

百度地图 API 类参考

目录

核心类	3
Map	3
MapOptions	9
Viewport	10
ViewportOptions	10
MapTypes	11
MapPanes	11
APIVersion	11
PanOptions	12
基础类	12
Point	12
Pixel	13
Bounds	13
Size	14
控件类	15
ControlNavigation	15
ControlOptions	16
ScaleControlOptions	16
CopyrightControlOptions	17
ControlAnchor	17
OverviewMapControl	17
LengthUnit	18
MapTypeControl	18
NavigationControl	18
OverviewMapControlOptions	19
CopyrightControl	19
NavigationControlType	20
ScaleControl	20
Copyright	20
覆盖物类	21
Overlay	21
Marker	21
LabelOptions	24
Polygon	24
IconOptions	26
InfoWindow	27
MarkerOptions	28
Polyline	28
PolygonOptions	30
Circle	31

InfoWindowOptions	32
Label	33
PolylineOptions	34
Icon	35
CircleOptions	36
右键菜单类	37
ContextMenu	37
MenuItem	37
MenuItemOptions	38
工具类	38
PushpinTool	38
PushpinToolOptions	39
DistanceTool	39
DragAndZoomTool	40
DragAndZoomTool	41
OptionsZoomType	41
地图图层类	41
TileLayer	41
TileLayerOptions	42
MapType	42
MapTypeOptions	43
TrafficLayer	43
TrafficLayerOptions	44
服务类	44
LocalSearch	44
TransitRoute	45
DrivingRoute	47
Step	48
LocalCity	48
LocalSearchOptions	49
TransitRouteOptions	49
DrivingRouteOptions	50
WalkingRoute	51
LocalCityOptions	52
RenderOptions	52
TransitPolicy	53
DrivingPolicy	53
WalkingRouteOptions	53
LocalCityResult	54
StatusCode	55
TransitRouteResult	55
DrivingRouteResult	56
WalkingRouteResult	56
TrafficControl	57

LocalResult	57
TransitRoutePlan	58
RoutePlan	58
Geocoder	59
Geolocation	60
LocalResultPoi	60
Line	61
Route	61
GeocoderResult	62
GeolocationResult	62
PoiType	63
LineType	63
RouteType	63
LocationOptions	64
PositionOptions	64
TaxiFare	64
TaxiFareDetail	65

核心类

Map

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>oom()</code>		
<code>enableDoubleClickZoom()</code>	<code>none</code>	启用双击放大，默认启用。
<code>disableDoubleClickZoom()</code>	<code>none</code>	禁用双击放大。
<code>enableKeyboard()</code>	<code>none</code>	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 $\frac{1}{2}$ 的大小。+、-键会使地图放大或缩小一级。
<code>disableKeyboard()</code>	<code>none</code>	禁用键盘操作。
<code>enableInertialDragging()</code>	<code>none</code>	启用地图惯性拖拽（默认为关闭状态）。 (自 1.1 新增)
<code>disableInertialDragging()</code>	<code>none</code>	禁用地图惯性拖拽 (自 1.1 新增)
<code>enableContinuousZoom()</code>	<code>none</code>	启用连续缩放效果（默认为禁用状态）。 (自 1.1 新增)
<code>disableContinuousZoom()</code>	<code>none</code>	禁用连续缩放效果 (自 1.1 新增)
<code>enablePinchToZoom()</code>	<code>none</code>	启用双指操作缩放（默认为开启状态）。 (自 1.1 新增)
<code>disablePinchToZoom()</code>	<code>none</code>	禁用双指操作缩放。 (自 1.1 新增)
<code>setDefaultCursor(cursor:String)</code>	<code>none</code>	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>	<code>String</code>	返回地图默认的鼠标指针样式。 (自 1.1 新增)
<code>setDraggingCursor(cursor:String)</code>	<code>none</code>	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>	<code>String</code>	返回拖拽地图时的鼠标指针样式。 (自 1.1 新增)
<code>highResolutionEnabled()</code>	<code>Boolean</code>	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> 。目前仅支持 iPhone4 及其后续机型。 (自 1.2 新增)

地图状态方法

方法	返回值	描述
<code>getBounds()</code>	<code>Bounds</code>	返回地图可视区域，以地理坐标表示。
<code>getCenter()</code>	<code>Point</code>	返回地图当前中心点。
<code>getDistance(start:Point, end:Point)</code>	<code>Number</code>	返回两点之间的距离，单位是米。 (自 1.1 新增)
<code>getMapType()</code>	<code>MapTypes</code>	返回地图类型。 (自 1.1 新增)
<code>getSize()</code>	<code>Size</code>	返回地图视图的大小，以像素表示。
<code>getViewport(view:Array<Point>[, viewportOptions:ViewportOptions])</code>	<code>Viewport</code>	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 <code>center</code> 和 <code>zoom</code> 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>	<code>Number</code>	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 String, zoom:Number)</code>	<code>no</code>	设置地图中心点地理坐标，并改变缩放等级。在创建地图实例后应立即调用此方法对地图进行初始化。 如果 <code>center</code> 类型为字符串时（自 1.2 版本支持），比如“北京”， <code>zoom</code> 可以忽略，地图将自动根据 <code>center</code> 调整缩放级别。
<code>panTo(center:Point[, opts:PanOptions])</code>	<code>no</code>	将地图的中心点更改为给定的点。如果该点在当前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（自 1.2 版本支持）。
<code>panBy(x:Number, y:Number[, opts:PanOptions])</code>	<code>no</code>	将地图在水平位置上移动 <code>x</code> 像素，垂直位置上移

<code>opts:PanOptions])</code>	ne	动 y 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。可以通过配置强制移动过程不使用动画效果（自 1.2 版本支持）。
<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	设置地图中心位置。自 1.2 版本开始， center 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 BMAP_PERSPECTIVE_MAP 时，需要在调用 centerAndZoom 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 BMAP_NORMAL_MAP 类型，则在调用 setMapType 切换地图类型时也要调用此方法。</p> <p>(自 1.1 新增)</p>
<code>setMapType(mapType:MapTypes)</code>	no ne	<p>设置地图类型。注意，当设置地图类型为 BMAP_PERSPECTIVE_MAP 时，需要调用 map.setCurrentCity 方法设置城市。</p> <p>(自 1.1 新增)</p>
<code>setViewport(view:Array<Point> Viewport[, viewportOptions:ViewportOptions])</code>	no ne	<p>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。</p> <p>(自 1.1 新增)</p>
<code>zoomTo(zoom:Number)</code>	no ne	<p>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。</p> <p>(自 1.2 废弃)</p>
<code>setZoom(zoom:Number)</code>	no ne	<p>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。</p> <p>(自 1.2 新增)</p>
<code>zoomIn()</code>	no ne	放大一级视图。

zoomOut()

no

缩小一级视图。

ne

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。
clearOverlays()	none	清除地图上所有覆盖物。
openInfoWindow(infoWindow:InfoWindow, point:Point)	none	在地图上打开信息窗口。
closeInfoWindow()	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。
pointToOverlayPixel(point:Point)	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)
overlayPixelToPoint(pixel:Pixel)	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)
getInfoWindow()	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 <code>null</code> 。 (自 1.1 新增)

<code>getOverlays()</code>	<code>Array<Overlay></code>	返回地图上的所有覆盖物。 (自 1.1 新增)
<code>getPanels()</code>	<code>MapPanels</code>	返回地图覆盖物容器集合。 (自 1.1 新增)

地图图层方法

方法	返回值	描述
<code>addTileLayer(tileLayer:TileLayer)</code>	<code>none</code>	添加一个自定义地图图层。
<code>removeTileLayer(tilelayer:TileLayer)</code>	<code>none</code>	移除一个自定义地图图层。

坐标变换

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

事件

事件	参数	描述
<code>click</code>	<code>event{type, target, point, pixel, overlay}</code>	左键单击地图时触发此事件。 当双击时，产生的事件序列为： <code>click click dblclick</code> (自 1.1 更新)
<code>dblclick</code>	<code>event{type, target, pixel, point}</code>	鼠标双击地图时会触发此事件。
<code>rightclick</code>	<code>event{type, target, point, pixel, overlay}</code>	右键单击地图时触发此事件。 当双击时，产生的事件序列为： <code>rightclick rightclick rightdblclick</code> (自 1.1 更新)
<code>rightdblclick</code>	<code>event{type, target, point, pixel, overlay}</code>	右键双击地图时触发此事件。 (自 1.1 新增)
<code>maptypechange</code>	<code>event{type, target}</code>	地图类型发生变化时触发此事件。 (自 1.1 新增)
<code>mousemove</code>	<code>event{type, target, point, pixel, overlay}</code>	鼠标在地图区域移动过程中触发此事件。 (自 1.1 更新)
<code>movestart</code>	<code>event{type, target}</code>	地图移动开始时触发此事件。
<code>moving</code>	<code>event{type, target}</code>	地图移动过程中触发此事件。
<code>moveend</code>	<code>event{type, target}</code>	地图移动结束时触发此事件。

<code>zoomstart</code>	<code>event{type, target}</code>	地图更改缩放级别开始时触发此事件。
<code>zoomend</code>	<code>event{type, target}</code>	地图更改缩放级别结束时触发此事件。
<code>addoverlay</code>	<code>event{type, target}</code>	当使用 <code>Map.addOverlay()</code> 方法向地图中添加单个覆盖物时会触发此事件。
<code>addcontrol</code>	<code>event{type, target}</code>	当使用 <code>Map.addControl()</code> 方法向地图中添加单个控件时会触发此事件。
<code>removecontrol</code>	<code>event{type, target}</code>	当使用 <code>Map.removeControl()</code> 方法移除单个控件时会触发此事件。
<code>removeoverlay</code>	<code>event{type, target}</code>	当使用 <code>Map.removeOverlay()</code> 方法移除单个覆盖物时会触发此事件。
<code>clearoverlays</code>	<code>event{type, target}</code>	当使用 <code>Map.clearOverlays()</code> 方法一次性移除全部覆盖物时会触发此事件。
<code>dragstart</code>	<code>event{type, target, pixel, point}</code>	开始拖拽地图时触发。
<code>dragging</code>	<code>event{type, target, pixel, point}</code>	拖拽地图过程中触发。
<code>dragend</code>	<code>event{type, target, pixel, point}</code>	停止拖拽地图时触发。
<code>addtilelayer</code>	<code>event{type, target}</code>	添加一个自定义地图图层时触发此事件。
<code>removetilelayer</code>	<code>event{type, target}</code>	移除一个自定义地图图层时触发此事件。
<code>load</code>	<code>event{type, target, pixel, point, zoom}</code>	调用 <code>Map.centerAndZoom()</code> 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。
<code>resize</code>	<code>event{type, target, size}</code>	地图可视区域大小发生变化时会触发此事件。

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>	Number	地图允许展示的最小级别。 (自 1.2 废弃)
<code>minZoom</code>	Number	地图允许展示的最小级别。 (自 1.2 新增)
<code>zoomLevelMax</code>	Number	地图允许展示的最大级别。 (自 1.2 废弃)
<code>maxZoom</code>	Number	地图允许展示的最大级别。 (自 1.2 新增)
<code>mapType</code>	MapTypes	地图类型，默认为 <code>BMAP_NORMAL_MAP</code> (自 1.1 新增)
<code>enableHighResolution</code>	Boolean	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)

Viewport

此类代表视野，不可实例化。

属性

属性	类型	描述
<code>center</code>	<code>Point</code>	视野中心点。 (自 1.1 新增)
<code>zoom</code>	<code>Number</code>	视野级别。 (自 1.1 新增)

ViewportOptions

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
<code>enableAnimation</code>	<code>Boolean</code>	是否启用动画效果移动地图，默认为 <code>true</code> 。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)
<code>margins</code>	<code>Array<Number></code>	视野调整的预留边距，例如： <code>margins: [30, 20, 0, 20]</code>

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 <code>map.setViewpoint</code> 计算出地图的级别为10,如果 <code>zoomFactor</code> 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)

MapTypes

此常量表示地图展示的类型。

常量

常量	描述
BMAP_NORMAL_MAP	此地图类型展示普通街道视图。 (自 1.1 新增)
BMAP_PERSPECTIVE_MAP	此地图类型展示透视图像视图。 (自 1.1 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。

属性

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

APIVersion

此常量表示 API 版本号，通过字符串进行描述。

常量

常量	描述
<code>BMAP_API_VERSION</code>	API 版本。例如在 API 1.2版本中，此常量返回字符串"1.2"。 (自 1.2 新增)

PanOptions

此常量表示 API 版本号，通过字符串进行描述。

常量

常量	描述
<code>BMAP_API_VERSION</code>	API 版本。例如在 API 1.2版本中，此常量返回字符串"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
<code>Point(lng:Number, lat:Number)</code>	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
<code>lng</code>	<code>Number</code>	地理经度。
<code>lat</code>	<code>Number</code>	地理纬度。

方法

方法	返回值	描述
<code>equals(other:Point)</code>	<code>Boolean</code>	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。
<code>toString()</code>	<code>String</code>	返回类型描述字符串。

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
<code>x</code>	<code>Number</code>	x 坐标。
<code>y</code>	<code>Number</code>	y 坐标。

方法

方法	返回值	描述
<code>equals(other:Pixel)</code>	<code>Boolean</code>	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。
<code>toString()</code>	<code>String</code>	返回类型描述字符串。

Bounds

此类表示地理坐标的矩形区域。矩形区域通过两个点的 x、y 坐标进行描述，其中数值较小的点位于矩形的左下角、数值较大的点位于矩形的右上角。

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 <code>sw</code> 表示矩形区域的西南角，参数 <code>ne</code> 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 x 坐标。

(自 1.2 废弃)

minY	Number	矩形左下角的 y 坐标。 (自 1.2 废弃)
maxX	Number	矩形右上角的 x 坐标。 (自 1.2 废弃)
maxY	Number	矩形右上角的 y 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 <code>true</code> 。
<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 <code>true</code> 。
<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 <code>true</code> 。
<code>intersects(other:Bounds)</code>	<code>Boolean</code>	计算与另一矩形的交集区域。
<code>extend(point:Point)</code>	<code>none</code>	放大此矩形，使其包含给定的点。
<code>getCenter()</code>	<code>Point</code>	返回矩形的中心点。
<code>isEmpty()</code>	<code>Boolean</code>	如果矩形为空，则返回 <code>true</code> 。
<code>getSouthWest()</code>	<code>Point</code>	返回矩形区域的西南角。 (自 1.2 新增)
<code>getNorthEast()</code>	<code>Point</code>	返回矩形区域的东北角。 (自 1.2 新增)
<code>toSpan()</code>	<code>Point</code>	返回矩形区域的跨度。 (自 1.2 新增)
<code>toString()</code>	<code>String</code>	返回类型描述字符串。

Size

此类以像素表示一个矩形区域的大小。`Size` 对象包含两个属性：`width`、`height`，分别表示矩形的水平方向的数值和竖直方向的数值。

属性

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

方法

方法	返回值	描述
<code>equals(other:Size)</code>	<code>Boolean</code>	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 <code>true</code> 。
<code>toString()</code>	<code>String</code>	返回类型描述字符串。

控件类

ControlNavigation

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>	<code>enum 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>	<code>enum ControlAnchor</code>	获取控件停靠的位置
<code>setOffset(offset:Size)</code>	<code>none</code>	设置控件停靠的偏移量
<code>getOffset()</code>	<code>Size</code>	获取控件停靠的偏移量
<code>show()</code>	<code>none</code>	显示控件
<code>hide()</code>	<code>none</code>	隐藏控件
<code>isVisible()</code>	<code>Boolean</code>	判断控件的可见性
<code>toString()</code>	<code>String</code>	返回描述类型字符串。如果需要，自定义控件需要自行实现此方法以获得描述自身类型的字符串。

ControlOptions

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>	<code>enum 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>	<code>enum NavigationControlType</code>	平移缩放控件的类型。

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。
---------------------	-------------------	---------

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
<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。

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。

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>	<code>none</code>	切换缩略地图控件的开合状态。

<code>setSize(size:Size)</code>	<code>none</code>	设置缩略地图的大小。
<code>getSize()</code>	<code>Size</code>	获取缩略地图的大小。

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。

LengthUnit

此常量表示长度单位制。

常量

常量	描述
<code>BMAP_UNIT_METRIC</code>	公制单位。
<code>BMAP_UNIT_IMPERIAL</code>	英制单位。

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
<code>MapTypeControl()</code>	创建地图类型控件。 (自 1.1 新增)

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>	<code>enum NavigationControlType</code>	获取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。

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

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
<code>CopyrightControl([opts:CopyrightControlOptions])</code>	创建一个版权控件实例。

方法

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

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。

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>	获取比例尺单位制。
<code>setUnit(unit:enum LengthUnit)</code>	<code>none</code>	设置比例尺单位制。

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
<code>id</code>	<code>Number</code>	该版权信息的唯一标识符。
<code>content</code>	<code>String</code>	该版权的文本信息，用于显示在地图上，支持 HTML 内容。

<code>bounds</code>	<code>Bounds</code>	该版权信息所适用的地理区域。
---------------------	---------------------	----------------

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>	<code>HTMLElement</code>	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>	<code>Boolean</code>	判断覆盖物是否可见。
<code>dispose()</code>	<code>none</code>	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。
<code>draw()</code>	<code>none</code>	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>	<code>none</code>	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)
<code>hide()</code>	<code>none</code>	隐藏覆盖物。 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 <code>none</code> 。 (自 1.1 更新)
<code>toString()</code>	<code>String</code>	返回类型描述字符串。

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(infoWindow: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)
setPosition(position:Point)	none	设置标注的地理坐标。 (自 1.2 新增)
getPoint()	Point	返回标注的地理坐标。 (自 1.2 废弃)
getPosition()	Point	返回标注的地理坐标。 (自 1.2 新增)
setOffset(offset:Size)	none	设置标注的偏移值。
getOffset()	Size	返回标注的偏移值。
getLabel()	Label	返回标注的文本标注。
setLabel(label:Label)	none	为标注添加文本标注。
setTitle(title:String)	none	设置标注的标题，当鼠标移至标注上时显示此标题。
getTitle()	String	返回标注的标题。
setTop(isTop:Boolean)	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>	<code>none</code>	开启标注拖拽功能。 (自 1.1 新增)
<code>disableDragging()</code>	<code>none</code>	关闭标注拖拽功能。 (自 1.1 新增)
<code>enableMassClear()</code>	<code>none</code>	允许覆盖物在 <code>map.clearOverlays</code> 方法中被清除 (自 1.1 新增)
<code>disableMassClear()</code>	<code>none</code>	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>	<code>none</code>	设置覆盖物的 <code>zIndex</code> (自 1.1 新增)

事件

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

<code>dragend</code>	<code>event{type, target, pixel, point}</code>	拖拽结束时触发此事件。 (自 1.1 新增)
<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

类表示 `Lable` 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>	<code>Point</code>	文本标注的坐标点。 (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)

Polygon

此类表示一个多边形覆盖物。

构造函数

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

方法

方法	返回值	描述
<code>setPoints(points:Array<Point>)</code>	<code>none</code>	设置多边型的点数组。
<code>getPoints()</code>	<code>Array<Point></code>	返回多边型的点数组。
<code>setStrokeColor(color:String)</code>	<code>none</code>	设置多边型的边线颜色，参数为合法的 CSS 颜色值。

<code>getStrokeColor()</code>	<code>String</code>	获得多边形的边线颜色。
<code>setFillColor(color:String)</code>	<code>none</code>	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。
<code>getFillColor()</code>	<code>String</code>	获取多边形的填充颜色。
<code>setStrokeOpacity(opacity:Number)</code>	<code>none</code>	设置多边形的边线透明度，取值范围0 - 1。
<code>getStrokeOpacity()</code>	<code>Number</code>	获取多边形的边线透明度。
<code>setFillOpacity(opacity:Number)</code>	<code>none</code>	设置多边形的填充透明度，取值范围0 - 1。
<code>getFillOpacity()</code>	<code>Number</code>	获取多边形的填充透明度。
<code>setStrokeWeight(weight:Number)</code>	<code>none</code>	设置边线的宽度，取值为大于等于1的整数。
<code>getStrokeWeight()</code>	<code>Number</code>	获取边线的宽度。
<code>setStrokeStyle(style:String)</code>	<code>none</code>	置是为实线或虚线，取值 <code>solid</code> 或 <code>dashed</code> 。
<code>getStrokeStyle()</code>	<code>String</code>	获取当前线样式状态，实线、虚线。
<code>setPointAt(index:Number, point: Point)</code>	<code>none</code>	修改指定位置的坐标。 (自 1.1 新增)
<code>getBounds()</code>	<code>Bounds</code>	获取覆盖物的地理区域范围。 (自 1.1 新增)
<code>enableEditing()</code>	<code>none</code>	开启编辑功能 (自 1.1 新增)
<code>disableEditing()</code>	<code>none</code>	关闭编辑功能 (自 1.1 新增)
<code>enableMassClear()</code>	<code>none</code>	允许覆盖物在 <code>map.clearOverlays</code> 方法中被清除 (自 1.1 新增)
<code>disableMassClear()</code>	<code>none</code>	禁止覆盖物在 <code>map.clearOverlays</code> 方法中被清除 (自 1.1 新增)

事件

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

	<code>pixel</code>	
<code>mousedown</code>	<code>event{type, target, point, pixel}</code>	鼠标在多边形上按下触发此事件。
<code>mouseup</code>	<code>event{type, target, point, pixel}</code>	鼠标在多边形释放触发此事件。
<code>mouseout</code>	<code>event{type, target, point, pixel}</code>	鼠标离开多边形时触发此事件。
<code>mouseover</code>	<code>event{type, target, point, pixel}</code>	当鼠标进入多边形区域时会触发此事件。
<code>remove</code>	<code>event{type, target}</code>	移除多边形时触发。
<code>lineupdate</code>	<code>event{type, target}</code>	覆盖物的属性发生变化时触发。 (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
<code>offset</code>	<code>Size</code>	图标定位点相对于其左上角的偏移值。 (自 1.2 废弃)
<code>anchor</code>	<code>Size</code>	图标定位点相对于其左上角的偏移值，默认等于图标宽度和高度的中间值。 (自 1.2 新增)
<code>imageOffset</code>	<code>Size</code>	图片相对于可视区域的偏移值。
<code>infoWindowOffset</code>	<code>Size</code>	当在标注上开启信息窗时，信息窗底部尖角相对于图标左上角的偏移位置。 (自 1.2 废弃)
<code>infoWindowAnchor</code>	<code>Size</code>	当在标注上开启信息窗时，信息窗底部尖角相对于图标左上角的偏移位置，默认等于图标的 <code>anchor</code> 。 (自 1.2 新增)
<code>printImageUrl</code>	<code>String</code>	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
<code>InfoWindow(content:String[, opts:InfoWindowOptions])</code>	创建一个信息窗实例,其中 content 支持 HTML 内容。

方法

方法	返回值	描述
<code>setWidth(width:Number)</code>	none	设置信息窗口的宽度,单位像素。取值范围: 220 - 730。
<code>setHeight(height:Number)</code>	none	设置信息窗口的高度,单位像素。取值范围: 60 - 650。
<code>redraw()</code>	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)</code>	none	设置信息窗口标题。支持 HTML 内容。
<code>setContent(content:String)</code>	none	设置信息窗口内容。支持 HTML 内容。
<code>enableMaximize()</code>	none	启用窗口最大化功能。
<code>disableMaximize()</code>	none	禁用窗口最大化功能。
<code>isOpen()</code>	Boolean	返回信息窗口的打开状态。
<code>setMaxContent(content:String)</code>	none	信息窗口最大化时所显示内容, 支持 HTML 内容。
<code>maximize()</code>	none	最大化信息窗口 (自 1.1 新增)
<code>restore()</code>	none	还原信息窗口 (自 1.1 新增)
<code>enableAutoPan()</code>	none	开启打开信息窗口时地图自动平移 (自 1.1 新增)
<code>disableAutoPan()</code>	none	关闭打开信息窗口时地图自动平移 (自 1.1 新增)
<code>enableCloseOnClick()</code>	none	开启点击地图时关闭信息窗口 (自 1.1 新增)
<code>disableCloseOnClick()</code>	none	关闭点击地图时关闭信息窗口 (自 1.1 新增)

事件

事件	参数	描述
close	event{type, target, point}	信息窗口被关闭时触发此事件。
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)

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，默认不启用。 (自 1.1 新增)

Polyline

此类表示地图上的折线覆盖物。

构造函数

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

方法

方法	返回值	描述
----	-----	----

<code>setPoints(points:Array<Point>)</code>	<code>none</code>	设置折线的点数组。 (自 1.2 废弃)
<code>setPath(path:Array<Point>)</code>	<code>none</code>	设置折线的点数组。 (自 1.2 新增)
<code>getPoints()</code>	<code>Array<Point></code>	返回折线的点数组。 (自 1.2 废弃)
<code>getPath()</code>	<code>Array<Point></code>	返回折线的点数组。 (自 1.2 新增)
<code>setStrokeColor(color:String)</code>	<code>none</code>	设置折线的颜色。
<code>getStrokeColor()</code>	<code>String</code>	返回折线的颜色。
<code>setStrokeOpacity(opacity:Number)</code>	<code>none</code>	设置透明度，取值范围0 - 1。
<code>getStrokeOpacity()</code>	<code>Number</code>	返回透明度。
<code>setStrokeWeight(weight:Number)</code>	<code>none</code>	设置线的宽度，范围为大于等于1的整数。
<code>getStrokeWeight()</code>	<code>Number</code>	返回线的宽度。
<code>setStrokeStyle(style:String)</code>	<code>none</code>	设置是为实线或虚线， <code>solid</code> 或 <code>dashed</code> 。
<code>getStrokeStyle()</code>	<code>String</code>	返回当前线样式状态，实线或者虚线。
<code>setPointAt(index:Number, point:Point)</code>	<code>none</code>	修改指定位置的坐标。 <code>Number</code> 从0开始计数。例如 <code>setPointAt(2, point2a)</code> 代表将折线的第3个点，坐标设为 <code>point2a</code> (自 1.1 新增)
<code>getBounds()</code>	<code>Bounds</code>	返回覆盖物的地理区域范围。 (自 1.1 新增)
<code>enableEditing()</code>	<code>none</code>	开启编辑功能 (自 1.1 新增)
<code>disableEditing()</code>	<code>none</code>	关闭编辑功能 (自 1.1 新增)
<code>enableMassClear()</code>	<code>none</code>	允许覆盖物在 <code>map.clearOverlays</code> 方法中被清除 (自 1.1 新增)
<code>disableMassClear()</code>	<code>none</code>	禁止覆盖物在 <code>map.clearOverlays</code> 方法中被清除 (自 1.1 新增)

事件

事件	参数			描述
click	event{type,	target,	point,	点击折线后会触发此事件。
	pixel}			
dblclick	event{type,	target,	point,	双击折线后会触发此事件。
	pixel}			
mousedown	event{type,	target,	point,	鼠标在折线上按下触发此事件。
	pixel}			
mouseup	event{type,	target,	point,	鼠标在折线释放触发此事件。
	pixel}			
mouseout	event{type,	target,	point,	鼠标离开折线时触发此事件。
	pixel}			
mouseover	event{type,	target,	point,	当鼠标进入折线区域时会触发此事件。
	pixel}			
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)

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	是否启用线编辑 (自 1.1 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
<code>Circle(center:Point, radius: Number)</code>	创建圆覆盖物。 (自 1.1 新增)

方法

方法	返回值	描述
<code>setCenter(center:Point)</code>	<code>none</code>	设置圆的中心点坐标。 (自 1.1 新增)
<code>getCenter()</code>	<code>Point</code>	返回圆的中心点坐标。 (自 1.1 新增)
<code>setRadius(radius: Number)</code>	<code>none</code>	设置圆的半径，单位为米。 (自 1.1 新增)
<code>getRadius()</code>	<code>Number</code>	获取圆的半径，单位为米。 (自 1.1 新增)
<code>getBounds()</code>	<code>Bounds</code>	获取覆盖物的地理区域范围。 (自 1.1 新增)
<code>enableEditing()</code>	<code>none</code>	开启编辑功能 (自 1.1 新增)
<code>disableEditing()</code>	<code>none</code>	关闭编辑功能 (自 1.1 新增)
<code>enableMassClear()</code>	<code>none</code>	允许覆盖物在 <code>map.clearOverlays</code> 方法中被清除 (自 1.1 新增)
<code>disableMassClear()</code>	<code>none</code>	禁止覆盖物在 <code>map.clearOverlays</code> 方法中被清除 (自 1.1 新增)

事件

事件	参数	描述
<code>click</code>	<code>event{type, target, point, pixel}</code>	鼠标点击圆形后会触发此事件。
<code>dblclick</code>	<code>event{type, target, point, pixel}</code>	鼠标双击圆形后会触发此事件。

<code>mousedown</code>	<code>event{type, target, point, pixel}</code>	鼠标在圆形上按下触发此事件。
<code>mouseup</code>	<code>event{type, target, point, pixel}</code>	鼠标在圆形释放触发此事件。
<code>mouseout</code>	<code>event{type, target, point, pixel}</code>	鼠标离开圆形时触发此事件。
<code>mouseover</code>	<code>event{type, target, point, pixel}</code>	当鼠标进入圆形区域时会触发此事件。
<code>remove</code>	<code>event{type, target}</code>	移除圆形时触发此事件。
<code>lineupdate</code>	<code>event{type, target}</code>	圆形覆盖物的属性发生变化时触发此事件。 (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
<code>width</code>	<code>Number</code>	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。
<code>height</code>	<code>Number</code>	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。
<code>maxWidth</code>	<code>Number</code>	信息窗最大化时的宽度，单位像素。取值范围：220 - 730。
<code>offset</code>	<code>Size</code>	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 <code>infoWindowOffset</code> 属性值，您可以为信息窗添加偏移量来改变默认位置。
<code>title</code>	<code>String</code>	信息窗标题文字，支持 HTML 内容。
<code>enableAutoPan</code>	<code>Boolean</code>	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)
<code>enableCloseOnMapClick</code>	<code>Boolean</code>	是否开启点击地图关闭信息窗口（默认开启）

nClick	an	(自 1.1 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>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>getPoint()</code>	Point	返回文本标注坐标。 (自 1.2 废弃)
<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>	<code>Size</code>	返回文本标注的偏移值。
<code>setTitle(title:String)</code>	<code>none</code>	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>	<code>String</code>	返回文本标注的标题。
<code>enableMassClear()</code>	<code>none</code>	允许覆盖物在 <code>map.clearOverlays</code> 方法中被清除 (自 1.1 新增)
<code>disableMassClear()</code>	<code>none</code>	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>	<code>none</code>	设置覆盖物的 <code>zIndex</code> 。 (自 1.1 新增)

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)

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
<code>strokeColor</code>	<code>String</code>	折线颜色
<code>strokeWeight</code>	<code>Number</code>	折线的宽度，以像素为单位。
<code>strokeOpacity</code>	<code>Number</code>	折线的透明度，取值范围0 - 1。

<code>strokeStyle</code>	<code>String</code>	折线的样式， <code>solid</code> 或 <code>dashed</code> 。
<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)
<code>enableEditing</code>	<code>Boolean</code>	是否启用线编辑 (自 1.1 新增)

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>	图标定位在地图上的位置相对于图标左上角的偏移值，默认偏移值为图标的中心位置，即图标的中心位置表示其真正的坐标点。 (自 1.2 废弃)
<code>anchor</code>	<code>Size</code>	图标定位在地图上的位置相对于图标左上角的偏移值，默认偏移值为图标的中心位置，即图标的中心位置表示其真正的坐标点。 (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>	当在标注上开启信息窗时，通过此属性可控制信息窗底部尖角相对于图标左上角的偏移位置，默认此偏移值等于标注的 <code>offset</code> 。 (自 1.2 废弃)
<code>infoWindowAnchor</code>	<code>Size</code>	当在标注上开启信息窗时，通过此属性可控制信息窗底部尖角相对于图标左上角的偏移位置，默认此偏移值等于标注的 <code>offset</code> 。 (自 1.2 新增)
<code>printImageUrl</code>	<code>String</code>	设置 <code>icon</code> 打印图片的 url，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 <code>icon</code> 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(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 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	当在标注上开启信息窗时，通过此方法可改变信息窗底部尖角相对于图标左上角的偏移位置。 (自 1.2 废弃)
<code>setInfoWindowAnchor(offset:Size)</code>	none	当在标注上开启信息窗时，通过此方法可改变信息窗底部尖角相对于图标左上角的偏移位置。 (自 1.2 新增)
<code>setPrintImageUrl(url:String)</code>	none	设置 <code>icon</code> 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 <code>icon</code> 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)
<code>toString()</code>	String	返回类型描述字符串。

CircleOptions

Circle 类构造函数的可选参数。

属性

属性	类型	描述
<code>enableMassClear</code>	Boolean	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)

<code>enableEditing</code>	<code>Boolean</code>	是否启用线编辑 (自 1.1 新增)
----------------------------	----------------------	-----------------------

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[, opts:MenuItemOptions])</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> 。

方法

方法	返回值	描述
<code>setText(text:String)</code>	<code>none</code>	设置菜单项显示的文本。
<code>enable()</code>	<code>none</code>	启用菜单项。
<code>disable()</code>	<code>none</code>	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>	<code>Number</code>	指定此菜单项的宽度，菜单以最长的菜单项宽度为准。

工具类

此类表示标注工具。此工具用来让使用者在地图上标注一个位置，可以通过该工具提供的事件来获得标注的位置。(自 1.2 废弃)

PushpinTool

此类表示标注工具。此工具用来让使用者在地图上标注一个位置，可以通过该工具提供的事件来获得标注的位置。

构造函数

构造函数	描述
<code>PushpinTool(map:Map[, opts:PushpinToolOptions])</code>	创建一个标注控件项。

方法

方法	返	描述
----	---	----

	回 值	
<code>open()</code>	<code>Boolean</code>	开启标注工具。返回值如果为 <code>false</code> ，则表明开启失败，可能有其他工具正处于开启状态。请先关闭其他工具再进行开启。
<code>close()</code>	<code>none</code>	关闭标注工具。
<code>setIcon(icon:Icon)</code>	<code>Icon</code>	设置标注的图标。
<code>getIcon()</code>	<code>Icon</code>	获取标注的图标。
<code>setCursor(cursor:String)</code>	<code>String</code>	设置标注的鼠标样式。
<code>getCursor()</code>	<code>String</code>	获取标注的鼠标样式。
<code>toString()</code>	<code>String</code>	返回类型描述字符串。

事件

事件	参数	描述
<code>markend</code>	<code>event{type, target, marker}</code>	在用户每完成一次标注时触发此事件。

PushpinToolOptions

此类表示 `PushpinTool` 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>icon</code>	<code>Icon</code>	标注所使用的图标。
<code>cursor</code>	<code>String</code>	标注所使用的鼠标样式。
<code>followText</code>	<code>String</code>	跟随鼠标移动的说明文字，默认为空。

DistanceTool

此类表示测距工具，用来让用户在地图上绘制折线并测定折线的距离。

构造函数

构造函数	描述
<code>DistanceTool(map:Map)</code>	创建一个绘制测距工具实例。

方法

方法	返回值	描述
<code>open()</code>	<code>Boolean</code>	开启测距工具。返回值如果为 <code>false</code> ，则表明开启失败，可能有其他工具正处于开启状态。请先关闭其他工具再进行开启。
<code>close()</code>	<code>none</code>	关闭测距工具。
<code>toString()</code>	<code>String</code>	返回类型描述字符串。

事件

事件	参数	描述
<code>drawend</code>	<code>event{type, target, points, polylines, distance}</code>	用户每次完成一次测距操作时触发此事件。

DragAndZoomTool

此类表示区域缩放控件。通过用户绘制的矩形区域的大小来确定地图的放大或缩小级别。

构造函数

构造函数	描述
<code>DragAndZoomTool(map:Map[, opts:DragAndZoomToolOptions])</code>	创建一个区域缩放控件实例。

方法

方法	返回值	描述
<code>open()</code>	<code>Boolean</code>	开启区域缩放工具。返回值如果为 <code>false</code> ，则表明开启失败，可能有其他工具正处于开启状态。请先关闭其他工具再进行开启。
<code>close()</code>	<code>none</code>	关闭区域缩放工具。
<code>toString()</code>	<code>String</code>	返回类型描述字符串。

事件

事件	参数	描述
----	----	----

<code>drawend</code>	<code>event{type, target, bounds}</code>	用户每完成一次缩放动作时触发此事件。
----------------------	--	--------------------

DragAndZoomTool

此类表示 `DragAndZoomTool` 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>zoomType</code>	<code>enum ZoomType</code>	表示用户绘制趋于结束后进行放大还是缩小操作，默认为放大。
<code>autoClose</code>	<code>Boolean</code>	每次完成缩放操作后是否自动退出，默认不退出。
<code>followText</code>	<code>String</code>	跟随鼠标移动的说明文字。

OptionsZoomType

此枚举常量表示区域缩放工具的缩放类型

常量

常量	描述
<code>BMAP_ZOOM_IN</code>	区域放大。
<code>BMAP_ZOOM_OUT</code>	区域缩小。

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>getTileUrl(point:Point, zoom:Number)</code>	<code>String</code>	抽象。向地图返回地图图块的网址，图块索引由 <code>point</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>	<code>String enum MapType</code>	返回此图块层的地图类型。
<code>getCopyright()</code>	<code>Number</code>	返回地图图层数据的版权对象。
<code>isTransparentPng()</code>	<code>Number</code>	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>	<code>Boolean</code>	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。
<code>tileUrlTemplate</code>	<code>String</code>	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}&y={Y}&z={Z}.png</code> 其中 <code>x</code> 和 <code>y</code> 分别指纬度和经度图块坐标， <code>z</code> 指缩放级别，比如： <code>http://yourhost/tile?x=3&y=27&z=5.png</code> 。如果您没有提供图块网址模板，您需要实现 <code>TileLayer.getTilesUrl()</code> 抽象方法。
<code>copyright</code>	<code>Copyright</code>	地图图层的版权信息。
<code>zIndex</code>	<code>Number</code>	图层的 <code>zIndex</code> 。 (自 1.2 新增)

MapType

此枚举常量表示地图图层的类型

构造函数

构造函数	描述
<code>MapType(name:String, layer:TileLayer[],</code>	创建 <code>MapType</code> 对象实例。

options:MapTypeOptions])

(自 1.2 新增)

方法

方法	返回值	描述
<code>getName()</code>	<code>String</code>	返回地图类型名称。 (自 1.2 新增)
<code>getTileLayer()</code>	<code>TileLayer</code>	返回地图类型对应的图层。 (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。

属性

属性	类型	描述
<code>minZoom</code>	<code>Number</code>	该类型地图的最小级别。 (自 1.2 新增)
<code>maxZoom</code>	<code>Number</code>	该类型地图的最大级别。 (自 1.2 新增)
<code>errorImageUrl</code>	<code>String</code>	当没有图块时所显示的错误图片地址。默认为透明的 gif 图片。 (自 1.2 新增)

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> <code>([opts:TrafficLayerOptions])</code>	创建交通流量图层 参数： <code>opts: TrafficLayerOptions</code> ，可选 <code>options</code> 参数指定应作为对象常量传递。如果可选参数提供 <code>predictDate</code> ，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
<code>predictDate</code>	<code>Object</code>	预测日期 <code>{</code> <code> hour: 小时（0-24），</code> <code> weekday: 星期（1-7）</code> <code>}</code> (自 1.1 新增)

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 <code>location</code> 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)</code>	<code>none</code>	根据检索词发起检索。
<code>searchInBounds(keyword:String, bounds:Bounds)</code>	<code>none</code>	根据范围和检索词发起范围检索。
<code>searchNearby(keyword:String, center:LocalResultPoi String Point[, radius:Number])</code>	<code>none</code>	根据中心点、半径与检索词发起周边检索。当 <code>center</code> 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>getResults()</code>	Loca lRes ult	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>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(capacity:Number)</code>	none	设置每页容量，范围检索支持1 - 50，其余支持1 - 10。此值只对下一次新检索有效。
<code>getPageCapacity()</code>	Numb er	获取每页容量
<code>setSearchCompleteCallback(callback:Function)</code>	none	设置检索结束后的回调函数
<code>setMarkersSetCallback(callback:Function)</code>	none	设置添加标注后的回调函数
<code>setInfoHtmlSetCallback(callback:Function)</code>	none	设置标注气泡创建时的回调函数
<code>setResultsHtmlSetCallback(callback:Function)</code>	none	设置结果列表创建后的回调函数
<code>getStatus()</code>	Stat usCo des	返回状态码
<code>toString()</code>	Stri ng	返回类型说明

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。 location 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。 start : 起点, 参数可以是关键字、坐标点(1.1版本支持) 或者 <code>LocalSearchPoi</code> 实例。 end : 终点, 参数可以是关键字、坐标点(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, 回调函数参数为 <code>TransitRouteResult</code> 对象
<code>setMarkersSetCallback(callback:Function)</code>	none	设置添加标注后的回调函数, 参数分别为起始点数组和换乘点数组
<code>setInfoHtmlSetCallback(callback:Function)</code>	none	设置气泡打开后的回调函数, 参数为起始点或换乘点和气泡的 DOM 元素
<code>setPolylinesSetCallback(callback:Function)</code>	none	设置添加路线后的回调函数, 参数分别为公交线路 <code>Line</code> 数组和步行线路 <code>Route</code> 数组

<code>setResultsHtmlSetCallback(callback:Function)</code>	<code>none</code>	设置结果列表创建后的回调函数
<code>getStatus()</code>	<code>StatusCodes</code>	返回状态码
<code>toString()</code>	<code>String</code>	返回类型说明

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)</code>	<code>none</code>	发起检索。 <code>start</code> : 起点，参数可以是关键字、坐标点（1.1版本支持）和 <code>LocalSearchPoi</code> 实例。 <code>end</code> : 终点，参数可以是关键字、坐标点（1.1版本支持）或 <code>LocalSearchPoi</code> 实例。
<code>getResults()</code>	<code>DrivingRouteResult</code>	返回最近一次检索的结果
<code>clearResults()</code>	<code>none</code>	清除最近一次检索的结果
<code>enableAutoViewport()</code>	<code>none</code>	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>	<code>none</code>	禁用自动调整地图层级。
<code>setLocation(location:Map Point String)</code>	<code>none</code>	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>	<code>none</code>	设置路线规划策略，参数为策略常量
<code>setSearchCompleteCallback(callback)</code>	<code>none</code>	设置检索结束后的回调函数，回调函数参数

<code>ck:Function)</code>		为 <code>DrivingRouteResult</code> 对象
<code>setMarkersSetCallback(callback:Function)</code>	<code>none</code>	设置添加标注后的回调函数，参数为起始点数组
<code>setInfoHtmlSetCallback(callback:Function)</code>	<code>none</code>	设置气泡打开后的回调函数，参数为起始点或关键点和气泡的 DOM 元素
<code>setPolylinesSetCallback(callback:Function)</code>	<code>none</code>	设置添加路线后的回调函数，参数为线路数组
<code>setResultsHtmlSetCallback(callback:Function)</code>	<code>none</code>	设置结果列表创建后的回调函数
<code>getStatus()</code>	<code>StatusCodes</code>	返回状态码
<code>toString()</code>	<code>String</code>	返回类型说明

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>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 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）。

LocalCity

此类用于获取用户所在的城市位置信息。

构造函数

构造函数	描述
<code>LocalCity([opts:LocalCityOptions])</code>	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。
<code>toString()</code>	<code>String</code>	返回类型说明。

LocalSearchOptions

此类表示 `LocalSearch` 构造函数的可选参数。

属性

属性	类型	描述
<code>renderOptions</code>	<code>RenderOptions</code>	结果呈现设置。
<code>pageCapacity</code>	<code>Number</code>	每页结果的数量（范围检索支持1到50，其余为1到10）
<code>onSearchComplete</code>	<code>Function</code>	检索完成后的回调函数。 参数： <code>results: LocalResult</code>
<code>onMarkersSet</code>	<code>Function</code>	标注添加完成后的回调函数。 参数： <code>pois: Array<LocalResultPoi></code> ，通过 <code>marker</code> 属性可得到其对应的标注。
<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。 <code>html: HTMLElement</code> ，气泡内的 Dom 元素
<code>onResultsHtmlSet</code>	<code>Function</code>	结果列表添加完成后的回调函数。 参数： <code>container: HTMLElement</code> ，结果列表所用的 HTML 元素

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
<code>renderOptions</code>	<code>RenderOptions</code>	搜索结果的呈现设置。
<code>policy</code>	<code>TransitPolicy</code>	公交导航的策略参数。
<code>pageCapacity</code>	<code>Number</code>	一页返回方案的个数，上限为5条。
<code>onSearchComplete</code>	<code>Function</code>	检索完成后的回调函数。 参数： <code>results: TransitRouteResult</code> ，公交导航结果
<code>onMarkersSet</code>	<code>Function</code>	标注添加完成后的回调函数。 参数： <code>pois: Array<LocalResultPoi></code> ，起点和目的地数组。 <code>transfers: Array<LocalResultPoi></code> ，公交车站数组。
<code>onPolylinesSet</code>	<code>Function</code>	折线添加完成后的回调函数。 参数： <code>lines: Array<Line></code> ，公交线路数组。 <code>routes: Array<Route></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>	气泡内容创建后的回调函数。 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>onResultsHtmlSet</code>	<code>Function</code>	结果列表添加完成后的回调函数。 参数： <code>container</code> : 结果列表所用的 HTML 元素

DrivingRouteOptions

此类表示 `DrivingRoute` 构造函数的可选参数。

属性

属性	类型	描述
<code>renderOptions</code>	<code>RenderOptions</code>	结果呈现设置。
<code>drivingPol</code>	<code>DrivingPo</code>	策略

icy	licy	
onSearchComplete	Function	检索完成后的回调函数。 参数： <code>results: DrivingRouteResult</code>
onMarkersSet	Function	标注添加完成后的回调函数。 参数： <code>pois: Array<LocalResultPoi></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。
onPolylineSet	Function	折线添加完成后的回调函数。 参数： <code>routes: Array<Route></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。
onInfoHtmlSet	Function	标注气泡内容创建后的回调函数。 参数： <code>poi: LocalResultPoi</code> , 通过 <code>marker</code> 属性可得到当前的标注。 <code>html: HTMLElement</code> , 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： <code>container</code> : 结果列表所用的 HTML 元素。

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。 <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>	<code>none</code>	发起检索。 <code>start</code> : 起点, 参数可以是关键字、坐标点(1.1版本支持) 或者 <code>LocalSearchPoi</code> 实例。 <code>end</code> : 终点, 参数可以是关键字、坐标点(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>	<code>none</code>	清除最近一次检索的结果
<code>enableAutoViewport()</code>	<code>none</code>	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>	<code>none</code>	禁用自动调整地图层级。
<code>setLocation(location:Map Point String)</code>	<code>none</code>	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>	<code>none</code>	设置检索结束后的回调函数，回调函数参数为 <code>WalkingRouteResult</code> 对象
<code>setMarkersSetCallback(callback:Function)</code>	<code>none</code>	设置添加标注后的回调函数，参数为起始点数组
<code>setInfoHtmlSetCallback(callback:Function)</code>	<code>none</code>	设置气泡打开后的回调函数，参数为起始点或关键点和气泡的 DOM 元素
<code>setPolylinesSetCallback(callback:Function)</code>	<code>none</code>	设置添加路线后的回调函数，参数为线路数组
<code>setResultsHtmlSetCallback(callback:Function)</code>	<code>none</code>	设置结果列表创建后的回调函数
<code>getStatus()</code>	<code>StatusCodes</code>	返回状态码
<code>toString()</code>	<code>String</code>	返回类型说明

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>RenderOptions</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。

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
<code>map</code>	<code>Map</code>	展现结果的地图实例。当指定此参数后，搜索结果的标注、线路等均会自动添加到此地图上。
<code>panel</code>	<code>String HTML Element</code>	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 <code>LocalCity</code> 无效。
<code>selectFirstResult</code>	<code>Boolean</code>	是否选择第一个检索结果。此属性仅对 <code>LocalSearch</code> 有效。
<code>autoViewport</code>	<code>Boolean</code>	检索结束后是否自动调整地图视野。此属性对 <code>LocalCity</code> 无效。

TransitPolicy

此常量表示公交方案的策略。

常量

常量	描述
<code>BMAP_TRANSIT_POLICY_LEAST_TIME</code>	最少时间
<code>BMAP_TRANSIT_POLICY_LEAST_TRANSFER</code>	最少换乘
<code>BMAP_TRANSIT_POLICY_LEAST_WALKING</code>	最少步行

DrivingPolicy

此枚举类型表示驾车方案的策略配置。

常量

常量	描述
<code>BMAP_DRIVING_POLICY_LEAST_TIME</code>	最少时间。
<code>BMAP_DRIVING_POLICY_LEAST_DISTANCE</code>	最短距离。
<code>BMAP_DRIVING_POLICY_AVOID_HIGHWAYS</code>	避开高速。

WalkingRouteOptions

此类表示 **WalkingRoute** 构造函数的可选参数。

属性

属性	类型	描述
renderOptions	String HTMLElement	搜索结果呈现设置。
onSearchComplete	Function	检索完成后的回调函数。 参数： results: WalkingRouteResult
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。
onPolylineSet	Function	折线添加完成后的回调函数。 参数： 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。
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 : 结果列表所用的 HTML 元素

LocalCityResult

此类表示 **LocalCity** 的定位结果。

属性

属性	类型	描述
center	LocalResultPoi	城市所在中心点。
level	Number	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 map 实例，则地图级别将根据您提供的地图大小进行调整。
name	String	城市名称。

StatusCode

此常量用于描述对象当前状态。

常量

常量	描述
<code>BMAP_STATUS_SUCCESS</code>	检索成功
<code>BMAP_STATUS_UNKNOWN_LOCATION</code>	位置结果未知
<code>BMAP_STATUS_UNKNOWN_ROUTE</code>	导航结果未知
<code>BMAP_STATUS_INVALID_KEY</code>	非法密钥
<code>BMAP_STATUS_INVALID_REQUEST</code>	非法请求
<code>BMAP_STATUS_PERMISSION_DENIED</code>	没有权限。 (自 1.1 新增)
<code>BMAP_STATUS_SERVICE_UNAVAILABLE</code>	服务不可用。 (自 1.1 新增)
<code>BMAP_STATUS_TIMEOUT</code>	超时。 (自 1.1 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。
--------------------------------	-------------------------------	-----------------------

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>	<code>enum 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。

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。

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
<code>city</code>	<code>String</code>	本次检索所在的城市

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。

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。

构造函数

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

方法

方法	返回值	描述
setPanelOffset (of fset:Size)	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。
show ()	none	显示实时路况。
hide ()	none	隐藏实时路况。

LocalResult

此类表示 **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

方法

方法	返回值	描述
----	-----	----

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

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>	<code>Route</code>	返回方案包含的某条步行线路。
<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>getDescription(includeHtml:Boolean)</code>	<code>String</code>	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。

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</code> <code>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</code> <code>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)

Geocoder

此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>	<code>none</code>	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>	<code>none</code>	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)
<code>toString()</code>	<code>String</code>	返回类型说明。

Geolocation

获取用户当前的位置，此方法将直接使用浏览器提供的定位功能，并将 wgs84坐标转换为百度坐标。

构造函数

构造函数	描述
<code>Geolocation()</code>	创建 <code>Geolocation</code> 对象 (自 1.1 新增)

方法

方法	返回值	描述
<code>getCurrentPosition(callback: function[, options: PositionOptions])</code>	<code>none</code>	获取用户当前位置。当定位成功时，回调函数的参数为 <code>GeolocationResult</code> 对象，否则为 <code>null</code> 。 (自 1.1 新增)
<code>getStatus()</code>	<code>StatusCode</code>	获取状态码，当定位成功后，状态码为： <code>BMAP_STATUS_SUCCESS</code> ，如果为其他状态码表示不能获取您当前的位置。 (自 1.1 新增)

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。

属性

属性	类型	描述
<code>title</code>	<code>String</code>	结果的名称标题。
<code>point</code>	<code>Point</code>	该结果所在的地理位置。
<code>url</code>	<code>String</code>	在百度地图详情页面展示该结果点的链接。
<code>address</code>	<code>String</code>	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。
<code>city</code>	<code>String</code>	所在城市。
<code>phoneNum</code>	<code>String</code>	电话，根据数据部分提供。

ber		
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。

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
title	String	公交线路全称
uid	String	公交线路 uid
type	enum LineType	公交线路类型

方法

方法	返回值	描述
getNumViaStops()	Number	返回公交线路途径的车站个数。
getGetOnStop()	LocalResultPoi	返回上车站。
getGetOffStop()	LocalResultPoi	返回下车站。
getPoints()	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。
getPolyline()	Polyline	返回公交线路对应的折线覆盖物。
getDistance([format:Boolean])	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> 。

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<Point></code>	返回路线的地理坐标点数组。
<code>getRouteType()</code>	<code>RouteTypes</code>	返回路线类型，可区分是驾车还是步行线路。

GeocoderResult

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>Object</code>	一个 JavaScript 对象，描述地址的层次结构，结构如下： <pre>{ streetNumber: 门牌号码, street: 街道名称, district: 区县名称, city: 城市名称, province: 省份名称 }</pre> (自 1.1 新增)
<code>surroundingPois</code>	<code>Array<LocalResultPoi></code>	附近的 POI 点 (自 1.1 新增)

GeolocationResult

此类作为 `getCurrentPosition` 回调函数的参数，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>accuracy</code>	<code>Number</code>	定位精确程度，单位为米。 (自 1.1 新增)

PoiType

此枚举常量表示地点的类型。

常量

常量	描述
<code>BMAP_POI_TYPE_NORMAL</code>	一般位置点
<code>BMAP_POI_TYPE_BUSSTOP</code>	公交车站位置点
<code>BMAP_POI_TYPE_SUBSTOP</code>	地铁车站位置点

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

RouteType

此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

BMAP_ROUTE_TYPE_WALKING	步行线路
-------------------------	------

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)

PositionOptions

此类作为 **getCurrentPosition** 的可选参数，不能实例化。

属性

属性	类型	描述
enableHighAccuracy	Boolean	要求浏览器获取最佳结果。 (自 1.1 新增)
timeout	Number	超时时间。 (自 1.1 新增)
maximumAge	Number	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。 (自 1.1 新增)

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

	<code>tail</code>	白天费用表示全天费用。
<code>distance</code>	<code>Number</code>	出租车里程，单位为米。
<code>remark</code>	<code>String</code>	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
<code>initialFare</code>	<code>Number</code>	出租车起步价。
<code>unitFare</code>	<code>Number</code>	出租车单价。
<code>totalFare</code>	<code>Number</code>	出租车费用总价。