

# 百度地图 JavaScript API

## v2.0 类参考文档

百度在线网络技术（北京）有限公司

2013 年 08 月 20 日 北京

## 法律说明

版权所有百度在线网络技术有限公司。

本文档包含的所有内容除特别声明之外，版权均属于百度在线网络技术有限公司所有，百度可在不作任何声明的情况下对本文档内容进行修改。

本文档中所使用的商标所有权属于该商标的所有者。

## 文件说明

文件标识:	Baidu_JavaScript API_Reference
当前版本:	2013_08_V2.0
作 者:	Baidu_Map_PM
创建日期:	2013-08-20
最新更新:	

## 更新说明

更新方式	类名	描述
新增	<b>MapStyle</b>	地图样式类，用于设置地图模板和显示元素。
新增	<b>Panorama 等</b>	全景地图类，提供展示全景图、获取检索数据等各种功能。

## 百度在线网络技术有限公司

总机: (+86 10) 59928888

邮箱: [mapapi@baidu.com](mailto:mapapi@baidu.com)

地址: 北京市海淀区上地信息路 9 号甲奎科科技大厦

邮编: 100085

百度地图 API 网址: <http://developer.baidu.com/map/>

百度地图 API 论坛: <http://bbs.lbsyun.baidu.com>

# 目录

核心类 .....	1
Map .....	1
构造函数.....	1
配置方法.....	1
地图状态方法.....	2
修改地图状态方法.....	3
控件方法.....	5
右键菜单方法.....	5
覆盖物方法.....	5
地图图层方法.....	6
坐标变换.....	6
全景方法.....	6
事件.....	7
PanOptions .....	9
属性.....	9
MapOptions.....	9
属性.....	9
MapStyle .....	10
属性.....	10
Viewport.....	10
属性.....	10
ViewportOptions .....	10
属性.....	10
APIVersion.....	11
常量.....	11
基础类 .....	11
Point.....	11
构造函数.....	11
属性.....	12
方法.....	12
Pixel .....	12
构造函数.....	12
属性.....	12
方法.....	12
Bounds .....	13
构造函数.....	13
属性.....	13
方法.....	13
Size.....	14
构造函数.....	14
属性.....	14
方法.....	14

控件类 .....	15
Control .....	15
构造函数.....	15
属性.....	15
方法.....	15
ControlAnchor .....	16
常量.....	16
NavigationControl.....	16
构造函数.....	16
方法.....	16
NavigationControlType.....	17
常量.....	17
NavigationControlOptions .....	17
属性.....	17
OverviewMapControl .....	18
构造函数.....	18
方法.....	18
事件.....	18
OverviewMapControlOptions.....	18
属性.....	18
ScaleControl.....	19
构造函数.....	19
方法.....	19
ScaleControlOptions .....	19
属性.....	19
LengthUnit .....	19
常量.....	20
CopyrightControl .....	20
构造函数.....	20
方法.....	20
Copyright .....	20
属性.....	20
CopyrightControlOptions.....	21
属性.....	21
MapTypeControl .....	21
构造函数.....	21
MapTypeControlOptions.....	21
属性.....	21
MapTypeControlType .....	22
常量.....	22
GeolocationControl.....	22
构造函数.....	22
方法.....	22
事件.....	23

GeolocationControlOptions .....	23
属性.....	23
StatusCode .....	23
StatusCode 常量 .....	23
PanoramaControl .....	24
构造函数.....	24
覆盖物类.....	24
Overlay.....	24
方法.....	24
Marker.....	25
构造函数.....	25
方法.....	25
事件.....	27
MarkerOptions .....	28
属性.....	28
Animation .....	28
常量.....	29
Icon .....	29
构造函数.....	29
属性.....	29
方法.....	30
IconOptions.....	30
属性.....	30
InfoWindow .....	31
构造函数.....	31
方法.....	31
事件.....	32
InfoWindowOptions.....	33
属性.....	33
Label .....	34
构造函数.....	34
方法.....	34
事件.....	35
LabelOptions.....	36
属性.....	36
Polyline .....	36
构造函数.....	36
方法.....	36
事件.....	38
PolylineOptions.....	38
属性.....	38
Polygon .....	39
构造函数.....	39
方法.....	39

事件.....	40
PolygonOptions.....	41
属性.....	41
Circle.....	42
构造函数.....	42
方法.....	42
事件.....	43
CircleOptions .....	44
属性.....	44
Hotspot.....	45
构造函数.....	45
方法.....	45
HotspotOptions .....	45
属性.....	45
MapPanes.....	46
属性.....	46
右键菜单类.....	46
ContextMenu.....	46
构造函数.....	47
方法.....	47
事件.....	47
MenuItem.....	47
构造函数.....	47
方法.....	48
MenuItemOptions .....	48
属性.....	48
地图类型类.....	48
MapType .....	48
构造函数.....	48
方法.....	49
常量.....	49
MapTypeOptions.....	49
属性.....	49
Projection .....	50
方法.....	50
MercatorProjection.....	50
方法.....	50
PerspectiveProjection.....	50
方法.....	50
地图图层类.....	51
TileLayer.....	51
构造函数.....	51
方法.....	51
TileLayerOptions .....	51

属性.....	51
CustomLayer .....	52
构造函数.....	52
事件.....	52
CustomPoi.....	52
属性.....	53
TrafficLayer .....	53
构造函数.....	53
TrafficLayerOptions.....	53
属性.....	54
PredictDate.....	54
属性.....	54
PanoramaCoverageLayer .....	54
构造函数.....	54
全景地图类.....	54
Panorama.....	54
构造函数.....	55
方法.....	55
事件.....	55
PanoramaOptions .....	56
属性.....	56
PanoramaLink .....	56
属性.....	56
PanoramaPov .....	56
属性.....	56
PanoramaService.....	57
构造函数.....	57
方法.....	57
PanoramaData .....	57
属性.....	57
PanoramaTileData.....	58
属性.....	58
服务类 .....	58
LocalSearch.....	58
构造函数.....	58
方法.....	59
LocalSearchOptions .....	60
属性.....	60
CustomData.....	61
属性.....	61
RenderOptions .....	61
属性.....	62
StatusCode .....	62
StatusCode 常量 .....	62

LocalResult .....	63
属性.....	63
方法.....	63
LocalResultPoi .....	64
属性.....	64
PoiType .....	64
常量.....	65
TransitRoute .....	65
构造函数.....	65
方法.....	65
TransitRouteOptions .....	66
属性.....	67
TransitPolicy .....	67
常量.....	67
TransitRouteResult .....	68
属性.....	68
方法.....	68
TransitRoutePlan .....	68
方法.....	68
Line .....	69
属性.....	69
方法.....	69
LineType .....	70
常量.....	70
DrivingRoute .....	70
构造函数.....	70
方法.....	70
DrivingRouteOptions .....	72
属性.....	72
DrivingPolicy .....	72
常量.....	73
DrivingRouteResult .....	73
属性.....	73
方法.....	73
TaxiFare .....	74
属性.....	74
TaxiFareDetail .....	74
属性.....	74
RoutePlan .....	74
方法.....	75
Route .....	75
方法.....	75
RouteType .....	76
常量.....	76



Step .....	76
方法.....	76
WalkingRoute.....	76
构造函数.....	77
方法.....	77
WalkingRouteOptions .....	78
属性.....	78
WalkingRouteResult .....	79
属性.....	79
方法.....	79
Geocoder .....	79
构造函数.....	79
方法.....	80
GeocoderResult .....	80
属性.....	80
AddressComponent .....	80
属性.....	81
LocationOptions.....	81
属性.....	81
LocalCity.....	81
构造函数.....	81
方法.....	81
LocalCityOptions .....	82
属性.....	82
LocalCityResult .....	82
属性.....	82
TrafficControl .....	82
构造函数.....	83
方法.....	83
Geolocation .....	83
构造函数.....	83
方法.....	83
GeolocationResult .....	84
属性.....	84
PositionOptions .....	84
属性.....	84
HighlightModes .....	84
常量.....	84
BusLineSearch .....	85
构造函数.....	85
方法.....	85
BusLineSearchOptions.....	86
属性.....	86
BusListResult .....	87

属性.....	87
方法.....	87
BusLine .....	87
属性.....	87
方法.....	87
BusListItem.....	88
属性.....	88
BusStation .....	88
属性.....	88
Autocomplete .....	88
构造函数.....	88
方法.....	89
事件.....	89
AutocompleteOptions .....	90
属性.....	90
AutocompleteResultPoi .....	90
属性.....	90
AutocompleteResult.....	91
属性.....	91
方法.....	91
Boundary.....	91
构造函数.....	91
方法.....	91

# 核心类

## Map

此类是地图 API 的核心类，用来实例化一个地图。

## 构造函数

构造函数	描述
<code>Map(container:String HTMLElement[, opts:MapOptions])</code>	在指定的容器内创建地图实例，之后需要调用 <code>Map.centerAndZoom()</code> 方法对地图进行初始化。未进行初始化的地图将不能进行任何操作。

## 配置方法

方法	返回值	描述
<code>enableDragging()</code>	none	启用地图拖拽，默认启用。
<code>disableDragging()</code>	none	禁用地图拖拽。
<code>enableScrollWheelZoom()</code>	none	启用滚轮放大缩小，默认禁用。
<code>disableScrollWheelZoom()</code>	none	禁用滚轮放大缩小。
<code>enableDoubleClickZoom()</code>	none	启用双击放大，默认启用。
<code>disableDoubleClickZoom()</code>	none	禁用双击放大。
<code>enableKeyboard()</code>	none	启用键盘操作，默认禁用。键盘的上、下、左、右键可连续移动地图。同时按下其中两个键可使地图进行对角移动。 <code>PgUp</code> 、 <code>PgDn</code> 、 <code>Home</code> 和 <code>End</code> 键会使地图平移其1/2的大小。 <code>+</code> 、 <code>-</code> 键会使地图放大或缩小一级。
<code>disableKeyboard()</code>	none	禁用键盘操作。

<code>enableInertialDragging()</code>	none	启用地图惯性拖拽，默认禁用。 (自 1.1 新增)
<code>disableInertialDragging()</code>	none	禁用地图惯性拖拽。 (自 1.1 新增)
<code>enableContinuousZoom()</code>	none	启用连续缩放效果，默认禁用。 (自 1.1 新增)
<code>disableContinuousZoom()</code>	none	禁用连续缩放效果。 (自 1.1 新增)
<code>enablePinchToZoom()</code>	none	启用双指操作缩放，默认启用。 (自 1.1 新增)
<code>disablePinchToZoom()</code>	none	禁用双指操作缩放。 (自 1.1 新增)
<code>enableAutoResize()</code>	none	启用自动适应容器尺寸变化，默认启用。 (自 1.2 新增)
<code>disableAutoResize()</code>	none	禁用自动适应容器尺寸变化。 (自 1.2 新增)
<code>setDefaultCursor(cursor:String)</code>	none	设置地图默认的鼠标指针样式。参数 <code>cursor</code> 应符合 CSS 的 <code>cursor</code> 属性规范。 (自 1.1 新增)
<code>getDefaultCursor()</code>	String	返回地图默认的鼠标指针样式。 (自 1.1 新增)
<code>setDraggingCursor(cursor:String)</code>	none	设置拖拽地图时的鼠标指针样式。参数 <code>cursor</code> 应符合 CSS 的 <code>cursor</code> 属性规范。 (自 1.1 新增)
<code>getDraggingCursor()</code>	String	返回拖拽地图时的鼠标指针样式。 (自 1.1 新增)
<code>setMinZoom(zoom:Number)</code>	none	设置地图允许的最小级别。取值不得小于地图类型所允许的最小级别。 (自 1.2 新增)
<code>setMaxZoom(zoom:Number)</code>	none	设置地图允许的最大级别。取值不得大于地图类型所允许的最大级别。 (自 1.2 新增)

## 地图状态方法

方法	返	描述
----	---	----

	返回值	
<code>getBounds()</code>	Bounds	返回地图可视区域，以地理坐标表示。
<code>getCenter()</code>	Point	返回地图当前中心点。
<code>getDistance(start:Point, end:Point)</code>	Number	返回两点之间的距离，单位是米。(自 1.1 新增)
<code>getMapType()</code>	MapType	返回地图类型。(自 1.2 新增)
<code>getSize()</code>	Size	返回地图视图的大小，以像素表示。
<code>getViewPort(view: Array&lt;Point&gt;[, viewportOptions: ViewportOptions])</code>	Viewport	根据提供的地理区域或坐标获得最佳的地图视野，返回的对象中包含 <b>center</b> 和 <b>zoom</b> 属性，分别表示地图的中心点和级别。此方法仅返回视野信息，不会将新的中心点和级别做用到当前地图上。(自 1.1 新增)
<code>getZoom()</code>	Number	返回地图当前缩放级别。

## 修改地图状态方法

方法	返回值	描述
<code>centerAndZoom(center:Point, zoom:Number)</code>	none	设初始化地图。 如果 <b>center</b> 类型为字符串时，比如“北京”， <b>zoom</b> 可以忽略，地图将自动根据 <b>center</b> 调整 <b>zoom</b> 级别。
<code>panTo(center:Point[, opts:PanoOptions])</code>	none	将地图的中心点更改为给定的点。如果该点在当前的地图视图中已经可见，则会以平滑动画的方式移动到中心点位置。可以通过配置强制移动过程不使用动画效果。
<code>panBy(x:Number, y:Number[, opts:PanoOptions])</code>	none	将地图在水平位置上移动 <b>x</b> 像素，垂直位置上移动 <b>y</b> 像素。如果指定的像素大于可视区域范围或者在配置中指定没有动画效果，则不执行滑动效

		果。
<code>reset()</code>	no ne	重新设置地图，恢复地图初始化时的中心点和级别。
<code>setCenter(center:Point String)</code>	no ne	设置地图中心点。 <code>center</code> 除了可以为坐标点以外，还支持城市名。
<code>setCurrentCity(city:String)</code>	no ne	<p>设置地图城市，注意当地图初始化时的类型设置为 <code>BMAP_PERSPECTIVE_MAP</code> 时，需要在调用 <code>centerAndZoom</code> 之前调用此方法设置地图所在城市。例如：</p> <pre>var map = new BMap.Map("container", {mapType: BMAP_PERSPECTIVE_MAP}); map.setCurrentCity("北京市"); map.centerAndZoom(new BMap.Point(116.404, 39.915), 18);</pre> <p>注意：初始化的坐标应与您设置的城市对应，否则地图将无法显示。如果地图初始化为 <code>BMAP_NORMAL_MAP</code> 类型，则在调用 <code>setMapType</code> 切换地图类型时也要调用此方法。（自 1.1 新增）</p>
<code>setMapType(mapType:MapTypes)</code>	no ne	设置地图类型。注意，当设置地图类型为 <code>BMAP_PERSPECTIVE_MAP</code> 时，需要调用 <code>map.setCurrentCity</code> 方法设置城市。（自 1.1 新增）
<code>setMapStyle(mapStyle:MapStyle)</code>	no ne	设置地图样式，样式包括地图底图样式和地图要素是否展示两部分。（自 1.5 新增）
<code>setViewport(view:Array&lt;Point&gt; Viewport[, viewportOptions:ViewportOptions])</code>	no ne	根据提供的地理区域或坐标设置地图视野，调整后的视野会保证包含提供的地理区域或坐标。（自 1.1 新增）
<code>zoomTo(zoom:Number)</code>	no ne	（自 1.2 废弃）
<code>setZoom(zoom:Number)</code>	no ne	将视图切换到指定的缩放等级，中心点坐标不变。注意：当有信息窗口在地图上打开时，地图缩放将保证信息窗口所在的坐标位置不动。（自 1.2 新增）
<code>highResolutionEnabled()</code>	Bo ol ea n	是否使用高分辨率底图。仅当 <code>mapOptions.enableHighResolution</code> 属性为 <code>true</code> 且设备支持高分辨率时返回 <code>true</code> 。
<code>zoomIn()</code>	no	放大一级视图。

	ne	
zoomOut()	no ne	缩小一级视图。
addHotspot(hotspot:Hotspot)	no ne	为地图添加热区。(自 1.2 新增)
removeHotspot(hotspot:Hotspot)	no ne	移除某个地图热区。(自 1.2 新增)
clearHotspots()	no ne	清空地图所有热区。(自 1.2 新增)

## 控件方法

方法	返回值	描述
addControl(control:Control)	none	将控件添加到地图，一个控件实例只能向地图中添加一次。
removeControl(control:Control)	none	从地图中移除控件。如果控件从未被添加到地图中，则该移除不起任何作用。
getContainer()	HTMLElement	返回地图的容器元素。当创建用户自定义控件时，需要自行实现 <code>Control.initialize()</code> 方法，并将控件的容器元素添加到地图上，通过此方法可获得地图容器。

## 右键菜单方法

方法	返回值	描述
addContextMenu(menu:ContextMenu)	none	添加右键菜单。
removeContextMenu(menu:ContextMenu)	none	移除右键菜单。

## 覆盖物方法

方法	返回值	描述
addOverlay(overlay:Overlay)	none	将覆盖物添加到地图中，一个覆盖物实例只能向地图中添加一次。
removeOverlay(overlay:Overlay)	none	从地图中移除覆盖物。如果覆盖物从未被添加到地图中，则该移除不起任何作用。

<code>clearOverlays()</code>	none	清除地图上所有覆盖物。
<code>openInfoWindow(infoWnd:InfoWindow, point:Point)</code>	none	在地图上打开信息窗口。
<code>closeInfoWindow()</code>	none	关闭在地图上打开的信息窗口。在标注上打开的信息窗口也可通过此方法进行关闭。
<code>pointToOverlayPixel(point:Point)</code>	Pixel	根据地理坐标获取对应的覆盖物容器的坐标，此方法用于自定义覆盖物。(自 1.1 新增)
<code>overlayPixelToPoint(pixel:Pixel)</code>	Point	根据覆盖物容器的坐标获取对应的地理坐标。(自 1.1 新增)
<code>getInfoWindow()</code>	InfoWindow Null	返回地图上处于打开状态的信息窗的实例。当地图没有打开的信息窗口时，此方法返回 null。(自 1.1 新增)
<code>getOverlays()</code>	Array<Overlay>	返回地图上的所有覆盖物。(自 1.1 新增)
<code>getPanes()</code>	MapPanes	返回地图覆盖物容器列表。(自 1.1 新增)

## 地图图层方法

方法	返回值	描述
<code>addTileLayer(tileLayer:TileLayer CustomLayer)</code>	none	添加一个自定义地图图层。自1.5新增 CustomLayer。
<code>removeTileLayer(tilelayer:TileLayer CustomLayer)</code>	none	移除一个自定义地图图层。自1.5新增 CustomLayer。
<code>getTileLayer(mapType:String)</code>	TileLayer	通过地图类型得到一个地图图层对象。

## 坐标变换

方法	返回值	描述
<code>pixelToPoint(pixel:Pixel)</code>	Point	像素坐标转换为经纬度坐标。
<code>pointToPixel(point:Point)</code>	Pixel	经纬度坐标转换为像素坐标。

## 全景方法

方法	返回值	描述
----	-----	----



getPanorama ()	Panorama	获取与 Map 类绑定的全景实例。 (自 v1.5 新增)
setPanorama(pano:Panorama)	none	将全景实例与 Map 类进行绑定。 (自 v1.5 新增)

## 事件

事件	参数	描述
click	{type, target, point, pixel, overlay}	左键单击地图时触发此事件。 当双击时，产生的事件序列为：click click dblclick (自 1.1 更新)
dblclick	{type, target, pixel, point}	鼠标双击地图时会触发此事件。
rightclick	{type, target, point, pixel, overlay}	右键单击地图时触发此事件。 当双击时，产生的事件序列为：rightclick rightclick rightdblclick (自 1.1 更新)
rightdblclick	{type, target, point, pixel, overlay}	右键双击地图时触发此事件。 (自 1.1 新增)
maptypechange	{type, target}	地图类型发生变化时触发此事件。 (自 1.1 新增)
mousemove	{type, target, point, pixel, overlay}	鼠标在地图区域移动过程中触发此事件。 (自 1.1 新增)
mouseover	{type, target}	鼠标移入地图区域时触发此事件。 (自 1.2 新增)
mouseout	{type, target}	鼠标移出地图区域时触发此事件。 (自 1.2 新增)
movestart	{type, target}	地图移动开始时触发此事件。
moving	{type, target}	地图移动过程中触发此事件。
moveend	{type, target}	地图移动结束时触发此事件。
zoomstart	{type, target}	地图更改缩放级别开始时触发此事件。

t		
zoomend	{type, target}	地图更改缩放级别结束时触发此事件。
addoverlay	{type, target}	当使用 Map.addOverlay()方法向地图中添加单个覆盖物时会触发此事件。
addcontrol	{type, target}	当使用 Map.addControl()方法向地图中添加单个控件时会触发此事件。
removecontrol	{type, target}	当使用 Map.removeControl()方法移除单个控件时会触发此事件。
removeoverlay	{type, target}	当使用 Map.removeOverlay()方法移除单个覆盖物时会触发此事件。
clearoverlays	{type, target}	当使用 Map.clearOverlays()方法一次性移除全部覆盖物时会触发此事件。
dragstart	{type, target, pixel, point}	开始拖拽地图时触发。
dragging	{type, target, pixel, point}	拖拽地图过程中触发。
dragend	{type, target, pixel, point}	停止拖拽地图时触发。
addtilelayer	{type, target}	添加一个自定义地图图层时触发此事件。
removetilelayer	{type, target}	移除一个自定义地图图层时触发此事件。
load	{type, target, pixel, point, zoom}	调用 Map.centerAndZoom()方法时会触发此事件。这表示位置、缩放层级已经确定，但可能还在载入地图图块。
resize	{type, target, size}	地图可视区域大小发生变化时会触发此事件。
hotspotclick	{type, target, hotspots}	点击热区时触发此事件。(自 1.2 新增)
hotspotover	{type, target, hotspots}	鼠标移至热区时触发此事件。(自 1.2 新增)
hotspotout	{type, target, hotspots}	鼠标移出热区时触发此事件。(自 1.2 新增)
tilesloaded	{type, target}	当地图所有图块完成加载时触发此事件。(自 1.2 新增)

## PanOptions

此类是 `panBy` 和 `panTo` 方法的可选参数，没有构造函数，通过对象字面量形式表示。

### 属性

属性	类型	描述
<code>noAnimation</code>	<code>Boolean</code>	是否在平移过程中禁止动画。（自1.2新增）

## MapOptions

此类表示 `Map` 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

### 属性

属性	类型	描述
<code>zoomLevelMin</code>	<code>Number</code>	地图允许展示的最小级别。 (自 1.2 废弃)
<code>zoomLevelMax</code>	<code>Number</code>	地图允许展示的最大级别。 (自 1.2 废弃)
<code>minZoom</code>	<code>Number</code>	地图允许展示的最小级别。 (自 1.2 新增)
<code>maxZoom</code>	<code>Number</code>	地图允许展示的最大级别。 (自 1.2 新增)
<code>mapType</code>	<code>MapType</code>	地图类型，默认为 <code>BMAP_NORMAL_MAP</code> 。 (自 1.1 新增)
<code>enableHighResolution</code>	<code>Boolean</code>	是否启用使用高分辨率地图。在 iPhone4 及其后续设备上，可以通过开启此选项获取更高分辨率的底图，默认不开启。 (自 1.2 新增)
<code>enableAutoResize</code>	<code>Boolean</code>	是否自动适应地图容器变化，默认启用。 (自 1.2 新增)

## MapStyle

该类用于设置地图样式，使用对象字面量形式表示，不可实例化。自 v1.5 新增。

### 属性

属性	类型	描述
features	Array	设置地图上展示的元素种类，支持 point（兴趣点）、road（道路）、river（河流）、land（陆地）、building（建筑物）。
style	string	设置地图底图样式，目前支持 normal（默认样式）、dark（深色样式）、light（浅色样式）三种。
styleId	string	LBS 云端配置的样式 id

## Viewport

此类代表视野，不可实例化，通过对象字面量形式表示。（自 1.1 新增）

### 属性

属性	类型	描述
center	Point	视野中心点。
zoom	Number	视野级别。

## ViewportOptions

此类作为 map.getViewport 与 map.setViewport 方法的可选参数，不可实例化。

### 属性

属性	类型	描述
enableAnimation	Boolean	是否启用动画效果移动地图，默认为 true。当调整后的级别与当前地图级别一致时，将使用动画效果移动地图。（自 1.1 新增）

margins	Array<Number>	视野调整的预留边距，例如： margins: [30, 20, 0, 20] 表示坐标点会限制在上述区域内。（自 1.1 新增）
zoomFactor	Number	地图级别的偏移量，您可以在方法得出的结果上增加一个偏移值。例如 map.setViewport 计算出地图的级别为10，如果 zoomFactor 为-1，则最终的地图级别为9。  (自 1.1 新增)
delay	Number	改变地图视野的延迟执行时间，单位毫秒，默认为200ms。此延时仅针对动画效果有效。  (自 1.1 新增)

## APIVersion

此常量表示 API 版本号，通过字符串进行描述。

## 常量

常量	类型	描述
BMAP_API_VERSION	String	返回当前 API 版本。例如，1.2版本返回字符串“1.2”。(自 1.2 新增)

## 基础类

### Point

此类表示一个地理坐标点。

## 构造函数

构造函数	描述
Point(lng:Number, lat:Number)	以指定的经度和纬度创建一个地理点坐标。

## 属性

属性	类型	描述
lng	Number	地理经度。
lat	Number	地理纬度。

## 方法

方法	返回值	描述
<code>equals(other:Point)</code>	Boolean	判断坐标点是否相等，当且仅当两点的经度和纬度均相等时返回 true。

## Pixel

此类表示地图上的一点，单位为像素。

## 构造函数

构造函数	描述
<code>Pixel(x:Number, y:Number)</code>	创建像素点对象实例。像素坐标的坐标原点为地图区域的左上角。

## 属性

属性	类型	描述
x	Number	x 坐标。
y	Number	y 坐标。

## 方法

方法	返回值	描述
<code>equals(other:Pixel)</code>	Boolean	判断坐标点是否相等，当且仅当两点的 x 坐标和 y 坐标均相等时返回 true。

## Bounds

此类表示地理坐标的矩形区域。

## 构造函数

构造函数	描述
<code>Bounds(minX:Number, minY:Number, maxX:Number, maxY:Number)</code>	创建一个包含所有给定点坐标的矩形区域。 (自 1.2 废弃)
<code>Bounds(sw:Point, ne:Point)</code>	创建一个包含所有给定点坐标的矩形区域。其中 <b>sw</b> 表示矩形区域的西南角，参数 <b>ne</b> 表示矩形区域的东北角。 (自 1.2 新增)

## 属性

属性	类型	描述
<code>minX</code>	<code>Number</code>	矩形左下角的 <b>x</b> 坐标。 (自 1.2 废弃)
<code>minY</code>	<code>Number</code>	矩形左下角的 <b>y</b> 坐标。 (自 1.2 废弃)
<code>maxX</code>	<code>Number</code>	矩形右上角的 <b>x</b> 坐标。 (自 1.2 废弃)
<code>maxY</code>	<code>Number</code>	矩形右上角的 <b>y</b> 坐标。 (自 1.2 废弃)

## 方法

方法	返回值	描述
<code>equals(other:Bounds)</code>	<code>Boolean</code>	当且仅当此矩形中的两点参数都等于其他矩形的两点参数时，返回 <b>true</b> 。
<code>containsPoint(point:Point)</code>	<code>Boolean</code>	如果点的地理坐标位于此矩形内，则返回 <b>true</b> 。
<code>containsBounds(bounds:Bounds)</code>	<code>Boolean</code>	传入的矩形区域完全包含于此矩形区域中，则返回 <b>true</b> 。

<code>intersects(other:Bounds)</code>	<b>Bounds</b>	计算与另一矩形的交集区域。
<code>extend(point:Point)</code>	<b>none</b>	放大此矩形，使其包含给定的点。
<code>getCenter()</code>	<b>Point</b>	返回矩形的中心点。
<code>isEmpty()</code>	<b>Boolean</b>	如果矩形为空，则返回 <b>true</b> 。
<code>getSouthWest()</code>	<b>Point</b>	返回矩形区域的西南角。 (自 1.2 新增)
<code>getNorthEast()</code>	<b>Point</b>	返回矩形区域的东北角。 (自 1.2 新增)
<code>toSpan()</code>	<b>Point</b>	返回矩形区域的跨度。 (自 1.2 新增)

## Size

此类以像素表示一个矩形区域的大小。

## 构造函数

构造函数	描述
<code>Size(width:Number, height:Number)</code>	以指定的宽度和高度创建一个矩形区域大小对象。

## 属性

属性	类型	描述
<code>width</code>	<b>Number</b>	水平方向数值。
<code>height</code>	<b>Number</b>	竖直方向的数值。

## 方法

方法	返回值	描述
<code>equals(other:Size)</code>	<b>Boolean</b>	当且仅当此矩形中的宽度和高度都等于其他矩形的宽度和高度时，返回 <b>true</b> 。



# 控件类

## Control

此类是所有控件的基类，您可以通过此类来实现自定义控件。所有控件均包含 **Control** 类的属性、方法和事件。通过 `Map.addControl()` 方法可将控件添加到地图上。

## 构造函数

构造函数	描述
<code>Control()</code>	创建一个控件原型实例，通过该原型实例可创建自定义控件。

## 属性

属性	类型	描述
<code>defaultAnchor</code>	enum <code>ControlAnchor</code>	控件默认的停靠位置。自定义控件时需要提供此属性，作为控件的默认停靠位置。
<code>defaultOffset</code>	<code>Size</code>	控件默认的位置偏移值。自定义控件时需要提供此属性，作为控件的默认偏移位置。

## 方法

方法	返回值	描述
<code>initialize(map:Map)</code>	<code>HTMLElement</code>	抽象方法。调用 <code>Map.addControl()</code> 方法添加控件时将调用此方法，从而实现该控件的初始化。自定义控件时需要实现此方法，并将元素的 DOM 元素在该方法中返回。DOM 元素需要添加到地图的容器中，使用 <code>map.getContainer()</code> 方法可获得地图容器元素。
<code>setAnchor(anchor:enum ControlAnchor)</code>	<code>none</code>	设置控件停靠的位置。
<code>getAnchor()</code>	enum <code>ControlAnchor</code>	返回控件停靠的位置。

<code>setOffset(offset:Size)</code>	none	设置控件停靠的偏移量。
<code>getOffset()</code>	Size	返回控件停靠的偏移量。
<code>show()</code>	none	显示控件。
<code>hide()</code>	none	隐藏控件。
<code>isVisible()</code>	Boolean	判断控件的可见性。

## ControlAnchor

此常量表示控件的定位。

### 常量

常量	描述
<code>BMAP_ANCHOR_TOP_LEFT</code>	控件将定位到地图的左上角。
<code>BMAP_ANCHOR_TOP_RIGHT</code>	控件将定位到地图的右上角。
<code>BMAP_ANCHOR_BOTTOM_LEFT</code>	控件将定位到地图的左下角。
<code>BMAP_ANCHOR_BOTTOM_RIGHT</code>	控件将定位到地图的右下角。

## NavigationControl

此类表示地图的平移缩放控件，可以对地图进行上下左右四个方向的平移和缩放操作。

### 构造函数

构造函数	描述
<code>NavigationControl([opts:NavigationControlOptions])</code>	创建一个特定样式的地图平移缩放控件。

### 方法

方法	返回值	描述
----	-----	----

<code>getType()</code>	enum <code>NavigationControlType</code>	返回平移缩放控件的类型。
<code>setType(type:enum NavigationControlType)</code>	<code>none</code>	设置平移缩放控件的类型。

## NavigationControlType

此常量表示平移缩放控件的类型

### 常量

常量	描述
<code>BMAP_NAVIGATION_CONTROL_LARGE</code>	标准的平移缩放控件（包括平移、缩放按钮和滑块）。
<code>BMAP_NAVIGATION_CONTROL_SMALL</code>	仅包含平移和缩放按钮。
<code>BMAP_NAVIGATION_CONTROL_PAN</code>	仅包含平移按钮。
<code>BMAP_NAVIGATION_CONTROL_ZOOM</code>	仅包含缩放按钮。

## NavigationControlOptions

此类表示 `NavigationControl` 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

### 属性

属性	类型	描述
<code>anchor</code>	enum <code>ControlAnchor</code>	控件的停靠位置。
<code>offset</code>	<code>Size</code>	控件的水平偏移值。
<code>type</code>	enum <code>NavigationControlType</code>	平移缩放控件的类型。
<code>showZoomInfo</code>	<code>Boolean</code>	是否显示级别提示信息。(自 1.1 新增)

# OverviewMapControl

此类表示缩略地图控件。

## 构造函数

构造函数	描述
<code>OverviewMapControl([opts:OverviewMapControlOptions])</code>	创建一个缩略地图控件实例。

## 方法

方法	返回值	描述
<code>changeView()</code>	none	切换缩略地图控件的开合状态。
<code>setSize(size:Size)</code>	none	设置缩略地图的大小。
<code>getSize()</code>	Size	返回缩略地图的大小。

## 事件

事件	参数	描述
<code>viewchanged</code>	<code>event{type, target, isOpen}</code>	缩略地图开合状态发生变化后触发此事件。
<code>viewchanging</code>	<code>event{type, target}</code>	缩略地图开合状态发生变化过程中触发此事件。

# OverviewMapControlOptions

此类表示 `OverviewMapControl` 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

## 属性

属性	类型	描述
<code>anchor</code>	<code>enum ControlAnchor</code>	控件的停靠位置。
<code>offset</code>	<code>Size</code>	控件的偏移值。

size	Size	缩略地图控件的大小。
isOpen	Boolean	缩略地图添加到地图后的开合状态，默认为关闭。

## ScaleControl

此类表示比例尺控件。

### 构造函数

构造函数	描述
<code>ScaleControl([opts:ScaleControlOptions])</code>	创建一个比例尺控件。

### 方法

方法	返回值	描述
<code>getUnit()</code>	<code>enum LengthUnit</code>	返回比例尺单位制。
<code>setUnit(unit:enum LengthUnit)</code>	<code>none</code>	设置比例尺单位制。

## ScaleControlOptions

此类表示 `ScaleControl` 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

### 属性

属性	类型	描述
<code>anchor</code>	<code>enum ControlAnchor</code>	控件的停靠位置。
<code>offset</code>	<code>Size</code>	控件的偏移值。

## LengthUnit

常量表示长度单位制。

## 常量

常量	描述
BMAP_UNIT_METRIC	公制单位。
BMAP_UNIT_IMPERIAL	英制单位。

## CopyrightControl

此类表示版权控件，您可以在地图上添加自己的版权信息。每一个版权信息需要包含如下内容：版权的唯一标识、版权内容和其适用的区域范围。

## 构造函数

构造函数	描述
CopyrightControl([opts:CopyrightControlOptions])	创建一个版权控件实例。

## 方法

方法	返回值	描述
addCopyright(copyright:Copyright)	none	添加版权信息。
removeCopyright(id)	none	移除版权信息。
getCopyright(id)	Copyright	返回单个版权信息。
getCopyrightCollection()	Array<Copyright>	返回版权信息列表。

## Copyright

此类表示一条版权信息。可作为 CopyrightControl.addCopyright()方法的参数。它没有构造函数，但可通过对象字面量形式表示。

## 属性

属性	类型	描述
----	----	----

id	Number	该版权信息的唯一标识符。
content	String	该版权的文本信息，用于显示在地图上，支持 HTML 内容。
bounds	Bounds	该版权信息所适用的地理区域。

## CopyrightControlOptions

此类表示 CopyrightControl 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

### 属性

属性	类型	描述
anchor	enum ControlAnchor	控件的停靠位置。
offset	Size	控件的偏移值。

## MapTypeControl

此类是负责切换地图类型的控件，此类继承 Control 所有功能。

### 构造函数

构造函数	描述
MapTypeControl([opts:CopyrightControlOptions])	创建地图类型控件。(自 1.1 新增)

## MapTypeControlOptions

此类表示 MapTypeControl 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示（自 1.2 新增）。

### 属性

属性	类型	描述
type	MapTypeContro	控件样式。

	lType	
mapTypes	Array<MapTypes>	控件展示的地图类型，默认为普通图、卫星图、卫星加路网混合图和三维图。通过此属性可配置控件展示的地图类型。

## MapTypeControlType

此常量表示 MapTypeControl 的外观样式（自 1.2 新增）。

### 常量

属性	描述
BMAP_MAPTYPE_CONTROL_HORIZONTAL	按钮水平方式展示，默认采用此类型展示。
BMAP_MAPTYPE_CONTROL_DROPDOWN	按钮呈下拉列表方式展示。

## GeolocationControl

此类是负责进行地图定位的控件，此类继承 Control 所有功能。

### 构造函数

构造函数	描述
GeolocationControl([opts:GeolocationControlOptions])	创建一个用来定位的控件（自 1.4 新增）

### 方法

方法	返回值	描述
location	none	开始进行定位。
getAddressComponent	AddressComponent	返回当前的定位信息。若当前还未定位，则返回 null。



## 事件

事件	参数	描述
locationSuccess	event{point, AddressComponent}	定位成功后触发此事件。
locationError	event(StatusCode)	定位失败后触发此事件。

## GeolocationControlOptions

此类表示 GeolocationControl 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示（自 1.4 新增）。

## 属性

属性	类型	描述
anchor	enum controlAnchor	控件的停靠位置，默认定位到地图的右下角。
offset	Size	控件的水平偏移值。
showAddressBar	Boolean	是否显示定位信息面板。默认显示定位信息面板。
enableAutoLocation	Boolean	添加控件时是否进行定位。默认添加控件时不进行定位。
locationIcon	Icon	可自定义定位中心点的 Icon 样式

## StatusCode

此常量用于描述对象当前状态。

## StatusCode 常量

常量	描述
BMAP_STATUS_PERMISSION_DENIED	没有权限。对应数值“6”。(自 1.1 新增)
BMAP_STATUS_SERVICE_UNAVAILABLE	服务不可用。对应数值“7”。(自 1.1 新增)

BMAP_STATUS_TIMEOUT	超时。对应数值“8”。(自 1.1 新增)
---------------------	-----------------------

## PanoramaControl

此类是负责切换至全景地图的控件，此类继承 **Control** 所有功能。

### 构造函数

构造函数	描述
<code>PanoramaControl()</code>	创建全景控件实例。

## 覆盖物类

### Overlay

覆盖物的抽象基类，所有覆盖物均继承基类的方法。此类不可实例化。

### 方法

方法	返回值	描述
<code>initialize(map:Map)</code>	HTMLElement	抽象方法，用于初始化覆盖物，当调用 <code>map.addOverlay</code> 时，API 将调用此方法。自定义覆盖物时需要实现此方法。自定义覆盖物时需要将覆盖物对应的 HTML 元素返回。(自 1.1 新增)
<code>isVisible()</code>	Boolean	判断覆盖物是否可见。
<code>dispose()</code>	none	释放覆盖物对象所占用的内存。可在覆盖物被移除后调用此方法，此后该覆盖物将不能再次添加到地图上。自1.1版本开始 API 会自动管理内存，您不再需要调用此方法。 (自 1.2 废弃)
<code>draw()</code>	none	抽象方法，当地图状态发生变化时，由系统调用对覆盖物进行绘制。自定义覆盖物需要实现此方法。 (自 1.1 新增)
<code>show()</code>	none	显示覆盖物。对于自定义覆盖物，此方法会自动将 <code>initialize</code> 方法返回的 HTML 元素样式的 <code>display</code> 属性设置为空。 (自 1.1 更新)

hide()	none	隐藏覆盖物。对于自定义覆盖物，此方法会自动将 initialize 方法返回的 HTML 元素样式的 display 属性设置为 none。  (自 1.1 更新)
--------	------	--

## Marker

此类表示地图上一个图像标注。

## 构造函数

构造函数	描述
Marker(point:Point[, opts:MarkerOptions])	创建一个图像标注实例。point 参数指定了图像标注所在的地理位置。

## 方法

方法	返回值	描述
openInfoWindow(infoWnd:InfoWindow)	none	打开信息窗。
closeInfoWindow()	none	关闭信息窗。
setIcon(icon:Icon)	none	设置标注所用的图标对象。
getIcon()	Icon	返回标注所用的图标对象。
setPoint(point:Point)	none	设置标注的地理坐标。  (自 1.2 废弃)
getPoint()	Point	返回标注的地理坐标。  (自 1.2 废弃)
setPosition(position:Point)	none	设置标注的地理坐标。  (自 1.2 新增)
getPosition()	Point	返回标注的地理坐标。  (自 1.2 新增)

<code>setOffset(offset:Size)</code>	none	设置标注的偏移值。
<code>getOffset()</code>	Size	返回标注的偏移值。
<code>getLabel()</code>	Label	返回标注的文本标注。
<code>setLabel(label:Label)</code>	none	为标注添加文本标注。
<code>setTitle(title:String)</code>	none	设置标注的标题，当鼠标移至标注上时显示此标题。
<code>getTitle()</code>	String	返回标注的标题。
<code>setTop(isTop:Boolean)</code>	none	将标注置于其他标注之上。默认情况下，纬度较低的标注会覆盖在纬度较高的标注之上，从而形成一种立体效果。通过此方法可使某个标注覆盖在其他所有标注之上。注意：如果在多个标注对象上调用此方法，则这些标注依旧按照纬度产生默认的覆盖效果。
<code>enableDragging()</code>	none	开启标注拖拽功能。 (自 1.1 新增)
<code>disableDragging()</code>	none	关闭标注拖拽功能。 (自 1.1 新增)
<code>enableMassClear()</code>	none	允许覆盖物在 <code>map.clearOverlays</code> 方法中被清除。 (自 1.1 新增)
<code>disableMassClear()</code>	none	禁止覆盖物在 <code>map.clearOverlays</code> 方法中被清除。 (自 1.1 新增)
<code>setZIndex(zIndex:Number)</code>	none	设置覆盖物的 <code>zIndex</code> 。 (自 1.1 新增)
<code>getMap()</code>	Map	返回覆盖物所在的 <code>map</code> 对象。 (自 1.2 新增)
<code>addContextMenu(menu:ContextMenu)</code>	none	添加右键菜单。 (自 1.2 新增)
<code>removeContextMenu(menu:ContextMenu)</code>	none	移除右键菜单。 (自 1.2 新增)
<code>setAnimation(animation:Animation Null)</code>	none	设置标注动画效果。如果参数为 <code>null</code> ，则取消动画效果。该方法需要在 <code>addOverlay</code> 方法后设置。

		(自 1.2 新增)
setShadow(shadow:Icon)	none	设置标注阴影图标。 (自 1.2 新增)
getShadow()	Icon	获取标注阴影图标。 (自 1.2 新增)
addEventListener(event:String, handler:Function)	none	添加事件监听函数
removeEventListener(event:String, handler:Function)	none	移除事件监听函数

## 事件

事件	参数	描述
click	event{type, target}	点击标注图标后会触发此事件。
dblclick	event{type, target, point,pixel}	双击标注图标后会触发此事件。
mousedown	event{type, target, point,pixel}	鼠标在标注图上按下触发此事件。
mouseup	event{type, target, point,pixel}	鼠标在标注图上释放触发此事件。
mouseout	event{type, target, point,pixel}	鼠标离开标注时触发此事件。
mouseover	event{type, target, point,pixel}	当鼠标进入标注图标区域时会触发此事件。
remove	event{type, target}	移除标注时触发。
infowindowclose	event{type, target}	信息窗在此标注上关闭时触发此事件。
infowindowopen	event{type, target}	信息窗在此标注上打开时触发此事件。
dragstart	event{type, target}	开始拖拽标注时触发此事件。 (自 1.1 新增)
dragging	event{type, target, pixel, point}	拖拽标注过程中触发此事件。 (自 1.1 新增)
dragend	event{type, target, pixel, point}	拖拽结束时触发此事件。 (自 1.1 新增)

rightclick	event{type, target}	右键点击标注时触发此事件。 (自 1.1 新增)
------------	---------------------	-----------------------------

## MarkerOptions

此类表示 Marker 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

### 属性

属性	类型	描述
offset	Size	标注的位置偏移值。
icon	Icon	标注所用的图标对象。
enableMassClear	Boolean	是否在调用 map.clearOverlays 清除此覆盖物，默认为 true。 (自 1.1 新增)
enableDragging	Boolean	是否启用拖拽，默认为 false。 (自 1.1 新增)
enableClicking	Boolean	是否响应点击事件。默认为 true。 (自 1.2 新增)
raiseOnDrag	Boolean	拖拽标注时，标注是否开启离开地图表面效果。默认为 false。 (自 1.2 新增)
draggingCursor	String	拖拽标注时的鼠标指针样式。此属性值需遵循 CSS 的 cursor 属性规范。 (自 1.2 新增)
shadow	Icon	阴影图标。 (自 1.2 新增)
title	String	鼠标移到 marker 上的显示内容。

## Animation

此常量表示标注的动画效果(自 1.2 新增)。

## 常量

常量	描述
BMAP_ANIMATION_DROP	坠落动画。
BMAP_ANIMATION_BOUNCE	跳动动画。

## Icon

此类表示标注覆盖物所使用的图标。

## 构造函数

构造函数	描述
<code>Icon(url:String, size:Size[, opts:IconOptions])</code>	以给定的图像地址和大小创建图标对象实例。

## 属性

属性	类型	描述
<code>offset</code>	<code>Size</code>	图标的定位点相对于图标左上角的偏移值。 (自 1.2 废弃)请使用 <code>anchor</code>
<code>anchor</code>	<code>Size</code>	图标的定位点相对于图标左上角的偏移值。 (自 1.2 新增)
<code>size</code>	<code>Size</code>	图标可视区域的大小。
<code>imageOffset</code>	<code>Size</code>	图标所用的图片相对于可视区域的偏移值，此功能的作用等同于 CSS 中的 <code>background-position</code> 属性。
<code>imageUrl</code>	<code>String</code>	图标所用图像资源的位置。
<code>infoWindowOffset</code>	<code>Size</code>	信息窗口开启位置相对于图标左上角的偏移值。 (自 1.2 废弃)
<code>infoWindowAnchor</code>	<code>Size</code>	信息窗口开启位置相对于图标左上角的偏移值。 (自 1.2 新增)
<code>printImageUrl</code>	<code>String</code>	设置 icon 打印图片的 url，该打印图片只针对 IE6 有效，解决 IE6 使用 PNG

1	g	滤镜导致的错位问题。如果您的 icon 没有使用 PNG 格式图片或者没有使用 CSS Sprites 技术，则可忽略此配置。 (自 1.1 新增)
---	---	---

## 方法

方法	返回值	描述
<code>setImageUrl(imageUrl:String)</code>	none	设置图片资源的地址。
<code>setSize(size:Size)</code>	none	设置图标可视区域的大小。
<code>setOffset(offset:Size)</code>	none	设置图标定位点相对于其左上角的偏移值。 (自 1.2 废弃)
<code>setAnchor(anchor:Size)</code>	none	设置图标定位点相对于其左上角的偏移值。 (自 1.2 新增)
<code>setImageOffset(offset:Size)</code>	none	设置图片相对于可视区域的偏移值。
<code>setInfoWindowOffset(offset:Size)</code>	none	设置信息窗口开启位置相对于图标左上角的偏移值。 (自 1.2 废弃)
<code>setInfoWindowAnchor(anchor:Size)</code>	none	设置信息窗口开启位置相对于图标左上角的偏移值。 (自 1.2 新增)
<code>setPrintImageUrl(url:String)</code>	none	设置 icon 的打印图片，该打印图片只针对 IE6 有效，解决 IE6 使用 PNG 滤镜导致的错位问题。如果您的 icon 没有使用 PNG 格式图片或者没有使用 CSS Sprites 技术，则可忽略此配置。 (自 1.1 新增)

## IconOptions

此类表示 Icon 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

## 属性

属性	类型	描述
offset	Size	图标的定位锚点。此点用来决定图标与地理位置的关系，是相对于图标左上角的偏移值，默认等于图标宽度和高度的中间值。



		(自 1.2 废弃)
anchor	Size	图标的定位锚点。此点用来决定图标与地理位置的关系，是相对于图标左上角的偏移值，默认等于图标宽度和高度的中间值。 (自 1.2 新增)
imageOffset	Size	图片相对于可视区域的偏移值。
infoWindowOffset	Size	信息窗口定位锚点。开启信息窗口时，信息窗口底部尖角相对于图标左上角的位置，默认等于图标的 anchor。 (自 1.2 废弃)
infoWindowAnchor	Size	信息窗口定位锚点。开启信息窗口时，信息窗口底部尖角相对于图标左上角的位置，默认等于图标的 anchor。 (自 1.2 新增)
printImageUrl	String	用于打印的图片，此属性只适用于 IE6，为了解决 IE6 在包含滤镜的情况下打印样式不正确的问题。 (自 1.1 新增)

## InfoWindow

此类表示地图上包含信息的窗口。

## 构造函数

构造函数	描述
InfoWindow(content:String HTMLElement[, opts:InfoWindowOptions])	创建一个信息窗实例，其中 content 支持 HTML 内容。 1.2版本开始 content 参数支持传入 DOM 结点。

## 方法

方法	返回值	描述
setWidth(width:Number)	none	设置信息窗口的宽度，单位像素。取值范围：220 - 730。
setHeight(height:Number)	none	设置信息窗口的高度，单位像素。取值范围：60 - 650。
redraw()	none	重绘信息窗口，当信息窗口内容发生变化时进行调用。

<code>setTitle(title:String HTMLElement)</code>	<code>none</code>	设置信息窗口标题。支持 HTML 内容。1.2版本开始 <code>title</code> 参数支持传入 DOM 结点。
<code>getTitle()</code>	<code>String HTMLElement</code>	返回信息窗口标题。 (自 1.2 新增)
<code>setContent(content:String HTMLElement)</code>	<code>none</code>	设置信息窗口内容。支持 HTML 内容。1.2版本开始 <code>content</code> 参数支持传入 DOM 结点。
<code>getContent()</code>	<code>String HTMLElement</code>	返回信息窗口内容。 (自 1.2 新增)
<code>getPosition()</code>	<code>Point</code>	返回信息窗口的位置。
<code>enableMaximize()</code>	<code>none</code>	启用窗口最大化功能。需要设置最大化后信息窗口里的内容，该接口才生效。
<code>disableMaximize()</code>	<code>none</code>	禁用窗口最大化功能。
<code>isOpen()</code>	<code>Boolean</code>	返回信息窗口的打开状态。
<code>setMaxContent(content:String)</code>	<code>none</code>	信息窗口最大化时所显示内容，支持 HTML 内容。
<code>maximize()</code>	<code>none</code>	最大化信息窗口(自 1.1 新增)
<code>restore()</code>	<code>none</code>	还原信息窗口(自 1.1 新增)
<code>enableAutoPan()</code>	<code>none</code>	开启打开信息窗口时地图自动平移。 (自 1.1 新增)
<code>disableAutoPan()</code>	<code>none</code>	关闭打开信息窗口时地图自动平移。 (自 1.1 新增)
<code>enableCloseOnClick()</code>	<code>none</code>	开启点击地图时关闭信息窗口。 (自 1.1 新增)
<code>disableCloseOnClick()</code>	<code>none</code>	关闭点击地图时关闭信息窗口。 (自 1.1 新增)
<code>addEventListener(event:String, handler:Function)</code>	<code>none</code>	添加事件监听函数
<code>removeEventListener(event:String, handler:Function)</code>	<code>none</code>	移除事件监听函数

## 事件

事件	参数	描述
<code>close</code>	<code>event{type, target,</code>	信息窗口被关闭时触发此事件。

	point}	
open	event{type, target, point}	信息窗口被打开时触发此事件。
maximize	event{type, target}	信息窗口最大化后触发此事件。(自 1.1 新增)
restore	event{type, target}	信息窗口还原时触发此事件。(自 1.1 新增)
clickclose	event{type, target}	点击信息窗口的关闭按钮时触发此事件。(自 1.1 新增)

## InfoWindowOptions

此类表示 InfoWindow 构造函数的可选参数，它没有构造函数，但可通过对象字面量形式表示。

### 属性

属性	类型	描述
width	Number	信息窗宽度，单位像素。取值范围：0, 220 - 730。如果您指定宽度为0，则信息窗口的宽度将按照其内容自动调整。
height	Number	信息窗高度，单位像素。取值范围：0, 60 - 650。如果您指定高度为0，则信息窗口的高度将按照其内容自动调整。
maxWidth	Number	信息窗最大化时的宽度，单位像素。取值范围：220 - 730。
offset	Size	信息窗位置偏移值。默认情况下在地图上打开的信息窗底端的尖角将指向其地理坐标，在标注上打开的信息窗底端尖角的位置取决于标注所用图标的 infoWindowOffset 属性值，您可以为信息窗添加偏移量来改变默认位置。
title	String	信息窗标题文字，支持 HTML 内容。
enableAutoPan	Boolean	是否开启信息窗口打开时地图自动移动（默认开启）。(自 1.1 新增)
enableCloseOnClick	Boolean	是否开启点击地图关闭信息窗口（默认开启）(自 1.1 新增)
enableMessage	Boolean	是否在信息窗里显示短信发送按钮（默认开启）(自1.5新增)。
message	String	自定义部分的短信内容，可选项。完整的短信内容包括：自定义部分+位

	g	置链接，不设置时，显示默认短信内容。（自1.5新增）
--	---	----------------------------

## Label

此类表示地图上的文本标注。

## 构造函数

构造函数	描述
<code>Label(content:String[, opts:LabelOptions])</code>	创建一个文本标注实例。 <code>point</code> 参数指定了文本标注所在的地理位置。

## 方法

方法	返回值	描述
<code>setStyle(styles:Object)</code>	none	设置文本标注样式，该样式将作用于文本标注的容器元素上。其中 <code>styles</code> 为 JavaScript 对象常量，比如： <code>setStyle({ color : "red", fontSize : "12px" })</code> 注意：如果 css 的属性名中包含连字符，需要将连字符去掉并将其后的字母进行大写处理，例如：背景色属性要写成： <code>backgroundColor</code> 。
<code>setContent(content:String)</code>	none	设置文本标注的内容。支持 HTML。
<code>setPoint(point:Point)</code>	none	设置文本标注坐标。仅当通过 <code>Map.addOverlay()</code> 方法添加的文本标注有效。（自 1.2 废弃）
<code>getPoint()</code>	Point	返回文本标注坐标。（自 1.2 废弃）
<code>setPosition(position:Point)</code>	none	设置文本标注坐标。仅当通过 <code>Map.addOverlay()</code> 方法添加的文本标注有效。（自 1.2 新增）
<code>getPosition()</code>	Point	获取 Marker 的地理坐标。（自 1.2 新增）
<code>setOffset(offset:Size)</code>	none	设置文本标注的偏移值。

<code>getOffset()</code>	Size	返回文本标注的偏移值。
<code>setTitle(title:String)</code>	none	设置文本标注的标题，当鼠标移至标注上时显示此标题。
<code>getTitle()</code>	String	返回文本标注的标题。
<code>enableMassClear()</code>	none	允许覆盖物在 <code>map.clearOverlays</code> 方法中被清除。(自 1.1 新增)
<code>disableMassClear()</code>	none	禁止覆盖物在 <code>map.clearOverlays</code> 方法中被清除。(自 1.1 新增)
<code>setZIndex(zIndex: Number)</code>	none	设置覆盖物的 <code>zIndex</code> 。(自 1.1 新增)
<code>setPosition(position:Point)</code>	none	设置地理坐标 (自1.2新增)
<code>getPosition()</code>	Point	返回地理坐标 (自1.2新增)
<code>getMap()</code>	Map	返回覆盖物所在的 <code>map</code> 对象。(自1.2新增)
<code>addEventListener(event:String, handler:Function)</code>	none	添加事件监听函数
<code>removeEventListener(event:String, handler:Function)</code>	none	移除事件监听函数

## 事件

事件	参数	描述
<code>click</code>	<code>event{type, target}</code>	点击文本标注后会触发此事件。
<code>dblclick</code>	<code>event{type, target}</code>	双击文本标注后会触发此事件。
<code>mousedown</code>	<code>event{type, target}</code>	鼠标在文本标注上按下触发此事件。
<code>mouseup</code>	<code>event{type, target}</code>	鼠标在文本标注释放触发此事件。
<code>mouseout</code>	<code>event{type, target}</code>	鼠标离开文本标注时触发此事件。
<code>mouseover</code>	<code>event{type, target}</code>	当鼠标进入文本标注区域时会触发此事件。
<code>remove</code>	<code>event{type, target}</code>	移除文本标注时触发。
<code>rightclick</code>	<code>event{type, target}</code>	右键点击标注时触发此事件。(自 1.1 新增)

## LabelOptions

此类表示 `Label` 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

### 属性

属性	类型	描述
<code>offset</code>	<code>Size</code>	文本标注的位置偏移值。
<code>point</code>	<code>Point</code>	文本标注的坐标点。(自 1.2 废弃)
<code>position</code> (原 <code>point</code> )	<code>Point</code>	文本标注的地理位置。(自 1.2 新增)
<code>enableMassClear</code>	<code>Boolean</code>	是否在调用 <code>map.clearOverlays</code> 清除此覆盖物，默认为 <code>true</code> 。(自 1.1 新增)

## Polyline

### 构造函数

使用浏览器的矢量制图工具（如果可用）在地图上绘制折线的地图叠加层。

构造函数	描述
<code>Polyline(points:Array&lt;Point&gt;[, opts:PolylineOptions])</code>	创建折线覆盖物对象

### 方法

方法	返回值	描述
<code>setPoints(points:Array&lt;Point&gt;)</code>	<code>none</code>	设置折线的点数组。(自 1.2 废弃)
<code>getPoints()</code>	<code>Array&lt;Point&gt;</code>	返回折线的点数组。(自 1.2 废弃)
<code>setPath(path:Array&lt;Point&gt;)</code>	<code>none</code>	设置折线的点数组 (自 1.2 新增)
<code>getPath()</code>	<code>Array&lt;</code>	返回折线的点数组 (自 1.2 新增)

	Point>	
setStrokeColor(color:String)	none	设置折线的颜色。
getStrokeColor()	String	返回折线的颜色。
setStrokeOpacity(opacity:Number)	none	设置透明度，取值范围0 - 1。
getStrokeOpacity()	Number	返回透明度。
setStrokeWeight(weight:Number)	none	设置线的宽度，范围为大于等于1的整数。
getStrokeWeight()	Number	返回线的宽度。
setStrokeStyle(style:String)	none	设置是为实线或虚线，solid 或 dashed。
getStrokeStyle()	String	返回当前线样式状态，实线或者虚线。
getBounds()	Bounds	返回覆盖物的地理区域范围。(自 1.1 新增)
enableEditing()	none	开启编辑功能。(自 1.1 新增)
disableEditing()	none	关闭编辑功能。(自 1.1 新增)
enableMassClear()	none	允许覆盖物在 map.clearOverlays 方法中被清除。(自 1.1 新增)
disableMassClear()	none	禁止覆盖物在 map.clearOverlays 方法中被清除。(自 1.1 新增)
setPointAt(index: Number, point: Point)	none	修改指定位置的坐标。Number 从0开始计数。例如 setPointAt(2, point2a)代表将折线的第3个点，坐标设为 point2a。(自 1.2 废弃)
setPositionAt(index: Number, point: Point)	none	修改指定位置的坐标。索引 index 从0开始计数。例如 setPointAt(2, point)代表将折线的第3个点的坐标设为 point(自 1.2 新增)
getMap()	Map	返回覆盖物所在的 map 对象。(自1.2新增)
addEventListener(event:String, handler:Function)	none	添加事件监听函数
removeEventListener(event:String, handler:Function)	none	移除事件监听函数

## 事件

事件	参数	描述
click	event{type, target, point, pixel}	点击折线后会触发此事件。
dblclick	event{type, target, point, pixel}	双击折线后会触发此事件。
mousedown	event{type, target, point, pixel}	鼠标在折线上按下触发此事件。
mouseup	event{type, target, point, pixel}	鼠标在折线释放触发此事件。
mousemove	event{type, target, point, pixel}	鼠标在折线上移动时触发此事件
mouseout	event{type, target, point, pixel}	鼠标离开折线时触发此事件。
mouseover	event{type, target, point, pixel}	当鼠标进入折线区域时会触发此事件。
remove	event{type, target}	移除折线时触发。
lineupdate	event{type, target}	覆盖物的属性发生变化时触发。(自 1.1 新增)

## PolylineOptions

此类表示 Polyline 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

## 属性

属性	类型	描述
strokeColor	String	折线颜色
strokeWeight	Number	折线的宽度，以像素为单位。
strokeOpacity	Number	折线的透明度，取值范围0 - 1。
strokeStyle	String	折线的样式，solid 或 dashed。
enableMassClear	Boolean	是否在调用 map.clearOverlays 清除此覆盖物，默认为 true。 (自 1.1 新增)



enableEditing	Boolean	是否启用线编辑，默认为 false。 (自 1.1 新增)
enableClicking	Boolean	是否响应点击事件，默认为 true。 (自 1.2 新增)

## Polygon

此类表示一个多边形覆盖物。

## 构造函数

构造函数	描述
Polygon(points:Array<Point>[, opts:PolygonOptions])	创建多边形覆盖物

## 方法

方法	返回值	描述
setPoints(points:Array<Point>)	none	设置多边型的点数组。(自 1.2 废弃)
getPoints()	Array<Point>	返回多边型的点数组。(自 1.2 废弃)
setPath(path:Array<Point>)	none	设置多边型的点数组 (自1.2新增)
getPath()	Array<Point>	返回多边型的点数组 (自1.2新增)
setStrokeColor(color:String)	none	设置多边型的边线颜色，参数为合法的 CSS 颜色值。
getStrokeColor()	String	返回多边型的边线颜色。
setFillColor(color:String)	none	设置多边形的填充颜色，参数为合法的 CSS 颜色值。 当参数为空字符串时，折线覆盖物将没有填充效果。
getFillColor()	String	返回多边形的填充颜色。
setStrokeOpacity(opacity:Number)	none	设置多边形的边线透明度，取值范围0 - 1。
getStrokeOpacity()	Number	返回多边形的边线透明度。

setFillOpacity(opacity: Number)	none	设置多边形的填充透明度，取值范围0 - 1。
getFillOpacity()	Number	返回多边形的填充透明度。
setStrokeWeight(weight: Number)	none	设置多边形边线的宽度，取值为大于等于1的整数。
getStrokeWeight()	Number	返回多边形边线的宽度。
setStrokeStyle(style: String)	none	设置多边形边线样式为实线或虚线，取值 solid 或 dashed。
getStrokeStyle()	String	返回多边形边线样式。
getBounds()	Bounds	返回覆盖物的地理区域范围。(自 1.1 新增)
enableEditing()	none	开启编辑功能(自 1.1 新增)
disableEditing()	none	关闭编辑功能(自 1.1 新增)
enableMassClear()	none	允许覆盖物在 map.clearOverlays 方法中被清除(自 1.1 新增)
disableMassClear()	none	禁止覆盖物在 map.clearOverlays 方法中被清除(自 1.1 新增)
setPointAt(index: Number, point: Point)	none	修改指定位置的坐标。Number 从0开始计数。例如 setPointAt(2, point2a)代表将折线的第3个点，坐标设为 point2a。(自 1.2 废弃)
setPositionAt(index: Number, point: Point)	none	修改指定位置的坐标。索引 index 从0开始计数。例如 setPointAt(2, point)代表将折线的第3个点的坐标设为 point(自 1.2 新增)
getMap()	Map	返回覆盖物所在的 map 对象。(自1.2新增)
addEventListener(event: String, handler: Function)	none	添加事件监听函数
removeEventListener(event: String, handler: Function)	none	移除事件监听函数

## 事件

事件	参数	描述
click	event{type, target, point, pixel}	点击多边形后会触发此事件。
dblclick	event{type, target, point,	双击多边形后会触发此事件。

	pixel}	
mousedown	event{type, target, point, pixel}	鼠标在多边形上按下触发此事件。
mouseup	event{type, target, point, pixel}	鼠标在多边形释放触发此事件。
mousemove	event{type, target, point, pixel}	鼠标在多边形上移动时触发此事件
mouseout	event{type, target, point, pixel}	鼠标离开多边形时触发此事件。
mouseover	event{type, target, point, pixel}	当鼠标进入多边形区域时会触发此事件。
remove	event{type, target}	移除多边形时触发。
lineupdate	event{type, target}	覆盖物的属性发生变化时触发。(自 1.1 新增)

## PolygonOptions

此类表示 Polygon 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

## 属性

属性	类型	描述
strokeColor	String	边线颜色。
fillColor	String	填充颜色。当参数为空时，折线覆盖物将没有填充效果。
strokeWeight	Number	边线的宽度，以像素为单位。
strokeOpacity	Number	边线透明度，取值范围0 - 1。
fillOpacity	Number	填充的透明度，取值范围0 - 1。
strokeStyle	String	边线的样式，solid 或 dashed。
enableMassClear	Boolean	是否在调用 map.clearOverlays 清除此覆盖物，默认为 true。 (自 1.1 新增)
enableEditing	Boolean	是否启用编辑功能，默认为 false。 (自 1.1 新增)

enableClicking	Boolean	是否响应点击事件，默认为 true。  (自 1.2 新增)
----------------	---------	--------------------------------------

## Circle

此类表示地图上的圆覆盖物。

## 构造函数

构造函数	描述
Circle(center:Point, radius: Number[, opts:CircleOptions])	创建圆覆盖物。(自 1.1 新增)

## 方法

方法	返回值	描述
setCenter(center:Point)	none	设置圆形的中心点坐标。
getCenter()	Point	返回圆形的中心点坐标。
setRadius(radius: Number)	none	设置圆形的半径，单位为米。
getRadius()	Number	返回圆形的半径，单位为米。
getBounds()	Bounds	返回圆形的地理区域范围。
setStrokeColor(color:String)	none	设置圆形的边线颜色，参数为合法的 CSS 颜色值。
getStrokeColor()	String	返回圆形的边线颜色。
setFillColor(color:String)	none	设置圆形的填充颜色，参数为合法的 CSS 颜色值。当参数为空字符串时，圆形覆盖物将没有填充效果。

<code>getFillColor()</code>	String	返回圆形的填充颜色。
<code>setStrokeOpacity(opacity: Number)</code>	none	设置圆形的边线透明度，取值范围0 - 1。
<code>getStrokeOpacity()</code>	Number	返回圆形的边线透明度。
<code>setFillOpacity(opacity: Number)</code>	none	设置圆形的填充透明度，取值范围0 - 1。
<code>getFillOpacity()</code>	Number	返回圆形的填充透明度。
<code>setStrokeWeight(weight: Number)</code>	none	设置圆形边线的宽度，取值为大于等于1的整数。
<code>getStrokeWeight()</code>	Number	返回圆形边线的宽度。
<code>setStrokeStyle(style: String)</code>	none	设置圆形边线样式为实线或虚线，取值 <b>solid</b> 或 <b>dashed</b> 。
<code>getStrokeStyle()</code>	String	返回圆形边线样式。
<code>enableEditing()</code>	none	开启编辑功能
<code>disableEditing()</code>	none	关闭编辑功能
<code>enableMassClear()</code>	none	允许覆盖物在 <code>map.clearOverlays</code> 方法中被清除
<code>disableMassClear()</code>	none	禁止覆盖物在 <code>map.clearOverlays</code> 方法中被清除
<code>getMap()</code>	Map	返回覆盖物所在的 <code>map</code> 对象。（自1.2新增）
<code>addEventListener(event: String, handler: Function)</code>	none	添加事件监听函数
<code>removeEventListener(event: String, handler: Function)</code>	none	移除事件监听函数

## 事件

事件	参数	描述
----	----	----

click	event{type, target, point, pixel}	鼠标点击圆形后会触发此事件。
dblclick	event{type, target, point, pixel}	鼠标双击圆形后会触发此事件。
mousedown	event{type, target, point, pixel}	鼠标在圆形上按下触发此事件。
mouseup	event{type, target, point, pixel}	鼠标在圆形释放触发此事件。
mousemove	event{type, target, point, pixel}	鼠标在圆形上移动时触发此事件。
mouseout	event{type, target, point, pixel}	鼠标离开圆形时触发此事件。
mouseover	event{type, target, point, pixel}	当鼠标进入圆形区域时会触发此事件。
remove	event{type, target}	移除圆形时触发此事件。
lineupdate	event{type, target}	圆形覆盖物的属性发生变化时触发此事件。

## CircleOptions

Circle 类构造函数的可选参数（1.1 版本新增）。它没有构造函数，但可通过对象字面量形式表示。

### 属性

属性	类型	描述
strokeColor	String	圆形边线颜色。
fillColor	String	圆形填充颜色。当参数为空时，圆形将没有填充效果。
strokeWeight	Number	圆形边线的宽度，以像素为单位。
strokeOpacity	Number	圆形边线透明度，取值范围0 - 1。
fillOpacity	Number	圆形填充的透明度，取值范围0 - 1。
strokeStyle	String	圆形边线的样式，solid 或 dashed。
enableMassClear	Boolean	是否在调用 map.clearOverlays 清除此覆盖物，默认为 true。
enableEditing	Boolean	是否启用线编辑，默认为 false。
enableClicking	Boolean	是否响应点击事件，默认为 true。

(自 1.2 新增)

## Hotspot

此类表示地图上的一个热区。(自 1.2 新增)

## 构造函数

构造函数	描述
<code>Hotspot(position:Point[, options:HotspotOptions])</code>	创建 <code>Hotspot</code> 对象实例。(自 1.2 新增)

## 方法

属性	返回值	描述
<code>getPosition()</code>	<code>Point</code>	获取热区位置坐标。(自 1.2 新增)
<code>setPosition(position:Point)</code>	<code>None</code>	设置热区位置坐标。(自 1.2 新增)
<code>getText()</code>	<code>String</code>	获取热区提示文本。(自 1.2 新增)
<code>setText(text:String)</code>	<code>None</code>	设置热区提示文本。(自 1.2 新增)
<code>getUserData()</code>	<code>Mix</code>	获取热区对应的用户数据。(自 1.2 新增)
<code>setUserData(data:Mix)</code>	<code>None</code>	设置热区对应的用户数据。(自 1.2 新增)

## HotspotOptions

此类是 `addHotspot` 方法的可选参数，没有构造函数，通过对象字面量形式表示。(自 1.2 新增)

## 属性

属性	类型	描述
<code>text</code>	<code>String</code>	当鼠标移至某一热区上时出现的文字提示。(自 1.2 新增)
<code>offsets</code>	<code>Array</code>	热区响应区域距中心点的扩展偏移值。数组的四个数值分别表示上、右、下、左距离中心点的扩展偏移量。默认偏移量为 <code>[5, 5, 5, 5]</code> 。(自 1.2 新增)

userData	Mix	由用户填入的自定义数据。(自 1.2 新增)
minZoom	Number	热区生效的最小级别。(自 1.2 新增)
maxZoom	Number	热区生效的最大级别。(自 1.2 新增)

## MapPanes

此类表示地图上所有覆盖物的容器集合，没有构造函数，通过对象字面量形式表示。通过 Map 的 getPanes 方法可获得该对象实例。

## 属性

属性	类型	描述
floatPane	HTMLElement	信息窗口所在的容器。
markerMouseTarget	HTMLElement	标注点击区域所在的容器。
floatShadow	HTMLElement	信息窗口阴影所在的容器。
labelPane	HTMLElement	文本标注所在的容器。
markerPane	HTMLElement	标注图标所在的容器。
markerShadow	HTMLElement	标注阴影所在的容器。
mapPane	HTMLElement	折线、多边形等矢量图形所在的容器。

## 右键菜单类

### ContextMenu

此类表示右键菜单。您可以在地图上添加自定义内容的右键菜单。



## 构造函数

构造函数	描述
<code>ContextMenu()</code>	创建一个右键菜单实例。

## 方法

方法	返回值	描述
<code>addItem(item:MenuItem)</code>	<code>none</code>	添加菜单项。
<code>getItem(index:Number)</code>	<code>MenuItem</code>	返回指定索引位置的菜单项，第一个菜单项的索引为 0。
<code>removeItem(item:MenuItem)</code>	<code>none</code>	移除菜单项。
<code>addSeparator()</code>	<code>none</code>	添加分隔符。
<code>removeSeparator(index:Number)</code>	<code>none</code>	移除指定索引位置的分隔符，第一个分隔符的索引为 0。

## 事件

事件	参数	描述
<code>open</code>	<code>event{type, target, point, pixel}</code>	右键菜单打开时触发，事件参数 <code>point</code> 和 <code>pixel</code> 分别表示菜单开启时的地理和像素坐标点。
<code>close</code>	<code>event{type, target, point, pixel}</code>	右键菜单关闭时触发，事件参数 <code>point</code> 和 <code>pixel</code> 分别表示菜单开启时的地理和像素坐标点。

## MenuItem

此类表示一个菜单项

## 构造函数

构造函数	描述
<code>MenuItem(text:String, callback:Function[,</code>	创建一个菜单项。当菜单项被点击时，系统将会以当前菜单弹出的地理坐标点作为参数调用回调函数 <code>callback</code> 。

```
opts:MenuItemOptions])
```

## 方法

方法	返回值	描述
<code>setText(text:String)</code>	none	设置菜单项显示的文本。
<code>enable()</code>	none	启用菜单项。
<code>disable()</code>	none	禁用菜单项。

## MenuItemOptions

此类表示 MenuItem 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

## 属性

属性	类型	描述
<code>width</code>	Number	指定此菜单项的宽度，菜单以最长的菜单项宽度为准。

## 地图类型类

## MapType

此类表示一种地图类型，您可以通过实例化此类自定义地图类型（自1.2新增）。

## 构造函数

方法	描述
<code>MapType(name:String, layers: TileLayer Array&lt;TileLayer&gt;[, options: MapTypeOptions])</code>	创建 MapType 对象实例。

## 方法

方法	返回值	描述
<code>getName()</code>	<code>String</code>	返回地图类型名称。
<code>getTileLayer()</code>	<code>TileLayer</code>	返回地图类型对应的图层。
<code>getMinZoom()</code>	<code>Number</code>	返回地图类型允许的最小级别。
<code>getMaxZoom()</code>	<code>Number</code>	返回地图类型允许的最大级别。
<code>getProjection()</code>	<code>Projection</code>	返回地图类型所使用的投影实例。
<code>getTextColor()</code>	<code>String</code>	返回地图类型对应的前景色。
<code>getTips()</code>	<code>String</code>	返回地图类型的提示说明，用于在地图类型控件中提示。

## 常量

常量	描述
<code>BMAP_NORMAL_MAP</code>	此地图类型展示普通街道视图。
<code>BMAP_PERSPECTIVE_MAP</code>	此地图类型展示透视图像视图。
<code>BMAP_SATELLITE_MAP</code>	此地图类型展示卫星视图。(自 1.2 新增)
<code>BMAP_HYBRID_MAP</code>	此地图类型展示卫星和路网的混合视图。(自 1.2 新增)

## MapTypeOptions

此类是 `MapType` 构造函数的可选参数，不能实例化，通过对象字面量形式表示（自1.2新增）。

## 属性

属性	类型	描述
<code>minZoom</code>	<code>Number</code>	该类型地图的最小级别。
<code>maxZoom</code>	<code>Number</code>	该类型地图的最大级别。
<code>errorImageUrl</code>	<code>String</code>	当没有图块时所显示的错误图片地址。默认为透明图。
<code>textColor</code>	<code>Number</code>	地图类型对应的前景色。
<code>tips</code>	<code>String</code>	提示说明信息，用于在地图类型控件中进行提示。

## Projection

此类表示地图投影抽象基类，不可实例化，但可通过 MapType 的 getProjection 方法获得（自1.2新增）。

### 方法

方法	返回值	描述
lngLatToPoint(lngLat:Point)	Pixel	抽象，根据球面坐标获得平面坐标。
pointToLngLat(point:Pixel)	Point	抽象，根据平面坐标获得球面坐标。

## MercatorProjection

此类表示地图投影抽象基类，不可实例化，但可通过 MapType 的 getProjection 方法获得（自1.2新增）。

### 方法

方法	返回值	描述
lngLatToPoint(lngLat:Point)	Pixel	抽象，根据球面坐标获得平面坐标。
pointToLngLat(point:Pixel)	Point	抽象，根据平面坐标获得球面坐标。

## PerspectiveProjection

此类表示透视地图投影类，一般通过 MapType 的 getProjection 方法获得实例（自1.2新增）。

### 方法

方法	返回值	描述
lngLatToPoint(lngLat:Point)	Pixel	根据球面坐标获得平面坐标。
pointToLngLat(point:Pixel)	Point	根据平面坐标获得球面坐标。

# 地图图层类

## TileLayer

此类表示一个地图图层，您可以向地图中添加自定义图层。

## 构造函数

构造函数	描述
<code>TileLayer([opts:TileLayerOptions])</code>	创建一个地图图层实例。

## 方法

方法	返回值	描述
<code>getTilesUrl(tileCoord: Pixel, zoom: Number)</code>	String	抽象。向地图返回地图图块的网址，图块索引由 <code>tileCoord</code> 的 <code>x</code> 和 <code>y</code> 属性在指定的缩放级别 <code>zoom</code> 提供。如果您在 <code>TileLayerOptions</code> 中提供了 <code>tileUrlTemplate</code> 参数，则可不实现此接口。
<code>getMapType()</code>	String   enum MapType	返回此图块层的地图类型。
<code>getCopyright()</code>	Number	返回地图图层数据的版权对象。
<code>isTransparentPng()</code>	Number	如果图层所用的图片为 PNG 格式并且包含透明信息，则返回 <code>true</code> 。

## TileLayerOptions

此类表示 `TileLayer` 构造函数的可选参数。

## 属性

属性	类型	描述
<code>transparentPng</code>	Boolean	是否使用了带有透明信息的 PNG。由于 IE6 不支持 PNG 透明，因此需要特殊处理。

tileUrlTemplate	String	指定图块网址模板，该模板可以针对每个图块请求而展开，以根据现有的图块坐标系引用唯一的图块。模板的格式应该为： <code>http://yourhost/tile?x={X}&amp;y={Y}&amp;z={Z}.png</code> 其中 X 和 Y 分别指纬度和经度图块坐标，Z 指缩放级别，比如： <code>http://yourhost/tile?x=3&amp;y=27&amp;z=5.png</code> 如果您没有提供图块网址模板，您需要实现 <code>TileLayer.getTileUrl()</code> 抽象方法。
copyright	Copyright	地图图层的版权信息。
zIndex	Number	图层的 <code>zIndex</code> 。

(自 1.2 新增)

## CustomLayer

此类表示用户自定义图层，您可以将存储在 LBS.云的数据通过该类以图层样式叠加到地图上。

## 构造函数

构造函数	描述
<code>CustomLayer(databoxId: Number)</code>	创建一个用户自定义图层实例。

## 事件

事件	参数	描述
Onhotspotclick	<code>event{type, target, customPoi}</code>	单击图层上的点时触发该事件。

## CustomPoi

此类表示 `CustomLayer` 图层上的点数据。它没有构造函数，通过 `onhotspotclick` 事件返回值 `customPoi` 获得。

## 属性

属性	类型	描述
poiId	String	数据的id

databoxId	String	数据集的id
title	String	结果的名称标题
address	String	地址
phoneNumber	String	电话
postcode	String	邮政编码
provinceCode	Number	结果所在省的编码
province	String	结果所在省的名称
cityCode	Number	结果所在城市的编码
city	String	结果所在城市的名称
districtCode	Number	结果所在区县的编码
district	String	结果所在区县的名称
point	Point	结果所在的地理位置
tags	Array <String>	结果的筛选标签
typeId	Number	结果的类别id
extendedData	Json	用户扩展数据，结构根据用户的自定义

## TrafficLayer

此类表示交通流量图层。

## 构造函数

构造函数	描述
<code>BMap.TrafficLayer</code> ([opts:TrafficLayerOptions])	创建交通流量图层。参数: opts: TrafficLayerOptions, 可选 options 参数指定应作为对象常量传递。如果可选参数提供 predictDate, 则将显示预测流量。否则显示实时流量。(自 1.1 新增)

## TrafficLayerOptions

此类是 TrafficLayer 构造函数的可选参数，没有构造函数，不能实例化。

## 属性

属性	类型	描述
predictDate	PredictDate	预测日期。 (自 1.1 新增)

## PredictDate

此类表示交通流量的预测日期，没有构造函数，通过对象字面量形式表示。

## 属性

属性	类型	描述
weekday	Number	预测日期，取值1到7，表示周一到周日。
hour	Number	预测小时数，取值0到23，表示当日的0点到23点。

## PanoramaCoverageLayer

此类表示全景覆盖的城市。

## 构造函数

构造函数	描述
PanoramaCoverageLayer()	创建全景覆盖区域图层的实例。(自 V1.5 新增)

## 全景地图类

### Panorama

此类用来展示某位置的全景视图，可以单独放置在一个 div 容器中，也可以放在 Map 类的容器中。



## 构造函数

构造函数	描述
<code>Panorama(container:String HTMLElement[, opts:PanoramaOptions])</code>	在给定的结点中创建全景。

## 方法

方法	返回值	描述
<code>getLinks()</code>	<code>Array&lt;PanoramaLink&gt;</code>	获取全景中道路指示信息。
<code>getId()</code>	<code>String</code>	获取当前全景的 id。
<code>getPosition()</code>	<code>Point</code>	获取当前全景的位置，通过经纬度描述。
<code>getPov()</code>	<code>PanoramaPov</code>	获取当前全景的视角。
<code>getZoom()</code>	<code>Number</code>	获取当前全景的级别。
<code>setId(id:String)</code>	<code>none</code>	设置全景的 id。
<code>setPosition(position:Point)</code>	<code>none</code>	设置全景的位置。
<code>setPov(pov:PanoramaPov)</code>	<code>none</code>	设置全景的视角。
<code>setZoom(zoom:Number)</code>	<code>none</code>	设置全景的级别。
<code>enableScrollWheelZoom()</code>	<code>none</code>	开启鼠标滚轮缩放功能。仅对 PC 上有效。
<code>disableScrollWheelZoom()</code>	<code>none</code>	关闭鼠标滚轮缩放功能。
<code>show()</code>	<code>none</code>	显示全景。
<code>hide()</code>	<code>none</code>	隐藏全景。

## 事件

事件	参数	描述
<code>position_changed</code>	<code>none</code>	全景位置发生变化时触发。
<code>links_changed</code>	<code>none</code>	全景相邻道路发生变化时触发，通常是在位置变化时，异步获取新数据之后触发。

pov_changed	none	全景视角发生变化时触发。
zoom_changed	none	全景级别发生变化时触发。

## PanoramaOptions

此类为 Panorama 类构造函数的可选参数，使用对象字面量形式表示，不可实例化。

### 属性

属性	类型	描述
navigationControl	boolean	是否显示全景的导航控件，默认为 true。
linksControl	boolean	是否显示道路指示控件。默认为 true。

## PanoramaLink

此类用来描述全景中道路指示中相邻全景的信息。使用对象字面量形式表示，不可实例化。

### 属性

属性	类型	描述
description	String	相连全景的描述信息。
heading	Number	相连道路的方向，正北方向为 0，正东为 90，正南为 180，正西为 270。
id	String	相邻全景的 id。

## PanoramaPov

此类描述全景的视角。使用对象字面量形式表示，不可实例化。

### 属性

属性	类型	描述
heading	Number	水平方向的角度，正北方向为 0，正东为 90，正南为 180，正西为 270。

pitch	Number	竖直方向的角度，向上最大到 90 度，向下最大到-90 度。（在某些场景下，俯角可能无法到达最大值）
-------	--------	--

## PanoramaService

此类用来检索全景数据信息

### 构造函数

构造函数	描述
PanoramaService()	创建检索全景数据信息类的实例。

### 方法

方法	返回值	描述
getPanoramaById(id:String, callback:function(data:PanoramaData));	none	根据全景 id 返回全景数据，当获取不到数据时，回调函数参数为 null。
getPanoramaByLocation(point:Point, callback:function(data:PanoramaData));	none	根据坐标返回距离此处最近的全景数据，当获取不到数据时，回调函数参数为 null。

## PanoramaData

通过 PanoramaService 类检索获得。使用对象字面量形式表示，不可实例化。

### 属性

属性	类型	描述
id	String	全景 id。
description	String	全景的描述信息。
links	Array<PanoramaLink>	全景相邻道路信息。

position	Point	全景的地理坐标。
tiles	PanoramaTileData	全景图块信息。

## PanoramaTileData

图块数据信息，不可实例化。

## 属性

属性	类型	描述
centerHeading	Number	整个全景图中心位置的方位（单位角度）
tileSize	Size	图块展示的尺寸（单位像素），注意这个尺寸与实际图片尺寸可能不同。
worldSize	Size	整个全景图的尺寸（单位像素）。

## 服务类

### LocalSearch

用于位置检索、周边检索和范围检索。

## 构造函数

构造函数	描述
<code>LocalSearch(location:Map Point String[, opts:LocalSearchOptions])</code>	创建一个搜索类实例，其中 <b>location</b> 表示检索区域，其类型可为地图实例、坐标点或城市名称的字符串。当参数为地图实例时，检索位置由当前地图中心点确定，且搜索结果的标注将自动加载到地图上，并支持调整地图视野层级；当参数为坐标时，检索位置由该点所在位置确定；当参数为城市名称时，检索会在该城市内进行。

## 方法

方法	返回值	描述
<code>search(keyword:String Array&lt;String&gt;[,option:Object])</code>	none	根据检索词发起检索。当keyword为数组时将同时执行多关键字的查询，最多支持10个关键字，多关键字自 1.2 版本支持。 <code>option:{forceLocal:Boolean, customData:CustomData}</code> <code>forceLocal</code> 表示是否将搜索范围约束在当前城市， <code>customData</code> 表示检索 lbs 云服务的数据
<code>searchInBounds(keyword:String Array&lt;String&gt;, bounds:Bounds[,option:Object])</code>	none	根据范围和检索词发起范围检索。当keyword为数组时将同时执行多关键字检索，最多支持10个关键字，多关键字自 1.2 版本支持。 <code>option:{customData:CustomData}</code> <code>customData</code> 表示检索 lbs 云服务的数据
<code>searchNearby(keyword:String Array&lt;String&gt;, center:LocalResultPoi String Point, radius:Number[,option:Object])</code>	none	根据中心点、半径与检索词发起周边检索。当keyword为数组时将同时执行多关键字的检索，最多支持10个关键字，多关键字自 1.2 版本支持。当center为字符串时，半径参数将忽略。注意： <code>Point</code> 类型的中心点自 1.1 版本支持。 <code>option:{customData:CustomData}</code> <code>customData</code> 表示检索 lbs 云服务的数据
<code>getResults()</code>	<code>LocalResult</code>   <code>Array&lt;LocalResult&gt;</code>	返回最近一次检索的结果。如果是多关键字范围检索，则返回一个 <code>LocalResult</code> 的数组，数组中的结果顺序和范围检索中多关键字数组中顺序一致。注意多关键字查询自 1.2 版本支持。
<code>clearResults()</code>	none	清除最近一次检索的结果
<code>gotoPage(page:Number)</code>	none	检索特定页面的结果
<code>enableAutoViewport()</code>	none	启用根据结果自动调整地图层级，当指定了搜索结果所展现的地图时有效。
<code>disableAutoViewport()</code>	none	禁用根据结果自动调整地图层级。
<code>enableFirstResultSelection()</code>	none	启用自动选择第一个检索结果。
<code>disableFirstResultSelection()</code>	none	禁用自动选择第一个检索结果。

<code>setLocation(location:Map Point String)</code>	none	设置检索范围，参数类型可以为地图实例、坐标点或字符串。例： <code>setLocation("北京市")</code>
<code>setPageCapacity</code>	none	设置每页容量，取值范围：1 - 100，对于多关键字检索，每页容量表示每个关键字返回结果的数量（例如当用2个关键字检索时，实际结果数量范围为：2 - 200）。此值只对下一次检索有效。
<code>getPageCapacity</code>	Number	返回每页容量，对于多关键字检索，返回每个关键字对应的页面容量。
<code>setSearchCompleteCallback</code>	none	设置检索结束后的回调函数。参数： <b>results:</b> LocalResult 或 Array<LocalResult>  如果是多关键字检索，回调函数参数为 LocalResult 的数组，数组中的结果顺序和检索中多关键字数组中顺序一致。
<code>setMarkersSetCallback(callback:Function)</code>	none	设置添加标注后的回调函数。参数： <b>pois:</b> Array<LocalResultPoi>，通过 <b>marker</b> 属性可得到其对应的标注。
<code>setInfoHtmlSetCallback(callback:Function)</code>	none	设置标注气泡创建时的回调函数。参数： <b>poi:</b> LocalResultPoi，通过其 <b>marker</b> 属性可得到当前的标注。 <b>html:</b> HTMLElement，气泡内的 Dom 元素
<code>setResultsHtmlSetCallback(callback:Function)</code>	none	设置结果列表创建后的回调函数。参数： <b>container:</b> HTMLElement，结果列表所用的 HTML 元素
<code>getStatus()</code>	StatusCodes	返回状态码。

## LocalSearchOptions

此类表示 LocalSearch 构造函数的可选参数。

## 属性

属性	类型	描述
----	----	----

renderOptions	LocalRenderOptions	结果呈现设置。
onMarkersSet	Function	标注添加完成后的回调函数。 参数: pois: Array<LocalResultPoi>, 通过 marker 属性可得到其对应的标注。
onInfoHtmlSet	Function	标注气泡内容创建后的回调函数。 参数: poi: LocalResultPoi, 通过其 marker 属性可得到当前的标注。 html: HTMLElement, 气泡内的 Dom 元素
onResultsHtmlSet	Function	结果列表添加完成后的回调函数。 参数: container: HTMLElement, 结果列表所用的 HTML 元素
pageCapacity	Number	设置每页容量, 取值范围: 1 - 100, 对于多关键字检索, 容量表示每个关键字的数量, 如果有2个关键字, 则实际检索结果数量范围为: 2 - 200。
onSearchComplete	Function	检索完成后的回调函数。 参数: results: LocalResult 或 Array<LocalResult> 如果是多关键字检索, 回调函数参数返回一个 LocalResult 的数组, 数组中的结果顺序和检索中多关键字数组中顺序一致。

## CustomData

此类表示检索lbs云服务的数据。它没有构造函数, 但可通过对象字面量形式表示。  
要检索lbs云服务的数据, 需要在引用api的时候在参数后加上lbs云平台的key。

如<script src="http://api.map.baidu.com/api?v=1.3&key=123456" type="text/javascript"></script>

## 属性

属性	类型	描述
databoxId	Number	可在lbs云平台上查看自己的databoxId

## RenderOptions

此类表示搜索结果呈现的配置。它没有构造函数, 但可通过对象字面量形式表示。

## 属性

属性	类型	描述
map	Map	展现结果的地图实例。当指定此参数后，搜索结果的标注、线路等都会自动添加到此地图上。
panel	String HTML Element	结果列表的 HTML 容器 id 或容器元素，提供此参数后，结果列表将在此容器中进行展示。此属性对 LocalCity 无效。
selectFirstResult	Boolean	是否选择第一个检索结果。此属性仅对 LocalSearch 有效。
autoViewport	Boolean	检索结束后是否自动调整地图视野。此属性对 LocalCity 无效。
highlightMode	HighlightModes	驾车结果展现中点击列表后的展现策略。

## StatusCode

此常量用于描述对象当前状态。

### StatusCode 常量

常量	描述
BMAP_STATUS_SUCCESS	检索成功。对应数值“0”。
BMAP_STATUS_CITY_LIST	城市列表。对应数值“1”。
BMAP_STATUS_UNKNOWN_LOCATION	位置结果未知。对应数值“2”。
BMAP_STATUS_UNKNOWN_ROUTE	导航结果未知。对应数值“3”。
BMAP_STATUS_INVALID_KEY	非法密钥。对应数值“4”。
BMAP_STATUS_INVALID_REQUEST	非法请求。对应数值“5”。
BMAP_STATUS_PERMISSION_DENIED	没有权限。对应数值“6”。(自 1.1 新增)
BMAP_STATUS_SERVICE_UNAVAILABLE	服务不可用。对应数值“7”。(自 1.1 新增)
BMAP_STATUS_TIMEOUT	超时。对应数值“8”。(自 1.1 新增)



## LocalResult

类表示 LocalSearch 的检索结果，没有构造函数，通过 LocalSearch.getResults()方法或 LocalSearch 的 onSearchComplete 回调函数的参数得到。

### 属性

属性	类型	描述
keyword	String	本次检索的关键词。
center	LocalResultPoi	周边检索的中心点（仅当周边检索时提供）。
radius	Number	周边检索的半径（仅当周边检索时提供）。
bounds	Bounds	范围检索的地理区域（仅当范围检索时提供）。
city	String	本次检索所在的城市。
moreResultsUrl	String	更多结果的链接，到百度地图进行搜索。
province	String	本次检索所在的省份。 (自 1.2 新增)
suggestions	Array<String>	搜索建议列表。（当关键词是拼音或拼写错误时给出的搜索建议） (自 1.2 新增)

### 方法

方法	返回值	描述
getPoi(i:Number)	LocalResultPoi	返回索引指定的结果。索引0表示第1条结果
getCurrentNumPois()	Number	返回当前页的结果数
getNumPois()	Number	返回总结果数
getNumPages()	Number	返回总页数
getPageIndex()	Number	返回页数序号
getCityList()	Array<Object>	返回城市列表。数组元素对象包含如下属性： city: String, 城市名 numResults: Number, 结果数

## LocalResultPoi

此类表示位置检索或路线规划的一个结果点，没有构造函数，可通过对象字面量形式表示。

### 属性

属性	类型	描述
title	String	结果的名称标题。
point	Point	该结果所在的地理位置。
url	String	在百度地图详情页面展示该结果点的链接。
detailUrl	String	该字段表示结果点的详情页面。
address	String	地址（根据数据部分提供）。注：当结果点类型为公交站或地铁站时，地址信息为经过该站点的所有车次。
city	String	所在城市。
phoneNumber	String	电话，根据数据部分提供。
postcode	String	邮政编码，根据数据部分提供。
type	enum PoiType	类型，根据数据部分提供。
isAccurate	Boolean	是否精确匹配。只适用 LocalSearch 的 search 方法检索的结果。 (自 1.2 新增)
province	String	所在省份。 (自 1.2 新增)
tags	Array<String>	POI 的标签，如商务大厦、餐馆等。目前只有 LocalSearch 的回调函数 onSearchComplete(result) 中的 result 和 Geocoder.getLocation 的回调函数的参数 GeocoderResult.surroundingPois 涉及的 LocalResultPoi 有 tags 字段。其他 API 涉及的 LocalResultPoi 没有该字段 (自 1.2 新增)

## PoiType

此枚举常量表示地点的类型。

## 常量

常量	描述
BMAP_POI_TYPE_NORMAL	一般位置点
BMAP_POI_TYPE_BUSSTOP	公交车站位置点
BMAP_POI_TYPE_SUBSTOP	地铁车站位置点

## TransitRoute

用于获取公交线路规划方案。

## 构造函数

构造函数	描述
<code>TransitRoute(location:Map Point String[, opts:TransitRouteOptions])</code>	创建一个公交导航实例。 <b>location</b> 表示检索区域，类型可为地图实例、坐标点或城市名称的字符串。当参数为地图实例时，检索位置由当前地图中心点确定；当参数为坐标时，检索位置由该点所在位置确定；当参数为城市名称时，检索会在该城市内进行。

## 方法

方法	返回值	描述
<code>search(start:String Point LocalResultPoi, end:String Point LocalResultPoi)</code>	none	发起检索。 <b>start</b> : 起点，参数可以是关键字、坐标点（自 1.1 版本支持）或者 <code>LocalSearchPoi</code> 实例。 <b>end</b> : 终点，参数可以是关键字、坐标点（自 1.1 版本支持）或者 <code>LocalSearchPoi</code> 实例。
<code>getResults()</code>	<code>TransitRouteResult</code>	返回最近一次检索的结果
<code>clearResults()</code>	none	清除最近一次检索的结果
<code>enableAutoViewport()</code>	none	启用自动调整地图层级，当指定了搜索结果所展现的地图时有效。

<code>disableAutoViewport()</code>	none	禁用自动调整地图层级。
<code>setPageCapacity(capacity:Number)</code>	none	设置每页返回方案个数（1-5），默认为5
<code>setLocation(location:Map Point String)</code>	none	设置检索范围，参数类型可以为地图实例、坐标点或字符串。例： <code>setLocation("北京市")</code>
<code>setPolicy(policy:TransitPolicy)</code>	none	设置路线规划策略，参数为策略常量
<code>setSearchCompleteCallback(callback:Function)</code>	none	设置检索结束后的回调函数。 参数： <code>results: TransitRouteResult</code> ，公交导航结果
<code>setMarkersSetCallback(callback:Function)</code>	none	设置添加标注后的回调函数。 参数： <code>pois: Array&lt;LocalResultPoi&gt;</code> ，起点和目的地数组。 <code>transfers: Array&lt;LocalResultPoi&gt;</code> ，公交车站数组。
<code>setInfoHtmlSetCallback(callback:Function)</code>	none	设置气泡打开后的回调函数。 参数： <code>poi: LocalResultPoi</code> ，表示当前气泡对应的点（可以是起点、终点或换乘车站） <code>html: HTMLElement</code> ，气泡内的 DOM 元素
<code>setPolylinesSetCallback(callback:Function)</code>	none	设置添加路线后的回调函数。 参数： <code>lines: Array&lt;Line&gt;</code> ，公交线路数组。 <code>routes: Array&lt;Route&gt;</code> ，步行线路数组，通过 <code>Route.getPolyline()</code> 方法可得到对应的折线覆盖物
<code>setResultsHtmlSetCallback(callback:Function)</code>	none	设置结果列表创建后的回调函数。 参数： <code>container: 结果列表所用的 HTML 元素</code>
<code>getStatus()</code>	StatusCodes	返回状态码
<code>toString()</code>	String	返回类型说明

## TransitRouteOptions

此类表示 `TransitRoute` 构造函数的可选参数。它没有构造函数，但可通过对象字面量形式表示。

## 属性

属性	类型	描述
renderOptions	TransitRenderOptions	搜索结果的呈现设置。
policy	TransitPolicy	公交导航的策略参数。
pageCapacity	Number	返回方案的个数。
onSearchComplete	Function	检索完成后的回调函数。参数: results: TransitRouteResult, 公交导航结果
onMarkersSet	Function	标注添加完成后的回调函数。参数: pois: Array<LocalResultPoi>, 起点和目的地数组。transfers: Array<LocalResultPoi>, 公交车站数组。
onInfoHtmlSet	Function	气泡内容创建后的回调函数。参数: poi: LocalResultPoi, 表示当前气泡对应的点 (可以是起点、终点或换乘车站) html: HTMLElement, 气泡内的 DOM 元素
onPolylineSet	Function	折线添加完成后的回调函数。参数: lines: Array<Line>, 公交线路数组。routes: Array<Route>, 步行线路数组, 通过 Route.getPolyline() 方法可得到对应的折线覆盖物。
onResultsHtmlSet	Function	结果列表添加完成后的回调函数。参数: container: 结果列表所用的 HTML 元素

## TransitPolicy

此常量表示公交方案的策略。

## 常量

常量	描述
BMAP_TRANSIT_POLICY_LEAST_TIME	最少时间。
BMAP_TRANSIT_POLICY_LEAST_TRANSFER	最少换乘。
BMAP_TRANSIT_POLICY_LEAST_WALKING	最少步行。
BMAP_TRANSIT_POLICY_AVOID_SUBWAYS	不乘地铁。 (自 1.2 新增)

## TransitRouteResult

此类表示路线导航的结果，没有构造函数，通过访问 `TransitRoute.getResults()` 方法或 `TransitRoute` 的 `onSearchComplete` 回调函数参数获得。

### 属性

属性	类型	描述
<code>policy</code>	<code>TransitPolices</code>	公交导航策略
<code>city</code>	<code>String</code>	本次检索所在的城市
<code>moreResultsUrl</code>	<code>String</code>	更多结果的链接，到百度地图进行搜索

### 方法

方法	返回值	描述
<code>getStart()</code>	<code>LocalResultPoi</code>	返回起点
<code>getEnd()</code>	<code>LocalResultPoi</code>	返回终点
<code>getNumPlans()</code>	<code>Number</code>	返回方案个数
<code>getPlan(i:Number)</code>	<code>TransitRoutePlan</code>	返回索引指定的方案。索引0表示第一条方案。

## TransitRoutePlan

此类表示一条公交出行方案。没有构造函数，通过 `TransitRouteResult.getPlan()` 方法获得。

### 方法

方法	返回值	描述
<code>getNumLines()</code>	<code>Number</code>	返回方案包含的公交线路段数。
<code>getLine(i:Number)</code>	<code>Line</code>	返回方案包含的某条公交线路。
<code>getNumRoutes()</code>	<code>Number</code>	返回方案包含的步行线路段数。

<code>getRoute(i:Number)</code>	Route	返回方案包含的某条步行线路。
<code>getDistance([format:Boolean])</code>	String   Number	返回方案总距离。当 <code>format</code> 参数为 <code>true</code> 时，返回方案距离字符串（包含单位），当 <code>format</code> 为 <code>false</code> 时，仅返回数值（单位为米）信息。默认参数为 <code>true</code> 。
<code>getDuration([format:Boolean])</code>	String   Number	返回方案总时间。当 <code>format</code> 参数为 <code>true</code> 时，返回描述时间的字符串（包含单位），当 <code>format</code> 为 <code>false</code> 时，仅返回数值（单位为秒）信息。默认参数为 <code>true</code> 。（自 1.1 新增）
<code>getDescription([includeHtml:Boolean])</code>	String	返回方案描述文本，默认包含 HTML 标签。当 <code>includeHtml</code> 为 <code>false</code> 时，方案描述不包含 HTML 标签。

## Line

此类表示一条公交线路。没有构造函数，通过 `TransitRoutePlan.getLine()` 方法得到。

## 属性

属性	类型	描述
<code>title</code>	String	公交线路全称
<code>type</code>	enum LineType	公交线路类型

## 方法

方法	返回值	描述
<code>getNumViaStops()</code>	Number	返回公交线路途径的车站个数。
<code>getGetOnStop()</code>	LocalResultPoi	返回上车站。
<code>getGetOffStop()</code>	LocalResultPoi	返回下车站。
<code>getPoints()</code>	Array<Point>	返回线路对应的地理坐标点数组，在公交方案中，地理坐标只给出方案涉及的部分。（自 1.2 废弃）
<code>getPath()</code>	Array<Point>	返回线路对应的地理坐标点数组，在公交方案中，地理坐标只给出方案涉及的部分。（自 1.2 新增）
<code>getPolyline()</code>	Polyline	返回公交线路对应的折线覆盖物。

<code>getDistance([format:Boolean])</code>	String Number	当 <code>format</code> 为 <code>true</code> 时，返回本段公交线路的距离字符串（包含单位），当 <code>format</code> 为 <code>false</code> 时仅返回数值（单位为米）。默认参数为 <code>true</code> 。
--	---------------	--

## LineType

此枚举类型表示不同类型的公共交通线路。

## 常量

常量	描述
<code>BMAP_LINE_TYPE_BUS</code>	公交车
<code>BMAP_LINE_TYPE_SUBWAY</code>	地铁
<code>BMAP_LINE_TYPE_FERRY</code>	渡轮

## DrivingRoute

此类用于获取驾车路线规划方案。

## 构造函数

构造函数	描述
<code>DrivingRoute(location:Map Point String[, opts:DrivingRouteOptions])</code>	创建一个驾车导航实例，其中 <code>location</code> 表示检索区域，类型可为地图实例、坐标点或城市名称的字符串。当参数为地图实例时，检索位置由地图当前的中心点确定；当参数为坐标时，检索位置由该点所在位置确定；当参数为城市名称时，检索会在该城市内进行。

## 方法

方法	返回值	描述
<code>search(start:String Point LocalResultPoi, end:String Point LocalResultPoi[,opti</code>	none	发起检索。  <b>start:</b> 起点，参数可以是关键字、坐标点（自 1.1 版本支持）和 <code>LocalSearchPoi</code> 实例。



<code>ons:object[])</code>		<p><b>end:</b> 终点, 参数可以是关键字、坐标点（自 1.1版本支持）或 <code>LocalSearchPoi</code> 实例。</p> <p><b>option:</b>{startCity:String,endCity:string}</p> <p><b>startCity</b> 表示是驾车查询的起点城市, 可以是城市名或者城市编码, <b>endCity</b> 表示驾车查询的终点城市, 可以是城市名或者城市编码。</p>
<code>getResults()</code>	DrivingRouteResult	返回最近一次检索的结果
<code>clearResults()</code>	none	清除最近一次检索的结果
<code>enableAutoViewport()</code>	none	启用自动调整地图层级, 当指定了搜索结果所展现的地图时有效。
<code>disableAutoViewport()</code>	none	禁用自动调整地图层级。
<code>setLocation(location:Map Point String)</code>	none	设置检索范围, 参数类型可以为地图实例、坐标点或字符串。例: <code>setLocation("北京市")</code>
<code>setPolicy(policy:DrivingPolicy)</code>	none	设置路线规划策略, 参数为策略常量
<code>setSearchCompleteCallback(callback:Function)</code>	none	<p>设置检索结束后的回调函数。</p> <p>参数: <code>results: DrivingRouteResult</code></p>
<code>setMarkersSetCallback(callback:Function)</code>	none	<p>设置添加标注后的回调函数。</p> <p>参数: <code>pois: Array&lt;LocalResultPoi&gt;</code>, 起点和目的地点数组, 通过 <code>marker</code> 属性可得到其对应的标注</p>
<code>setInfoHtmlSetCallback(callback:Function)</code>	none	<p>设置气泡打开后的回调函数。</p> <p>参数: <code>poi: LocalResultPoi</code>, 通过 <code>marker</code> 属性可得到当前的标注。html: <code>HTMLElement</code>, 气泡内的 DOM 元素。</p>
<code>setPolylinesSetCallback(callback:Function)</code>	none	<p>设置添加路线后的回调函数。</p> <p>参数: <code>routes: Array&lt;Route&gt;</code>, 驾车线路数组, 通过 <code>Route.getPolyline()</code>方法可得到对应的折线覆盖物。</p>
<code>setResultsHtmlSetCallback(callback:Function)</code>	none	<p>设置结果列表创建后的回调函数。</p> <p>参数: <code>container:</code> 结果列表所用的 HTML 元素。</p>
<code>getStatus()</code>	StatusC	返回状态码

	odes	
toString()	String	返回类型说明

## DrivingRouteOptions

此类表示 DrivingRoute 构造函数的可选参数。

### 属性

属性	类型	描述
renderOptions	DrivingRenderOptions	结果呈现设置。
policy	DrivingPolicy	驾车策略
onSearchComplete	Function	检索完成后的回调函数。参数： results: DrivingRouteResult
onMarkersSet	Function	标注添加完成后的回调函数。 参数： pois: Array<LocalResultPoi>，起点和目的地点数组，通过 marker 属性可得到其对应的标注。
onInfoHtmlSet	Function	标注气泡内容创建后的回调函数。 参数： poi: LocalResultPoi，通过 marker 属性可得到当前的标注。 html: HTMLElement，气泡内的 DOM 元素。
onPolylineSets	Function	折线添加完成后的回调函数。 参数： routes: Array<Route>，驾车线路数组，通过 Route.getPolyline()方法可得到对应的折线覆盖物。
onResultsHtmlSet	Function	结果列表添加完成后的回调函数。 参数： container: 结果列表所用的 HTML 元素。

## DrivingPolicy

此枚举类型表示驾车方案的策略配置。

## 常量

常量	描述
BMAP_DRIVING_POLICY_LEAST_TIME	最少时间。
BMAP_DRIVING_POLICY_LEAST_DISTANCE	最短距离。
BMAP_DRIVING_POLICY_AVOID_HIGHWAYS	避开高速。

## DrivingRouteResult

此类表示路线导航的结果，没有构造函数，通过 `DrivingRoute.getResults()` 方法或 `DrivingRoute` 的 `onSearchComplete` 回调函数参数获得。

## 属性

属性	类型	描述
<code>policy</code>	enum <code>DrivingPolicy</code>	驾车导航策略。
<code>city</code>	<code>String</code>	本次检索所在的城市。
<code>moreResultsUrl</code>	<code>String</code>	更多结果的链接，到百度地图进行搜索。
<code>taxiFare</code>	<code>TaxiFare</code>	出租车费用信息。注意，此属性在部分城市有效，无效时为 <code>null</code> 。（自 1.1 新增）

## 方法

方法	返回值	描述
<code>getStart()</code>	<code>LocalResultPoi</code>	返回起点。
<code>getEnd()</code>	<code>LocalResultPoi</code>	返回终点。
<code>getNumPlans()</code>	<code>Number</code>	返回方案个数。
<code>getPlan(i:Number)</code>	<code>RoutePlan</code>	返回索引指定的方案。索引0表示第一条方案。

## TaxiFare

此类表示出租车费用信息，没有构造函数，通过对象字面量形式表示。

### 属性

属性	类型	描述
day	TaxiFareDetail	白天费用。
night	TaxiFareDetail	夜间费用。注意，部分城市没有夜间费用，此时此属性为 null，且同时白天费用表示全天费用。
distance	Number	出租车里程，单位为米。
remark	String	出租车备注信息。

## TaxiFareDetail

此类表示出租车具体费用信息，没有构造函数，通过对象字面量形式表示。

### 属性

属性	类型	描述
initialFare	Number	出租车起步价。
unitFare	Number	出租车单价。
totalFare	Number	出租车费用总价。

## RoutePlan

此类表示一条驾车或步行出行方案。它没有构造函数，可通过 `DrivingRouteResult.getPlan()` 方法或 `WalkingRouteResult` 类的 `getPlan()` 方法获得。

## 方法

方法	返回值	描述
<code>getNumRoutes()</code>	<code>Number</code>	返回方案包含的线路的个数。
<code>getRoute(i:Number)</code>	<code>Route</code>	返回方案中索引指定的线路。索引0表示第一条线路。
<code>getDistance([format:Boolean])</code>	<code>String Number</code>	返回方案总距离。当 <code>format</code> 参数为 <code>true</code> 时，返回方案距离字符串（包含单位），当 <code>format</code> 为 <code>false</code> 时，仅返回数值（单位为米）信息。默认参数为 <code>true</code> 。
<code>getDuration([format:Boolean])</code>	<code>String Number</code>	返回方案总时间。当 <code>format</code> 参数为 <code>true</code> 时，返回描述时间的字符串（包含单位），当 <code>format</code> 为 <code>false</code> 时，仅返回数值（单位为秒）信息。默认参数为 <code>true</code> 。 (自 1.1 新增)
<code>getDragPois()</code>	<code>Array&lt;LocalResultPoi&gt;</code>	获取通过拖拽方式产生的途径点。注意，这里的 <code>LocalResultPoi</code> 只有 <code>title</code> 和 <code>point</code> 属性。 (自 1.2 新增)

## Route

此类表示一条驾车或步行路线。

## 方法

方法	返回值	描述
<code>getNumSteps()</code>	<code>Number</code>	返回路线包含的关键点个数。
<code>getStep(i:Number)</code>	<code>Step</code>	返回索引指定的关键点，驾车和步行适用。索引0表示第一个关键点。
<code>getDistance(format:Boolean)</code>	<code>String Number</code>	返回路线距离，当 <code>format</code> 为 <code>false</code> 时仅返回数值。
<code>getIndex()</code>	<code>Number</code>	返回本路线在方案中的索引位置。
<code>getPolyline()</code>	<code>Polyline</code>	返回路线对应的覆盖物，仅当结果自动添加到地图上时有效。
<code>getPoints()</code>	<code>Array&lt;Point&gt;</code>	返回路线的地理坐标点数组。(自 1.2 废弃)

<code>getPath()</code>	<code>Array&lt;Point&gt;</code>	返回路线的地理坐标点数组。（自 1.2 新增）
<code>getRouteType()</code>	<code>RouteTypes</code>	返回路线类型，可区分是驾车还是步行线路。

## RouteType

常量表示不同的线路类型。

### 常量

常量	描述
<code>BMAP_ROUTE_TYPE_DRIVING</code>	驾车线路
<code>BMAP_ROUTE_TYPE_WALKING</code>	步行线路

## Step

此类表示驾车或步行路线中的一个关键点。它没有构造函数，通过 `Route.getStep()` 方法获得。

### 方法

方法	返回值	描述
<code>getPoint()</code>	<code>Point</code>	返回关键点地理坐标。
<code>getPosition()</code>	<code>Point</code>	返回关键点地理坐标。（自 1.2 新增）
<code>getIndex()</code>	<code>Number</code>	返回本关键点在路线中的位置索引
<code>getDescription(includeHtml:Boolean)</code>	<code>String</code>	返回关键点描述文本，默认包含 HTML 标签。当 <code>includeHtml</code> 为 <code>false</code> 时，描述文本不包含 HTML 标签。
<code>getDistance(format:Boolean)</code>	<code>String</code>   <code>Number</code>	返回到下一个关键点的距离，当 <code>format</code> 为 <code>false</code> 时仅返回数值（单位为米）。

## WalkingRoute

用于获取步行路线规划方案。

## 构造函数

构造函数	描述
<code>WalkingRoute(location:Map Point String[, opts:WalkingRouteOptions])</code>	创建一个步行导航实例。 <b>location</b> 表示检索区域，类型可为地图实例、坐标点或城市名称的字符串。当参数为地图实例时，检索位置由当前地图中心点确定；当参数为坐标时，检索位置由该点所在位置确定；当参数为城市名称时，检索会在该城市内进行。

## 方法

方法	返回值	描述
<code>search(start:String Point LocalResultPoi, end:String Point LocalResultPoi)</code>	none	发起检索。  <b>start</b> : 起点，参数可以是关键字、坐标点（自 1.1 版本支持）或者 <code>LocalSearchPoi</code> 实例。 <b>end</b> : 终点，参数可以是关键字、坐标点（自 1.1 版本支持）或者 <code>LocalSearchPoi</code> 实例。
<code>getResults()</code>	<code>WalkingRouteResult</code>	返回最近一次检索的结果
<code>clearResults()</code>	none	清除最近一次检索的结果
<code>enableAutoViewport()</code>	none	启用自动调整地图层级，当指定了搜索结果所展现的地图时有效。
<code>disableAutoViewport()</code>	none	禁用自动调整地图层级。
<code>setLocation(location:Map Point String)</code>	none	设置检索范围，参数类型可以为地图实例、坐标点或字符串。例： <code>setLocation("北京市")</code>
<code>setSearchCompleteCallback(callback:Function)</code>	none	设置检索结束后的回调函数。  参数： <code>results: WalkingRouteResult</code> 。
<code>setMarkersSetCallback(callback:Function)</code>	none	设置添加标注后的回调函数。  参数： <code>pois: Array&lt;LocalResultPoi&gt;</code> ，起点和目的地点数组。通过 <code>marker</code> 属性可得到其对应的标注。
<code>setInfoHtmlSetCallback(callback:Function)</code>	none	设置气泡打开后的回调函数。  参数： <code>poi: LocalResultPoi</code> ，通过其 <code>marker</code>

		属性可得到当前的标注。 <code>html</code> : <code>HTMLElement</code> , 气泡内的 DOM 元素。
<code>setPolylinesSetCallback(callback: Function)</code>	<code>none</code>	设置添加路线后的回调函数。 参数: <code>routes: Array&lt;Route&gt;</code> , 步行线路数组, 通过 <code>Route.getPolyline()</code> 方法可得到对应的折线覆盖物。
<code>setResultsHtmlSetCallback(callback: Function)</code>	<code>none</code>	设置结果列表创建后的回调函数。 参数: <code>container</code> : 结果列表所用的 HTML 元素
<code>getStatus()</code>	<code>StatusCodes</code>	返回状态码
<code>toString()</code>	<code>String</code>	返回类型说明

## WalkingRouteOptions

此类表示 `WalkingRoute` 构造函数的可选参数。

## 属性

属性	类型	描述
<code>renderOptions</code>	<code>WalkingRenderOptions</code>	搜索结果呈现设置。
<code>onSearchComplete</code>	<code>Function</code>	检索完成后的回调函数。 参数: <code>results: WalkingRouteResult</code>
<code>onMarkersSet</code>	<code>Function</code>	标注添加完成后的回调函数。 参数: <code>pois: Array&lt;LocalResultPoi&gt;</code> , 起点和目的地点数组,。通过 <code>marker</code> 属性可得到其对应的标注。
<code>onPolylinesSet</code>	<code>Function</code>	折线添加完成后的回调函数。 参数: <code>routes: Array&lt;Route&gt;</code> , 步行线路数组, 通过 <code>Route.getPolyline()</code> 方法可得到对应的折线覆盖物。
<code>onInfoHtmlSet</code>	<code>Function</code>	标注气泡内容创建后的回调函数。 参数: <code>poi: LocalResultPoi</code> , 通过其 <code>marker</code> 属性可得到当前的标



		注。 html: HTMLElement, 气泡内的 DOM 元素。
onResultsH tmlSet	Function	结果列表添加完成后的回调函数。  参数： container: 结果列表所用的 HTML 元素

## WalkingRouteResult

此类表示路线导航的结果，没有构造函数，通过访问 `WalkingRoute.getResults()` 方法或 `WalkingRoute` 的 `onSearchComplete` 回调函数参数获得。

### 属性

属性	类型	描述
city	String	本次检索所在的城市

### 方法

方法	返回值	描述
<code>getStart()</code>	LocalResultPoi	返回起点。
<code>getEnd()</code>	LocalResultPoi	返回终点。
<code>getNumPlans()</code>	Number	返回方案个数。
<code>getPlan(i:Number)</code>	RoutePlan	返回索引指定的方案。索引0表示第一条方案。

## Geocoder

类用于获取用户的地址解析。

### 构造函数

构造函数	描述
<code>Geocoder()</code>	创建一个地址解析器的实例

## 方法

方法	返回值	描述
<code>getPoint(address:String, callback:Function, city:String)</code>	none	对指定的地址进行解析。如果地址定位成功，则以地址所在的坐标点 <code>Point</code> 为参数调用回调函数。否则，回调函数的参数为 <code>null</code> 。 <code>city</code> 为地址所在的城市名，例如“北京市”。
<code>getLocation(point:Point, callback:function[, options:LocationOptions])</code>	none	对指定的坐标点进行反向地址解析。如果解析成功，则回调函数的参数为 <code>GeocoderResult</code> 对象，否则回调函数的参数为 <code>null</code> 。(自 1.1 新增)

## GeocoderResult

此类表示 `Geocoder` 的地址解析结果。它在地址解析的回调函数的参数中返回，不可实例化。

## 属性

属性	类型	描述
<code>point</code>	<code>Point</code>	坐标点。(自 1.1 新增)
<code>address</code>	<code>String</code>	地址描述。(自 1.1 新增)
<code>addressComponents</code>	<code>AddressComponent</code>	结构化的地址描述。(自 1.1 新增)
<code>surroundingPois</code>	<code>Array&lt;LocalResultPoi&gt;</code>	附近的 POI 点。(自 1.1 新增)
<code>business</code>	<code>String</code>	商圈字段，代表此点所属的商圈 (自 1.1 新增)

## AddressComponent

此类表示地址解析结果的层次化地址信息，没有构造函数，通过对象字面量形式表示。

## 属性

属性	类型	描述
streetNumber	String	门牌号码。
street	String	街道名称。
district	String	区县名称。
city	String	城市名称。
province	String	省份名称。

## LocationOptions

此类表示 Geocoder 的地址解析请求的可选参数。它不可实例化。

## 属性

属性	类型	描述
poiRadius	Number	附近 POI 所处于的最大半径，默认为100米。(自 1.1 新增)
numPois	Number	返回的 POI 点个数，默认为10个。取值范围(自 1.1 新增)

## LocalCity

此类用于获取用户所在的城市位置信息。(根据用户 IP 自动定位到城市)

## 构造函数

构造函数	描述
LocalCity([opts:LocalCityOptions])	创建一个获取本地城市位置的实例

## 方法

方法	返回值	描述
----	-----	----

<code>get(callback:Function)</code>	<code>none</code>	当获取城市信息后，回调函数会被调用，其参数为类型为 <code>LocalCityResult</code> 对象。
-------------------------------------	-------------------	--

## LocalCityOptions

此类表示 `LocalCity` 的可选参数。它没有构造函数，但可通过对象字面量表示。

### 属性

属性	类型	描述
<code>renderOptions</code>	<code>LocalRenderOptions</code>	结果呈现设置，当给定 <code>map</code> 参数时，改地图将自动将视野定位到当前城市。

## LocalCityResult

此类表示 `LocalCity` 的定位结果。

### 属性

属性	类型	描述
<code>center</code>	<code>LocalResultPoi</code>	城市所在中心点。
<code>level</code>	<code>Number</code>	展示当前城市的最佳地图级别，如果您在使用此对象时提供了 <code>map</code> 实例，则地图级别将根据您提供的地图大小进行调整。
<code>name</code>	<code>String</code>	城市名称。

## TrafficControl

此类表示交通流量控件，它继承 `Control` 类，包含该类的所有方法。该控件的停靠位置常量仅支持 `BMAP_CONTROL_ANCHOR_TOP_RIGHT`，但可修改其偏移位置（自 1.2 废弃）。

## 构造函数

构造函数	描述
<code>TrafficControl()</code>	创建一个交通流量的控件实例，该实例继承自 <b>Control</b> ，支持此类的所有方法。

## 方法

方法	返回值	描述
<code>setPanelOffset(offset:Size)</code>	none	设置路况面板的水平和垂直偏移位置。面板与按钮停靠位置一致，始终位于地图区域的右上角。
<code>show()</code>	none	显示实时路况。
<code>hide()</code>	none	隐藏实时路况。

## Geolocation

返回用户当前的位置。此方法利用浏览器的 **geolocation** 接口获取用户当前位置，不支持的浏览器将无法获取。(自 1.1 新增)

## 构造函数

构造函数	描述
<code>Geolocation()</code>	创建 <b>Geolocation</b> 对象实例。

## 方法

方法	返回值	描述
<code>getCurrentPosition(callback:function[, options:PositionOptions])</code>	none	返回用户当前位置。当定位成功时，回调函数的参数为 <b>GeolocationResult</b> 对象，否则为 <b>null</b> 。
<code>getStatus()</code>	StatusCode	返回状态码，当定位成功后，状态码为： <b>BMAP_STATUS_SUCCESS</b> ，如果为其他状态码表示不能获取您当前的位置。

## GeolocationResult

此类作为 Geolocation 的 `getCurrentPosition` 方法的回调函数参数，不可实例化(自 1.1 新增)。

### 属性

属性	类型	描述
<code>point</code>	<code>Point</code>	坐标点。
<code>accuracy</code>	<code>Number</code>	定位精确程度，单位为米。

## PositionOptions

此类作为 `getCurrentPosition` 的可选参数，不能实例化(自 1.1 新增)。

### 属性

属性	类型	描述
<code>enableHighAccuracy</code>	<code>Boolean</code>	要求浏览器获取最佳结果。
<code>timeout</code>	<code>Number</code>	超时时间。
<code>maximumAge</code>	<code>Number</code>	允许返回指定时间内的缓存结果。如果此值为0，则浏览器将立即获取新定位结果。

## HighlightModes

此常量用于描述对象当前状态。

### 常量

常量	描述
<code>BMAP_HIGHLIGHT_STEP</code>	驾车结果展现中点击列表后的展现点步骤。

BMAP\_HIGHLIGHT\_ROUTE

驾车结果展现中点击列表后的展现路段。

## BusLineSearch

公交线路搜索类。

### 构造函数

构造函数	描述
<code>BusLineSearch(location:Map Point String[, options:BusLineSearchOptions])</code>	创建公交线搜索类。其中 <b>location</b> 表示检索区域，其类型可为地图实例、坐标点或城市名称的字符串。当参数为地图实例时，检索位置由当前地图中心点确定；当参数为坐标时，检索位置由该点所在位置确定；当参数为城市名称时，检索会在该城市内进行。 (自 1.2 新增)

### 方法

方法	返回值	描述
<code>getBusList(keyword:String)</code>	none	在用户配置的回调函数中返回公交列表结果，其类型为 <b>BusListResult</b> (自 1.2 新增)
<code>getBusLine(busListItem:BusListItem)</code>	none	在用户配置的回调函数中返回该条线路的公交信息，其类型为 <b>BusLine</b> 类型(自 1.2 新增)
<code>clearResults()</code>	none	清除本次公交线检索结果(自 1.2 新增)
<code>enableAutoViewport()</code>	none	启用自动调整地图视野功能(自 1.2 新增)
<code>disableAutoViewport()</code>	none	禁用自动调整地图视野功能(自 1.2 新增)
<code>setLocation(location:Map Point String)</code>	none	设置检索范围，参数类型可以为地图实例、坐标点或字符串。例： <code>setLocation("北京市")</code> (自 1.2 新增)
<code>getStatus()</code>	Status Codes	返回状态码(自 1.2 新增)
<code>toString()</code>	String	返回类型说明(自 1.2 新增)
<code>setGetBusListCompleteCallback(callback:Function)</code>	none	设置公交列表查询后的回调函数。参数： <b>rs</b> : <b>BusListResult</b> 类型(自 1.2 新增)
<code>setGetBusLineCompleteCallback</code>	none	设置公交线路查询后的回调函数。参数： <b>rs</b> : <b>BusLine</b> 类

ack(callback:Function)		型(自 1.2 新增)
setBusListHtmlSetCallback(callback:Function)	none	公交列表结果页渲染后回调函数。参数: container: HTMLElement, 结果列表所用的 HTML 元素(自 1.2 新增)
setBusLineHtmlSetCallback(callback:Function)	none	公交线路结果页渲染后回调函数。参数: container: HTMLElement, 结果列表所用的 HTML 元素(自 1.2 新增)
setPolylinesSetCallback(callback:Function)	none	添加公交线时候回调函数。参数: ply:Polyline 公交线路几何对象(自 1.2 新增)
setMarkersSetCallback(callback:Function)	none	添加公交站点时候回调函数。参数: sts:Array<Marker> 公交站坐标组成的 Marker 对象数组(自 1.2 新增)

## BusLineSearchOptions

### 属性

属性	类型	描述
renderOptions	RenderOptions	RenderOptions 结果呈现设置。(自 1.2 新增)
onGetBusListComplete	Function	设置公交列表查询后的回调函数。参数: rs: BusListResult 类型(自 1.2 新增)
onGetBusLineComplete	Function	设置公交线路查询后的回调函数。参数: rs: BusLine 类型(自 1.2 新增)
onBusListHtmlSet	Function	公交列表结果页渲染后回调函数。参数: container: HTMLElement, 结果列表所用的 HTML 元素(自 1.2 新增)
onBusLineHtmlSet	Function	公交线路结果页渲染后回调函数。参数: container: HTMLElement, 结果列表所用的 HTML 元素(自 1.2 新增)
onPolylinesSet	Function	添加公交线时候回调函数。参数: ply:Polyline 公交线路几何对象(自 1.2 新增)
onMarkersSet	Function	添加公交站点时候回调函数。参数: sts:Array<Marker> 公交站坐标组成的 Marker 对象数组(自 1.2 新增)



## BusListResult

### 属性

属性	返回值	描述
keyword	String	本次检索关键字(自 1.2 新增)
city	String	本次检索所在城市(自 1.2 新增)
moreResultsUrl	String	到百度地图查看 url(自 1.2 新增)

### 方法

方法	返回值	描述
getNumBusList()	Number	公交列表个数(自 1.2 新增)
getBusListItem(i: Number)	BusListItem	获取某一个具体的公交列表中的对象。0表示上行，1表示下行。(自 1.2 新增)

## BusLine

表示公交线路结果的公交线，没有构造函数，通过检索回调函数获得。

### 属性

属性	返回值	描述
name	String	线路名称(自 1.2 新增)
startTime	String	首班车时间(自 1.2 新增)
endTime	String	末班车时间(自 1.2 新增)
company	String	公交线路所属公司(自 1.2 新增)

### 方法

方法	返回值	描述
----	-----	----

<code>getNumBusStations()</code>	Number	获取公交站点个数(自 1.2 新增)
<code>getBusStation(i: Number)</code>	BusStation	获取某一个具体的公交站对象(自 1.2 新增)
<code>getPath()</code>	Array<Point>	返回公交线地理坐标点数组。(自 1.2 新增)
<code>getPolyline()</code>	Polyline	获取公交线几何对象, 仅当结果自动添加到地图上时有效(自 1.2 新增)

## BusListItem

### 属性

属性	返回值	描述
name	String	公交线名称(自 1.2 新增)

## BusStation

### 属性

属性	返回值	描述
name	String	站点名称(自 1.2 新增)
position	Point	站点坐标(自 1.2 新增)

## Autocomplete

Autocomplete 是结果提示、自动完成类。

### 构造函数

构造函数	描述
<code>Autocomplete([options:AutocompleteOptions])</code>	创建自动完成的实例。

## 方法

方法	返回值	描述
<code>show()</code>	<code>none</code>	显示提示列表
<code>hide()</code>	<code>none</code>	隐藏提示列表
<code>setTypes(types:type&lt;Array&gt;)</code>	<code>none</code>	修改请求数据类型。
<code>setLocation(location:String  Map Point)</code>	<code>none</code>	设置检索区域
<code>search(keywords:string)</code>	<code>none</code>	发起某个关键字的提示请求，会引起 <code>onSearchComplete</code> 的回调
<code>getResults()</code>	<code>AutocompleteResult</code>	获取结果列表
<code>setInputValue()</code>	<code>none</code>	设置绑定的 <code>input</code> 控件的值，且不会出现下拉列表。  (自 1.3 新增)
<code>dispose()</code>	<code>none</code>	销毁自动完成对象。(自 1.3 新增)

## 事件

事件	参数	描述
<code>onconfirm</code>	<code>{type,target,item}</code>	回车选中某条记录后触发  <code>item</code> : {  <code>index</code> : 1 /*高亮的记录，所属返回结果的 index*/ <code>,value</code> : {}/*结果数据，见 <code>AutocompleteResultPoi*/</code> }
<code>onhighlight</code>	<code>{type,target,fromitem,toitem}</code>	键盘或者鼠标移动，某条记录高亮之后，触发  <code>fromitem</code> : { /*上一条记录的信息*/  <code>index</code> : 2 /*高亮的记录，所属返回结果的 index*/ <code>,value</code> : {}/*结果数据，见 <code>AutocompleteResultPoi*/</code> },  <code>toitem</code> : {}/*当前记录的信息，与 <code>fromitem</code> 结构一

		致*/}
--	--	------

## AutocompleteOptions

### 属性

属性	类型	描述
location	String Map Point	设定返回结果的所属范围。例如“北京市”。
types	Array<String>	返回数据类型。默认值为所有数据 “city”：城市数据
onSearchComplete	Function	在 input 框中输入字符后，发起列表检索，完成后的回调函数。 参数：AutocompleteResult
input	String HTMLElement	文本输入框元素或其 id

## AutocompleteResultPoi

### 属性

属性	类型	描述
province	String	省名
City	String	城市名
district	String	区县名称
street	String	街道名称
streetNumber	String	门牌号码
business	String	商户名

## AutocompleteResult

### 属性

属性	类型	描述
keyword	String	检索关键字

### 方法

方法	返回值	描述
getPoi()	Array<AutocompleteResultPoi>	结果数组
getNumPois()	Number	结果总数

## Boundary

此类表示一个行政区域的边界。

### 构造函数

构造函数	描述
Boundary()	创建行政区域搜索的对象实例。

### 方法

方法	返回值	描述
get(name:String, callback:function)	null	返回行政区域的边界。  name: 查询省、直辖市、地级市、或县的名称。  callback: 执行查询后，数据返回到客户端的回调函数，数据以回调函数的参数形式返回。返回结果是一个数组，数据格式如下：  arr[0] = "x1, y1; x2, y2; x3, y3; ..."  arr[1] = "x1, y1; x2, y2; x3, y3; ..."

		<pre>arr[2] = "x1, y1; x2, y2; ..."</pre> <p>...</p> <p>否则回调函数的参数为 <code>null</code>。(自1.3自增)</p>
--	--	---