.
	projection	**IProjection**		Проекция координат в глобальные пиксели.
	shadow	Boolean		Флаг наличия тени. Значение по умолчанию: true
	shadowLayout	Function|String		Макет тени. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета).
	zIndex	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
events	**IEventManager**	Менеджер событий. Унаследовано от **IDomEventEmitter**.

События

Имя	Описание
autopanbegin	Начало автоматического перемещения центра карты, вызванного методом autoPan. Экземпляр класса **Event**.
autopanend	Окончание автоматического перемещения центра карты, вызванного методом autoPan. Экземпляр класса **Event**.

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
close	Закрытие балуна. Экземпляр класса **Event**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
open	Открытие балуна. Экземпляр класса **Event**.
userclose	Закрытие балуна пользователем. Экземпляр класса **Event**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
autoPan()	**Balloon**	Автоматически смещает центр карты таким образом, чтобы балун оказался в видимой области.
close()	**Balloon**	Закрывает балун.
getData()	Object	Возвращает данные балуна.
getOverlay()	**overlay.html.Balloon**	Возвращает ссылку на оверлей балуна.
getPosition()	Number[]	Возвращает координаты балуна.
isOpen()	Boolean	Возвращает состояние балуна: открыт/закрыт.
open(position, preventAutoPan)	**Balloon**	Открывает балун на карте в точке с заданными координатами. Если балун уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией **IProjection**, которая содержится в опциях.
setData(data)	**Balloon**	Устанавливает данные балуна.
setPosition(position, preventAutoPan)	**Balloon**	Устанавливает новую позицию балуна. Формат и суть координат определяется проекцией **IProjection**, которая содержится в опциях.

Описание событий

autopanbegin

```
autopanbegin
```

Начало автоматического перемещения центра карты, вызванного методом autoPan. Экземпляр класса **Event**.

autopanend

```
autopanend
```

Окончание автоматического перемещения центра карты, вызванного методом autoPan. Экземпляр класса **Event**.

close

```
close
```

Закрытие балуна. Экземпляр класса **Event**.

open

```
open
```

Открытие балуна. Экземпляр класса **Event**.

userclose

```
userclose
```

Закрытие балуна пользователем. Экземпляр класса **Event**.

Описание методов

autoPan

```
{Balloon} autoPan()
```

Автоматически смещает центр карты таким образом, чтобы балун оказался в видимой области.

Возвращает ссылку на себя.

close

```
{Balloon} close()
```

Закрывает балун.

Возвращает ссылку на себя.

getData

```
{Object} getData()
```

Возвращает данные балуна.

getOverlay

```
{overlay.html.Balloon} getOverlay()
```

Возвращает ссылку на оверлей балуна.

getPosition

```
{Number[]} getPosition()
```

Возвращает координаты балуна.

isOpen

```
{Boolean} isOpen()
```

Возвращает состояние балуна: открыт/закрыт.

Возвращает true, если балун открыт, false — в обратном случае.

Пример:

```
// Открываем балун при двойном щелчке на карте
// или просто обновляем содержимое (случайное число),
// если балун уже был открыт.
var balloon = null;

myMap.events.add('dblclick', function () {
    if (balloon && balloon.isOpen()) {
        balloon.setData({ content: Math.random() });
    } else {
        balloon = myMap.balloon.open(myMap.getCenter(), Math.random());
    }
});
```

open

```
{Balloon} open(position, preventAutoPan)
```

Открывает балун на карте в точке с заданными координатами. Если балун уже открыт, перемещает его в указанную точку. Формат и суть координат определяется проекцией **IProjection**, которая содержится в опциях.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	Описание
position	Number[]	Координаты балуна.
preventAutoPan	Boolean	Флаг, отключающий автоматический сдвиг карты для данного вызова. Значение по умолчанию: false

setData

```
{Balloon} setData(data)
```

Устанавливает данные балуна.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
data	Object	✔	Данные.

* Обязательный параметр/опция.

setPosition

```
{Balloon} setPosition(position, preventAutoPan)
```

Устанавливает новую позицию балуна. Формат и суть координат определяется проекцией **IProjection**, которая содержится в опциях.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
position	Number[]	✔	Новые координаты.
preventAutoPan	Boolean		Флаг, отключающий автоматический сдвиг карты для данного вызова. Значение по умолчанию: false

* Обязательный параметр/опция.

behavior

behavior.DblClickZoom

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IBehavior**.

Поведение "масштабирование карты двойным щелчком кнопки мыши".

Конструктор | Поля | События | Методы

Конструктор

```
behavior.DblClickZoom(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции.
	centering	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. Значение по умолчанию: true
	duration	Number	Время анимации масштабирования двойным щелком кнопки мыши (0 — анимации нет). Значение по умолчанию: 200

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**.

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • 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**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от **IBehavior**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

behavior.Drag

Входит в пакеты package.controls, package.full, package.map, package.standard.

Расширяет **IBehavior**.

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

Конструктор | Поля | События | Методы

Конструктор

```
behavior.Drag(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
`options`		Object	Опции.
	`actionCursor`	String	Курсор для поведения **behavior.Drag** при перетаскивании карты. Значение по умолчанию: 'grabbing'

Параметр	Свойства параметра	Тип	Описание
	`cursor`	String	Курсор для поведения **behavior.Drag** при наведении на карту. Значение по умолчанию: 'grab'
	`inertia`	Boolean	Включает кинетическую инерцию движения в конце перетаскивания. Значение по умолчанию: true
	`inertiaDuration`	Number	Продолжительность инерции в мс. Значение по умолчанию: 400
	`tremor`	Integer	Минимальное смещение курсора после нажатия кнопки мыши, после которого начинает двигаться карта. Значение по умолчанию: 2

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**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • 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**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от **IBehavior**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

behavior.LeftMouseButtonMagnifier

Входит в пакеты package.controls, package.full, package.map, package.standard.

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	Свойства параметра	Тип	Описание
`options`		Object	Опции.
	`actionCursor`	String	Курсор во время выбора области для увеличения при включенном поведении **behavior.LeftMouseButtonMagnifier**. Значение по умолчанию: 'crosshair'
	`cursor`	String	Курсор при включенном поведении **behavior.LeftMouseButtonMagnifier**. Значение по умолчанию: 'zoom'
	`duration`	Number	Продолжительность анимации увеличения области при использовании поведения **behavior.LeftMouseButtonMagnifier**, мс.

Параметр	Свойства параметра	Тип	Описание
			Значение по умолчанию: 300

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**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • 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**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от **IBehavior**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

behavior.MultiTouch

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IBehavior**.

Поведение "масштабирование карты мультисенсорным касанием".

Конструктор | Поля | События | Методы

Конструктор

```
behavior.MultiTouch(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции.
	tremor	Number	Минимальное движение по экрану устройства (в пикселях), при котором сработает поведение **behavior.MultiTouch**. Значение по умолчанию: 2

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**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • 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**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от **IBehavior**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

behavior.RightMouseButtonMagnifier

Входит в пакеты package.full, package.map, package.standard.

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	Свойства параметра	Тип	Описание
`options`		Object	Опции.
	`actionCursor`	String	Курсор во время выбора области для увеличения при включенном поведении **behavior.RightMouseButtonMagnifier**. Значение по умолчанию: 'crosshair'
	`duration`	Number	Продолжительность анимации при выделении области при помощи поведения **behavior.RightMouseButtonMagnifier**, мс. Значение по умолчанию: 300

Поля

Имя	Тип	Описание
events	**IEventManager**	Менеджер событий.

Имя	Тип	Описание
		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**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • 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**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от **IBehavior**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

behavior.RouteEditor

Входит в пакеты package.full, package.route.

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.
enable	Включение поведения. Унаследовано от IBehavior.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
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.

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение. Унаследован от IBehavior.
enable()		Включает поведение. Унаследован от IBehavior.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от IBehavior.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание событий

routechange

```
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.

behavior.Ruler

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **IBehavior**.

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

Конструктор | Поля | События | Методы

Конструктор

```
behavior.Ruler()
```

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**.
enable	Включение поведения. Унаследовано от **IBehavior**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель.

Имя	Описание
	Унаследовано от **IChild**.

Методы

Имя	Возвращает	Описание
close()		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**	Возвращает ссылку на родительский объект. 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**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.
setState(state)		Восстанавливает состояние линейки из закодированной строки. Смотрите behavior.Ruler.getState.

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

```
close()
```

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

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	Тип	*	Описание
state	String	✔	Закодированное состояние линейки.

* Обязательный параметр/опция.

behavior.ScrollZoom

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IBehavior**.

Поведение "масштабирование карты колесом мыши".

Конструктор | Поля | События | Методы

Конструктор

```
behavior.ScrollZoom(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции.
	maximumDelta	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). Значение по умолчанию: 5

Параметр	Свойства параметра	Тип	Описание
	`speed`	Number	Скорость увеличения/уменьшения масштаба карты при включенном поведении **behavior.ScrollZoom**, уровней масштаба в секунду. Значение по умолчанию: 5

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**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • 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**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
isEnabled()	Boolean	Проверяет, включено ли поведение. Унаследован от **IBehavior**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект.

Имя	Возвращает	Описание
		Унаследован от **IChildOnMap**.

behavior.storage

Входит в пакеты package.full, package.map, package.standard.

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" — перемещание 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 двойным касанием (например, пальцами 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.

Circle

Входит в пакеты package.editor, package.full, package.geoObjects.

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.

См.: GeoObject geometry.Circle

Параметры:

Параметр	Тип	*	Описание
geometry	**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.
properties	Object|**IDataManager**		Данные геообъекта. Соответствуют данным класса **GeoObject**.
options	Object		Опции геообъекта. Соответствуют опциям класса **GeoObject**.

* 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	Null	Редактор геометрии "Круг" на данный момент не реализован.
events	**event.Manager**	Менеджер событий. Унаследовано от **GeoObject**.
geometry	**geometry.Circle**	Геометрия типа "Круг".
hint	**geoObject.Hint**	Хинт геообъекта. Унаследовано от **GeoObject**.

Имя	Тип	Описание
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: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что 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**. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса **Balloon**. Унаследовано от **GeoObject**.
beforedrag	Событие, предшествующее событию drag. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 drag будет отменено. Унаследовано от **GeoObject**.
beforedragstart	Событие, предшествующее событию dragstart. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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. Унаследовано от **GeoObject**.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
drag	Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге. Унаследовано от **GeoObject**.
dragend	Завершение перетаскивания геообъекта. Экземпляр класса **Event**. Унаследовано от **GeoObject**.
dragstart	Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • domEvent — исходное DOM-событие (в виде объекта **DomEvent**), если оно имеется. Унаследовано от **GeoObject**.
editorstatechange	Изменение состояния редактора геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие редактора геометрии. Унаследовано от **GeoObject**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Имя	Описание
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от **IGeoObject**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от **IGeoObject**.
propertieschange	Изменение данных геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от **IGeoObject**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getMap()	**Map**	Возвращает ссылку на карту.

Имя	Возвращает	Описание
		Унаследован от **IParentOnMap**.
getOverlay()	**IOverlay**|null	Возвращает оверлей геообъекта, либо null. Унаследован от **IGeoObject**.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

Описание полей

editor

```
Null editor
```

Редактор геометрии "Круг" на данный момент не реализован.

geometry

```
geometry.Circle geometry
```

Геометрия типа "Круг".

cluster

cluster.Balloon

Менеджер балуна кластера.

Конструктор | Поля | События | Методы

Конструктор

```
cluster.Balloon(cluster)
```

Параметры:

Параметр	Тип	*	Описание
cluster	IGeoObject	✔	Объект-кластер.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	event.Manager	Менеджер событий балуна геообъекта.

События

Имя	Описание
close	Произошло закрытие балуна.
open	Произошло открытие балуна.

Методы

Имя	Возвращает	Описание
close()		Закрывает балун.
isOpen()	Boolean	Проверяет, открыт ли балун в данный момент.
open(anchorPixels)	Balloon	Открывает балун.

Описание полей

events

```
event.Manager events
```

Менеджер событий балуна геообъекта.

Описание событий

close

```
close
```

Произошло закрытие балуна.

open

```
open
```

Произошло открытие балуна.

Описание методов

close

```
close()
```

Закрывает балун.

isOpen

```
{Boolean} isOpen()
```

Проверяет, открыт ли балун в данный момент.

Возвращает true, если балун открыт, false — иначе.

open

```
{Balloon} open(anchorPixels)
```

Открывает балун.

Возвращает экземляр балуна.

Параметры:

Параметр	Тип	Описание
anchorPixels	Number[]	Точка привязки балуна в глобальных пиксельных координатах. По умолчанию используются текущие координаты центра карты.

Cluster

Расширяет **IGeoObject**.

Кластер геообъектов. Используется по умолчанию в **Clusterer**.

Конструктор | Поля | События | Методы

Конструктор

```
Cluster(geometry, properties, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		**IPointGeometry**	✔	Геометрия кластера.
properties		**IDataManager**	✔	Данные кластера.
	geoObjects	**IGeoObject**[]	✔	Массив геообъектов, которые находятся в данном кластере.
options		Object		Опции кластера. Помимо частных опций, балун кластера поддерживает те же опции, что и {@link.Balloon}. Опции для балуна кластера указываются с префиксом 'balloon'.

Параметр	Свойства параметра	Тип	*	Описание
	`balloonContentBodyLayout`	Function|String		Макет балуна кластера. Можно передать конструктор объекта с интерфейсом **ILayout** или ключ одного из стандартных макетов. Каждый стандартный макет имеет некоторые собственные опции. • 'cluster#balloonContentBody' — Макет с двумя колонками. В левой колонке (sidebar) находится список геообъектов. В правой колонке информация о геообъекте. **Опции макета** • balloonMainContentLayout Макет отображения данных о геообъекте в балуне кластера. (Тип: конструктор объекта

Параметр	Свойства параметра	Тип	*	Описание
				с интерфейсом **ILayout** или ключ макета) Значение по умолчанию: 'cluster#balloonMainContent' • balloonSidebarWidth Ширина поля со списком объектов в балуне кластера в пикселях. Значение по умолчанию: 125 • 'cluster#balloonCarouselContent' — Макет с тремя колонками. В центральной колонке представлена информация о геобъекте. В боковых колонках располагаются стрелки перехода на следующий и предыдущий геообъект. В нижней части балуна находится меню навигации (pager). **Опции макета** • balloonContentItemLayout Макет отображения данных геообъекта в балуне кластера. (Тип: конструктор объекта

Параметр	Свойства параметра	Тип	*	Описание

				с интерфейсом **ILayout** или ключ макета) Значение по умолчанию: 'cluster#balloonCarouselContentItem'
				• balloonCycling Цикличность списка при навигации боковыми стрелками. Значение по умолчанию: true
				• balloonPagerSize Количество элементов навигации в нижней панели. Значение по умолчанию: 9
				• balloonPagerType Тип нижней панели навигации. Может принимать значения 'numeric' и 'marker'.
				  • numeric — отображение номера геообъекта в списке;
				  • marker — отображение маркеров без номеров. Рекомендуется

Параметр	Свойства параметра	Тип	*	Описание
				использовать при количестве элементов в кластере меньшем или равном значению options.balloonPagerSize. Значение по умолчанию: numeric • balloonPagerVisible Отображать ли панель навигации. Значение по умолчанию: true • 'cluster#balloonAccordionContent' — Информация о геообъектах в виде списка. После клика по элементу под ним разворачивается информация о геообъекте. **Опции макета** • balloonAccordionItemTitleLayout Макет заголовка элемента списка. По клику разворачивается детальная информация о геообъекте. (Тип: конструктор объекта

Параметр	Свойства параметра	Тип	*	Описание
				с интерфейсом **ILayout** или ключ макета) Значение по умолчанию: 'cluster#balloonAccordionItemTitle' • balloonAccordionItemContentLayout Макет информации о геообъекте. Отображается после клика по элементу списка. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета) Значение по умолчанию: 'cluster#balloonAccordionItemContent' • balloonAccordionShowIcons Опция, которая определяет отображается ли иконка геообъекта в макете 'cluster#balloonAccordionItemTitle'. Значение по умолчанию: true Значение по умолчанию: 'cluster#balloonContentBody'

Параметр	Свойства параметра	Тип	*	Описание
	balloonHeight	Number		Высота балуна кластера. Стандартное значение зависит от макета. • 'cluster#balloonContentBody' — 210 пикселей • 'cluster#balloonCarouselContent' — 177 пикселей • 'cluster#balloonAccordionContent' — 283 пикселей
	balloonWidth	Number		Ширина балуна кластера. Стандартное значение зависит от макета. • 'cluster#balloonContentBody' — 475 пикселей • 'cluster#balloonCarouselContent' — 308 пикселей • 'cluster#balloonAccordionContent' — 305 пикселей Обратите внимание, что если ширина балуна кластера будет больше максимально допустимой ширины балуна (опция maxWidth **Balloon**), то макет будет обрезан.
	cursor	String		Курсор над меткой кластера. Значение по умолчанию: 'pointer'
	disableClickZoom	Boolean		Флаг, запрещающий увеличение коэффициента масштабирования карты при клике на кластер. Значение по умолчанию: false
	hideIconOnBalloonOpen	Boolean		Скрывать иконку при открытии балуна. Значение по умолчанию: true

Параметр	Свойства параметра	Тип	*	Описание
	iconContentLayout	Function|String		Макет содержимого метки кластера. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета). Значение по умолчанию: 'cluster#iconContent'
	iconLayout	Function|String		Макет метки кластера (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета). Значение по умолчанию: 'cluster#icon'
	icons	Object[]		Массив, описывающий иконки для стандартной реализации кластера.
	interactivityModel	String		Модель интерактивности кластера. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**. Значение по умолчанию: 'default#geoObject'
	numbers	Number[]		Массив, описывающий граничные значения для размеров кластеров. Значение по умолчанию: [10, 100]
	openBalloonOnClick	Boolean		Опция, позволяющая запретить открытие балуна при клике на кластере. По умолчанию открытие балуна разрешено. Количество иконок, описанных в опции "icons", должно быть на 1 больше, чем чисел в данном массиве. Значение по умолчанию: true

* 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]
 }],
 // При размере кластера до 100 будет использована картинка 'small.jpg'.
 // 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 ym.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"
    clusterBalloonContentBodyLayout: "cluster#balloonCarouselContent",
    // Запрещаем зацикливание списка при постраничной навигации
    clusterBalloonCycling: false,
    // Настройка внешнего вида панели навигации. Элементами панели навигации
будут маркеры.
    clusterBalloonPagerType: "marker",
    // Количество элементов в панели навигации
    clusterBalloonPagerSize: 6
});
```

3.

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

Поля

Имя	Тип	Описание
balloon	**cluster.Balloon**	Балун кластера.
events	**IEventManager**	Менеджер событий. Унаследовано от **IDomEventEmitter**.
geometry	**IGeometry**	Геометрия геообъекта. Унаследовано от **IGeoObject**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.

Имя	Тип	Описание
properties	IDataManager	Менеджер данных геообъекта. Унаследовано от IGeoObject.
state	data.Manager	Состояние кластера. Определяется следующими полями: • activeObject — Ссылка на активный объект кластера. Активным объектом является тот, который в данный момент выбран в балуне кластера.

События

Имя	Описание
addtomap	Добавление кластера на карту. Экземпляр класса Event.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от IGeoObject.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Имя	Описание
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от **IGeoObject**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от **IGeoObject**.
propertieschange	Изменение данных геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от **IGeoObject**.
removefrommap	Удаление кластера с карты. Экземпляр класса **Event**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getBounds()	Number[][]|null	Возвращает географические координаты прямоугольной области, охватывающей все элементы кластера.
getGeoObjects()	IGeoObject[]	Возвращает массив геообъектов, образующих кластер.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getOverlay()	IOverlay|null	Возвращает оверлей геообъекта, либо null. Унаследован от IGeoObject.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание полей

balloon

```
cluster.Balloon balloon
```

Балун кластера.

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 событий

addtomap

```
addtomap
```

Добавление кластера на карту. Экземпляр класса Event.

removefrommap

```
removefrommap
```

Удаление кластера с карты. Экземпляр класса Event.

Описание методов

getBounds

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

Возвращает географические координаты прямоугольной области, охватывающей все элементы кластера.

getGeoObjects

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

Возвращает массив геообъектов, образующих кластер.

Clusterer

Входит в пакеты package.clusters, package.full.

Расширяет **ICollection**.

Кластеризатор геообъектов.

Конструктор | Поля | События | Методы

Конструктор

```
Clusterer(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции. Опции для дочерних объектов-кластеров задаются с префиксом cluster.
	gridSize	Number	Размер ячейки кластера в пикселях. Значение по умолчанию: 64
	groupByCoordinates	Boolean	Специальный режим работы кластеризатора при котором кластеры образуются только из геобъектов с одинаковыми координатами. Значение по умолчанию: false

Параметр	Свойства параметра	Тип	Описание
	`margin`	Number|Number[]	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 горизонтальные и вертикальные отступы соответственно. Если задан массив из 4х чисел, то это отступы top, right, bottom, left. Значение по умолчанию: 10
	`maxZoom`	Number[]	Максимальный коэффициент масштабирования карты, на котором происходит кластеризация объектов. Даже если кластеризация отключена, будут показаны только объекты в видимой области карты. Значение по умолчанию: 23
	`minClusterSize`	Number	Минимальное количество объектов, образующих кластер. Значение по умолчанию: 2
	`preset`	String	Ключ предустановленных опций кластеризатора. Список ключей, доступных в пакете package.clusters, содержится в описании **option.presetStorage**.
	`showInAlphabeticalOrder`	Boolean	Показывать метки в балуне в алфавитном порядке при нажатии на кластер. Геообъекты кластера сортируются по специальным полям в данных этих геообъектов — clusterCaption (или balloonContentHeader, если предыдущее поле не определено). По умолчанию геообъекты показываются в порядке добавления в кластеризатор. Значение по умолчанию: false

Параметр	Свойства параметра	Тип	Описание
	synchAdd	Boolean	Добавлять объекты на карту сразу. По умолчанию объекты добавляются на карту группами через определенные интервалы времени. Это делается для того, чтобы медленные браузеры справлялись с нагрузкой. Значение по умолчанию: false
	zoomMargin	Number|Number[]	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 top, 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 top, right, bottom, left. Значение по умолчанию: 0

Примеры:

1.

```
// создание кластеризатора
// создадим карту, на которой необходимо кластеризовать геообъекты
var map = new ymaps.Map('mapsID', {center: [56.034, 36.992], zoom: 8}),
// создадим массив геообъектов
myGeoObjects = [];
myGeoObjects[0] = new ymaps.GeoObject({
    geometry: {type: "Point", coordinates: [56.034, 36.992]},
    properties: {
        clusterCaption: 'Геообъект №1',
        balloonContentBody: 'Содержимое балуна геообъекта №1.'
    }
});
myGeoObjects[1] = new ymaps.GeoObject({
    geometry: {type: "Point", coordinates: [56.021, 36.983]},
```

```
    properties: {
        clusterCaption: 'Геообъект №2',
        balloonContentBody: 'Содержимое балуна геообъекта №2.'
    }
});

// создадим кластеризатор и запретим приближать карту при клике на кластеры
clusterer = new ymaps.Clusterer({clusterDisableClickZoom: true});
clusterer.add(myGeoObjects);
map.geoObjects.add(clusterer);
```

2.

```
// Открытие балуна кластера с выбранным объектом.

// Поскольку по умолчанию объекты добавляются асинхронно,
// обработку данных можно делать только после события, сигнализирующего об
// окончании добавления объектов на карту.
cluster.events.add('objectsaddtomap', function () {

    // Получим данные о состоянии объекта внутри кластера.
    var geoObjectState = cluster.getObjectState(myGeoObjects[1]);
    // Проверяем, находится ли объект находится 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]);
            geoObjectState.cluster.balloon.open();

        } else {
            // Если объект не попал в кластер, открываем его собственный балун.
            myGeoObjects[1].balloon.open();
        }
    }

});
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter.
options	option.Manager	Менеджер опций кластера.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от ICollection.
objectsaddtomap	Окончание добавления объектов на карту при асинхронной обработке. Экземпляр класса Event. Имена свойств, доступных через метод get: • 'type' — тип события • 'target' — ссылка на кластеризатор
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection.

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 кластеризатора.
getIterator()	**IIterator**	Возвращает итератор по коллекции. Унаследован от **ICollection**.
getObjectState(geoObject)	Object	Функция для получения информации о текущем состоянии объекта, добавленного в кластеризатор.
refresh()	**Clusterer**	Функция, которая заново генерирует кластеры из геообъектов и отображает их на карте.
remove(objects)	**Clusterer**	Удаляет геообъекты из кластеризатора.
removeAll()	**Clusterer**	Удаляет все геообъекты из кластеризатора.

Описание полей

options

```
option.Manager options
```

Менеджер опций кластера.

Пример:

```
clusterer.options.set({
    gridSize: 100,
    disableClickZoom: true
});
```

Описание событий

objectsaddtomap

```
objectsaddtomap
```

Окончание добавления объектов на карту при асинхронной обработке. Экземпляр класса **Event**. Имена свойств, доступных через метод get:

- 'type' — тип события
- 'target' — ссылка на кластеризатор

Описание методов

add

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

Добавляет геообъект или массив геообъектов в кластеризатор.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
objects	IGeoObject|IGeoObject[]	✔	Массив геообъектов или единичный геообъект.

* Обязательный параметр/опция.

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.

Параметры:

Параметр	Тип	*	Описание
center	Number[]	✔	Центр кластера в геокоординатах.
geoObjects	IGeoObject[]	✔	Массив плейсмарков в кластере.

* Обязательный параметр/опция.

getBounds

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

Возвращает географические координаты прямоугольной области, охватывающей все элементы кластеризатора.

getObjectState

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

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

Возвращает объект с полями:

- 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	Тип	*	Описание
geoObject	IGeoObject	✔	Гео-объект, для которого нужно получить состояние.

* Обязательный параметр/опция.

refresh

```
{Clusterer} refresh()
```

Функция, которая заново генерирует кластеры из геообъектов и отображает их на карте.

Возвращает ссылку на себя.

remove

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

Удаляет геообъекты из кластеризатора.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
objects	IGeoObject|IGeoObject[]	✔	Массив геообъектов.

* Обязательный параметр/опция.

removeAll

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

Удаляет все геообъекты из кластеризатора.

Возвращает ссылку на себя.

collection

collection.Item

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IChildOnMap**, **ICustomizable**, **IEventEmitter**, **IParentOnMap**.

Базовый класс элемента коллекции объектов карты.

Конструктор | Поля | События | Методы

Конструктор

```
collection.Item(options)
```

Параметры:

Параметр	Тип	Описание
options	Object	Опции обьекта.

Пример:

```
// Пример реализации собственного элемента управления на основе наследования от
collection.Item.
// Элемент управления отображает название объекта, который находится в центре
карты.
var bind = ymaps.util.bind,
    map = new ymaps.Map(
        'map', {
            center: [55.819543, 37.611619],
            zoom: 6
        }
    ),
// Создаем собственный класс.
    CustomControlClass = function(options) {
        CustomControlClass.superclass.constructor.call(this, options);
        this._$content = null;
        this._geocoderPromise = null;
    };
// И наследуем его от collection.Item.
ymaps.util.augment(CustomControlClass, ymaps.collection.Item, {
    onAddToMap: function (map) {
        CustomControlClass.superclass.onAddToMap.call(this, map);
        // Создаем HTML-элемент с текстом.
        var parentDomContainer = this.getParent().getChildElement(this);
        this._$content = $('<div class="customControl"></
div>').appendTo(parentDomContainer);
        this._mapEventGroup = this.getMap().events.group();
        // Запрашиваем данные после изменения положения карты.
        this._mapEventGroup.add('boundschange', this._createRequest, this);
        // Сразу же запрашиваем название места.
        this._createRequest();
    },

    onRemoveFromMap: function () {
        this._$content.remove();
        this._mapEventGroup.removeAll();
        this._stopPendingRequest();
        CustomControlClass.superclass.onRemoveFromMap.call(this);
    },

    _createRequest: function() {
        // Отменяем ожидание предыдущих данных.
        this._stopPendingRequest();
```

```
        // Запрашиваем информацию о месте по координатам центра карты.
        this._geocoderPromise =
ymaps.geocode(this.getMap().getCenter().join(','), {
            // Указываем, что ответ должен быть в формате JSON.
            json: true,
            // Устанавливаем лимит на кол-во записей в ответе.
            results: 1
        }).then(
            bind(this._onServerResponse, this)
        );
    },

    _stopPendingRequest: function () {
        if (this._geocoderPromise) {
            this._geocoderPromise.reject();
            this._geocoderPromise = null;
        }
    },

    _onServerResponse: function(result) {
        // Данные от сервера были получены и теперь их необходимо отобразить.
        // Описание ответа в формате JSON.
        var members = result.GeoObjectCollection.featureMember,
            geoObjectData = (members && members.length) ? members[0].GeoObject :
null;
        if (geoObjectData) {
            this._
$content.text(geoObjectData.metaDataProperty.GeocoderMetaData.text);
        }
    }
});

var customControl = new CustomControlClass();
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: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.

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()		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	Тип	*	Описание
map	Map	✔	Объект карты, на которую добавлен элемент.

* Обязательный параметр/опция.

onRemoveFromMap

```
onRemoveFromMap()
```

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

setParent

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

Устанавливает родителя для заданного элемента коллекции.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
parent	IParentOnMap	✔	Родительский объект.

* Обязательный параметр/опция.

Collection

Входит в пакеты package.full, package.map, package.standard.

Расширяет **ICollection**, **collection.Item**.

Базовая реализация коллекции объектов на карте.

Конструктор | Поля | События | Методы

Конструктор

```
Collection(options)
```

Параметры:

Параметр	Тип	Описание
options	Object	Опции коллекции.

Пример:

```
// Пример реализации собственного меню на основе наследования от Collection.
var bind = ymaps.util.bind,
    map = new ymaps.Map(
        'map', {
            center: [55.819543, 37.611619],
            zoom: 6
        }
    ),
// Создаем собственный класс дочернего элемента меню.
    MenuItem = function (data, options) {
        MenuItem.superclass.constructor.call(this, options);
        // Создаем менеджер данных.
        // Если первым аргументом передали строчку, то обрачиваем её в хеш.
       this._data = new ymaps.data.Manager((typeof data == 'string') ? { content:
data } : data);
        this._$content = null;
    };

// Наследуем MenuItem от collection.Item.
ymaps.util.augment(MenuItem, ymaps.collection.Item, {
    onAddToMap: function (map) {
        MenuItem.superclass.onAddToMap.call(this, map);
        var parentDomContainer = this.getParent().getChildElement(this);
        this._$content = $('<div class="menuItem"></
div>').appendTo(parentDomContainer);
        // Создаем слушателя изменения поля 'content' в _data.
        this._dataMonitor = new ymaps.Monitor(this._data);
        this._dataMonitor.add('content', this._applyContent, this);
        this._applyContent(this._data.get('content', ''));
        // Пример использования менеджера событий domEvent.manager.
        this._eventsGroup = ymaps.domEvent.manager.group(this._$content[0]);
        this._eventsGroup.add('click', function () {
            this.events.fire('click');
        }, this);
    },

    onRemoveFromMap: function () {
        this._$content.remove();
        this._dataMonitor.removeAll();
        this._eventsGroup.removeAll();
        MenuItem.superclass.onRemoveFromMap.call(this);
    },

    getData: function () {
        return this._data;
```

```
    },

    _applyContent: function (newValue) {
        this._$content.text(newValue);
    }
});

// Создаем собственный класс меню.
var Menu = function (options) {
    Menu.superclass.constructor.call(this, options);
    this._$content = null;
};

// Наследуем Menu от Collection.
ymaps.util.augment(Menu, ymaps.Collection, {
    onAddToMap: function (map) {
        Menu.superclass.onAddToMap.call(this, map);
        var parentDomContainer = this.getParent().getChildElement(this);
        this._$content = $('<div class="menu"></
div>').appendTo(parentDomContainer);
    },

    onRemoveFromMap: function () {
        this._$content.remove();
        Menu.superclass.onRemoveFromMap.call(this);
    },

    getChildElement: function (chilElement) {
        return this._$content;
    }
});

// Пример использования пользовательского элемента управления "Меню".
var menu = new Menu(),
    menuItem1 = new MenuItem('Пункт 1'),
    menuItem2 = new MenuItem({content: 'Второй пункт'}),
    menuItem3 = new MenuItem('Третий пункт');

menu.add(menuItem1);
menu.add(menuItem2);
menu.add(menuItem3);
map.controls.add(menu, {top: 5, left: 5});
// Изменяем данные в первом пункте меню.
menuItem1.getData().set('content', 'Первый пункт');
// Удаляем последний пункт из меню.
menu.remove(menuItem3);

// Можно прослушивать события не каждого отдельного элемента, а сразу всей
коллекции.
menu.events.add('click', function (event) {
    // Запрашиваем порядковый номер элемента, который проинициировал событие.
    console.log('menu item #' + menu.indexOf(event.get('target')));
});
```

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: • child — добавленный дочерний элемент.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
remove	Произошло удаление дочернего объекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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 диапозона номеров коллекции.
getAll()	Object[]	Возвращает массив со всеми элементами коллекции.
getIterator()	**IIterator**	Возвращает итератор для перебора элементов коллекции.
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**.

Имя	Возвращает	Описание
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()		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 из коллекции.
setParent(parent)	**collection.Item**	Устанавливает родителя для заданного элемента коллекции. Унаследован от **collection.Item**.

Описание событий

add

```
add
```

Произошло добавление дочернего объекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- child — добавленный дочерний элемент.

remove

```
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	Тип	*	Описание
child	**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	Тип	*	Описание
callback	Function	✔	Функция-обработчик. На вход получает элемент коллекции. При возвращении функцией значения false перебор останавливается.
context	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	Тип	*	Описание
filterFunction	Function	✔	Функция, которая является фильтром для объектов коллекции. Первым параметром принимает элемент из коллекции. Должна возвращать булевое значение.

* Обязательный параметр/опция.

get

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

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

Параметры:

Параметр	Тип	*	Описание
index	Number	✔	Порядковый номер элемента в коллекции.

* Обязательный параметр/опция.

getAll

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

Возвращает массив со всеми элементами коллекции.

getIterator

```
{IIterator} 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	Тип	*	Описание
childToFind	Object	✔	Искомый объект.

* Обязательный параметр/опция.

remove

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

Удаляет элемент из коллекции.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
child	collection.Item	✔	Элемент, который необходимо удалить.

* Обязательный параметр/опция.

removeAll

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

Удаляет все элементы из коллекции.

Возвращает ссылку на себя.

control

control.Button

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **ICustomizable**, **ISelectableControl**.

Элемент управления "Кнопка". Добавляется в тулбар.

Конструктор | Поля | События | Методы

Конструктор

```
control.Button(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
params		Object|String	Параметры кнопки или строка — содержимое кнопки в виде HTML.
	data	Object	Данные кнопки.
	data.content	String	Содержимое кнопки в виде HTML.
	data.image	String	Иконка кнопки, если есть.
	data.imageDisabled	String	Иконка для "отключенного" состояния кнопки.
	data.imageSelected	String	Иконка для "выбранного" состояния кнопки.
	data.title	String	Текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.
options		Object	Опции кнопки.
	layout	Function|String	Макет элемента управления. В конструктор макета передается объект, содержащий поля: • 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**.

Параметр	Свойства параметра	Тип	Описание
			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. (Тип: конструктор объекта с интерфейсом **ISelectableControlLayout** или ключ макета).
	`minWidth`	Number	Минимальная ширина кнопки. Значение по умолчанию: 0
	`position`	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, left: 5 }

Параметр	Свойства параметра	Тип	Описание
	selectOnClick	Boolean	Опция, описывающая поведение кнопки. • true — кнопка становится "нажатой" после клика; • false — кнопка не меняет свой внешний вид после клика на нее. Значение по умолчанию: true
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Примеры:

1.

```
// Пример 1.
// Создание кнопки и добавление ее на карту.
var button = new ymaps.control.Button({
         data: {
             // Зададим иконку для кнопки
             image: 'images/button.jpg',
             // Текст на кнопке.
             content: 'Сохранить',
             // Текст всплывающей подсказки.
             title: 'Нажмите для сохранения маршрута'
         }
    }, {
        // Зададим опции для кнопки.
        selectOnClick: false
    });
map.controls.add(button, {top: 5, right: 5});
```

2.

```
// Пример 2.
// Создание кнопки с пользовательским макетом
var button = new ymaps.control.Button({
        data: {
            content: 'Красная кнопка',
            title: 'Нажмите на кнопку'
        }
    }, {
        layout: ymaps.templateLayoutFactory.createClass(
            // Если кнопка не нажата, к ней применится css-стиль 'myButton'
            // Если кнопка нажата, к ней применятся css-стили 'myButton' и
'myButtonSelected'.
            "<div class='myButton [if state.selected]myButtonSelected[endif]'
title='$[data.title]'>" +
            "$[data.content]" +
            "</div>"
        )
    });
map.controls.add(button);
```

Поля

Имя	Тип	Описание
data	data.Manager	Данные кнопки. Имена полей, доступных через метод data.Manager.get: • image — иконка кнопки, если есть; • imageDisabled — иконка для "отключенного" состояния кнопки; • imageSelected — для "выбранного" состояния кнопки; • content — содержимое кнопки в виде HTML; • title — текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.
state	data.Manager	Состояние кнопки. Имена полей, доступных через метод data.Manager.get: • selected — признак, нажата ли кнопка; • enabled — признак, активна ли кнопка; • pressed — признак того, что кнопка зажата пользователем.

События

Имя	Описание
click	Нажатие на кнопку. Экземпляр класса Event.
deselect	Элемент управления не выбран. Унаследовано от ISelectableControl.
disable	Элемент управления стал недоступным. Унаследовано от ISelectableControl.
enable	Элемент управления стал доступным. Унаследовано от ISelectableControl.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • 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.
getParent()	IControlParent	Возвращает ссылку на родительский объект. 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. Унаследован от IControl.

Описание полей

data

```
data.Manager data
```

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

- image — иконка кнопки, если есть;
- imageDisabled — иконка для "отключенного" состояния кнопки;
- imageSelected — для "выбранного" состояния кнопки;
- content — содержимое кнопки в виде HTML;
- title — текст всплывающей подсказки, которая появляется при наведении на кнопку курсора мыши.

state

```
data.Manager state
```

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

- selected — признак, нажата ли кнопка;
- enabled — признак, активна ли кнопка;
- pressed — признак того, что кнопка зажата пользователем.

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

```
click
```

Нажатие на кнопку. Экземпляр класса **Event**.

control.Group

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **IControl**, **IControlParent**, **ICustomizable**, **IMapObjectCollection**.

Группа (упорядоченная коллекция) элементов управления.

Конструктор | Поля | События | Методы

Конструктор

```
control.Group(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
params		Object|**IControl**[]	Настройки группы элементов управления. Если в качестве настроек передан массив элементов управления, он трактуется как список дочерних элементов.
	items	**IControl**[]	Список дочерних элементов.
options		Object	Опции.

Параметр	Свойства параметра	Тип	Описание
	`position`	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, left: 5 }
	`visible`	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные группы.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**data.Manager**	Состояние группы.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от **ICollection**.

Имя	Описание
childcontainerchange	Изменение дом-элемента, в который добавляются дочерние элементы группы. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newChildContainerElement — новый дом-элемент для дочерних элементов; • oldChildContainerElement — старый элемент для дочерних элементов.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection.

Методы

Имя	Возвращает	Описание
add(item, positionIndex)	control.Group	Добавляет элемент управления в группу. Элемент можно добавлять, передав экземпляр класса, реализующего интерфейс IControl, или по ключу. Возожные значения ключей: • "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" control.MapTools; • "miniMap" — обзорная карта control.MiniMap; • "scaleLine" — масштабный отрезок control.ScaleLine; • "searchControl" — панель поиска control.SearchControl; • "trafficControl" — панель пробок control.TrafficControl; • "typeSelector" — панель переключения типа карты control.TypeSelector; • "zoomControl" — ползунок масштаба control.ZoomControl; • "smallZoomControl" — простой ползунок масштаба control.SmallZoomControl.

Имя	Возвращает	Описание
each(callback, context)	control.Group	Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.
filter(callback)	IChildOnMap[]	Функция для выборки дочерних элементов по условию.
get(key)	IChildOnMap	Возвращает ссылку на объект контрола, добавленного по ключу или по индексу.
getAll()	IChildOnMap[]	Возвращает массив дочерних элементов контрола.
getChildElement(child)	HTMLElement	Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться. Унаследован от IControlParent.
getContainer()	HTMLElement	Возвращает контейнер, в который добавляются дочерние элементы контрола.
getIterator()	IIterator	Возвращает итератор по коллекции. Унаследован от ICollection.
getLength()	Number	Возвращает количество дочерних элементов.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getParent()	IControlParent	Возвращает ссылку на родительский объект. Унаследован от IControl.
indexOf(childToFind)	Integer	Возвращает индекс дочернего элемента в группе.
remove(item)	control.Group	Удаляет элемент управления из группы.
removeAll()	control.Group	Удаляет все дочерние элементы из группы.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IControl.

Описание полей

data

```
data.Manager data
```

Данные группы.

state

```
data.Manager state
```

Состояние группы.

Описание событий

childcontainerchange

```
childcontainerchange
```

Изменение дом-элемента, в который добавляются дочерние элементы группы. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- newChildContainerElement — новый дом-элемент для дочерних элементов;
- oldChildContainerElement — старый элемент для дочерних элементов.

Описание методов

add

```
{control.Group} add(item, positionIndex)
```

Добавляет элемент управления в группу. Элемент можно добавлять, передав экземпляр класса, реализующего интерфейс **IControl**, или по ключу.

Возожные значения ключей:

- "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**;
- "miniMap" — обзорная карта **control.MiniMap**;
- "scaleLine" — масштабный отрезок **control.ScaleLine**;
- "searchControl" — панель поиска **control.SearchControl**;
- "trafficControl" — панель пробок **control.TrafficControl**;
- "typeSelector" — панель переключения типа карты **control.TypeSelector**;
- "zoomControl" — ползунок масштаба **control.ZoomControl**;
- "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
item	IControl|String	✔	Элемент управления или его ключ.
positionIndex	Integer		Позиция вставки элемента. Если не задана, элемент добавляется в конец списка.

* Обязательный параметр/опция.

Пример:

```
var groupControl = new ymaps.control.Group();
// Добавляем элемент управления типами карты.
groupControl.add(new ymaps.control.TypeSelector(), 1);
// Добавляем стандарный набор элементов управления картой.
groupControl.add(new ymaps.control.MapTools(), 0);
map.controls.add(groupControl);
```

each

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

Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
`callback`	Function	✔	Функция-обработчик.
`context`	Object		Контекст выполнения функции.

* Обязательный параметр/опция.

filter

```
{IChildOnMap[]} filter(callback)
```

Функция для выборки дочерних элементов по условию.

Возвращает массив дочерних элементов.

Параметры:

Параметр	Тип	*	Описание
`callback`	Function	✔	Функция-обработчик дочерних элементов контрола. Получает на вход ссылку на элемент контрола и возвращает значение true/false. Дочерние элементы, для которых функция-обработчик возвращает true, будут добавлены в результирующий массив.

* Обязательный параметр/опция.

get

```
{IChildOnMap} get(key)
```

Возвращает ссылку на объект контрола, добавленного по ключу или по индексу.

Параметры:

Параметр	Тип	*	Описание
`key`	String|Number	✔	

* Обязательный параметр/опция.

getAll

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

Возвращает массив дочерних элементов контрола.

getContainer

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

Возвращает контейнер, в который добавляются дочерние элементы контрола.

getLength

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

Возвращает количество дочерних элементов.

indexOf

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

Возвращает индекс дочернего элемента в группе.

Параметры:

Параметр	Тип	*	Описание
childToFind	String|**IControl**	✔	Дочерний элемент или его ключ.

* Обязательный параметр/опция.

remove

```
{control.Group} remove(item)
```

Удаляет элемент управления из группы.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
item	**IControl**|String	✔	Элемент управления или его ключ.

* Обязательный параметр/опция.

removeAll

```
{control.Group} removeAll()
```

Удаляет все дочерние элементы из группы.

Возвращает ссылку на себя.

control.ListBox

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **control.Group**.

Класс для создания элемента управления в виде раскрывающегося списка.

Конструктор | Поля | События | Методы

Конструктор

```
control.ListBox(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
`params`		Object	Параметры элемента управления.
	`data`	Object	Данные.
	`data.title`	String	Заголовок списка.
	`items`	**IControl**[]	Пункты списка.
`options`		Object	Опции элемента управления.
	`collapseTimeout`	Number	Временной интервал, по истечении которого раскрытый список автоматически закрывается. Значение по умолчанию: 3000
	`expandOnClick`	Boolean	Флаг, разрешающий автоматическое разворачивание/ сворачивание списка по клику. Значение по умолчанию: true
	`layout`	Function|String	Макет элемента управления. В конструктор макета передается объект, содержащий поля: • 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**.

Параметр	Свойства параметра	Тип	Описание
			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. (Тип: конструктор объекта, реализующий интерфейсы: • **IGroupControlLayout**; • **IExpandableControlLayout**. или ключ макета).
	`minWidth`	Number	Минимальная ширина выпадающего списка. Значение по умолчанию: 0
	`position`	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, left: 5 }
	`visible`	Boolean	Признак того, что элемент управления отображается.

Параметр	Свойства параметра	Тип	Описание
			Значение по умолчанию: true

Примеры:

1.

```
// Пример 1.
// Обработка клика на элементах списка.
var cityList = new ymaps.control.ListBox({
        data: {
            title: 'Выбрать город'
        },
        items: [
            new ymaps.control.ListBoxItem({data: {content: 'Москва'}}),
            new ymaps.control.ListBoxItem({data: {content: 'Новосибирск'}}),
            new ymaps.control.ListBoxSeparator(),
            new ymaps.control.ListBoxItem({data: {content: '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);
```

2.

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

// Выпадающий список по умолчанию реагирует на события 'click' и автоматически
// меняет состояние на развернутое 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' [if state.expanded]style='display:
block;'[else]style='display: none;'[endif] >" +
        "</div >", {

        build: function() {
            MyListBoxLayout.superclass.build.call(this);
            this.childContainerElement = $('#my-list-box').get(0);
            // Каждый раз при перестроении будем генерировать событие,
            // сигнализирующее о смене контейнера для дочерних элементов.
            // 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({}, {layout: MyListBoxLayout});
```

Поля

Имя	Тип	Описание
data	data.Manager	Данные выпадающего списка. Имена полей, доступных через метод data.Manager.get: • title — заголовок выпадающего списка.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.
state	data.Manager	Состояние выпадающего списка. Имена полей, доступных через метод data.Manager.get: • expanded — признак, раскрыт ли список.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от ICollection.
childcontainerchange	Изменение дом-элемента, в который добавляются дочерние элементы группы. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newChildContainerElement — новый дом-элемент для дочерних элементов; • oldChildContainerElement — старый элемент для дочерних элементов. Унаследовано от control.Group.
click	Щелчок по заголовку списка. Экземпляр класса Event.
collapse	Список закрыт. Экземпляр класса Event.
expand	Список открыт. Экземпляр класса Event.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection.

Методы

Имя	Возвращает	Описание
add(item, positionIndex)	**control.Group**	Добавляет элемент управления в группу. Элемент можно добавлять, передав экземпляр класса, реализующего интерфейс **IControl**, или по ключу. Возожные значения ключей: • "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**; • "miniMap" — обзорная карта **control.MiniMap**; • "scaleLine" — масштабный отрезок **control.ScaleLine**; • "searchControl" — панель поиска **control.SearchControl**; • "trafficControl" — панель пробок **control.TrafficControl**; • "typeSelector" — панель переключения типа карты **control.TypeSelector**; • "zoomControl" — ползунок масштаба **control.ZoomControl**; • "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**. Унаследован от **control.Group**.
collapse()		Сворачивает список.
each(callback, context)	**control.Group**	Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик. Унаследован от **control.Group**.
expand()		Раскрывает список.
filter(callback)	**IChildOnMap**[]	Функция для выборки дочерних элементов по условию. Унаследован от **control.Group**.
get(key)	**IChildOnMap**	Возвращает ссылку на объект контрола, добавленного по ключу или по индексу. Унаследован от **control.Group**.
getAll()	**IChildOnMap**[]	Возвращает массив дочерних элементов контрола. Унаследован от **control.Group**.
getChildElement(child)	HTMLElement	Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться. Унаследован от **IControlParent**.
getContainer()	HTMLElement	Возвращает контейнер, в который добавляются дочерние элементы контрола. Унаследован от **control.Group**.

Имя	Возвращает	Описание
getIterator()	IIterator	Возвращает итератор по коллекции. Унаследован от ICollection.
getLength()	Number	Возвращает количество дочерних элементов. Унаследован от control.Group.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getParent()	IControlParent	Возвращает ссылку на родительский объект. Унаследован от IControl.
indexOf(childToFind)	Integer	Возвращает индекс дочернего элемента в группе. Унаследован от control.Group.
isExpanded()	Boolean	Возвращает признак того, находится ли контрол в развернутом состоянии.
remove(item)	control.Group	Удаляет элемент управления из группы. Унаследован от control.Group.
removeAll()	control.Group	Удаляет все дочерние элементы из группы. Унаследован от control.Group.
setMinWidth(width)		Устанавливает ширину списка.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IControl.
setTitle(html)		Задает новый заголовок раскрывающегося списка.

Описание полей

data

```
data.Manager data
```

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

- title — заголовок выпадающего списка.

Пример:

```
// Добавим на карту выпадающий список и будем менять его заголовок
// 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');
    })
    .add('collapse', function () {
        listBox.data.set('title', 'Список свернут');
    });
map.controls.add(listBox, {top: 10, left: 10});
// ...
map.controls.remove(listBox);
// После удаления элемента с карты удалим слушателей.
listBoxListener.removeAll();
```

state

```
data.Manager state
```

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

- expanded — признак, раскрыт ли список.

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

```
click
```

Щелчок по заголовку списка. Экземпляр класса **Event**.

collapse

```
collapse
```

Список закрыт. Экземпляр класса **Event**.

expand

```
expand
```

Список открыт. Экземпляр класса **Event**.

Описание методов

collapse

```
collapse()
```

Сворачивает список.

expand

```
expand()
```

Раскрывает список.

isExpanded

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

Возвращает признак того, находится ли контрол в развернутом состоянии.

setMinWidth

```
setMinWidth(width)
```

Устанавливает ширину списка.

Параметры:

Параметр	Тип	*	Описание
`width`	Number	✔	Ширина списка.

* Обязательный параметр/опция.

setTitle

```
setTitle(html)
```

Задает новый заголовок раскрывающегося списка.

Параметры:

Параметр	Тип	*	Описание
`html`	String	✔	Заголовок.

* Обязательный параметр/опция.

control.ListBoxItem

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **ICustomizable**, **ISelectableControl**.

Элемент раскрывающегося списка.

Конструктор | Поля | События | Методы

Конструктор

```
control.ListBoxItem(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
`params`		Object|String	Параметры элемента либо строка — контент элемента.
	`data`	Object	Данные элемента.
	`data.content`	String	Содержимое элемента.
`options`		Object	Опции элемента управления.
	`checkbox`	Boolean	Опция, которая определяет внешний вид стандартного макета. • true — слева от содержимого элемента отображается чекбокс; • false — элемент выделяется галочкой справа от содержимого. Значение по умолчанию: false

Параметр	Свойства параметра	Тип	Описание
	`layout`	Function|String	Макет элемента управления. В конструктор макета передается объект, содержащий поля: • control — ссылка на элемент управления; • options — менеджер опций элемента управления control.ListBoxItem.options; • data — менеджер данных элемента управления **control.ListBoxItem.data**; • state — менеджер состояния элемента управления **control.ListBoxItem.state**. 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.ListBoxItem.state** в зависимости от полученных команд. (Тип: конструктор объекта с интерфейсом **ISelectableControlLayout** или ключ макета).
	`selectOnClick`	Boolean	Флаг, разрешающий автоматическое выделение элемента списка по клику. • true — элемент списка выделяется после клика; • false — элемент списка не меняет свой внешний вид после клика. Значение по умолчанию: false
	`visible`	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Поля

Имя	Тип	Описание
data	data.Manager	Данные элемента списка. Имена полей, доступных через метод data.Manager.get: • content — содержимое элемента списка в виде HTML.
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 data.Manager.get: • selected — признак, выбран ли элемент списка; • enabled — 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.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • 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.

Имя	Возвращает	Описание
disable()		Делает элемент управления недоступным (запрещены пользовательские действия). Унаследован от **ISelectableControl**.
enable()		Делает элемент управления доступным (разрешены пользовательские действия). Унаследован от **ISelectableControl**.
getParent()	**IControlParent**	Возвращает ссылку на родительский объект. 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. 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.

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

```
click
```

Выбор элемента списка.

control.ListBoxSeparator

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **IControl**, **ICustomizable**.

Разделитель в выпадающем списке.

Конструктор | Поля | События | Методы

Конструктор

```
control.ListBoxSeparator(data, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
data		Object	Данные.
options		Object	Опции.
	layout	Function|String	Макет разделителя. В конструктор макета передается объект, содержащий поля: • control — ссылка на элемент управления; • options — менеджер опций элемента управления control.ListBoxSeparator.options; • data — менеджер данных элемента управления **control.ListBoxSeparator.data**; • state — менеджер состояния элемента управления **control.ListBoxSeparator.state**. (Тип: конструктор объекта, реализующий интерфейс IControl, или ключ макета).
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные разделителя.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**data.Manager**	Состояние разделителя.

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.

Методы

Имя	Возвращает	Описание
getParent()	**IControlParent**	Возвращает ссылку на родительский объект. Унаследован от **IControl**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IControl**.

Описание полей

data

```
data.Manager data
```

Данные разделителя.

state

```
data.Manager state
```

Состояние разделителя.

control.MapTools

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **control.ToolBar**.

Тулбар со стандартными кнопками.

Конструктор | Поля | События | Методы

Конструктор

```
control.MapTools(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
`params`		Object|**IControl**[]|String[]	Параметры тулбара или список кнопок, которые будут добавлены по умолчанию.
	`items`	**IControl**[]|String[]	Кнопки которые будут добавленны в тулбар. Могут задаваться экземплярами класса, реализующего интерфейс IControl, либо ключами. Для этих кнопок будет созданна специальная группа. Можно использовать следующие ключи: • "default" — короткий синоним для стандартного набора кнопок; • "drag" — кнопка "Навигация"; • "magnifier" — кнопка "Выделение"; • "ruler" — кнопка "Линейка". По умолчанию автоматически добавляются стандартные кнопки "Навигация", "Выделение" и "Линейка". Получить ссылку на группу, содержащую эти объекты, можно будет через метод getDefaultGroup.
`options`		Object	Опции элемента управления.
	`position`	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах.

Параметр	Свойства параметра	Тип	Описание
			Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, right: 5 }
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Пример:

```
var mapTools = new ymaps.control.MapTools({ items: ["drag", "magnifier"]});
// получаем доступ к кнопке, добавленной по умолчанию
mapTools.getDefaultGroup().get("drag");
```

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные тулбара. Унаследовано от **control.ToolBar**.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**data.Manager**	Состояние элемента управления. Имена полей, доступных через метод data.Manager.get: • contentWidth — общая ширина дочерних элементов. Унаследовано от **control.ToolBar**.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от **ICollection**.
childcontainerchange	Изменение дом-элемента, в который добавляются дочерние элементы группы. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newChildContainerElement — новый дом-элемент для дочерних элементов; • oldChildContainerElement — старый элемент для дочерних элементов. Унаследовано от **control.Group**.

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
remove	Произошло удаление дочернего объекта. Унаследовано от **ICollection**.

Методы

Имя	Возвращает	Описание
add(item, positionIndex)	**control.Group**	Добавляет элемент управления в группу. Элемент можно добавлять, передав экземпляр класса, реализующего интерфейс **IControl**, или по ключу. Возожные значения ключей: • "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**; • "miniMap" — обзорная карта **control.MiniMap**; • "scaleLine" — масштабный отрезок **control.ScaleLine**; • "searchControl" — панель поиска **control.SearchControl**; • "trafficControl" — панель пробок **control.TrafficControl**; • "typeSelector" — панель переключения типа карты **control.TypeSelector**; • "zoomControl" — ползунок масштаба **control.ZoomControl**; • "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**. Унаследован от **control.Group**.
each(callback, context)	**control.Group**	Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик. Унаследован от **control.Group**.
filter(callback)	**IChildOnMap**[]	Функция для выборки дочерних элементов по условию. Унаследован от **control.Group**.

Имя	Возвращает	Описание
get(key)	**IChildOnMap**	Возвращает ссылку на объект контрола, добавленного по ключу или по индексу. Унаследован от **control.Group**.
getAll()	**IChildOnMap**[]	Возвращает массив дочерних элементов контрола. Унаследован от **control.Group**.
getChildElement(child)	HTMLElement	Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться. Унаследован от **IControlParent**.
getContainer()	HTMLElement	Возвращает контейнер, в который добавляются дочерние элементы контрола. Унаследован от **control.Group**.
getDefaultGroup()	**control.RadioGroup**	Возвращает ссылку на группу стандартных элементов управления картой.
getIterator()	**IIterator**	Возвращает итератор по коллекции. Унаследован от **ICollection**.
getLength()	Number	Возвращает количество дочерних элементов. Унаследован от **control.Group**.
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getParent()	**IControlParent**	Возвращает ссылку на родительский объект. Унаследован от **IControl**.
indexOf(childToFind)	Integer	Возвращает индекс дочернего элемента в группе. Унаследован от **control.Group**.
remove(item)	**control.Group**	Удаляет элемент управления из группы. Унаследован от **control.Group**.
removeAll()	**control.Group**	Удаляет все дочерние элементы из группы. Унаследован от **control.Group**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IControl**.

Описание методов

getDefaultGroup

```
{control.RadioGroup} getDefaultGroup()
```

Возвращает ссылку на группу стандартных элементов управления картой.

control.MiniMap

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **IChildOnMap**, **ICustomizable**.

Элемент управления "обзорная карта".

Конструктор | Поля | События | Методы

Конструктор

```
control.MiniMap(state, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
state		Object	Состояние обзорной карты.
	expanded	Boolean	true — обзорная карта развернута, false — свернута. Значение по умолчанию: true
	type	String|**MapType**	Тип карты. Список доступных типов карты: • 'yandex#map' — тип карты "схема"; • 'yandex#satellite' — тип карты "спутник"; • 'yandex#hybrid' — тип карты "гибрид"; • 'yandex#publicMap' — тип карты "народная карта"; • 'yandex#publicMapHybrid' — тип карты "народный гибрид". Ключи 'yandex#publicMap' и 'yandex#publicMapHybrid' доступны только для России и Украины. Значение по умолчанию: 'yandex#map'
options		Object	Опции обзорной карты.

Параметр	Свойства параметра	Тип	Описание
	layout	Function|String	Макет элемента управления. В конструктор макета передается объект, содержащий поля: • control — ссылка на элемент управления; • options — менеджер опций элемента управления control.MiniMap.options; • data — менеджер данных элемента управления **control.MiniMap.data**; • state — менеджер состояния элемента управления **control.MiniMap.state**. 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.MiniMap.state** в зависимости от полученных команд. (Тип: конструктор объекта с интерфейсом **IExpandableControlLayout** или ключ макета).
	position	Object	Положение обзорной карты относительно контейнера основной карты. Значение по умолчанию: { left: 5, bottom: 5 }
	size	Integer[]	Размеры обзорной карты. Значение по умолчанию: [128, 90]
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true
	zoomOffset	Number	Разность уровней масштабирования обзорной карты и основной карты. Значение по умолчанию: 5

Пример:

```
// Добавление мини-карты со спутниковой подложкой, размером 90x90.
map.controls.add(new ymaps.control.MiniMap(
    { type: 'yandex#satellite' },
    { size: [90, 90] }
));
```

Поля

Имя	Тип	Описание
data	data.Manager	Данные элемента управления.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.
state	data.Manager	Состояние мини-карты. Имена полей, доступных через метод data.Manager.get: • type — тип мини-карты; • expanded — признак, развернута ли панель.

События

Имя	Описание
collapse	Мини-карта свернута. Экземпляр класса Event.
expand	Мини-карта развернута. Экземпляр класса Event.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
typechange	Изменился тип мини-карты. Экземпляр класса Event.

Методы

Имя	Возвращает	Описание
collapse()	control.MiniMap	Сворачивает обзорную карту.
expand()	control.MiniMap	Разворачивает обзорную карту.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
getType()	String|MapType	Возвращает текущий тип карты.
isExpanded()	Boolean	Возвращает true — обзорная карта развернута, false — свёрнута.
setParent(parent)	IChildOnMap	Устанавливает родительский объект.

Имя	Возвращает	Описание
		Унаследован от **IChildOnMap**.
setType(newType)		Задаёт новый тип карты.

Описание полей

data

```
data.Manager data
```

Данные элемента управления.

state

```
data.Manager state
```

Состояние мини-карты. Имена полей, доступных через метод data.Manager.get:

- type — тип мини-карты;
- expanded — признак, развернута ли панель.

Пример:

```
myMap.get('miniMap').state.set('yandex#satellite');
```

Описание событий

collapse

```
collapse
```

Мини-карта свернута. Экземпляр класса **Event**.

expand

```
expand
```

Мини-карта развернута. Экземпляр класса **Event**.

typechange

```
typechange
```

Изменился тип мини-карты. Экземпляр класса **Event**.

Описание методов

collapse

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

Сворачивает обзорную карту.

Возвращает ссылку на себя.

expand

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

Разворачивает обзорную карту.

Возвращает ссылку на себя.

getType

```
{String|MapType} getType()
```

Возвращает текущий тип карты.

isExpanded

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

Возвращает true — обзорная карта развернута, false — свёрнута.

setType

```
setType(newType)
```

Задаёт новый тип карты.

Параметры:

Параметр	Тип	*	Описание
newType	String|MapType	✔	Новый тип карты.

* Обязательный параметр/опция.

control.RadioGroup

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **control.Group**.

Группа переключаемых элементов. В группе всегда выбран только один элемент.

Конструктор | Поля | События | Методы

Конструктор

```
control.RadioGroup(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
params		Object|ISelectableControl[]	Настройки группы элементов управления. Если в качестве настроек передан массив элементов управления, он трактуется как список дочерних элементов.
	items	ISelectableControl[]	Список дочерних элементов.
options		Object	Опции.

Параметр	Свойства параметра	Тип	Описание
	position	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, left: 5 }
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Пример:

```
// Создание радио-группы элементов управления.
var radioGroup = new ymaps.control.RadioGroup({
        items: [
            new ymaps.control.Button({ data: { content: 'Сохранить' } }),
            new ymaps.control.Button({ data: { content: 'Включить' } })
        ]
    });
map.controls.add(radioGroup, {left: 10, bottom: 20});
// Вторая кнопка будет выбрана по умолчанию.
radioGroup.get(1).select();
```

Поля

Имя	Тип	Описание
data	data.Manager	Данные группы. Унаследовано от control.Group.
events	IEventManager	Менеджер событий.

Имя	Тип	Описание
		Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**data.Manager**	Состояние группы. Унаследовано от **control.Group**.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от **ICollection**.
childcontainerchange	Изменение дом-элемента, в который добавляются дочерние элементы группы. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newChildContainerElement — новый дом-элемент для дочерних элементов; • oldChildContainerElement — старый элемент для дочерних элементов. Унаследовано от **control.Group**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
remove	Произошло удаление дочернего объекта. Унаследовано от **ICollection**.
select	Выбран один из элементов группы. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • item — ссылка на выбранный элемент группы.

Методы

Имя	Возвращает	Описание
add(item, positionIndex)	**control.Group**	Добавляет элемент управления в группу. Элемент можно добавлять, передав экземпляр класса, реализующего интерфейс **IControl**, или по ключу.

Имя	Возвращает	Описание
		Возожные значения ключей: • "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**; • "miniMap" — обзорная карта **control.MiniMap**; • "scaleLine" — масштабный отрезок **control.ScaleLine**; • "searchControl" — панель поиска **control.SearchControl**; • "trafficControl" — панель пробок **control.TrafficControl**; • "typeSelector" — панель переключения типа карты **control.TypeSelector**; • "zoomControl" — ползунок масштаба **control.ZoomControl**; • "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**. Унаследован от **control.Group**.
each(callback, context)	**control.Group**	Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик. Унаследован от **control.Group**.
filter(callback)	**IChildOnMap**[]	Функция для выборки дочерних элементов по условию. Унаследован от **control.Group**.
get(key)	**IChildOnMap**	Возвращает ссылку на объект контрола, добавленного по ключу или по индексу. Унаследован от **control.Group**.
getAll()	**IChildOnMap**[]	Возвращает массив дочерних элементов контрола. Унаследован от **control.Group**.
getChildElement(child)	HTMLElement	Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться. Унаследован от **IControlParent**.
getContainer()	HTMLElement	Возвращает контейнер, в который добавляются дочерние элементы контрола. Унаследован от **control.Group**.
getIterator()	**IIterator**	Возвращает итератор по коллекции. Унаследован от **ICollection**.
getLength()	Number	Возвращает количество дочерних элементов. Унаследован от **control.Group**.
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getParent()	**IControlParent**	Возвращает ссылку на родительский объект.

Имя	Возвращает	Описание
		Унаследован от **IControl**.
getSelectedIndex()	Number	Возвращает индекс выбранного элемента либо -1, если ни один элемент не выбран.
indexOf(childToFind)	Integer	Возвращает индекс дочернего элемента в группе. Унаследован от **control.Group**.
remove(item)	**control.Group**	Удаляет элемент управления из группы. Унаследован от **control.Group**.
removeAll()	**control.Group**	Удаляет все дочерние элементы из группы. Унаследован от **control.Group**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IControl**.

Описание событий

select

```
select
```

Выбран один из элементов группы. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- item — ссылка на выбранный элемент группы.

Описание методов

getSelectedIndex

```
{Number} getSelectedIndex()
```

Возвращает индекс выбранного элемента либо -1, если ни один элемент не выбран.

control.RollupButton

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **control.RadioGroup**.

Вертикальная группа кнопок в тулбаре.

Конструктор | Поля | События | Методы

Конструктор

```
control.RollupButton(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
`params`		Object|**IControl**[]	Параметры группы или массив кнопок.
	`items`	**IControl**[]	Массив кнопок.

Параметр	Свойства параметра	Тип	Описание
`options`		Object	Опции элемента управления.
	`layout`	Function|String	Макет элемента управления. В конструктор макета передается объект, содержащий поля: • control — ссылка на элемент управления; • options — менеджер опций элемента управления control.RollupButton.options; • data — менеджер данных элемента управления control.RollupButton.data; • state — менеджер состояния элемента управления **control.RollupButton.state**. 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.RollupButton.state** в зависимости от полученных команд. (Тип: конструктор объекта, реализующий интерфейсы: • **IGroupControlLayout**; • **IExpandableControlLayout**. или ключ макета).

Параметр	Свойства параметра	Тип	Описание
	position	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, left: 5 }
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Пример:

```
// Создание пользовательской группы кнопок
// В примере используется jQuery, загруженный с http://yandex.st/jquery/1.6.4/
jquery.min.js
var myRollupLayout = ymaps.templateLayoutFactory.createClass(
    " <div id='rollup-header'>[if state.expanded]Свернуть[else]Развернуть[endif]
</div>" +
    // В этом контейнере будут располагаться дочерние элементы.
    " <div id='rollup-body'> </div>", {

        build: function () {
            myRollupLayout.superclass.build.call(this);
            this.checkExpandCallback = ymaps.util.bind(this.checkExpanded,
this);
            $('#rollup-header').bind('click', this.checkExpandCallback);

            // Поскольку макет будет перестраиваться много раз из-за
            // изменения значения поля state.expanded,
            // нужно при каждом перестроении сообщать об изменении контейнера
            // дочерних элементов через событие интерфейса IGroupControlLayout.
            this.events.fire('childcontainerchange', {
```

```
                newChildContainerElement: $('#rollup-body').get(0),
                oldChildContainerElement: this.childContainerElement
            });
            this.childContainerElement = $('#rollup-body').get(0);
        },

        clear: function () {
            $('#rollup-header').unbind('click', this.checkExpandCallback);
            myRollupLayout.superclass.clear.call(this);
        },

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

        checkExpanded: function () {
            // При клике на переключающий элемент проверяем состояние
            // элемента управления и генерируем одно из событий интерфейса
            // IExpandableControlLayout.
            if (this.getData().state.get('expanded')) {
                this.events.fire('collapse');
            } else {
                this.events.fire('expand');
            }
        }
    }),

    rollupButton = new ymaps.control.RollupButton({
        items: [
            new ymaps.control.Button('Киев'),
            new ymaps.control.Button('Львов'),
            new ymaps.control.Button('Харьков')
        ]
    }, {
        layout: myRollupLayout
    });

map.controls.add(rollupButton, {right: 5, top: 5});
```

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные группы. Унаследовано от **control.Group**.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**data.Manager**	Состояние вертикальной группы. Имена полей, доступных через метод data.Manager.get: • expanded — признак, раскрыт ли список.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от **ICollection**.

Имя	Описание
childcontainerchange	Изменение дом-элемента, в который добавляются дочерние элементы группы. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newChildContainerElement — новый дом-элемент для дочерних элементов; • oldChildContainerElement — старый элемент для дочерних элементов. Унаследовано от **control.Group**.
collapse	Список закрыт. Экземпляр класса **Event**.
expand	Список открыт. Экземпляр класса **Event**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
remove	Произошло удаление дочернего объекта. Унаследовано от **ICollection**.
select	Выбран один из элементов группы. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • item — ссылка на выбранный элемент группы. Унаследовано от **control.RadioGroup**.

Методы

Имя	Возвращает	Описание
add(item, positionIndex)	**control.Group**	Добавляет элемент управления в группу. Элемент можно добавлять, передав экземпляр класса, реализующего интерфейс **IControl**, или по ключу.

Имя	Возвращает	Описание
		Возожные значения ключей: • "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**; • "miniMap" — обзорная карта **control.MiniMap**; • "scaleLine" — масштабный отрезок **control.ScaleLine**; • "searchControl" — панель поиска **control.SearchControl**; • "trafficControl" — панель пробок **control.TrafficControl**; • "typeSelector" — панель переключения типа карты **control.TypeSelector**; • "zoomControl" — ползунок масштаба **control.ZoomControl**; • "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**. Унаследован от **control.Group**.
each(callback, context)	**control.Group**	Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик. Унаследован от **control.Group**.
filter(callback)	**IChildOnMap**[]	Функция для выборки дочерних элементов по условию. Унаследован от **control.Group**.
get(key)	**IChildOnMap**	Возвращает ссылку на объект контрола, добавленного по ключу или по индексу. Унаследован от **control.Group**.
getAll()	**IChildOnMap**[]	Возвращает массив дочерних элементов контрола. Унаследован от **control.Group**.
getChildElement(child)	HTMLElement	Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться. Унаследован от **IControlParent**.
getContainer()	HTMLElement	Возвращает контейнер, в который добавляются дочерние элементы контрола. Унаследован от **control.Group**.
getIterator()	**IIterator**	Возвращает итератор по коллекции. Унаследован от **ICollection**.
getLength()	Number	Возвращает количество дочерних элементов. Унаследован от **control.Group**.
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getParent()	**IControlParent**	Возвращает ссылку на родительский объект.

Имя	Возвращает	Описание
		Унаследован от **IControl**.
getSelectedIndex()	Number	Возвращает индекс выбранного элемента либо -1, если ни один элемент не выбран. Унаследован от **control.RadioGroup**.
indexOf(childToFind)	Integer	Возвращает индекс дочернего элемента в группе. Унаследован от **control.Group**.
remove(item)	**control.Group**	Удаляет элемент управления из группы. Унаследован от **control.Group**.
removeAll()	**control.Group**	Удаляет все дочерние элементы из группы. Унаследован от **control.Group**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IControl**.

Описание полей

state

```
data.Manager state
```

Состояние вертикальной группы. Имена полей, доступных через метод data.Manager.get:

- expanded — признак, раскрыт ли список.

Пример:

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

Описание событий

collapse

```
collapse
```

Список закрыт. Экземпляр класса **Event**.

expand

```
expand
```

Список открыт. Экземпляр класса **Event**.

control.RouteEditor

Входит в пакеты package.full, package.route.

Расширяет Button.

Элемент управления "Редактор маршрутов".

Конструктор | Методы

Конструктор

```
control.RouteEditor(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
params		Object	Параметры кнопки.
options		Object	Опции элемента управления.
	position	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, left: 98 }
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Пример:

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

// можно добавить контрол через тулбар
map.controls.add(new ymaps.control.ToolBar([
    new ymaps.control.RouteEditor()
]));
```

Методы

Имя	Возвращает	Описание
getRoute()	router.Route	Возвращает маршрут.

Описание методов

getRoute

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

Возвращает маршрут.

control.ScaleLine

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **IControl**.

Элемент управления "Масштабная линейка".

Конструктор | Поля | События | Методы

Конструктор

```
control.ScaleLine(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции масштабной линейки.
	layout	Function|String	Макет элемента управления. (Тип: конструктор объекта с интерфейсом **ILayout** или ключ макета).
	position	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах.

Параметр	Свойства параметра	Тип	Описание
			Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { right: 7, bottom: 50 }
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Пример:

```
map.controls.add('scaleLine');
```

Поля

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

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.

Методы

Имя	Возвращает	Описание
getParent()	IControlParent	Возвращает ссылку на родительский объект. Унаследован от IControl.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IControl.

control.SearchControl

Входит в пакеты package.full, package.search, package.standard.

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.

Конструктор | Поля | События | Методы

Конструктор

```
control.SearchControl(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции.
	boundedBy	Number[][]	Прямоугольная область на карте, где предположительно находится искомый объект. При ранжировании объекты, находящиеся внутри указанной области, будут получать больший приоритет.
	kind	String	Вид топонима (только для обратного геокодирования). Список возможных значений: • house — дом; • street — улица; • metro — станция метро; • district — район города; • locality — населенный пункт (город/поселок/деревня/село/...). Значение по умолчанию: 'house'

Параметр	Свойства параметра	Тип	Описание
	`layout`	Function|String	Макет элемента управления. В конструктор макета передается объект, содержащий поля: • 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**. 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. (Тип: конструктор объекта с интерфейсом **ISearchControlLayout** или ключ макета).
	`noCentering`	Boolean	false — автоматически располагать центр карты так, чтобы объект был виден целиком. true — не изменять центр карты при показе найденного объекта. Значение по умолчанию: false
	`noPlacemark`	Boolean	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. Значение по умолчанию: false

Параметр	Свойства параметра	Тип	Описание
	noPopup	Boolean	true — не показывать выпадающий список результатов, false — показывать. Значение по умолчанию: false
	position	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, left: 104 }
	provider	IGeocodeProvider|String	Провайдер геокодирования. Можно воспользоваться одним из стандартных провайдеров: • 'yandex#map' — поиск по карте; • 'yandex#publicMap'; — поиск по народной карте. Значение по умолчанию: 'yandex#map'
	resultsPerPage	Integer	Количество результатов поиска на странице. Максимальное значение — 20. Значение по умолчанию: 3

Параметр	Свойства параметра	Тип	Описание
	searchCoordOrder	String	Определяет каким образом нужно интрепретировать координаты в запросе. По умолчанию координаты будут обрабатываться как широта-долгота. Значение по умолчанию: 'latlong'
	strictBounds	Boolean	Искать только внутри области, заданной опцией boundedBy. Объекты вне указанной области попадать в выдачу не будут.
	useMapBounds	Boolean	Флаг, позволяющий учитывать при поиске границы видимой области карты. При значении true, рассчитанная область видимости имеет больший приоритет, чем заданная через boundedBy.
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true
	width	Integer	Ширина элемента (в пикселах). Минимальное значение: 200. Значение по умолчанию: 240

Пример:

```
// Пример 1.
// Создаем карту и панель поиска
var map = new ym.Map('map', {
        center: [51.700684, 39.23291],
        zoom: 4
    }),
    searchControl = new ym.control.SearchControl({ provider:
'yandex#publicMap' });

// добавляем панель на карту в нужную позицию
map.controls.add(searchControl, { left: '40px', top: '10px' });
```

Поля

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

Имя	Тип	Описание
state	data.Manager	Состояние элемента управления. Имена полей, доступных через метод data.Manager.get: • results — массив, содержащий результаты поиска; • currentIndex — индекс текущего выбранного элемента; • pageNumber — текущая страница выдачи результатов; • found — общее количество найденных результатов; • request — текущий активный запрос; • suggest — исправленный запрос; • expanded — признак, свернут или развернут элемент управления; • opened — признак, показана ли панель выдачи результата.

События

Имя	Описание
close	Событие закрытия поисковой панели. Экземпляр класса Event.
collapse	Событие схлопывания поисковой панели. Экземпляр класса Event.
expand	Событие развертывания поисковой панели. Экземпляр класса Event.
load	Событие получения результатов поиска с сервера. Экземпляр класса Event.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
open	Событие открытия поисковой панели. Экземпляр класса Event.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
pagechange	Событие смены страницы с результатами поиска. Экземпляр класса Event.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
resultselect	Событие выбора результата поиска. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • resultIndex — индекс выбранного результата.
resultshow	Событие показа выбранного результата поиска. Индекс выбранного результата можно получить через свойство события 'resultIndex'. Экземпляр класса Event.
submit	Событие отправления поискового запроса на сервер. Экземпляр класса Event.

Методы

Имя	Возвращает	Описание
close()	control.SearchControl	Закрывает панель с результатами поиска.
collapse()	control.SearchControl	Сворачивает панель с результатами поиска.
expand()	control.SearchControl	Разворачивает панель с результатами поиска.
getCurrentPage()	Integer	Возвращает индекс текущий страницы с результатами запроса. Индексы начинаются с 0.
getPageCount()	Integer	Возвращает количество страниц результатов текущего поискового запроса.
getParent()	IControlParent	Возвращает ссылку на родительский объект. Унаследован от IControl.
getRequestString()	String	Возвращает строку, содержащую текущий поисковый запрос.
getResult(index)	util.Promise	Предоставляет доступ к результату по заданному индексу. Это асинхронный метод. Если результат с заданным индексом еще не загружен, то он загрузится.
getResultCount()	Integer	Возвращает количество результатов текущего поискового запроса.
getSelectedIndex()	Integer	Возвращает индекс текущего выбранного элемента.
getSuggestString()	String	Возвращает строку, содержащую исправленный поисковый запрос, если была допущена опечатка.
isExpanded()	Boolean	Возвращает true, если панель с результатами поиска развернута, false — в противном случае.
isOpen()	Boolean	Возвращает true, если панель с результатами поиска открыта, false — в противном случае.
open()	control.SearchControl	Открывает панель с результатами поиска.
search(request)	util.Promise	Осуществляет поиск.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IControl.
showPage(page)	util.Promise	Показывает страницу с результатами запроса по заданному индексу. Индексы страниц начинаются с 0.
showResult(index)	util.Promise	Отображает результат с заданным индексом.

Описание полей

state

```
data.Manager state
```

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

- results — массив, содержащий результаты поиска;
- currentIndex — индекс текущего выбранного элемента;
- pageNumber — текущая страница выдачи результатов;
- found — общее количество найденных результатов;
- request — текущий активный запрос;
- suggest — исправленный запрос;
- expanded — признак, свернут или развернут элемент управления;
- opened — признак, показана ли панель выдачи результата.

Описание событий

close

```
close
```

Событие закрытия поисковой панели. Экземпляр класса **Event**.

collapse

```
collapse
```

Событие схлопывания поисковой панели. Экземпляр класса **Event**.

expand

```
expand
```

Событие развертывания поисковой панели. Экземпляр класса **Event**.

load

```
load
```

Событие получения результатов поиска с сервера. Экземпляр класса **Event**.

open

```
open
```

Событие открытия поисковой панели. Экземпляр класса **Event**.

pagechange

```
pagechange
```

Событие смены страницы с результатами поиска. Экземпляр класса **Event**.

resultselect

```
resultselect
```

Событие выбора результата поиска. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- resultIndex — индекс выбранного результата.

resultshow

```
resultshow
```

Событие показа выбранного результата поиска. Индекс выбранного результата можно получить через свойство события 'resultIndex'. Экземпляр класса **Event**.

submit

```
submit
```

Событие отправления поискового запроса на сервер. Экземпляр класса **Event**.

Описание методов

close

```
{control.SearchControl} close()
```

Закрывает панель с результатами поиска.

Возвращает ссылку на себя.

collapse

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

Сворачивает панель с результатами поиска.

Возвращает ссылку на себя.

expand

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

Разворачивает панель с результатами поиска.

Возвращает ссылку на себя.

getCurrentPage

```
{Integer} getCurrentPage()
```

Возвращает индекс текущий страницы с результатами запроса. Индексы начинаются с 0.

getPageCount

```
{Integer} getPageCount()
```

Возвращает количество страниц результатов текущего поискового запроса.

Возвращает количество страниц.

getRequestString

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

Возвращает строку, содержащую текущий поисковый запрос.

Возвращает строка, содержащая текущий поисковый запрос.

getResult

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

Предоставляет доступ к результату по заданному индексу. Это асинхронный метод. Если результат с заданным индексом еще не загружен, то он загрузится.

возвращает объект типа **util.Promise**.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс результата, начинается с 0.

* Обязательный параметр/опция.

Пример:

```
searchControl.getResult(5).then(function (result) {
    alert(result.properties.get('name'));
});
```

getResultCount

```
{Integer} getResultCount()
```

Возвращает количество результатов текущего поискового запроса.

Возвращает количество результатов.

getSelectedIndex

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

Возвращает индекс текущего выбранного элемента.

getSuggestString

```
{String} getSuggestString()
```

Возвращает строку, содержащую исправленный поисковый запрос, если была допущена опечатка.

Возвращает строка, содержащая исправленный поисковый запрос.

isExpanded

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

Возвращает true, если панель с результатами поиска развернута, false — в противном случае.

Возвращает true если панель развернута.

isOpen

```
{Boolean} isOpen()
```

Возвращает true, если панель с результатами поиска открыта, false — в противном случае.

Возвращает true если панель развернута.

open

```
{control.SearchControl} open()
```

Открывает панель с результатами поиска.

Возвращает ссылку на себя.

search

```
{util.Promise} search(request)
```

Осуществляет поиск.

возвращает объект типа **util.Promise**.

Параметры:

Параметр	Тип	*	Описание
request	String	✔	Запрос.

* Обязательный параметр/опция.

Пример:

```
searchControl.search('Москва').then(function (geoObjectsArr) {

    // geoObjectsArr - это массив геообъектов, содержащий результаты запроса.

     // выводит свойство name первого геообъекта из результатов запроса
    alert(geoObjectsArr[0].properties.get('name'));

});
```

showPage

```
{util.Promise} showPage(page)
```

Показывает страницу с результатами запроса по заданному индексу. Индексы страниц начинаются с 0.

возвращает объект типа **util.Promise**.

Параметры:

Параметр	Тип	*	Описание
page	Integer	✔	Индекс страницы, которую нужно показать.

* Обязательный параметр/опция.

Пример:

```
searchControl.showPage(1).then(
     function (pageIndex) { alert('Показана страница ' + pageIndex); }
);
```

showResult

```
{util.Promise} showResult(index)
```

Отображает результат с заданным индексом.

возвращает объект типа **util.Promise**.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс результата, начинается с 0.

* Обязательный параметр/опция.

control.SmallZoomControl

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **IControl**, **ICustomizable**.

Класс для создания элемента управления "Простой элемент управления масштабом карты". Позволяет уменьшать или увеличивать коэффициент масштабирования карты на 1.

Конструктор | Поля | События | Методы

Конструктор

```
control.SmallZoomControl(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции элемента управления.
	duration	Number	Время анимации маштабирования карты, мс. Значение по умолчанию: 200
	layout	Function|String	Макет элемента управления. В конструктор макета передается объект, содержащий поля: - control — ссылка на элемент управления; - options — менеджер опций элемента управления control.SmallZoomControl.options; - data — менеджер данных элемента управления **control.SmallZoomControl.data**; - state — менеджер состояния элемента управления **control.SmallZoomControl.state**. Макет меняет свой внешний вид на основе данных, состояния и опций элемента управления. Элемент управления, в свою очередь, реагирует

Параметр	Свойства параметра	Тип	Описание
			на интерфейсные события макета и меняет значения полей **control.SmallZoomControl .state** в зависимости от полученных команд. (Тип: конструктор объекта с интерфейсом **IZoomControlLayout** или ключ макета).
	position	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: - top — отступ от верхнего края контейнера карты в пикселах; - right — отступ от правого края контейнера карты в пикселах; - bottom — отступ от нижнего края контейнера карты в пикселах; - left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 75, left: 5 }
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Примеры:

1.

```
// Пример 1.
// Добавление стандартного элемента управления масштабом.
map.controls.add('smallZoomControl', { top: 70, right: 5 });
```

2.

```
// Пример 2.
// Создание пользовательского элемента управления масштабом.
// В примере используется jQuery, загруженный с http://yandex.st/jquery/1.6.4/
jquery.min.js
var MyZoomLayout = ymaps.templateLayoutFactory.createClass("<div>" +
        "<div id='zoom-in'>+</div>" +
        "<div id='zoom-out'>-</div>" +
    "</div>", {

    build: function () {
        // Сначала вызываем родительский метод.
        MyZoomLayout.superclass.build.call(this);
        // Запомним функции-обработчики, за которым закреплен контекст,
        // чтобы впоследствии от них отписаться.
        this.zoomInCallback = ymaps.util.bind(this.zoomIn, this);
        this.zoomOutCallback = ymaps.util.bind(this.zoomOut, this);
        $('#zoom-in').bind('click', this.zoomInCallback);
        $('#zoom-out').bind('click', this.zoomOutCallback);
    },

   clear: function () {
       // Снимаем слушателей.
       $('#zoom-in').unbind('click', this.zoomInCallback);
       $('#zoom-out').unbind('click', this.zoomOutCallback);
       // А затем вызываем родительский метод.
       MyZoomLayout.superclass.clear.call(this);
   },

   zoomIn: function () {
       var map = this.getData().control.getMap();
       // Посылаем событие, которое описано в интерфейсе IZoomControlLayout.
       this.events.fire('zoomchange', {
           oldZoom: map.getZoom(),
           // При клике на '+' будет послана команда увеличить
           // текущий коэффициент масштабирования на 1
           newZoom: map.getZoom() + 1
       });
   },

   zoomOut: function () {
       var map = this.getData().control.getMap();
       this.events.fire('zoomchange', {
           oldZoom: map.getZoom(),
           // При клике на '-' будет послана команда уменьшить
           // текущий коэффициент масштабирования на 1
           newZoom: map.getZoom() - 1
       });
   }
});
var customZoom = new ymaps.control.SmallZoomControl({layout: MyZoomLayout});
```

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные элемента управления.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.

Имя	Тип	Описание
state	data.Manager	Состояние элемента управления. Имена полей, доступных через метод data.Manager.get: • zoom — текущий коэффициент масштабирования; • zoomRange — текущий диапазон коэффициентов масштабирования.

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
zoomchange	Коэффициент масштабирования карты изменен. Экземпляр класса Event. • newZoom — новый коэффициент масштабирования карты; • oldZoom — старый коэффициент масштабирования карты.

Методы

Имя	Возвращает	Описание
getParent()	IControlParent	Возвращает ссылку на родительский объект. Унаследован от IControl.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IControl.

Описание полей

data

```
data.Manager data
```

Данные элемента управления.

state

```
data.Manager state
```

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

- zoom — текущий коэффициент масштабирования;
- zoomRange — текущий диапазон коэффициентов масштабирования.

Описание событий

zoomchange

```
zoomchange
```

Коэффициент масштабирования карты изменен. Экземпляр класса **Event**.

- newZoom — новый коэффициент масштабирования карты;
- oldZoom — старый коэффициент масштабирования карты.

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:

- "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**;
- "miniMap" — обзорная карта **control.MiniMap**;
- "scaleLine" — масштабный отрезок **control.ScaleLine**;
- "searchControl" — панель поиска **control.SearchControl**;
- "trafficControl" — панель пробок **control.TrafficControl**;
- "typeSelector" — панель переключения типа карты **control.TypeSelector**;
- "zoomControl" — ползунок масштаба **control.ZoomControl**;
- "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**.

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.ToolBar

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **control.Group**.

Элемент управления "Тулбар".

Конструктор | Поля | События | Методы

Конструктор

```
control.ToolBar(params, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
params		Object|IControl[]	Настройки группы элементов управления. Если в качестве настроек передан массив элементов управления, он трактуется как список дочерних элементов.
	items	IControl[]	Список дочерних элементов.
options		Object	опции.
	position	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах. Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, left: 5 }
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Пример:

```
map.controls.add(new ymaps.control.ToolBar([
    new ymaps.control.Button({data: { content: 'Кнопка раз' }}),
    new ymaps.control.Button({data: { content: 'Кнопка два' }})
]));
```

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные тулбара.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**data.Manager**	Состояние элемента управления. Имена полей, доступных через метод data.Manager.get: • contentWidth — общая ширина дочерних элементов.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от **ICollection**.
childcontainerchange	Изменение дом-элемента, в который добавляются дочерние элементы группы. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newChildContainerElement — новый дом-элемент для дочерних элементов; • oldChildContainerElement — старый элемент для дочерних элементов. Унаследовано от **control.Group**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
remove	Произошло удаление дочернего объекта. Унаследовано от **ICollection**.

Методы

Имя	Возвращает	Описание
add(item, positionIndex)	**control.Group**	Добавляет элемент управления в группу. Элемент можно добавлять, передав экземпляр класса, реализующего интерфейс **IControl**, или по ключу. Возожные значения ключей: • "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**; • "miniMap" — обзорная карта **control.MiniMap**; • "scaleLine" — масштабный отрезок **control.ScaleLine**; • "searchControl" — панель поиска **control.SearchControl**; • "trafficControl" — панель пробок **control.TrafficControl**; • "typeSelector" — панель переключения типа карты **control.TypeSelector**; • "zoomControl" — ползунок масштаба **control.ZoomControl**; • "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**. Унаследован от **control.Group**.
each(callback, context)	**control.Group**	Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик. Унаследован от **control.Group**.
filter(callback)	**IChildOnMap**[]	Функция для выборки дочерних элементов по условию. Унаследован от **control.Group**.
get(key)	**IChildOnMap**	Возвращает ссылку на объект контрола, добавленного по ключу или по индексу. Унаследован от **control.Group**.
getAll()	**IChildOnMap**[]	Возвращает массив дочерних элементов контрола. Унаследован от **control.Group**.
getChildElement(child)	HTMLElement	Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться. Унаследован от **IControlParent**.
getContainer()	HTMLElement	Возвращает контейнер, в который добавляются дочерние элементы контрола. Унаследован от **control.Group**.
getIterator()	**IIterator**	Возвращает итератор по коллекции. Унаследован от **ICollection**.

Имя	Возвращает	Описание
getLength()	Number	Возвращает количество дочерних элементов. Унаследован от **control.Group**.
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getParent()	**IControlParent**	Возвращает ссылку на родительский объект. Унаследован от **IControl**.
indexOf(childToFind)	Integer	Возвращает индекс дочернего элемента в группе. Унаследован от **control.Group**.
remove(item)	**control.Group**	Удаляет элемент управления из группы. Унаследован от **control.Group**.
removeAll()	**control.Group**	Удаляет все дочерние элементы из группы. Унаследован от **control.Group**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IControl**.

Описание полей

data

```
data.Manager data
```

Данные тулбара.

state

```
data.Manager state
```

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

- contentWidth — общая ширина дочерних элементов.

control.ToolBarSeparator

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **IControl**, **ICustomizable**.

Создает разделитель на тулбаре

Конструктор | Поля | События | Методы

Конструктор

```
control.ToolBarSeparator(width)
```

Класс для создания разделителя на тулбаре.

Параметры:

Параметр	Тип	*	Описание
`width`	number	✔	Ширина разделителя (в пикселах).

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные разделителя.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**data.Manager**	Состояние разделителя.

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.

Методы

Имя	Возвращает	Описание
getParent()	**IControlParent**	Возвращает ссылку на родительский объект. Унаследован от **IControl**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IControl**.

Описание полей

data

```
data.Manager data
```

Данные разделителя.

state

```
data.Manager state
```

Состояние разделителя.

control.TrafficControl

Входит в пакеты package.full, package.traffic.

Расширяет **IControl**, **ICustomizable**.

Панель управления пробками на карте.

Конструктор | Поля | События | Методы

Конструктор

```
control.TrafficControl(state, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
state		Object	Состояние контрола.
	providerKey	String	Ключ провайдера пробок, показываемых на карте. • 'traffic#actual' — пробки "сейчас" • 'traffic#archive' — пробки "обычно" Значение по умолчанию: 'traffic#actual'
	shown	Boolean	Показаны ли данные по пробкам на карте. Значение по умолчанию: false
options		Object	Опции контрола.

Параметр	Свойства параметра	Тип	Описание
	`layout`	Function|String	Макет элемента управления. В конструктор макета передается объект, содержащий поля: • control — ссылка на элемент управления; • options — менеджер опций элемента управления control.TrafficControl.options; • data — менеджер данных элемента управления **control.TrafficControl.data**; • state — менеджер состояния элемента управления **control.TrafficControl.state**. 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.TrafficControl.state** в зависимости от полученных команд. (Тип: конструктор объекта с интерфейсом **ITrafficControlLayout** или ключ макета).
	`position`	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах.

Параметр	Свойства параметра	Тип	Описание
			Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: {top: 5, right: 120}
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Примеры:

1.

```
// Пример 1.
// Добавление на карту элемента управления пробками
// с включенными пробками "сейчас".
var trafficControl = new ymaps.control.TrafficControl({shown: true});
map.controls.add(trafficControl, {top: 10, left: 10});
```

2.

```
// Пример 2.
// В примере используется jQuery, загруженный с http://yandex.st/jquery/1.6.4/
jquery.min.js
// Добавление на карту пользовательской панели управления пробками.
var MyTrafficControlLayout = ymaps.templateLayoutFactory.createClass("<div
id='traffic'>" +
    "<div id='shown'>[if state.shown]hide[else]show[endif]</div>" +
    "[if state.shown]" +
        "<div id='now'>now</div>" +
        "<div id='usually'>usually</div>" +
    "[endif]" +
    "</div>", {

    build: function () {
        // Вызываем родительский метод.
        MyTrafficControlLayout.superclass.build.call(this);
        // Запоминаем функции-обработчики, для которых закреплен контекст,
        // чтобы потом от них отписаться.
        this.actualCallback = ymaps.util.bind(this.setActualProvider, this);
        this.archiveCallback = ymaps.util.bind(this.setArchiveProvider, this);
        this.shownCallback = ymaps.util.bind(this.checkShown, this);
        // Подписываемся на клики на элементах панели.
        $('#shown').bind('click', this.shownCallback);
        $('#now').bind('click', this.actualCallback);
        $('#usually').bind('click', this.archiveCallback);
     },
     clear: function () {
         // Отписываемся от событий на элементах панели.
         $('#shown').unbind('click', this.shownCallback);
         $('#now').unbind('click', this.actualCallback);
         $('#usually').unbind('click', this.archiveCallback);
         // Вызываем родительский метод.
         MyTrafficControlLayout.superclass.clear.call(this);
```

```
    },

    setActualProvider: function () {
        // Для смены провайдера инициируем событие, описанное в
ITrafficControlLayout.
        this.events.fire('providerkeychange', {newProviderKey:
'traffic#actual'});
    },

    setArchiveProvider: function () {
        // Для смены провайдера инициируем событие, описанное в
ITrafficControlLayout.
        this.events.fire('providerkeychange', {newProviderKey:
'traffic#archive'});
    },

    checkShown: function () {
        // Проверяем состояние элемента управления,
        // и в зависимости от результата генерируем одно из событий
        // интерфейса ITrafficControlLayout.
        if (this.getData().state.get('shown')) {
            this.events.fire('hide');
        } else {
            this.events.fire('show');
        }
    }
});
var trafficControl = new ymaps.control.TrafficControl({}, {layout:
MyTrafficControlLayout});
```

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные панели.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**data.Manager**	Состояние панели. Имена полей, доступных через метод data.Manager.get: • shown — 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.

События

Имя	Описание
collapse	Панель пробок свернута. Экземпляр класса **Event**.
expand	Панель пробок развернута. Экземпляр класса **Event**.
hide	Пробки скрыты. Экземпляр класса **Event**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
providerkeychange	Сменился ключ провайдера. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newProviderKey — новое значение ключа провайдера; • oldProviderKey — старое значение ключа.
show	Показаны пробки. Экземпляр класса Event.

Методы

Имя	Возвращает	Описание
collapse()		Свернуть панель пробок.
expand()		Развернуть панель пробок.
getParent()	IControlParent	Возвращает ссылку на родительский объект. Унаследован от IControl.
getProvider(key)	ITrafficProvider	Возвращает экземпляр провайдера пробок.
hide()		Скрыть провайдер пробок с карты.
isExpanded()	Boolean	Возвращает признак, развернута ли панель.
isShown()	Boolean	Возвращает признак, показан ли провайдер пробок на карте.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IControl.
show()		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:

- shown — 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

```
collapse
```

Панель пробок свернута. Экземпляр класса **Event**.

expand

```
expand
```

Панель пробок развернута. Экземпляр класса **Event**.

hide

```
hide
```

Пробки скрыты. Экземпляр класса **Event**.

providerkeychange

```
providerkeychange
```

Сменился ключ провайдера. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- newProviderKey — новое значение ключа провайдера;
- oldProviderKey — старое значение ключа.

show

```
show
```

Показаны пробки. Экземпляр класса **Event**.

Описание методов

collapse

```
collapse()
```

Свернуть панель пробок.

expand

```
expand()
```

Развернуть панель пробок.

getProvider

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

Возвращает экземпляр провайдера пробок.

Параметры:

Параметр	Тип	Описание
key	String	Ключ провайдера информации о пробках. Список доступных ключей: • '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('info
LayerShown', true);
```

hide

```
hide()
```

Скрыть провайдер пробок с карты.

isExpanded

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

Возвращает признак, развернута ли панель.

isShown

```
{Boolean} isShown()
```

Возвращает признак, показан ли провайдер пробок на карте.

show

```
show()
```

Показать провайдер пробок на карте.

control.TypeSelector

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **control.ListBox**.

Элемент управления "Типы карты". В данный элемент управления можно добавлять элементы списка, которые описывают типы карты, а также дополнительные элементы. Элементы, описывающие типы карты, объединены в группу **control.Group**. Группа по умолчанию добавлена первым элементом. Доступ к этой группе можно получить через метод getMapTypeItemsGroup.

Конструктор | Поля | События | Методы

Конструктор

```
control.TypeSelector(params, options, fixTitle)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
params		String[]|MapType[]|Object	Объект с описаниями параметров элемента управления. Если передан массив, то он трактуется как массив типов карты.
	data.title	String	Заголовок списка.
	mapTypes	String[]|MapType[]	Массив конструкторов типов карты или ключей. Список доступных типов карты: • 'yandex#map' — тип карты "схема"; • 'yandex#satellite' — тип карты "спутник"; • 'yandex#hybrid' — тип карты "гибрид"; • 'yandex#publicMap' — тип карты "народная карта"; • 'yandex#publicMapHybrid' — тип карты "народный гибрид". Ключи 'yandex#publicMap' и 'yandex#publicMapHybrid' доступны только для России и Украины.
	data	Object	Данные списка.
options		Object	Опции элемента управления.
	position	Object	Положение элемента управления над картой. Задается в виде объекта со следующими полями: • top — отступ от верхнего края контейнера карты в пикселах; • right — отступ от правого края контейнера карты в пикселах; • bottom — отступ от нижнего края контейнера карты в пикселах; • left — отступ от левого края контейнера карты в пикселах.

Параметр	Свойства параметра	Тип	Описание
			Если при этом заданы одновременно и top, и bottom, то значение bottom игнорируется. Аналогично, если заданы одновременно и left, и right, то значение right игнорируется. Если элемент управления добавляется в группу элементов (например, в тулбар или раскрывающийся список), то значение position не применяется. Значение по умолчанию: { top: 5, right: 5 }
	visible	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true
fixTitle		Boolean	Флаг, отвечающий за поведение заголовка элемента управления. • true — заголовок элемента управления не изменяется при смене типа карты; • false — заголовок элемента управления изменяется в соответствии с выбранным типом карты. Если тип карты не выбран, будет отображаться значение, заданное в params.data.title. Значение по умолчанию: false

Пример:

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

Поля

Имя	Тип	Описание
data	**data.Manager**	Данные элемента управления. Имена полей, доступных через метод data.Manager.get: • title — заголовок выпадающего списка.
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.

Имя	Тип	Описание
state	data.Manager	Состояние выпадающего списка. Имена полей, доступных через метод data.Manager.get: • expanded — признак, раскрыт ли список. Унаследовано от control.ListBox.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от ICollection.
childcontainerchange	Изменение дом-элемента, в который добавляются дочерние элементы группы. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newChildContainerElement — новый дом-элемент для дочерних элементов; • oldChildContainerElement — старый элемент для дочерних элементов. Унаследовано от control.Group.
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.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection.

Методы

Имя	Возвращает	Описание
add(item, positionIndex)	control.Group	Добавляет элемент управления в группу. Элемент можно добавлять, передав экземпляр класса, реализующего интерфейс IControl, или по ключу.

Имя	Возвращает	Описание
		Возожные значения ключей: • "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**; • "miniMap" — обзорная карта **control.MiniMap**; • "scaleLine" — масштабный отрезок **control.ScaleLine**; • "searchControl" — панель поиска **control.SearchControl**; • "trafficControl" — панель пробок **control.TrafficControl**; • "typeSelector" — панель переключения типа карты **control.TypeSelector**; • "zoomControl" — ползунок масштаба **control.ZoomControl**; • "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**. Унаследован от **control.Group**.
addMapType(mapType, position)	TypeSelector	Добавляет тип карты в список.
collapse()		Сворачивает список. Унаследован от **control.ListBox**.
each(callback, context)	**control.Group**	Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик. Унаследован от **control.Group**.
expand()		Раскрывает список. Унаследован от **control.ListBox**.
filter(callback)	**IChildOnMap**[]	Функция для выборки дочерних элементов по условию. Унаследован от **control.Group**.
get(key)	**IChildOnMap**	Возвращает ссылку на объект контрола, добавленного по ключу или по индексу. Унаследован от **control.Group**.
getAll()	**IChildOnMap**[]	Возвращает массив дочерних элементов контрола. Унаследован от **control.Group**.
getChildElement(child)	HTMLElement	Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться. Унаследован от **IControlParent**.
getContainer()	HTMLElement	Возвращает контейнер, в который добавляются дочерние элементы контрола. Унаследован от **control.Group**.
getIterator()	**IIterator**	Возвращает итератор по коллекции. Унаследован от **ICollection**.
getLength()	Number	Возвращает количество дочерних элементов.

Имя	Возвращает	Описание
		Унаследован от **control.Group**.
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getMapTypeItemsGroup()	**control.Group**	Возвращает группу, которая содержит элементы списка, описывающие типы карты.
getParent()	**IControlParent**	Возвращает ссылку на родительский объект. Унаследован от **IControl**.
indexOf(childToFind)	Integer	Возвращает индекс дочернего элемента в группе. Унаследован от **control.Group**.
isExpanded()	Boolean	Возвращает признак того, находится ли контрол в развернутом состоянии. Унаследован от **control.ListBox**.
remove(item)	**control.Group**	Удаляет элемент управления из группы. Унаследован от **control.Group**.
removeAll()	**control.Group**	Удаляет все дочерние элементы из группы. Унаследован от **control.Group**.
removeAllMapTypes()	TypeSelector	Удаляет все типы карты из элемента управления.
removeMapType(mapType)	TypeSelector	Удаляет тип карты.
setMinWidth(width)		Устанавливает ширину списка. Унаследован от **control.ListBox**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IControl**.
setTitle(html)		Задает новый заголовок раскрывающегося списка. Унаследован от **control.ListBox**.

Описание полей

data

```
data.Manager data
```

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

- title — заголовок выпадающего списка.

Пример:

```
// Добавим на карту выпадающий список и будем менять его заголовок
// в зависимости от того, свернут список или развернут.
var typeSelector = new ymaps.control.TypeSelector();
// Зададим опцию, запрещающую менять заголовок списка.
typeSelector.options.set('fixTitle', true);
// Создадим группу слушателей событий.
var typeSelectorListener = typeSelector.events.group()
    .add('expand', function () {
        typeSelector.data.set('title', 'Список развернут');
    })
    .add('collapse', function () {
        typeSelector.data.set('title', 'Список свернут');
    });
map.controls.add(typeSelector, {top: 10, left: 10});
// ...
map.controls.remove(typeSelector);
// После удаления элемента с карты удалим слушателей.
typeSelectorListener.removeAll();
```

Описание методов

addMapType

```
{TypeSelector} addMapType(mapType, position)
```

Добавляет тип карты в список.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
mapType	String|**MapType**	✔	Тип карты.
position	Integer		Позиция в списке (если не указана, новый тип карты добавляется в конец списка).

* Обязательный параметр/опция.

Пример:

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

getMapTypeItemsGroup

```
{control.Group} getMapTypeItemsGroup()
```

Возвращает группу, которая содержит элементы списка, описывающие типы карты.

removeAllMapTypes

```
{TypeSelector} removeAllMapTypes()
```

Удаляет все типы карты из элемента управления.

Возвращает ссылку на себя.

removeMapType

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

Удаляет тип карты.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
`mapType`	String|**MapType**	✔	

* Обязательный параметр/опция.

control.ZoomControl

Входит в пакеты package.controls, package.full, package.standard.

Расширяет **control.SmallZoomControl**.

Элемент управления "Ползунок масштаба".

Конструктор | Поля | События | Методы

Конструктор

```
control.ZoomControl(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
`options`		Object	Опции элемента управления.
	`customTips`	Object[]	Массив описаний подсказок. Каждое описание — это объект с двумя полями: • `index {Integer}` — коэффициент масштабирования, которому соответствует подсказка; • `value {String}` — текст подсказки.
	`noTips`	Boolean	Включает/выключает показ всплывающих подсказок: true — подсказки выключены, false — включены. Подсказки показываются в стороне от шкалы масштаба при наведении на нее курсора мыши. Значение по умолчанию: false
	`visible`	Boolean	Признак того, что элемент управления отображается. Значение по умолчанию: true

Пример:

```
map.controls.add('zoomControl', { top: 75, left: 5 });
```

Поля

Имя	Тип	Описание
data	data.Manager	Данные элемента управления. Унаследовано от control.SmallZoomControl.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.
state	data.Manager	Состояние элемента управления. Имена полей, доступных через метод data.Manager.get: • zoom — текущий коэффициент масштабирования; • zoomRange — текущий диапазон коэффициентов масштабирования; • hints — массив описаний подсказок, применяющийся в макете элемента управления.

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
zoomchange	Коэффициент масштабирования карты изменен. Экземпляр класса Event. • newZoom — новый коэффициент масштабирования карты; • oldZoom — старый коэффициент масштабирования карты. Унаследовано от control.SmallZoomControl.

Методы

Имя	Возвращает	Описание
getParent()	IControlParent	Возвращает ссылку на родительский объект. Унаследован от IControl.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IControl.

Описание полей

state

```
data.Manager state
```

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

- zoom — текущий коэффициент масштабирования;
- zoomRange — текущий диапазон коэффициентов масштабирования;
- hints — массив описаний подсказок, применяющийся в макете элемента управления.

coordSystem

coordSystem.cartesian

Входит в пакеты package.full, package.map, package.standard.

Статический объект.

Экземпляр класса **ICoordSystem**

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

Методы

Параметры:

Параметр	Тип	Описание
`scale`	Number|Number[]	Масштаб одного деления на оси. Может быть числом или парой чисел для каждой из осей. Значение по умолчанию: 1

Методы

Имя	Возвращает	Описание
getDistance(point1, point2)	Number	Возвращает кратчайшее (вдоль геодезической линии) расстояние между двумя заданными точками (в метрах).
solveDirectProblem(startPoint, direction, distance)	Object	Решает так называемую первую (прямую) геодезическую задачу: где мы окажемся, если выйдем из указанной точки в указанном направлении и продём, не сворачивая, указанное расстояние.
solveInverseProblem(startPoint, endPoint, reverseDirection)	Object	Решает так называемую вторую (обратную) геодезическую задачу: построить кратчайший маршрут между двумя точками на картографируемой поверхности и определелить расстояние и направление движения.

coordSystem.geo

Входит в пакеты package.full, package.map, package.standard.

Статический объект.

Экземпляр класса **ICoordSystem**

Объект, описывающий геометрию земной поверхности. Позволяет строить кратчайшие маршруты (геодезические линии) между точками земной поверхности и находить расстояния.

Методы

Методы

Имя	Возвращает	Описание
getDistance(point1, point2)	Number	Возвращает кратчайшее (вдоль геодезической линии) расстояние между двумя заданными точками (в метрах).
solveDirectProblem(startPoint, direction, distance)	Object	Решает так называемую первую (прямую) геодезическую задачу: где мы окажемся, если выйдем из указанной точки в указанном направлении и продём, не сворачивая, указанное расстояние.
solveInverseProblem(startPoint, endPoint, reverseDirection)	Object	Решает так называемую вторую (обратную) геодезическую задачу: построить кратчайший маршрут между двумя точками на картографируемой поверхности и определелить расстояние и направление движения.

data

data.Manager

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IDataManager**, **IFreezable**.

Менеджер произвольных данных.

Конструктор | Поля | События | Методы

Конструктор

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

Параметры:

Параметр	Тип	*	Описание
data	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
set(path, value)	data.Manager	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 '.'.
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable.
unset(path)	data.Manager	Очищает заданные поля данных.
unsetAll()	data.Manager	Очищает все поля данных.

Описание методов

get

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

Возвращает значение поля данных с заданным именем.

Параметры:

Параметр	Тип	*	Описание
path	String	✔	Строка с именем поля данных. Имя может адресовать вложенные поля, т.е. содержать '.'.
defaultValue	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	Тип	*	Описание
path	Object|String	✔	Строка, содержащая имя поля данных, либо объект вида {"имя поля": "значение"}.
value	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>'
});
```

unset

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

Очищает заданные поля данных.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
path	String|String[]	✔	Имя поля, либо массив имён.

* Обязательный параметр/опция.

unsetAll

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

Очищает все поля данных.

Возвращает ссылку на себя.

DomEvent

Расширяет **IDomEvent**.

Объект-событие. Предоставляет прокси-методы для доступа к полям и методам 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 объект-событие, описывающий DOM-событие.

Параметры:

Параметр	Тип	*	Описание
originalEvent	Object	✔	DOM-событие.
type	String		Тип события. Если не указан, берётся из originalEvent.type.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
callMethod(name)		Вызывает указанный метод у исходного DOM-события. Второй и последующие аргументы передаются методу при вызове. Унаследован от **IDomEvent**.
get(name)	Object	Возвращает свойство события по ключу. С помощью этого метода можно получать доступ как к свойствам оригинального события, так и к дополнительным свойствам, предоставляемым API карт. Унаследован от **IDomEvent**.
isImmediatePropagationStopped()		Возвращает true, если распространение события внутри системы событий API Яндекс.Карт остановлено. Это внутренний признак, никак не связанный с пропагацией события по DOM-дереву. Унаследован от **IDomEvent**.

domEvent

domEvent.manager

Входит в пакеты package.full, package.map, package.standard.

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/MSPointerDown) с одной точкой прикосновения транслируется в последовательность событий mouseenter, mousemove и mousedown;
- Событие перемещения касания (touchmove/MSPointerMove) с одной точкой прикосновения транслируется в событие mousemove;
- События завершения касания (touchend/MSPointerUp) или отмены (touchcancel/MSPointerCancel) 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 DOM-элемента.
var block = document.getElementById('block');
ymaps.domEvent.manager
    .add(block, 'click', function (event) {
        console.log(event.get('type')); // click
    })
    .add(block, 'mouseleave', function (event) {
        console.log(event.get('type')); // mouseleave
    })
    // Задание одного слушателя на несколько событий.
    .add(block, ['mousedown', 'mouseup'], function (event) {
        console.log(event.get('type')); // mousedown / mouseup
    });
```

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.

```
// Исполнение слушателя в контексте определенного объекта.
var block = document.getElementById('block'),
    // Определяем класс.
    someClass = function () {
        this.property = 'value';
    },
    // Создаем реализацию класса.
    someObj = new someClass();
ymaps.domEvent.manager.add(block, 'click', function (event) {
    console.log(this.property + ' ' + event.get('type')); // value click
}, 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
domEvent.manager.**add**(htmlElement, types, callback, context, capture)	✔	**domEvent.manager**	Добавляет слушателя DOM-событий объекта.
domEvent.manager.**group**(htmlElement, capture)	✔	**event.Group**	Создает контейнер для прослушивания событий DOM-элемента.
domEvent.manager.**remove**(htmlElement, types, callback, context, capture)	✔	**domEvent.manager**	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	Тип	*	Описание
htmlElement	HTMLElement	✔	DOM-элемент, события которого необходимо прослушивать.
types	String|String[]	✔	Тип или типы событий.
callback	Function	✔	Функция-обработчик события.
context	Object		Контекст исполнения функции-обработчика.
capture		✔	

* Обязательный параметр/опция.

group

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

Создает контейнер для прослушивания событий DOM-элемента.

Возвращает контейнер событий DOM-элемента.

Параметры:

Параметр	Тип	*	Описание
htmlElement	HTMLElement	✔	Элемент, события которого необходимо прослушивать.
capture		✔	

* 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	Тип	*	Описание
htmlElement	HTMLElement	✔	DOM-элемент, события которого прослушиваются.
types	String|String[]	✔	Тип или типы событий.
callback	Function|String	✔	Функция-обработчик события либо уникальный id пары callback-context.
context	Object		Контекст исполнения функции-обработчика. Значение по умолчанию: window
capture		✔	

* Обязательный параметр/опция.

domEvent.MultiPointer

Расширяет IMultiTouchEvent.

Объект, описывающий multitouch-событие, которое было вызвано несколькими событиями MSPointerEvent. Предоставляет прокси-методы для доступа к полям и методам DOM-события. Возвращаемые значения нормализованы с учётом особенностей ie10.
Конструктор

Конструктор

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

Создаёт объект, описывающий multitouch-событие.
Параметры:

Параметр	Тип	*	Описание
originalEvent	Object	✔	pointer-событие.
type	String		Тип события. Если не указан, считается равным 'multi' + originalEvent.type.

* 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

Конструктор

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

Создаёт объект-событие, описывающий multitouch-событие.
Параметры:

Параметр	Тип	*	Описание
originalEvent	Object	✔	multitouch-событие.
type	String		Тип события. Если не указан, считается равным 'multi' + originalEvent.type.

* Обязательный параметр/опция.

domEvent.Pointer

Расширяет IMultiTouchEvent.

Объект-событие. Предоставляет прокси-методы для доступа к полям и методам DOM-события (одиночное касание экрана в ie10). Возвращаемые значения нормализованы с учётом особенностей ie10.
Конструктор

Конструктор

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

Создаёт объект, описывающий pointer-событие.
Параметры:

Параметр	Тип	*	Описание
originalEvent	Object	✔	DOM-событие.
type	String		Тип события. Если не указан, берётся из originalEvent.type.

* 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

Конструктор

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

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

Параметр	Тип	*	Описание
originalEvent	Object	✔	DOM-событие.
type	String		Тип события. Если не указан, берётся из originalEvent.type.

* Обязательный параметр/опция.

Event

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IEvent**.

Объект-событие. Предоставляет методы для доступа к полям и методам объекта originalObject с возможностью их переопределения.

Конструктор | Методы

Конструктор

```
Event(originalEvent, isPropagationStopped, isDefaultPrevented)
```

Создаёт объект-событие.

Параметры:

Параметр	Тип	*	Описание
originalEvent	Object	✔	Исходные данные.
isPropagationStopped	Boolean		true — пропагирование запрещено, false — нет. Значение по умолчанию: false
isDefaultPrevented	Boolean		true — действие по умолчанию запрещено, false — нет. Значение по умолчанию: false

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
callMethod(name)	Object	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.
get(name)	Object	Возвращает значение поля из originalEvent. originalEvent всегда имеет следующие поля: • type — строковый тип события; • target — ссылка на объект, породивший событие.
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.
isPropagationStopped()	Boolean	Проверяет, остановлено ли распространение события по иерархии объектов и коллекций в системе событий API Яндекс.Карт.

Имя	Возвращает	Описание
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	Тип	*	Описание
name	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.

Возвращает значение.

Параметры:

Параметр	Тип	*	Описание
name	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.

event

event.Group

Расширяет **IEventGroup**.

Слушатель событий.

Конструктор | Поля | Методы

Конструктор

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

Создает экземпляр контейнера событий.

Параметры:

Параметр	Тип	*	Описание
events	**event.Manager**	✔	Менеджер событий, для которого создается контейнер.

* Обязательный параметр/опция.

Пример:

```
// Создадим контейнер событий
var listeners = events.group()
   .add('click', function () { alert('click!') })
   .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	**event.Manager**	Менеджер событий, для которого создан контейнер.
types	Object	Список обработчиков, которые были созданы через данный контейнер. Представляет собой ассоциативный массив, в котором каждому типу событий сопоставлен список listenerId обработчиков в виде JavaScript-объекта.
typesCount	Number	Количество разных типов событий, которые слушают через данный контейнер

Методы

Имя	Возвращает	Описание
add(types, callback, context)	**IEventGroup**	Добавляет обработчик события. Унаследован от **IEventGroup**.
remove(types, callback, context)	**IEventGroup**	Удаляет обработчик события. Удалить можно только те обработчики, которые были добавлены через данный экземпляр контейнера. Унаследован от **IEventGroup**.

Имя	Возвращает	Описание
removeAll()	**IEventGroup**	Удаляет все обработчики событий, которые были добавлены через данный экземпляр контейнера. Унаследован от **IEventGroup**.

Описание полей

events

```
event.Manager events
```

Менеджер событий, для которого создан контейнер.

types

```
Object types
```

Список обработчиков, которые были созданы через данный контейнер. Представляет собой ассоциативный массив, в котором каждому типу событий сопоставлен список listenerId обработчиков в виде JavaScript-объекта.

typesCount

```
Number typesCount
```

Количество разных типов событий, которые слушают через данный контейнер

event.Manager

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IEventManager**.

Менеджер событий с распространением. 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.

Конструктор | Методы

Конструктор

```
event.Manager(options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции менеджера событий.
	context	Object	Объект, для которого создан менеджер событий.
	controllers	**IEventWorkflowController**[]	Контроллер или контроллеры менеджера событий.

Параметр	Свойства параметра	Тип	Описание
	parent	IEventManager	Родительский менеджер событий.

Методы

Имя	Возвращает	Описание
add(types, callback, context)	IEventManager	Добавляет новую подписку. Унаследован от IEventManager.
createEventObject(type, event, target)	Event	Функция, которая создает объект-событие. Вызывается в методе fire, когда переданный объект не является экземпляром класса Event или его наследником.
fire(type, event)	event.Manager	Бросает событие.
getParent()	IEventManager	Возвращает родительский менеджер событий.
group()	IEventGroup	Возвращает контейнер событий, привязанный к данному менеджеру. Унаследован от IEventManager.
remove(types, callback, context)	IEventManager	Удаляет существующую подписку. Унаследован от IEventManager.
setParent(parent)	event.Manager	Задаёт 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	Тип	*	Описание
type	String	✔	Тип события.
event	Object	✔	Объект, описывающий событие.
target	Object	✔	Объект, на котором произошло событие.

* Обязательный параметр/опция.

fire

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

Бросает событие.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
type	String	✔	Тип события.
event	Object|**Event**		Объект, описывающий событие. Если передан хэш с данными, то для него будет вызван метод createEventObject, и дальнейшие действия будут производиться с новым созданным событием. Значение isImmediatePropagationStopped() проверяется в каждой фазе распространения событий. Значение isPropagationStopped() проверяется перед фазой bubblingEvent. Значение isDefaultPrevented() проверяется перед фазой defaultaction.

* Обязательный параметр/опция.

getParent

```
{IEventManager} getParent()
```

Возвращает родительский менеджер событий.

setParent

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

Задаёт родительский менеджер событий.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
parent	**IEventManager**	✔	Родительский менеджер событий.

* Обязательный параметр/опция.

event.PriorityGroup

Расширяет **IEventGroup**.

Слушатель событий.

Конструктор | Поля | Методы

Конструктор

```
event.PriorityGroup(events)
```

Создает экземпляр контейнера событий.

Параметры:

Параметр	Тип	*	Описание
events	event.PriorityManager	✔	Менеджер событий, для которого создается контейнер.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	event.PriorityManager	Менеджер событий, для которого создан контейнер.

Методы

Имя	Возвращает	Описание
add(types, callback, context, priority)	event.PriorityGroup	Добавляет обработчик события.
remove(types, callback, context, priority)	event.PriorityGroup	Удаляет обработчик события. Удалить можно только те обработчики, которые были добавлены через данный экземпляр контейнера.
removeAll()	event.PriorityGroup	Удаляет все обработчики событий, которые были добавлены через данный экземпляр контейнера.

Описание полей

events

```
event.PriorityManager events
```

Менеджер событий, для которого создан контейнер.

Описание методов

add

```
{event.PriorityGroup} add(types, callback, context, priority)
```

Добавляет обработчик события.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или типы событий.
callback	Function	✔	Функция-обработчик.
context	Object		Контекст исполнения функции-обработчика.
priority		✔	

* Обязательный параметр/опция.

remove

```
{event.PriorityGroup} remove(types, callback, context, priority)
```

Удаляет обработчик события. Удалить можно только те обработчики, которые были добавлены через данный экземпляр контейнера.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или типы событий.
callback	Function	✔	Функция-обработчик.
context	Object		Контекст исполнения функции-обработчика.
priority		✔	

* Обязательный параметр/опция.

removeAll

```
{event.PriorityGroup} removeAll()
```

Удаляет все обработчики событий, которые были добавлены через данный экземпляр контейнера.

Возвращает ссылку на себя.

event.PriorityManager

Входит в пакеты package.full, package.map, package.standard.

Расширяет **event.Manager**.

Менеджер событий с приоритетами.
Конструктор | Методы

Конструктор

```
event.PriorityManager(options)
```

Создает экземпляр менеджера событий с приоритетами для некоторого объекта. Позволяет добавлять обработчики событий с указанием приоритета — при бросании событий обработчики будут вызваны в порядке убывания приоритета.

Приоритет должен быть целым числом (возможно, отрицательным). Если приоритет не указан, то он считается наименьшим из возможных, и обработчик будет вызван последним.

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

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции менеджера событий.
	controllers	IWorkflowController[]	Контроллер или контроллеры менеджера событий.

Пример:

```
// Создадим менеджер событий
var events = new ymaps.event.Manager();
events
    .add('click', function () {
        alert('click 1!');
    }, null, 2)
    .add('click', function () {
        alert('click 2!');
    })
    .add('click', function () {
        alert('click 3!');
    }, null, 1);
// Выполняем действия
// ...
// Бросим событие:
events.fire('click', new ymaps.Event());
// Сначала будет выведено 'click 1!', потом 'click 3!',
// потом 'click 2!' - в порядке убывания приоритетов
// (если приоритет не указан, он считается самым малым).
```

Методы

Имя	Возвращает	Описание
add(types, callback, context, priority)	**event.PriorityManager**	Добавляет новую подписку.
createEventObject(type, event, target)	**Event**	Функция, которая создает объект-событие. Вызывается в методе fire, когда переданный объект не является экземпляром класса Event или его наследником.
fire(type, event)	**event.PriorityManager**	Бросает событие.
getManagerByPriority(priority)	event.manager.Base	Перебирает вложенные менеджеры событий и возвращает экземпляр с нужным priority. Если такого нет — создаёт его.
getParent()	**IEventManager**	Возвращает родительский менеджер событий.
group()	**event.PriorityGroup**	Возвращает контейнер событий, привязанный к данному менеджеру.
once(types, callback, context, priority)	**event.PriorityManager**	Добавляет слушателя, который вызовет функцию-обработчик только 1 раз.
remove(types, callback, context, priority)	**event.PriorityManager**	Удаляет существующую подписку. Для удаления подписки требуется указать тот же приоритет, который был использован при её добавлении.
setParent(parent)	**event.Manager**	Задаёт родительский менеджер событий.

Описание методов

add

```
{event.PriorityManager} add(types, callback, context, priority)
```

Добавляет новую подписку.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или массив типов события.
callback	Function	✔	Функция-обработчик события.
context	Object		Контекст исполнения обработчика.
priority	Integer		Приоритет подписки. Если не указан, считается наименьшим из возможных.

* Обязательный параметр/опция.

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	Тип	*	Описание
type	String	✔	Тип события.
event	Object	✔	Объект, описывающий событие.
target	Object	✔	Объект, на котором произошло событие.

* Обязательный параметр/опция.

fire

```
{event.PriorityManager} fire(type, event)
```

Бросает событие.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
type	String	✔	Тип события.

Параметр	Тип	*	Описание
event	Object|Event		Объект, описывающий событие. Если передан хэш с данными, то для него будет вызван метод createEventObject, и дальнейшие действия будут производиться с новым созданным событием. Значение isImmediatePropagationStopped() проверяется в каждой фазе распространения событий. Значение isPropagationStopped() проверяется перед фазой bubblingEvent. Значение isDefaultPrevented() проверяется перед фазой defaultaction.

* Обязательный параметр/опция.

getManagerByPriority

```
{event.manager.Base} getManagerByPriority(priority)
```

Перебирает вложенные менеджеры событий и возвращает экземпляр с нужным priority. Если такого нет — создаёт его.

Возвращает менеджер событий.

Параметры:

Параметр	Тип	Описание
priority	Number	Приоритет.

getParent

```
{IEventManager} getParent()
```

Возвращает родительский менеджер событий.

group

```
{event.PriorityGroup} group()
```

Возвращает контейнер событий, привязанный к данному менеджеру.

once

```
{event.PriorityManager} once(types, callback, context, priority)
```

Добавляет слушателя, который вызовет функцию-обработчик только 1 раз.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или массив типов события.

Параметр	Тип	*	Описание
callback	Function	✔	Функция-обработчик события.
context	Object		Контекст исполнения обработчика.
priority	Integer		Приоритет подписки. Если не указан, считается наименьшим из возможных.

* Обязательный параметр/опция.

remove

```
{event.PriorityManager} remove(types, callback, context, priority)
```

Удаляет существующую подписку. Для удаления подписки требуется указать тот же приоритет, который был использован при её добавлении.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или массив типов события.
callback	Function	✔	Функция-обработчик события.
context	Object		Контекст исполнения обработчика.
priority	Integer		Приоритет подписки. Если не указан, считается наименьшим из возможных.

* Обязательный параметр/опция.

setParent

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

Задаёт родительский менеджер событий.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
parent	IEventManager	✔	Родительский менеджер событий.

* Обязательный параметр/опция.

formatter

Входит в пакеты package.full, package.map, package.standard.

Статический объект.

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

Методы

Методы

Имя	Возвращает	Описание
distance(value, significantDigits)	String	Возвращает отформатированное согласно локальным правилам и переведённое в нужную систему единиц измерения строковое представление расстояния.
duration(value, significantDigits)	String	Возвращает строковое представление интервала времени.

Описание методов

distance

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

Возвращает отформатированное согласно локальным правилам и переведённое в нужную систему единиц измерения строковое представление расстояния.

Параметры:

Параметр	Тип	*	Описание
value	Number	✔	Длина в метрах.
significantDigits	Integer		Число значащих цифр в ответе. Значение по умолчанию: 2

* Обязательный параметр/опция.

duration

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

Возвращает строковое представление интервала времени.

Параметры:

Параметр	Тип	*	Описание
value	Number	✔	Интервал времени в секундах.
significantDigits	Integer		Число значащих цифр в ответе. Значение по умолчанию: 2

* Обязательный параметр/опция.

geocode

Входит в пакеты package.full, package.geocode, package.search, package.standard.

Статическая функция.

Обрабатывает запросы геокодирования. Результат запроса может быть представлен в формате JSON или в виде объекта **GeoObjectCollection**.

Возвращает: объект-promise. (тип **util.Promise**).

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
request		String|Number[]	✔	Адрес, для которого необходимо определить координаты (прямое геокодирование) или координаты, для которых необходимо определить адрес (обратное геокодирование).
options		Object		Опции.
	boundedBy	Number[][]		Прямоугольная область на карте, в которой предположительно находится искомый объект.
	json	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**. Значение по умолчанию: false
	kind	String		Вид топонима (только для обратного геокодирования).

Параметр	Свойства параметра	Тип	*	Описание
				Список возможных значений: • house — дом; • street — улица; • metro — станция метро; • district — район города; • locality — населенный пункт (город/поселок/деревня/село/...). Значение по умолчанию: 'house'
	provider	IGeocodeProvider|String		Провайдер геокодирования. Можно воспользоваться одним из стандартных провайдеров: • 'yandex#map' — поиск по карте; • 'yandex#publicMap' — поиск по народной карте. Значение по умолчанию: 'yandex#map'
	results	Integer		Максимальное количество возвращаемых результатов. Значение по умолчанию: 10
	searchCoordOrder	String		Определяет каким образом нужно интрепретировать координаты в запросе.
	skip	Integer		Число результатов, которое необходимо пропустить. Значение по умолчанию: 0
	strictBounds	Boolean		Искать только внутри области, заданной опцией boundedBy. Значение по умолчанию: false

* Обязательный параметр/опция.

Пример:

```
// Пример 1
// Осуществляет поиск объекта с именем "Москва".
// Полученный результат сразу отображается на карте.
var myGeocoder = ymaps.geocode("Москва");
myGeocoder.then(
    function (res) {
        map.geoObjects.add(res.geoObjects);
    },
    function (err) {
        // обработка ошибки
    }
);

// Пример 2
// реализует интерфейс IGeocodeProvider
var randomPointProvider = {
     geocode: function (request, options) {
         var promise = new ymaps.util.Promise(),
             geoObjects = new ymaps.GeoObjectArray(),
             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, // геообъекты поисковой выдачи
             // метаинформация ответа
             metaData: {
                 geocoder: {
                     request: request, // строка обработанного запроса
                     found: results, // количество найденных результатов
                     results: results, // количество возвращенных результатов
                     skip: options.skip || 0 // количество пропущенных
результатов
                 }
             }
         };

         // выполняем асинхронную обработку
         setTimeout(function () {
             promise.resolve(response);
         }, 0);

         return promise;
     }
},
    myGeocoder = ymaps.geocode("Москва", { provider: randomPointProvider });

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

geolocation

Входит в пакеты package.full, package.map, package.standard.

Статический объект.

Предоставляет информацию о местоположении пользователя на основе его IP адреса. Объект geolocation содержит следующие поля:

- longitude — долгота;
- latitude — широта;
- zoom — масштаб (если доступен);
- city — название города;
- region — название региона;
- country — название страны.

Название города, региона и страны выводится на языке, указанном в параметре lang при подключении API.

geometry

geometry.base

geometry.base.Circle

Входит в пакеты package.full, package.geometries.

Расширяет **IBaseCircleGeometry**.

Базовая геометрия "Круг".

Конструктор | Поля | События | Методы

Конструктор

```
geometry.base.Circle(coordinates, radius)
```

Параметры:

Параметр	Тип	Описание
coordinates	Number[]|null	Координаты центра круга. Значение по умолчанию: null
radius	Number	Радиус круга. Значение по умолчанию: 0

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	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. Унаследовано от **ICircleGeometryAccess**.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри круга. Унаследован от **ICircleGeometryAccess**.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.
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 окружности точку, ближайшую к anchorPosition. Унаследован от **ICircleGeometryAccess**.
getCoordinates()	Number[]|null	Возвращает координаты центра круга. Унаследован от **ICircleGeometryAccess**.
getRadius()	Number	Возвращает радиус круга. Унаследован от **ICircleGeometryAccess**.
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**	Задает координаты центра круга. Унаследован от **ICircleGeometryAccess**.
setRadius(radius)	**ICircleGeometryAccess**	Задает 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

Входит в пакеты package.full, package.geometries.

Расширяет **IBaseLinearRingGeometry**.

Базовая геометрия "Замкнутый контур".

Конструктор | Поля | События | Методы

Конструктор

```
geometry.base.LinearRing(coordinates, fillRule)
```

Параметры:

Параметр	Тип	Описание
coordinates	Number[][]	Координаты геометрии. Значение по умолчанию: []
fillRule	String	Строковый идентификатор, определяющий алгоритм 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. Значение по умолчанию: "evenOdd"

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	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. Унаследовано от ILinearRingGeometryAccess.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри контура. Унаследован от ILinearRingGeometryAccess.
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
get(index)	Number[]	Возвращает координаты точки с заданным индексом. Унаследован от ILinearRingGeometryAccess.
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 контура. Унаследован от ILinearRingGeometryAccess.
getClosest(anchorPosition)	Object	Ищет на контуре точку, ближайшую к anchorPosition. Унаследован от ILinearRingGeometryAccess.
getCoordinates()	Number[][]	Возвращает массив координат геометрии. Унаследован от ILinearRingGeometryAccess.

Имя	Возвращает	Описание
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**	Задает координаты точки с заданным индексом. Унаследован от **ILinearRingGeometryAccess**.
setCoordinates(coordinates)	**ILinearRingGeometryAccess**	Задает массив координат геометрии. Унаследован от **ILinearRingGeometryAccess**.
setFillRule(fillRule)	**ILinearRingGeometryAccess**	Задает алгоритм заливки контура. Унаследован от **ILinearRingGeometryAccess**.
splice(index, number)	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 **ILinearRingGeometryAccess**.
unfreeze()	**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. (тип **geometry.base.LinearRing**).

Параметры:

Параметр	Тип	*	Описание
`encodedCoordinates`	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. (тип String).

Параметры:

Параметр	Тип	*	Описание
`geometry`	**geometry.base.LinearRing**	✔	Геометрия.

* Обязательный параметр/опция.

geometry.base.LineString

Входит в пакеты package.full, package.geometries.

Расширяет **IBaseLineStringGeometry**.

Базовая геометрия "Ломаная линия".

Конструктор | Поля | События | Методы

Конструктор

```
geometry.base.LineString(coordinates)
```

Параметры:

Параметр	Тип	Описание
coordinates	Number[][]	Координаты геометрии. Значение по умолчанию: []

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
events	**event.Manager**	Менеджер событий геометрии.

События

Имя	Описание
change	Изменение координат. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты. Унаследовано от **ILineStringGeometryAccess**.

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**.

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)	**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**.
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**.
set(index, coordinates)	**ILineStringGeometryAccess**	Задает координаты точки с заданным индексом. Унаследован от **ILineStringGeometryAccess**.
setCoordinates(coordinates)	**ILineStringGeometryAccess**	Задает массив координат геометрии. Унаследован от **ILineStringGeometryAccess**.
splice(index, number)	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.

Имя	Возвращает	Описание
		Унаследован от ILineStringGeometryAccess.
unfreeze()	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. (тип geometry.base.LineString).

Параметры:

Параметр	Тип	*	Описание
encodedCoordinates	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. (тип String).

Параметры:

Параметр	Тип	*	Описание
geometry	geometry.base.LineString	✔	Геометрия.

* Обязательный параметр/опция.

geometry.base.Point

Входит в пакеты package.full, package.geometries.

Расширяет **IBasePointGeometry**.

Базовая геометрия "Точка".

Конструктор | Поля | События | Методы

Конструктор

```
geometry.base.Point(coordinates)
```

Параметры:

Параметр	Тип	Описание
coordinates	Number[]|null	Координаты точки. Значение по умолчанию: null

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
events	**event.Manager**	Менеджер событий геометрии.

События

Имя	Описание
change	Изменение координат. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты. Унаследовано от **IPointGeometryAccess**.

Методы

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**.

Имя	Возвращает	Описание
getCoordinates()	Number[]|null	Возвращает координаты точки. Унаследован от IPointGeometryAccess.
getType()	String	Возвращает строку "Point". Унаследован от IBasePointGeometry.
setCoordinates(coordinates)	IPointGeometryAccess	Задает координаты точки. Унаследован от IPointGeometryAccess.

Описание полей

events

```
event.Manager events
```

Менеджер событий геометрии.

geometry.base.Polygon

Входит в пакеты package.full, package.geometries.

Расширяет **IBasePolygonGeometry**.

Базовая геометрия "Многоугольник".

Конструктор | Поля | События | Методы

Конструктор

```
geometry.base.Polygon(coordinates, fillRule)
```

Параметры:

Параметр	Тип	Описание
coordinates	Number[][][]	Координаты геометрии. Значение по умолчанию: []

Параметр	Тип	Описание
fillRule	String	Строковый идентификатор, определяющий алгоритм 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. Значение по умолчанию: "evenOdd"

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	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. Унаследовано от IPolygonGeometryAccess.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри многоугольника. Унаследован от IPolygonGeometryAccess.
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
get(index)	Number[][]	Возвращает координаты контура с заданным индексом. Унаследован от IPolygonGeometryAccess.
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.

Имя	Возвращает	Описание
		Унаследован от **IPolygonGeometryAccess**.
getType()	String	Возвращает строку "Polygon". Унаследован от **IBasePolygonGeometry**.
insert(index, path)	**IPolygonGeometryAccess**	Добавляет новый контур с заданным индексом. Унаследован от **IPolygonGeometryAccess**.
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**	Задает координаты контура с заданным индексом. Унаследован от **IPolygonGeometryAccess**.
setCoordinates(coordinates)	**IPolygonGeometryAccess**	Задает массив координат геометрии. Унаследован от **IPolygonGeometryAccess**.
setFillRule(fillRule)	**IPolygonGeometryAccess**	Задает 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. (тип **geometry.base.Polygon**).

Параметры:

Параметр	Тип	*	Описание
encodedCoordinates	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. (тип String).

Параметры:

Параметр	Тип	*	Описание
geometry	**geometry.base.Polygon**	✔	Геометрия.

* Обязательный параметр/опция.

geometry.base.Rectangle

Входит в пакеты package.full, package.geometries.

Расширяет **IBaseRectangleGeometry**.

Базовая геометрия "Прямоугольник".

Конструктор | Поля | События | Методы

Конструктор

```
geometry.base.Rectangle(coordinates)
```

Параметры:

Параметр	Тип	Описание
coordinates	Number[][]|null	Массив, содержащий координаты двух противоположных углов прямоугольника. Значение по умолчанию: null

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	Изменение координат углов. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты углов; • newCoordinates — новые координаты углов. Унаследовано от **IRectangleGeometryAccess**.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри прямоугольника. Унаследован от **IRectangleGeometryAccess**.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.

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**.
getClosest(anchorPosition)	Object	Ищет на контуре прямоугольника точку, ближайшую к anchorPosition. Унаследован от **IRectangleGeometryAccess**.
getCoordinates()	**IRectangleGeometryAccess**	Возвращает координаты двух противоположных углов прямоугольника. Унаследован от **IRectangleGeometryAccess**.
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**	Задает 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

Входит в пакеты package.editor, package.full, package.geoObjects, package.geometries.

Расширяет **ICircleGeometry**.

Геометрия "Круг".

См.: Circle

Конструктор | Поля | События | Методы

Конструктор

```
geometry.Circle(coordinates, radius, options)
```

См.: Circle

Параметры:

Параметр	Свойства параметра	Тип	Описание
coordinates		Number[]|null	Координаты центра круга. Значение по умолчанию: null
radius		Number	Радиус круга в метрах. Значение по умолчанию: 0
options		Object	Опции геометрии.
	geodesic	Boolean	Включает отображение с использованием геодезических линий. Значение по умолчанию: false
	pixelRendering	String	Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений: • jumpy — отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещается во время движения карты; • static — отображение всегда располагается в начальном мире и не перемещается при движении карты. Значение по умолчанию: "jumpy"
	projection	**IProjection**	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
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: • oldCoordinates — старые координаты центра; • newCoordinates — новые координаты центра; • oldRadius — старый радиус; • newRadius — новый радиус. Унаследовано от **ICircleGeometryAccess**.
mapchange	Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**; • newPixelGeometry — новая пиксельная геометрия **IPixelGeometry**. Унаследовано от **IGeometry**.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри круга. Унаследован от **ICircleGeometryAccess**.

Имя	Возвращает	Описание
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.
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**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

geometry.LineString

Входит в пакеты package.editor, package.full, package.geoObjects, package.geometries.

Расширяет **ILineStringGeometry**.

Геометрия "Ломаная линия".

См.: Polyline

Конструктор | Поля | События | Методы

Конструктор

```
geometry.LineString(coordinates, options)
```

См.: Polyline

Параметры:

Параметр	Свойства параметра	Тип	Описание
`coordinates`		Number[][]	Координаты геометрии. Значение по умолчанию: []
`options`		Object	Опции геометрии.
	`coordRendering`	String	Строковый идентификатор, определяющий алгоритм пересчета координат геометрии в пиксельные координаты. Для геометрии "Ломаная линия" может принимать одно из двух значений: • shortestPath — алгоритм, учитывающий зацикленность проекции по осям, и формирующий пиксельные координаты так, чтобы расстояние между двумя соседними точками было минимальным; • straightPath — алгоритм, не учитывающий зацикленность проекции.
	`geodesic`	Boolean	Включает отображение с использованием геодезических линий. Значение по умолчанию: false

Параметр	Свойства параметра	Тип	Описание
	pixelRendering	String	Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений: • jumpy — отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещается во время движения карты; • static — отображение всегда располагается в начальном мире и не перемещается при движении карты. Значение по умолчанию: "jumpy"
	projection	IProjection	Проекция.
	simplification	Boolean	Включает симплификацию при рендеринге пиксельной геометрии. Значение по умолчанию: true

Пример:

```
// Создаем инстанцию геометрии линии (указываем координаты вершин).
var lineStringGeometry = new ymaps.geometry.LineString([
    [30, 50], [31, 51], [32, 52]
]),
// Создаем инстанцию геообъекта и передаем нашу геометрию в конструктор.
    lineStringGeoObject = new ymaps.GeoObject({ geometry: lineStringGeometry });

lineStringGeometry.event.add('change', function (e) {
   alert([e.get('newCoordinates'), e.get('oldCoordinates')]);
});

// Изменяем вершины через свойство geometry геообъекта (задаем новые координаты
второй точке в линии).
lineStringGeoObject.geometry
    .set(1, [20, 40])
    .remove(2);
// Или напрямую.
lineStringGeometry
    .set(1, [20, 40])
    .remove(2);
// Также доступ к lineStringGeometry можно получить через
lineStringGeoObject.geometry.
```

Поля

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

Имя	Тип	Описание
		Унаследовано от **ICustomizable**.

События

Имя	Описание
change	Изменение координат. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты. Унаследовано от **ILineStringGeometryAccess**.
mapchange	Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**; • newPixelGeometry — новая пиксельная геометрия **IPixelGeometry**. Унаследовано от **IGeometry**.

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 — юго-западный угол области; второй элемент — северо-восточный угол соответственно. Унаследован от **IGeometry**.
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**.

Имя	Возвращает	Описание
getCoordinates()	Number[][]	Возвращает массив координат геометрии. Унаследован от **ILineStringGeometryAccess**.
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**.
set(index, coordinates)	**ILineStringGeometryAccess**	Задает координаты точки с заданным индексом. Унаследован от **ILineStringGeometryAccess**.
setCoordinates(coordinates)	**ILineStringGeometryAccess**	Задает массив координат геометрии. Унаследован от **ILineStringGeometryAccess**.
setMap(map)		Задает карту. Унаследован от **IGeometry**.
splice(index, number)	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**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

geometry.LineString.fromEncodedCoordinates

Статическая функция.

Создает геометрию **geometry.LineString** на основе строки закодированных в Base64 координат.

Возвращает: Геометрию. (тип **geometry.LineString**).

Параметры:

Параметр	Тип	*	Описание
encodedCoordinates	String	✔	Закодированные в Base64 координаты вершин ломаной линии.

* Обязательный параметр/опция.

Пример:

```
var base64Coords = "gMPJAYDw-gJAQg8AQEIPAEBCDwBAQg8A",
    geometry = ymaps.geometry.LineString.fromEncodedCoordinates(base64Coords);
    polyline = new ymaps.Polyline(geometry);
```

geometry.LineString.getPixelRadius

Статическая функция.

Возвращает пиксельный радиус круга с центром в заданой точке на заданном масштабе.

Возвращает: Пиксельный радиус. (тип Number).

Параметры:

Параметр	Тип	*	Описание
coordCenter	Number[]	✔	Центр круга.
radius	Number	✔	Радиус круга.
projection	**IProjection**	✔	Проекция.
zoom	Number	✔	Масштаб.

* Обязательный параметр/опция.

geometry.LineString.toEncodedCoordinates

Статическая функция.

Возвращает: Строку закодированных в Base64 координат для заданного объекта геометрии. (тип String).

Параметры:

Параметр	Тип	*	Описание
geometry	**geometry.LineString**	✔	Геометрия.

* Обязательный параметр/опция.

geometry.pixel

geometry.pixel.Circle

Входит в пакеты package.full, package.geometries, package.overlays.

Расширяет **IPixelCircleGeometry**.

Пиксельная геометрия "Круг".

Конструктор | Поля | Методы

Конструктор

```
geometry.pixel.Circle(coordinates, radius, metaData)
```

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[]|null	✔	Координаты центра круга.
radius	Number|null	✔	Радиус круга.
metaData	Object		Метаданные.

* Обязательный параметр/опция.

Поля

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

Методы

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**.
getCoordinates()	Number[]	Возвращает координаты центра круга. Унаследован от **IPixelCircleGeometry**.
getMetaData()	Object	Возвращает метаданные пиксельной геометрии. Унаследован от **IPixelGeometry**.
getRadius()	Number	Возвращает радиус круга. Унаследован от **IPixelCircleGeometry**.
getType()	String	Возвращает идентификатор типа геометрии.

Имя	Возвращает	Описание
		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

Входит в пакеты package.full, package.geometries, package.overlays.

Расширяет **IPixelLineStringGeometry**.

Пиксельная геометрия "Ломаная линия".

Конструктор | Поля | Методы

Конструктор

```
geometry.pixel.LineString(coordinates, metaData)
```

Параметры:

Параметр	Тип	*	Описание
`coordinates`	Number[][]	✔	Координаты линии.
`metaData`	Object		Метаданные.

* Обязательный параметр/опция.

Поля

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

Методы

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**.
getClosest(anchorPosition)	Object	Ищет на ломаной линии точку, ближайшую к anchorPosition. Унаследован от **IPixelLineStringGeometry**.
getCoordinates()	Number[][]	Возвращает координаты линии.

Имя	Возвращает	Описание
		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.MultiPolygon

Расширяет **IPixelPolygonGeometry**.

Пиксельная геометрия "Многоугольник из нескольких фигур".

Конструктор | Поля | Методы

Конструктор

```
geometry.pixel.MultiPolygon(coordinates, fillRule, metaData)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
`coordinates`		Number[][][][]	✔	Координаты многоугольников.
`fillRule`		String	✔	Строковый идентификатор, определяющий алгоритм заливки многоугольников. Может принимать одно из двух значений: • evenOdd — алгоритм, определяющий, находится ли точка в области заполнения, путем рисования луча от этой точки

Параметр	Свойства параметра	Тип	*	Описание
				до бесконечности и 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 и 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`		Object		Метаданные.

Параметр	Свойства параметра	Тип	*	Описание
	convex	Boolean		Признак выпуклости многоугольника. true — выпуклый, false — невыпуклый. Для выпуклых многоугольников быстрее рассчитывается попадание точки в многоугольник. Значение по умолчанию: false

* 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**.
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**.
getCoordinates()	Number[][][]	Возвращает координаты многоугольника. Унаследован от **IPixelPolygonGeometry**.
getFillRule()	String	Возвращает идентификатор алгоритма заливки многоугольника. Унаследован от **IPixelPolygonGeometry**.
getLength()	Integer	Возвращает количество контуров в многоугольнике. Унаследован от **IPixelPolygonGeometry**.
getMetaData()	Object	Возвращает метаданные пиксельной геометрии.

Имя	Возвращает	Описание
		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

Входит в пакеты package.full, package.geometries, package.overlays.

Расширяет **IPixelPointGeometry**.

Пиксельная геометрия "Точка".

Конструктор | Поля | Методы

Конструктор

```
geometry.pixel.Point(position, metaData)
```

Параметры:

Параметр	Тип	*	Описание
`position`	Number[]|null	✔	Координаты точки.
`metaData`	Object		Метаданные.

* Обязательный параметр/опция.

Поля

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

Методы

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**.
getCoordinates()	Number[]	Возвращает координаты точки.

Имя	Возвращает	Описание
		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

Входит в пакеты package.full, package.geometries, package.overlays.

Расширяет **IPixelPolygonGeometry**.

Пиксельная геометрия "Многоугольник".

Конструктор | Поля | Методы

Конструктор

```
geometry.pixel.Polygon(coordinates, fillRule, metaData)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
`coordinates`		Number[][][]	✔	Координаты многоугольника.
`fillRule`		String	✔	Строковый идентификатор, определяющий алгоритм 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

Параметр	Свойства параметра	Тип	*	Описание
				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 бесконечност и 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`		Object		Метаданные.

Параметр	Свойства параметра	Тип	*	Описание
	convex	Boolean		Признак выпуклости многоугольника. true — выпуклый, false — невыпуклый. Для выпуклых многоугольников быстрее рассчитывается попадание точки в многоугольник. Значение по умолчанию: false

* 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**.
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**.
getCoordinates()	Number[][][]	Возвращает координаты многоугольника. Унаследован от **IPixelPolygonGeometry**.
getFillRule()	String	Возвращает идентификатор алгоритма заливки многоугольника. Унаследован от **IPixelPolygonGeometry**.
getLength()	Integer	Возвращает количество контуров в многоугольнике. Унаследован от **IPixelPolygonGeometry**.
getMetaData()	Object	Возвращает метаданные пиксельной геометрии.

Имя	Возвращает	Описание
		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

Входит в пакеты package.full, package.geometries, package.overlays.

Расширяет **IPixelRectangleGeometry**.

Пиксельная геометрия "Прямоугольник".

Конструктор | Поля | Методы

Конструктор

```
geometry.pixel.Rectangle(coordinates, metaData)
```

Параметры:

Параметр	Тип	Описание
`coordinates`	Number[][]|null	Координаты двух противоположных углов прямоугольника. Значение по умолчанию: null
`metaData`	Object	Метаданные.

Поля

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

Методы

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**.
getClosest(anchorPosition)	Object	Ищет на прямоугольнике точку, ближайшую к anchorPosition.

Имя	Возвращает	Описание
		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

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geocode, package.geometries, package.search, package.standard, package.traffic.

Расширяет **IPointGeometry**.

Геометрия "Точка".

См.: Placemark

Конструктор | Поля | События | Методы

Конструктор

```
geometry.Point(position, options)
```

См.: Placemark

Параметры:

Параметр	Свойства параметра	Тип	Описание
position		Number[]	Координаты точки. Значение по умолчанию: null
options		Object	Опции геометрии.
	pixelRendering	String	Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений: • jumpy — отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещается во время движения карты; • static — отображение всегда располагается в начальном мире и не перемещается при движении карты. Значение по умолчанию: "jumpy"
	projection	**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.
    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.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.

События

Имя	Описание
change	Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты. Унаследовано от IPointGeometryAccess.
mapchange	Сменилась карта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия IPixelGeometry; • newPixelGeometry — новая пиксельная геометрия IPixelGeometry. Унаследовано от IGeometry.

Методы

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.
getCoordinates()	Number[]|null	Возвращает координаты точки. Унаследован от IPointGeometryAccess.
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.

Имя	Возвращает	Описание
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

Входит в пакеты package.editor, package.full, package.geoObjects, package.geometries.

Расширяет **IPolygonGeometry**.

Геометрия "Многоугольник".

См.: Polygon

Конструктор | Поля | События | Методы

Конструктор

```
geometry.Polygon(coordinates, fillRule, options)
```

См.: Polygon

Параметры:

Параметр	Свойства параметра	Тип	Описание
`coordinates`		Number[][][]	Координаты геометрии. Трехмерный массив, элементами которого являются двумерные массивы координат контуров многоугольника. Первый элемент описывает внешний контур, остальные — внутренние. Значение по умолчанию: []

Параметр	Свойства параметра	Тип	Описание
`fillRule`		String	Строковый идентификатор, определяющий алгоритм 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. Значение по умолчанию: "evenOdd"
`options`		Object	Опции геометрии.

Параметр	Свойства параметра	Тип	Описание
	`coordRendering`	String	Строковый идентификатор, определяющий алгоритм пересчета координат геометрии в пиксельные координаты. Может принимать одно из двух значений: • shortestPath — алгоритм, учитывающий зацикленность проекции по осям, и формирующий пиксельные координаты так, чтобы расстояние между двумя соседними точками было минимальным; • straightPath — алгоритм, не учитывающий зацикленность проекции. Значение по умолчанию: "shortestPath"
	`geodesic`	Boolean	Включает отображение с использованием геодезических линий. Значение по умолчанию: false
	`pixelRendering`	String	Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений: • jumpy — отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещается во время движения карты; • static — отображение всегда располагается в начальном мире и не перемещается при движении карты. Значение по умолчанию: "jumpy"
	`projection`	**IProjection**	Проекция.
	`simplification`	Boolean	Включает симплификацию при рендеринге пиксельной геометрии.

Параметр	Свойства параметра	Тип	Описание
			Значение по умолчанию: true

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: • oldCoordinates — старые координаты; • newCoordinates — новые координаты; • oldFillRule — старый алгоритм заливки; • newFillRule — новые алгоритм заливки. Унаследовано от IPolygonGeometryAccess.
mapchange	Сменилась карта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия IPixelGeometry; • newPixelGeometry — новая пиксельная геометрия IPixelGeometry. Унаследовано от IGeometry.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри многоугольника. Унаследован от **IPolygonGeometryAccess**.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.
get(index)	Number[][]	Возвращает координаты контура с заданным индексом. Унаследован от **IPolygonGeometryAccess**.
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**.
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**.
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 "Polygon". Унаследован от **IPolygonGeometry**.
insert(index, path)	**IPolygonGeometryAccess**	Добавляет новый контур с заданным индексом.

Имя	Возвращает	Описание
		Унаследован от **IPolygonGeometryAccess**.
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**	Задает координаты контура с заданным индексом. Унаследован от **IPolygonGeometryAccess**.
setCoordinates(coordinates)	**IPolygonGeometryAccess**	Задает массив координат геометрии. Унаследован от **IPolygonGeometryAccess**.
setFillRule(fillRule)	**IPolygonGeometryAccess**	Задает алгоритм заливки многоугольника. Унаследован от **IPolygonGeometryAccess**.
setMap(map)		Задает карту. Унаследован от **IGeometry**.
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**.

geometry.Polygon.fromEncodedCoordinates

Статическая функция.

Создает геометрию **geometry.Polygon** на основе строки закодированных в Base64 координат.

Возвращает: Геометрию. (тип **geometry.Polygon**).

Параметры:

Параметр	Тип	*	Описание
`encodedCoordinates`	String	✔	Закодированные в Base64 координаты вершин ломаной линии.

* Обязательный параметр/опция.

Пример:

```
var base64Coords =
"AAAAAAAAAAAAAAAAQEtMAEBLTAAAAAAAAAAAAMC0s__AtLP_AAAAAA==;QEIPAEBCDwAAAAAQEIPAE
BCDwAAAAAAAAAAMC98P_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. (тип String).

Параметры:

Параметр	Тип	*	Описание
geometry	geometry.Polygon	✔	Геометрия.

* Обязательный параметр/опция.

geometry.Rectangle

Входит в пакеты package.editor, package.full, package.geoObjects, package.geometries.

Расширяет **IRectangleGeometry**.

Геометрия "Прямоугольник".

См.: Rectangle

Конструктор | Поля | События | Методы

Конструктор

```
geometry.Rectangle(coordinates, options)
```

См.: Rectangle

Параметры:

Параметр	Свойства параметра	Тип	Описание
coordinates		Number[][]|null	Массив, содержащий координаты двух противоположных углов прямоугольника. Значение по умолчанию: null
options		Object	Опции геометрии.

Параметр	Свойства параметра	Тип	Описание
	`coordRendering`	String	Строковый идентификатор, определяющий алгоритм пересчета координат геометрии в пиксельные координаты. Для геометрии "Прямоугольник" может принимать одно из трех значений: • shortestPath — алгоритм, учитывающий зацикленность проекции по осям, и формирующий пиксельные координаты так, чтобы расстояние между противоположными углами было минимальным; • straightPath — алгоритм, не учитывающий зацикленность проекции; • boundsPath — 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.
	`geodesic`	Boolean	Включает отображение с использованием геодезических линий. Значение по умолчанию: false

Параметр	Свойства параметра	Тип	Описание
	pixelRendering	String	Метод расчета пиксельных координат отображения в зацикленных проекциях. Опция может принимать одно из следующих значений: • jumpy — отображение располагается как можно ближе к центру области показа карты и может скачкообразно перемещается во время движения карты; • static — отображение всегда располагается в начальном мире и не перемещается при движении карты. Значение по умолчанию: "jumpy"
	projection	IProjection	Проекция.

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	Менеджер событий. Унаследовано от IEventEmitter.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.

События

Имя	Описание
change	Изменение координат углов. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты углов; • newCoordinates — новые координаты углов. Унаследовано от IRectangleGeometryAccess.

Имя	Описание
mapchange	Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**; • newPixelGeometry — новая пиксельная геометрия **IPixelGeometry**. Унаследовано от **IGeometry**.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри прямоугольника. Унаследован от **IRectangleGeometryAccess**.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.
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 контуре прямоугольника точку, ближайшую к anchorPosition. Унаследован от **IRectangleGeometryAccess**.
getCoordinates()	**IRectangleGeometryAccess**	Возвращает координаты двух противоположных углов прямоугольника. Унаследован от **IRectangleGeometryAccess**.
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**.

Имя	Возвращает	Описание
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от **IFreezable**.
setCoordinates(coordinates)	**IRectangleGeometryAccess**	Задает координаты двух противоположных углов прямоугольника. Унаследован от **IRectangleGeometryAccess**.
setMap(map)		Задает карту. Унаследован от **IGeometry**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

geometryEditor

geometryEditor.LineString

Входит в пакеты package.editor, package.full.

Расширяет **IGeometryEditor**.

Редактор геометрии "Ломаная линия".

Конструктор | Поля | События | Методы

Конструктор

```
geometryEditor.LineString(geometry, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		**ILineStringGeometry**	✔	Геометрия "Ломаная линия".
options		Object		Опции редактора геометрии. Через опции редактора геометрии можно задавать как опции самого редактора, так и опции его составных частей: • опции направляющих линий с префиксом guideLines.
	dblClickHandler	Function		Обработчик двойного щелчка по вершине. Принимает ссылку на модель редактируемой вершины. По умолчанию, обработчик определен функцией, удаляющей соответствующую вершину.
	drawingCursor	Boolean		Курсор мыши в режиме добавления новых вершин. Значение по умолчанию: "arrow"
	drawOver	Boolean		Разрешает ставить точки поверх объектов карты в режиме добавления новых вершин. Значение по умолчанию: true

Параметр	Свойства параметра	Тип	*	Описание
	edgeLayout	Function		Класс макета для промежуточных меток.
	maxPoints	Number		Максимально допустимое число вершин в ломаной линии. Значение по умолчанию: Infinity
	menuManager	Function		Диспетчер контекстного меню, открывающегося при щелчке по вершине. Принимает два аргумента: • массив объектов, описывающих пункты контекстного меню для этой вершины; • ссылку на модель редактируемой вершины. Диспетчер может изменять данные в переданном массиве. Должен возвращать массив объектов, описывающих пункты контекстного меню.
	vertexLayout	Function		Класс макета для меток на вершинах ломаной линии.

* 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.

Имя	Тип	Описание
		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".

События

Имя	Описание
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. 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. 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.

Имя	Описание
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. 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. 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.
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.
edgedragend	Завершение перетаскивания промежуточной метки. Экземпляр класса **Event**.
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.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
statechange	Изменение состояния редактора геометрии. Экземпляр класса **Event**. Унаследовано от **IGeometryEditor**.

Имя	Описание
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.
vertexdragend	Завершение перетаскивания вершины. Экземпляр класса Event.
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.
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.

Методы

Имя	Возвращает	Описание
getView()	geometryEditor.view.Path	Возвращает ссылку на объект отображения редактора.
startDrawing()		Включает режим добавления новых вершин в ломаную линию.
startEditing()		Включает режим редактирования. Унаследован от IGeometryEditor.
stopDrawing()		Отключает режим добавления новых вершин в ломаную линию.
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 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

```
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.

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

```
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.

Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие edgedragstart, будет отменено.

beforevertexadd

```
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

```
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.

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

```
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.

Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие vertexdragstart, будет отменено.

beforevertexdraw

```
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.

edgedrag

```
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.

edgedragend

```
edgedragend
```

Завершение перетаскивания промежуточной метки. Экземпляр класса **Event**.

edgedragstart

```
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.

vertexadd

```
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

```
vertexdrag
```

Перетаскивание вершины. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге.

vertexdragend

```
vertexdragend
```

Завершение перетаскивания вершины. Экземпляр класса **Event**.

vertexdragstart

```
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.

vertexdraw

```
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

getView

```
{geometryEditor.view.Path} getView()
```

Возвращает ссылку на объект отображения редактора.

startDrawing

```
startDrawing()
```

Включает режим добавления новых вершин в ломаную линию.

stopDrawing

```
stopDrawing()
```

Отключает режим добавления новых вершин в ломаную линию.

geometryEditor.model

geometryEditor.model.ChildLinearRing

Расширяет **geometryEditor.model.ChildLineString**.

Модель дочернего замкнутого контура.

Конструктор | Поля | Методы

Конструктор

```
geometryEditor.model.ChildLinearRing(geometry, editor, pixels, parent)
```

Параметры:

Параметр	Тип	*	Описание
geometry	IBaseLinearRingGeometry	✔	Редактируемая геометрия.
editor	IGeometryEditor	✔	Ссылка на редактор геометрии.
pixels	Number[][]	✔	Пиксельные данные модели.
parent	IGeometryEditorModel	✔	Родительская модель данных.

* Обязательный параметр/опция.

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.

Имя	Возвращает	Описание
		Унаследован от **geometryEditor.model.ChildLineString**.
setIndex(index)		Задает индекс дочерней ломаной в родительской модели. Унаследован от **geometryEditor.model.ChildLineString**.
setPixels(pixels)		Задает 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

Расширяет **IGeometryEditorChildModel**.

Модель дочерней ломаной линии.

Конструктор | Поля | Методы

Конструктор

```
geometryEditor.model.ChildLineString(geometry, editor, pixels, parent)
```

Параметры:

Параметр	Тип	*	Описание
geometry	**IBaseLineStringGeometry**	✔	Редактируемая геометрия.
editor	**IGeometryEditor**	✔	Ссылка на редактор геометрии.
pixels	Number[][]	✔	Пиксельные данные модели.
parent	**IGeometryEditorModel**	✔	Родительская модель данных.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
editor	**IGeometryEditor**	Редактор геометрии. Унаследовано от **IGeometryEditorChildModel**.
events	**IEventManager**	Менеджер событий.

Имя	Тип	Описание
		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.
getEdgeModels()	**geometryEditor.model.Edge**[]	Возвращает массив моделей промежуточных меток.
getIndex()	Integer	Возвращает индекс дочерней ломаной в родительской модели.
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.
setIndex(index)		Задает индекс дочерней ломаной в родительской модели.
setPixels(pixels)		Задает 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.

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)
```

Задает индекс дочерней ломаной в родительской модели.

Параметры:

Параметр	Тип	*	Описание
index	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	Тип	*	Описание
start	Integer	✔	Индекс, начиная с которого производится удаление и добавление вершин.
deleteCount	Integer	✔	Количество удаляемых вершин.

* Обязательный параметр/опция.

geometryEditor.model.ChildVertex

Расширяет **IGeometryEditorChildModel**.

Модель дочерней вершины.

Конструктор | Поля | Методы

Конструктор

```
geometryEditor.model.ChildVertex(geometry, editor, pixels, parent)
```

Параметры:

Параметр	Тип	*	Описание
geometry	IBasePointGeometry	✔	Редактируемая геометрия.
editor	IGeometryEditor	✔	Ссылка на редактор геометрии.
pixels	Number[]	✔	Пиксельные данные модели.
parent	IGeometryEditorModel	✔	Родительская модель данных.

* Обязательный параметр/опция.

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)		Задает глобальные пиксельные координаты вершины.
setIndex(index)		Задает индекс дочерней вершины в родительской модели.
setNextVertex(nextVertex)		Устанавливает ссылку на модель следующей вершины.
setPixels(pixels)		Задает пиксельные данные модели. Унаследован от IGeometryEditorChildModel.

Имя	Возвращает	Описание
setPrevVertex(prevVertex)		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)
```

Задает глобальные пиксельные координаты вершины.

Параметры:

Параметр	Тип	*	Описание
pixels	Number[]	✔	Глобальные пиксельные координаты вершины.

* Обязательный параметр/опция.

setIndex

```
setIndex(index)
```

Задает индекс дочерней вершины в родительской модели.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс дочерней вершины.

* Обязательный параметр/опция.

setNextVertex

```
setNextVertex(nextVertex)
```

Устанавливает ссылку на модель следующей вершины.

Параметры:

Параметр	Тип	*	Описание
nextVertex	geometryEditor.model.ChildVertex|null	✔	Модель следующей вершины.

* Обязательный параметр/опция.

setPrevVertex

```
setPrevVertex(prevVertex)
```

Устанавливает ссылку на модель предыдущей вершины.

Параметры:

Параметр	Тип	*	Описание
prevVertex	geometryEditor.model.ChildVertex|null	✔	Модель предыдущей вершины.

* Обязательный параметр/опция.

geometryEditor.model.Edge

Расширяет **IGeometryEditorRootModel**.

Модель промежуточной метки

Конструктор | Поля | Методы

Конструктор

```
geometryEditor.model.Edge(geometry, editor)
```

Параметры:

Параметр	Тип	*	Описание
geometry	geometryEditor.model.EdgeGeometry	✔	Геометрия промежуточной метки.
editor	IGeometryEditor	✔	Ссылка на редактор геометрии.

* Обязательный параметр/опция.

Поля

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

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel.
getNextVertex()	geometryEditor.model.ChildVertex|null	Возвращает ссылку на модель следующей вершины.
getPixels()	Number[]	Возвращает пиксельные данные модели. Унаследован от IGeometryEditorModel.
getPrevVertex()	geometryEditor.model.ChildVertex|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	Тип	*	Описание
nextVertex	geometryEditor.model.ChildVertex|null	✔	Модель следующей вершины.

* Обязательный параметр/опция.

setPrevVertex

```
setPrevVertex(prevVertex)
```

Устанавливает ссылку на модель предыдущей вершины.

Параметры:

Параметр	Тип	*	Описание
prevVertex	geometryEditor.model.ChildVertex|null	✔	Модель предыдущей вершины.

* Обязательный параметр/опция.

geometryEditor.model.EdgeGeometry

Расширяет **IGeometry**.

Геометрия промежуточной метки.

Конструктор | Поля | События | Методы

Конструктор

```
geometryEditor.model.EdgeGeometry(prevVertex, nextVertex, options)
```

Параметры:

Параметр	Тип	*	Описание
prevVertex	geometryEditor.model.ChildVertex	✔	Модель предшествующей вершины.
nextVertex	geometryEditor.model.ChildVertex	✔	Модель следующей вершины.
options	Object		Опции геометрии.

* Обязательный параметр/опция.

Поля

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

События

Имя	Описание
mapchange	Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**; • newPixelGeometry — новая пиксельная геометрия **IPixelGeometry**.

Имя	Описание
	Унаследовано от **IGeometry**.

Методы

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**.
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 идентификатор типа геометрии. Унаследован от **IBaseGeometry**.
setMap(map)		Задает карту. Унаследован от **IGeometry**.

geometryEditor.model.RootLineString

Расширяет **IGeometryEditorRootModel**.

Модель корневой ломаной линии.

Конструктор | Поля | Методы

Конструктор

```
geometryEditor.model.RootLineString(geometry, editor)
```

Параметры:

Параметр	Тип	*	Описание
`geometry`	**ILineStringGeometry**	✔	Редактируемая геометрия.
`editor`	**IGeometryEditor**	✔	Ссылка на редактор геометрии.

* Обязательный параметр/опция.

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 deleteCount.

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	Тип	*	Описание
start	Integer	✔	Индекс, начиная с которого производится удаление и добавление вершин.
deleteCount	Integer	✔	Количество удаляемых вершин.

* Обязательный параметр/опция.

geometryEditor.model.RootPolygon

Расширяет **IGeometryEditorRootModel**.

Модель корневого полигона.

Конструктор | Поля | Методы

Конструктор

```
geometryEditor.model.RootPolygon(geometry, editor)
```

Параметры:

Параметр	Тип	*	Описание
geometry	**IPolygonGeometry**	✔	Редактируемая геометрия.
editor	**IGeometryEditor**	✔	Ссылка на редактор геометрии.

* Обязательный параметр/опция.

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.ChildLinearRing**[]	Возвращает массив моделей дочерних контуров.
getPixels()	Number[]	Возвращает пиксельные данные модели. Унаследован от **IGeometryEditorModel**.
splicePaths(start, deleteCount)	Number[][]	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.

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	Тип	*	Описание
start	Integer	✔	Индекс, начиная с которого производится удаление и добавление контуров.
deleteCount	Integer	✔	Количество удаляемых контуров.

* Обязательный параметр/опция.

geometryEditor.Point

Входит в пакеты package.editor, package.full.

Расширяет **IGeometryEditor**.

Редактор геометрии "Точка".

Конструктор | Поля | События | Методы

Конструктор

```
geometryEditor.Point(geometry, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		**IPointGeometry**	✔	Геометрия "Точка".
options		Object		Опции редактора геометрии. Через опции редактора геометрии можно задавать как опции самого редактора, так и опции его составных частей: • опции направляющих линий с префиксом guideLines.
	dblClickHandler	Function		Обработчик двойного щелчка по вершине. Принимает ссылку на модель редактируемой вершины. По умолчанию, обработчик определен функцией, удаляющей соответствующую вершину.
	drawingCursor	Boolean		Курсор мыши в режиме рисования. Значение по умолчанию: "arrow"
	drawOver	Boolean		Разрешает ставить точки поверх объектов карты в режиме рисования. Значение по умолчанию: true

* 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: • editing — определяет включен ли редактор. Тип — Boolean. Значение по умолчанию — false. • drawing — 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	Описание
startDrawing()	Включает режим рисования точки.
startEditing()	Включает режим редактирования. Унаследован от IGeometryEditor.
stopDrawing()	Отключает режим рисования точки.
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

```
startDrawing()
```

Включает режим рисования точки.

stopDrawing

```
stopDrawing()
```

Отключает режим рисования точки.

geometryEditor.Polygon

Входит в пакеты package.editor, package.full.

Расширяет **IGeometryEditor**.

Редактор геометрии "Многоугольник".

Конструктор | Поля | События | Методы

Конструктор

```
geometryEditor.Polygon(geometry, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
`geometry`		**IPolygonGeometry**	✔	Геометрия "Многоугольник".
`options`		Object		Опции редактора геометрии. Через опции редактора геометрии можно задавать как опции самого редактора, так и опции его составных частей: • опции направляющих линий с префиксом `guideLines`.
	`dblClickHandler`	Function		Обработчик двойного щелчка по вершине. Принимает ссылку на модель редактируемой вершины. По умолчанию, обработчик определен функцией, удаляющей соответствующую вершину.

Параметр	Свойства параметра	Тип	*	Описание
	drawingCursor	Boolean		Курсор мыши в режиме добавления новых вершин. Значение по умолчанию: "arrow"
	drawOver	Boolean		Разрешает ставить точки поверх объектов карты в режиме добавления новых вершин. Значение по умолчанию: true
	edgeLayout	Function		Класс макета для промежуточных меток.
	maxPoints	Number		Максимально допустимое число вершин в многоугольнике. Значение по умолчанию: Infinity
	menuManager	Function		Диспетчер контекстного меню, открывающегося при щелчке по вершине. Принимает два аргумента: • массив объектов, описывающих пункты контекстного меню для этой вершины; • ссылку на модель редактируемой вершины. Диспетчер может изменять данные в переданном массиве. Должен возвращать массив объектов, описывающих пункты контекстного меню.
	vertexLayout	Function		Класс макета для меток на вершинах многоугольника.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий.

Имя	Тип	Описание
		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: • 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.

События

Имя	Описание
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. 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. Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие edgedragstart, будет отменено.

Имя	Описание
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.
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. 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. 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.
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.

Имя	Описание
edgedragend	Завершение перетаскивания промежуточной метки. Экземпляр класса **Event**.
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.
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: • 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.
vertexdragend	Завершение перетаскивания вершины. Экземпляр класса **Event**.
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.
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.

Методы

Имя	Возвращает	Описание
getView()	**geometryEditor.view.MultiPath**	Возвращает ссылку на объект отображения редактора.
startDrawing()		Включает режим добавления новых вершин в многоугольник.
startEditing()		Включает режим редактирования. Унаследован от **IGeometryEditor**.
stopDrawing()		Отключает режим добавления новых вершин в многоугольник.

Имя	Возвращает	Описание
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 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

```
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.

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

```
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.

Если у данного события вызвать метод Event.preventDefault, то последующее перетаскивание, как и событие edgedragstart, будет отменено.

beforevertexadd

```
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

```
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.

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

```
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.

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("target").properties.get("vertexModel"),
        pathModel = vertexModel.getParent();
    if (pathModel.getIndex() == 0 && vertexModel.getIndex() == 0) {
        event.preventDefault();
    }
});
```

beforevertexdraw

```
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.

edgedrag

```
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.

edgedragend

```
edgedragend
```

Завершение перетаскивания промежуточной метки. Экземпляр класса **Event**.

edgedragstart

```
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.

vertexadd

```
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

```
vertexdrag
```

Перетаскивание вершины. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге.

vertexdragend

```
vertexdragend
```

Завершение перетаскивания вершины. Экземпляр класса **Event**.

vertexdragstart

```
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.

vertexdraw

```
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

getView

```
{geometryEditor.view.MultiPath} getView()
```

Возвращает ссылку на объект отображения редактора.

startDrawing

```
startDrawing()
```

Включает режим добавления новых вершин в многоугольник.

stopDrawing

```
stopDrawing()
```

Отключает режим добавления новых вершин в многоугольник.

geometryEditor.view

geometryEditor.view.Edge

Отображение промежуточной метки.

Конструктор | Методы

Конструктор

```
geometryEditor.view.Edge(model, editor, params)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
model		geometryEditor.model.Edge	✔	Модель промежуточной метки.
editor		IGeometryEditor	✔	Ссылка на редактор геометрии.
params		Object		Параметры отображения.
	placemarkCollection	GeoObjectArray|null		Коллекция в которую должны быть добавлена промежуточная метка. Если коллекция не передана, метка добавляется в корневую коллекцию карты.

* Обязательный параметр/опция.

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.

Возвращает геообъект промежуточной метки.

geometryEditor.view.MultiPath

Отображение множества контуров.

Конструктор | Методы

Конструктор

```
geometryEditor.view.MultiPath(model, editor, params)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
model		geometryEditor.model.RootPolygon	✔	Модель многоугольника.
editor		IGeometryEditor	✔	Ссылка на редактор геометрии.
params		Object		Параметры отображения.
	edgePlacemarkCollection	GeoObjectArray|null		Родительская коллекция для коллекции промежуточных меток. Если коллекция не передана, будет использована корневая коллекция карты.
	vertexPlacemarkCollection	GeoObjectArray|null		Родительская коллекция для коллекции вершин. Если коллекция не передана, будет использована корневая коллекция карты.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
getEdgePlacemarks()	GeoObjectArray	Возвращает коллекцию промежуточных меток дочерних отображений контуров.
getPathViews()	geometryEditor.view.Path[]	Возвращает массив отображений дочерних контуров.
getVertexPlacemarks()	GeoObjectArray	Возвращает коллекцию меток вершин дочерних отображений контуров.

Описание методов

getEdgePlacemarks

```
{GeoObjectArray} getEdgePlacemarks()
```

Возвращает коллекцию промежуточных меток дочерних отображений контуров.

getPathViews

```
{geometryEditor.view.Path[]} getPathViews()
```

Возвращает массив отображений дочерних контуров.

getVertexPlacemarks

```
{GeoObjectArray} getVertexPlacemarks()
```

Возвращает коллекцию меток вершин дочерних отображений контуров.

geometryEditor.view.Path

Отображение контура.

Конструктор | Методы

Конструктор

```
geometryEditor.view.Path(model, editor, params)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
model		geometryEditor.model.ChildLinearRing|geometryEditor.model.RootLineString	✔	Модель контура.
editor		IGeometryEditor	✔	Ссылка на редактор геометрии.
params		Object		Параметры отображения.

Параметр	Свойства параметра	Тип	*	Описание
	edgePlacemarkCollection	GeoObjectArray|null		Родительская коллекция для коллекции промежуточных меток. Если коллекция не передана, будет использована корневая коллекция карты.
	vertexPlacemarkCollection	GeoObjectArray|null		Родительская коллекция для коллекции вершин. Если коллекция не передана, будет использована корневая коллекция карты.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
getEdgePlacemarks()	GeoObjectArray	Возвращает коллекцию промежуточных меток, входящих в контур.
getEdgeViews()	geometryEditor.view.Edge[]	Возвращает массив отображений промежуточных меток.
getVertexPlacemarks()	GeoObjectArray	Возвращает коллекцию меток вершин, входящих в контур.
getVertexViews()	geometryEditor.view.Vertex[]	Возвращает массив отображений дочерних вершин.

Описание методов

getEdgePlacemarks

```
{GeoObjectArray} getEdgePlacemarks()
```

Возвращает коллекцию промежуточных меток, входящих в контур.

getEdgeViews

```
{geometryEditor.view.Edge[]} getEdgeViews()
```

Возвращает массив отображений промежуточных меток.

getVertexPlacemarks

```
{GeoObjectArray} getVertexPlacemarks()
```

Возвращает коллекцию меток вершин, входящих в контур.

getVertexViews

```
{geometryEditor.view.Vertex[]} getVertexViews()
```

Возвращает массив отображений дочерних вершин.

geometryEditor.view.Vertex

Отображение вершины.

Конструктор | Методы

Конструктор

```
geometryEditor.view.Vertex(model, editor, params)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
model		geometryEditor.model.ChildVertex	✔	Модель вершины.
editor		IGeometryEditor	✔	Ссылка на редактор геометрии.
params		Object		Параметры отображения.
	placemarkCollection	GeoObjectArray|null		Коллекция в которую должна быть добавлена метка вершины. Если коллекция не передана, метка добавляется в корневую коллекцию карты.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
getPlacemark()	GeoObject	Возвращает ссылку на геообъект метки, обозначающей вершину. В данных метки, под именем "model", доступна ссылка на модель данных вершины.

Описание методов

getPlacemark

```
{GeoObject} getPlacemark()
```

Возвращает ссылку на геообъект метки, обозначающей вершину. В данных метки, под именем "model", доступна ссылка на модель данных вершины.

Возвращает геообъект метки, обозначающей вершину.

GeoObject

Входит в пакеты package.editor, package.full, package.geoObjects.

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)
```

См.: Placemark Polyline Polygon Circle Rectangle

Параметры:

Параметр	Свойства параметра	Тип	Описание
feature		Object	Описание геообъекта.
	geometry	**IGeometry**|Object	Геометрия геообъекта. Может задаваться как экземпляр класса, реализующего интерфейс **IGeometry**, либо в виде хэша с полями type и coordinates.

Параметр	Свойства параметра	Тип	Описание
	properties	IDataManager|Object	Данные геообъекта. Могут задаваться как экземпляр класса, реализующего интерфейс IDataManager, либо в виде хэша. При значениях опций по умолчанию, геообъектом трактуются следующие поля данных: • iconContent — содержимое иконки геообъекта; • hintContent — содержимое всплывающей подсказки геообъекта; • balloonContent — содержимое балуна геообъекта; • balloonContentHeader — содержимое заголовка балуна геообъекта; • balloonContentBody — содержимое основой части балуна геообъекта; • balloonContentFooter — содержимое нижней части балуна геообъекта. Поле balloonContent является кратким обозначением для поля balloonContentBody, но при одновременном задании balloonContentBody 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.

Параметр	Свойства параметра	Тип	Описание
`options`		Object	Опции геообъекта. Используя этот параметр можно задавать как опции самого геообъекта, так и опции его составных частей: • Опции балуна геообъекта с префиксом `balloon`. • Опции всплывающей подсказки геообъекта с префиксом `hint`. • Опции редактора геометрии геообъекта с префиксом `editor`. 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.
	`balloonIconImageHref`	String	URL графического файла иконки в балуне кластера **Cluster**.
	`balloonIconImageOffset`	Number[]	Пиксельный сдвиг иконки в балуне кластера **Cluster** по отношению к родительскому элементу.
	`balloonIconImageSize`	Number[]	Размер иконки в балуне кластера **Cluster** в пикселях.
	`cursor`	String	Вид курсора над геообъектом. Значение по умолчанию: "pointer"
	`draggable`	Boolean	Определяет возможность перетаскивания геообъекта. Значение по умолчанию: false
	`fill`	Boolean	Наличие заливки фигуры. Значение по умолчанию: true
	`fillColor`	String	Цвет заливки.

Параметр	Свойства параметра	Тип	Описание
			Значение по умолчанию: "0066ff99"
	fillImageHref	String	Фоновое изображение. При включении данной опции в режиме заливки **stretch** значение fillColor игнорируется.
	fillMethod	String	Тип заливки фоном. Может принимать одно из двух значений: • 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. Значение по умолчанию: 'stretch'
	fillOpacity	Number	Прозрачность заливки. Значение по умолчанию: 1
	hasBalloon	Boolean	Определяет наличие поля balloon у геообъекта. Значение по умолчанию: true
	hasHint	Boolean	Определяет наличие поля hint у геообъекта. Значение по умолчанию: true
	hideIconOnBalloonOpen	Boolean	Скрывать иконку при открытии балуна. Значение по умолчанию: true
	iconColor	String	Цвет иконки (для растягивающихся под содержимое иконок).
	iconContentLayout	Function|String	Макет содержимого иконки. (Тип: конструктор объекта с интерфейсом ILayout или его ключ в хранилище).
	iconContentOffset	Number[]	Пиксельный сдвиг содержимого иконки относительно родительского элемента. Используется в макете default#imageWithContent.
	iconContentPadding	Number[]	Отступ для содержимого в иконке.

Параметр	Свойства параметра	Тип	Описание
	iconContentSize	Number[]	Размер содержимого. Используется в макете default#imageWithContent.
	iconImageClipRect	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. Значение по умолчанию: [[0, 0], [{imageWidth}, {imageHeight}]]
	iconImageHref	String	URL графического файла иконки.
	iconImageOffset	Number[]	Пиксельный сдвиг изображения иконки внутри родительского элемента.
	iconImageSize	Number[]	Размер иконки в пикселях.
	iconLayout	Function|String	Макет иконки. (Тип: конструктор объекта с интерфейсом ILayout или его ключ в хранилище).
	iconMaxHeight	Number	Максимальная высота иконки с содержимым.
	iconMaxWidth	Number	Максимальная ширина иконки с содержимым.
	iconOffset	Number[]	Пиксельное смещение иконки относительно заданной позиции.
	iconShadow	Boolean	Флаг наличия тени у иконки. Значение по умолчанию: false
	iconShadowImageClipRect	Number[][]	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. Значение по умолчанию: [[0, 0], [{imageWidth}, {imageHeight}]]
	iconShadowImageHref	String	URL графического файла тени иконки.

Параметр	Свойства параметра	Тип	Описание
	iconShadowImageOffset	Number[]	Пиксельный сдвиг изображения тени иконки внутри родительского элемента.
	iconShadowImageSize	Number[]	Размер тени иконки.
	iconShadowLayout	Function|String	Макет тени иконки. (Тип: конструктор объекта с интерфейсом ILayout или его ключ в хранилище).
	iconShadowOffset	Number[]	Пиксельное смещение тени иконки относительно заданной позиции.
	interactivityModel	String	Модель интерактивности. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**. Значение по умолчанию: "default#geoObject"
	opacity	Number	Прозрачность. Значение по умолчанию: 1
	openBalloonOnClick	Boolean	Определяет показывать ли балун при щелчке на геообъекте. Значение по умолчанию: true
	openEmptyBalloon	Boolean	Определяет показывать ли пустой балун при щелчке на геообъекте. Значение по умолчанию: false
	outline	Boolean	Наличие обводки фигуры. Значение по умолчанию: true
	overlayFactory	String|**IOverlayFactory**	Фабрика оверлеев. На основе типа геометрии геообъекта создает те или иные оверлеи для отображения геообъекта. Список доступных фабрик смотрите в описании **geoObject.overlayFactory.storage**. Значение по умолчанию: "default#interactive"
	pane	**IPane**|String	Контейнер оверлея геообъекта. Значение по умолчанию: "overlays"
	preset	String	Ключ предустановленных опций геообъекта. Список ключей, доступных в пакетах package.standard и package.full, содержится в описании **option.presetStorage**.

Параметр	Свойства параметра	Тип	Описание
	showEmptyHint	Boolean	Определяет показывать ли пустой хинт при наведении указателя мыши на геообъект. Значение по умолчанию: false
	showHintOnHover	Boolean	Определяет показывать ли хинт при наведении указателя мыши на геообъект. Значение по умолчанию: true
	strokeColor	String	Цвет линии или обводки. Значение по умолчанию: "0066ffff"
	strokeOpacity	Number	Прозрачность линии или обводки. Значение по умолчанию: 1
	strokeStyle	String	Стиль линии или обводки.
	strokeWidth	Number	Толщина линии или обводки. Значение по умолчанию: 1
	visible	Boolean	Определяет видимость геообъекта. Значение по умолчанию: true
	zIndex	Number	z-index геообъекта в обычном состоянии. Наименее приоритетный.
	zIndexActive	Number	z-index геообъекта с открытым балуном. Наиболее приоритетный.
	zIndexDrag	Number	z-index геообъекта при перетаскивании.
	zIndexHover	Number	z-index геообъекта при наведении на него указателя мыши.

Примеры:

1.

```
var myGeoObject = new ymaps.GeoObject({
        // Описываем геометрию типа "Ломаная линия".
        geometry: {
            type: "LineString",
            coordinates: [
                [37.61, 55.75], [13.38, 52.51]
            ]
        },
        // Описываем данные геообъекта.
        properties: {
            hintContent: "Москва-Берлин"
        }
```

```
    }, {
        // Включаем отображение в форме геодезических кривых.
        geodesic: true,
        // Задаем ширину в 5 пикселей.
        strokeWidth: 5,
        // Задаем цвет линии.
        strokeColor: "#F008"
    });
// Добавляем геообъект на карту.
geoMap.geoObjects.add(myGeoObject);
```

2.

```
var myGeoObject = new ymaps.GeoObject({
        // Описываем геометрию типа "Точка".
        geometry: {
            type: "Point",
            coordinates: [37.61, 55.75]
        },
        // Описываем данные геообъекта.
        properties: {
            hintContent: "Москва",
            balloonContentHeader: "Москва",
            balloonContentBody: "Столица России",
            population: 11848762
        }
    }, {
        // Задаем пресет метки с точкой без содержимого.
        preset: "twirl#redDotIcon",
        // Включаем возможность перетаскивания.
        draggable: true,
        // Переопределяем макет содержимого нижней части балуна.
        balloonContentFooterLayout: ymaps.templateLayoutFactory.createClass(
            'население: $[properties.population], координаты: $
[geometry.coordinates]'
        ),
        // Отключаем задержку закрытия всплывающей подсказки.
        hintHideTimeout: 0
    });
// Добавляем геообъект 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**	Геометрия геообъекта. Унаследовано от **IGeoObject**.
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: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что геообъект перетаскивается.

События

Имя	Описание
balloonclose	Закрытие балуна. Экземпляр класса **Event**.
balloonopen	Открытие балуна на геообъекте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса **Balloon**.
beforedrag	Событие, предшествующее событию drag. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 drag будет отменено.
beforedragstart	Событие, предшествующее событию dragstart. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
drag	Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге.
dragend	Завершение перетаскивания геообъекта. Экземпляр класса **Event**.
dragstart	Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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.

Имя	Описание
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от **IGeoObject**.

Имя	Описание
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от IGeoObject.
propertieschange	Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от IGeoObject.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getOverlay()	IOverlay|null	Возвращает оверлей геообъекта, либо null. Унаследован от IGeoObject.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание полей

balloon

```
geoObject.Balloon balloon
```

Балун геообъекта.

Пример:

```
// Открывает балун геообъекта и позиционирует его максимально близко к центру
карты.
geoObject.balloon.open(myMap.getCenter());
```

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.

hint

```
geoObject.Hint hint
```

Хинт геообъекта.

Пример:

```
// Показывает хинт над геообъектом.
geoObject.hint.show();
```

options

```
option.Manager options
```

Менеджер опций геообъекта.

properties

```
data.Manager properties
```

Менеджер данных геообъекта.

state

```
data.Manager state
```

Состояние геообъекта. Определяется следующими полями:

- active — признак того, что на геообъекте открыт балун.
- hover — признак того, что в данный момент на геообъект наведен указатель мыши.
- drag — признак того, что геообъект перетаскивается.

Описание событий

balloonclose

```
balloonclose
```

Закрытие балуна. Экземпляр класса **Event**.

balloonopen

```
balloonopen
```

Открытие балуна на геообъекте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- balloon — экземпляр класса **Balloon**.

beforedrag

```
beforedrag
```

Событие, предшествующее событию drag. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 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

```
beforedragstart
```

Событие, предшествующее событию dragstart. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- 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

```
drag
```

Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге.

dragend

```
dragend
```

Завершение перетаскивания геообъекта. Экземпляр класса **Event**.

dragstart

```
dragstart
```

Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- domEvent — исходное DOM-событие (в виде объекта **DomEvent**), если оно имеется.

editorstatechange

```
editorstatechange
```

Изменение состояния редактора геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- originalEvent — оригинальное событие редактора геометрии.

geoObject

geoObject.Balloon

Менеджер балуна геообъекта. Позволяет управлять балуном геообъекта, открывать и скрывать его. 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.: GeoObject.balloon

Конструктор | Поля | События | Методы

Конструктор

```
geoObject.Balloon(geoObject)
```

См.: GeoObject.balloon

Параметры:

Параметр	Тип	*	Описание
geoObject	Object	✔	Геообъект.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	**event.Manager**	Менеджер событий балуна геообъекта.

События

Имя	Описание
close	Закрытие балуна. Экземпляр класса **Event**.
open	Открытие балуна. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса **Balloon**.

Методы

Имя	Возвращает	Описание
calculatePixelPosition(anchorPixels)		Определяет пиксельные координаты для открытия балуна на геообъекте.
close(callback)		Закрывает балун.
isOpen()	Boolean	Проверяет, открыт ли балун в данный момент.
open(anchorPixels)	**Balloon**	Открывает балун.

Описание полей

events

```
event.Manager events
```

Менеджер событий балуна геообъекта.

Описание событий

close

```
close
```

Закрытие балуна. Экземпляр класса **Event**.

open

```
open
```

Открытие балуна. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- balloon — экземпляр класса **Balloon**.

Описание методов

calculatePixelPosition

```
calculatePixelPosition(anchorPixels)
```

Определяет пиксельные координаты для открытия балуна на геообъекте.

Параметры:

Параметр	Тип	Описание
anchorPixels	Number[]	Координаты точки привязки. Если указать данный параметр, то результатом работы функции будет точка геообъекта, которая является ближайшей к заданной точке, иначе балун откроется по возможности ближе к центру карты.

Пример:

```
// Открываем балун с привязкой к центру геообъекта, а не к центру карты.
// Данный метод имеет смысл для линий или для полигонов с вырезами, на которые
может приходиться его центр,
// в случае чего нужно найти ближайшую к центру точку на одном из контуров,
расположенных поблизости.
var polygonBounds = polygon.geometry.getBounds(),
    polygonCenter = [
        (polygonBounds[0][0] + polygonBounds[1][0]) / 2,
        (polygonBounds[0][1] + polygonBounds[1][1]) / 2
    ],
    polygonPixelCenter =
myMap.options.get('projection').toGlobalPixels(polygonCenter, myMap.getZoom()),
    balloonPosition =
polygon.balloon.calculatePixelPosition(polygonPixelCenter);

polygon.balloon.open(balloonPosition);
```

close

```
close(callback)
```

Закрывает балун.

Параметры:

Параметр	Тип	*	Описание
callback		✔	

* Обязательный параметр/опция.

isOpen

```
{Boolean} isOpen()
```

Проверяет, открыт ли балун в данный момент.

Возвращает true, если балун открыт, false — иначе.

open

```
{Balloon} open(anchorPixels)
```

Открывает балун.

Возвращает экземпляр балуна.

Параметры:

Параметр	Тип	Описание
anchorPixels	Number[]	Точка привязки балуна в глобальных пиксельных координатах. По умолчанию используются текущие координаты центра карты.

Пример:

```
// Отключает открытие хинта при наведении мыши и открывает вместо него балун.
myMap.geoObjects.options.set({
    showHintOnHover: false
});

myMap.geoObjects.events.add('mouseenter', function (e) {
    var geoObject = e.get('target'),
        position = e.get('globalPixelPosition'),
        balloon = geoObject.balloon.open(position);

    balloon.events.add('mouseleave', function () {
        balloon.close();
    });
});
```

geoObject.Hint

Менеджер хинта геообъекта. Позволяет управлять хинтом геообъекта, открывать и скрывать его. 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.: GeoObject.hint

Конструктор | Поля | События | Методы

Конструктор

```
geoObject.Hint(geoObject)
```

См.: GeoObject.hint

Параметры:

Параметр	Тип	*	Описание
`geoObject`	Object	✔	Геообъект.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	event.Manager	Менеджер событий хинта геообъекта.

События

Имя	Описание
hide	Скрытие хинта. Экземпляр класса Event.
show	Показ хинта над геообъектом. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • hint — экземпляр класса Hint.

Методы

Имя	Возвращает	Описание
hide(callback, urgently)		Скрывает хинт.
isShown()	Boolean	Проверяет состояние хинта на карте. Проксирует вызов Hint.isShown().
move(pixelPosition)		Передвигает хинт, если он в данный момент показан.
show(pixelPosition)	Hint	Показывает хинт.

Описание полей

events

```
event.Manager events
```

Менеджер событий хинта геообъекта.

Описание событий

hide

```
hide
```

Скрытие хинта. Экземпляр класса Event.

show

```
show
```

Показ хинта над геообъектом. Экземпляр класса Event. Имена полей, доступных через метод Event.get:

- hint — экземпляр класса Hint.

Описание методов

hide

```
hide(callback, urgently)
```

Скрывает хинт.

Параметры:

Параметр	Тип	Описание
callback	Function	Функция, которая будет вызвана после скрытия хинта.
urgently	Boolean	Если true, то хинт будет скрыт моментально, без таймаута. Значение по умолчанию: false

Пример:

```
// Моментально (без таймаута) скрываем хинт при клике на нем.
// Примечание: такое поведение также можно реализовать, задав геообъекту опцию
hintHideTimeout = false.
myPoint.hint.events.add('mouseleave', function () {
    myPoint.hint.hide(function () {
        alert('Пока, хинт!');
    }, true);
});
```

isShown

```
{Boolean} isShown()
```

Проверяет состояние хинта на карте. Проксирует вызов **Hint**.isShown().

Возвращает true, если хинт показывается, false, — если спрятан.

move

```
move(pixelPosition)
```

Передвигает хинт, если он в данный момент показан.

Параметры:

Параметр	Тип	*	Описание
pixelPosition	Number[]	✔	Новая позиция хинта в глобальных пиксельных координатах.

* Обязательный параметр/опция.

Пример:

```
// Заставляем хинт двигаться при наведении на геообъект плавными круговыми
движениями в размеренном темпе.
var interval = null;

point.hint.events
    .add('show', function (e) {
        var angle = 0,
            position =
point.options.get('projection').toGlobalPixels(point.geometry.getCoordinates(),
myMap.getZoom());
```

```
        interval = setInterval(function () {
            point.hint.move([
                position[0] + 50 + Math.cos(angle) * 50,
                position[1] + 50 + Math.sin(angle) * 50
            ]);
            angle += Math.PI / 90;
            if (angle >= 2 * Math.PI) {
                angle = 0;
            }
        }, 50);
    })
    .add('hide', function () {
        if (interval) {
            clearInterval(interval);
        }
    });
```

show

```
{Hint} show(pixelPosition)
```

Показывает хинт.

Возвращает экземпляр хинта.

Параметры:

Параметр	Тип	*	Описание
pixelPosition	Number[]	✔	Позиция хинта в глобальных пиксельных координатах.

* Обязательный параметр/опция.

Пример:

```
// Отключает открытие хинта при наведении мыши и открывает его при клике,
// вместо балуна.
myMap.geoObjects.options.set({
    showHintOnHover: false,
    openBalloonOnClick: false
});

myMap.geoObjects.events.add('click', function (e) {
    var geoObject = e.get('target'),
        position = e.get('globalPixelPosition');

    geoObject.hint.show(position);
});
```

geoObject.overlayFactory

geoObject.overlayFactory.hotspot

Входит в пакеты package.editor, package.full, package.geoObjects.

Статический объект.

Экземпляр класса **geoObject.OverlayFactory**

Фабрика оверлеев, отображающихся как активные области. Создает следующие оверлеи для геометрий:

- **geometry.Point** — оверлей **overlay.hotspot.Placemark**.
- **geometry.LineString** — оверлей **overlay.hotspot.Polyline**.
- **geometry.Polygon** — оверлей **overlay.hotspot.Polygon**.
- **geometry.Rectangle** — оверлей **overlay.hotspot.Rectangle**.
- **geometry.Circle** — оверлей **overlay.hotspot.Circle**.

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	**util.Storage**	Добавляет объект в хранилище.
createOverlay(pixelGeometry, data, options)	**IOverlay**	Создает и возвращает экземпляр оверлея на основе типа пиксельной геометрии.
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.

geoObject.overlayFactory.interactive

Входит в пакеты package.editor, package.full, package.geoObjects, package.geoXml, package.route, package.traffic.

Статический объект.

Экземпляр класса **geoObject.OverlayFactory**

Фабрика интерактивных оверлеев. Создает следующие оверлеи для геометрий:

- **geometry.Point** — оверлей **overlay.html.Placemark**.
- **geometry.LineString** — оверлей **overlay.interactiveGraphics.Polyline**.
- **geometry.Polygon** — оверлей **overlay.interactiveGraphics.Polygon**.
- **geometry.Rectangle** — оверлей **overlay.interactiveGraphics.Rectangle**.
- **geometry.Circle** — оверлей **overlay.interactiveGraphics.Circle**.

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	**util.Storage**	Добавляет объект в хранилище.
createOverlay(pixelGeometry, data, options)	**IOverlay**	Создает и возвращает экземпляр оверлея на основе типа пиксельной геометрии.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом или исходный key, в случае, если это не строка.

Имя	Возвращает	Описание
remove(key)	**util.Storage**	Удаляет пару "ключ: значение" из хранилища.

geoObject.overlayFactory.interactiveGraphics

Входит в пакеты package.editor, package.full, package.geoObjects.

Статический объект.

Экземпляр класса **geoObject.OverlayFactory**

Фабрика интерактивных графических оверлеев. Создает следующие оверлеи для геометрий:

- **geometry.Point** — оверлей **overlay.interactiveGraphics.Placemark**.
- **geometry.LineString** — оверлей **overlay.interactiveGraphics.Polyline**.
- **geometry.Polygon** — оверлей **overlay.interactiveGraphics.Polygon**.
- **geometry.Rectangle** — оверлей **overlay.interactiveGraphics.Rectangle**.
- **geometry.Circle** — оверлей **overlay.interactiveGraphics.Circle**.

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	**util.Storage**	Добавляет объект в хранилище.
createOverlay(pixelGeometry, data, options)	**IOverlay**	Создает и возвращает экземпляр оверлея на основе типа пиксельной геометрии.
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.

geoObject.overlayFactory.staticGraphics

Входит в пакеты package.editor, package.full, package.geoObjects.

Статический объект.

Экземпляр класса **geoObject.OverlayFactory**

Фабрика неинтерактивных графических оверлеев. Создает следующие оверлеи для геометрий:

- **geometry.Point** — оверлей **overlay.staticGraphics.Placemark**.
- **geometry.LineString** — оверлей **overlay.staticGraphics.Polyline**.
- **geometry.Polygon** — оверлей **overlay.staticGraphics.Polygon**.
- **geometry.Rectangle** — оверлей **overlay.staticGraphics.Rectangle**.
- **geometry.Circle** — оверлей **overlay.staticGraphics.Circle**.

Методы

Методы

Имя	Возвращает	Описание
add(key, object)	**util.Storage**	Добавляет объект в хранилище.
createOverlay(pixelGeometry, data, options)	**IOverlay**	Создает и возвращает экземпляр оверлея на основе типа пиксельной геометрии.
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.

geoObject.overlayFactory.storage

Входит в пакеты package.editor, package.full, package.geoObjects.

Статический объект.

Экземпляр класса **util.Storage**

Хранилище фабрик оверлеев геообъектов. Фабрики оверлеев позволяют определять тип отображения геообъектов. Список доступных фабрик оверлеев:

- 'default#interactive' — фабрика интерактивных оверлеев **geoObject.overlayFactory.interactive**.
- 'default#interactiveGraphics' — фабрика интерактивных графических оверлеев **geoObject.overlayFactory.interactiveGraphics**.
- 'default#staticGraphics' — фабрика неинтерактивных графических оверлеев **geoObject.overlayFactory.staticGraphics**.
- 'default#hotspot' — оверлеев, отображающихся как активные области **geoObject.overlayFactory.hotspot**.

Методы

Пример:

```
// Отображает интерактивную метку с использованием графической подсистемы.
var marker = new ymaps.Placemark([37, 40], {
    hintContent: "hint",
    balloonContent: "balloon"
}, {
    preset: "twirl#greenIcon",
    overlayFactory: "default#interactiveGraphics"
});
myMap.geoObjects.add(marker);
```

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.

geoObject.OverlayFactory

Расширяет **IOverlayFactory**, **util.Storage**.

Фабрика оверлеев для геообъекта.

Конструктор | Методы

Конструктор

```
geoObject.OverlayFactory()
```

Методы

Имя	Возвращает	Описание
add(key, object)	**util.Storage**	Добавляет объект в хранилище. Унаследован от **util.Storage**.
createOverlay(pixelGeometry, data, options)	**IOverlay**	Создает и возвращает экземпляр оверлея на основе типа пиксельной геометрии. Унаследован от **IOverlayFactory**.
get(key)	Object	Возвращает объект, хранящийся под указанным ключом или исходный key, в случае, если это не строка. Унаследован от **util.Storage**.
remove(key)	**util.Storage**	Удаляет пару "ключ: значение" из хранилища. Унаследован от **util.Storage**.

GeoObjectArray

Входит в пакеты package.editor, package.full, package.geoObjects, package.standard.

Расширяет **IGeoObject**, **IMapObjectCollection**.

Упорядоченная группа геообъектов на базе массива. Обеспечивает быстрые операции на основе порядковых индексов, но поиск объектов в данной группе медленнее чем в коллекции на основе двусвязного списка. Смотрите также **GeoObjectCollection**.

Конструктор | Поля | События | Методы

Конструктор

```
GeoObjectArray(feature, options)
```

Создает упорядоченную группу геообъектов.

Параметры:

Параметр	Свойства параметра	Тип	Описание
feature		Object	Описание группы. Функционально соответствует описанию геообъекта. Смотрите **GeoObject**.
	geometry	**IGeometry**|Object	Геометрия группы.
	properties	**IDataManager**|Object	Данные группы.
options		Object	Опции группы. Здесь можно задавать все опции описанные в классе **GeoObject**. Значения опций будут применяться как к самой группе, так и к ее дочерним объектам, если для них данные опции не заданы.

Пример:

```
// Создаем группу геообъектов и задаем опции.
var myGroup = new ymaps.GeoObjectArray({}, {
    preset: "twirl#airplaneIcon",
    strokeWidth: 4,
    geodesic: true
});

// Добавляем в группу метки и линию.
myGroup.add(new ymaps.Placemark([13.38, 52.51]));
myGroup.add(new ymaps.Placemark([30.30, 50.27]));
myGroup.add(new ymaps.Polyline([[13.38, 52.51], [30.30, 50.27]]));

// Добавляем группу на карту.
myMap.geoObjects.add(myGroup);
// Устанавливаем карте центр и масштаб так, чтобы охватить группу целиком.
myMap.setBounds(myGroup.getBounds());
```

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
geometry	IGeometry	Геометрия геообъекта. Унаследовано от IGeoObject.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.
properties	IDataManager	Менеджер данных геообъекта. Унаследовано от IGeoObject.
state	IDataManager	Состояние геообъекта. Определяется следующими полями: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что геообъект перетаскивается. Унаследовано от IGeoObject.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от ICollection.
boundschange	Изменение координат географической области, охватывающей группу и все ее дочерние геообъекты. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldBounds — предыдущая ограничивающая область; • newBounds — новая ограничивающая область.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от IGeoObject.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от **IGeoObject**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
pixelboundschange	Изменение пиксельных координат области, охватывающей группу и все ее дочерние геообъекты. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldPixelBounds — предыдущая ограничивающая область; • newPixelBounds — новая ограничивающая область.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от IGeoObject.
propertieschange	Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от IGeoObject.
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
add(child, index)	GeoObjectArray	Добавляет геообъект в группу.
each(callback, context)		Перебирает все элементы группы, вызывая для каждого функцию-обработчик.
get(index)	IGeoObject|null	Возвращает ссылку на дочерний геообъект, либо null, если объекта с заданным индексом нет в группе.
getBounds()	Number[][]|null	Возвращает географические координаты области, охватывающей группу и все ее дочерние геообъекты.
getIterator()	IIterator	Возвращает итератор по группе.
getLength()	GeoObjectArray	Возвращает количество геообъектов в группе.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getOverlay()	IOverlay|null	Возвращает оверлей геообъекта, либо null. Унаследован от IGeoObject.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
getPixelBounds()	Number[][]|null	Возвращает глобальные пиксельные координаты области, охватывающей группу и все ее дочерние геообъекты.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
indexOf(object)	Integer	Возвращает индекс геообъекта в группе или -1, если объект не найден.
remove(child)	**GeoObjectArray**	Удаляет геообъект из группы.
removeAll()	**GeoObjectArray**	Удаляет из группы все геообъекты.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.
splice(index, number)	**IGeoObject**[]	Удаляет геообъекты из группы. Если необходимо, вставляет на их место другие. Объекты, которые будут вставлены вместо удаляемых, передаются как дополнительные параметры (после параметра number).

Описание событий

boundschange

```
boundschange
```

Изменение координат географической области, охватывающей группу и все ее дочерние геообъекты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldBounds — предыдущая ограничивающая область;
- newBounds — новая ограничивающая область.

pixelboundschange

```
pixelboundschange
```

Изменение пиксельных координат области, охватывающей группу и все ее дочерние геообъекты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldPixelBounds — предыдущая ограничивающая область;
- newPixelBounds — новая ограничивающая область.

Описание методов

add

```
{GeoObjectArray} add(child, index)
```

Добавляет геообъект в группу.

Возвращает ссылку на группу.

Параметры:

Параметр	Тип	*	Описание
child	**IGeoObject**	✔	Дочерний геообъект.
index	Integer		Индекс, в который добавляется новый объект. Если индекс не задан, то объект будет добавлен в конец группы.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

each

```
each(callback, context)
```

Перебирает все элементы группы, вызывая для каждого функцию-обработчик.

Параметры:

Параметр	Тип	*	Описание
callback	Function	✔	Функция-обработчик.
context	Object		Контекст выполнения функции.

* Обязательный параметр/опция.

Пример:

```
// Отображаем индекс геообъекта в группе как содержимое его иконки.
myGroup.events.add(["add", "remove"], function () {
    this.each(function (el, i) {
        el.properties.set("iconContent", i);
    })
}, myGroup);
```

get

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

Возвращает ссылку на дочерний геообъект, либо null, если объекта с заданным индексом нет в группе.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс дочернего геообъекта.

* Обязательный параметр/опция.

getBounds

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

Возвращает географические координаты области, охватывающей группу и все ее дочерние геообъекты.

getIterator

```
{IIterator} getIterator()
```

Возвращает итератор по группе.

getLength

```
{GeoObjectArray} getLength()
```

Возвращает количество геообъектов в группе.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getPixelBounds

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

Возвращает глобальные пиксельные координаты области, охватывающей группу и все ее дочерние геообъекты.

indexOf

```
{Integer} indexOf(object)
```

Возвращает индекс геообъекта в группе или -1, если объект не найден.

Параметры:

Параметр	Тип	*	Описание
object	Object	✔	Геообъект.

* Обязательный параметр/опция.

remove

```
{GeoObjectArray} remove(child)
```

Удаляет геообъект из группы.

Возвращает ссылку на группу.

Параметры:

Параметр	Тип	*	Описание
child	IGeoObject	✔	Дочерний геообъект.

* Обязательный параметр/опция.

Пример:

```
// При клике по геообъекту удаляем его из группы.
myGroup.events.add("click", function (e) {
    if (e.get("target").getParent() == this) {
        this.remove(e.get("target"));
    }
}, myGroup);
```

removeAll

```
{GeoObjectArray} removeAll()
```

Удаляет из группы все геообъекты.

Возвращает ссылку на группу.

splice

```
{IGeoObject[]} splice(index, number)
```

Удаляет геообъекты из группы. Если необходимо, вставляет на их место другие. Объекты, которые будут вставлены вместо удаляемых, передаются как дополнительные параметры (после параметра number).

Возвращает массив удаленных геообъектов.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс геообъекта, с которого начинается удаление.
number	Integer	✔	Количество удаляемых геообъектов.

* Обязательный параметр/опция.

Пример:

```
// Удаляет второй объект.
myGroup.splice(1, 1);
// Вставляет новый объект obj во вторую позицию.
myGroup.splice(1, 0, obj);
// Заменяет второй объект на новый объект obj.
myGroup.splice(1, 1, obj);
```

GeoObjectCollection

Входит в пакеты package.editor, package.full, package.geoObjects, package.standard.

Расширяет **IGeoObject**, **IMapObjectCollection**.

Коллекция геообъектов на базе двусвязного списка. Обеспечивает быстрое добавление и удаление объектов, но не поддерживает работу с порядковыми индексами. Смотрите также **GeoObjectArray**.

Конструктор | Поля | События | Методы

Конструктор

```
GeoObjectCollection(feature, options)
```

Создает коллекцию геообъектов.

Параметры:

Параметр	Свойства параметра	Тип	Описание
feature		Object	Описание коллекции. Функционально соответствует описанию геообъекта. Смотрите **GeoObject**.
	geometry	**IGeometry**|Object	Геометрия коллекции.
	properties	**IDataManager**|Object	Данные коллекции.
options		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 заданы.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Создаем коллекцию геообъектов.
var myCollection = new ymaps.GeoObjectCollection();

// Добавляем метки в коллекцию.
myCollection.add(new ymaps.Placemark([37.61, 55.75]));
myCollection.add(new ymaps.Placemark([13.38, 52.51]));
myCollection.add(new ymaps.Placemark([30.30, 50.27]));

// При наведении на одну из меток подсвечиваем коллекцию целиком.
myCollection.events
    .add("mouseenter", function () {
        myCollection.options.set("preset", "twirl#redIcon");
    })
    .add("mouseleave", function () {
        myCollection.options.unset("preset");
    });

// Добавляем коллекцию на карту.
myMap.geoObjects.add(myCollection);
// Устанавливаем карте центр и масштаб так, чтобы охватить коллекцию целиком.
myMap.setBounds(myCollection.getBounds());
```

Поля

Имя	Тип	Описание
events	**IEventManager**	Менеджер событий. Унаследовано от **IDomEventEmitter**.
geometry	**IGeometry**	Геометрия геообъекта. Унаследовано от **IGeoObject**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
properties	**IDataManager**	Менеджер данных геообъекта. Унаследовано от **IGeoObject**.
state	**IDataManager**	Состояние геообъекта. Определяется следующими полями: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что геообъект перетаскивается. Унаследовано от **IGeoObject**.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от **ICollection**.
boundschange	Изменение координат географической области, охватывающей коллекцию и все ее дочерние геообъекты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldBounds — предыдущая ограничивающая область; • newBounds — новая ограничивающая область.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от IDomEventEmitter.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от IGeoObject.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
pixelboundschange	Изменение пиксельных координат области, охватывающей коллекцию и все ее дочерние геообъекты. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldPixelBounds — предыдущая ограничивающая область; • newPixelBounds — новая ограничивающая область.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от IGeoObject.
propertieschange	Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от IGeoObject.
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
add(child)	GeoObjectCollection	Добавляет геообъект в коллекцию.
each(callback, context)		Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getBounds()	Number[][]|null	Возвращает географические координаты области, охватывающей коллекцию и все ее дочерние геообъекты.
getIterator()	**IIterator**	Возвращает итератор по коллекции.
getLength()	Integer	Возвращает количество геообъектов в коллекции.
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getOverlay()	**IOverlay**|null	Возвращает оверлей геообъекта, либо null. Унаследован от **IGeoObject**.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
getPixelBounds()	Number[][]|null	Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и все ее дочерние геообъекты.
remove(child)	**GeoObjectCollection**	Удаляет геообъект из коллекции.
removeAll()	**GeoObjectCollection**	Удаляет из коллекции все геообъекты.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

Описание событий

boundschange

```
boundschange
```

Изменение координат географической области, охватывающей коллекцию и все ее дочерние геообъекты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldBounds — предыдущая ограничивающая область;
- newBounds — новая ограничивающая область.

pixelboundschange

```
pixelboundschange
```

Изменение пиксельных координат области, охватывающей коллекцию и все ее дочерние геообъекты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldPixelBounds — предыдущая ограничивающая область;
- newPixelBounds — новая ограничивающая область.

Описание методов

add

```
{GeoObjectCollection} add(child)
```

Добавляет геообъект в коллекцию.

Возвращает ссылку на коллекцию.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
child	IGeoObject	✔	Дочерний геообъект.

* Обязательный параметр/опция.

each

```
each(callback, context)
```

Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Параметры:

Параметр	Тип	*	Описание
callback	Function	✔	Функция-обработчик.
context	Object		Контекст выполнения 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:

```
// Ищем в коллекции геообъект с геометрией "Ломаная линия".
var iterator = myGroup.getIterator(),
    object;
while (object = iterator.getNext()) {
    if (object.geometry.getType() == "LineString") {
        break;
    }
}
```

getLength

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

Возвращает количество геообъектов в коллекции.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getPixelBounds

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

Возвращает глобальные пиксельные координаты области, охватывающей коллекцию и все ее дочерние геообъекты.

remove

```
{GeoObjectCollection} remove(child)
```

Удаляет геообъект из коллекции.

Возвращает ссылку на коллекцию.

Параметры:

Параметр	Тип	*	Описание
child	IGeoObject	✔	Дочерний геообъект.

* Обязательный параметр/опция.

removeAll

```
{GeoObjectCollection} removeAll()
```

Удаляет из коллекции все геообъекты.

Возвращает ссылку на коллекцию.

geoQuery

Входит в пакеты package.full, package.geoQuery.

Статическая функция.

Формирует набор данных из указанного источника и создает на его основе экземпляр GeoQueryResult.

Возвращает: Результат, содержащий данные из источника. (тип **GeoQueryResult**).

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
source	Object	✔	Источник геообъектов: • IGeoObject — объект, реализующий соответствующий интерфейс; • IGeoObject[] — массив объектов, реализующих соответствующий интерфейс; • ICollection — коллекция объектов, реализующих интерфейс IGeoObject; • ICollection[] — массив коллекций объектов, реализующих интерфейс IGeoObject; • util.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.

* 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Создание GeoQueryResult из util.Promise.
var result = ymaps.geoQuery(ymaps.geocode('Сыромятнический
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]]
                    }
                },
                {
                    type: 'FeatureCollection',
                    features: [
                        {
                            type: 'Feature',
                            geometry: {
                                type: 'Point',
                                coordinates: [12, 41]
                            },
                            properties: {
                              name: 'point'
                            },
                            options: {
                                preset: 'twirl#yellowIcon'
                            }
                        }
                    ]
                }
            ]
      });
// Неточечные объекты добавим 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

Входит в пакеты package.full, package.geoQuery.

Расширяет **IPromiseProvider**.

Результат геозапроса.

Конструктор | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
GeoQueryResult(source)
```

Параметры:

Параметр	Тип	*	Описание
source	Object	✔	Источник геообъектов: • 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.

* Обязательный параметр/опция.

Примеры:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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 = geoQuery(myMap.geoObjects).searchIntersect(myMap);
alert("Количество геообъектов в видимой области карты: " + result.getLength());
```

4.

```
// Создание GeoQueryResult из util.Promise.
var result = ymaps.geoQuery(ymaps.geocode('Сыромятнический
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: 'twirl#yellowIcon'
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
                                }
                            }
                        ]
                    }
                ]
        });
// Неточечные объекты добавим 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());
```

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)	**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)	**GeoQueryResult**	Возвращает ссылку на себя.
get(index)	**IGeoObject**	Метод 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.
getClosestTo(object, radius)	**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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getExtreme(key)	Number	Метод, возвращающий 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[]	Метод, возвращающий 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)	Number	Метод, вычисляющий максимальный коэффициент масштабирования, при котором в видимую область карты попадают все объекты.
indexOf(item)	Number	Возвращает индекс элемента в выборке. Если элемент не найден, возвращается -1.
intersect(result)	**GeoQueryResult**	Метод создает новую выборку, содержащую общие элементы для двух других выборок.
isReady()	Boolean	Метод, возвращающий признак готовности выборки.
map(callback, context)	**GeoQueryResult**	Метод, вызывающий метод callback для всех элементов выборки и формирующий новую выборку на основе полученных результатов.
remove()	**GeoQueryResult**	Удаляет объекты из результата. Не изменяет исходный объект, а создает новый, содержащий результирующий набор геообъектов.
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.
removeFrom(collection)	**GeoQueryResult**	Метод для удаления объектов из коллекции.
removeFromMap(map)	**GeoQueryResult**	Метод для удаления объектов с карты.
reverse()	**GeoQueryResult**	Переставляет элементы выборки в обратном порядке и возвращает новую выборку.
search(condition)	**GeoQueryResult**	Метод для поиска объектов выбоки, удовлетворяющих условиям.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
searchContaining(object)	**GeoQueryResult**	Метод, создающий новую выборку из объектов, содержащих указанный объект. Обратите внимание, что многие геообъекты требуют добавления на карту для корректных расчетов.
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 properties 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, raduis)	**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(onResolve, onReject)	**GeoQueryResult**	Подписка на "обещание".
unsetOptions(keys)	**GeoQueryResult**	Метод, позволяющий обнулять значение опций всем элементам выборки.
unsetProperties(path)	**GeoQueryResult**	Метод позволяющий обнулить значение поля properties 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
source	Object	✔	Источник геообъектов: • IGeoObject — объект, реализующий соответствующий интерфейс; • IGeoObject[] — массив объектов, реализующих соответствующий интерфейс; • ICollection — коллекция объектов, реализующих интерфейс IGeoObject; • ICollection[] — массив коллекций объектов, реализующих интерфейс IGeoObject; • util.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.

* Обязательный параметр/опция.

Примеры:

1.

```
// Добавление в GeoQueryResult одиночного геообъекта.
var placemark = new ymaps.Placemark([34, 56]);
myGeoQueryResult.add(placemark);
```

2.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// 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.getLenght());
    // 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),
    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	Тип	*	Описание
events	String|String[]	✔	Тип или массив типов событий, на которые совершается подписка.
callback	Function	✔	Функция-обработчик.
context	Object	✔	Контекст выполнения функции-обработчика.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
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	Тип	*	Описание
collection	ICollection	✔	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	Тип	*	Описание
map	Map	✔	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)
```

Метод, позволяющий установить видимую область карты так, чтобы видны были все объекты из выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
map	Map	✔	Карта.

* Обязательный параметр/опция.

Примеры:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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(myMap);
// 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(myMap);
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	Тип	Описание
options	Object	Опции, передаваемые в конструктор объекта **Clusterer**.

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)
```

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
callback	Function	✔	Функция-обработчик. Вызывается для каждого элемента выборки и получает его на вход.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Скроем красные метки в видимой области карты.
ymaps.geoQuery(placemarks).searchIntersect(myMap).each(function(pm) {
    if (pm.options.get('preset') == 'twirl#redIcon') {
        myMap.geoObjects.remove(pm);
    }
});
```

get

```
{IGeoObject} get(index)
```

Метод возвращает элемент выборки по индексу.

Возвращает объект.

Параметры:

Параметр	Тип	*	Описание
index	Number	✔	Индекс элементов.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример при синхронной обработке.
var result = ymaps.geoQuery(placemarks).sort('lat'),
    // Самый южный объект.
    southObject = result.get(0),
    // Самый северный объект.
    northObject = rosult.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.
        northObject = result.get(result.getLength() - 1);
});
```

getBounds

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

Возвращает географические координаты области, охватывающей объекты результата.

Пример:

```
// Устанавливаем центр и масштаб карты так, чтобы отобразить весь результат
целиком.
myMap.setBounds(myResult.getBounds());
```

getCenter

```
{Number[]} getCenter(map)
```

Метод, возвращающий центр области, охватывающей объекты результата, в географических координатах.

Возвращает координаты центра области в географических координатах.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
map	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 ссылка на геообъект или null в случае пустой выборки.

Параметры:

Параметр	Тип	*	Описание
map	Map	✔	Карта.

* Обязательный параметр/опция.

Пример:

```
// Пример работы при синхронных операциях.
// Откроем балун у объекта, ближайшего к центру видимой области карты.
ymaps.geoQuery(objects).getCentralObject(myMap).balloon.open();
```

getClosestTo

```
{IGeoObject|null} getClosestTo(object, radius)
```

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
object	Object	✔	Объект, относительно которого будет производится поиск. Может принимать следующие значения: • IGeoObject — объект, реализующий интефейс **IGeoObject**; • IGeometry — объект, реализующий интерфейс **IGeometry**; • Map — карта. В этом случае опорным объектом будет прямоугольная граница карты; • Number[] — координаты точки; • Object — Json-описание геометрии. Содержит поля type и coordinates. При описании круга также необходимо поле radius. При описании многоугольника можно указывать необязательное поле fillRule.
radius	Number		Радиус окружности в метрах.

* 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. Map.
// Найдем объект, ближайший к границе видимой области карты.
var edgeObject = result.getClosestTo(myMap);

// 4. Объект, ближайщий к точке.
var closestObject = result.getClosestTo([34, 53]);
```

2.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Пример использования при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('Париж')).addToMap(myMap);
// Дождемся 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 найдется.
    if (closestObject) {
        closestObject.balloon.open();
    }
});
```

getExtreme

```
{Number} getExtreme(key)
```

Метод, возвращающий максимальные и минимальные значения координат среди координат объектов выборки.

Возвращает соответствующее значение координаты.

Параметры:

Параметр	Тип	*	Описание
key	String	✔	Ключ для получения данных. Может принимать следующие значения: • top; • right; • bottom; • left.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример работы при синхронных операциях.
alert('Самая северная координата: ',
ymaps.geoQuery(myMap.geoObjects).getExtreme('top'));
```

2.

```
// Пример работы при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('Новгород'));
// Дождемся 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
key	String	✔	Ключ для поиска объекта. Может принимать следующие значения: • 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 найдется.
    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	Тип	*	Описание
map	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)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Метод, возвращающий центр области, охватывающей объекты результата, в глобальных пиксельных координатах для текущего зума карты.

Возвращает координаты центра области в глобальных пиксельных координатах.

Параметры:

Параметр	Тип	*	Описание
map	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 при синхронных операциях.
// Поиск элементов, на которые попадают координаты клика.
myMap.events.add('click', function (event) {
    var iterator = ymaps.geoQuery(myMap.geoObjects)
            .searchContaining(event.getCoordinates())
            .getIterator(),
        obj;
   while (obj = iterator.getNext()) {
       // 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(),
        obj;
    while (obj = iterator.getNext()) {
        // Совершаем необходимые действия над геообъектом.
    }

});
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getLength

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

Возвращает количество элементов в результате.

Примеры:

1.

```
var result = ymaps.geoQuery(myMap.geoObject).searchIntersect(myPolygon);
alert('Количество геообъектов, пересекающих многоугольник: ' +
result.getLenght());
```

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)
```

Метод, вычисляющий максимальный коэффициент масштабирования, при котором в видимую область карты попадают все объекты.

Возвращает максимальный коэффициент масштабирования карты.

Параметры:

Параметр	Тип	*	Описание
map	Map	✔	Карта, для которой ведется расчет.

* 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);
```

indexOf

```
{Number} indexOf(item)
```

Возвращает индекс элемента в выборке. Если элемент не найден, возвращается -1.

Параметры:

Параметр	Тип	*	Описание
item	IGeoObject	✔	Искомый объект.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// 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 новая выборка, содержащая результат пересечения.

Параметры:

Параметр	Тип	*	Описание
result	GeoQueryResult	✔	Выборка, с которой пересекают исходную.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример пересечения при синхронных операциях.
var result = ymaps.geoQuery(placemarks),
    greenObjects = result.search('properties.color="green"'),
    roundObjects = result.search('properties.shape="round"'),
    greenRoundObjects = greenObjects.intersect(roundObjects);
alert('Количество круглых зеленых объектов: ' + greenRoundObjects.getLength());
```

2.

```
// Пример пересечения при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('Ивановка')),
    filteredByLat = result.search('lat > 56'),
    filteredByLong = result.search('long > 36'),
    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()
```

Метод, возвращающий признак готовности выборки.

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.
}
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
callback	Function	✔	Функция-обработчик. Принимает на вход элемент выборки. Возвращает экземпляр IGeoObject.
context	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(myMap),
// Также добавим на карту метки, обозначающие центры окружностей.
    centers = circlesResult.map(function (object) {
        return new ymaps.Placemark(object.geometry.getCenter());
    }).addToMap(myMap);
```

remove

```
{GeoQueryResult} remove()
```

Удаляет объекты из результата. Не изменяет исходный объект, а создает новый, содержащий результирующий набор геообъектов.

Возвращает новый объект с результирующим набором геообъектов.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
		Object	✔	Объекты могут быть представлены в разном виде: • IGeoObject — объект, реализующий соответствующий интерфейс; • IGeoObject[] — массив объектов, реализующих соответствующий интерфейс; • ICollection — коллекция объектов, реализующих интерфейс IGeoObject; • ICollection[] — массив коллекций объектов, реализующих интерфейс IGeoObject; • GeoQueryResult — объект класса GeoQueryResult. Обратите внимание, что при асинхронных операциях нужно дождаться готовности результата для его корректного удаления; • util.Promise — объект класса util.Promise. Должен быть зарезовлен массивом геообъектов или объектом с полем geoObjects.

* Обязательный параметр/опция.

Примеры:

1.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
var objects = [
        new ymaps.Placemark([34, 56]),
        new ymaps.Rectangle([[34, 56], [36, 57]])
    ],
    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])
    ],
    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 из GeoQueryResult коллекции геообъектов.
// 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"'),
    result2 = ymaps.geoQuery(placemarks).search('properties.shape="circle"'),
    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	Тип	*	Описание
events	String|String[]	✔	Тип или массив типов событий, на которые была сделана подписка.
callback	Function	✔	Функция-обработчик, которую указали при подписке.
context	Object	✔	Контекст, который был указан при подписке.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
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('cl
ick', callback);
```

removeFrom

```
{GeoQueryResult} removeFrom(collection)
```

Метод для удаления объектов из коллекции.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
collection	ICollection	✔	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	Тип	*	Описание
map	Map	✔	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.
    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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Использование при синхронных запросах.
var result = ymaps.geoQuery(myMap.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, можно начинать пользоваться.
});
```

search

```
{GeoQueryResult} search(condition)
```

Метод для поиска объектов выбоки, удовлетворяющих условиям.

Возвращает новую выборку, содержащую результаты поиска.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
condition	String|Function	✔	Строковый шаблон для поиска или функция-фильтр. Строковый шаблон имеет структуру "<fieldname> <condition> <expression>". Возможные значения <fieldname>: • '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 geometry.getType(); • 'geometry.coordinates.<index>' — координаты, результат выполнения метода geometry.getCoordinates(). Для получения доступа к координате следует указывать ее индексы через точку — 'geometry.coordinates.0.1'; • 'properties.<path>' — значение поля данных; • 'options.<key>' — значение опции.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
			Возможные значения <condition>: • '>'; • '>='; • '==', '='; • '!='; • '<='; • '<'; • 'rlike', 'regexp' — соответствие значения <fieldname> некоторому регулярному выражению. Возможные значения <expression>: • Число; • Строка — обязательно указывается в кавычках; • true; • false; • null; • undefined. Функция-фильтр принимает на вход объект из выборки и возвращает true/false. True — объект попадет в результирующую выборку, false — не попадет.

* Обязательный параметр/опция.

Примеры:

1.

```
// Пример поиска объектов при синхронных операциях.
var result = ymaps.geoQuery(myMap.geoObjects);

     // Поиск объектов с определенным типом геометрии. Обратите внимание, что
// значение поля указано в кавычках - это строка.
result.search('geomtry.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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// 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(myMap)
        .applyBoundsToMap(myMap);
});
```

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	Тип	*	Описание
object	Object	✔	Объект, относительно которого будет производится поиск. Может принимать следующие значения: • IGeoObject — объект, реализующий интефейс **IGeoObject**; • IGeometry — объект, реализующий интерфейс **IGeometry**; • Map — карта. В этом случае опорным объектом будет прямоугольная граница карты; • Number[] — координаты точки. Если передан один параметр, трактуется как точка. Если передано два аргумента, трактуется как центр окружности; • Number[][] — координаты прямоугольной области; • Object — Json-описания геометрии.

* Обязательный параметр/опция.

Примеры:

1.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// 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(myMap);
// 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(myMap);
    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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
object	Object	✔	Объект, относительно которого будет производится поиск. Может принимать следующие значения: • IGeoObject — объект, реализующий интефейс **IGeoObject**; • IGeometry — объект, реализующий интерфейс **IGeometry**; • Map — карта. В этом случае опорным объектом будет прямоугольная граница карты; • Object — Json-описание геометрии. Содержит поля type и coordinates. При описании круга также необходимо поле radius. При описании многоугольника можно указывать необязательное поле fillRule; • Number[][] — координаты прямоугольной области.

* 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 objectsInsidePolygon = result.searchInside(polygon);

// 2. IGeometry.
var objectsInsideGeometry = result.searchInside(polygon.geometry);

// 3. Map.
// Найдем объекты, целиком лежащие в видимой области карты.
var visibleObject = result.searchInside(myMap);
```

2.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Пример использования при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('Иваново'))
        .setOptions('visible', false)
        .addToMap(myMap);
// Дождемся 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
`object`		Object	✔	Объект, относительно которого будет производится поиск. Может принимать следующие значения: • IGeoObject — объект, реализующий интефейс **IGeoObject**; • IGeometry — объект, реализующий интерфейс **IGeometry**; • Map — карта. В этом случае опорным объектом будет прямоугольная граница карты; • Number[] — координаты точки. Если передан один параметр, трактуется как точка. Если передано два аргумента, трактуется как центр окружности; • Number[][] — координаты прямоугольной области; • Object — Json-описание геометрии. Содержит поля type и coordinates. При описании круга также необходимо поле radius. При описании многоугольника можно указывать необязательное поле fillRule.
`options`		Object		Опции.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	considerOccurance	Object		Флаг, показывающий, считаем ли мы пересечением, когда одна фигура полностью содержится в другой. Значение по умолчанию: true

* 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 objectsIntersectPolygon = result.searchIntersect(polygon);

// 2. IGeometry.
var objectsIntersectGeometry = result.searchIntersect(polygon.geometry);

// 3. Map.
// 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(myMap,
{considerOccurance: false});
```

2.

```
// Пример использования при асинхронных операциях.
var result = ymaps.geoQuery(ymaps.geocode('кафе'))
        .setOptions('visible', false)
        .addToMap(myMap);

// Дождемся ответа от сервера и покажем объекты, попадающие в видимую область
карты.
result.then(function () {
    result.searchIntersect(myMap).setOptions('visible', true);
});
```

setOptions

```
{GeoQueryResult} setOptions(key, value)
```

Метод, позволяющий задать значение опций всем элементам выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
key	String|Object	✔	Название опции или хэш с опциями и их значениями.
value	Object		Значение опции.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
path	String	✔	Имя поля, которому присваивается значение. Может содержать ".".
value	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	Тип	*	Описание
begin	Number	✔	Индекс начального элемента в срезе.
end	Number		Индекс элемента в выборке, на котором срез завершится. При этом последним в новом срезе будет элемент с индексом end-1.

* Обязательный параметр/опция.

Примеры:

1.

```
// Выполним срез при синхронной обработке.
var result = ymaps.geoQuery(map.geoObjects).slice(0, 10);
alert('Количество элементов в новой выборке:', result.getLength());
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

2.

```
// Взятие среза при асинхронной обработке.
var result = ymaps.geoQuery(ymaps.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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
`comparator`	String|Function	✔	Строка с шаблоном сортировки или функция-компаратор. Строка-шаблон может быть представлена в виде '<fieldname> [<order>=asc]'. Возможные значения <fieldname>: • 'lat' — широта. Поддерживается только для точеченых объектов; • 'lng', 'long' — долгота. Поддерживается только для точеченых объектов; • 'x' — глобальные пиксельные координаты по оси x. Поддерживается только для точеченых объектов; • 'y' — глобальные пиксельные координаты по оси y. Поддерживается только для точеченых объектов; • 'geometry.type' — тип геометрии, результат выполнения метода geometry.getType(); • 'geometry.coordinates.<index>' — координаты, результат выполнения метода geometry.getCoordinates(). Для получения доступа к координате следует указывать ее индексы через точку — 'geometry.coordinates.0.1'; • 'properties.<path>' — значение поля данных; • 'options.<key>' — значение опции. Возможные значения необязательного параметра <order>: • 'asc' — сортировка по возрастанию; • 'desc' — сортировка по убыванию.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
			Функция-компаратор принимает на вход два элемента выборки. Возвращаемые значения: Если первый объект больше второго, функция возвращает значение > 0; Если первый объект равен второму, функция возвращает 0; Если первый объект меньше второго, функция возвращает < 0.

* Обязательный параметр/опция.

Пример:

```
// Пример использования 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, raduis)
```

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
object	Object	✔	Объект, до которого будет считаться расстояние. Может принимать следующие значения: • IGeoObject — объект, реализующий интефейс **IGeoObject**; • IGeometry — объект, реализующий интерфейс **IGeometry**; • Map — карта. В этом случае опорным объектом будет прямоугольная граница карты; • Number[] — координаты точки; • Number[][] — координаты прямоугольной области; • Object — Json-описание геометрии. Содержит поля type и coordinates. При описании круга также необходимо поле radius. При описании многоугольника можно указывать необязательное поле fillRule.
raduis		✔	

* Обязательный параметр/опция.

Примеры:

1.

```
// Примеры использования метода при различных входных данных.
var result = ymaps.geoQuery(objects).addToMap(myMap);

// 1. IGeoObject.
var polyline = new ymaps.Polyline([[35, 65], [35, 66], [34, 62], [34, 63]]);
myMap.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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// 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(onResolve, onReject)
```

Подписка на "обещание".

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	Описание
onResolve	Function	Функция-обработчик, которая вызывается, если обещание выполнено.
onReject	Function	Функция-обработчик, которая вызывается, если обещание не выполнено (произошла ошибка).

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	Тип	*	Описание
keys	String|String	✔	Название или массив названий опций, которые должны быть аннулированы.

* Обязательный параметр/опция.

Пример:

```
result.unsetOptions('visible');
```

unsetProperties

```
{GeoQueryResult} unsetProperties(path)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Метод позволяющий обнулить значение поля properties у всех элементов выборки.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
path	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);
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

geoXml

geoXml.load

Входит в пакеты package.full, package.geoXml.

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 верхней коллекции геообъектов GPX-файла доступны следующие пресеты:

- 'gpx#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;
- 'gpx#plain' — элементы gpx-коллекции ведут себя как обычные геообъекты;

Возвращает: 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);

(тип **util.Promise**).

Параметры:

Параметр	Тип	*	Описание
url	String	✔	URL файла с географическими данными.

* Обязательный параметр/опция.

Пример:

```
// Создание и инициализация карты map...

// Загрузка и отображение ymapsml-файла с сервиса "Мои карты"
ymaps.geoXml.load('http://maps.yandex.ru/export/usermaps/
HNQ5uTUgbjy6L0dW2uReUjSoXb1Ad7jw/')
     .then(function (res) {
         // добавление элементов ymapsml-файла на карте
         map.geoObjects.add(res.geoObjects);
         // Выставление границ и типа к карты.
         // границы res.mapState.getBounds() применяются к карте асинхронно,
         // т.к. необходимо получить информацию о доступных масштабах для данных
границ
         // тип карты res.mapState.getType() применяется к карте синхронно
         if (res.mapState) {
             res.mapState.applyToMap(map).then(function () {
                 alert('К карте применены границы ' +
res.mapState.getBounds().toString());
             });
         }
         // если информация о границах не предоставлена в repr:View YMapsML
файла,
         // можно применить применить gml:boundedBy верхнего элемента
ymaps:GeoObjectCollection
         else if (res.geoObjects.properties.get('boundedBy')) {
             map.setBounds(res.geoObjects.properties.get('boundedBy'), {
                 checkZoomRange: true
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
            });
        }
    });

// Загрузка и отображение kml-файла
ymaps.geoXml.load('http://api.yandex.ru/maps/doc/jsapi/1.x/examples/kml/
demonstration.xml')
    .then(function (res) {
        map.geoObjects.add(res.geoObjects);
    });

// Загрузка и отображение gpx-файла
ymaps.geoXml.load('http://karmatsky.narod2.ru/MskChel2.xml')
    .then(function (res) {
        res.geoObjects.options.set({
            balloonContentBodyLayout: ymaps.templateLayoutFactory.createClass(
            // будем показывать в балуне только свойство name геообъекта и
скорость
            '$[properties.name] $[properties.velocity]'
            )
        });

        map.geoObjects.add(res.geoObjects);
        // метаданные о границах из gpx-файла 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
            });
        }
    });
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getZoomRange

Входит в пакеты package.full, package.map, package.standard.

Статическая функция.

Определяет доступный диапазон уровней масштабирования в указанной точке для указанного типа карты. Возвращает объект-promise, который будет подтверждён и получит в качестве параметра массив из двух чисел — максимальный и минимальный уровень масштабирования в данной точке.

Возвращает: Объект-promise. (тип **util.Promise**).

Параметры:

Параметр	Тип	*	Описание
mapType	String|**MapType**	✔	Тип карты. Строка-ключ из **mapType.storage**, либо экземпляр класса **MapType**.
coords	Number[]	✔	Координаты точки, для которой нужно узнать диапазон доступных уровней масштабирования.

* 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]
        });
    });
});
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

graphics

graphics.style

graphics.style.color

Статический объект.

Задает цвет графической фигуры в форматах #RGB,#RGBA,#RRGGBB,#RRGGBBAA,rgb(r,b,a),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 отображениея VML(IE<9) могут быть использованны 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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

solid

```
solid
```

Сплошная линия

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Hint

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geoXml, package.geocode, package.route, package.search, package.standard.

Расширяет **IDomEventEmitter**.

Хинт — всплывающая подсказка, которая может отображать любое HTML-содержимое. Хинты возвращаются из методов show менеджеров хинта (карты, геообъектов, слоёв активных областей и т.д.). Не создавайте их самостоятельно, если в этом нет необходимости.
Конструктор | Поля | События | Методы

Конструктор

```
Hint(map, data, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
map		**Map**	✔	Ссылка на объект карты.
data		Object		Данные.
options		Object		Опции.
	contentLayout	Function|String		Макет для содержимого хинта. (Тип: конструктор объекта с интерфейсом ILayout).
	fitPane	Boolean		Флаг, заставляющий хинт сдвигать свою позицию, чтобы не выходить за границы контейнера. Значение по умолчанию: true
	hideTimeout	Number		Задержка закрытия (в мс). Значение по умолчанию: 700
	holdByMouse	Boolean		Флаг, оменяющий закрытие хинта, находящегося под курсором. Значение по умолчанию: true
	interactivityModel	String		Ключ модели интерактивности. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**.
	layout	Function|String		Внешний макет хинта. (Тип: конструктор объекта с интерфейсом ILayout).

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	overlay	IOverlay		Класс оверлея. Значение по умолчанию: overlay.Label
	pane	IPane|String		Контейнер оверлея хинта. "outers" — вне карты, "floats" — в пределах карты. Значение по умолчанию: outers
	showTimeout	Number		Задержка открытия (в мс). Значение по умолчанию: 150
	zIndex	String		z-index подсказки.

* Обязательный параметр/опция.

Пример:

```
// Создаем независимый экземпляр хинта и показываем его в центре карты.
(new ymaps.Hint(myMap, { content: 'Hello' }, {
    projection: ymaps.{@link projection.wgs84Mercator},
    layout: ymaps.{@link templateLayoutFactory}.createClass('$[content]')
})).show(myMap.getCenter());
```

Поля

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

События

Имя	Описание
beforedestroy	Событие, предшествующее уничтожению хинта. Экземпляр класса Event.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
destroy	Уничтожение хинта. Экземпляр класса Event.
hide	Скрытие хинта. Экземпляр класса Event.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
show	Отображение хинта на карте. Экземпляр класса **Event**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy(callback, urgently)	**Hint**	Уничтожает хинт. Предварительно генерирует событие beforedestroy, при отмене которого уничтожения хинта не произойдет. После уничтожения хинта генерируется событие destroy.
getData()	Object	Возвращает данные хинта.
getOverlay()	**IOverlay**	Возвращает оверлей хинта.
getPosition()	Number[]	Возвращает позицию хинта.
hide(urgently)	**Hint**	Скрывает хинт.
isShown()	Boolean	Возвращает состояние хинта: открыт/скрыт.
setData(data)	**Hint**	Устанавливает данные хинта.
setPosition(position)	**Hint**	Устанавливает координаты хинта в проекции, заданной в опциях.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
show(position)	**Hint**	Отображает хинт на карте в точке с заданными координатами. Если хинт уже отображен, перемещает его в указанную точку.

Описание событий

beforedestroy

```
beforedestroy
```

Событие, предшествующее уничтожению хинта. Экземпляр класса **Event**.

destroy

```
destroy
```

Уничтожение хинта. Экземпляр класса **Event**.

hide

```
hide
```

Скрытие хинта. Экземпляр класса **Event**.

show

```
show
```

Отображение хинта на карте. Экземпляр класса **Event**.

Описание методов

destroy

```
{Hint} destroy(callback, urgently)
```

Уничтожает хинт. Предварительно генерирует событие beforedestroy, при отмене которого уничтожения хинта не произойдет. После уничтожения хинта генерируется событие destroy.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
`callback`	Function		Функция, которая будет вызвана после уничтожения балуна. Если событие beforedestroy было отменено, в функцию первым параметром будет передано сообщение о том, что хинт не был уничтожен.
`urgently`	Boolean	✔	Скрыть хинт сразу, без временной задержки и без учета положения курсора.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getData

```
{Object} getData()
```

Возвращает данные хинта.

getOverlay

```
{IOverlay} getOverlay()
```

Возвращает оверлей хинта.

getPosition

```
{Number[]} getPosition()
```

Возвращает позицию хинта.

hide

```
{Hint} hide(urgently)
```

Скрывает хинт.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
urgently	Boolean	✔	Скрыть хинт сразу, без временной задержки и без учета положения курсора.

* Обязательный параметр/опция.

isShown

```
{Boolean} isShown()
```

Возвращает состояние хинта: открыт/скрыт.

Возвращает true, если хинт открыт, false — в обратном случае.

Пример:

```
// Открываем хинт (по таймауту в 1 сек.):
var hint = myMap.hint.show(myMap.getCenter(), { content: 'loading' },
{ showTimeout: 1000 });

// Выполняем AJAX-запрос:
$.getJSON('hint-data.json')
    .success(function (data) {
        // Обновляем данные если хинт открыт, иначе вешаемся на событие
открытия:
        if (hint.isShown()) {
            hint.setData(data);
        } else {
            myMap.hint.events.add('show', function () {
                hint.setData(data);
            });
        }
    });
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

setData

```
{Hint} setData(data)
```

Устанавливает данные хинта.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
data	Object	✔	Данные.

* Обязательный параметр/опция.

setPosition

```
{Hint} setPosition(position)
```

Устанавливает координаты хинта в проекции, заданной в опциях.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
position	Number[]	✔	Новая позиция.

* Обязательный параметр/опция.

show

```
{Hint} show(position)
```

Отображает хинт на карте в точке с заданными координатами. Если хинт уже отображен, перемещает его в указанную точку.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	Описание
position	Number[]	Координаты хинта.

Пример:

```
// Показывает хинт на карте, а затем скрывает и вновь показывает его с интервалом
в 1 сек.
var hint = myMap.hint.show(myMap.getCenter(), 'Test');
setInterval(function () {
    hint.isShown() ? hint.hide() : hint.show(myMap.getCenter());
}, 1000);
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

hotspot

hotspot.Layer

Входит в пакеты package.full, package.standard.

Расширяет **IChildOnMap**, **ICustomizable**.

Слой активных областей.

Конструктор | Поля | События | Методы

Конструктор

```
hotspot.Layer(objectSource, options)
```

Создает слой активных областей. Каждая отдельная область в составе слоя представляет собой фигуру — **hotspot.Shape**.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
objectSource		IHotspotObjectSource	✔	Источник объектов слоя.
options		Object		Опции слоя. Опции для балунов **Balloon** и хинтов **Hint** хотспотного слоя нужно указывать с префиксами 'balloon' и 'hint'.
	cursor	String		Тип курсора. Значение по умолчанию: 'pointer'
	dontChangeCursor	Boolean		Опция, позволяющая запретить изменение курсора при наведении на объект слоя. По умолчанию курсоры изменяются. Значение по умолчанию: false
	hasBalloon	Boolean		Флаг, показывающий, есть ли у слоя балуны. Если значение флага false, у слоя не будет создано поле .balloon. Значение по умолчанию: true
	hasHint	Boolean		Флаг, показывающий, есть ли у слоя хинты. Если значение флага false, у слоя не будет создано поле .hint. Значение по умолчанию: true

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	`interactivityModel`	String		Модель интерактивности слоя. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**. Значение по умолчанию: 'default#layer'
	`openBalloonOnClick`	Boolean		Опция, позволяющая запретить открытие балуна при клике на хотспотном объекте. По умолчанию открытие балуна разрешено. Значение по умолчанию: true
	`openEmptyBalloon`	String		Открывать балун с пустым содержимым. Значение по умолчанию: false
	`showEmptyHint`	String		Показывать всплывающую подсказку с пустым содержимым. Значение по умолчанию: false
	`showHintOnHover`	Boolean		Опция, позволяющая запретить показ хинта при наведении на хотспотный объект. По умолчанию показ хинтов разрешен. Значение по умолчанию: true
	`zIndex`	Number		Значение zIndex слоя. Когда активный объект ищется в пределах одного слоя, то при одновременном попадании курсора в два объекта выбирается объект с наибольшими значением zIndex. При дальнейшей работе — например при определении, фигура какого слоя является более приоритетной, используется значение zIndex слоя.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Пример:

```
// Создаем источник хотспотных данных. Не указываем значение ключа,
// в результате чего 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 открывать балуны без содержимого.
        showEmptyBalloon: true
    });
```

Поля

Имя	Тип	Описание
balloon	hotspot.layer.Balloon	Объект для управления балунами слоя хотспотов.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter.
hint	hotspot.layer.Hint	Пространство имен для работы с хинтами слоя хотспотов.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.

События

Имя	Описание
addtomap	Слой добавлен на карту.
balloonclose	
balloonopen	
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
removefrommap	Слой удален с карты.

Методы

Имя	Возвращает	Описание
getObjectSource()	IHotspotObjectSource	Возвращает источник объектов слоя активных областей.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
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
```

Объект для управления балунами слоя хотспотов.

Пример:

```
hotspotLayer.options.set({
   // Запрещаем показ всплывающей подсказки по наведению.
   showHintOnHover: false,
   // Разрешаем открытие пустого балуна.
   openEmptyBalloon: true
});
// открываем балун по наведению на активный объект
hotspotLayer.events.add('mouseenter', function (e) {
   // Получаем объект, на который навели курсор.
   var activeShape = e.get('activeShape');
   hotspotLayer.balloon.open(
       e.get('globalPixelPosition'),
       // Данные для балуна получаем из объекта.
       activeShape.getProperties(),
       // Опции получаем из слоя хотспотов.
       hotspotLayer.options
   );
}, this);
```

hint

```
hotspot.layer.Hint hint
```

Пространство имен для работы с хинтами слоя хотспотов.

Пример:

```
hotspotLayer.options.set({
    // Запрещаем открытие балуна по клику.
    openBalloonOnClick: false,
    // Разрешаем показ пустых всплывающих подсказок.
    showEmptyHint: true
});
// Показываем хинт по клику на активном объекте.
hotspotLayer.events.add('click', function (e) {
   // Получаем объект, на котором произошел клик.
   var activeShape = e.get('activeShape');
   hotspotLayer.hint.show(
       e.get('globalPixelPosition'),
       // Данные для хинта получаем из объекта.
       activeShape.getProperties(),
       // Опции получаем из слоя хотспотов.
       hotspotLayer.options
   );
}, this);
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание событий

addtomap

```
addtomap
```

Слой добавлен на карту.

balloonclose

```
balloonclose
```

balloonopen

```
balloonopen
```

removefrommap

```
removefrommap
```

Слой удален с карты.

Описание методов

getObjectSource

```
{IHotspotObjectSource} getObjectSource()
```

Возвращает источник объектов слоя активных областей.

Пример:

```
// Обновление источника данных хотспотного слоя.
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.layer

hotspot.layer.Balloon

Менеджер балуна слоя активных областей.

Конструктор | События | Методы

Конструктор

```
hotspot.layer.Balloon(layer)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
layer	hotspot.Layer	✔	Слой, на котором ведется работа с балунами.

* Обязательный параметр/опция.

События

Имя	Описание
close	
open	

Методы

Имя	Возвращает	Описание
close()		Закрывает балун слоя хотспотов, если он открыт на карте.
isOpen()	Boolean	Проверяет, открыт ли балун хотспотного слоя на карте.
open(globalPixelPosition, data, options)	Balloon	Открывает балун в заданной точке.

Описание событий

close

```
close
```

open

```
open
```

Описание методов

close

```
close()
```

Закрывает балун слоя хотспотов, если он открыт на карте.

isOpen

```
{Boolean} isOpen()
```

Проверяет, открыт ли балун хотспотного слоя на карте.

Возвращает признак открытия балуна.

open

```
{Balloon} open(globalPixelPosition, data, options)
```

Открывает балун в заданной точке.

Возвращает ссылку на балун.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
globalPixelPosition	Number[]	✔	Пиксельные координаты точки, в которой требуется открыть балун.
data	Object		Данные балуна.
options	Object		Опции балуна.

* Обязательный параметр/опция.

hotspot.layer.Hint

Менеджер хинта слоя активных областей.

Конструктор | Методы

Конструктор

```
hotspot.layer.Hint(layer)
```

Параметры:

Параметр	Тип	*	Описание
layer	hotspot.Layer	✔	Слой активных областей.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
hide(callback, urgently)		Скрывает хинт.
isShown()	Boolean	Проверяет состояние хинта на карте.
show(globalPixelPosition, data, options)	Hint	Показывает хинт.

Описание методов

hide

```
hide(callback, urgently)
```

Скрывает хинт.

Параметры:

Параметр	Тип	*	Описание
callback	Function	✔	Функция, которая вызывается при закрытии балуна.
urgently	Boolean	✔	Спрятать хинт немедленно. При значении false хинт спрячется по таймауту.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

isShown

```
{Boolean} isShown()
```

Проверяет состояние хинта на карте.

Возвращает true, если хинт показывается, false — если спрятан.

show

```
{Hint} show(globalPixelPosition, data, options)
```

Показывает хинт.

Возвращает экземпляр хинта.

Параметры:

Параметр	Тип	*	Описание
globalPixelPosition	Number[]	✔	Позиция хинта в глобальных пиксельных координатах.
data	Object		Данные.
options	Object		Опции.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

hotspot.ObjectSource

Входит в пакеты package.full, package.standard.

Расширяет **IHotspotObjectSource**.

Стандартная реализация интерфейса **IHotspotObjectSource**. Работает со стандартным форматом ответа сервера.

Конструктор | Поля | События | Методы

Конструктор

```
hotspot.ObjectSource(tileUrlTemplate, keyTemplate|Function, options)
```

Возвращает источник данных для слоя активных областей.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
tileUrlTemplate		String|Function	✔	Шаблон URL данных, соответствующих тайлу. Поддерживаются специальные конструкции и возможность использования функции по аналогии с **Layer**.
keyTemplate|Function		String		Шаблон идентификатора тайла. Используется для формирования padding в jsonp-запросе за данными. Задается по аналогии с предыдущим параметром. Все символы, не являющиеся буквой, цифрой или символом '_',

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
				будут заменены на '_'. Если параметр не задан, то padding запроса будет сгенерирован автоматически. Примеры преобразований при tileNumber=[3, 1], zoom=9: • 'myCallback=%x' — 'myCallback_3' • '%c' — 'x_3_y_1_z_9' • 'callback2_%c' — 'callback2_x_3_y_1_z_9' • 'callback%test' — 'callback_test'
options		Object		Опции источника данных.
	bounds	Bounds		Область на карте, для которой есть данные или null, если данные есть везде. Опция для стандартной реализации метода restrict. Значение по умолчанию: null
	maxZoom	Bounds		Максимальный уровень масштабирования, для которого делаются запросы за данными. Опция для стандартной реализации метода restrict. Значение по умолчанию: 0
	minZoom	Bounds		Минимальный уровень масштабирования, для которого делаются запросы за данными. Опция для стандартной реализации метода restrict. Значение по умолчанию: 0

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	noCache	Boolean		false — использовать встроенный кэш браузера, true — не использовать (к каждому URL будет автоматически приписываться случайный GET-параметр, чтобы исключить кэширование тайлов браузером). Значение по умолчанию: false

* Обязательный параметр/опция.

Пример:

```
// Пример стандартного формата ответа сервера
hotspot_callback({
    // ответ может содержать поля "data" или "error".
    "data": {
        "type": "FeatureCollection",
        // Массив хотспотных объектов.
        "features": [{
            "type": "Feature",
            // Данные объекта.
            // Передаются в конструктор hotspot.Shape в качестве данных.
            "properties": {
                "hintContent": "Содержимое текстовой подсказки.",
                "balloonContentBody": "Содержимое балуна.",
                "balloonContentHeader": "Заголовок балуна.",
                "balloonContentFooter": "Нижняя часть балуна.",
                // Можно задавать свойство balloonContent вместо Body/Header/
Footer

                // Обязательное поле, описывающее hotspot.Shape.
                "HotspotMetaData": {
                    // Идентификатор хотспотного объекта.
                   // hotspot.Shape.getId по умолчанию возвращает значение этого
поля.
                    "id": 10469893,
                    "zIndex": 10,
                    // Данные, на основе которых создается геометрия
hotspot.Shape.
                    // 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.
                    //       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.
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
                    "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]
                            ]
                        ]
                    }
                }
            },
            // Реальная геометрия объекта.
            // Необязательное поле.
            // Передается в hotspot.Shape и доступна через метод
hotspot.Shape.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 следующего хотспотного объекта...
            }
        }]
    }
});
```

Поля

Имя	Тип	Описание
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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.
requestObjects(layer, tileNumber, zoom, callback)		Строит массив объектов IHotspotShape, соответствующих определенному слою, тайлу и коэффициенту масштабирования карты, и передает его в функцию callback. Унаследован от IHotspotObjectSource.
restrict(layer, tileNumber, zoom)	Boolean	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.
setKeyTemplate(template)		Задает новый шаблон идентификатора тайла.
setTileUrlTemplate(template)		Задает 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 при формировании padding в jsonp-запросе за данными.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
tileNumber	Number[]	✔	Номер тайла (тайловые координаты).
zoom	Integer	✔	Уровень масштаба.

* Обязательный параметр/опция.

getKeyTemplate

```
{String} getKeyTemplate()
```

Возвращает шаблон идентификатора тайла.

getTileUrl

```
{String} getTileUrl(tileNumber, zoom)
```

Возвращает URL данных для конкретного тайла.

Параметры:

Параметр	Тип	*	Описание
tileNumber	Number[]	✔	Номер тайла (тайловые координаты).
zoom	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 + '&x=' + tileNumber[0] +
'&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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
layer	hotspot.Layer	✔	Слой, которому принадлежат объекты.
res	Object	✔	Ответ сервера.
callback	Function	✔	Функция-обработчик.
tileNumber	Number[]	✔	Номер тайла, для которого пришел ответ.
zoom	Number	✔	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	Тип	*	Описание
layer	hotspot.Layer	✔	Слой активных областей.
tileNumber	Number[]	✔	Номер тайла.
zoom	Integer	✔	Коэффициент масштабирования.

* Обязательный параметр/опция.

Пример:

```
// Пример переопределения метода restrict
// Допустим, что данные есть только для Мурманска и Новосибирска.
var myMap = new ymaps.Map('map', {center: [32.5, 68.9] , zoom: 9}),
    geoBounds = [
         [[31.729958, 69.369182], [34.203324, 68.666473]], // Мурманск
         [[82.179084, 55.341085], [83.725642, 54.670738]]  // Новосибирск
    ],
    projection = myMap.options.get('projection');

var myHotspotSource = new ymaps.hotspot.ObjectSource('http://www.myDomain.ru/
tiles/?%c', '%c');
myHotspotSource.restrict = function(layer, tileNumber, zoom) {

   // Вычислим пиксельные границы городов для данного уровня масштабирования.
   var boundsFromPoints = ymaps.util.bounds.fromPoints,
       toGlobalPixels = projection.toGlobalPixels,
       pixelBounds = [
           boundsFromPoints(
               toGlobalPixels(geoBounds[0][0]], zoom),
               toGlobalPixels(geoBounds[0][1]], zoom)
           ),
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
            boundsFromPoints(
                toGlobalPixels(geoBounds[1][0]), zoom),
                toGlobalPixels(geoBounds[1][1]), zoom)
            )
        ],

        // Вычислим пиксельные границы тайла
        leftTop = [tileNumber[0] * 256, tileNumber[1] * 256],
        tileBounds = [leftTop, [leftTop[0] + 256, leftTop[1] + 256]],
        intersects = ymaps.util.bounds.intersects;
    // Если пиксельные границы тайла пересекаются с пиксельными границами
указанных областей,
    // то нужно будет отправить запрос за данными
    if (intersects(pixelBounds[0], tileBounds) || (intersects(pixelBounds[1],
tileBounds)) {
            return false;
    }

    // Для всех остальных тайлов данных у этого источника нет.
    return true;
}
```

setKeyTemplate

```
setKeyTemplate(template)
```

Задает новый шаблон идентификатора тайла.

Параметры:

Параметр	Тип	*	Описание
template	String	✔	Шаблон идентификатора.

* Обязательный параметр/опция.

setTileUrlTemplate

```
setTileUrlTemplate(template)
```

Задает новый шаблон URL данных для тайла.

Параметры:

Параметр	Тип	*	Описание
template	String	✔	Шаблон URL

* Обязательный параметр/опция.

hotspot.Shape

Входит в пакеты package.full, package.standard.

Расширяет **IHotspotShape**.

Фигура хотспотного объекта.

Конструктор | Поля | События | Методы

Конструктор

```
hotspot.Shape(feature, options)
```

Создает фигуру для отображения объекта-хотспота.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
feature		Object	✔	
	geometry	Object|**IGeometry**		Реальная геометрия объекта.
	pixelGeometry	**IPixelGeometry**		Пиксельные координаты фигуры.
	properties	Object|**IDataManager**		Данные объекта.
options		Object		Данные объекта.
	interactivityModel	String		Ключ модели интерактивности фигуры. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**. Значение по умолчанию: 'default#layer'
	zIndex	Number		zIndex фигуры.

* 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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
renderedgeometrychange	Событие, оповещающее о смене хотспотной геометрии. Унаследовано от **IHotspotShape**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getContainer()	IHotspotContainer	Возвращает контейнер хотспотных шейпов. Унаследован от **IHotspotShape**.
getGeometry()	Object	Метод для получения реальной геометрии объекта. Унаследован от **IHotspotShape**.
getId()	Number	Возвращает идентификатор объекта. Унаследован от **IHotspotShape**.
getProperties()	Object	Возвращает данные объекта. Унаследован от **IHotspotShape**.
getRenderedGeometry()	**IHotspotShapeGeometry**	Возвращает пиксельную геометрию хотспотного объекта. Унаследован от **IHotspotShape**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
setContainer(container)		Метод для добавления хотспотного шейпа в контейнер. Унаследован от **IHotspotShape**.
setGeometry(geometry)		Метод для задания реальной геометрии объекта. Унаследован от **IHotspotShape**.
setId(id)		
setProperties(properties)		
setRenderedGeometry(renderedGeometry)		

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

interactivityModel

interactivityModel.storage

Входит в пакеты package.full, package.map, package.standard.

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 получает 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 получает все DOM-события. На карту прокидываются события 'wheel' и 'mousedown'. У события 'mousedown' выставляется флаг 'preventDefault'=true. Если на карте включены поведения 'scrollZoom' или 'magnifier', они будут работать через объекты с данной моделью интерактивности, в отличие от объектов с моделью 'default#opaque'.
- 'default#layer' — объект получает все DOM-события. На карту прокидываются события 'wheel' и 'mousedown'. Если на карте включены поведения 'scrollZoom', 'drag' или 'magnifier', они будут работать через объекты с данной моделью интерактивности.
- 'default#transparent' — объект получает все DOM-события, а затем 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

layer

layer.storage

Входит в пакеты package.full, package.map, package.standard.

Статический объект.

Экземпляр класса **util.Storage**

Хранилище слоев.

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.tile

layer.tile.CanvasTile

Входит в пакеты package.full, package.standard.

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)
```

Создаёт картиночный canvas-тайл. Доступен по ключу "default#canvas" в хранилище классов тайлов.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
`url`		String	✔	URL изображения.
`options`		Object		Опции.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	notFoundTile	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. Значение по умолчанию: null
	tileAnimationDuration	Number		Продолжительность анимации прозрачности изображения. Значение по умолчанию: 350

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий тайла. Унаследовано от ICanvasTile.

События

Имя	Описание
ready	Событие готовности тайла. Унаследовано от ICanvasTile.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает тайл. Унаследован от ICanvasTile.
isReady()	Boolean	Проверяет готовность тайла. Унаследован от ICanvasTile.
renderAt(context, canvasSize, bounds, animate)		Отрисовывает картиночный тайл на 2d-контексте объекта canvas. Унаследован от ICanvasTile.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

layer.tile.DomTile

Входит в пакеты package.full, package.standard.

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)
```

Создаёт картиночный dom-тайл. Доступен по ключу "default#dom" в хранилище классов тайлов. Создаёт картиночный DOM-тайл.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
url		String	✔	URL изображения.
options		Object		Опции.
	notFoundTile	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. Значение по умолчанию: null
	tileAnimationDuration	Number		Продолжительность анимации прозрачности изображения при рисовании в мс (применяется только в браузерах, поддерживающих CSS Transition для свойства opacity). Значение по умолчанию: 300

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий тайла. Унаследовано от IDomTile.

События

Имя	Описание
ready	Событие готовности тайла. Унаследовано от IDomTile.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает тайл. Унаследован от IDomTile.
isReady()	Boolean	Проверяет готовность тайла. Унаследован от IDomTile.
renderAt(context, clientBounds, animate)		Добавляет тайл в родительский HTML-элемент. Унаследован от IDomTile.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

layer.tileContainer

layer.tileContainer.CanvasContainer

Входит в пакеты package.full, package.standard.

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	Свойства параметра	Тип	*	Описание
layer		**ILayer**	✔	Слой.
options		Object		Опции контейнера.
	notFoundTile	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. Значение по умолчанию: null
	tileClass	**ICanvasTile**		Класс тайлов, используемых контейнером. Должен реализовывать интерфейс **ICanvasTile**. Значение по умолчанию: 'default#canvas'
	tileTransparent	Boolean		Флаг, показывающий, являются ли тайлы контейнера прозрачными.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
				Значение по умолчанию: false

* Обязательный параметр/опция.

Поля

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

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.

Методы

Имя	Возвращает	Описание
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
getTile(tileNumber, tileZoom, priority)	ICanvasTile	Фабричная функция создания тайлов.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание методов

getTile

```
{ICanvasTile} getTile(tileNumber, tileZoom, priority)
```

Фабричная функция создания тайлов.

Возвращает экземпляр тайла.

Параметры:

Параметр	Тип	*	Описание
tileNumber	Number[]	✔	Номер тайла.
tileZoom	Number	✔	Масштаб тайла.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
priority	Number	✔	Приоритет загрузки.

* Обязательный параметр/опция.

layer.tileContainer.DomContainer

Входит в пакеты package.full, package.standard.

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	Свойства параметра	Тип	*	Описание
layer		**ILayer**	✔	Слой.
options		Object		Опции контейнера.
	notFoundTile	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. Значение по умолчанию: null
	tileClass	**IDomTile**		Класс тайлов, используемых контейнером. Должен реализовывать интерфейс **IDomTile**. Значение по умолчанию: 'default#dom'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	`tileTransparent`	Boolean		Флаг, показывающий, являются ли тайлы контейнера прозрачными. Значение по умолчанию: false

* Обязательный параметр/опция.

Поля

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

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
ready	Событие готовности всех тайлов.

Методы

Имя	Возвращает	Описание
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
getTile(tileNumber, tileZoom, priority)	IDomTile	Фабричная функция создания тайлов.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание событий

ready

```
ready
```

Событие готовности всех тайлов.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

getTile

```
{IDomTile} getTile(tileNumber, tileZoom, priority)
```

Фабричная функция создания тайлов.

Возвращает экземпляр тайла.

Параметры:

Параметр	Тип	*	Описание
tileNumber	Number[]	✔	Номер тайла.
tileZoom	Number	✔	Масштаб тайла.
priority	Number	✔	Приоритет загрузки.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Layer

Входит в пакеты package.full, package.map, package.standard.

Расширяет **ILayer**, **IPositioningContext**.

Тайловый слой. Позволяет отображать на карте подложку, состоящую из тайлов.

Конструктор | Поля | События | Методы

Конструктор

```
Layer(tileUrlTemplate, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
tileUrlTemplate		String|Function	✔	Строковый шаблон URL тайла, либо функция, которая генерирует URL тайла. Для строкового шаблона поддерживается подстановка: • %c — заменяется на x=номер[0]&y=номер[1]&z=уровень масштабирования; • %x — заменяется на номер[0] • %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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
				несколькими доменами. В качестве n нужно указывать степени двойки (2, 4, 16 и т.д.). Если шаблон имеет вид %d, то n=4. Функция-шаблон получает на вход три параметра: • tileNumber — массив из двух чисел, номера тайла по x и y; • tileZoom — уровень масштабирования; • Возвращает строку-URL.
options		Object		Опции.
	brightness	Number		Яркость слоя. Задается числом от 0 до 1. 0 соответствует черному цвету, 1 — белому. Значение по умолчанию: 0.5
	notFoundTile	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. Значение по умолчанию: null
	pane	IPane		Родительский контейнер слоёв. Значение по умолчанию: 'layers'
	projection	Object		Проекция слоя.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	tileSize	Number[]		Размер тайлов слоя. Значение по умолчанию: [256, 256]
	tileTransparent	Boolean		Флаг, показывающий, являются ли тайлы слоя прозрачными. Значение по умолчанию: false
	zIndex	Number		Z-index слоя в контейнере слоёв. Значение по умолчанию: constants.zIndex.layer

* Обязательный параметр/опция.

Пример:

```
// Добавим 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('&copy; 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: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IChild**.
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. Возвращает яркость слоя. В зависимости от суммарной яркости слоев, добавленных на карту, выбирается цвет логотипа и копирайтов на карте. Унаследован от **ILayer**.
getCopyrights(coords, zoom)	**util.Promise**	Необязательный метод. Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах. Унаследован от **ILayer**.
getPane()	**IPane**	Возвращает контейнер, в котором находится слой.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
getTileSize(zoom)	Number[]	Возвращает размеры тайлов в зависимости от масштаба.
getTileUrl(tileNumber, tileZoom)	String|null	Возвращает URL тайла по его номеру и уровню масштабирования или null если для запрашиваемого участка нет данных.
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**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getZoomRange(point)	**util.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 зацикленность карты по осям x, y).
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.
setTileUrlTemplate(tileUrlTemplate)		
toClientPixels(globalPixelPoint)	Number[]	Преобразует глобальные пиксельные координаты в клиентские. Унаследован от **IPositioningContext**.
update()		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	Тип	*	Описание
`clientPixelPoint`	Number	✔	Точка в клиентских пиксельных координатах.
`tileZoom`	Number	✔	Уровень масштаба тайла.

* Обязательный параметр/опция.

getPane

```
{IPane} getPane()
```

Возвращает контейнер, в котором находится слой.

Возвращает контейнер.

getTileSize

```
{Number[]} getTileSize(zoom)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает размеры тайлов в зависимости от масштаба.

Возвращает размеры тайла по горизонтали и вертикали для указанного коэффициента масштабирования.

Параметры:

Параметр	Тип	*	Описание
zoom	Number	✔	Значение коэффициента мастабирования.

* Обязательный параметр/опция.

Пример:

```
// Будем показывать 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('&copy; OpenStreetMap contributors, CC-BY-SA');
```

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	Тип	*	Описание
tileNumber		✔	
tileZoom		✔	

* Обязательный параметр/опция.

Пример:

```
// Определим свою 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()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
tileNumber	Integer[]	✔	Номер тайла.
tileZoom	Integer	✔	Уровень масштаба 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	Тип	*	Описание
number	Integer[]	✔	Номер тайла.
tileZoom	Integer	✔	Уровень масштабирования тайла.

* Обязательный параметр/опция.

setTileUrlTemplate

```
setTileUrlTemplate(tileUrlTemplate)
```

Параметры:

Параметр	Тип	*	Описание
tileUrlTemplate	String|Function	✔	Строковый шаблон URL тайла, либо функция, которая его генерирует.

* Обязательный параметр/опция.

update

```
update()
```

Удаляет старые тайлы и запрашивает новые.

LayerCollection

Входит в пакеты package.full, package.map, package.standard.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Расширяет **ILayer**, **IMapObjectCollection**.

Коллекция слоев.

Конструктор | Поля | События | Методы

Конструктор

```
LayerCollection()
```

Коллекция слоев.

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: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
remove	Произошло удаление дочернего объекта. Унаследовано от **ICollection**.
zoomrangechange	Событие изменения доступной информации о диапазоне уровней масштабирования. Унаследовано от **ILayer**.

Методы

Имя	Возвращает	Описание
add(child)	**LayerCollection**	Добавляет в коллекцию дочерний объект.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
each(callback, context)		Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.
getBrightness()	Number	Возвращает яркость слоя в виде числа от 0 до 1.
getCopyrights(coords, zoom)	**util.Promise**	Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах. Если точка и коэффициент масштабирования не указаны, будут использованы центр и коэффициент масштабирования карты.
getIterator()	**IIterator**	Возвращает итератор по коллекции. Унаследован от **ICollection**.
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
getZoomRange(coords)	**util.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.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

Описание методов

add

```
{LayerCollection} add(child)
```

Добавляет в коллекцию дочерний объект.

Возвращает ссылку на себя.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
child	ILayer|String	✔	Слой (ключ хранилища layer.storage или экземпляр класса ILayer), который нужно добавить.

* Обязательный параметр/опция.

Пример:

```
// Допустим, мы хотим добавить в свою коллекцию несколько слоёв.
var layerCollection = new ymaps.LayerCollection(),
    customLayer = new ymaps.Layer(
         'http://tile.openstreetmap.org/%z/%x/%y.png', {
         projection: ymaps.projection.sphericalMercator
    }),
    // Слой можно задать ключём из layer.storage.
    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	Тип	*	Описание
callback	Function	✔	Функция-обработчик.
context	Object		Контекст выполнения функции.

* Обязательный параметр/опция.

getBrightness

```
{Number} getBrightness()
```

Возвращает яркость слоя в виде числа от 0 до 1.

getCopyrights

```
{util.Promise} getCopyrights(coords, zoom)
```

Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах. Если точка и коэффициент масштабирования не указаны, будут использованы центр и коэффициент масштабирования карты.

Возвращает объект-promise.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	Описание
coords	Number[]	Точка на карте, для которой запрашивается информация об авторских правах.
zoom	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 promise = new ymaps.util.Promise();
    $.ajax('url/to/copyrights/provider?ll=' +
        (coords || map.getCenter()).join(',') + '&z=' +
        (zoom || map.getZoom()),
        function (res) {
        promise.resolve(res || []);
    });
    return promise;
};
```

getZoomRange

```
{util.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	Тип	*	Описание
coords		✔	

* Обязательный параметр/опция.

Пример:

```
// Допустим, наш слой отрисован для масштабов 2-15 по всей земле
myLayer.getZoomRange = function () {
    var promise = new ymaps.util.Promise();
    promise.resolve([2, 15]);
    return promise;
}
```

remove

```
{LayerCollection} remove(child)
```

Удаляет из коллекции дочерний объект.

Возвращает ссылку на себя.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
child	ILayer|String	✔	Слой (строка-ключ из layer.storage или экземпляр класса ILayer), который нужно удалить.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

layout

layout.Image

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geocode, package.search, package.standard.

Расширяет **ILayout**.

Класс для создания макетов, содержащих картинку.

Конструктор | Поля | События | Методы

Конструктор

```
layout.Image()
```

Создает экземпляр макета-картинки.

Параметры:

Параметр	Свойства параметра	Тип	Описание
	options.imageHref	String	URL графического файла.
	options.imageClipRect	Number[][]	Координаты отображаемой области исходного изображения в пикселях.
	options.imageOffset	Number[]	Смещение слоя с картинкой относительно точки привязки.
	options.imageSize	Number[]	Размеры слоя с картинкой.

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от **ILayout**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **ILayout**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от **ILayout**.
getData()	Object	Возвращает объект данных макета. Унаследован от **ILayout**.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от **ILayout**.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от **ILayout**.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от **ILayout**.

layout.ImageWithContent

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geocode, package.search, package.standard.

Расширяет **layout.Image**.

Класс для создания макетов, состоящих из картинки и содержимого.

Конструктор | Поля | События | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
layout.ImageWithContent()
```

Создает экземпляр макета-картинки с содержимым.

Параметры:

Параметр	Свойства параметра	Тип	Описание
	`options.contentLayout`	Function|String	Макет содержимого. (Тип: конструктор объекта с интерфейсом ILayout или его ключ в хранилище).
	`options.contentOffset`	Number[]	Смещение слоя с содержимым относительно слоя с картинкой.
	`options.contentSize`	Number[]	Размеры слоя с содержимым.

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от **ILayout**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **ILayout**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от **ILayout**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от **ILayout**.
getData()	Object	Возвращает объект данных макета. Унаследован от **ILayout**.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от **ILayout**.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от **ILayout**.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от **ILayout**.

layout.storage

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geoXml, package.geocode, package.route, package.search, package.standard.

Статический объект.

Экземпляр класса **util.Storage**

Хранилище классов лейаутов.

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

layout.templateBased

layout.templateBased.Base

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geoXml, package.geocode, package.route, package.search, package.standard.

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)
```

См.: templateLayoutFactory

Параметры:

Параметр	Тип	*	Описание
data	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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от **ILayout**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от **ILayout**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
build()		Строит экземпляр макета на основе шаблона и добавляет его в родительский HTML-элемент.
clear()		Удаляет содержимое макета из DOM.
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от ILayout.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от ILayout.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от ILayout.
getData()	Object	Возвращает объект данных макета. Унаследован от ILayout.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от ILayout.
isEmpty()	Boolean	Возвращает признак того, что макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие. Унаследован от ILayout.
rebuild()		Перестраивает макет.
setData(data)		Задает данные макета. Унаследован от ILayout.
setParentElement(parent)		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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

clear

```
clear()
```

Удаляет содержимое макета из DOM.

rebuild

```
rebuild()
```

Перестраивает макет.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

load

Статическая функция.

Загружает новые пакеты API.

Параметры:

Параметр	Тип	*	Описание
packages	String[]|String	✔	Имена пакетов, которые требуется загрузить.
callback	Function		Функция, которая будет вызвана при успешной загрузке пакетов.
context	Object		Контекст выполнения.

* Обязательный параметр/опция.

Пример:

```
<script src="http://api-maps.yandex.ru/2.0/?load=package.map"type="text/
javascript"></script>
<script type="text/javascript">
    if (window.location.pathname == '/special-page') {
        // Для этой страницы нужны пробки и панель поиска
        ymaps.load(['package.traffic', 'package.search']);
    }
</script>
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

map

map.action

map.action.Continuous

Входит в пакеты package.full, package.map, package.standard.

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**.
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. 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**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
isActive()	Boolean	Проверяет, исполняется ли движение картой в данный момент.
tick(tick)	map.action.Continuous	Выполняет один шаг движения карты.

Описание методов

isActive

```
{Boolean} isActive()
```

Проверяет, исполняется ли движение картой в данный момент.

Возвращает true — движение в данный момент исполняется, false — нет.

tick

```
{map.action.Continuous} tick(tick)
```

Выполняет один шаг движения карты.

Возвращает ссылку на себя.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
tick		Object	✔	Параметры движения.
	duration	Number		Длительность исполнения движения, в миллисекундах. Значение по умолчанию: 0
	globalPixelCenter	Number[]		Новый центр карты в глобальных пикселах. Должен быть задан один из параметров: либо pixelOffset, либо globalPixelCenter
	pixelOffset	Number[]		Смещение в пикселях относительно предыдущего центра. Должен быть задан один из параметров: либо pixelOffset, либо globalPixelCenter.
	timingFunction	String		Тайминг-функция. Значение по умолчанию: 'linear'
	zoom	Number		Новый коэффициент масштабирования карты. Если не указан, масштаб карты не изменяется.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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)
```

См.: Map.action

Параметры:

Параметр	Тип	*	Описание
map	**Map**	✔	Карта.

* Обязательный параметр/опция.

Пример:

```
// Создадим сложное движение: каждые 100 мс центр
// карты смещается на случайную величину.

// Создадим экземпляр сложного движения.
var action = new ymaps.map.action.Continuous();
// Запустим его выполнение на карте.
myMap.action.execute(action);

// Запомним пиксельный центр карты и уровень масштабирования.
var center = myMap.getGlobalPixelCenter(),
    zoom = myMap.getZoom();
// Каждые 100 миллисекунд будем генерировать случайный сдвиг.
var interval = window.setInterval(function () {
    center[0] += Math.round(Math.random() * 100) - 50;
    center[1] += Math.round(Math.random() * 100) - 50;
    // Генерируем новый сдвиг карты.
    action.tick({
        globalPixelCenter: center,
        zoom: zoom
    });
}, 100);

// Как только пользователь сдвинет карту, наше движение
// прекратит выполняться и произойдёт событие end.
var listener = action.events.once('end', function () {
    window.clearInterval(interval);
});
```

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
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.
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.

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 окончанию коррекции необходимо вернуть скорректированные значения.
stop()		Останавливает выполнение действия на карте.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание событий

begin

```
begin
```

Событие, возникающее, когда действие началось. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- action — запущенное действие.

break

```
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

```
end
```

Событие, возникающее при остановке действия. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- action — остановленное действие.

tick

```
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

```
tickcomplete
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
action	**IMapAction**	✔	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:

```
// Логирует 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 окончанию коррекции необходимо вернуть скорректированные значения.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
userFunction	Function	✔	Пользовательская функция коррекции шагов.

* Обязательный параметр/опция.

Пример:

```
// Сделаем так, чтобы пользователь не мог перетащить центр карты
// за пределы МКАД.
var mkad = [
    [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

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IMapAction**.

Простое движение карты. Движение выполняется сразу после передачи его в map.action.Manager.

Конструктор | Поля | События | Методы

Конструктор

```
map.action.Single(tick)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Создаёт простое (один шаг) движение карты.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
`tick`		Object	✔	Параметры движения.
	`callback`	Function		Функция, которая будет вызвана после выполнения действия. Принимает в качестве параметра ошибку или null, если действие выполнилось успешно.
	`center`	Number[]		Новый центр карты в геокоординатах.
	`checkZoomRange`	Boolean		Флаг, показывающий, нужно ли проверять новый коэффициент мастабирования 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. Значение по умолчанию: false
	`duration`	Number		Время исполнения движения, в миллисекундах. Значение по умолчанию: 0
	`globalPixelCenter`	Number[]		Новый центр карты в глобальных пикселах. При одновременном задании параметров center и globalPixelCenter, параметр center игнорируется.
	`timingFunction`	String		Тайминг-функция. Значение по умолчанию: 'linear'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	zoom	Number		Новый коэффициент масштабирования карты.

* 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: • 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
end()		Останавливает движение. Унаследован от IMapAction.
isActive()	Boolean	Проверяет, исполняется ли движение картой в данный момент.

Описание методов

isActive

```
{Boolean} isActive()
```

Проверяет, исполняется ли движение картой в данный момент.

Возвращает true — движение в данный момент исполняется, false — нет.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

map.Balloon

Расширяет **IEventEmitter**.

Менеджер балуна карты. У каждой карты уже есть собственный менеджер балуна, доступный как map.balloon. В один момент на карте может быть открыт только один балун, управляемый менеджером. Не создавайте экземпляры этого класса без необходимости.

См.: Map.balloon

Конструктор | Поля | События | Методы

Конструктор

```
map.Balloon(map)
```

См.: Map.balloon

Параметры:

Параметр	Тип	*	Описание
map	**Map**	✔	Ссылка на объект карты.

* Обязательный параметр/опция.

Поля

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

События

Имя	Описание
close	Произошло закрытие балуна. Экземпляр класса **Event**.
open	Произошло открытие балуна. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • balloon — ссылка на экземпляр класса **Balloon**.

Методы

Имя	Возвращает	Описание
close(callback)		Закрывает балун.
isOpen()	Boolean	Проверяет состояние балуна на карте.
open(position, data, options)	**Balloon**	Открывает балун на карте в заданной точке. Возвращает экземпляр класса **Balloon**.

Описание событий

close

```
close
```

Произошло закрытие балуна. Экземпляр класса **Event**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

open

```
open
```

Произошло открытие балуна. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- balloon — ссылка на экземпляр класса **Balloon**.

Описание методов

close

```
close(callback)
```

Закрывает балун.

Параметры:

Параметр	Тип	*	Описание
callback		✔	

* Обязательный параметр/опция.

isOpen

```
{Boolean} isOpen()
```

Проверяет состояние балуна на карте.

Возвращает true, если балун на карте открыт, false — в обратном случае.

Пример:

```
// Открываем балун при двойном щелчке на карте
// или просто обновляем содержимое (случайное число),
// если балун уже был открыт.
var balloon = null;

myMap.events.add('dblclick', function () {
    if (myMap.balloon.isOpen()) {
        balloon.setData({ content: Math.random() });
    } else {
        balloon = myMap.balloon.open(myMap.getCenter(), Math.random());
    }
});
```

open

```
{Balloon} open(position, data, options)
```

Открывает балун на карте в заданной точке. Возвращает экземпляр класса **Balloon**.

Возвращает балун.

Параметры:

Параметр	Тип	*	Описание
position	Number[]	✔	Позиция балуна. Указывается в гео-координатах текущей проекции.
data	Object|String		Данные для отображения.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
options	Object		Опции.

* Обязательный параметр/опция.

Пример:

```
// Открываем балун на карте, через 1 секунду меняем его содержимое, еще через 5
секунд закрываем.
var balloon = myMap.balloon.open(myMap.getCenter(), { content: 'Hello Yandex!' },
{ closeButton: true });

setTimeout(function () {
    balloon.setData({ content: 'Этот балун закроется через 3 секунды.' })
    setTimeout(function () { myMap.balloon.close(); }, 3000);
}, 1000)
```

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)
```

См.: Map.behaviors

Параметры:

Параметр	Тип	*	Описание
map	**Map**	✔	Карта.
behaviors	String|String[]		Список поведений карты, включенных сразу при создании карты. По умолчанию — "drag", "dblClickZoom", "rightMouseButtonMagnifier" для настольных браузеров, "drag", "dblClickZoom" и "multiTouch" — для мобильных.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
			Возможные значения ключей: • "default" — короткий синоним для включения/отключения поведений карты по умолчанию; • "drag" — перемещание 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**; Добавлять и удалять классы поведений можно через хранилище поведений **behavior.storage**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
options	Object		Опции поведений. Можно задавать следующие опции: • 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	Менеджер событий. Унаследовано от IEventEmitter.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
behaviors	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	Тип	*	Описание
behaviors	String|String[]	✔	Список подключаемых поведений.

* Обязательный параметр/опция.

Пример:

```
myMap.behaviors.enable(['ruler', 'multiTouch']);
```

get

```
{IBehavior} get(behaviorName)
```

Возвращает экземпляр поведения по ключу.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
behaviorName	String	✔	Имя поведения.

* Обязательный параметр/опция.

Пример:

```
myMap.behaviors.get('drag');
```

isEnabled

```
{Boolean} isEnabled(behaviorName)
```

Проверяет, включено ли в настоящий момент поведение.

Возвращает true, если поведение включено, в противном случае — false.

Параметры:

Параметр	Тип	*	Описание
behaviorName	String	✔	Идентификатор поведения.

* Обязательный параметр/опция.

Пример:

```
// Если поведение "drag" отключено, то подключаем его
if (!(myMap.behaviors.isEnabled('drag'))) {
    myMap.behaviors.enable('drag');
}
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

map.Container

Расширяет **IEventEmitter**.

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)
```

См.: Map.container

Параметры:

Параметр	Тип	*	Описание
parentElement	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 **IEventEmitter**.

События

Имя	Описание
sizechange	Событие, возникающее при изменении размера контейнера карты.

Методы

Имя	Возвращает	Описание
fitToViewport(preservePixelPosition)		Вызывается при изменении размера контейнера карты для того, чтобы карта применила новый размер.
getElement()	HTMLElement	Возвращает HTML-элемент карты.
getOffset()	Number[]	Возвращает смещение контейнера карты в пикселях относительно левого верхнего угла документа.
getSize()	Number[]	Возвращает размеры контейнера карты в пикселях.

Описание событий

sizechange

```
sizechange
```

Событие, возникающее при изменении размера контейнера карты.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

fitToViewport

```
fitToViewport(preservePixelPosition)
```

Вызывается при изменении размера контейнера карты для того, чтобы карта применила новый размер.

Параметры:

Параметр	Тип	Описание
preservePixelPosition	Boolean	Сохранять положение центра карты.

Пример:

```
// Изменим размеры контейнера карты
map.container.getElement().style.width = '300px';
// Инициируем 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.

getSize

```
{Number[]} getSize()
```

Возвращает размеры контейнера карты в пикселях.

map.control

map.control.Manager

Расширяет **IMapObjectCollection**.

Менеджер элементов управления карты. У каждой карты уже есть собственный менеджер контролов, доступный как map.controls. Не инстанцируйте экземпляры этого класса без необходимости.

См.: Map.controls

Конструктор | Поля | События | Методы

Конструктор

```
map.control.Manager(map, controls)
```

См.: Map.controls

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
map	Map	✔	Ссылка на объект карты.
controls	String[]|IControl[]	✔	Элементы управления, заданные экземплярами классов, реализующих интерфейс **IControl** либо ключами. Возможные значения ключей: • "mapTools" — стандартный тулбар с кнопками "Навигация", "Выделение" и "Линейка" **control.MapTools**; • "miniMap" — обзорная карта **control.MiniMap**; • "scaleLine" — масштабный отрезок **control.ScaleLine**; • "searchControl" — панель поиска **control.SearchControl**; • "trafficControl" — панель пробок **control.TrafficControl**; • "typeSelector" — панель переключения типа карты **control.TypeSelector**; • "zoomControl" — ползунок масштаба **control.ZoomControl**; • "smallZoomControl" — простой ползунок масштаба **control.SmallZoomControl**.

* Обязательный параметр/опция.

Пример:

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

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
		Унаследовано от **ICustomizable**.

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от **ICollection**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
remove	Произошло удаление дочернего объекта. Унаследовано от **ICollection**.

Методы

Имя	Возвращает	Описание
add(control, position)	**map.control.Manager**	Добавляет элемент управления на карту.
each(callback, context)		Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.
getIterator()	**IIterator**	Возвращает итератор по коллекции. Унаследован от **ICollection**.
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
remove(control)	**map.control.Manager**	Удаляет элемент управления с карты.

Описание методов

add

```
{map.control.Manager} add(control, position)
```

Добавляет элемент управления на карту.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
control	**IControl**|String	✔	Элемент управления или его ключ.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
position	Object		Позиция элемента управления на карте. Задаётся в виде объекта с полями left/right и top/bottom (например, так: { left: 5, top: 5 }). При указании этого параметра экземпляру элемента управления будет задана опция position с указанным значением.

* Обязательный параметр/опция.

each

```
each(callback, context)
```

Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Параметры:

Параметр	Тип	*	Описание
callback	Function	✔	Функция-обработчик.
context	Object		Контекст выполнения функции.

* Обязательный параметр/опция.

remove

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

Удаляет элемент управления с карты.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
control	IControl|String	✔	Элемент управления или его ключ.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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 инстанцируйте экземпляры этого класса без необходимости.

См.: **Map.converter**

Конструктор | Методы

Конструктор

```
map.Converter(map)
```

См.: **Map.converter**

Параметры:

Параметр	Тип	*	Описание
map	**Map**	✔	Ссылка на карту.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
globalToPage(globalPixelPoint)	Number[]	Преобразовывает глобальные пиксельные координаты точки в локальные.
pageToGlobal(pagePixelPoint)	Number[]	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	Тип	*	Описание
globalPixelPoint	Number[]	✔	Пиксельные координаты точки, которые необходимо преобразовать.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// 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	Тип	*	Описание
pagePixelPoint	Number[]	✔	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

Расширяет **IEventEmitter**.

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 | Поля | События | Методы

Конструктор

```
map.Copyrights(map)
```

См.: Map.copyrights

Параметры:

Параметр	Тип	*	Описание
map	**Map**	✔	Карта.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Пример:

```
// Добавим статическую информацию об авторских правах на карту
var accessor = map.copyrights.add('&copy; Герард Меркатор');
// ...
// Удалим информацию об авторских правах
accessor.remove();
```

Поля

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

События

Имя	Описание
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;

Методы

Имя	Возвращает	Описание
add(customCopyrights)	ICopyrightsAccessor	Добавляет на карту статическую (не зависящую от текущего центра и масштаба карты) информацию об авторских правах.
addProvider(provider)	map.Copyrights	Добавляет нового провайдера информации об авторских правах.
get(point, zoom)	util.Promise	Определяет информацию об авторских правах в указанной точке. Возвращает объект-promise, который будет подтверждён и передаст в качестве параметра массив строк/DOM-элементов.
removeProvider(provider)	map.Copyrights	Удаляет провайдера информации об авторских правах.

Описание событий

change

```
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;

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
customCopyrights	String|HTMLElement|String[]|HTMLElement[]	✔	Информация об авторских правах в виде строки, DOM-элемента либо массива строк/DOM-элементов.

* Обязательный параметр/опция.

addProvider

```
{map.Copyrights} addProvider(provider)
```

Добавляет нового провайдера информации об авторских правах.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
provider	ICopyrightsProvider	✔	Провайдер.

* Обязательный параметр/опция.

Пример:

```
// Создадим динамический провайдер информации об авторских правах,
// который будет получать актуальную информацию об авторских правах для
// конкретной точки карты с удаленного сервера.
var myProvider = {
        getCopyrights: function (center, zoom) {
            var promise = new ymaps.util.Promise();
            $.ajax('http://some.server/copyrights/?ll=' +
                center.join(',') = '&z=' + zoom, {
                // Сервер должен вернуть массив строк
                success: function (res) {
                    promise.resolve(res);
                }
            });
            return promise;
        }
    };
// Добавим провайдер на карту; теперь при изменении центра или уровня
// масштабирования менеджер информации об авторских правах карты будет
// перезапрашивать акутальную информацию с удаленного сервера и отображать
// полученные данные автоматически.
map.copyrights.addProvider(myProvider);
```

get

```
{util.Promise} get(point, zoom)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Определяет информацию об авторских правах в указанной точке. Возвращает объект-promise, который будет подтверждён и передаст в качестве параметра массив строк/DOM-элементов.

Возвращает объект-promise.

Параметры:

Параметр	Тип	Описание
point	Number[]	Точка (в географических координатах), для которой требуется определить информацию об авторских правах. Если не указана, берётся текущий центр карты.
zoom	Number	Уровень масштабирования, для которого требуется определить информацию об авторских правах. Если не указан, берётся текущий уровень масштабирования карты.

removeProvider

```
{map.Copyrights} removeProvider(provider)
```

Удаляет провайдера информации об авторских правах.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
provider	ICopyrightsProvider	✔	Провайдер.

* Обязательный параметр/опция.

map.GeoObjects

Расширяет **IMapObjectCollection**.

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)
```

См.: Map.geoObjects

Параметры:

Параметр	Тип	*	Описание
map	Map	✔	Карта

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
options	Object	✔	Опции геообъектов. Через опции map.geoObjects можно задавать настройки геообъектов, добавленных на карту: • опции кластеризаторов с префиксом clusterer; • опции кластеров с префиксом cluster;

* Обязательный параметр/опция.

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.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection.

Методы

Имя	Возвращает	Описание
add(object)	ICollection	Добавляет в коллекцию дочерний объект. Унаследован от ICollection.
each(callback, context)		Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.
getBounds()	Number[][]|null	Возвращает географические координаты области, охватывающей все геообъекты.
getIterator()	IIterator	Возвращает итератор по коллекции. Унаследован от ICollection.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getPixelBounds()	Number[][]|null	Возвращает глобальные пиксельные координаты области, охватывающей все геообъекты.
remove(object)	ICollection	Удаляет из коллекции дочерний объект. Унаследован от ICollection.

Описание методов

each

```
each(callback, context)
```

Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Параметры:

Параметр	Тип	*	Описание
callback	Function	✔	Функция-обработчик.
context	Object		Контекст выполнения функции.

* Обязательный параметр/опция.

Пример:

```
// Найдём на карте геообъект по его id
map.geoObjects.each(function (geoObject) {
    if (geoObject.properties.get('id') == 'some id') {
        // do something
        ...
        return false;
    }
});
```

getBounds

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

Возвращает географические координаты области, охватывающей все геообъекты.

Пример:

```
// Устанавливаем центр и масштаб карты так, чтобы отобразить все геообъекты
целиком.
myMap.setBounds(myMap.geoObjects.getBounds());
```

getPixelBounds

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

Возвращает глобальные пиксельные координаты области, охватывающей все геообъекты.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

map.Hint

Расширяет **IEventEmitter**.

Менеджер хинта карты. У каждой карты уже есть собственный менеджер хинта, доступный как map.hint. В один момент на карте может быть открыт только один хинт, управляемый менеджером. Не создавайте экземпляры этого класса без необходимости.

См.: Map.hint

Конструктор | Поля | События | Методы

Конструктор

```
map.Hint(map)
```

См.: Map.hint

Параметры:

Параметр	Тип	*	Описание
map	**Map**	✔	Ссылка на объект карты.

* Обязательный параметр/опция.

Поля

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

События

Имя	Описание
hide	Событие, возникающее после скрытия hint-а. Экземпляр класса **Event**.
show	Событие, возникающее после отображения hint-а на карте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • hint — ссылка на экземпляр класса **Hint**.

Методы

Имя	Возвращает	Описание
hide(callback, urgently)		Скрывает хинт.
isShown()	Boolean	Проверяет состояние хинта.
show(position, data, options)	**Hint**	Показывает хинт на карте в заданной точке. Возвращает экземпляр класса **Hint**

Описание событий

hide

```
hide
```

Событие, возникающее после скрытия hint-а. Экземпляр класса **Event**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

show

```
show
```

Событие, возникающее после отображения hint-а на карте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- hint — ссылка на экземпляр класса **Hint**.

Описание методов

hide

```
hide(callback, urgently)
```

Скрывает хинт.

Параметры:

Параметр	Тип	*	Описание
callback		✔	
urgently		✔	

* Обязательный параметр/опция.

isShown

```
{Boolean} isShown()
```

Проверяет состояние хинта.

Возвращает true — если хинт в данный момент открыт, false — в обратном случае.

Пример:

```
// Открываем хинт (по таймауту в 1 сек.)
var hint = myMap.hint.show(myMap.getCenter(), { content: 'loading' },
{ showTimeout: 1000 });

// Выполняем AJAX-запрос:
$.getJSON('hint-data.json')
    .success(function (data) {
        // Обновляем данные если хинт открыт, иначе вешаемся на событие
открытия.
        if (myMap.hint.isShown()) {
            hint.setData(data);
        } else {
            myMap.hint.events.add('show', function () {
                hint.setData(data);
            });
        }
    });
```

show

```
{Hint} show(position, data, options)
```

Показывает хинт на карте в заданной точке. Возвращает экземпляр класса **Hint**

Возвращает экземпляр хинта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
position	Number[]	✔	Позиция привязки хинта.
data	Object|String		Данные для отображения.
options	Object		Опции хинта.

* Обязательный параметр/опция.

Пример:

```
// Показываем в хинте координаты клика на карте.
geoMap.events.add('click', function (e) {
    geoMap.hint.show(e.get('coordPosition'), e.get('coordPosition').join(', '),
{
        showTimeout: false,
        hideTimeout: 500
    });
});
```

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.

См.: Map.layers

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
map		Map	✔	Карта
options		Object	✔	Опции слоев карты. Через опции map.layers можно задавать настройки слоев, добавленных на карту. Опции слоев хотспотов задаются с префиксом 'hotspotLayer'.
	trafficImageZIndex	Number		Z-index картиночного слоя пробок. Значение по умолчанию: 201

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	`trafficInfoZIndex`	Number		Приоритет хотспотного слоя инфоточек. Значение по умолчанию: 1
	`trafficJamZIndex`	Number		Приоритет хотспотного слоя пробок. Значение по умолчанию: 0

* Обязательный параметр/опция.

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: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
remove	Произошло удаление дочернего объекта. Унаследовано от **ICollection**.
zoomrangechange	Событие изменения доступной информации о диапазоне уровней масштабирования. Унаследовано от **ILayer**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
add(object)	ICollection	Добавляет в коллекцию дочерний объект. Унаследован от ICollection.
each(callback, context)		Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.
getBrightness()	Number	Необязательный метод. Возвращает яркость слоя. В зависимости от суммарной яркости слоев, добавленных на карту, выбирается цвет логотипа и копирайтов на карте. Унаследован от ILayer.
getCopyrights(coords, zoom)	util.Promise	Необязательный метод. Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах. Унаследован от ILayer.
getIterator()	IIterator	Возвращает итератор по коллекции. Унаследован от ICollection.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
getZoomRange(point)	util.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.
remove(object)	ICollection	Удаляет из коллекции дочерний объект. Унаследован от ICollection.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание методов

each

```
each(callback, context)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Перебирает все элементы коллекции, вызывая для каждого функцию-обработчик.

Параметры:

Параметр	Тип	*	Описание
`callback`	Function	✔	Функция-обработчик.
`context`	Object		Контекст выполнения функции.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

map.pane

map.pane.Manager

Менеджер контейнеров карты. У каждой карты уже есть свой менеджер контейнеров, доступный как map.panes. Не инстанцируйте экземпляры этого класса без необходимости.

См.: Map.panes

Конструктор | Методы

Конструктор

```
map.pane.Manager(map)
```

См.: Map.panes

Параметры:

Параметр	Тип	*	Описание
map	Map	✔	Карта.

* Обязательный параметр/опция.

Пример:

```
// Добавим watermark над контейнером карты.
// Для этого изменим background у контейнера событий.
map.panes.get('events').getElement().style.backgroundImage = 'my/background/
image';
```

Методы

Имя	Возвращает	Описание
get(key)	IPane	Возвращает контейнер карты с соответствующим ключом. Список ключей и контейнеров карты: • 'controls': IPane — контейнер элементов управлениия карты; • 'events': IPane — контейнер, на котором слушаются события карты; • 'floats': IPane — контейнер элементов (например, хинтов), размещенных на карте; • 'graphics': IPane — контейнер графики; • 'layers': IPane — контейнер слоев карты; • 'outers': IPane — контейнер элементов (например, хинтов), выступающих за границы карты; • 'overlays': IPane — контейнер геообъектов карты; • 'shadows': IPane — контейнер для размещения макетов теней объектов карты .

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

get

```
{IPane} get(key)
```

Возвращает контейнер карты с соответствующим ключом. Список ключей и контейнеров карты:

- 'controls': **IPane** — контейнер элементов управлениия карты;
- 'events': **IPane** — контейнер, на котором слушаются события карты;
- 'floats': **IPane** — контейнер элементов (например, хинтов), размещенных на карте;
- 'graphics': **IPane** — контейнер графики;
- 'layers': **IPane** — контейнер слоев карты;
- 'outers': **IPane** — контейнер элементов (например, хинтов), выступающих за границы карты;
- 'overlays': **IPane** — контейнер геообъектов карты;
- 'shadows': **IPane** — контейнер для размещения макетов теней объектов карты

.

Параметры:

Параметр	Тип	*	Описание
key	String	✔	Ключ.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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)
```

См.: Map.zoomRange

Параметры:

Параметр	Тип	*	Описание
map	**Map**	✔	Карта.
constraints	Number[]	✔	Массив, содержащий минимальный и максимальный зум карты.

* Обязательный параметр/опция.

Поля

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

События

Имя	Описание
change	Произошло изменение диапазона масштабов карты.

Методы

Имя	Возвращает	Описание
get(coords)	**util.Promise**	Определяет доступный в указанной точке диапазон уровней масштабирования. Возвращает объект-promise, который будет подтверждён и получит в качестве параметра массив из двух чисел — максимальный и минимальный уровень масштабирования в данной точке.
getCurrent()	Number[]	Возвращает текущее (последнее полученное) значение минимального и максимального уровня масштабирования карты.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание событий

change

```
change
```

Произошло изменение диапазона масштабов карты.

Описание методов

get

```
{util.Promise} get(coords)
```

Определяет доступный в указанной точке диапазон уровней масштабирования. Возвращает объект-promise, который будет подтверждён и получит в качестве параметра массив из двух чисел — максимальный и минимальный уровень масштабирования в данной точке.

Возвращает объект-promise.

Параметры:

Параметр	Тип	*	Описание
coords		✔	

* Обязательный параметр/опция.

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Map

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IDomEventEmitter**.

Класс для создания и управления картой.

Конструктор | Поля | События | Методы

Конструктор

```
Map(element, state, options)
```

Создает экземпляр карты.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
element		Object|String	✔	Ссылка на HTML-элемент, в котором размещается карта, или идентификатор этого HTML-элемента.
state		Object	✔	Параметры карты.
	behaviors	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 двойным щелчком и выделением области правой кнопкой мыши на остальных устройствах. Можно указывать любые ключи, которые поддерживаются **map.behavior.Manager**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	bounds	Number[][]		Геокоординаты начальной области показа карты. При инициализации карты может быть задана либо пара state.zoom-state.center, либо данная область. Если совместно со state.bounds будут заданы параметры state.zoom, state.center, они будут проигнорированы.
	center	Number[]		Геокоординаты центра карты. Должны быть заданы совместно со state.zoom.
	type	String|MapType		Тип карты. Может быть ключом или экземпляром класса MapType. Список доступных ключей: • 'yandex#map' — тип карты "схема"; • 'yandex#satellite' — тип карты "спутник"; • 'yandex#hybrid' — тип карты "гибрид"; • 'yandex#publicMap' — тип карты "народная карта"; • 'yandex#publicMapHybrid' — тип карты "народный гибрид". Ключи 'yandex#publicMap' и 'yandex#publicMapHybrid' доступны только для России и Украины. Значение по умолчанию: 'yandex#map'
	zoom	Number		Коэффициент масштабирования карты. Должен быть задан совместно со state.center.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
`options`		Object		Опции карты. Через опции карты можно задавать настройки как самой карты, так и добавленных на неё объектов: • опции поведений карты; • опции балуна карты с префиксом `balloon`; • опции хинта карты с префиксом `hint`; • опции гео-объектов с префиксом `geoObject`; • опции слоев с префиксом `layer`; • опции слоев хотспотов с префиксом `hotspotLayer`; Опции, которые трактует непосредственно сама карта, перечислены ниже.
	`adjustZoomOnTypeChange`	Number		Выбранного типа карты с текущим кооэффициентом масштабирования тайлов может не оказаться. При значении true кооэффициент масштабирования будет изменен на ближайший допустимый. Значение по умолчанию: false

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	`autoFitToViewport`	String		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 map.container.fitToViewport. Доступны следующие значения: • 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 map.container.fitToViewport, чтобы подстроить карту под него. После слежение будет остановлено; • always — постоянно следить за изменением состояния отображения контейнера карты. Значение по умолчанию: 'ifNull'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	avoidFractionalZoom	Boolean		true — карта не будет останавливаться на дробных значениях коэффициента масштабирования, false — будет. Значение по умолчанию: true
	maxAnimationZoomDifference	Number		Максимальная разница между текущим и выставляемым коэффициентом масштабирования карты, при котором масштабирование будет происходить плавно. Значение по умолчанию: 5
	maxZoom	Number		Максимальный коэффициент масштабирования карты. Значение по умолчанию: 23
	minZoom	Number		Минимальный коэффициент масштабирования карты. Значение по умолчанию: 0
	projection	IProjection		Проекция карты. Значение по умолчанию: ymaps.projection.wgs84Mercator
	restrictMapArea	Boolean|Number[][]		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. Значение по умолчанию: false

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	yandexMapAutoSwitch	Boolean		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. Значение по умолчанию: true

* Обязательный параметр/опция.

Примеры:

1.

```
// Инициализация карты из известных центра и уровня масштабирования
var myMap = new ymaps.Map('map', {
          center: [55.74954, 37.621587],
          zoom: 10,
          // включаем масштабирование карты колесом
          behaviors: ['default', 'scrollZoom']
    });
```

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 : 10
    });
});
```

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	**map.control.Manager**	Элементы управления картой.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
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.CursorManager	Менеджер курсоров карты.
events	event.PriorityManager	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 слоёв 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: • 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.
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	Закрыт балун
balloonopen	Открыт балун на карте. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • balloon — экземпляр балуна.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
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; • newCenter — новый центр карты в геокоординатах; • oldZoom — предыдущий уровень машсштабирования; • 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.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
destroy	Карта уничтожена.
hinthide	Скрыт хинт.
hintshow	Показан хинт на карте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • hint — экземпляр хинта.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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**. Имена полей, доступных через метод Event.get: • oldType — предыдущий тип карты; • newType — новый тип карты.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает карту.
getBounds()	Number[][]	Возвращает двумерный массив геокоординат левого нижнего и правого верхнего углов области показа карты.
getCenter()	Number[]	Возвращает географические координаты текущего центра карты.
getGlobalPixelCenter()	Number[]	Возвращает глобальные пиксельные координаты текущего центра карты.
getType()	String|**MapType**	Возвращает текущий тип карты.
getZoom()	Number	Возвращает текущий коэффициент масштабирования карты.
panTo(center, options)	**Map**	Устанавливает центр карты. Если передан массив точек, то карта будет перемещаться от одной точки к другой.
setBounds(bounds, options)	**Map**	Позиционирует карту для показа переданного района.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
setCenter(center, zoom, options)	Map	Устанавливает центр и коэффициент масштабирования карты. Центр задается в географических координатах.
setGlobalPixelCenter(globalPixelCenter, zoom, options)	Map	Устанавливает центр и коэффициент масштабирования карты. Центр задается в глобальных пиксельных координатах.
setType(type)	Map	Устанавливает тип карты.
setZoom(zoom, options)	Map	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.

Пример:

```
myMap.balloon.open(myMap.getCenter(), {
    content: 'Hello Yandex!'
}, {
    closeButton: true
});
```

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Пример:

```
// Подгоним размер карты под новый размер контейнера
// (например, если изменилась верстка страницы или карта была инициализирована
// в скрытом состоянии)
map.container.fitToViewport();
```

controls

```
map.control.Manager controls
```

Элементы управления картой.

Пример:

```
myMap.controls.add('zoomControl', {right: '40px', top: '5px'});
```

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('&copy; Vassily Poupkine');
```

cursors

```
util.CursorManager cursors
```

Менеджер курсоров карты.

events

```
event.PriorityManager events
```

Менеджер событий карты. Поддерживает подписку с приоритетами. Бросает события типа **MapEvent**.

Примеры:

1.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Поставим метку по клику над картой
map.events.add('click', function (e) {
    // Географические координаты точки клика можно узнать
    // посредством вызова .get('coordPosition')
    var position = e.get('coordPosition');
    map.geoObjects.add(new ymaps.Placemark(position));
});
```

2.

```
// Запретим пользователю уводить карту за пределы заданной области
// и подыматься выше определённого уровня масштабирования.

// Геокоординаты области, которой мы хотим ограничить просмотр карты
var bounds = [[45, 20], [65, 60]];
// Запретим пользователю подыматься на уровни масштабирования меньше 5
map.options.set('minZoom', 5);
// Будем слушать изменение области показа карты и плавно возвращать
// пользователя обратно в пределы заданной области.
map.events.add('boundschange', function (e) {
    // Новая область показа карты
    var newBounds = e.get('newBounds'),
    // Считаем, насколько нужно сместить центр карты, чтобы
    // вернуть пользователя обратно в заданные границы
        offset = [
            bounds[0][0] > newBounds[0][0] ?
                bounds[0][0] - newBounds[0][0] :
               (bounds[1][0] < newBounds[1][0] ? bounds[1][0] - newBounds[1][0] :
0),
            bounds[0][1] > newBounds[0][1] ?
                bounds[0][1] - newBounds[0][1] :
               (bounds[1][1] < newBounds[1][1] ? bounds[1][1] - newBounds[1][1] :
0)
        ];
     // Если пользователь вышел за границы, возвращаем его обратно
     if (offset[0] || offset[1]) {
         map.setCenter([
             map.getCenter()[0] + offset[0],
             map.getCenter()[1] + offset[1]
         ]);
     }
});
```

geoObjects

```
map.GeoObjects geoObjects
```

Менеджер геообъектов карты.

hint

```
map.Hint hint
```

Хинт карты.

Пример:

```
myMap.hint.show(myMap.getCenter(), {
    hintContent: 'Hello Yandex!'
}, {
    hintHasHideButton: true
});
```

layers

```
map.layer.Manager layers
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Менеджер слоёв карты.

См.: Layer

Пример:

```
// Добавим слой пользовательских тайлов на карту
map.layers.add(new ymaps.Layer('http://some.server/tiles?&c'));
```

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

```
actionbegin
```

Началось новое плавное движение карты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- action — запущенное действие.

actionbreak

```
actionbreak
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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

```
actionend
```

Окончание плавного движения карты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- action — остановленное действие.

actiontick

```
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

```
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

```
balloonclose
```

Закрыт балун

balloonopen

```
balloonopen
```

Открыт балун на карте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- balloon — экземпляр балуна.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

boundschange

```
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;
- newCenter — новый центр карты в геокоординатах;
- oldZoom — предыдущий уровень машсштабирования;
- 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

```
destroy
```

Карта уничтожена.

hinthide

```
hinthide
```

Скрыт хинт.

hintshow

```
hintshow
```

Показан хинт на карте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- hint — экземпляр хинта.

optionschange

```
optionschange
```

Изменились опции карты.

sizechange

```
sizechange
```

Изменился размер карты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldSize — предыдущий размер карты;
- newSize — новый размер карты.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

typechange

```
typechange
```

Изменился тип карты. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldType — предыдущий тип карты;
- newType — новый тип карты.

Описание методов

destroy

```
destroy()
```

Уничтожает карту.

getBounds

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

Возвращает двумерный массив геокоординат левого нижнего и правого верхнего углов области показа карты.

getCenter

```
{Number[]} getCenter()
```

Возвращает географические координаты текущего центра карты.

getGlobalPixelCenter

```
{Number[]} getGlobalPixelCenter()
```

Возвращает глобальные пиксельные координаты текущего центра карты.

Пример:

```
// Сдвинем карту на 10 пикселей влево
var position = map.getGlobalPixelCenter();
map.setGlobalPixelCenter([ position[0] - 10, position[1] ]);
```

getType

```
{String|MapType} getType()
```

Возвращает текущий тип карты.

getZoom

```
{Number} getZoom()
```

Возвращает текущий коэффициент масштабирования карты.

panTo

```
{Map} panTo(center, options)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Устанавливает центр карты. Если передан массив точек, то карта будет перемещаться от одной точки к другой.

Возвращает ссылку на карту.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
`center`		Number[]|Object[]	✔	Центр карты или массив точек для последовательного обхода.
`options`		Object	✔	Опции.
	`callback`	Function		Функция, которая вызывается по окончании выполнения действия. В функцию передается параметр err. В случае корректного выполнения действия err=null, в остальных случаях параметр является признаком ошибки.
	`checkZoomRange`	Boolean		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. Значение по умолчанию: false
	`delay`	Number		Время задержки выполнения очередного перемещения в милисекундах. Значение по умолчанию: 1000
	`duration`	Number		Время анимации в миллисекундах. Значение по умолчанию: 500

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	flying	Boolean		Разрешить уменьшать и затем увеличивать зум карты при перемещении между точками. Если расстояние между точками не превышает двух размеров контейнера карты, то карта просто переместит центр в указанную точку. Если расстояние больше, то при включенной опции flying карта выставит себе меньший коэффициент масштабирования, затем сместится к конечной точке и выставит требуемый коэффициент масштабирования (эмуляция полета над картой). Если options.flying=false, то карта переместится в конечную точку мгновенно, без анимации. Значение по умолчанию: true
	timingFunction	String		Тайминг-функция. Может задаваться как с помощью отдельных коэффициентов: 'cubic-bezier(0,1,1,0)', так и строкой: 'linear', 'ease', 'ease-in', 'ease-out', 'ease-in-out'. Значение по умолчанию: 'ease-in-out'

* Обязательный параметр/опция.

Пример:

```
// Полёт из Калининграда во Владивосток через Москву
map.setCenter([54.704815, 20.466380], 10);
map.panTo([
    [ 55.751574, 37.573856 ],
    [ 43.134091, 131.928478 ]
], {
    callback: function () {
        alert('Прилетели!');
    }
});
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

setBounds

```
{Map} setBounds(bounds, options)
```

Позиционирует карту для показа переданного района.

Возвращает ссылку на себя.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
bounds		Number[][]	✔	Границы области показа.
options		Object		Опции.
	callback	Function		Функция, которая вызывается по окончании выполнения действия. В функцию передается параметр err. В случае корректного выполнения действия err=null, в остальных случаях параметр является признаком ошибки.
	checkZoomRange	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. Значение по умолчанию: false
	duration	Number		Время анимации в миллисекундах. Значение по умолчанию: 0
	preciseZoom	Boolean		Возможность использования дробных коэффициентов масштабирования.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
				Значение по умолчанию: false
	timingFunction	Function		Тайминг-функция.
	zoomMargin	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. Значение по умолчанию: 0

* Обязательный параметр/опция.

Пример:

```
// Новые центр и коэффициент масштабирования карты вычисляются на основе
текущего
// состояния 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.
myMap.setBounds([[60,-40], [20,60]], {
     checkZoomRange: true,
     callback: function(err) {
         if (err) {
             // Не удалось показать заданный регион
             // ...
         }
     }
});
```

setCenter

```
{Map} setCenter(center, zoom, options)
```

Устанавливает центр и коэффициент масштабирования карты. Центр задается в географических координатах.

Возвращает ссылку на себя.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
center		Number[]	✔	Геокоординаты центра карты.
zoom		Number		Коэффициент масштабирования карты.
options		Object		Опции карты.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	callback	Function		Функция, которая вызывается по окончании выполнения действия. В функцию передается параметр err. В случае корректного выполнения действия err=null, в остальных случаях параметр является признаком ошибки.
	checkZoomRange	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. Значение по умолчанию: false
	duration	Number		Время анимации в миллисекундах. Значение по умолчанию: 0
	timingFunction	Function		Тайминг-функция.

* Обязательный параметр/опция.

Пример:

```
myMap.setCenter([40, 50], 3, {
    checkZoomRange: true
});
```

setGlobalPixelCenter

```
{Map} setGlobalPixelCenter(globalPixelCenter, zoom, options)
```

Устанавливает центр и коэффициент масштабирования карты. Центр задается в глобальных пиксельных координатах.

Возвращает ссылку на карту.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
globalPixelCenter		Number[]	✔	Пиксельные координаты нового центра карты.
zoom		Number		Коэффициент масштабирования карты.
options		Object		Опции.
	callback	Function		Функция, которая вызывается по окончании выполнения действия. В функцию передается параметр err. В случае корректного выполнения действия err=null, в остальных случаях параметр является признаком ошибки.
	checkZoomRange	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. Значение по умолчанию: false
	duration	Number		Время анимации в миллисекундах. Значение по умолчанию: 0
	timingFunction	Function		Тайминг-функция.

* Обязательный параметр/опция.

Пример:

```
// Сдвинем карту на 10 пикселей влево
var position = map.getGlobalPixelCenter();
map.setGlobalPixelCenter([ position[0] - 10, position[1] ]);
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

setType

```
{Map} setType(type)
```

Устанавливает тип карты.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
type	String|MapType	✔	Тип карты. Может быть ключом или или экземпляром класса MapType. Список доступных ключей: • 'yandex#map' — тип карты "схема"; • 'yandex#satellite' — тип карты "спутник"; • 'yandex#hybrid' — тип карты "гибрид"; • 'yandex#publicMap' — тип карты "народная карта"; • 'yandex#publicMapHybrid' — тип карты "народный гибрид". Ключи 'yandex#publicMap' и 'yandex#publicMapHybrid' доступны только для России и Украины.

* Обязательный параметр/опция.

Пример:

```
myMap.setType('yandex#hybrid');
```

setZoom

```
{Map} setZoom(zoom, options)
```

Устанавливает коэффициент масштабирования карты.

Возвращает ссылку на себя.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
zoom		Number	✔	Коэффициент масштабирования карты.
options		Object		Опции.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	callback	Function		Функция, которая вызывается по окончании выполнения действия. В функцию передается параметр err. В случае корректного выполнения действия err=null, в остальных случаях параметр является признаком ошибки.
	checkZoomRange	Boolean		Проверка возможности установить указанный коэффициент масштабирования. Значение по умолчанию: false
	duration	Number		Время анимации в миллисекундах. Значение по умолчанию: 0

* Обязательный параметр/опция.

Пример:

```
myMap.setZoom(4, {duration: 1000});
```

MapEvent

Входит в пакеты package.full, package.map, package.standard.

Расширяет **Event**.

Объект, описывающий произошедшее над картой событие. Имена полей, доступных через метод Event.get:

- coords — географические координаты точки, в которой произошло событие (также доступно под именем coordPosition);
- globalPixels — координаты события в глобальных пикселях от левого верхнего угла мира (также доступно под именем globalPixelPosition);
- pagePixels — координаты события в 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, map)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
originalEvent	Object	✔	Данные, связанные с событием.
map	Map	✔	Карта, над которой произошло событие.

* Обязательный параметр/опция.

Примеры:

1.

```
// Откроем балун в точке клика по карте
map.events.add('click', function (e) {
    map.balloon.open(e.get('coordPosition'), 'Щелк!');
});
```

2.

```
// Симулируем двойной клик над картой
map.events.fire('dblclick', new ymaps.MapEvent({
    type: 'dblclick',
    position: [300, 400],
    target: map
}, map));

// Также можно передавать в метод fire просто хэш с данными.
// Менеджер событий карты автоматически создаст объект MapEvent.
map.events.fire('dblclick', {
    position: [300, 400]
});
```

Методы

Имя	Возвращает	Описание
callMethod(name)	Object	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. Унаследован от Event.
get(name)	Object	Возвращает значение поля из originalEvent. originalEvent всегда имеет следующие поля: • type — строковый тип события; • target — ссылка на объект, породивший событие. Унаследован от Event.
getMap()	Map	Возвращает карту, над которой произошло событие.
isDefaultPrevented()	Boolean	Проверяет, отменена ли реакция по умолчанию на событие в рамках системы событий API Яндекс.Карт. Унаследован от Event.
isImmediatePropagationStopped()	Boolean	Определяет, прекращено ли распространение события в системе событий API Яндекс.Карт.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **Event**.
isPropagationStopped()	Boolean	Проверяет, остановлено ли распространение события по иерархии объектов и коллекций в системе событий API Яндекс.Карт. Унаследован от **Event**.
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. Унаследован от **Event**.
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. Унаследован от **Event**.
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**.

Описание методов

getMap

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

Возвращает карту, над которой произошло событие.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

mapType

mapType.storage

Входит в пакеты package.full, package.map, package.standard.

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('Мой тип карты', ['my#layer']);
// Добавляем его в хранилище типов карты
ymaps.mapType.storage.add('my#mapType', myType);
// Теперь можем устанавливать свой тип карте
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

MapType

Входит в пакеты package.full, package.map, package.standard.

Тип карты.

Конструктор | Методы

Конструктор

```
MapType(name, layers)
```

Создает экземпляр типа карты.

Параметры:

Параметр	Тип	*	Описание
name	String	✔	Название типа.
layers	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 () {
        var promise = new ymaps.util.Promise();
        promise.resolve('Data, imagery and map information provided by MapQuest,
Open Street Map and contributors, CC-BY-SA');
        return promise;
    };
    // Диапазон доступных масштабов
    layer.getZoomRange = function () {
        var promise = new ymaps.util.Promise();
        promise.resolve([0, 18]);
        return promise;
    };

    return layer;
};
// Добавим слой под ключом
ymaps.layer.storage.add('mq#aerial', MQLayer);
// Создадим тип карты, состоящий из слоёв '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 задавать наш тип карты любой карте
map.setType('mq_ya#hybrid');
// И миникарте тоже
map.controls.add(new ymaps.control.MiniMap({ type: 'mq_ya#hybrid' }));
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
getLayers()	Function[]|String[]	Возвращает список слоев данного типа карты — массив конструкторов или ключей для слоев.
getName(map)	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(map)
```

Возвращает название типа карты.

Параметры:

Параметр	Тип	*	Описание
map		✔	

* Обязательный параметр/опция.

Monitor

Входит в пакеты package.full, package.map, package.standard.

Объект, отслеживающий изменения определенных полей данных на заданном менеджере данных. Также может использоваться для отслеживания изменений опций.

Конструктор | Методы

Конструктор

```
Monitor(dataManager)
```

Параметры:

Параметр	Тип	*	Описание
dataManager	IDataManager|IOptionManager	✔	Менеджер данных.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// 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	Инициирует проверку изменения значений наблюдаемых полей данных.
get(name)	Object	Возвращает текущее значение одного из наблюдаемых полей данных.
remove(name)	Monitor	Отключает наблюдение за определенным полем, либо группой полей данных.
removeAll()	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	Свойства параметра	Тип	*	Описание
name		String|String[]	✔	Имя, либо массив имен полей данных, за которыми устанавливается наблюдение.
changeCallback		Function	✔	Обработчик изменения поля данных, либо одного из полей данных группы.
context		Object		Контекст исполнения обработчика изменений данных, а также опциональных обработчиков.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
params		Object		Опциональные параметры.
	compareCallback	Function		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 compareCallbacks.
	compareCallbacks	Object		Хэш вида {имя поля данных: ссылка на обработчик}. Данный параметр позволяет задать индивидуальные обработчики сравнения значений для различных полей данных в группе.
	defaultValue	Object		Значение по умолчанию, используемое в том случае, если поле данных не определено.
	defaultValues	Object		Хэш вида {имя поля данных: значение по умолчанию}. Данный параметр позволяет задать индивидуальные значения по умолчанию для различных полей данных в группе.
	resolveCallback	Function		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 resolveCallbacks.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	resolveCallbacks	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()
```

Инициирует проверку изменения значений наблюдаемых полей данных.

Возвращает ссылку на себя.

get

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

Возвращает текущее значение одного из наблюдаемых полей данных.

Параметры:

Параметр	Тип	*	Описание
name	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	Тип	*	Описание
name	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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

option

option.Manager

Входит в пакеты package.full, package.map, package.standard.

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	Тип	Описание
options	Object	Хэш опций.
parent	IOptionManager	Родительский менеджер опций.
name	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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Пример использования опции "preset".
var optionManager = new ymaps.option.Manager({
        preset: 'twirl#blueIcon'
    }),
    subOptionManager = new ymaps.option.Manager();
// Данных нет, так как subOptionManager пуст.
alert(subOptionManager.get('iconImageSize');
// Связываем два менеджера.
subOptionManager.setParent(optionManager);
// [37, 42] - значение получено из пресета в родительском менеджере.
alert(subOptionManager.get('iconImageSize');
// Перекрываем значение iconImageSize на уровне subOptionManager.
subOptionManager.set('iconImageSize', [10, 12]);
// [10, 12] - значение получено из subOptionManager.
alert(subOptionManager.get('iconImageSize');
// Отменяем перекрытие iconImageSize.
subOptionManager.unset('iconImageSize');
// [37, 42] - значение снова получено 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	Изменение опций. Унаследовано от IOptionManager.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.

Методы

Имя	Возвращает	Описание
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
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. Унаследован от IOptionManager.
getAll()	Object	Возвращает ссылку на внутренний хэш хранящий значения опций. Унаследован от IOptionManager.
getName()	String	Возвращает имя менеджера опций.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **IOptionManager**.
getNative(key)	Object	Возвращает значение заданной опции, определенное на данном уровне иерархии опций, т.е. в данном менеджере. Унаследован от **IOptionManager**.
getParent()	**IOptionManager**|null	Возвращает родительский менеджер опций. Унаследован от **IOptionManager**.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от **IFreezable**.
resolve(key, name)	Object	Метод, предназначенный для вызова дочерними менеджерами опций. Унаследован от **IOptionManager**.
set(key, value)	**option.Manager**	Задает 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.
setName(name)		Задает имя менеджера опций. Унаследован от **IOptionManager**.
setParent(parent)	**IChild**	Задает родительский менеджер опций. Унаследован от **IOptionManager**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.
unset(keys)	**option.Manager**	Стирает значения заданных опций в данном менеджере.
unsetAll()	**option.Manager**	Стирает значения всех опции в данном менеджере.

Описание методов

set

```
{option.Manager} set(key, value)
```

Задает 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
key	Object|String	✔	Имя опции, либо хеш вида {"имя опции": "значение опции"}.
value	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 {@link OptionManager.change}.
```

2.

```
// Задание опций раздельно.
myMap.options
    .set("dblClickZoomCentering", true)
    .set("dblClickFloatZoom", true);
// После каждого set будет генерироваться событие {@link OptionManager.change}.
```

3.

```
// Использование freeze для минимизации количества событий {@link
OptionManager.change}.
myMap.options.freeze();
myMap.options.set({
    dblClickZoomCentering: true,
    dblClickFloatZoom: true
});
myMap.options.set('cursor', 'zoom');
myMap.unfreeze();
// Генерируется одно событие {@link OptionManager.change}.
```

unset

```
{option.Manager} unset(keys)
```

Стирает значения заданных опций в данном менеджере.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
keys	String|String[]	✔	Имя опции или массив имен опций, значения которых должны быть аннулированы.

* Обязательный параметр/опция.

Пример:

```
map.options.unset(['dblClickZoomCentering', 'dblClickFloatZoom']);
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

unsetAll

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

Стирает значения всех опции в данном менеджере.

Возвращает ссылку на себя.

Пример:

```
var geoObject = new ymaps.Placemark([37, 55], {}, {preset:'twirl#blueIcon'});
myMap.geoObjects.add(geoObject);
// Произвольным образом меняем стиль.
geoObject.options.set({
    iconImageHref: 'http://mysite.ru/icon.png',
    iconImageSize: [16, 16]
});
// Восстанавливаем первоначальный вид.
geoObject.options
    // Во избежание двойной реакции геообъекта
    // на изменение опций сначала вызываем freeze, а после задания
    // всех значений вызываем unfreeze.
    .freeze()
    .unsetAll()
    .set('preset','twirl#blueIcon')
    .unfreeze();
```

option.presetStorage

Входит в пакеты package.full, package.map, package.standard.

Статический объект.

Экземпляр класса **util.Storage**

Хранилище предустановленных опций. Список ключей, доступных при подключении пакетов package.full и package.standard:
Метки без содержимого

Значок	Ключ	Значок	Ключ
	'twirl#blueIcon'		'twirl#orangeIcon'
	'twirl#darkblueIcon'		'twirl#pinkIcon'
	'twirl#darkgreenIcon'		twirl#redIcon'
	'twirl#darkorangeIcon'		'twirl#violetIcon'
	'twirl#greenIcon'		'twirl#whiteIcon'

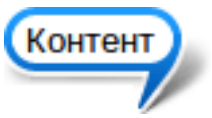
Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

	'twirl#greyIcon'		'twirl#yellowIcon'
	'twirl#lightblueIcon'		'twirl#brownIcon'
	'twirl#nightIcon'		'twirl#blackIcon'

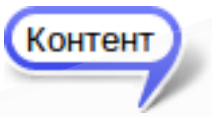
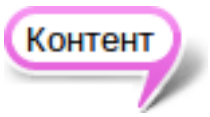
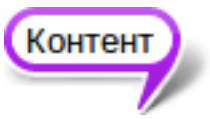
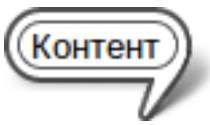
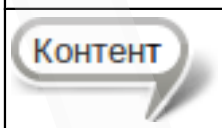
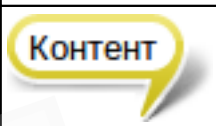
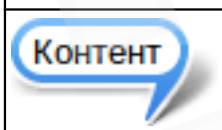
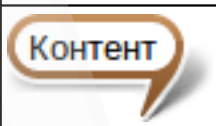
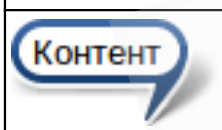
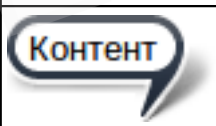
Метки без содержимого с точкой в центре

Значок	Ключ	Значок	Ключ
	'twirl#blueDotIcon'		'twirl#orangeDotIcon'
	'twirl#darkblueDotIcon'		'twirl#pinkDotIcon'
	'twirl#darkgreenDotIcon'		'twirl#redDotIcon'
	'twirl#darkorangeDotIcon'		'twirl#violetDotIcon'
	'twirl#greenDotIcon'		'twirl#whiteDotIcon'
	'twirl#greyDotIcon'		'twirl#yellowDotIcon'
	'twirl#lightblueDotIcon'		'twirl#brownDotIcon'
	'twirl#nightDotIcon'		'twirl#blackDotIcon'

Метки с текстом (иконки тянутся под контент)

Значок	Ключ	Значок	Ключ
	'twirl#blueStretchyIcon'		'twirl#orangeStretchyIcon'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

	'twirl#darkblueStretchyIcon'		'twirl#pinkStretchyIcon'
	'twirl#darkgreenStretchyIcon'		'twirl#redStretchyIcon'
	'twirl#darkorangeStretchyIcon'		'twirl#violetStretchyIcon'
	'twirl#greenStretchyIcon'		'twirl#whiteStretchyIcon'
	'twirl#greyStretchyIcon'		'twirl#yellowStretchyIcon'
	'twirl#lightblueStretchyIcon'		'twirl#brownStretchyIcon'
	'twirl#nightStretchyIcon'		'twirl#blackStretchyIcon'

Стандартные значки (пиктограммы)

Значок	Ключ	Значок	Ключ	Значок	Ключ
	'twirl#airplaneIcon'		'twirl#fishingIcon'		'twirl#skiingIcon'
	'twirl#anchorIcon'		'twirl#gasStationIcon'		'twirl#smartphoneIcon'
	'twirl#badmintonIcon'		'twirl#gymIcon'		'twirl#workshopIcon'
	'twirl#bankIcon'		'twirl#hospitalIcon'		'twirl#storehouseIcon'
	'twirl#barIcon'		'twirl#houseIcon'		'twirl#swimmingIcon'
	'twirl#barberShopIcon'		'twirl#keyMasterIcon'		'twirl#tailorShopIcon'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

	'twirl#bicycleIcon'		'twirl#mailPostIcon'		'twirl#tennisIcon'
	'twirl#bowlingIcon'		'twirl#metroKievIcon'		'twirl#tireIcon'
	'twirl#buildingsIcon'		'twirl#metroMoscowIcon'		'twirl#truckIcon'
	'twirl#busIcon'		'twirl#metroStPetersburgIcon'		'twirl#trainIcon'
	'twirl#cafeIcon'		'twirl#metroYekaterinburgIcon'		'twirl#tramwayIcon'
	'twirl#campingIcon'		'twirl#motobikeIcon'		'twirl#trolleybusIcon'
	'twirl#carIcon'		'twirl#mushroomIcon'		'twirl#wifiIcon'
	'twirl#cellularIcon'		'twirl#phoneIcon'		'twirl#wifiLogoIcon'
	'twirl#cinemaIcon'		'twirl#photographerIcon'		'twirl#turnLeftIcon'
	'twirl#downhillSkiingIcon'		'twirl#pingPongIcon'		'twirl#turnRightIcon'
	'twirl#dpsIcon'		'twirl#restaurauntIcon'		'twirl#arrowDownLeftIcon'
	'twirl#dryCleanerIcon'		'twirl#shipIcon'		'twirl#arrowDownRightIcon'
	'twirl#electricTrainIcon'		'twirl#shopIcon'		'twirl#arrowLeftIcon'
	'twirl#factoryIcon'		'twirl#skatingIcon'		'twirl#arrowRightIcon'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

	'twirl#theaterIcon'		'twirl#stadiumIcon'		'twirl#arrowUpIcon'

Значки кластеров

Маленький значок	Средний значок	Большой значок	Ключ
			'twirl#blackClusterIcons'
			'twirl#invertedBlackClusterIcons'
			'twirl#blueClusterIcons'
			'twirl#invertedBlueClusterIcons'
			'twirl#brownClusterIcons'
			'twirl#invertedBrownClusterIcons'
			'twirl#darkblueClusterIcons'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Маленький значок	Средний значок	Большой значок	Ключ
			'twirl#invertedDarkblueClusterIcons'
			'twirl#darkgreenClusterIcons'
			'twirl#invertedDarkgreenClusterIcons'
			'twirl#darkorangeClusterIcons'
			'twirl#invertedDarkorangeClusterIcons'
			'twirl#greenClusterIcons'
			'twirl#invertedGreenClusterIcons'
			'twirl#greyClusterIcons'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Маленький значок	Средний значок	Большой значок	Ключ
			'twirl#invertedGreyClusterIcons'
			'twirl#lightblueClusterIcons'
			'twirl#invertedLightblueClusterIcons'
			'twirl#nightClusterIcons'
			'twirl#invertedNightClusterIcons'
			'twirl#orangeClusterIcons'
			'twirl#invertedOrangeClusterIcons'
			'twirl#pinkClusterIcons'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Маленький значок	Средний значок	Большой значок	Ключ
			'twirl#invertedPinkClusterIcons'
			'twirl#redClusterIcons'
			'twirl#invertedRedClusterIcons'
			'twirl#violetClusterIcons'
			'twirl#invertedVioletClusterIcons'
			'twirl#whiteClusterIcons'
			'twirl#invertedWhiteClusterIcons'
			'twirl#yellowClusterIcons'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Маленький значок	Средний значок	Большой значок	Ключ
			'twirl#invertedYellowClusterIcons'

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

overlay

overlay.hotspot

overlay.hotspot.Base

Расширяет **IOverlay**.

Базовый хотспотный оверлей.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.hotspot.Base(geometry, data, options)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
geometry		**IPixelGeometry**	Пиксельная геометрия фигуры.
data		**IDataManager**|Object	Данные.
options		Object	Опции.
	cursor	String	Курсор при наведении мыши.
	interactivityModel	String	Модель интерактивности. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**. Значение по умолчанию: 'default#geoObject'

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от **IOverlay**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от **IOverlay**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
createHotspotGeometry(geometry, data)	**IPixelGeometry**	Возвращает новая пиксельная геометрия для хотспота.
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

Описание методов

createHotspotGeometry

```
{IPixelGeometry} createHotspotGeometry(geometry, data)
```

Возвращает новая пиксельная геометрия для хотспота.

Параметры:

Параметр	Тип	*	Описание
geometry	**IPixelGeometry**	✔	Пиксельная геометрия.
data	**IDataManager**	✔	Менеджер данных.

* Обязательный параметр/опция.

overlay.hotspot.Circle

Входит в пакеты package.full, package.overlays.

Расширяет **overlay.hotspot.Base**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Хотспотный оверлей круга.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.hotspot.Circle()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от IOverlay.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от IOverlay.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IOverlay.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
createHotspotGeometry(geometry, data)	**IPixelGeometry**	Возвращает новая пиксельная геометрия для хотспота. Унаследован от **overlay.hotspot.Base**.
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

overlay.hotspot.Placemark

Входит в пакеты package.full, package.overlays.

Расширяет **overlay.hotspot.Base**.

Хотспотный оверлей точки.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.hotspot.Placemark()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от **IOverlay**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от **IOverlay**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
createHotspotGeometry(geometry, data)	IPixelGeometry	Возвращает новая пиксельная геометрия для хотспота. Унаследован от overlay.hotspot.Base.
getData()	Object	Возвращает объект данных макета. Унаследован от IOverlay.
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay.
getMap()	Map|null	Возвращает ссылку на текущую карту. Унаследован от IOverlay.
setData(data)		Задает данные оверлея. Унаследован от IOverlay.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от IOverlay.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от IOverlay.

overlay.hotspot.Polygon

Входит в пакеты package.full, package.overlays.

Расширяет **overlay.hotspot.Base**.

Хотспотный оверлей многоугольника.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.hotspot.Polygon()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от **IOverlay**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от **IOverlay**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
createHotspotGeometry(geometry, data)	**IPixelGeometry**	Возвращает новая пиксельная геометрия для хотспота. Унаследован от **overlay.hotspot.Base**.
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

overlay.hotspot.Polyline

Входит в пакеты package.full, package.overlays.

Расширяет **overlay.hotspot.Base**.

Хотспотный оверлей линии.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.hotspot.Polyline()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от IOverlay.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от IOverlay.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IOverlay.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
createHotspotGeometry(geometry, data)	**IPixelGeometry**	Возвращает новая пиксельная геометрия для хотспота. Унаследован от **overlay.hotspot.Base**.
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

overlay.hotspot.Rectangle

Входит в пакеты package.full, package.overlays.

Расширяет **overlay.hotspot.Base**.

Хотспотный оверлей прямоугольника.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.hotspot.Rectangle()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от **IOverlay**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от **IOverlay**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
createHotspotGeometry(geometry, data)	**IPixelGeometry**	Возвращает новая пиксельная геометрия для хотспота. Унаследован от **overlay.hotspot.Base**.
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

overlay.html

overlay.html.Balloon

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

HTML-оверлей балуна.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.html.Balloon(geometry, data, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		**IPixelPointGeometry**	✔	Пиксельная геометрия фигуры.
data		**IDataManager**|Object		Данные.
options		Object		Опции.
	cursor	String		Курсор при наведении мыши.
	interactivityModel	String		Модель интерактивности. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**. Значение по умолчанию: 'default#opaque'
	layout	Function|String		Макет. (Тип: конструктор объекта с интерфейсом ILayout).
	offset	Array		Смещение в пикселях. Значение по умолчанию: [0,0]
	pane	String		Контейнер, в который будет помещен макет балуна. Значение по умолчанию: "overlays"
	shadow	Boolean		Флаг наличия тени. Значение по умолчанию: true
	shadowLayout	Function|String		Макет тени (Тип: конструктор объекта с интерфейсом ILayout).

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	shadowOffset	Array		Смещение тени в пикселях. Значение по умолчанию: [0,0]
	shadowPane	Array		Контейнер, в который будет помещен макет тени балуна. Значение по умолчанию: "shadows"
	zIndex	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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от IOverlay.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от IOverlay.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getBalloonElement()	HTMLElement	Возвращает родительский элемент макета балуна.
getBalloonLayout()	**ILayout**	Возвращает макет балуна.
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getElement()	HTMLElement	Возвращает родительский элемент макета балуна.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay.
getLayout()	ILayout	Возвращает макет балуна.
getMap()	Map|null	Возвращает ссылку на текущую карту. Унаследован от IOverlay.
getShadowElement()	HTMLElement	Возвращает родительский элемент макета тени балуна.
getShadowLayout()	ILayout	Возвращает макет тени балуна.
setData(data)		Задает данные оверлея. Унаследован от IOverlay.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от IOverlay.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от IOverlay.

Описание методов

getBalloonElement

```
{HTMLElement} getBalloonElement()
```

Возвращает родительский элемент макета балуна.

getBalloonLayout

```
{ILayout} getBalloonLayout()
```

Возвращает макет балуна.

getElement

```
{HTMLElement} getElement()
```

Возвращает родительский элемент макета балуна.

getLayout

```
{ILayout} getLayout()
```

Возвращает макет балуна.

getShadowElement

```
{HTMLElement} getShadowElement()
```

Возвращает родительский элемент макета тени балуна.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getShadowLayout

```
{ILayout} getShadowLayout()
```

Возвращает макет тени балуна.

overlay.html.Label

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

Простой HTML-оверлей.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.html.Label(geometry, data, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		IPixelPointGeometry	✔	Пиксельная геометрия фигуры.
data		IDataManager|Object		Данные.
options		Object		Опции.
	cursor	String		Курсор при наведении мыши.
	interactivityModel	String		Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage. Значение по умолчанию: 'default#opaque'
	layout	ILayout|String		Макет. Значение по умолчанию: theme.twirl.label.Layout
	pane	String		Контейнер, в который будет помещен оверлей. Значение по умолчанию: "overlays"
	zIndex	Number		z-index элемента.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от IOverlay.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от IOverlay.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IOverlay.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getElement()	HTMLElement	Возвращает родительский элемент макета хинта.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getLayout()	**ILayout**	Возвращает макет хинта.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

getElement

```
{HTMLElement} getElement()
```

Возвращает родительский элемент макета хинта.

getLayout

```
{ILayout} getLayout()
```

Возвращает макет хинта.

overlay.html.Placemark

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

HTML-оверлей метки.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.html.Placemark(geometry, data, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		IPixelPointGeometry	✔	Пиксельная геометрия.
data		IDataManager|Object		Данные.
options		Object		Опции.
	cursor	String		Курсор при наведении мыши.
	interactivityModel	String		Модель интерактивности. Доступные ключи и их значения перечислены в описании interactivityModel.storage. Значение по умолчанию: 'default#geoObject'
	layout	Function|String		Макет. (Тип: конструктор объекта с интерфейсом ILayout).
	offset	Array		Смещение в пикселях. Значение по умолчанию: [0,0]

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	pane	String		Контейнер, в который будет помещен макет метки. Значение по умолчанию: "overlays"
	shadow	Boolean		Флаг наличия тени. Значение по умолчанию: false
	shadowLayout	Function|String		Макет тени. (Тип: конструктор объекта с интерфейсом ILayout).
	shadowOffset	Array		Смещение тени в пикселях. Значение по умолчанию: [0,0]
	shadowPane	Array		Контейнер, в который будет помещен макет тени метки. Значение по умолчанию: "shadows"
	zIndex	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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от **IOverlay**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от **IOverlay**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета. Унаследован от IOverlay.
getElement()	HTMLElement	Возвращает родительский элемент макета иконки.
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию. Унаследован от IOverlay.
getIconElement()	HTMLElement	Возвращает родительский элемент макета иконки.
getIconLayout()	ILayout	Возвращает макет иконки.
getLayout()	ILayout	Возвращает макет иконки.
getMap()	Map|null	Возвращает ссылку на текущую карту. Унаследован от IOverlay.
getShadowElement()	HTMLElement	Возвращает родительский элемент макета тени иконки.
getShadowLayout()	ILayout	Возвращает макет тени иконки.
setData(data)		Задает данные оверлея. Унаследован от IOverlay.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от IOverlay.
setMap(map)		Задает 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

```
{ILayout} getIconLayout()
```

Возвращает макет иконки.

getLayout

```
{ILayout} getLayout()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает макет иконки.

getShadowElement

```
{HTMLElement} getShadowElement()
```

Возвращает родительский элемент макета тени иконки.

getShadowLayout

```
{ILayout} getShadowLayout()
```

Возвращает макет тени иконки.

overlay.html.rectangle

overlay.html.rectangle.Layout

Расширяет **ILayout**.

Собственный макет для HTML оверлея прямоугольника.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.html.rectangle.Layout(dataSet)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
dataSet		Object	✔	
	options.borderRadius	Number		Радиус скругления углов.
	options.fillColor	String		Цвет заливки **graphics.style.color**.
	options.fillImageHref	String		Фоновое изображение. При включении данной опции значение fillColor игнорируется.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	`options.fillMethod`	String		Тип заливки фоном. Может принимать одно из двух значений: • 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. Значение по умолчанию: 'stretch'
	`options.fillOpacity`	Number		Прозрачность заливки.
	`options.opacity`	Number		Общая прозрачность.
	`options.strokeColor`	String		Цвет линии **graphics.style.color**.
	`options.strokeStyle`	Number[]|String		Стиль обводки, поддерживаемый стандартным CSS-свойством *border-style*.
	`options.strokeWidth`	Number		Ширина линии.

* 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 ввиду, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от **ILayout**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от **ILayout**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от IDomEventEmitter.
parentelementchange	Изменение родительского элемента. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от ILayout.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от ILayout.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от ILayout.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от ILayout.
getData()	Object	Возвращает объект данных макета. Унаследован от ILayout.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от ILayout.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от ILayout.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от ILayout.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

overlay.html.Rectangle

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

HTML-оверлей прямоугольника.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.html.Rectangle(geometry, data, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		**IPixelRectangleGeometry**	✔	Пиксельная геометрия фигуры.
data		**IDataManager**|Object		Данные.
options		Object		Опции оверлея. Используя этот параметр можно задавать как опции самого оверлея, так и опции его макета.
	cursor	String		Курсор при наведении мыши.
	interactivityModel	String		Модель интерактивности. Доступные ключи и их значения перечислены в описании **interactivityModel.storage**. Значение по умолчанию: 'default#geoObject'
	layout	**ILayout**|String		Макет. Значение по умолчанию: overlay.html.rectangle.Layout
	pane	String		Контейнер, в который будет помещен оверлей. Значение по умолчанию: "overlays"
	zIndex	Number		z-index элемента.

* Обязательный параметр/опция.

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от IOverlay.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от IOverlay.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IOverlay.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getElement()	HTMLElement	Возвращает родительский элемент макета прямоугольника.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getLayout()	**ILayout**	Возвращает макет прямоугольника.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

Описание методов

getElement

```
{HTMLElement} getElement()
```

Возвращает родительский элемент макета прямоугольника.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getLayout

```
{ILayout} getLayout()
```

Возвращает макет прямоугольника.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

overlay.interactiveGraphics

overlay.interactiveGraphics.Circle

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

Интерактивный оверлей круга.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.interactiveGraphics.Circle(geometry, data, options)
```

Параметры:

Параметр	Тип	Описание
geometry	**IPixelCircleGeometry**	Пиксельная геометрия фигуры.
data	**IDataManager**|Object	Данные.
options	Object	Опции. • Содержит в себе без префиксов опции **overlay.hotspot.Circle** и **overlay.staticGraphics.Circle**

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от **IOverlay**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от **IOverlay**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

overlay.interactiveGraphics.Placemark

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

Интерактивный оверлей метки-изображения.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.interactiveGraphics.Placemark(geometry, data, options)
```

Параметры:

Параметр	Тип	Описание
`geometry`	**IPixelPointGeometry**	Пиксельная геометрия фигуры.
`data`	**IDataManager**|Object	Данные.
`options`	Object	Опции. • Содержит в себе без префиксов опции **overlay.hotspot.Polygon** и overlay.staticGraphics.CanvasPlacemark

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от **IOverlay**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от **IOverlay**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

overlay.interactiveGraphics.Polygon

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

Интерактивный оверлей многоугольника.

Конструктор | Поля | События | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
overlay.interactiveGraphics.Polygon(geometry, data, options)
```

Параметры:

Параметр	Тип	*	Описание
geometry	IPixelPolygonGeometry	✔	Пиксельная геометрия фигуры.
data	IDataManager|Object		Данные.
options	Object		Опции. • Содержит в себе без префиксов опции overlay.hotspot.Polygon и overlay.staticGraphics.Polygon

* 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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от IOverlay.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от IOverlay.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

overlay.interactiveGraphics.Polyline

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

Интерактивный оверлей линии.

Конструктор | Поля | События | Методы

Конструктор

```
overlay.interactiveGraphics.Polyline(geometry, data, options)
```

Параметры:

Параметр	Тип	*	Описание
geometry	**IPixelLineStringGeometry**	✔	Пиксельная геометрия фигуры.
data	**IDataManager**|Object		Данные.
options	Object		Опции. • Содержит в себе без префиксов опции **overlay.hotspot.Polyline** и **overlay.staticGraphics.Polyline**

* Обязательный параметр/опция.

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от **IOverlay**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от **IOverlay**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию. Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

overlay.interactiveGraphics.Rectangle

Входит в пакеты package.full, package.overlays.

Расширяет **IOverlay**.

Интерактивный оверлей прямоугольника.

Конструктор | Поля | События | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
overlay.interactiveGraphics.Rectangle(geometry, data, options)
```

Параметры:

Параметр	Тип	Описание
geometry	IPixelRectangleGeometry	Пиксельная геометрия фигуры.
data	IDataManager|Object	Данные.
options	Object	Опции. • Содержит в себе без префиксов опции overlay.hotspot.Rectangle и overlay.staticGraphics.Rectangle

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные. Унаследовано от IOverlay.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия. Унаследовано от IOverlay.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IOverlay**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета. Унаследован от **IOverlay**.
getGeometry()	**IPixelGeometry**	Возвращает текущую пиксельную геометрию.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **IOverlay**.
getMap()	**Map**|null	Возвращает ссылку на текущую карту. Унаследован от **IOverlay**.
setData(data)		Задает данные оверлея. Унаследован от **IOverlay**.
setGeometry(geometry)		Задает пиксельную геометрию оверлея. Унаследован от **IOverlay**.
setMap(map)		Задает карту, на которой должен отобразиться оверлей. Унаследован от **IOverlay**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

overlay.staticGraphics

overlay.staticGraphics.Circle

Входит в пакеты package.full, package.overlays.

Расширяет overlay.staticGraphics.Base.

Эллипс.

Конструктор

Конструктор

```
overlay.staticGraphics.Circle(geometry, data, options)
```

Графический обьект Эллипс.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		IPixelCircleGeometry	✔	Геометрия.
data		Object	✔	Данные.
options		Object	✔	Опции.
	fill	Boolean		Наличие заливки graphics.style.color.
	fillColor	String		Цвет заливки.
	fillImageHref	String		Фоновое изображение. При включении данной опции значение fillColor игнорируется.
	fillMethod	String		Тип заливки фоном. Может принимать одно из двух значений: • 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. Значение по умолчанию: 'stretch'

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	fillOpacity	Number		Прозрачность заливки.
	opacity	Number		Общая прозрачность.
	outline	Boolean		Наличие обводки.
	separateContainer	Boolean		Производит отрисовку в отдельном слое.
	strokeColor	String		Цвет линии graphics.style.color.
	strokeOpacity	Number		Прозрачность контура.
	strokeStyle	Number[]|String		Стиль контура(не поддерживается в режиме Canvas) graphics.style.stroke.
	strokeWidth	Number		Ширина линии.

* Обязательный параметр/опция.

overlay.staticGraphics.Placemark

Входит в пакеты package.full, package.overlays.

Расширяет overlay.staticGraphics.Base.

Создает изображение.

Конструктор

Конструктор

```
overlay.staticGraphics.Placemark(geometry, data, options)
```

Графический обьект Изображение.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		IPixelPointGeometry	✔	Геометрия.
data		Object	✔	Данные.
options		Object	✔	Опции.
	imageClipRect	Number[][]		Область обрезки исходного изображения. Может быть использована для реализации поддержки спрайтов. Значение по умолчанию: false
	imageHref	String	✔	Адрес изображения.
	imageOffset	String		Сдвиг маркера относительно левого верхнего угла.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	imageSize	String		Размер изображения.
	separateContainer	Boolean		Производит отрисовку в отдельном слое.

* Обязательный параметр/опция.

overlay.staticGraphics.Polygon

Входит в пакеты package.full, package.overlays.

Расширяет overlay.staticGraphics.Base.

Полигон.

Конструктор

Конструктор

```
overlay.staticGraphics.Polygon(geometry, data, options)
```

Графический обьект "Полигон".

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		**IPixelPolygonGeometry**	✔	Геометрия.
data		Object	✔	Данные.
options		Object	✔	Опции.
	fill	String		Наличие заливки.
	fillColor	String		Цвет заливки **graphics.style.color**.
	fillImageHref	String		Фоновое изображение. При включении данной опции значение fillColor игнорируется.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	fillMethod	String		Тип заливки фоном. Может принимать одно из двух значений: • 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. Значение по умолчанию: 'stretch'
	fillOpacity	Number		Прозрачность заливки.
	opacity	Number		Общая прозрачность.
	outline	String		Наличие обводки.
	separateContainer	Boolean		Производит отрисовку в отдельном слое.
	strokeColor	String		Цвет линии **graphics.style.color**.
	strokeOpacity	Number		Прозрачность контура.
	strokeStyle	Number[]|String		Стиль контура **graphics.style.stroke**.
	strokeWidth	Number		Ширина линии.

* Обязательный параметр/опция.

overlay.staticGraphics.Polyline

Входит в пакеты package.full, package.overlays.

Расширяет overlay.staticGraphics.Base.

Линия.

Конструктор

Конструктор

```
overlay.staticGraphics.Polyline(geometry, data, options)
```

Графический обьект Линия.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		IPixelLineStringGeometry	✔	Геометрия.
data		Object	✔	Данные.
options		Object	✔	Опции.
	opacity	Number		Общая прозрачность.
	separateContainer	Boolean		Производит отрисовку в отдельном слое.
	strokeColor	String		Цвет линии graphics.style.color.
	strokeOpacity	Number		Прозрачность контура.
	strokeStyle	Number[]|String		Стиль контура graphics.style.stroke.
	strokeWidth	Number		Ширина линии.

* Обязательный параметр/опция.

overlay.staticGraphics.Rectangle

Входит в пакеты package.full, package.overlays.

Расширяет overlay.staticGraphics.Base.

Прямоугольник.

Конструктор

Конструктор

```
overlay.staticGraphics.Rectangle(geometry, data, options)
```

Графический обьект Прямоугольник.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
geometry		IPixelRectangleGeometry	✔	Геометрия.
data		Object	✔	Данные.
options		Object	✔	Опции.
	borderRadius	Number		Радиус скругления углов.
	fill	String		Наличие заливки.
	fillColor	String		Цвет заливки graphics.style.color.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	fillImageHref	String		Фоновое изображение. При включении данной опции значение fillColor игнорируется.
	fillMethod	String		Тип заливки фоном. Может принимать одно из двух значений: • 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. Значение по умолчанию: 'stretch'
	fillOpacity	Number		Прозрачность заливки.
	opacity	Number		Общая прозрачность.
	outline	String		Наличие обводки.
	separateContainer	Boolean		Производит отрисовку в отдельном слое.
	strokeColor	String		Цвет линии **graphics.style.color**.
	strokeOpacity	Number		Прозрачность контура.
	strokeStyle	Number[]|String		Стиль обводки (не поддерживается в режиме Canvas) **graphics.style.stroke**.
	strokeWidth	Number		Ширина линии.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Placemark

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geocode, package.search, package.standard.

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.

См.: GeoObject geometry.Point

Параметры:

Параметр	Тип	*	Описание
geometry	Number[]|Object|IPointGeometry	✔	Координаты метки или хэш с описанием геометрии или ссылка на объект точечной геометрии.
properties	Object|**IDataManager**		Данные метки. Соответствуют данным класса **GeoObject**.
options	Object		Опции метки. Соответствуют опциям класса **GeoObject**.

* Обязательный параметр/опция.

Пример:

```
// Создаем метку.
var placemark = new ymaps.Placemark([48, 40], {
    balloonContent: '<img src="http://img-fotki.yandex.ru/get/
6114/82599242.2d6/0_88b97_ec425cf5_M" />',
    iconContent: "Азербайджан"
}, {
    preset: "twirl#yellowStretchyIcon",
    // Отключаем кнопку закрытия балуна.
    balloonCloseButton: false,
    // Балун будем открывать и закрывать кликом по иконке метки.
    hideIconOnBalloonOpen: false
});
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".

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
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: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что 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**. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса **Balloon**. Унаследовано от **GeoObject**.
beforedrag	Событие, предшествующее событию drag. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 drag будет отменено. Унаследовано от **GeoObject**.
beforedragstart	Событие, предшествующее событию dragstart. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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. Унаследовано от **GeoObject**.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
drag	Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге. Унаследовано от **GeoObject**.
dragend	Завершение перетаскивания геообъекта. Экземпляр класса **Event**. Унаследовано от **GeoObject**.
dragstart	Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • domEvent — исходное DOM-событие (в виде объекта **DomEvent**), если оно имеется. Унаследовано от **GeoObject**.
editorstatechange	Изменение состояния редактора геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие редактора геометрии. Унаследовано от **GeoObject**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от IDomEventEmitter.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от IDomEventEmitter.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от IDomEventEmitter.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от IGeoObject.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от IGeoObject.
propertieschange	Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от IGeoObject.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **IParentOnMap**.
getOverlay()	**IOverlay**|null	Возвращает оверлей геообъекта, либо null. Унаследован от **IGeoObject**.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

Описание полей

editor

```
geometryEditor.Point editor
```

Редактор геометрии "Точка".

geometry

```
geometry.Point geometry
```

Геометрия типа "Точка".

Polygon

Входит в пакеты package.editor, package.full, package.geoObjects.

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.

См.: GeoObject geometry.Polygon

Параметры:

Параметр	Тип	*	Описание
geometry	Number[][][]|Object|IPolygonGeometry	✔	Координаты вершин ломаных, определяющих внешнюю и внутренние границы многоугольника, хэш-объект с параметрами геометрии или ссылка на объект геометрии. Внутренняя граница может отсутствовать.
properties	Object|**IDataManager**		Данные геообъекта. Соответствуют данным класса **GeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
options	Object		Опции геообъекта. Соответствуют опциям класса **GeoObject**.

* 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**.
state	**data.Manager**	Состояние геообъекта. Определяется следующими полями: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что геообъект перетаскивается. Унаследовано от **GeoObject**.

События

Имя	Описание
balloonclose	Закрытие балуна. Экземпляр класса **Event**. Унаследовано от **GeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
balloonopen	Открытие балуна на геообъекте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса **Balloon**. Унаследовано от **GeoObject**.
beforedrag	Событие, предшествующее событию drag. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 drag будет отменено. Унаследовано от **GeoObject**.
beforedragstart	Событие, предшествующее событию dragstart. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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. Унаследовано от **GeoObject**.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
drag	Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге. Унаследовано от **GeoObject**.
dragend	Завершение перетаскивания геообъекта. Экземпляр класса **Event**. Унаследовано от **GeoObject**.
dragstart	Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • domEvent — исходное DOM-событие (в виде объекта **DomEvent**), если оно имеется. Унаследовано от **GeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
editorstatechange	Изменение состояния редактора геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие редактора геометрии. Унаследовано от **GeoObject**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от IGeoObject.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от IGeoObject.
propertieschange	Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от IGeoObject.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getOverlay()	IOverlay|null	Возвращает оверлей геообъекта, либо null. Унаследован от IGeoObject.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание полей

editor

```
geometryEditor.Polygon editor
```

Редактор геометрии "Многоугольник".

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

geometry

```
geometry.Polygon geometry
```

Геометрия типа "Многоугольник".

Polyline

Входит в пакеты package.editor, package.full, package.geoObjects.

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.

См.: GeoObject geometry.LineString

Параметры:

Параметр	Тип	*	Описание
geometry	Number[][]|Object|ILineStringGeometry	✔	Координаты вершин, хэш-объект с параметрами геометрии или ссылка на объект геометрии ломаной линии.
properties	Object|**IDataManager**		Данные геообъекта. Соответствуют данным класса **GeoObject**.
options	Object		Опции геообъекта. Соответствуют опциям класса **GeoObject**.

* Обязательный параметр/опция.

Пример:

```
// Создаем ломаную линию.
var polyline = new ymaps.Polyline([
    [-80, 60], [-90, 50], [-60, 40], [-80, 60]
], {
    hintContent: "Ломаная линия"
}, {
    draggable: true,
    strokeColor: '#000000',
    strokeWidth: 4,
    // Первой цифрой задаем длину штриха. Второй цифрой задаем длину разрыва.
    strokeStyle: '1 5'
});
// Добавляем линию на карту.
myMap.geoObjects.add(polyline);
// Устанавливаем карте границы линии.
myMap.setBounds(polyline.geometry.getBounds());
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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.
state	data.Manager	Состояние геообъекта. Определяется следующими полями: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что 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. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса Balloon. Унаследовано от GeoObject.
beforedrag	Событие, предшествующее событию drag. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 drag будет отменено. Унаследовано от GeoObject.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
beforedragstart	Событие, предшествующее событию dragstart. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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. Унаследовано от **GeoObject**.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
drag	Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге. Унаследовано от **GeoObject**.
dragend	Завершение перетаскивания геообъекта. Экземпляр класса **Event**. Унаследовано от **GeoObject**.
dragstart	Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • domEvent — исходное DOM-событие (в виде объекта **DomEvent**), если оно имеется. Унаследовано от **GeoObject**.
editorstatechange	Изменение состояния редактора геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие редактора геометрии. Унаследовано от **GeoObject**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от **IGeoObject**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от **IGeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
propertieschange	Изменение данных геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от **IGeoObject**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getOverlay()	**IOverlay**|null	Возвращает оверлей геообъекта, либо null. Унаследован от **IGeoObject**.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

Описание полей

editor

```
geometryEditor.LineString editor
```

Редактор геометрии "Ломаная линия".

geometry

```
geometry.LineString geometry
```

Геометрия типа "Ломаная линия".

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

projection

projection.Cartesian

Входит в пакеты package.full, package.map, package.standard.

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^zoom.

Параметры:

Параметр	Тип	*	Описание
bounds	Object[]	✔	Массив из двух точек — координат левого нижнего и правого верхнего углов прямоугольной координатной области.
cycled	Boolean[]		Массив из признаков зацикленности карты по x и y. Значение по умолчанию: [false, false]
scale	Number|Number[]		Масштаб одного деления на оси. Может быть числом или парой чисел для каждой из осей. Значение по умолчанию: 1

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
fromGlobalPixels(globalPixelPoint, zoom)	Number[]	Преобразует пиксельные координаты на указанном уровне масштабирования в координаты проекции. Унаследован от **IProjection**.
getCoordSystem()	**ICoordSystem**	Возвращает координатную систему, используемую проекцией. Необязательный метод. В случае его отсутствия будет использована координатная система **coordSystem.cartesian**. Унаследован от **IProjection**.
isCycled()	Boolean[]	Показатель зацикленности проекции. Унаследован от **IProjection**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
toGlobalPixels(coordPoint, zoom)	Number[]	Преобразует координаты проекции в глобальные пиксельные на указанном уровне масштабирования. Унаследован от **IProjection**.

projection.sphericalMercator

Входит в пакеты package.full, package.map, package.standard.

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. Необязательный метод. В случае его отсутствия будет использована координатная система **coordSystem.cartesian**.
isCycled()	Boolean[]	Показатель зацикленности проекции.
toGlobalPixels(coordPoint, zoom)	Number[]	Преобразует координаты проекции в глобальные пиксельные на указанном уровне масштабирования.

projection.wgs84Mercator

Входит в пакеты package.full, package.map, package.standard.

Статический объект.

Экземпляр класса **IProjection**

Меркаторовская проекция на референсный эллипсоид wgs84. Используется Яндекс.Картами по умолчанию.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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. Необязательный метод. В случае его отсутствия будет использована координатная система **coordSystem.cartesian**.
isCycled()	Boolean[]	Показатель зацикленности проекции.
toGlobalPixels(coordPoint, zoom)	Number[]	Преобразует координаты проекции в глобальные пиксельные на указанном уровне масштабирования.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

ready

Статическая функция.

Выполняет переданную в параметре функцию, когда API готов к использованию.

Параметры:

Параметр	Тип	*	Описание
callback	Function	✔	Функция, которая будет вызвана при успешной загрузке API.
context	Object		Контекст выполнения.

* Обязательный параметр/опция.

Пример:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
       <script src="http://api-maps.yandex.ru/2.x/?load=package.full&mode=debug"
type="text/javascript"></script>
        <script type="text/javascript">
            ymaps.ready(function(){
                var myMap = new ymaps.Map("map", {
                        center:[55.76, 37.64], // Москва
                        zoom:2
                    });
             });
        </script>
    </head>

    <body>
        <div id="map" style="width: 800px; height: 300px"></div>
    </body>
</html>
```

Rectangle

Входит в пакеты package.editor, package.full, package.geoObjects.

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.

См.: GeoObject geometry.Rectangle

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
geometry	Number[][]|Object|IRectangleGeometry	✔	Координаты двух противоположных углов, хэш-объект с параметрами геометрии или ссылка на объект геометрии прямоугольника.
properties	Object|**IDataManager**		Данные геообъекта. Соответствуют данным класса **GeoObject**.
options	Object		Опции геообъекта. Соответствуют опциям класса **GeoObject**.

* Обязательный параметр/опция.

Пример:

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

// Создаем прямоугольник на основе границы круга.
var rectangle = new ymaps.Rectangle(circle.geometry.getBounds(), {}, {
    fill: false,
    coordRendering: "boundsPath",
    strokeWidth: 4
});
// Добавляем прямоугольник на карту на карту.
myMap.geoObjects.add(rectangle);

// При изменении геометрии круга обновляем координаты прямоугольника.
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**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
state	data.Manager	Состояние геообъекта. Определяется следующими полями: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что 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. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса Balloon. Унаследовано от GeoObject.
beforedrag	Событие, предшествующее событию drag. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 drag будет отменено. Унаследовано от GeoObject.
beforedragstart	Событие, предшествующее событию dragstart. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • 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. Унаследовано от GeoObject.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
drag	Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге. Унаследовано от **GeoObject**.
dragend	Завершение перетаскивания геообъекта. Экземпляр класса **Event**. Унаследовано от **GeoObject**.
dragstart	Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • domEvent — исходное DOM-событие (в виде объекта **DomEvent**), если оно имеется. Унаследовано от **GeoObject**.
editorstatechange	Изменение состояния редактора геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие редактора геометрии. Унаследовано от **GeoObject**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от **IGeoObject**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от **IGeoObject**.
propertieschange	Изменение данных геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от **IGeoObject**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getOverlay()	**IOverlay**|null	Возвращает оверлей геообъекта, либо null. Унаследован от **IGeoObject**.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание полей

editor

```
Null editor
```

Редактор геометрии "Прямоугольник" на данный момент не реализован.

geometry

```
geometry.Rectangle geometry
```

Геометрия типа "Прямоугольник".

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

regions

regions.load

Входит в пакеты package.full, package.regions.

Статическая функция.

Предоставляет доступ к геометрии различных регинов.

Возвращает: (тип imports.util.Promise).

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
region		String	✔	ISO_3166-1 код страны (RU, UA, BY, KZ).
options		Object	✔	Опции отображения.
	lang	String		Язык (ru, uk, en, be).
	quality	Number		Уровень качества. Доступные значения: • 0 — минимальное качество; • 1 — стандартное качество; • 2 — повышенное качество; • 3 — высокое качество. Уровень качества влияет на точность представления кривых и обьем файла данных. Значение по умолчанию: 1

* Обязательный параметр/опция.

Пример:

```
ymaps.regions.load('RU', {
    lang: 'en'
}).then(function (result) {
    geoMap.geoObjects.add(result.geoObjects);
});
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

route

Входит в пакеты package.full, package.route.

Статическая функция.

Прокладывает маршрут через заданные точки.

Возвращает: Объект-promise, который подтверждается при успешном построении маршрута и получает в качестве параметра экземпляр класса **router.Route**, соответствующий построенному маршруту. (тип **util.Promise**).

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
points		Object[]	✔	Набор точек, через которые должен проходить маршрут. В массиве могут быть заданы строки — "Москва", координаты точки — [30, 50] и объекты, описывающие точку, — { type: 'viaPoint', point: 'Москва' } или { type: 'wayPoint', point: [30, 50] }. Через тип 'viaPoint' обозначается транзитная точка, т.е. точка, через которую нужно проехать без остановки.
options		Object		Опции построения маршрута.
	avoidTrafficJams	Boolean		true — строить маршрут с учетом пробок, false — без учета пробок. При использовании опции учитывайте, что объезд пробок не всегда возможен. Значение по умолчанию: false
	boundedBy	Number[][]		Область на карте, где предположительно находятся искомые объекты. Используется, если точки маршрута заданы почтовым адресом, а не координатами.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	mapStateAutoApply	Boolean		Флаг, позволяющий автоматически установить центр и коэффициент масштабирования карты так, чтобы построенный маршрут был виден целиком. Значение по умолчанию: false
	searchCoordOrder	String		Определяет каким образом нужно интрепретировать координаты в запросе. Используется, если точки маршрута заданы почтовым адресом, а не координатами.
	strictBounds	Boolean		Искать только внутри области, заданной опцией boundedBy. Значение по умолчанию: false

* Обязательный параметр/опция.

Пример:

```
// Строим маршрут из Королева в Красногорск через Химки и Мытищи, где Мытищи -
транзитная точка.
// Красногорск зададим координатами.
ymaps.route([
     'Королев',
     { type: 'viaPoint', point: 'Мытищи' },
     'Химки',
     { type: 'wayPoint', point: [55.811511, 37.312518] }
 ], {
     mapStateAutoApply: true
 }).then(function (route) {
     route.getPaths().options.set({
         // в балуне выводим только информацию о времени движения с учетом
пробок
         balloonContenBodyLayout: ymaps.templateLayoutFactory.createClass('$
[properties.humanJamsTime]'),
         // можно выставить настройки графики маршруту
         strokeColor: '0000ffff',
         opacity: 0.9
     });
     // добавляем маршрут на карту
     map.geoObjects.add(route);
 });
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

router

router.Editor

Редактор маршрутов.

Конструктор | События

Конструктор

```
router.Editor(route, options)
```

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
route		router.Route	✔	Маршрут, к которому должен быть привязан редактор.
options		Object		Опции.
	addViaPoints	Boolean		true — добавление транзитных точек разрешено, false — запрещено. Значение по умолчанию: true
	addWayPoints	Boolean		true — добавление путевых точек при клике на карте разрешено, false — запрещено. Значение по умолчанию: false
	editViaPoints	Boolean		true — редактирование (перемещение) транзитных точек разрешено, false — запрещено. Значение по умолчанию: true
	editWayPoints	Boolean		true — редактирование (перемещение) путевых точек разрешено, false — запрещено. Значение по умолчанию: true
	removeViaPoints	Boolean		true — удаление транзитных точек двойным кликом разрешено, false — запрещено. Значение по умолчанию: true

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	removeWayPoints	Boolean		true — удаление путевых точек двойным кликом разрешено, false — запрещено. Значение по умолчанию: false

* Обязательный параметр/опция.

События

Имя	Описание
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 событий

routeupdate

```
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

```
start
```

Включение редактора.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

stop

```
stop
```

Выключение редактора.

viapointadd

```
viapointadd
```

Добавление транзитной точки. Добавляемую транзитную точку можно получить через e.get('viaPoint')

viapointdragend

```
viapointdragend
```

Завершение перетаскивания транзитной точки. Путевую точку можно получить через e.get('viaPoint')

viapointdragstart

```
viapointdragstart
```

Начало перетаскивания транзитной точки. Путевую точку можно получить через e.get('viaPoint')

viapointremove

```
viapointremove
```

Удаление транзитной точки. Удаляемую транзитную точку можно получить через e.get('viaPoint')

waypointadd

```
waypointadd
```

Добавление путевой точки. Добавляемую путевую точку можно получить через e.get('wayPoint')

waypointdragend

```
waypointdragend
```

Завершение перетаскивания путевой точки. Путевую точку можно получить через e.get('wayPoint')

waypointdragstart

```
waypointdragstart
```

Начало перетаскивания путевой точки. Путевую точку можно получить через e.get('wayPoint')

waypointremove

```
waypointremove
```

Удаление путевой точки. Удаляемую путевую точку можно получить через e.get('wayPoint')

router.Path

Расширяет **GeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Объект, описывающий часть (путь) машрута. Маршрут может содержать несколько путей, каждый путь соединяет две путевые точки. Для путей доступны следующие пресеты:

- 'router#interactivePath' — обеспечивает вывод информации о сегменте пути при клике по полилинии пути. При использовании этого пресета в макете балуна становятся доступными, как свойства геообъекта, следующие данные сегмента **router.Segment**: action, humanAction, angle, lenght, humanLength, time, humanTime, jamsTime, humanJamsTime, street. Используется по умолчанию.
- 'router#plainPath' — путь ведет себя как обычный геообъект.

См.: route

Конструктор | Поля | События | Методы

Конструктор

```
router.Path(feature, options)
```

См.: route

Параметры:

Параметр	Тип	*	Описание
feature	Object	✔	Геометрия и свойства пути.
options	Object		Опции пути.

* Обязательный параметр/опция.

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**	Геометрия геообъекта. Унаследовано от **IGeoObject**.
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: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что геообъект перетаскивается. Унаследовано от **GeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
balloonclose	Закрытие балуна. Экземпляр класса **Event**. Унаследовано от **GeoObject**.
balloonopen	Открытие балуна на геообъекте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса **Balloon**. Унаследовано от **GeoObject**.
beforedrag	Событие, предшествующее событию drag. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 drag будет отменено. Унаследовано от **GeoObject**.
beforedragstart	Событие, предшествующее событию dragstart. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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. Унаследовано от **GeoObject**.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
drag	Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге. Унаследовано от **GeoObject**.
dragend	Завершение перетаскивания геообъекта. Экземпляр класса **Event**. Унаследовано от **GeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
dragstart	Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • domEvent — исходное DOM-событие (в виде объекта **DomEvent**), если оно имеется. Унаследовано от **GeoObject**.
editorstatechange	Изменение состояния редактора геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие редактора геометрии. Унаследовано от **GeoObject**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от IDomEventEmitter.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от IGeoObject.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от IGeoObject.
propertieschange	Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от IGeoObject.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getOverlay()	IOverlay|null	Возвращает оверлей геообъекта, либо null. Унаследован от IGeoObject.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
getSegments()	router.Segment[]	Возвращает сегменты пути.
getTime()	Integer	Возвращает время проезда пути в секундах.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. 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()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает время проезда пути в секундах.

router.Route

Расширяет **IGeoObject**, **IParentOnMap**.

Объект, описывающий проложенный маршрут.

См.: route

Конструктор | Поля | События | Методы

Конструктор

```
router.Route(json, options)
```

См.: route

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
json		Object	✔	Описание маршрута в формате json.
options		Object		Опции маршрута.
	mapStateAutoApply	Boolean		Флаг, позволяющий автоматически установить центр и коэффициент масштабирования карты так, чтобы построенный маршрут был виден целиком. Значение по умолчанию: false

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
editor	**router.Editor**	Редактор маршрутов.
events	**IEventManager**	Менеджер событий. Унаследовано от **IDomEventEmitter**.
geometry	**IGeometry**	Геометрия геообъекта. Унаследовано от **IGeoObject**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
properties	**IDataManager**	Менеджер данных геообъекта. Унаследовано от **IGeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
state	**IDataManager**	Состояние геообъекта. Определяется следующими полями: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что геообъект перетаскивается. Унаследовано от **IGeoObject**.

События

Имя	Описание
boundsapply	Событие выставления границ маршрута карте при заданной опции options.mapStateAutoApply.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от **IGeoObject**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от **IGeoObject**.
propertieschange	Изменение данных геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от **IGeoObject**.
update	Событие обновления маршрута при включенном редакторе маршрутов.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
getHumanJamsTime()	String	Возвращает строковое представление времени проезда маршрута с единицами измерения с учетом пробок.
getHumanLength()	String	Возвращает строковое представление длины маршрута с единицами измерения.
getHumanTime()	String	Возвращает строковое представление времени проезда маршрута с единицами измерения.
getJamsTime()	Integer	Возвращает время проезда маршрута в секундах с учетом пробок.
getLength()	Number	Возвращает длину маршрута в метрах.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getOverlay()	IOverlay|null	Возвращает оверлей геообъекта, либо null. Унаследован от IGeoObject.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
getPaths()	GeoObjectArray	Возвращает коллекцию путей, из которых состоит маршрут.
getTime()	Integer	Возвращает время проезда маршрута в секундах.
getViaPoints()	GeoObjectArray	Возвращает коллекцию транзитных точек маршрута.
getWayPoints()	GeoObjectArray	Возвращает коллекцию путевых точек маршрута.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание полей

editor

```
router.Editor editor
```

Редактор маршрутов.

Пример:

```
// Начало редактирования маршрута.
ymaps.route(['Москва', 'Питер'], function (route) {
    route.editor.start();
    // ...
    // Завершение редактирования маршрута.
    route.editor.stop();
});
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание событий

boundsapply

```
boundsapply
```

Событие выставления границ маршрута карте при заданной опции options.mapStateAutoApply.

update

```
update
```

Событие обновления маршрута при включенном редакторе маршрутов.

Описание методов

getHumanJamsTime

```
{String} getHumanJamsTime()
```

Возвращает строковое представление времени проезда маршрута с единицами измерения с учетом пробок.

getHumanLength

```
{String} getHumanLength()
```

Возвращает строковое представление длины маршрута с единицами измерения.

getHumanTime

```
{String} getHumanTime()
```

Возвращает строковое представление времени проезда маршрута с единицами измерения.

getJamsTime

```
{Integer} getJamsTime()
```

Возвращает время проезда маршрута в секундах с учетом пробок.

getLength

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

Возвращает длину маршрута в метрах.

getPaths

```
{GeoObjectArray} getPaths()
```

Возвращает коллекцию путей, из которых состоит маршрут.

getTime

```
{Integer} getTime()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает время проезда маршрута в секундах.

getViaPoints

```
{GeoObjectArray} getViaPoints()
```

Возвращает коллекцию транзитных точек маршрута.

getWayPoints

```
{GeoObjectArray} getWayPoints()
```

Возвращает коллекцию путевых точек маршрута.

router.Segment

Объект, описывающий сегмент маршрута. Сегмент — часть маршрута от одного манёвра до другого.

Конструктор | Методы

Конструктор

```
router.Segment(feature)
```

Параметры:

Параметр	Тип	*	Описание
feature	Object	✔	Свойства и геометрия сегмента.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
getAction()	String	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	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.
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 сегмента в метрах.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
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()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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 сегмента 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getTime

```
{Integer} getTime()
```

Возвращает время проезда сегмента в секундах.

router.ViaPoint

Расширяет **GeoObject**.

Объект, описывающий транзитную точку маршрута.

Конструктор | Поля | События | Методы

Конструктор

```
router.ViaPoint(feature, options)
```

Параметры:

Параметр	Тип	*	Описание
feature	Object	✔	Свойства и геометрия.
options	Object		Опции.

* Обязательный параметр/опция.

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**	Геометрия геообъекта. Унаследовано от **IGeoObject**.
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: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что геообъект перетаскивается.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
		Унаследовано от **GeoObject**.

События

Имя	Описание
balloonclose	Закрытие балуна. Экземпляр класса **Event**. Унаследовано от **GeoObject**.
balloonopen	Открытие балуна на геообъекте. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • balloon — экземпляр класса **Balloon**. Унаследовано от **GeoObject**.
beforedrag	Событие, предшествующее событию drag. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг 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 drag будет отменено. Унаследовано от **GeoObject**.
beforedragstart	Событие, предшествующее событию dragstart. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • 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. Унаследовано от **GeoObject**.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
drag	Перетаскивание геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pixelOffset — массив из двух чисел, описывающий пиксельный сдвиг на данном шаге. Унаследовано от **GeoObject**.
dragend	Завершение перетаскивания геообъекта. Экземпляр класса **Event**. Унаследовано от **GeoObject**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
dragstart	Начало перетаскивания геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • domEvent — исходное DOM-событие (в виде объекта **DomEvent**), если оно имеется. Унаследовано от **GeoObject**.
editorstatechange	Изменение состояния редактора геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие редактора геометрии. Унаследовано от **GeoObject**.
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии. Унаследовано от **IGeoObject**.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от IDomEventEmitter.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет. Унаследовано от IGeoObject.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия. Унаследовано от IGeoObject.
propertieschange	Изменение данных геообъекта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных. Унаследовано от IGeoObject.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
getOverlay()	IOverlay|null	Возвращает оверлей геообъекта, либо null. Унаследован от IGeoObject.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. 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. Унаследован от IChildOnMap.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

getPathIndex

```
{Integer} getPathIndex()
```

Возвращает индекс пути, в котором расположена точка.

getSegmentIndex

```
{Integer} getSegmentIndex()
```

Возвращает индекс сегмента пути, в котором расположена точка.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

templateLayoutFactory

Входит в пакеты package.clusters, package.editor, package.full, package.geoObjects, package.geoXml, package.geocode, package.route, package.search, package.standard.

Статический объект.

Фабрика для создания макетов из шаблонов. Позволяет создавать классы, реализующие интерфейс **ILayout**, используя язык шаблонов. Шаблон представляет собой фрагмент html-кода со следующими подстановками:

- $[имя_поля|значение_по_умолчанию] — подставляет указанное поле данных либо значение по умолчанию, если такого поля нет;
- $[[имя_поля]] — для отображения вложенных макетов; шаблонизатор, встретив такую конструкцию, попытается использовать значение в поле как ключ или класс вложенного макета;
- [if имя_поля]шаблон1[else]шаблон2[endif] — в зависимости от значения поля будет подставлен или шаблон1, или шаблон2; секция [else] может отсутствовать;
- [ifdef имя_поля]шаблон1[else]шаблон2[endif] — то же, что и конструкция if, но условие считается истинным при любом значении поля, кроме undefined.

См.: layout.templateBased.Base

Методы

Пример:

```
// В примерах используется jQuery, загруженный с http://yandex.st/jquery/1.6.4/
jquery.min.js

// Пример 1. Применение макетов к геообъекту.
// Создание макета контента балуна. Здесь обращаемся к опции балуна
contentBodyLayout, заданной в геообъекте
// с префиксом balloon. В данном случае options.contentBodyLayout - вложенный
макет.
var myBalloonContentLayout = ymaps.templateLayoutFactory.createClass('<p>$
[[options.contentBodyLayout]]</p>');
// Создание макета основного содержимого контента балуна.
var counter = 0,
    myBalloonContentBodyLayout = ymaps.templateLayoutFactory.createClass(
    '<b>$[properties.name]</b><br /><button id="counter-button">Counter</button>
<i id="count"></i>', {
        build: function () {
            myBalloonContentBodyLayout.superclass.build.call(this);
            $('#counter-button').bind('click', this.onCounterClick);
            $('#count').html(counter);
        },
        clear: function () {
            $('#counter-button').unbind('click', this.onCounterClick);
            myBalloonContentBodyLayout.superclass.clear.call(this);
        },
        onCounterClick: function () {
            $('#count').html(++counter);
        }
    });

// Создание макета для контента метки.
var myIconContentLayout = ymaps.templateLayoutFactory.createClass(
    '<h3 id="layout-element">$[properties.name]</h3>' +
    '[if properties.description]<i>$[properties.description]</i>[else]$
[properties.metaDataProperty.description][endif]', {
        build: function () {
            // необходим вызов родительского метода, чтобы добавить содержимое
макета в DOM
            this.constructor.superclass.build.call(this);
            $('#layout-element').bind('mouseover', this.onNameHover);
        },
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
        clear: function () {
            $('#layout-element').unbind('mouseover', this.onNameHover);
            this.constructor.superclass.clear.call(this);
        },

        onNameHover: function () {
            $('#layout-element').css('color', getRandomColor());
        }
    });

var getRandomColor = function () {
        return 'rgb(' +
            [Math.floor(Math.random() * 256), Math.floor(Math.random() * 256),
Math.floor(Math.random() * 256) ].join(',') +
        ')';
    };

var myGeoObject = new ymaps.GeoObject({
         geometry: {
             type: "Point",
             coordinates: [55.650625, 37.62708]
         },
         properties: {
             name: 'my name',
             metaDataProperty: {
                 description: 'metaData description'
             }
         }
     }, {
         balloonContentBodyLayout: myBalloonContentBodyLayout,
         balloonContentLayout: myBalloonContentLayout,
         iconContentLayout: myIconContentLayout,
         // Выставляем тянущийся макет иконки, чтобы вместился вложенный макет
myIconContentLayout.
         preset: 'twirl#nightStretchyIcon'
     });

map.geoObjects.add(myGeoObject);
// Установим через 5 секунд свойство в геообъекте. Макет контента иконки отследит
это обновление и обновит содержимое.
setTimeout(function () {
    myGeoObject.properties.set('description', 'my description after timeout');
}, 5000);

// Пример 2. Открытие балуна на карте с заданными макетами.
map.balloon.open(map.getCenter(), {
    myBodyContent: '<b>body content</b>',
    myFooterContent: 'footer content'
}, {
    contentBodyLayout: ymaps.templateLayoutFactory.createClass('<p>$
[myBodyContent]</p>'),
    contentFooterLayout: ymaps.templateLayoutFactory.createClass('<i>$
[myFooterContent]</i>')
});
```

Методы

Имя	Статический	Возвращает	Описание
templateLayoutFactory.**createClass**(template, overrides)	✔	Function	Возвращает конструктор макета. Созданный класс будет унаследован от класса **layout.templateBased.Base** с переопределением списка методов, указанных в overrides.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

createClass

```
{Function}  <static> templateLayoutFactory.createClass(template, overrides)
```

Возвращает конструктор макета. Созданный класс будет унаследован от класса **layout.templateBased.Base** с переопределением списка методов, указанных в overrides.

Параметры:

Параметр	Тип	*	Описание
template	String	✔	Шаблон html-содержимого макетов.
overrides	Object		Переопределения родительских методов. Можно переопределять/ дополнять методы build, clear и rebuild.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

theme

theme.twirl

theme.twirl.balloon

theme.twirl.balloon.Layout

Расширяет **IBalloonLayout**.

Класс для создания стандартного балуна.

Конструктор | Поля | События | Методы

Конструктор

```
theme.twirl.balloon.Layout()
```

Создает экземпляр стандартного макета балуна. Доступен под ключом twirl#balloon.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
	options.contentLayout	**ILayout**	✔	HTML содержимое макета.
	options.closeButton	Boolean		Флаг наличия кнопки закрытия.
	options.closeButtonLayout	**ILayout**		Макет кнопки закрытия
	options.maxHeight	String		Максимально допустимая высота.
	options.maxWidth	String		Максимально допустимая ширина.

* 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 ввиду, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от ILayout.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от IDomEventEmitter.
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. Унаследовано от IDomEventEmitter.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от **ILayout**.
userclose	Нажата кнопка закрытия. Унаследовано от **IBalloonLayout**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от **ILayout**.
getData()	Object	Возвращает объект данных макета. Унаследован от **ILayout**.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от **ILayout**.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от **ILayout**.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание событий

clientboundschange

```
clientboundschange
```

Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента;
- newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

theme.twirl.geoObject

theme.twirl.geoObject.layout

theme.twirl.geoObject.layout.StretchyIcon

Расширяет **ILayout**.

Класс для создания макета для метки с содержимым.

Конструктор | Поля | События | Методы

Конструктор

```
theme.twirl.geoObject.layout.StretchyIcon()
```

Создает экземпляр стандартного макета который растягивается под свое содержимое. Доступен под ключом twirl#stretchyIcon.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
	options.contentLayout	**ILayout**	✔	Макет содержимого.
	options.maxHeight	String		Максимально допустимая высота.
	options.maxWidth	String		Максимально допустимая ширина.

* 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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от **ILayout**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **ILayout**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от **ILayout**.
getData()	Object	Возвращает объект данных макета. Унаследован от **ILayout**.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от **ILayout**.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от **ILayout**.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

traffic

traffic.provider

traffic.provider.Actual

Входит в пакеты package.full, package.traffic.

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	Свойства параметра	Тип	Описание
options		Object	Опции провайдера. Опции для слоев провайдера задаются через опции глобальной коллекции слоев **Map.layers**. • Опции для картиночного слоя **Layer** задаются с префиксом 'trafficImage'; • Опции для слоя хотспотов **hotspot.Layer** задаются с префиксом 'trafficJam'; • Опции для слоя инфоточек задаются с префиксом 'trafficInfo'. Слой инфоточек является экземпляром класса hotspot.Layer.
	autoUpdate	Boolean	Флаг, включающий автоматическое обновление данных о пробках. Автоматическое обновление происходит только в том случае, если на карте каждые 4 минуты возникает событие mousemove. Если событие не возникает, пробки перестают обновляться до появления нового события. Значение по умолчанию: true

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	Описание
state		Object	Состояние провайдера.
	infoLayerShown	Boolean	Флаг, включающий отображение слоя дорожных событий. Значение по умолчанию: false

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 слоев пробок и инфоточек.
myMap.layers.options.set({
    // Название опции сформировано из опции хотспотного слоя
    // 'openBalloonOnClick' путем добавления префикса 'trafficJam'.
    trafficJamOpenBalloonOnClick: false,
    // Аналогично формируется название опции для слоя инфоточек.
    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 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".

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
getMap()	Map|null	Возвращает ссылку на карту. Унаследован от ITrafficProvider.
setMap(Ссылка)		
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

Входит в пакеты package.full, package.traffic.

Расширяет ITrafficProvider.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Свойства параметра	Тип	Описание
options		Object	Опции провайдера. Опции для слоев провайдера задаются через опции глобальной коллекции слоев **Map.layers**. • Опции для картиночного слоя **Layer** задаются с префиксом 'trafficImage'; • Опции для слоя хотспотов **hotspot.Layer** задаются с префиксом 'trafficJam'.
	showCurrentTimeFirst	Boolean	При первом показе архивных данных выставить время, близкое к текущему. Значение по умолчанию: true
state		Object	Состояние провайдера.
	timestamp	Number	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).

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.
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
    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 балунов по клику для слоев пробок.
myMap.layers.options.set({
    // Название опции сформировано из опции хотспотного слоя
    // 'openBalloonOnClick' путем добавления префикса 'trafficJam'.
    trafficJamOpenBalloonOnClick: false
});

// ...
// Удаление 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 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.

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(Ссылка)		

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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('Провайдер готов предоставлять данные.');
    }
});
```

traffic.provider.Forecast

Входит в пакеты package.full, package.traffic.

Расширяет **ITrafficProvider**.

Провайдер прогноза пробок. Доступен в хранилище провайдеров под ключом 'traffic#forecast'.

Конструктор | Поля | События | Методы

Конструктор

```
traffic.provider.Forecast(options, state)
```

Создает провайдер прогноза пробок.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Свойства параметра	Тип	Описание
options		Object	Опции провайдера. Опции для слоев провайдера задаются через опции глобальной коллекции слоев **Map.layers**. • Опции для картиночного слоя **Layer** задаются с префиксом 'trafficImage'; • Опции для слоя хотспотов **hotspot.Layer** задаются с префиксом 'trafficJam'.
	autoUpdate	Boolean	Флаг, включающий автоматическое обновление данных о пробках. Автоматическое обновление происходит только в том случае, если на карте каждые 4 минуты возникает событие mousemove. Если событие не возникает, пробки перестают обновляться до появления нового события. Значение по умолчанию: true
state		Object	Состояние провайдера.
	timeOffset	Boolean	временной интервал между текущим временем и временем прогноза. Измеряется в секундах и кратен 15 минутам (900 секундам). // Создадим провайдер пробок с прогнозом на полчаса вперед // и разместим его на карте. var forecastProvider = new ymaps.traffic.provider.Forecast({}, {timeOffset: 30 * 60}); forecastProvider.setMap(myMap); // Запретим показ балунов по клику для слоев пробок. myMap.layers.options.set({ // Название опции сформировано из опции хотспотного слоя // 'openBalloonOnClick' путем добавления префикса 'trafficJam'. trafficJamOpenBalloonOnClick: false }); // ... // Удаление провайдера с карты. forecastProvider.setMap(null);

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	Описание
			Значение по умолчанию: 15

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 data.Manager.get: • isInited — признак, готов ли провайдер предоставлять данные; • timestamp — текущее время в формате unix timestamp 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. • timeOffset — временной интервал между текущим временем и временем прогноза. Измеряется в секундах и кратен 15 минутам (900 секундам) • isotime — 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(Ссылка)		

Описание полей

state

```
data.Manager state
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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

- isInited — признак, готов ли провайдер предоставлять данные;
- timestamp — текущее время в формате unix timestamp 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.
- timeOffset — временной интервал между текущим временем и временем прогноза. Измеряется в секундах и кратен 15 минутам (900 секундам)
- isotime — строка, содержащая текущую дату в формате "YYYY-MM-DDThh:mm:ss±hhmm".

traffic.provider.storage

Статический объект.

Экземпляр класса **util.Storage**

Хранилище провайдеров.

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

util

util.augment

Входит в пакеты package.full, package.map, package.standard.

Статическая функция.

Базовая функция, реализующая наследование в JavaScript. Реализует наследование прототипа без исполнения конструктора родителя. К дочернему классу приписывается поле superclass, указывающее на прототип родительского класса.

Возвращает: Прототип дочернего класса. (тип Object).

Параметры:

Параметр	Тип	*	Описание
ChildClass	Function	✔	Дочерний класс.
ParentClass	Function	✔	Родительский класс.
override	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 от ParentClass
ymaps.util.augment(ChildClass, ParentClass, {
    // переопределяем в наследнике метод foo
    foo: function () {
        // Вызываем метод родительского класса
        ChildClass.superclass.foo.call(this);
        alert('Child!');
    }
});
```

util.bind

Входит в пакеты package.full, package.map, package.standard.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Статическая функция.

Связывает переданную функцию с переданным контекстом исполнения.

Параметры:

Параметр	Тип	*	Описание
callback	Function	✔	Функция.
context	Object	✔	Контекст исполнения.

* Обязательный параметр/опция.

Пример:

```
var myObject = {
        name: 'test!'
    };
ymaps.geocode.load('Москва')
    then(ymaps.util.bind(function (res) {
        alert(this.name); // test!
    }, myObject));
```

util.bounds

Входит в пакеты package.full, package.map, package.standard.

Статический объект.

Позволяет работать с границами карты.

util.cursor

util.cursor.Accessor

Объект, предоставляющий доступ к добавленному на карту курсору.

Конструктор | Методы

Конструктор

```
util.cursor.Accessor(key)
```

Параметры:

Параметр	Тип	*	Описание
key	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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
remove()		Удаляет курсор с карты.
setKey(key)		Задает 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.

Возвращает .

remove

```
remove()
```

Удаляет курсор с карты.

setKey

```
setKey(key)
```

Задает 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	Тип	*	Описание
key	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	Тип	*	Описание
element	HTMLElement	✔	DOM-элемент, для которого выполняется установка курсоров.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// 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
events	event.Manager	Менеджер событий менеджера курсоров.

События

Имя	Описание
change	Изменение курсора.

Методы

Имя	Возвращает	Описание
push(key)	util.cursor.Accessor	Устанавливает новый курсор и дописывает его в стэк курсоров объекта.

Описание полей

events

```
event.Manager events
```

Менеджер событий менеджера курсоров.

Описание событий

change

```
change
```

Изменение курсора.

Описание методов

push

```
{util.cursor.Accessor} push(key)
```

Устанавливает новый курсор и дописывает его в стэк курсоров объекта.

Возвращает объект, предоставляющий доступ к добавленному на карту курсору.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
`key`	String	✔	Курсор. Допустимые значения: • "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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

util.Dragger

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IEventEmitter**.

Специальный инструмент, который предоставляет механизм для реализации перетаскивания элементов на странице.

Конструктор | Поля | События | Методы

Конструктор

```
util.Dragger(params)
```

Параметры:

Параметр	Свойства параметра	Тип	Описание
params		Object	Параметры драггера.
	autoStartElement	HTMLElement| **IDomEventEmitter**	DOM-элемент или реализация интерфейса IDomEventEmitter, нажатие на который запустит драггер.
	byRightButton	Boolean	Для начала работы драггера используется правая кнопка мыши. На устройствах без поддержки мыши перетаскивание правой кнопкой не будет работать. Значение по умолчанию: false
	tremor	Object	Объект, который описывает чувствительность драггера при запуске. Этот параметр необходим для предотвращения запуска драггера по клику.
	tremor.offset	Number	Минимальное смещение в пикселях от стартовой точки, необходимое для запуска драггера. Значение по умолчанию: 3
	tremor.time	Number	Интервал времени после вызова метода start до начала работы драггера. Указывается в миллисекундах. Значение по умолчанию: 500

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// Пример перетаскивания DOM-элемента на карту.
// На месте отпускания DOM-элемента создается Placemark.
var myMap = new ymaps.Map('map', {center: [35.76, 37.67], zoom: 5}),
// У DOM-элемента должно быть выставлено CSS-свойство position: absolute.
    element = document.getElementById('someId'),
    dragger = new ymaps.util.Dragger({
        autoStartElement: element
    }),
    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);
        // Получаем географические координаты по точке окончания работы
драггера.
        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
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
start	Начало работы драггера. Это событие происходит не в момент нажатия, а при первом измении положения после нажатия. 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.
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.

Методы

Имя	Возвращает	Описание
destroy()		Останавливает работу драггера и удаляет прослушивание события "mousedown" у элемента autoStartElement.
isDragging()	Boolean	Возвращает работает ли сейчас драггер.
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 или по событию нажатия клавиши клавиатуры.
stop()		Останавливает драггер. Этот метод можно использовать для досрочной остановки работы драггера.

Описание событий

move

```
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

```
start
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Начало работы драггера. Это событие происходит не в момент нажатия, а при первом измении положения после нажатия. 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.

stop

```
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 или по событию нажатия клавиши клавиатуры.

Параметры:

Параметр	Тип	*	Описание
event	**DomEvent**	✔	Инициализирующее событие.

* Обязательный параметр/опция.

stop

```
stop()
```

Останавливает драггер. Этот метод можно использовать для досрочной остановки работы драггера.

util.extend

Входит в пакеты package.full, package.map, package.standard.

Статическая функция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Функция, копирующая свойства из одного или нескольких JavaScript-объектов в другой JavaScript-объект.

Параметры:

Параметр	Тип	*	Описание
target	Object	✔	Целевой JavaScript-объект. Будет модифицирован в результате работы функции.
source	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'
});
// Получим в итоге: {
//     prop1: 'a',
//     prop2: 'c',
//     prop3: 'e'
// }
```

util.math

util.math.areEqual

Входит в пакеты package.full, package.map, package.standard.

Статическая функция.

Производит сравнение двух точек с допустимой погрешностью.

Возвращает: Результат сравнения. (тип Boolean).

Параметры:

Параметр	Тип	*	Описание
first	Number[]	✔	Массив для сравнения.
second	Number[]	✔	Массив, который будем сравнивать.
diff	Number		Точность сравнения.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

util.math.cycleRestrict

Входит в пакеты package.full, package.map, package.standard.

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. (тип Number).

Параметры:

Параметр	Тип	*	Описание
value	Number	✔	Ограничиваемое значение.
min	Number	✔	Минимальный предел.
max	Number	✔	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

Входит в пакеты package.full, package.map, package.standard.

Статическая функция.

Ограничивает цифровое значение на входе заданными минимальным и максимальным пределами.

Возвращает: Ограниченное значение. (тип Number).

Параметры:

Параметр	Тип	*	Описание
value	Number	✔	Ограничиваемое значение.
min	Number	✔	Минимальный предел.
max	Number	✔	Максимальный предел.

* Обязательный параметр/опция.

Пример:

```
ymaps.util.math.restrict(-250, -180, 180) = -180
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

util.Promise

Входит в пакеты package.full, package.map, package.standard.

Расширяет **IPromiseProvider**.

Обещание выполнения функций-обработчиков.

Конструктор | Методы

Конструктор

```
util.Promise()
```

Обеспечивает асинхронный отложенный вызов функций-обработчиков. Функции-обработчики выполняются в порядке добавления. Если функция-обработчик возвращает значение, то оно будет передано в следующий обработчик. Если функция-обработчик возвращает undefined, в следующий обработчик будет передан тот же входной параметр, что и в текущую функцию.

Пример:

```
// создаем экземпляр
var promise = new ymaps.util.Promise(),
    a,
    b,
    c,
    d;
// навешиваем события
promise.then(function (res) { return res + 1; })
       .then(function (res) { a = res; })
       .then(function (res) { b = res; return 3; }),
       .then(function (res) { c = res; });
promise.then(function (res) { d = res; });

// если функция возвращает не undefined, то возвращаемое значение передается в
// следующую по цепочке функцию, иначе в нее передается значение, поступившее на
// вход функции стоящей выше

// выполняем наше "обещание"
promise.resolve(1);
a == 2; // true
b == 2; // true
c == 3; // true
d == 1; // true
```

Методы

Имя	Возвращает	Описание
reject(error)		Прерываем слушание.
resolve(value)		Вызывает функции-обработчики, когда "обещание" выполнено.
then(onResolve, onReject)	**util.Promise**	Подписка на "обещание".

Описание методов

reject

```
reject(error)
```

Прерываем слушание.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	Описание
error	String|Object	Ошибка, подаваемая на функцию-обработчик.

resolve

```
resolve(value)
```

Вызывает функции-обработчики, когда "обещание" выполнено.

Параметры:

Параметр	Тип	Описание
value	Object	Значение, подающееся на функцию-обработчик.

then

```
{util.Promise} then(onResolve, onReject)
```

Подписка на "обещание".

Возвращает указатель на объект Promise.

Параметры:

Параметр	Тип	Описание
onResolve	Function	Функция-обработчик, которая вызывается, если обещание выполнено.
onReject	Function	Функция-обработчик, которая вызывается, если обещание не выполнено (произошла ошибка).

util.Storage

Входит в пакеты package.full, package.map, package.standard.

Хранилище объектов по ключу.

Конструктор | Методы

Конструктор

```
util.Storage()
```

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

add

```
{util.Storage} add(key, object)
```

Добавляет объект в хранилище.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
key	String	✔	Ключ.
object	Object	✔	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	Тип	*	Описание
key	String|Object	✔	Ключ.

* Обязательный параметр/опция.

remove

```
{util.Storage} remove(key)
```

Удаляет пару "ключ: значение" из хранилища.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
key	String	✔	Ключ.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Интерфейсы

IBalloonLayout

Расширяет **ILayout**.

Класс для создания макета балуна.

Конструктор | Поля | События | Методы

Конструктор

```
IBalloonLayout()
```

Интерфейс макета балуна.

Параметры:

Параметр	Свойства параметра	Тип	*	Описание
	options.contentLayout	**ILayout**	✔	HTML содержимое макета.
	options.closeButton	Boolean		Флаг наличия кнопки закрытия.
	options.closeButtonLayout	**ILayout**		Макет кнопки закрытия
	options.maxHeight	String		Максимально допустимая высота.
	options.maxWidth	String		Максимально допустимая ширина.

* 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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от **ILayout**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **ILayout**.
userclose	Нажата кнопка закрытия.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от **ILayout**.
getData()	Object	Возвращает объект данных макета. Унаследован от **ILayout**.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от **ILayout**.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от **ILayout**.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от **ILayout**.

Описание событий

userclose

```
userclose
```

Нажата кнопка закрытия.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

IBaseCircleGeometry

Расширяет **IBaseGeometry**, **ICircleGeometryAccess**.

Интерфейс геометрии "Круг".

Конструктор | Поля | События | Методы

Конструктор

```
IBaseCircleGeometry()
```

Поля

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

События

Имя	Описание
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. Унаследовано от **ICircleGeometryAccess**.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри круга. Унаследован от **ICircleGeometryAccess**.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.
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 окружности точку, ближайшую к anchorPosition. Унаследован от **ICircleGeometryAccess**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getCoordinates()	Number[]|null	Возвращает координаты центра круга. Унаследован от ICircleGeometryAccess.
getRadius()	Number	Возвращает радиус круга. Унаследован от ICircleGeometryAccess.
getType()	String	Возвращает строку "Circle".
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.
setRadius(radius)	ICircleGeometryAccess	Задает радиус круга. Унаследован от ICircleGeometryAccess.
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable.

Описание методов

getType

```
{String} getType()
```

Возвращает строку "Circle".

IBaseGeometry

Расширяет IEventEmitter.

Интерфейс базовой геометрии.

Конструктор | Поля | Методы

Конструктор

```
IBaseGeometry()
```

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

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.
getType()	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
events	IEventManager	Менеджер событий. Унаследовано от **IEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
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. Унаследовано от ILinearRingGeometryAccess.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри контура. Унаследован от ILinearRingGeometryAccess.
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
get(index)	Number[]	Возвращает координаты точки с заданным индексом. Унаследован от ILinearRingGeometryAccess.
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 контура. Унаследован от ILinearRingGeometryAccess.
getClosest(anchorPosition)	Object	Ищет на контуре точку, ближайшую к anchorPosition. 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **ILinearRingGeometryAccess**.
getType()	String	Возвращает строку "LinearRing".
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**	Задает координаты точки с заданным индексом. Унаследован от **ILinearRingGeometryAccess**.
setCoordinates(coordinates)	**ILinearRingGeometryAccess**	Задает массив координат геометрии. Унаследован от **ILinearRingGeometryAccess**.
setFillRule(fillRule)	**ILinearRingGeometryAccess**	Задает алгоритм заливки контура. Унаследован от **ILinearRingGeometryAccess**.
splice(index, number)	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 **ILinearRingGeometryAccess**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

Описание методов

getType

```
{String} getType()
```

Возвращает строку "LinearRing".

IBaseLineStringGeometry

Расширяет **IBaseGeometry**, **ILineStringGeometryAccess**.

Интерфейс геометрии "Ломаная линия".

Конструктор | Поля | События | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
IBaseLineStringGeometry()
```

Поля

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

События

Имя	Описание
change	Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты. Унаследовано от ILineStringGeometryAccess.

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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **ILineStringGeometryAccess**.
getType()	String	Возвращает строку "LineString".
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**.
set(index, coordinates)	**ILineStringGeometryAccess**	Задает координаты точки с заданным индексом. Унаследован от **ILineStringGeometryAccess**.
setCoordinates(coordinates)	**ILineStringGeometryAccess**	Задает массив координат геометрии. Унаследован от **ILineStringGeometryAccess**.
splice(index, number)	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**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

Описание методов

getType

```
{String} getType()
```

Возвращает строку "LineString".

IBasePointGeometry

Расширяет **IBaseGeometry**, **IPointGeometryAccess**.

Интерфейс геометрии "Точка".

Конструктор | Поля | События | Методы

Конструктор

```
IBasePointGeometry()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Поля

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

События

Имя	Описание
change	Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты. Унаследовано от IPointGeometryAccess.

Методы

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.
getCoordinates()	Number[]|null	Возвращает координаты точки. Унаследован от IPointGeometryAccess.
getType()	String	Возвращает строку "Point".
setCoordinates(coordinates)	IPointGeometryAccess	Задает координаты точки. Унаследован от IPointGeometryAccess.

Описание методов

getType

```
{String} getType()
```

Возвращает строку "Point".

IBasePolygonGeometry

Расширяет IBaseGeometry, IPolygonGeometryAccess.

Интерфейс геометрии "Многоугольник".

Конструктор | Поля | События | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
IBasePolygonGeometry()
```

Поля

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

События

Имя	Описание
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. Унаследовано от IPolygonGeometryAccess.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри многоугольника. Унаследован от IPolygonGeometryAccess.
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
get(index)	Number[][]	Возвращает координаты контура с заданным индексом. Унаследован от IPolygonGeometryAccess.
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		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".
insert(index, path)	**IPolygonGeometryAccess**	Добавляет новый контур с заданным индексом. Унаследован от **IPolygonGeometryAccess**.
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**	Задает координаты контура с заданным индексом. Унаследован от **IPolygonGeometryAccess**.
setCoordinates(coordinates)	**IPolygonGeometryAccess**	Задает массив координат геометрии. Унаследован от **IPolygonGeometryAccess**.
setFillRule(fillRule)	**IPolygonGeometryAccess**	Задает 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**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.

События

Имя	Описание
change	Изменение координат углов. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты углов; • newCoordinates — новые координаты углов. Унаследовано от **IRectangleGeometryAccess**.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри прямоугольника. Унаследован от **IRectangleGeometryAccess**.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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**.
getClosest(anchorPosition)	Object	Ищет на контуре прямоугольника точку, ближайшую к anchorPosition. Унаследован от **IRectangleGeometryAccess**.
getCoordinates()	**IRectangleGeometryAccess**	Возвращает координаты двух противоположных углов прямоугольника. Унаследован от **IRectangleGeometryAccess**.
getType()	String	Возвращает строку "Rectangle".
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от **IFreezable**.
setCoordinates(coordinates)	**IRectangleGeometryAccess**	Задает координаты двух противоположных углов прямоугольника. Унаследован от **IRectangleGeometryAccess**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

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)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	Описание
options	Object	Опции поведения.

Поля

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

События

Имя	Описание
disable	Выключение поведения.
enable	Включение поведения.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.

Методы

Имя	Возвращает	Описание
disable()		Выключает поведение.
enable()		Включает поведение.
getParent()	IParentOnMap	Возвращает ссылку на родительский объект. Унаследован от IChildOnMap.
isEnabled()	Boolean	Проверяет, включено ли поведение.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IChildOnMap.

Описание событий

disable

```
disable
```

Выключение поведения.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

enable

```
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

Интерфейс тайлов, умеющих отображаться на 2d-контексте объекта canvas.

Конструктор | Поля | События | Методы

Конструктор

```
ICanvasTile(url)
```

Параметры:

Параметр	Тип	*	Описание
url	String	✔	URL тайла.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий тайла.

События

Имя	Описание
ready	Событие готовности тайла.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает тайл.
isReady()	Boolean	Проверяет готовность тайла.
renderAt(context, canvasSize, bounds, animate)		Отрисовывает картиночный тайл на 2d-контексте объекта canvas.

Описание полей

events

```
IEventManager events
```

Менеджер событий тайла.

Описание событий

ready

```
ready
```

Событие готовности тайла.

Описание методов

destroy

```
destroy()
```

Уничтожает тайл.

isReady

```
{Boolean} isReady()
```

Проверяет готовность тайла.

Возвращает true — тайл готов, false — тайл не готов.

renderAt

```
renderAt(context, canvasSize, bounds, animate)
```

Отрисовывает картиночный тайл на 2d-контексте объекта canvas.

Параметры:

Параметр	Тип	*	Описание
context	Object	✔	2d-контекст объекта canvas.
canvasSize	Number[]	✔	размеры HTML-элемента canvas.
bounds	Number[][]	✔	Область в клиентских координатах, в которой нужно отрисовать тайл.
animate	Boolean		true — анимировать рендеринг, false — нет.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
			Значение по умолчанию: false

* Обязательный параметр/опция.

IChild

Расширяет **IEventEmitter**.

Интерфейс объекта, имеющего родителя.

Конструктор | Поля | События | Методы

Конструктор

```
IChild()
```

Поля

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

События

Имя	Описание
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель.

Методы

Имя	Возвращает	Описание
getParent()	Object	Возвращает ссылку на родительский объект.
setParent(parent)	**IChild**	Устанавливает родительский объект.

Описание событий

parentchange

```
parentchange
```

Сменился родительский объект.
Поля данных:

- oldParent — старый родитель;
- newParent — новый родитель.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

getParent

```
{Object} getParent()
```

Возвращает ссылку на родительский объект.

setParent

```
{IChild} setParent(parent)
```

Устанавливает родительский объект.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
parent	Object	✔	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
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
getParent()	IParentOnMap	Возвращает ссылку на родительский объект.
setParent(parent)	IChildOnMap	Устанавливает родительский объект.

Описание событий

mapchange

```
mapchange
```

Сменилась карта. Поля данных:

- oldMap — старая карта;
- newMap — новая карта.

Описание методов

getParent

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

Возвращает ссылку на родительский объект.

setParent

```
{IChildOnMap} setParent(parent)
```

Устанавливает родительский объект.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
parent	IParentOnMap	✔	Родительский объект.

* Обязательный параметр/опция.

ICircleGeometry

Расширяет ICircleGeometryAccess, IGeometry.

Интерфейс геометрии "Круг".

Конструктор | Поля | События | Методы

Конструктор

```
ICircleGeometry()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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: • oldCoordinates — старые координаты центра; • newCoordinates — новые координаты центра; • oldRadius — старый радиус; • newRadius — новый радиус. Унаследовано от ICircleGeometryAccess.
mapchange	Сменилась карта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия IPixelGeometry; • newPixelGeometry — новая пиксельная геометрия IPixelGeometry. Унаследовано от IGeometry.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри круга. Унаследован от ICircleGeometryAccess.
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
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".
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**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

Описание методов

getType

```
{String} getType()
```

Возвращает строку "Circle".

ICircleGeometryAccess

Расширяет **IFreezable**.

Интерфейс доступа к геометрии "Круг".

Конструктор | Поля | События | Методы

Конструктор

```
ICircleGeometryAccess()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Поля

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

События

Имя	Описание
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.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри круга.
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
getClosest(anchorPosition)	Object	Ищет на окружности точку, ближайшую к anchorPosition.
getCoordinates()	Number[]|null	Возвращает координаты центра круга.
getRadius()	Number	Возвращает радиус круга.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable.
setCoordinates(coordinates)	ICircleGeometryAccess	Задает координаты центра круга.
setRadius(radius)	ICircleGeometryAccess	Задает радиус круга.
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable.

Описание событий

change

```
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

contains

```
{Boolean} contains(position)
```

Проверяет, лежит ли переданная точка внутри круга.

Возвращает признак принадлежности точки кругу.

Параметры:

Параметр	Тип	*	Описание
position	Number[]	✔	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	Тип	*	Описание
anchorPosition	Number[]	✔	Координаты точки, для которой расчитывается 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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

setCoordinates

```
{ICircleGeometryAccess} setCoordinates(coordinates)
```

Задает координаты центра круга.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[]|null	✔	Координаты центра круга.

* Обязательный параметр/опция.

setRadius

```
{ICircleGeometryAccess} setRadius(radius)
```

Задает радиус круга.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
radius	Number	✔	Радиус круга.

* Обязательный параметр/опция.

ICollection

Расширяет **IEventEmitter**.

Интерфейс коллекции.

Конструктор | Поля | События | Методы

Конструктор

```
ICollection()
```

Поля

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

События

Имя	Описание
add	Произошло добавление дочернего объекта.
remove	Произошло удаление дочернего объекта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
add(object)	ICollection	Добавляет в коллекцию дочерний объект.
getIterator()	IIterator	Возвращает итератор по коллекции.
remove(object)	ICollection	Удаляет из коллекции дочерний объект.

Описание событий

add

```
add
```

Произошло добавление дочернего объекта.

remove

```
remove
```

Произошло удаление дочернего объекта.

Описание методов

add

```
{ICollection} add(object)
```

Добавляет в коллекцию дочерний объект.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
object	Object	✔	Добавляемый объект.

* Обязательный параметр/опция.

getIterator

```
{IIterator} getIterator()
```

Возвращает итератор по коллекции.

remove

```
{ICollection} remove(object)
```

Удаляет из коллекции дочерний объект.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
object	Object	✔	Удаляемый объект.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

* Обязательный параметр/опция.

IControl

Расширяет **IChildOnMap**.

Элемент управления.

Конструктор | Поля | События | Методы

Конструктор

```
IControl(options)
```

Интерфейс элемента управления.

Параметры:

Параметр	Тип	Описание
options	Object	Опции элемента управления.

Поля

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

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IChildOnMap**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.

Методы

Имя	Возвращает	Описание
getParent()	**IControlParent**	Возвращает ссылку на родительский объект.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект.

Описание методов

getParent

```
{IControlParent} getParent()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает ссылку на родительский объект.

setParent

```
{IChildOnMap} setParent(parent)
```

Устанавливает родительский объект.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
parent	IControlParent	✔	Родительский объект.

* Обязательный параметр/опция.

IControlParent

Расширяет IParentOnMap.

Интерфейс родительского объекта для элемента управления.

Конструктор | События | Методы

Конструктор

```
IControlParent()
```

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.

Методы

Имя	Возвращает	Описание
getChildElement(child)	HTMLElement	Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.

Описание методов

getChildElement

```
{HTMLElement} getChildElement(child)
```

Возвращает для своего дочернего объекта HTML-элемент, в котором он должен размещаться.

Возвращает HTML-элемент.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
child	IControl	✔	Дочерний объект.

* Обязательный параметр/опция.

ICoordSystem

Интерфейс координатной системы карты. Этот интерфейс необходимо реализовать в случае использования нестандартных (например, цилиндрических) координат.

См.: coordSystem.geo coordSystem.cartesian

Конструктор | Методы

Конструктор

```
ICoordSystem()
```

См.: coordSystem.geo coordSystem.cartesian

Методы

Имя	Возвращает	Описание
getDistance(point1, point2)	Number	Возвращает кратчайшее (вдоль геодезической линии) расстояние между двумя заданными точками (в метрах).
solveDirectProblem(startPoint, direction, distance)	Object	Решает так называемую первую (прямую) геодезическую задачу: где мы окажемся, если выйдем из указанной точки в указанном направлении и продём, не сворачивая, указанное расстояние.
solveInverseProblem(startPoint, endPoint, reverseDirection)	Object	Решает так называемую вторую (обратную) геодезическую задачу: построить кратчайший маршрут между двумя точками на картографируемой поверхности и определелить расстояние и направление движения.

Описание методов

getDistance

```
{Number} getDistance(point1, point2)
```

Возвращает кратчайшее (вдоль геодезической линии) расстояние между двумя заданными точками (в метрах).

Параметры:

Параметр	Тип	*	Описание
point1	Number[]	✔	Первая точка.
point2	Number[]	✔	Вторая точка.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

* Обязательный параметр/опция.

Пример:

```
// Рассчитаем расстояние 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 продём, 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 с полями:

- startPoint — начальная точка в геокоординатах,
- startDirection — начальное направление движения,
- endPoint — конечная точка в геокоординатах,
- endDirection — конечное направление движения,
- distance — расстояние,
- pathFunction — Функция, принимающая число от 0 до 1 (доля пройденного пути) и возвращающая структуру с полями point и direction.

.

Параметры:

Параметр	Тип	*	Описание
`startPoint`	Number[]	✔	Точка отправления.
`direction`	Number[]	✔	Направление. Задаётся в виде вектора (приращение координат) [dlat, dlon] либо [dlon, dlat], в завивимости от параметра coordorder. Для того, чтобы получить из заданного таким образом направления азимут (угол между направлением движения и направлением на север), необходимо вычислить

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Тип	*	Описание
			арктангенс величины dlat/dlon (в JavaScript — стандартная функция Math.atan2(dlat, dlon)); для того, чтобы по известному азимуту а вычислить направление движения, нужно положить dlat = cos(a), dlon = sin(a).
distance	Number	✔	Пройденное расстояние.

* 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('аэропорт Домодедово').then(function (res) {
    var startPoint = res.geoObjects.get(0).geometry.getCoordinates(),
    // Движемся на северо-восток, азимут 45 градусов
    // или pi/4 радиан.
        azimuth = Math.PI / 4,
    // Направление движения
        direction = [Math.sin(azimuth), Math.cos(azimuth)],
    // Путевая функция
        path = ymaps.coordSystem.geo
            .solveDirectProblem(startPoint, direction, 2e5).pathFunction;

    // Изобразим путь на карте с помощью меток, проставленных 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 определелить 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 с полями:

- startPoint — начальная точка в геокоординатах,
- startDirection — начальное направление движения,
- endPoint — конечная точка в геокоординатах,
- endDirection — конечное направление движения,
- distance — расстояние,
- pathFunction — Функция пути, принимающая число от 0 до 1 (доля пройденного пути) и возвращающая структуру с полями point и direction.

.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
startPoint	Number[]	✔	Точка отправления.
endPoint	Number[]	✔	Точка прибытия.
reverseDirection	Boolean		Направление движения. false — выбрать кратчайшую дугу, true — выбрать обратную кратчайшей. Значение по умолчанию: false

* Обязательный параметр/опция.

Пример:

```
// Построим кратчайший маршрут от Калининграда до Владивостока
// 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: 'twirl#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()
```

См.: map.Copyrights.add

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Поля

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

События

Имя	Описание
copyrightschange	Событие изменения информации об авторских правах. Унаследовано от ICopyrightsProvider.

Методы

Имя	Возвращает	Описание
getCopyrights(coords, zoom)	util.Promise	Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах. Унаследован от ICopyrightsProvider.
remove()		Удаляет добавленную через данный объект информацию об авторских правах. Унаследован от ICopyrightsProvider.
setCopyrights(copyrights)		Задаёт новое значение для добавленной через данный объект информации об авторских правах. Унаследован от ICopyrightsProvider.

ICopyrightsProvider

Расширяет IEventEmitter.

Провайдер информации об авторских правах.

Конструктор | Поля | События | Методы

Конструктор

```
ICopyrightsProvider()
```

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
copyrightschange	Событие изменения информации об авторских правах.

Методы

Имя	Возвращает	Описание
getCopyrights(coords, zoom)	util.Promise	Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах.
remove()		Удаляет добавленную через данный объект информацию об авторских правах.
setCopyrights(copyrights)		Задаёт новое значение для добавленной через данный объект информации об авторских правах.

Описание событий

copyrightschange

```
copyrightschange
```

Событие изменения информации об авторских правах.

Описание методов

getCopyrights

```
{util.Promise} getCopyrights(coords, zoom)
```

Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах.

Возвращает объект-promise.

Параметры:

Параметр	Тип	*	Описание
coords	Number[]	✔	Точка на карте, для которой запрашивается информация об авторских правах.
zoom	Number	✔	Уровень масштабирования, для которого запрашивается информация об авторских правах.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
copyrights	String|HTMLElement| String[]|HTMLElement[]	✔	Информация об авторских правах.

* Обязательный параметр/опция.

ICustomizable

Интерфейс объекта, который можно настраивать с помощью глобального реестра опций.

Конструктор | Поля | События

Конструктор

```
ICustomizable()
```

Поля

Имя	Тип	Описание
options	IOptionManager	Менеджер опций.

События

Имя	Описание
optionschange	Изменение в опциях объекта.

Описание полей

options

```
IOptionManager options
```

Менеджер опций.

Описание событий

optionschange

```
optionschange
```

Изменение в опциях объекта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

IDataManager

Расширяет **IEventEmitter**.

Интерфейс менеджера произвольных данных.

Конструктор | Поля | События | Методы

Конструктор

```
IDataManager()
```

Поля

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

События

Имя	Описание
change	Изменение данных.

Методы

Имя	Возвращает	Описание
get(path, defaultValue)	Object	Возвращает значение поля данных.

Описание событий

change

```
change
```

Изменение данных.

Описание методов

get

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

Возвращает значение поля данных.

Параметры:

Параметр	Тип	*	Описание
path	String	✔	Строка с именем свойства; может содержать '.'.
defaultValue	Object	✔	Значение по умолчанию.

* Обязательный параметр/опция.

Пример:

```
dataManager.get("icon.shadow.size", [10, 10]);
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

IDomEvent

Расширяет **IEvent**.

Объект-обёртка, предоставляющий доступ к DOM-событию. Посредством методов get и callMethod можно получать доступ к полям и методам оригинального DOM-события. При этом автоматически выполняется нормализация (приведение к стандарту) для тех полей и методов, реализация которых различна в разных браузерах. Так, event.callMethod('stopPropagation') остановит пропагацию события во всех браузерах, включая Internet Explorer.

Конструктор | Методы

Конструктор

```
IDomEvent(originalEvent)
```

Параметры:

Параметр	Тип	*	Описание
originalEvent	Object	✔	Исходное DOM-событие.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
callMethod(name)		Вызывает указанный метод у исходного DOM-события. Второй и последующие аргументы передаются методу при вызове.
get(name)	Object	Возвращает свойство события по ключу. С помощью этого метода можно получать доступ как к свойствам оригинального события, так и к дополнительным свойствам, предоставляемым API карт.
isImmediatePropagationStopped()		Возвращает true, если распространение события внутри системы событий API Яндекс.Карт остановлено. Это внутренний признак, никак не связанный с пропагацией события по DOM-дереву.

Описание методов

callMethod

```
callMethod(name)
```

Вызывает указанный метод у исходного DOM-события. Второй и последующие аргументы передаются методу при вызове.

Параметры:

Параметр	Тип	*	Описание
name	String	✔	Имя метода.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

get

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

Возвращает свойство события по ключу. С помощью этого метода можно получать доступ как к свойствам оригинального события, так и к дополнительным свойствам, предоставляемым API карт.

Возвращает значение свойства.

Параметры:

Параметр	Тип	*	Описание
name	String	✔	Имя свойства. Поддерживаются дополнительные свойства: • 'propagatedData' — данные события, которые сохраняются при его пропагировании (распространении по DOM-дереву).

* Обязательный параметр/опция.

isImmediatePropagationStopped

```
isImmediatePropagationStopped()
```

Возвращает true, если распространение события внутри системы событий API Яндекс.Карт остановлено. Это внутренний признак, никак не связанный с пропагацией 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
events	**IEventManager**	Менеджер событий.

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.
multitouchend	Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях.
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.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.

Описание полей

events

```
IEventManager events
```

Менеджер событий.

Описание событий

click

```
click
```

Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.

contextmenu

```
contextmenu
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

dblclick

```
dblclick
```

Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

mousedown

```
mousedown
```

Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

mouseenter

```
mouseenter
```

Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

mouseleave

```
mouseleave
```

Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

mousemove

```
mousemove
```

Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

mouseup

```
mouseup
```

Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

multitouchend

```
multitouchend
```

Окончание мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях.

multitouchmove

```
multitouchmove
```

Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

multitouchstart

```
multitouchstart
```

Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях.

wheel

```
wheel
```

Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

IDomTile

Интерфейс тайлов, представляющих из себя DOM-объект.

Конструктор | Поля | События | Методы

Конструктор

```
IDomTile(url)
```

Параметры:

Параметр	Тип	*	Описание
url	String	✔	URL тайла.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	**IEventManager**	Менеджер событий тайла.

События

Имя	Описание
ready	Событие готовности тайла.

Методы

Имя	Возвращает	Описание
destroy()		Уничтожает тайл.
isReady()	Boolean	Проверяет готовность тайла.
renderAt(context, clientBounds, animate)		Добавляет тайл в родительский HTML-элемент.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание полей

events

```
IEventManager events
```

Менеджер событий тайла.

Описание событий

ready

```
ready
```

Событие готовности тайла.

Описание методов

destroy

```
destroy()
```

Уничтожает тайл.

isReady

```
{Boolean} isReady()
```

Проверяет готовность тайла.

Возвращает true — тайл готов, false — тайл не готов.

renderAt

```
renderAt(context, clientBounds, animate)
```

Добавляет тайл в родительский HTML-элемент.

Параметры:

Параметр	Тип	*	Описание
context	HTMLElement	✔	Родительский HTML-элемент
clientBounds	Number[][]	✔	Область в клиентских координатах, которую должен занять тайл.
animate	Boolean	✔	true — анимировать рендеринг, false — нет.

* Обязательный параметр/опция.

IEvent

Событие, бросаемое менеджером событий IEventManager.

Конструктор | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
IEvent()
```

Методы

Имя	Возвращает	Описание
isImmediatePropagationStopped()	Boolean	Проверяет, прервано ли событие кем-то из слушателей. Этот метод не является обязательным; если он отсутствует, то прервать обработку события нельзя.

Описание методов

isImmediatePropagationStopped

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

Проверяет, прервано ли событие кем-то из слушателей. Этот метод не является обязательным; если он отсутствует, то прервать обработку события нельзя.

Возвращает true, если обработка события прервана, false — в противном случае.

IEventController

Интерфейс контроллера событий. Позволяет контролировать подписки и отписки от событий на определённом менеджере событий.

Конструктор | Методы

Конструктор

```
IEventController()
```

Методы

Имя	Описание
onStartListening(events, type)	Вызывается при первой подписке на указанный тип событий через указанный менеджер событий. Этот метод является необязательным.
onStopListening(events, type)	Вызывается, когда определённый тип событий перестаёт слушаться на указанном менеджере событий (удалена последняя подписка). Этот метод является необязательным.

Описание методов

onStartListening

```
onStartListening(events, type)
```

Вызывается при первой подписке на указанный тип событий через указанный менеджер событий. Этот метод является необязательным.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
events	IEventManager	✔	Менеджер событий.
type	String	✔	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 необязательным.

Параметры:

Параметр	Тип	*	Описание
events	IEventManager	✔	Менеджер событий.
type	String	✔	Тип события.

* Обязательный параметр/опция.

IEventEmitter

Интерфейс объекта, у которого можно слушать события.

Конструктор | Поля

Конструктор

```
IEventEmitter()
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий.

Описание полей

events

```
IEventManager events
```

Менеджер событий.

IEventGroup

Контейнер событий.

Конструктор | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
IEventGroup(events)
```

Параметры:

Параметр	Тип	*	Описание
events	event.Manager	✔	Менеджер событий, для которого создается контейнер.

* Обязательный параметр/опция.

Методы

Имя	Возвращает	Описание
add(types, callback, context)	IEventGroup	Добавляет обработчик события.
remove(types, callback, context)	IEventGroup	Удаляет обработчик события. Удалить можно только те обработчики, которые были добавлены через данный экземпляр контейнера.
removeAll()	IEventGroup	Удаляет все обработчики событий, которые были добавлены через данный экземпляр контейнера.

Описание методов

add

```
{IEventGroup} add(types, callback, context)
```

Добавляет обработчик события.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или типы событий.
callback	Function	✔	Функция-обработчик. В качестве параметра в функцию передается объект, описывающий событие. Может быть либо произвольным объектом, либо реализовывать интерфейс IEvent.
context	Object		Контекст исполнения функции-обработчика.

* Обязательный параметр/опция.

remove

```
{IEventGroup} remove(types, callback, context)
```

Удаляет обработчик события. Удалить можно только те обработчики, которые были добавлены через данный экземпляр контейнера.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или типы событий.
callback	Function	✔	Функция-обработчик. В качестве параметра в функцию передается объект, описывающий событие. Может быть либо произвольным объектом, либо реализовывать интерфейс **IEvent**.
context	Object		Контекст исполнения функции-обработчика.

* Обязательный параметр/опция.

removeAll

```
{IEventGroup} removeAll()
```

Удаляет все обработчики событий, которые были добавлены через данный экземпляр контейнера.

Возвращает ссылку на себя.

IEventManager

Менеджер событий. Большинство объектов API имеют поле events, реализующее данный интерфейс. При помощи менеджера событий можно подписываться и отписываться от событий, а также инициировать сами события.

Конструктор | Методы

Конструктор

```
IEventManager()
```

Методы

Имя	Возвращает	Описание
add(types, callback, context)	**IEventManager**	Добавляет новую подписку.
fire(type, eventObject)	**IEventManager**	Бросает событие.
getParent()	**IEventManager**|null	Возвращает ссылку на родительский менеджер событий.
group()	**IEventGroup**	Возвращает контейнер событий, привязанный к данному менеджеру.
remove(types, callback, context)	**IEventManager**	Удаляет существующую подписку.
setParent(parent)		Устанавливает родительский менеджер событий.

Описание методов

add

```
{IEventManager} add(types, callback, context)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Добавляет новую подписку.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или массив типов события.
callback	Function	✔	Функция-обработчик события. В качестве параметра в функцию передается объект, описывающий событие. Может быть либо произвольным объектом, либо реализовывать интерфейс **IEvent**.
context	Object		Контекст исполнения обработчика.

* Обязательный параметр/опция.

fire

```
{IEventManager} fire(type, eventObject)
```

Бросает событие.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
type	String	✔	Тип события.
eventObject	Object|**IEvent**		Объект, описывающий событие. Может быть либо произвольным объектом, либо реализовывать интерфейс **IEvent**. Во втором случае после вызова каждого обработчика будет проверяться значение метода isImmediatePropagationStopped() — если оно истинно, оповещение о событии немедленно прекращается.

* Обязательный параметр/опция.

getParent

```
{IEventManager|null} getParent()
```

Возвращает ссылку на родительский менеджер событий.

group

```
{IEventGroup} group()
```

Возвращает контейнер событий, привязанный к данному менеджеру.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

remove

```
{IEventManager} remove(types, callback, context)
```

Удаляет существующую подписку.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
types	String|String[]	✔	Тип или массив типов события.
callback	Function	✔	Функция-обработчик события. В качестве параметра в функцию передается объект, описывающий событие. Может быть либо произвольным объектом, либо реализовывать интерфейс **IEvent**.
context	Object		Контекст исполнения обработчика.

* Обязательный параметр/опция.

setParent

```
setParent(parent)
```

Устанавливает родительский менеджер событий.

Параметры:

Параметр	Тип	*	Описание
parent	**IEventManager**|null	✔	Родительский менеджер событий.

* Обязательный параметр/опция.

IEventWorkflowController

Расширяет **IEventController**.

Интерфейс контроллера событий, позволяющего влиять на распространение событий по дереву.

Конструктор | Методы

Конструктор

```
IEventWorkflowController()
```

Методы

Имя	Описание
onAfterEventFiring(events, type, event)	Функция, вызываемая после окончания обработки события менеджером событий. Этот метод является необязательным.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
onBeforeEventFiring(events, type, event)	Функция, вызываемая до начала обработки события менеджером событий. Этот метод является необязательным.
onStartListening(events, type)	Вызывается при первой подписке на указанный тип событий через указанный менеджер событий. Этот метод является необязательным. Унаследован от **IEventController**.
onStopListening(events, type)	Вызывается, когда определённый тип событий перестаёт слушаться на указанном менеджере событий (удалена последняя подписка). Этот метод является необязательным. Унаследован от **IEventController**.

Описание методов

onAfterEventFiring

```
onAfterEventFiring(events, type, event)
```

Функция, вызываемая после окончания обработки события менеджером событий. Этот метод является необязательным.

Параметры:

Параметр	Тип	*	Описание
events	**IEventManager**	✔	Менеджер событий.
type	String	✔	Тип события.
event	**IEvent**		Событие.

* Обязательный параметр/опция.

onBeforeEventFiring

```
onBeforeEventFiring(events, type, event)
```

Функция, вызываемая до начала обработки события менеджером событий. Этот метод является необязательным.

Параметры:

Параметр	Тип	*	Описание
events	**IEventManager**	✔	Менеджер событий.
type	String	✔	Тип события.
event	**IEvent**		Событие.

* Обязательный параметр/опция.

IExpandableControlLayout

Расширяет **ILayout**.

Интерфейс макета, который может находится в свернутом и развернутом состоянии.

Конструктор | Поля | События | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
IExpandableControlLayout()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от ILayout.
collapse	Событие, инициирующее сворачивание объекта.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от ILayout.
expand	Событие, инициирующее разворачивание объекта.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от **ILayout**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от **ILayout**.
getData()	Object	Возвращает объект данных макета. Унаследован от **ILayout**.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от **ILayout**.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от **ILayout**.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от **ILayout**.

Описание событий

collapse

```
collapse
```

Событие, инициирующее сворачивание объекта.

expand

```
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()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий объекта.

События

Имя	Описание
change	Изменение внутреннего состояния объекта.

Методы

Имя	Возвращает	Описание
freeze()	IFreezable	Переводит объект в "замороженный" режим.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false.
unfreeze()	IFreezable	Переводит объект в активный режим.

Описание полей

events

```
IEventManager events
```

Менеджер событий объекта.

Описание событий

change

```
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает Ссылку на себя.

IGeocodeProvider

Интерфейс провайдера геокодера.

Конструктор | Методы

Конструктор

```
IGeocodeProvider()
```

Методы

Имя	Возвращает	Описание
geocode(request, options)	util.Promise	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.

Описание методов

geocode

```
{util.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	Свойства параметра	Тип	*	Описание
request		String	✔	Строка запроса.
options		Object		Опции.
	boundedBy	Number[][]		Прямоугольная область на карте, где предположительно находится искомый объект. Должна быть задана как массив, например [[30, 40], [50, 50]].
	results	Number		Максимальное количество возвращаемых результатов.
	skip	Number		Число результатов, которое необходимо пропустить.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметр	Свойства параметра	Тип	*	Описание
	strictBounds	Boolean		Искать только внутри области, заданной опцией boundedBy.

* Обязательный параметр/опция.

IGeometry

Расширяет **IBaseGeometry**, **ICustomizable**.

Интерфейс геометрии.

Конструктор | Поля | События | Методы

Конструктор

```
IGeometry()
```

Поля

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

События

Имя	Описание
mapchange	Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — новая карта.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
pixelgeometrychange	Изменилась пиксельная геометрия. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**; • newPixelGeometry — новая пиксельная геометрия **IPixelGeometry**.

Методы

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.
getMap()	**Map**|null	Возвращает текущую карту.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getPixelGeometry(options)	IPixelGeometry	Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry.
setMap(map)		Задает карту.

Описание событий

mapchange

```
mapchange
```

Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldMap — старая карта;
- newMap — новая карта.

pixelgeometrychange

```
pixelgeometrychange
```

Изменилась пиксельная геометрия. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**;
- newPixelGeometry — 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)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты.

Параметры:

Параметр	Тип	Описание
options	Object	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	Тип	*	Описание
map	Map|null	✔	Ссылка на карту.

* Обязательный параметр/опция.

IGeometryEditor

Расширяет **ICustomizable**, **IEventEmitter**.

Интерфейс редактора геометрии.

Конструктор | Поля | События | Методы

Конструктор

```
IGeometryEditor()
```

Поля

Имя	Тип	Описание
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
geometry	**IGeometry**	Редактируемая геометрия.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.
state	**IDataManager**	Состояние редактора геометрии.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
statechange	Изменение состояния редактора геометрии. Экземпляр класса Event.

Методы

Имя	Описание
startEditing()	Включает режим редактирования.
stopEditing()	Отключает режим редактирования.

Описание полей

geometry

```
IGeometry geometry
```

Редактируемая геометрия.

state

```
IDataManager state
```

Состояние редактора геометрии.

Описание событий

statechange

```
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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
IGeometryEditorChildModel(geometry, editor, pixels, parent)
```

Параметры:

Параметр	Тип	*	Описание
geometry	IBaseGeometry	✔	Редактируемая дочерняя геометрия. Интерфейс IBaseGeometry не подразумевает наличие события pixelgeometrychange, поэтому пиксельные данные получаются от родительской модели данных.
editor	IGeometryEditor	✔	Ссылка на редактор геометрии.
pixels	Number[]	✔	Пиксельные данные модели.
parent	IGeometryEditorModel	✔	Родительская модель данных.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
editor	IGeometryEditor	Редактор геометрии.
events	IEventManager	Менеджер событий. Унаследовано от IEventEmitter.
geometry	IBaseGeometry	Геометрия модели.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel.
getParent()	IGeometryEditorModel	Возвращает родительскую модель данных.
getPixels()	Number[]	Возвращает пиксельные данные модели. Унаследован от IGeometryEditorModel.
setPixels(pixels)		Задает пиксельные данные модели.

Описание полей

editor

```
IGeometryEditor editor
```

Редактор геометрии.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

geometry

```
IBaseGeometry geometry
```

Геометрия модели.

Описание методов

getParent

```
{IGeometryEditorModel} getParent()
```

Возвращает родительскую модель данных.

setPixels

```
setPixels(pixels)
```

Задает пиксельные данные модели.

Параметры:

Параметр	Тип	*	Описание
pixels	Number[]	✔	Пиксельные данные.

* Обязательный параметр/опция.

IGeometryEditorModel

Расширяет **IEventEmitter**.

Интерфейс модели данных редактора геометрии.

Конструктор | Поля | Методы

Конструктор

```
IGeometryEditorModel(geometry, editor)
```

Параметры:

Параметр	Тип	*	Описание
geometry	**IBaseGeometry**	✔	Редактируемая геометрия.
editor	**IGeometryEditor**	✔	Ссылка на редактор геометрии.

* Обязательный параметр/опция.

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
destroy()		Деструктор.
getPixels()	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	Тип	*	Описание
geometry	IGeometry	✔	Редактируемая геометрия.
editor	IGeometryEditor	✔	Ссылка на редактор геометрии.

* Обязательный параметр/опция.

Поля

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

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Унаследован от IGeometryEditorModel.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getPixels()	Number[]	Возвращает пиксельные данные модели. Унаследован от IGeometryEditorModel.

IGeoObject

Расширяет IChildOnMap, ICustomizable, IDomEventEmitter, IParentOnMap.

Интерфейс геообъекта.

Конструктор | Поля | События | Методы

Конструктор

```
IGeoObject()
```

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий. Унаследовано от IDomEventEmitter.
geometry	IGeometry	Геометрия геообъекта.
options	IOptionManager	Менеджер опций. Унаследовано от ICustomizable.
properties	IDataManager	Менеджер данных геообъекта.
state	IDataManager	Состояние геообъекта. Определяется следующими полями: • active — признак того, что на геообъекте открыт балун. • hover — признак того, что в данный момент на геообъект наведен указатель мыши. • drag — признак того, что геообъект перетаскивается.

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
geometrychange	Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие геометрии.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IParentOnMap**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
overlaychange	Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newOverlay — новый оверлей, либо null, если оверлея нет; • oldOverlay — старый оверлей, либо null, если оверлея нет.
parentchange	Сменился родительский объект.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
pixelgeometrychange	Изменение пиксельной геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • newPixelGeometry — новая пиксельная геометрия; • oldPixelGeometry — старая пиксельная геометрия.
propertieschange	Изменение данных геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • originalEvent — оригинальное событие менеджера данных.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
getMap()	**Map**	Возвращает ссылку на карту. Унаследован от **IParentOnMap**.
getOverlay()	**IOverlay**|null	Возвращает оверлей геообъекта, либо null.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

Описание полей

geometry

```
IGeometry geometry
```

Геометрия геообъекта.

Пример:

```
// При изменении координат геометрии геообъекта устанавливаем карте границы так,
// чтобы охватить геообъект целиком.
myMap.geoObjects.add(myGeoObject);
myGeoObject.events.add("geometrychange", function () {
    myMap.setBounds(myGeoObject.geometry.getBounds());
});
```

properties

```
IDataManager properties
```

Менеджер данных геообъекта.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// При изменении данных выводим пользовательские идентификатор,
// если объект у объекта поле synchronized ложно.
myGeoObject.events.add("propertieschange", function () {
    if (!myGeoObject.properties.get("synchronized")) {
        console.log(myGeoObject.properties.get("myID"));
    }
});
```

state

```
IDataManager state
```

Состояние геообъекта. Определяется следующими полями:

- active — признак того, что на геообъекте открыт балун.
- hover — признак того, что в данный момент на геообъект наведен указатель мыши.
- drag — признак того, что геообъект перетаскивается.

Описание событий

geometrychange

```
geometrychange
```

Изменение геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- originalEvent — оригинальное событие геометрии.

overlaychange

```
overlaychange
```

Изменение оверлея геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- newOverlay — новый оверлей, либо null, если оверлея нет;
- oldOverlay — старый оверлей, либо null, если оверлея нет.

pixelgeometrychange

```
pixelgeometrychange
```

Изменение пиксельной геометрии геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- newPixelGeometry — новая пиксельная геометрия;
- oldPixelGeometry — старая пиксельная геометрия.

propertieschange

```
propertieschange
```

Изменение данных геообъекта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- originalEvent — оригинальное событие менеджера данных.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

getOverlay

```
{IOverlay|null} getOverlay()
```

Возвращает оверлей геообъекта, либо null.

IGroupControlLayout

Расширяет **ILayout**.

Интерфейс макета группового элемента управления.

Конструктор | Поля | События | Методы

Конструктор

```
IGroupControlLayout(data)
```

Параметры:

Параметр	Тип	*	Описание
data	Object	✔	Данные макета.

* Обязательный параметр/опция.

Поля

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

События

Имя	Описание
childcontainerchange	Изменение контейнера дочерних элементов. Поля данных: • newChildContainerElement — новый элемент для дочерних объектов; • oldChildContainerElement — старый элемент для дочерних объектов.
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **ILayout**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от **ILayout**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
parentelementchange	Изменение родительского элемента. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от ILayout.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от ILayout.
getChildContainerElement()	Object	Возвращает ссылку на DOM-элемент, в который должны быть добавлены дочерние элементы.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от ILayout.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от ILayout.
getData()	Object	Возвращает объект данных макета. Унаследован от ILayout.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от ILayout.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от ILayout.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от ILayout.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание событий

childcontainerchange

```
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.

IHotspotObjectSource

Расширяет **ICustomizable**.

Источник объектов для слоёв активных областей.

Конструктор | Поля | События | Методы

Конструктор

```
IHotspotObjectSource()
```

Поля

Имя	Тип	Описание
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.

Методы

Имя	Описание
cancelLastRequest()	Отменяет последний запрос за данными.
requestObjects(layer, tileNumber, zoom, callback)	Строит массив объектов **IHotspotShape**, соответствующих определенному слою, тайлу и коэффициенту масштабирования карты, и передает его в функцию callback.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

cancelLastRequest

```
cancelLastRequest()
```

Отменяет последний запрос за данными.

requestObjects

```
requestObjects(layer, tileNumber, zoom, callback)
```

Строит массив объектов **IHotspotShape**, соответствующих определенному слою, тайлу и коэффициенту масштабирования карты, и передает его в функцию callback.

Параметры:

Параметр	Тип	*	Описание
layer	**hotspot.Layer**	✔	Слой активных областей.
tileNumber	Number[]	✔	Тайловые координаты.
zoom	Number	✔	Коэффициент масштабирования.
callback	Function	✔	Функция-обработчик.

* Обязательный параметр/опция.

IHotspotShape

Расширяет **ICustomizable**, **IDomEventEmitter**.

Интерфейс фигуры, определяющей геометрию активной области.
Конструктор | Поля | События | Методы

Конструктор

```
IHotspotShape()
```

Создает фигуру, определяющую геометрию активной области.

Примечание:
Все координаты задаются в пикселах на текущем масштабе, отсчитываемых от левого верхнего угла мира.

Поля

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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
renderedgeometrychange	Событие, оповещающее о смене хотспотной геометрии.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
getContainer()	IHotspotContainer	Возвращает контейнер хотспотных шейпов.
getGeometry()	Object	Метод для получения реальной геометрии объекта.
getId()	Number	Возвращает идентификатор объекта.
getProperties()	Object	Возвращает данные объекта.
getRenderedGeometry()	IHotspotShapeGeometry	Возвращает пиксельную геометрию хотспотного объекта.
setContainer(container)		Метод для добавления хотспотного шейпа в контейнер.
setGeometry(geometry)		Метод для задания реальной геометрии объекта.
setId(id)		
setProperties(properties)		
setRenderedGeometry(renderedGeometry)		

Описание событий

renderedgeometrychange

```
renderedgeometrychange
```

Событие, оповещающее о смене хотспотной геометрии.

Описание методов

getContainer

```
{IHotspotContainer} getContainer()
```

Возвращает контейнер хотспотных шейпов.

getGeometry

```
{Object} getGeometry()
```

Метод для получения реальной геометрии объекта.

Возвращает объект-геометрию.

getId

```
{Number} getId()
```

Возвращает идентификатор объекта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getProperties

```
{Object} getProperties()
```

Возвращает данные объекта.

getRenderedGeometry

```
{IHotspotShapeGeometry} getRenderedGeometry()
```

Возвращает пиксельную геометрию хотспотного объекта.

setContainer

```
setContainer(container)
```

Метод для добавления хотспотного шейпа в контейнер.

Параметры:

Параметр	Тип	*	Описание
container	IHotspotContainer	✔	Контейнер, в который добавляется шейп.

* Обязательный параметр/опция.

setGeometry

```
setGeometry(geometry)
```

Метод для задания реальной геометрии объекта.

Параметры:

Параметр	Тип	*	Описание
geometry	Object	✔	Реальная геометрия объекта.

* Обязательный параметр/опция.

setId

```
setId(id)
```

Параметры:

Параметр	Тип	*	Описание
id	Number	✔	Идентификатор объекта.

* Обязательный параметр/опция.

setProperties

```
setProperties(properties)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
properties	Object	✔	Данные объекта.

* Обязательный параметр/опция.

setRenderedGeometry

```
setRenderedGeometry(renderedGeometry)
```

Параметры:

Параметр	Тип	*	Описание
renderedGeometry	IPixelGeometry	✔	Пиксельная геометрия объекта.

* Обязательный параметр/опция.

IHotspotShapeGeometry

Расширяет **ICustomizable**.

Интерфейс фигуры, определяющей геометрию активной области.
Конструктор | Поля | События

Конструктор

```
IHotspotShapeGeometry()
```

Создает геометрию активной области.

Примечание:
Все координаты задаются в пикселах на текущем масштабе, отсчитываемых от левого верхнего угла мира.

Поля

Имя	Тип	Описание
options	IOptionManager	Менеджер опций. Унаследовано от **ICustomizable**.

События

Имя	Описание
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.

IIterator

Итератор списка, который позволяет получить доступ к элементам списка.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор | Методы

Конструктор

```
IIterator()
```

Методы

Имя	Возвращает	Описание
getNext()	Object|null	Возвращает ссылку на очередной объект списка. При первом вызове возвращает ссылку на первый элемент. Возвращает null, если достигнут конец списка.

Описание методов

getNext

```
{Object|null} getNext()
```

Возвращает ссылку на очередной объект списка. При первом вызове возвращает ссылку на первый элемент. Возвращает null, если достигнут конец списка.

Возвращает ссылку на очередной объект или null — признак конца списка.

ILayer

Расширяет **IChildOnMap**, **ICustomizable**, **IEventEmitter**.

Интерфейс слоя карты.

Конструктор | Поля | События | Методы

Конструктор

```
ILayer()
```

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.
copyrightschange	Событие изменения доступной информации об авторских правах.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IChildOnMap**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от **IChild**.
zoomrangechange	Событие изменения доступной информации о диапазоне уровней масштабирования.

Методы

Имя	Возвращает	Описание
getBrightness()	Number	Необязательный метод. Возвращает яркость слоя. В зависимости от суммарной яркости слоев, добавленных на карту, выбирается цвет логотипа и копирайтов на карте.
getCopyrights(coords, zoom)	**util.Promise**	Необязательный метод. Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах.
getParent()	**IParentOnMap**	Возвращает ссылку на родительский объект. Унаследован от **IChildOnMap**.
getZoomRange(point)	**util.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.
setParent(parent)	**IChildOnMap**	Устанавливает родительский объект. Унаследован от **IChildOnMap**.

Описание событий

brightnesschange

```
brightnesschange
```

Событие изменения яркости слоя.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

copyrightschange

```
copyrightschange
```

Событие изменения доступной информации об авторских правах.

zoomrangechange

```
zoomrangechange
```

Событие изменения доступной информации о диапазоне уровней масштабирования.

Описание методов

getBrightness

```
{Number} getBrightness()
```

Необязательный метод. Возвращает яркость слоя. В зависимости от суммарной яркости слоев, добавленных на карту, выбирается цвет логотипа и копирайтов на карте.

Возвращает значение яркости от 0 до 1 (0 — нулевая яркость, 1 — максимальная яркость).

getCopyrights

```
{util.Promise} getCopyrights(coords, zoom)
```

Необязательный метод. Запрашивает информацию об авторских правах в указанной точке с указанным коэффициентом масштабирования. Возвращает объект-promise, который будет подтверждён и передаст в качестве результата массив строк или DOM-элементов с информацией об авторских правах.

Возвращает объект-promise.

Параметры:

Параметр	Тип	*	Описание
coords	Number[]	✔	Точка на карте, для которой запрашивается информация об авторских правах.
zoom	Number	✔	Уровень масштабирования, для которого запрашивается информация об авторских правах.

* Обязательный параметр/опция.

getZoomRange

```
{util.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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
point	Number[]	✔	Точка

* Обязательный параметр/опция.

ILayout

Расширяет **IDomEventEmitter**.

Интерфейс макета.

Конструктор | Поля | События | Методы

Конструктор

```
ILayout(data)
```

Параметры:

Параметр	Тип	*	Описание
data	Object	✔	Данные макета.

* 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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать.
getData()	Object	Возвращает объект данных макета.
getParentElement()	HTMLElement	Возвращает родительский html-элемент.
isEmpty()	Boolean	Возвращает признак того, что макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.
setData(data)		Задает данные макета.
setParentElement(parent)		Добавляет макет в DOM-дерево.

Описание событий

clientboundschange

```
clientboundschange
```

Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента;
- newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента.

emptinesschange

```
emptinesschange
```

Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldEmptiness — старое значение признака;
- newEmptiness — новое значение признака.

parentelementchange

```
parentelementchange
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- oldParentElement — старый родительский элемент;
- newParentElement — новый родительский элемент.

Описание методов

destroy

```
destroy()
```

Деструктор. Вызывается при завершении работы с макетом.

getClientBoundingRect

```
{Number[][]} getClientBoundingRect()
```

Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода.

Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу.

getClientBounds

```
{Number[][]} getClientBounds()
```

Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать.

getData

```
{Object} getData()
```

Возвращает объект данных макета.

getParentElement

```
{HTMLElement} getParentElement()
```

Возвращает родительский html-элемент.

isEmpty

```
{Boolean} isEmpty()
```

Возвращает признак того, что макет пуст, т.е. не имеет содержимого. Данный признак используется для сокрытия пустых объектов, таких как хинт, балун и другие.

setData

```
setData(data)
```

Задает данные макета.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
data	Object	✔	Данные макета.

* Обязательный параметр/опция.

setParentElement

```
setParentElement(parent)
```

Добавляет макет в DOM-дерево.

Параметры:

Параметр	Тип	*	Описание
parent	HTMLElement|null	✔	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
events	**IEventManager**	Менеджер событий объекта. Унаследовано от **IFreezable**.

События

Имя	Описание
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри контура.
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
get(index)	Number[]	Возвращает координаты точки с заданным индексом.
getChildGeometry(index)	IPointGeometryAccess	Создает и возвращает объект IPointGeometryAccess для заданной вершины контура.
getClosest(anchorPosition)	Object	Ищет на контуре точку, ближайшую к anchorPosition.
getCoordinates()	Number[][]	Возвращает массив координат геометрии.
getFillRule()	String	Возвращает идентификатор алгоритма заливки.
getLength()	Integer	Возвращает количество точек в геометрии.
insert(index, coordinates)	ILinearRingGeometryAccess	Добавляет новую точку с заданным индексом.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable.
remove(index)	Number[]	Удаляет точку с заданным индексом.
set(index, coordinates)	ILinearRingGeometryAccess	Задает координаты точки с заданным индексом.
setCoordinates(coordinates)	ILinearRingGeometryAccess	Задает массив координат геометрии.
setFillRule(fillRule)	ILinearRingGeometryAccess	Задает алгоритм заливки контура.
splice(index, number)	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.
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable.

Описание событий

change

```
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

contains

```
{Boolean} contains(position)
```

Проверяет, лежит ли переданная точка внутри контура.

Возвращает признак принадлежности точки контуру.

Параметры:

Параметр	Тип	*	Описание
position	Number[]	✔	Координаты точки.

* Обязательный параметр/опция.

get

```
{Number[]} get(index)
```

Возвращает координаты точки с заданным индексом.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс точки.

* Обязательный параметр/опция.

getChildGeometry

```
{IPointGeometryAccess} getChildGeometry(index)
```

Создает и возвращает объект **IPointGeometryAccess** для заданной вершины контура.

Возвращает объект геометрии "Точка", соответствующий заданной вершине.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Поля nextPointIndex, prevPointIndex могут отсутствовать, если position совпадает с одной из вершин контура.

Параметры:

Параметр	Тип	*	Описание
anchorPosition	Number[]	✔	Координаты точки, для которой расчитывается 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	Тип	*	Описание
index	Integer	✔	Индекс точки.
coordinates	Number[]	✔	Координаты точки.

* Обязательный параметр/опция.

remove

```
{Number[]} remove(index)
```

Удаляет точку с заданным индексом.

Возвращает координаты удаленной точки.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс точки.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

* Обязательный параметр/опция.

set

```
{ILinearRingGeometryAccess} set(index, coordinates)
```

Задает координаты точки с заданным индексом.

Возвращает ссылка на себя.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс точки.
coordinates	Number[]	✔	Координаты точки.

* Обязательный параметр/опция.

setCoordinates

```
{ILinearRingGeometryAccess} setCoordinates(coordinates)
```

Задает массив координат геометрии.

Возвращает ссылка на себя.

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[][]	✔	Координаты геометрии.

* Обязательный параметр/опция.

setFillRule

```
{ILinearRingGeometryAccess} setFillRule(fillRule)
```

Задает алгоритм заливки контура.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
fillRule	String	✔	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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс, начиная с которого производится удаление и добавление точек.
number	Integer	✔	Количество удаляемых точек.

* Обязательный параметр/опция.

ILineStringGeometry

Расширяет **IGeometry**, **ILineStringGeometryAccess**.

Интерфейс геометрии "Ломаная линия".

Конструктор | Поля | События | Методы

Конструктор

```
ILineStringGeometry()
```

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: • oldCoordinates — старые координаты; • newCoordinates — новые координаты. Унаследовано от **ILineStringGeometryAccess**.
mapchange	Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**; • newPixelGeometry — новая пиксельная геометрия **IPixelGeometry**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IGeometry**.

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 — юго-западный угол области; второй элемент — северо-восточный угол соответственно. Унаследован от **IGeometry**.
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**.
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".
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **ILineStringGeometryAccess**.
set(index, coordinates)	**ILineStringGeometryAccess**	Задает координаты точки с заданным индексом. Унаследован от **ILineStringGeometryAccess**.
setCoordinates(coordinates)	**ILineStringGeometryAccess**	Задает массив координат геометрии. Унаследован от **ILineStringGeometryAccess**.
setMap(map)		Задает карту. Унаследован от **IGeometry**.
splice(index, number)	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**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

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
events	**IEventManager**	Менеджер событий объекта. Унаследовано от **IFreezable**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
change	Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты.

Методы

Имя	Возвращает	Описание
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
get(index)	Number[]	Возвращает координаты точки с заданным индексом.
getChildGeometry(index)	IPointGeometryAccess	Создает и возвращает объект IPointGeometryAccess для заданной вершины ломаной линии.
getClosest(anchorPosition)	Object	Ищет на ломаной линии точку, ближайшую к anchorPosition.
getCoordinates()	Number[][]	Возвращает массив координат геометрии.
getLength()	Integer	Возвращает количество точек в геометрии.
insert(index, coordinates)	ILineStringGeometryAccess	Добавляет новую точку с заданным индексом.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable.
remove(index)	Number[]	Удаляет точку с заданным индексом.
set(index, coordinates)	ILineStringGeometryAccess	Задает координаты точки с заданным индексом.
setCoordinates(coordinates)	ILineStringGeometryAccess	Задает массив координат геометрии.
splice(index, number)	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.
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable.

Описание событий

change

```
change
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
index	Integer	✔	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	Тип	*	Описание
index	Integer	✔	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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
anchorPosition	Number[]	✔	Координаты точки, для которой расчитывается 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('coordPosition')).closestPointIndex
    );
});
```

getCoordinates

```
{Number[][]} getCoordinates()
```

Возвращает массив координат геометрии.

getLength

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

Возвращает количество точек в геометрии.

insert

```
{ILineStringGeometryAccess} insert(index, coordinates)
```

Добавляет новую точку с заданным индексом.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс точки.
coordinates	Number[]	✔	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('coordPosition'))
});
```

remove

```
{Number[]} remove(index)
```

Удаляет точку с заданным индексом.

Возвращает координаты удаленной точки.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс точки.

* Обязательный параметр/опция.

set

```
{ILineStringGeometryAccess} set(index, coordinates)
```

Задает координаты точки с заданным индексом.

Возвращает ссылка на себя.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс точки.
coordinates	Number[]	✔	Координаты точки.

* Обязательный параметр/опция.

setCoordinates

```
{ILineStringGeometryAccess} setCoordinates(coordinates)
```

Задает массив координат геометрии.

Возвращает ссылка на себя.

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[][]	✔	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	Тип	*	Описание
index	Integer	✔	Индекс, начиная с которого производится удаление и добавление точек.
number	Integer	✔	Количество удаляемых точек.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
// добавляем новую точку в начало линии по клику на карте
myMap.events.add('click', function (e) {
    myLineString.splice(0, 0, myLineString.getLength(), e.get('coordPosition'))
});
```

IMapAction

Расширяет **IEventEmitter**.

Интерфейс объекта, управляющего движением карты.

Конструктор | Поля | События | Методы

Конструктор

```
IMapAction()
```

Поля

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

События

Имя	Описание
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.

Методы

Имя	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.
end()	Останавливает движение.

Описание событий

end

```
end
```

Событие, оповещающее карту об окончании движения.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

tick

```
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	Тип	*	Описание
mapActionManager	**map.action.Manager**	✔	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
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
add	Произошло добавление дочернего объекта. Унаследовано от ICollection.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IParentOnMap.
optionschange	Изменение в опциях объекта. Унаследовано от ICustomizable.
remove	Произошло удаление дочернего объекта. Унаследовано от ICollection.

Методы

Имя	Возвращает	Описание
add(object)	ICollection	Добавляет в коллекцию дочерний объект. Унаследован от ICollection.
getIterator()	IIterator	Возвращает итератор по коллекции. Унаследован от ICollection.
getMap()	Map	Возвращает ссылку на карту. Унаследован от IParentOnMap.
remove(object)	ICollection	Удаляет из коллекции дочерний объект. Унаследован от ICollection.

IMapState

Объект-описание состояния карты. Объекты с таким интерфейсом возвращаются сервисом загрузки YMapsML.

См.: geoXml.load

Конструктор | Методы

Конструктор

```
IMapState()
```

См.: geoXml.load

Методы

Имя	Возвращает	Описание
applyToMap(map)	util.Promise	Применяет состояние к переданной карте.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

applyToMap

```
{util.Promise} applyToMap(map)
```

Применяет состояние к переданной карте.

Возвращает объект-promise.

Параметры:

Параметр	Тип	*	Описание
map	Map	✔	Карта.

* Обязательный параметр/опция.

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
events	IEventManager	Менеджер событий объекта. Унаследовано от IFreezable.

События

Имя	Описание
change	Изменение опций.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.

Методы

Имя	Возвращает	Описание
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
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.
getAll()	Object	Возвращает ссылку на внутренний хэш хранящий значения опций.
getName()	String	Возвращает имя менеджера опций.
getNative(key)	Object	Возвращает значение заданной опции, определенное на данном уровне иерархии опций, т.е. в данном менеджере.
getParent()	IOptionManager|null	Возвращает родительский менеджер опций.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable.
resolve(key, name)	Object	Метод, предназначенный для вызова дочерними менеджерами опций.
setName(name)		Задает имя менеджера опций.
setParent(parent)	IChild	Задает родительский менеджер опций.
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable.

Описание событий

change

```
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.

Возвращает значение заданной опции, либо defaultValue, если значение опции не определено.

Параметры:

Параметр	Тип	*	Описание
key	String	✔	Имя опции.
defaultValue	Object		Значение по умолчанию.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
key	String	✔	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	Тип	*	Описание
key	String	✔	Имя опции.
name	String		Имя дочернего менеджера опций.

* Обязательный параметр/опция.

setName

```
setName(name)
```

Задает имя менеджера опций.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
name	String	✔	Имя менеджера опций.

* Обязательный параметр/опция.

setParent

```
{IChild} setParent(parent)
```

Задает родительский менеджер опций.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
parent	IOptionManager|null	✔	Родительский менеджер опций.

* Обязательный параметр/опция.

IOverlay

Расширяет ICustomizable, IDomEventEmitter.

Интерфейс оверлея.

Конструктор | Поля | События | Методы

Конструктор

```
IOverlay()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
datachange	Изменение данных. Поля данных: • oldData — старые данные; • newData — новые данные.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
geometrychange	Изменение геометрии. Поля данных: • oldGeometry — старая пиксельная геометрия; • newGeometry — новая пиксельная геометрия.
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
optionschange	Изменение в опциях объекта. Унаследовано от **ICustomizable**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
getData()	Object	Возвращает объект данных макета.
getGeometry()	IPixelGeometry	Возвращает текущую пиксельную геометрию.
getMap()	Map|null	Возвращает ссылку на текущую карту.
setData(data)		Задает данные оверлея.
setGeometry(geometry)		Задает пиксельную геометрию оверлея.
setMap(map)		Задает карту, на которой должен отобразиться оверлей.

Описание событий

datachange

```
datachange
```

Изменение данных. Поля данных:

- oldData — старые данные;
- newData — новые данные.

geometrychange

```
geometrychange
```

Изменение геометрии. Поля данных:

- oldGeometry — старая пиксельная геометрия;
- newGeometry — новая пиксельная геометрия.

mapchange

```
mapchange
```

Сменилась карта. Поля данных:

- oldMap — старая карта;
- newMap — новая карта.

Описание методов

getData

```
{Object} getData()
```

Возвращает объект данных макета.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

getGeometry

```
{IPixelGeometry} getGeometry()
```

Возвращает текущую пиксельную геометрию.

getMap

```
{Map|null} getMap()
```

Возвращает ссылку на текущую карту.

setData

```
setData(data)
```

Задает данные оверлея.

Параметры:

Параметр	Тип	*	Описание
data	Object	✔	Данные оверлея.

* Обязательный параметр/опция.

setGeometry

```
setGeometry(geometry)
```

Задает пиксельную геометрию оверлея.

Параметры:

Параметр	Тип	*	Описание
geometry	IPixelGeometry	✔	Геометрия в глобальных пиксельных координатах.

* Обязательный параметр/опция.

setMap

```
setMap(map)
```

Задает карту, на которой должен отобразиться оверлей.

Параметры:

Параметр	Тип	*	Описание
map	Map|null	✔	Ссылка на карту.

* Обязательный параметр/опция.

IOverlayFactory

Интерфейс фабрики оверлеев.

Конструктор | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
IOverlayFactory()
```

Методы

Имя	Возвращает	Описание
createOverlay(pixelGeometry, data, options)	IOverlay	Создает и возвращает экземпляр оверлея на основе типа пиксельной геометрии.

Описание методов

createOverlay

```
{IOverlay} createOverlay(pixelGeometry, data, options)
```

Создает и возвращает экземпляр оверлея на основе типа пиксельной геометрии.

Возвращает оверлей.

Параметры:

Параметр	Тип	*	Описание
pixelGeometry	IPixelGeometry	✔	Пиксельная геометрия оверлея.
data	Object		Данные оверлея.
options	Object		Опции оверлея.

* Обязательный параметр/опция.

IPane

Расширяет IEventEmitter, IPositioningContext.

Контейнер, позволяющий объектам карты отображаться и реагировать на движения карты. Позволяет позиционировать внутри своего HTML-элемента дочерние объекты.

Конструктор | Поля | События | Методы

Конструктор

```
IPane(map)
```

Параметры:

Параметр	Тип	*	Описание
map	Map	✔	Карта.

* Обязательный параметр/опция.

Поля

Имя	Тип	Описание
events	IEventManager	Менеджер событий.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
		Унаследовано от **IEventEmitter**.

События

Имя	Описание
actionbegin	Событие начала движения контейнера (если контейнер может двигаться).
actionend	Событие окончания движения контейнера (если контейнер может двигаться).
viewportchange	Событие изменения области показа объектов контейнера.
zoomchange	Событие изменения уровня масштабирования.

Методы

Имя	Возвращает	Описание
fromClientPixels(clientPixelPoint)	Number[]	Преобразует клиентские пиксельные координаты в глобальные. Унаследован от **IPositioningContext**.
getElement()	HTMLElement	Возвращает DOM-элемент контейнера.
getViewport()	Number[][]	Возвращает область показа объектов в клиентских координатах.
getZoom()	Number	Возвращает уровень масштабирования.
toClientPixels(globalPixelPoint)	Number[]	Преобразует глобальные пиксельные координаты в клиентские. Унаследован от **IPositioningContext**.

Описание событий

actionbegin

```
actionbegin
```

Событие начала движения контейнера (если контейнер может двигаться).

actionend

```
actionend
```

Событие окончания движения контейнера (если контейнер может двигаться).

viewportchange

```
viewportchange
```

Событие изменения области показа объектов контейнера.

zoomchange

```
zoomchange
```

Событие изменения уровня масштабирования.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

getElement

```
{HTMLElement} getElement()
```

Возвращает DOM-элемент контейнера.

getViewport

```
{Number[][]} getViewport()
```

Возвращает область показа объектов в клиентских координатах.

getZoom

```
{Number} getZoom()
```

Возвращает уровень масштабирования.

IParentOnMap

Интерфейс родительского объекта, относящегося к определенному объекту карты.

Конструктор | События | Методы

Конструктор

```
IParentOnMap()
```

События

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта.

Методы

Имя	Возвращает	Описание
getMap()	**Map**	Возвращает ссылку на карту.

Описание событий

mapchange

```
mapchange
```

Сменилась карта. Поля данных:

- oldMap — старая карта;
- newMap — новая карта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

getMap

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

Возвращает ссылку на карту.

IPixelCircleGeometry

Расширяет **IPixelGeometry**.

Интерфейс пиксельной геометрии "Круг".

Конструктор | Поля | Методы

Конструктор

```
IPixelCircleGeometry(coordinates, metaData)
```

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[]	✔	Координаты центра круга.
metaData	Object		метаданные

* Обязательный параметр/опция.

Поля

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

Методы

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**.
getCoordinates()	Number[]	Возвращает координаты центра круга.
getMetaData()	Object	Возвращает метаданные пиксельной геометрии. Унаследован от **IPixelGeometry**.
getRadius()	Number	Возвращает радиус круга.
getType()	String	Возвращает идентификатор типа геометрии.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		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**.

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
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.

Методы

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**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getMetaData()	Object	Возвращает метаданные пиксельной геометрии.
getType()	String	Возвращает идентификатор типа геометрии. Унаследован от IBaseGeometry.
scale(factor)	IPixelGeometry	Создает отмасштабированную копию геометрии.
shift(offset)	IPixelGeometry	Создает копию геометрии сдвинутую на заданную величину.

Описание методов

getMetaData

```
{Object} getMetaData()
```

Возвращает метаданные пиксельной геометрии.

scale

```
{IPixelGeometry} scale(factor)
```

Создает отмасштабированную копию геометрии.

Возвращает отмасштабированную копию геометрии.

Параметры:

Параметр	Тип	*	Описание
factor	Number	✔	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	Тип	*	Описание
offset	Number[]	✔	Величина сдвига по осям.

* Обязательный параметр/опция.

Пример:

```
// Сдвигаем все координаты геометрии на 200 пикселей влево
var shifted = myPixelGeometry.shift([-200, 0]);
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

IPixelLineStringGeometry

Расширяет **IPixelGeometry**.

Интерфейс пиксельной геометрии "Ломаная линия".

Конструктор | Поля | Методы

Конструктор

```
IPixelLineStringGeometry(coordinates, metaData)
```

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[][]	✔	Координаты линии.
metaData	Object		метаданные

* Обязательный параметр/опция.

Поля

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

Методы

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**.
getClosest(anchorPosition)	Object	Ищет на ломаной линии точку, ближайшую к anchorPosition.
getCoordinates()	Number[][]	Возвращает координаты линии.
getLength()	Integer	Возвращает количество точек в геометрии.
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **IPixelGeometry**.

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	Тип	*	Описание
anchorPosition	Number[]	✔	Координаты точки, для которой расчитывается ближайшая точка на ломаной.

* Обязательный параметр/опция.

getCoordinates

```
{Number[][]} getCoordinates()
```

Возвращает координаты линии.

getLength

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

Возвращает количество точек в геометрии.

IPixelPointGeometry

Расширяет **IPixelGeometry**.

Интерфейс пиксельной геометрии "Точка".

Конструктор | Поля | Методы

Конструктор

```
IPixelPointGeometry(coordinates, metaData)
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[]	✔	Координаты точки.
metaData	Object		метаданные

* Обязательный параметр/опция.

Поля

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

Методы

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.
getCoordinates()	Number[]	Возвращает координаты точки.
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.

Описание методов

getCoordinates

```
{Number[]} getCoordinates()
```

Возвращает координаты точки.

IPixelPolygonGeometry

Расширяет IPixelGeometry.

Интерфейс пиксельной геометрии "Многоугольник".

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор | Поля | Методы

Конструктор

```
IPixelPolygonGeometry(coordinates, fillRule, metaData)
```

Параметры:

Параметр	Тип	*	Описание
`coordinates`	Number[][][]	✔	Координаты многоугольника.
`fillRule`	String	✔	Строковый идентификатор, определяющий алгоритм 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.
`metaData`	Object		метаданные

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

* 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.
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.
getCoordinates()	Number[][][]	Возвращает координаты многоугольника.
getFillRule()	String	Возвращает идентификатор алгоритма заливки многоугольника.
getLength()	Integer	Возвращает количество контуров в многоугольнике.
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.

Описание методов

contains

```
{Boolean} contains(position)
```

Проверяет лежит ли переданная точка внутри многоугольника.

Возвращает признак принадлежности точки многоугольнику.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
position	Number[]	✔	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	Тип	*	Описание
anchorPosition	Number[]	✔	Координаты точки, для которой расчитывается ближайшая точка на контуре многоугольника.

* Обязательный параметр/опция.

getCoordinates

```
{Number[][][]} getCoordinates()
```

Возвращает координаты многоугольника.

getFillRule

```
{String} getFillRule()
```

Возвращает идентификатор алгоритма заливки многоугольника.

getLength

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

Возвращает количество контуров в многоугольнике.

IPixelRectangleGeometry

Расширяет **IPixelGeometry**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Интерфейс пиксельной геометрии "Прямоугольник".

Конструктор | Поля | Методы

Конструктор

```
IPixelRectangleGeometry(coordinates, metaData)
```

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[][]	✔	Координаты двух противоположных углов прямоугольника.
metaData	Object		метаданные

* Обязательный параметр/опция.

Поля

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

Методы

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**.
getClosest(anchorPosition)	Object	Ищет на прямоугольнике точку, ближайшую к anchorPosition.
getCoordinates()	Number[][]	Возвращает координаты двух противоположных углов прямоугольника.
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**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
anchorPosition	Number[]	✔	Координаты точки, для которой расчитывается 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
events	**IEventManager**	Менеджер событий. Унаследовано от **IEventEmitter**.
options	**IOptionManager**	Менеджер опций. Унаследовано от **ICustomizable**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

События

Имя	Описание
change	Изменение координат. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты. Унаследовано от IPointGeometryAccess.
mapchange	Сменилась карта. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия IPixelGeometry; • newPixelGeometry — новая пиксельная геометрия IPixelGeometry. Унаследовано от IGeometry.

Методы

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.
getCoordinates()	Number[]|null	Возвращает координаты точки. Унаследован от IPointGeometryAccess.
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".
setCoordinates(coordinates)	IPointGeometryAccess	Задает координаты точки. Унаследован от IPointGeometryAccess.
setMap(map)		Задает карту. Унаследован от IGeometry.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

getType

```
{String} getType()
```

Возвращает строку "Point".

IPointGeometryAccess

Интерфейс доступа к геометрии "Точка".

Конструктор | События | Методы

Конструктор

```
IPointGeometryAccess()
```

События

Имя	Описание
change	Изменение координат. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты; • newCoordinates — новые координаты.

Методы

Имя	Возвращает	Описание
getCoordinates()	Number[]|null	Возвращает координаты точки.
setCoordinates(coordinates)	**IPointGeometryAccess**	Задает координаты точки.

Описание событий

change

```
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

setCoordinates

```
{IPointGeometryAccess} setCoordinates(coordinates)
```

Задает координаты точки.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[]|null	✔	Координаты точки.

* Обязательный параметр/опция.

IPolygonGeometry

Расширяет **IGeometry**, **IPolygonGeometryAccess**.

Интерфейс геометрии "Многоугольник".

Конструктор | Поля | События | Методы

Конструктор

```
IPolygonGeometry()
```

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: • oldCoordinates — старые координаты; • newCoordinates — новые координаты; • oldFillRule — старый алгоритм заливки; • newFillRule — новые алгоритм заливки. Унаследовано от **IPolygonGeometryAccess**.
mapchange	Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — новая карта. Унаследовано от **IGeometry**.
optionschange	Изменение в опциях объекта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **ICustomizable**.
pixelgeometrychange	Изменилась пиксельная геометрия. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**; • newPixelGeometry — новая пиксельная геометрия **IPixelGeometry**. Унаследовано от **IGeometry**.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри многоугольника. Унаследован от **IPolygonGeometryAccess**.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.
get(index)	Number[][]	Возвращает координаты контура с заданным индексом. Унаследован от **IPolygonGeometryAccess**.
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**.
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**.
getMap()	**Map**|null	Возвращает текущую карту. Унаследован от **IGeometry**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getPixelGeometry(options)	IPixelGeometry	Возвращает пиксельную геометрию, соответствующую данной геометрии, ее опциям и состоянию карты. Унаследован от IGeometry.
getType()	String	Возвращает строку "Polygon".
insert(index, path)	IPolygonGeometryAccess	Добавляет новый контур с заданным индексом. Унаследован от IPolygonGeometryAccess.
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	Задает координаты контура с заданным индексом. Унаследован от IPolygonGeometryAccess.
setCoordinates(coordinates)	IPolygonGeometryAccess	Задает массив координат геометрии. Унаследован от IPolygonGeometryAccess.
setFillRule(fillRule)	IPolygonGeometryAccess	Задает алгоритм заливки многоугольника. Унаследован от IPolygonGeometryAccess.
setMap(map)		Задает карту. Унаследован от IGeometry.
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 методов

getType

```
{String} getType()
```

Возвращает строку "Polygon".

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

IPolygonGeometryAccess

Расширяет **IFreezable**.

Интерфейс доступа к геометрии "Многоугольник".

Конструктор | Поля | События | Методы

Конструктор

```
IPolygonGeometryAccess()
```

Поля

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

События

Имя	Описание
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.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри многоугольника.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим. Унаследован от **IFreezable**.
get(index)	Number[][]	Возвращает координаты контура с заданным индексом.
getChildGeometry(index)	**ILinearRingGeometryAccess**	Создает и возвращает объект **ILinearRingGeometryAccess** для заданного контура.
getClosest(anchorPosition)	Object	Ищет на контуре многоугольника точку, ближайшую к anchorPosition.
getCoordinates()	Number[][][]	Возвращает массив координат геометрии.
getFillRule()	String	Возвращает идентификатор алгоритма заливки.
getLength()	Integer	Возвращает количество контуров в геометрии.
insert(index, path)	**IPolygonGeometryAccess**	Добавляет новый контур с заданным индексом.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **IFreezable**.
remove(index)	**ILinearRingGeometryAccess**	Удаляет контур с заданным индексом.
set(index, path)	**IPolygonGeometryAccess**	Задает координаты контура с заданным индексом.
setCoordinates(coordinates)	**IPolygonGeometryAccess**	Задает массив координат геометрии.
setFillRule(fillRule)	**IPolygonGeometryAccess**	Задает алгоритм заливки многоугольника.
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.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

Описание событий

change

```
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	Тип	*	Описание
position	Number[]	✔	Координаты точки.

* Обязательный параметр/опция.

Пример:

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

```
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.setMap(myMap);

// Проверка, входит ли точка клика в полигон, с заданной выше геометрией.
myMap.events.add('click', function (e) {
    alert(myPolygon.contains(e.get('coordPosition')) ? 'Ранил!' : 'Мимо!');
});
```

get

```
{Number[][]} get(index)
```

Возвращает координаты контура с заданным индексом.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	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	Тип	*	Описание
index	Integer	✔	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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
anchorPosition	Number[]	✔	Координаты точки, для которой расчитывается 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]]
    ]),
    myPoint = new ymaps.Placemark([74, 52]),
    // Построим кратчайшую нормаль от контура полигона до точки:
    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",
// теперь точка входит в полигон.
myPolygon.geometry.setFillRule('nonZero');
alert(myPolygon.geometry.contains(middlePoint)); // => true
```

getLength

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


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Возвращает количество контуров в геометрии.

insert

```
{IPolygonGeometryAccess} insert(index, path)
```

Добавляет новый контур с заданным индексом.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс контура.
path	Number[][]	✔	Координаты контура.

* Обязательный параметр/опция.

remove

```
{ILinearRingGeometryAccess} remove(index)
```

Удаляет контур с заданным индексом.

Возвращает удаленный контур.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс контура.

* Обязательный параметр/опция.

set

```
{IPolygonGeometryAccess} set(index, path)
```

Задает координаты контура с заданным индексом.

Возвращает ссылка на себя.

Параметры:

Параметр	Тип	*	Описание
index	Integer	✔	Индекс контура.
path	Number[][]	✔	Координаты контура.

* Обязательный параметр/опция.

setCoordinates

```
{IPolygonGeometryAccess} setCoordinates(coordinates)
```

Задает массив координат геометрии.

Возвращает ссылка на себя.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[][][]	✔	Координаты геометрии.

* Обязательный параметр/опция.

setFillRule

```
{IPolygonGeometryAccess} setFillRule(fillRule)
```

Задает алгоритм заливки многоугольника.

Возвращает ссылку на себя.

Параметры:

Параметр	Тип	*	Описание
fillRule	String	✔	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	Тип	*	Описание
index	Integer	✔	Индекс, начиная с которого производится удаление и добавление контуров.
number	Integer	✔	Количество удаляемых контуров.

* Обязательный параметр/опция.

IPositioningContext

Интерфейс контекста позиционирования — объекта, позволяющего позиционировать внутри себя объект, заданный глобальными пиксельными координатами.

Конструктор | Методы

Конструктор

```
IPositioningContext()
```


Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Методы

Имя	Возвращает	Описание
fromClientPixels(clientPixelPoint)	Number[]	Преобразует клиентские пиксельные координаты в глобальные.
getZoom()	Number	Возвращает текущий уровень масштабирования, в котором работает контекст позиционирования.
toClientPixels(globalPixelPoint)	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	Тип	*	Описание
clientPixelPoint	Number[]	✔	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	Тип	*	Описание
globalPixelPoint	Number[]	✔	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).

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор | Методы

Конструктор

```
IProjection()
```

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. Необязательный метод. В случае его отсутствия будет использована координатная система **coordSystem.cartesian**.
isCycled()	Boolean[]	Показатель зацикленности проекции.
toGlobalPixels(coordPoint, zoom)	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	Тип	*	Описание
globalPixelPoint	Number[]	✔	Точка в пиксельных координатах.
zoom	Number	✔	Уровень масштабирования.

* Обязательный параметр/опция.

getCoordSystem

```
{ICoordSystem} getCoordSystem()
```

Возвращает координатную систему, используемую проекцией. Необязательный метод. В случае его отсутствия будет использована координатная система **coordSystem.cartesian**.

Возвращает координатную систему.

isCycled

```
{Boolean[]} isCycled()
```

Показатель зацикленности проекции.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

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	Тип	*	Описание
coordPoint	Number[]	✔	Точка в координатах проекции.
zoom	Number	✔	Уровень масштабирования.

* Обязательный параметр/опция.

IPromiseProvider

Объект, выполняющий "обещание".

Конструктор | Методы

Конструктор

```
IPromiseProvider()
```

Методы

Имя	Возвращает	Описание
then(onResolve, onReject)	**IPromiseProvider**	Возвращает объект-обещание.

Описание методов

then

```
{IPromiseProvider} then(onResolve, onReject)
```

Возвращает объект-обещание.

Возвращает ссылку на себя или новый объект-обещание.

Параметры:

Параметр	Тип	*	Описание
onResolve	Function	✔	Функция-обработчик, которая вызывается, если обещание выполнено.
onReject	Function	✔	Функция-обработчик, которая вызывается, если обещание не выполнено (произошла ошибка).

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

IRectangleGeometry

Расширяет **IGeometry**, **IRectangleGeometryAccess**.

Интерфейс геометрии "Прямоугольник".

Конструктор | Поля | События | Методы

Конструктор

```
IRectangleGeometry()
```

Поля

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

События

Имя	Описание
change	Изменение координат углов. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты углов; • newCoordinates — новые координаты углов. Унаследовано от **IRectangleGeometryAccess**.
mapchange	Сменилась карта. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldMap — старая карта; • newMap — 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: • oldPixelGeometry — старая пиксельная геометрия **IPixelGeometry**; • newPixelGeometry — новая пиксельная геометрия **IPixelGeometry**. Унаследовано от **IGeometry**.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри прямоугольника. Унаследован от **IRectangleGeometryAccess**.
freeze()	**IFreezable**	Переводит объект в "замороженный" режим.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
		Унаследован от **IFreezable**.
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 контуре прямоугольника точку, ближайшую к anchorPosition. Унаследован от **IRectangleGeometryAccess**.
getCoordinates()	**IRectangleGeometryAccess**	Возвращает координаты двух противоположных углов прямоугольника. Унаследован от **IRectangleGeometryAccess**.
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".
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от **IFreezable**.
setCoordinates(coordinates)	**IRectangleGeometryAccess**	Задает координаты двух противоположных углов прямоугольника. Унаследован от **IRectangleGeometryAccess**.
setMap(map)		Задает карту. Унаследован от **IGeometry**.
unfreeze()	**IFreezable**	Переводит объект в активный режим. Унаследован от **IFreezable**.

Описание методов

getType

```
{String} getType()
```

Возвращает строку "Rectangle".

IRectangleGeometryAccess

Расширяет **IFreezable**.

Интерфейс доступа к геометрии "Прямоугольник".

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор | Поля | События | Методы

Конструктор

```
IRectangleGeometryAccess()
```

Поля

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

События

Имя	Описание
change	Изменение координат углов. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldCoordinates — старые координаты углов; • newCoordinates — новые координаты углов.

Методы

Имя	Возвращает	Описание
contains(position)	Boolean	Проверяет, лежит ли переданная точка внутри прямоугольника.
freeze()	IFreezable	Переводит объект в "замороженный" режим. Унаследован от IFreezable.
getClosest(anchorPosition)	Object	Ищет на контуре прямоугольника точку, ближайшую к anchorPosition.
getCoordinates()	IRectangleGeometryAccess	Возвращает координаты двух противоположных углов прямоугольника.
isFrozen()	Boolean	Возвращает true, если объект находится в "замороженном" режиме, иначе false. Унаследован от IFreezable.
setCoordinates(coordinates)	IRectangleGeometryAccess	Задает координаты двух противоположных углов прямоугольника.
unfreeze()	IFreezable	Переводит объект в активный режим. Унаследован от IFreezable.

Описание событий

change

```
change
```

Изменение координат углов. Экземпляр класса Event. Имена полей, доступных через метод Event.get:

- oldCoordinates — старые координаты углов;
- newCoordinates — новые координаты углов.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Описание методов

contains

```
{Boolean} contains(position)
```

Проверяет, лежит ли переданная точка внутри прямоугольника.

Возвращает признак принадлежности точки прямоугольнику.

Параметры:

Параметр	Тип	*	Описание
position	Number[]	✔	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	Тип	*	Описание
anchorPosition	Number[]	✔	Координаты точки, для которой расчитывается ближайшая точка на прямоугольнике.

* Обязательный параметр/опция.

getCoordinates

```
{IRectangleGeometryAccess} getCoordinates()
```

Возвращает координаты двух противоположных углов прямоугольника.

setCoordinates

```
{IRectangleGeometryAccess} setCoordinates(coordinates)
```

Задает координаты двух противоположных углов прямоугольника.

Возвращает ссылка на себя.

Параметры:

Параметр	Тип	*	Описание
coordinates	Number[][]	✔	Координаты углов.

* Обязательный параметр/опция.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

ISearchControlLayout

Расширяет **IExpandableControlLayout**.

Интерфейс макета элемента управления "Поиск по карте".

Конструктор | Поля | События | Методы

Конструктор

```
ISearchControlLayout()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от **ILayout**.
close	Событие, инициирующее скрытие панели с результатами поиска.
collapse	Событие, инициирующее сворачивание объекта. Унаследовано от **IExpandableControlLayout**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от **ILayout**.
expand	Событие, инициирующее разворачивание объекта.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IExpandableControlLayout**.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
open	Событие, инициирующее показ панели с результатами поиска.
pagechange	Событие, инициирующее смену страницы с результатами поиска. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • pageNumber — номер страницы, которую нужно показать.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от **ILayout**.
search	Событие, инициирующее поиск объектов. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • request — строка, содержащая запрос.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
showresult	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.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от ILayout.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от ILayout.
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от ILayout.
getData()	Object	Возвращает объект данных макета. Унаследован от ILayout.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от ILayout.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от ILayout.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от ILayout.

Описание событий

close

```
close
```

Событие, инициирующее скрытие панели с результатами поиска.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

open

```
open
```

Событие, инициирующее показ панели с результатами поиска.

pagechange

```
pagechange
```

Событие, инициирующее смену страницы с результатами поиска. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- pageNumber — номер страницы, которую нужно показать.

search

```
search
```

Событие, инициирующее поиск объектов. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get:

- request — строка, содержащая запрос.

showresult

```
showresult
```

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.

ISelectableControl

Расширяет **IControl**.

Интерфейс переключаемого и выбираемого элемента управления.

Конструктор | Поля | События | Методы

Конструктор

```
ISelectableControl()
```

Поля

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

События

Имя	Описание
deselect	Элемент управления не выбран.
disable	Элемент управления стал недоступным.
enable	Элемент управления стал доступным.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mapchange	Сменилась карта. Поля данных: • oldMap — старая карта; • newMap — новая карта. Унаследовано от IChildOnMap.
parentchange	Сменился родительский объект. Поля данных: • oldParent — старый родитель; • newParent — новый родитель. Унаследовано от IChild.
select	Элемент управления выбран.

Методы

Имя	Возвращает	Описание
deselect()		Отменяет выбор элемента управления (выключает).
disable()		Делает элемент управления недоступным (запрещены пользовательские действия).
enable()		Делает элемент управления доступным (разрешены пользовательские действия).
getParent()	IControlParent	Возвращает ссылку на родительский объект. Унаследован от IControl.
isEnabled()	Boolean	Возвращает true — элемент управления доступен, false — недоступен.
isSelected()	Boolean	Возвращает true — элемент управления выбран, false — не выбран.
select()		Выбирает (включает) элемент управления.
setParent(parent)	IChildOnMap	Устанавливает родительский объект. Унаследован от IControl.

Описание событий

deselect

```
deselect
```

Элемент управления не выбран.

disable

```
disable
```

Элемент управления стал недоступным.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

enable

```
enable
```

Элемент управления стал доступным.

select

```
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

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
ISelectableControlLayout()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от ILayout.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
deselect	Событие, инициирующее выход элемента из состояния "выбран".
emptinesschange	Изменение признака пустоты макета. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от ILayout.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
	Унаследовано от **IDomEventEmitter**.
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
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. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от **ILayout**.
select	Событие, инициирующее переход элемента в состояние "выбран".
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от **ILayout**.
getData()	Object	Возвращает объект данных макета. Унаследован от **ILayout**.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от **ILayout**.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от **ILayout**.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от **ILayout**.

Описание событий

deselect

```
deselect
```

Событие, инициирующее выход элемента из состояния "выбран".

select

```
select
```

Событие, инициирующее переход элемента в состояние "выбран".

ITrafficControlLayout

Расширяет **IExpandableControlLayout**.

Интерфейс макета элемента управления "Панель управления пробками".

Конструктор | Поля | События | Методы

Конструктор

```
ITrafficControlLayout()
```

Поля

Имя	Тип	Описание
events	**IEventManager**	Менеджер событий.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Тип	Описание
		Унаследовано от **IDomEventEmitter**.

События

Имя	Описание
click	Однократное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от **ILayout**.
collapse	Событие, инициирующее сворачивание объекта. Унаследовано от **IExpandableControlLayout**.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от **ILayout**.
expand	Событие, инициирующее разворачивание объекта. Унаследовано от **IExpandableControlLayout**.
hide	Событие, инициирующее удаление пробок с карты. Скрыть пробки.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от **ILayout**.
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.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от ILayout.
getData()	Object	Возвращает объект данных макета. Унаследован от ILayout.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от ILayout.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от ILayout.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от ILayout.

Описание событий

hide

```
hide
```

Событие, инициирующее удаление пробок с карты. Скрыть пробки.

providerkeychange

```
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

```
show
```

Событие, инициирующее показ пробок на карте.

ITrafficProvider

Расширяет ICustomizable, IEventEmitter.

Интерфейс провайдера данных о пробках.

Конструктор | Поля | События | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
ITrafficProvider()
```

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
getMap()	Map|null	Возвращает ссылку на карту.
setMap(Ссылка)		

Описание методов

getMap

```
{Map|null} getMap()
```

Возвращает ссылку на карту.

setMap

```
setMap(Ссылка)
```

Параметры:

Параметр	Тип	*	Описание
Ссылка	Map|null	✔	на карту.

* Обязательный параметр/опция.

IZoomControlLayout

Расширяет ILayout.

Интерфейс макета элемента управления "Ползунок масштаба".

Конструктор | Поля | События | Методы

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Конструктор

```
IZoomControlLayout()
```

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 введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
clientboundschange	Изменение координат верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Это событие не является обязательным; если оно отсутствует, то метод getClientBounds тоже не должен присутствовать. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldClientBounds — старые значения верхней левой и нижней правой точек прямоугольника элемента; • newClientBounds — новые значения верхней левой и нижней правой точек прямоугольника элемента. Унаследовано от ILayout.
contextmenu	Вызов контекстного меню на элементе. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
dblclick	Двойное нажатие левой кнопки мыши на объекте. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
emptinesschange	Изменение признака пустоты макета. Экземпляр класса Event. Имена полей, доступных через метод Event.get: • oldEmptiness — старое значение признака; • newEmptiness — новое значение признака. Унаследовано от ILayout.
mousedown	Нажатие кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseenter	Наведение курсора на объект. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.
mouseleave	Вывод курсора за пределы объекта. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в domEvent.manager. Унаследовано от IDomEventEmitter.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Описание
mousemove	Перемещение курсора над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
mouseup	Отпускание кнопки мыши над объектом. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание 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 с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchmove	Повторящееся событие при мультисенсорном управлении. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
multitouchstart	Начало мультисенсорного управления. Данное событие доступно только на устройствах, которые поддерживают множественные сенсорные прикосновения. Возвращает реализацию интерфейса IMultiTouchEvent с информацией о прикосновениях. Унаследовано от **IDomEventEmitter**.
parentelementchange	Изменение родительского элемента. Экземпляр класса **Event**. Имена полей, доступных через метод Event.get: • oldParentElement — старый родительский элемент; • newParentElement — новый родительский элемент. Унаследовано от **ILayout**.
wheel	Скролл мышкой. При использовании следует иметь введу, что при сенсорном вводе события мыши будут эмулироваться. Более детально описание в **domEvent.manager**. Унаследовано от **IDomEventEmitter**.
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.

Методы

Имя	Возвращает	Описание
destroy()		Деструктор. Вызывается при завершении работы с макетом. Унаследован от **ILayout**.
getClientBoundingRect()	Number[][]	Старое название метода getClientBounds. Оставлено для обратной совместимости. Крайне не рекомендуется одновременно реализовывать оба метода. Унаследован от **ILayout**.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Имя	Возвращает	Описание
getClientBounds()	Number[][]	Возвращает координаты верхней левой и нижней правой точек внешнего прямоугольника элемента по отношению к родительскому элементу. Этот метод не является обязательным; если он отсутствует, то событие clientboundschange тоже не должно присутствовать. Унаследован от ILayout.
getData()	Object	Возвращает объект данных макета. Унаследован от ILayout.
getParentElement()	HTMLElement	Возвращает родительский html-элемент. Унаследован от ILayout.
isEmpty()	Boolean	Возвращает признак того, что 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)		Задает данные макета. Унаследован от ILayout.
setParentElement(parent)		Добавляет макет в DOM-дерево. Унаследован от ILayout.

Описание событий

zoomchange

```
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.

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/

Яндекс

Справочник API Яндекс.Карт

Версия 2.0.31

2.07.2013

Antenna House XSL Formatter (Evaluation) http://www.antennahouse.com/