



Html5 API 接口文档 V1.0

天地图有限公司

2016 年 4 月

Html5 API 接口文档 V1.0	1
1 地图主类	6
1.1 Map 类	6
1.1.1 MapOptions 属性	6
1.1.2 构造函数	6
1.1.3 配置方法	6
1.1.4 地图状态方法	7
1.1.5 修改地图状态方法	8
1.1.6 坐标变换	9
1.1.7 覆盖物方法	9
1.1.8 控件方法	10
1.1.9 地图图层方法	10
1.1.10 事件	10
2 图层类	11
2.1 TileLayer 类	12
2.2.1 TileLayerOptions 属性	12
2.2.2 构造函数	12
2.2.3 方法	12
2.2.4 事件	13
2.2 TileLayer.TDT 类	13
2.3.1 TileLayer.TDTOptions	13
2.3.2 构造函数	13
2.3.3 方法	14
2.3.4 事件	14
2.3 TileLayer.WMS 类	14
2.4.1 TileLayer.WMSOptions 属性	14
2.4.2 构造函数	15
2.4.3 方法	15
2.4.4 事件	15
3 控件类	15
3.1 Control 类	16
3.1.1 ControlPosition 常量	16
3.1.2 ControlOptions 属性	16
3.1.3 方法	16
3.2 Control.Zoom 类	17
3.2.1 Control.ZoomOptions 属性	17
3.2.2 构造函数	17
3.3 Control.Scale 类	17
3.3.1 Control.ScaleOptions 属性	17
3.3.2 构造函数	18
3.3.3 方法	18
3.4 Control.Copyright 类	18
3.4.1 Control.CopyrightOptions 属性	18

3.4.2 构造函数.....	18
3.4.3 方法.....	18
3.5 Control.OverviewMap 类.....	19
3.4.4 Control.OverviewMapOptions 属性.....	19
3.4.5 构造函数.....	19
3.4.6 方法.....	20
3.4.7 事件.....	20
4 覆盖物类.....	20
4.1 Overlay 类.....	20
4.1.1 OverlayOptions 属性.....	21
4.1.2 方法.....	21
4.2 Label 类.....	21
4.2.1 LabelOptions 属性.....	22
4.2.2 构造函数.....	22
4.2.3 方法.....	22
4.2.4 事件.....	23
4.3 Marker 类.....	23
4.3.1 MarkerOptions 属性.....	23
4.3.2 构造函数.....	24
4.3.3 方法.....	24
4.3.4 事件.....	25
4.4 Icon 类.....	25
4.4.1 IconOptions 属性.....	25
4.4.2 构造函数.....	25
4.4.3 方法.....	26
4.5 InfoWindow 类.....	26
4.5.1 InfoWindowOptions 属性.....	26
4.5.2 构造函数.....	27
4.5.3 方法.....	27
4.5.4 事件.....	28
4.6 Polyline 类.....	28
4.6.1 PolylineOptions 属性.....	28
4.6.2 构造函数.....	28
4.6.3 方法.....	28
4.6.4 事件.....	29
4.7 Polygon 类.....	29
4.7.1 PolygonOptions 属性.....	30
4.7.2 构造函数.....	30
4.7.3 方法.....	30
4.7.4 事件.....	31
4.8 Rectangle 类.....	31
4.8.1 RectangleOptions 属性.....	32
4.8.2 构造函数.....	32
4.8.3 方法.....	32

4.8.4 事件.....	33
4.9 Circle 类.....	33
4.9.1 CircleOptions 属性.....	34
4.9.2 构造函数.....	34
4.9.3 方法.....	34
4.9.4 事件.....	35
5 工具类.....	35
5.1 MarkTool 类.....	36
5.1.1 MarkToolOptions 属性.....	36
5.1.2 构造函数.....	36
5.1.3 方法.....	36
5.1.4 事件.....	37
5.2 PolygonTool 类.....	37
5.2.1 PolygonToolOptions 属性.....	37
5.2.2 构造函数.....	37
5.2.3 方法.....	37
5.2.4 事件.....	38
5.3 PolylineTool 类.....	38
5.3.1 PolylineToolOptions 属性.....	39
5.3.2 构造函数.....	39
5.3.3 方法.....	39
5.3.4 事件.....	40
5.4 RectangleTool 类.....	40
5.4.1 RectangleToolOptions 属性.....	40
5.4.2 构造函数.....	40
5.4.3 方法.....	41
5.4.4 事件.....	41
5.5 CircleTool 类.....	42
5.4.5 CircleToolOptions 属性.....	42
5.4.6 构造函数.....	42
5.4.7 方法.....	42
5.4.8 事件.....	43
6 组件类.....	43
6.1 MarkerClusterer 类.....	43
6.1.1 MarkerClustererOptions 属性.....	43
6.1.2 构造函数.....	44
6.1.3 方法.....	44
6.2 CoordinatePickup 类.....	45
6.2.1 CoordinatePickupOptions 属性.....	45
6.2.2 构造函数.....	45
6.2.3 方法.....	45
7 右键菜单.....	46
7.1 MenuItem 类.....	46
7.1.1 构造函数.....	46

7.1.2 方法.....	46
7.2 ContextMenu 类.....	46
7.2.1 ContextMenuOptions 属性.....	46
7.2.2 构造函数.....	46
7.2.3 方法.....	47
7.2.4 事件.....	47
8 实体类.....	47
8.1 LngLat 类.....	47
8.1.1 LngLatOptions 属性.....	47
8.1.2 构造函数.....	48
8.1.3 方法.....	48
8.2 LngLatBounds 类.....	48
8.2.1 构造函数.....	48
8.2.2 方法.....	48
8.3 Point 类.....	49
8.3.1 PointOptions 属性.....	49
8.3.2 构造函数.....	49
8.3.3 方法.....	49
9 服务类.....	50
9.1 搜索.....	50
9.1.1 LocalSearch 类.....	50
9.1.2 LocalSearchResult 类.....	51
9.2 逆地理编码.....	56
9.2.1 Geocoder 类.....	56
9.2.2 GeocoderResult 类.....	56
9.2.3 AddressComponent 类.....	57
9.3 公交.....	57
9.3.1 TransitRoute.....	57
9.3.2 TransitRouteResult.....	59
9.3.3 TransitRoutePlan.....	59
9.3.4 TransitRouteLine.....	59
9.3.5 BusLineSearch.....	61
9.3.6 BusListResult.....	62
9.3.7 BusLine.....	62
9.4 驾车.....	63
9.4.1 DrivingRoute.....	63
9.4.2 DrivingRouteResult.....	65
9.4.3 RoutePlan.....	65
9.4.4 Route.....	65
9.4.5 Step.....	66

1 地图主类

1.1 Map 类

此类是地图 API 的核心类，用来实例化一个地图。

1.1.1 MapOptions 属性

属性	类型	默认值	说明
projection	String	"EPSG:900913"	指定地图的投影方式，目前支持的地图投影方式有：EPSG:900913(墨卡托投影)，EPSG:4326(大地平面投影)。
minZoom	number	null	地图允许展示的最小级别。
maxZoom	number	null	地图允许展示的最大级别。
maxBounds	LngLatBound	null	当这个选项被设置后，地图被限制在给定的地理边界内，当用户平移将地图拖动到视图以外的范围时会出现弹回的效果，并且也不允许缩小视图到给定范围以外的区域(这取决于地图的尺寸)。使用 setMaxBounds 方法可以动态地设置这种约束。
center	LngLat	null	地图的初始化中心点。
zoom	number	null	地图的初始化级别。

1.1.2 构造函数

构造函数	说明
Map(container:String HTMLElement[,opts:MapOptions])	在指定的容器内创建地图实例，之后需要调用 Map.centerAndZoom()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。 参数说明： container：用于显示地图的 DIV 对象。 opts：地图属性对象，请参考 MapOptions。

1.1.3 配置方法

方法	返回值	说明
enableDrag()	none	启用地图拖拽，默认启用。
disableDrag()	none	禁用地图拖拽。
isDrag()	boolean	是否启用地图拖拽，true 表示启用，false 表示

		禁止。
enableScrollWheelZoom()	none	启用滚轮放大缩小，默认启用。
disableScrollWheelZoom()	none	禁用滚轮放大缩小。
isScrollWheelZoom()	boolean	是否启用滚轮放大缩小，true 表示启用，false 表示禁止。
enableDoubleClickZoom()	none	启用双击放大，默认启用。
disableDoubleClickZoom()	none	禁用双击放大。
isDoubleClickZoom()	boolean	是否启用双击放大，true 表示启用，false 表示禁止。
enableKeyboard()	none	启用键盘操作，默认启用。
disableKeyboard()	none	禁用键盘操作。
isKeyboard()	boolean	是否启用键盘操作，true 表示启用，false 表示禁止。
enableInertia()	none	启用地图惯性拖拽，默认启用。
disableInertia()	none	禁用地图惯性拖拽。
isInertia()	boolean	是否启用地图惯性拖拽，true 表示启用，false 表示禁止。
enableContinuousZoom()	none	启用连续缩放效果，默认启用。
disableContinuousZoom()	none	禁用连续缩放效果。
isContinuousZoom()	boolean	是否启用连续缩放效果，true 表示启用，false 表示禁止。
enablePinchToZoom()	none	启用双指操作缩放，默认启用。
disablePinchToZoom()	none	禁用双指操作缩放。
isPinchToZoom()	boolean	是否启用操作缩放，true 表示启用，false 表示禁止。
enableAutoResize()	none	启用自动适应容器尺寸变化，默认启用。
disableAutoResize()	none	禁用自动适应容器尺寸变化。

1.1.4 地图状态方法

方法	返回值	说明
getCode()	String	返回地图投影类型信息。
getBounds()	LngLatBounds	返回地图可视区域，以地理坐标表示。
getCenter()	LngLat	返回地图当前中心点。
getSize()	Point	返回地图视图的大小，以像素表示。
getZoom()	number	返回地图当前缩放级别。
getDistance(start:LngLat,end:LngLat)	number	返回两点之间的距离，单位是米。 参数说明： start: 起点地理坐标。 end: 终点地理坐标。
getViewport(view:Array<LngLat>)	json	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 center 和

		zoom 属性，分别表示地图的中心点和级别。
--	--	------------------------

1.1.5 修改地图状态方法

方法	返回值	说明
centerAndZoom(lnglat:LngLat,zoom:Number)	none	初始化定位地图。 参数说明： lnglat: 地图的中心点地理坐标。 zoom: 地图的比例尺级别。
panTo(lnglat:LngLat[,zoom:Number])	none	将地图的中心点变换到指定的地理坐标，如果同时指定了缩放等级，则同时缩放到指定的等级。 参数说明： lnglat: 地图的中心点地理坐标。 zoom: 地图的比例尺级别。
panBy(position: Point)	none	将地图在水平位置上移动 x 像素，垂直位置上移动 y 像素。 参数说明： position: 像素坐标。
setZoom(zoom:Number)	none	将视图缩放到指定的缩放等级，中心点坐标不变。 参数说明： zoom: 地图显示级别。
zoomIn()	none	放大一级视图。
zoomOut()	none	缩小一级视图。
checkResize()	none	通知地图其容器大小已更改。在更改了容器对象的大小后调用此方法，以便地图能够调整自己适合于新的大小。
setMinZoom(level:Number)	none	设置地图最小显示级别。 参数说明： level: 地图显示最小级别。
setMaxZoom(level:Number)	none	设置地图最大显示级别。 参数说明： level: 地图显示最大级别。
setMaxBounds(bounds:LngLatBounds)	none	设置地图的显示范围。 参数说明： bounds: 地图坐标范围 LngLatBounds。
setViewport(view:Array<LngLat>)	none	根据提供的坐标点数组设置地图视野，调整后的视野会保证包含提供的坐标点。

1.1.6 坐标变换

方法	返回值	说明
containerPointToLngLat(pixel:Point)	LngLat	将地图上相对于 container 的像素坐标转化为地理坐标，参数是 Point，返回值类型是 LngLat。pixel 参数是像素对象。
lngLatToContainerPoint(lnglat:LngLat)	Point	将地理坐标转化为地图上点的像素坐标，相对于 container 左上角，参数是 LngLat，返回值是 Point。
lngLatToLayerPoint(lnglat:LngLat)	Point	将地理坐标转化为地图图层上点的像素坐标，相对于地图图层左上角，参数是 LngLat，返回值是 Point。
layerPointToLngLat(pixel:Point)	LngLat	返回给定地图上点的地理坐标系。 将地图上相对于地图图层的像素坐标转化为地理坐标，参数是 Point，返回值类型是 LngLat。pixel 参数是像素对象。

1.1.7 覆盖物方法

方法	返回值	说明
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。 参数说明： overlay: 覆盖物对象。
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。 参数说明： overlay: 覆盖物对象。
clearOverLays()	none	清除地图上所有覆盖物。
getOverlays()	Array<Overlay>	返回地图上的所有覆盖物。
getPanes()	MapPanes	返回地图覆盖物容器列表。
openInfoWindow(indow:InfoWindow, lnglat:Lnglat)	none	在地图上打开信息窗口。 参数说明： indow: 信息窗口对象 InfoWindow。 lnglat: 在地图上打开的地理位置坐标。
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。

1.1.8 控件方法

方法	返回值	说明
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。 参数说明： control: 控件对象。
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。 参数说明： control: 控件对象。
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 Control.initialize() 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。

1.1.9 地图图层方法

方法	返回值	说明
getLayers()	array<TileLayer>	获取所有叠加层对象，可以自己判断需要移除哪些叠加层。
addLayer(layer:TileLayer)	none	给地图添加一个叠加层对象。 参数说明： layer: TileLayer 对象。
removeLayer(layer:TileLayer)	none	移除一个叠加层对象。 参数说明： layer: TileLayer 对象。
clearLayers()	none	移除所有叠加层对象。

1.1.10 事件

事件	参数	描述
click	{ type,target,lnglat,containerPoint }	左键单击地图时触发此事件。
dblclick	{ type,target,lnglat,containerPoint }	鼠标双击地图时会触发此事件。
contextmenu	{ type,target,lnglat,containerPoint }	右键单击地图时触发此事件。
mousemove	{ type,target,lnglat,containerPoint }	鼠标在地图区域移动过程中触发此事件。
mouseover	{ type, target,lnglat,containerPoint }	鼠标移入地图区域时触发此事件。
mouseout	{ type, target,lnglat,containerPoint }	鼠标移出地图区域时触发此事件。
movestart	{ type, target }	地图移动开始时触发此事件。

move	{ 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。
zoomend	{ type, target }	地图更改缩放级别结束时触发此事件。
addoverlay	{ type, target, addoverlay }	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。
removeoverlay	{ type, target, removeoverlay }	当使用 Map.removeOverlay() 方法移除单个覆盖物时会触发此事件。
addcontrol	{ type, target, addcontrol }	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, removecontrol }	当使用 Map.removeControl() 方法移除单个控件时会触发此事件。
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 }	开始拖拽地图时触发。
drag	{ type, target }	拖拽地图过程中触发。
dragend	{ type, target }	停止拖拽地图时触发。
layeradd	{ type, target, layer }	添加一个自定义地图图层时触发此事件。
layerremove	{ type, target, layer }	移除一个自定义地图图层时触发此事件。
load	{ type, target }	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瓦片。
resize	{ type, target, newSize, oldSize }	地图可视区域大小发生变化时会触发此事件。
levels	{ type, target, minzoom, maxzoom }	调用 setMinZoom 和 setMaxZoom 时会触发此事件。
touchstart	{ type, target, lnglat, containerPoint }	触摸开始时触发此事件，仅适用移动设备。
touchmove	{ type, target, lnglat, containerPoint }	触摸移动时触发此事件，仅适用移动设备。
touchend	{ type, target, lnglat, containerPoint }	触摸结束时触发此事件，仅适用移动设备。
longpress	{ type, target, lnglat, containerPoint }	长按事件，仅适用移动设备。

2 图层类

2.1 TileLayer 类

通过此类实现在地图上叠加自定义的地图图块层。

2.2.1 TileLayerOptions 属性

属性	类型	默认值	说明
minZoom	number	0	此图层的最低缩放级别。
maxZoom	number	18	此图层的最高缩放级别。
errorTileUrl	string	""	当没有瓦片时所显示的错误图片地址。
opacity	number	1.0	设置图层的透明度 (0.0-1.0)。默认值为 1.0 不透明。
zIndex	number	null	图层的显示顺序。
bounds	LngLatBounds	null	设置指定范围内显示瓦片。

2.2.2 构造函数

构造函数	说明
TileLayer(url:String,opts:TileLayerOptions)	以指定的参数创建 TTileLayer 对象,这个对象是在地图上叠加栅格地图的时候用到的。 参数说明： url: 图层服务地址。 opts: 图层属性对象，请参考 TileLayerOptions。

2.2.3 方法

方法	返回值	说明
setOpacity(opacity:Number)	this	改变图层的透明度。 参数说明： opacity: 取值范围 0-1 之间。
setZIndex(zIndex:Number)	this	设置图层的叠放顺序。 参数说明： zIndex: 大于 1 的整数。
refresh()	this	重新加载所有添加到此 TileLayer 的可见瓦片。
setUrl(url:String)	this	更新图层的 URL 地址并重绘他们。 参数说明： url: 图层服务地址。
getContainer()	HTMLelement	瓦片的标签容器。

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。

2.2.4 事件

事件	参数	描述
loading	{type, target}	当瓦片图层开始加载瓦片时触发。
load	{type, target}	当瓦片图层加载完可见瓦片后触发。
tileloadstart	{type, target, coords, tile}	瓦片请求和开始加载时触发。
tileload	{type, target, coords, tile}	在加载瓦片时触发。
tileunload	{type, target, coords, tile}	在瓦片被移除时触发（比如打开了 unloadInvisibleTiles）。
tileerror	{type, target, coords, tile}	瓦片错误时触发。

2.2 TileLayer.TDT 类

通过此类实现在地图上叠加天地图矢量地图服务，此接口只支持叠加经纬度投影的矢量地图服务。

2.3.1 TileLayer.TDTOptions

属性	类型	默认值	说明
minZoom	number	0	此图层的最低缩放级别。
maxZoom	number	18	此图层的最高缩放级别。
errorTileUrl	string	""	当没有瓦片时所显示的错误图片地址。
attribution	string	""	用来描述了图层信息。
opacity	number	1.0	设置图层的透明度（0.0-1.0）。默认值为 1.0 不透明。
zIndex	number	null	图层的显示顺序。
bounds	LngLatBounds	null	设置指定范围内显示瓦片。

2.3.2 构造函数

构造函数	说明
TileLayer.TDT(opts:TileLayer.TDTOptions)	以指定的参数创建 TTileLayer.TDT 对象,这个对象是在地图上叠加栅格地图的时候用到的。 参数说明：

	opts: 天地图图层属性对象, 请参考 TileLayer. TDTOptions。
--	---

2.3.3 方法

方法	返回值	说明
setOpacity(opacity:Number)	this	改变图层的透明度。
setZIndex(zIndex:Number)	this	设置图层的叠放顺序。
refresh ()	this	重新加载所有添加到此 TileLayer 的可见瓦片。
setUrl(url:String)	this	更新图层的 URL 地址并重绘他们。
getContainer()	HTMLElement	瓦片的标签容器。
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。

2.3.4 事件

事件	参数	描述
loading	{ type, target }	当瓦片图层开始加载瓦片时触发。
load	{ type, target }	当瓦片图层加载完可见瓦片后触发。
tileloadstart	{ type, target, coords, tile }	瓦片请求和开始加载时触发。
tileload	{ type, target, coords, tile }	在加载瓦片时触发。
tileunload	{ type, target, coords, tile }	在瓦片被移除时触发（比如打开了 unloadInvisibleTiles）。
tileerror	{ type, target, coords, tile }	瓦片错误时触发。

2.3 TileLayer.WMS 类

通过此类实现在地图上叠加自定义的 WMS 地图图块层。

2.4.1 TileLayer.WMSOptions 属性

属性	类型	默认值	说明
layers	string	""	用","分隔的多个图层列表。
styles	string	""	每个请求图层的用","分隔的描述样式。
format	string	"image/jpeg"	输出图像的类型。
transparent	boolean	false	输出图像背景是否透明。

version	string	"1.1.1"	请求服务的版本
srs	String	" EPSG:900913"	地图投影类型。

2.4.2 构造函数

构造函数	说明
TileLayer.WMS(opts:TileLayer.WMSOptions)	以指定的参数创建 TileLayer.WMS 对象,这个对象是在地图上叠加自定义栅格地图。 参数说明： opts: WMS 属性对象，请参考 TileLayer.WMSOptions。

2.4.3 方法

方法	返回值	说明
setOpacity(opacity:Number)	this	改变图层的透明度。
setZIndex(zIndex:Number)	this	设置图层的叠放顺序。
refresh ()	this	重新加载所有添加到此 TileLayer 的可见瓦片。
setUrl(url:String)	this	更新图层的 URL 地址并重绘他们。
getContainer()	HTMLElement	瓦片的标签容器。
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。

2.4.4 事件

事件	参数	描述
loading	{ type, target }	当瓦片图层开始加载瓦片时触发。
load	{ type, target }	当瓦片图层加载完可见瓦片后触发。
tileloadstart	{ type, target, coords, tile }	瓦片请求和开始加载时触发。
tileload	{ type, target, coords, tile }	在加载单个瓦片时触发。
tileunload	{ type, target, coords, tile }	在单个瓦片被移除时触发。
tileerror	{ type, target, coords, tile }	单个瓦片错误时触发。

3 控件类

3.1 Control 类

此类是所有控件类的基类，你可以通过此类来自定义控件，所有控件均包含 Control 类的属性、方法和事件。

3.1.1 ControlPosition 常量

常量	说明
T_ANCHOR_TOP_LEFT	控件将定位到地图的左上角。
T_ANCHOR_TOP_RIGHT	控件将定位到地图的右上角。
T_ANCHOR_BOTTOM_LEFT	控件将定位到地图的左下角。
T_ANCHOR_BOTTOM_RIGHT	控件将定位到地图的右下角。

3.1.2 ControlOptions 属性

属性	类型	默认值	说明
position	enumControlPosition	T_ANCHOR_TOP_RIGHT	控件的停靠位置。参数请参见 ControlPosition 常量。

3.1.3 方法

方法	返回值	说明
setPosition(position:enumControlPosition)	none	设置控件的位置。参数请参见 ControlPosition 常量。
getPosition()	enumControlPosition	返回控件的位置。
onAdd(map:Map)	none	向地图上添加叠加物。当调用 map.addControl 时，API 将调用此方法。自定义控件时需要实现此方法。自定义控件时需要将控件对应的 HTML 元素返回。
onRemove()	none	移除控件，释放控件对象所占用的内存。自定义控件时需要实现此方法。
getContainer()	HTMLElement	返回控件所在的容器的标签。
show()	none	显示控件。
hide()	none	隐藏控件。
isVisible()	boolean	判断控件的可见性。
setOffset(offset:Point)	none	设置控件停靠的偏移量。
getOffset()	Point	返回控件停靠的偏移量。
setOptions(opt:JSON)	JSON	对 ControlOptions 属性值赋值。

3.2 Control.Zoom 类

地图缩放控件，可以缩放和移动地图。

3.2.1 Control.ZoomOptions 属性

属性	类型	默认值	说明
position	enumControlPosition	T_ANCHOR_TOP_RIGHT	控件初始位置,在四角位置之一。
zoomInText	string	"+"	放大层级按钮的文字显示。
zoomOutText	string	"-"	缩小层级按钮的文字显示。
zoomInTitle	string	"放大"	放大层级按钮的标题显示。
zoomOutTitle	string	"缩小"	缩小层级按钮的标题显示。

3.2.2 构造函数

构造函数	说明
Control.Zoom(opts:Control.ZoomOptions)	创建一个地图缩放控件。 参数说明： opts：地图缩放控件属性对象，请参考 Control.ZoomOptions。

3.3 Control.Scale 类

地图比例尺控件，用来实时的显示地图的当前比例尺。

3.3.1 Control.ScaleOptions 属性

属性	类型	默认值	说明
position	String	T_ANCHOR_BOTTOM_LEFT	控制的位置（在地图的某一角）。

3.3.2 构造函数

构造函数	说明
Control.Scale(opts:Control.ScaleOptions)	创建一个地图缩放控件。 参数说明： opts: 比例尺属性对象，请参考 Control.ScaleOptions。

3.3.3 方法

方法	返回值	说明
setColor(color:String)	none	设置设置比例尺控件的颜色。 参数说明： color: 用于配置比例尺的颜色，例如“#000000”、“red”。

3.4 Control.Copyright 类

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。

3.4.1 Control.CopyrightOptions 属性

属性	类型	默认值	说明
id	string	null	该版权信息的唯一标识符。
content	string	null	该版权的文本信息，用于显示在地图上，支持 HTML 内容。
bounds	LngLatBounds	null	该版权信息所适用的地理区域。

3.4.2 构造函数

构造函数	说明
Control.Copyright()	创建一个版权控件。

3.4.3 方法

方法	返回值	说明
addCopyright(copyright:CopyrightOptions)	none	添加一个版权信息。

		参数说明： copyright：请参考 CopyrightOptions。
removeCopyright(copyright:CopyrightOptions)	none	移除版权信息。 参数说明： copyright：请参考 CopyrightOptions。
getCopyright(string:id)	CopyrightOptions	获得单个版权信息。id 参数 为获取单个版权信息的唯一 标识。 参数说明： id：唯一标识。
getCopyrightCollection()	array	获得版权信息列表。

3.5 Control.OverviewMap 类

鹰眼地图控件，用来显示一个鹰眼地图，继承自 Control 基类，拥有基类的所有方法。

3.4.4 Control.OverviewMapOptions 属性

属性	类型	默认值	说明
anchor	string	T_ANCHOR_TOP_RIGHT	控件的停靠位置。 属性说明： T_ANCHOR_TOP_LEFT：表示左上。 T_ANCHOR_TOP_RIGHT：表示右上。 T_ANCHOR_BOTTOM_LEFT：表示左下。 T_ANCHOR_BOTTOM_RIGHT：表示右下。
size	Point	Point(0,0)	缩略地图控件的大小。
isOpen	boolean	false	缩略地图添加到地图后的开合状态，默认为 关闭。true 表示显示，false 表示隐藏。

3.4.5 构造函数

构造函数	说明
OverviewMapControl(opts: Control.OverviewMapOptions)	创建一个 Control.OverviewMap 控件。

3.4.6 方法

方法	返回值	说明
setButtonImage(imgOpen:String,imgClose:String)	none	设置鹰眼地图的按钮图片,要同时设置鹰眼地图打开和关闭状态下的两张图片。 imgOpen: 鹰眼地图在打开状态下的按钮图片。 imgClose: 鹰眼地图在关闭状态下的按钮图片。
setBorderColor(color:String)	none	设置鹰眼地图和主地图之间空隙的背景颜色。
setRectBorderColor(color:String)	none	设置鹰眼地图上的矩形框边框颜色。
setRectBackColor(color:String)	none	设置鹰眼地图上的矩形框背景颜色。
changeView()	none	切换鹰眼地图的开-合状态。
isOpen()	boolean	返回该鹰眼的视图是否被打开。
getMiniMap()	Map	返回该鹰眼的地图对象。
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。

3.4.7 事件

事件	参数	说明
viewchange	{type,target,isOpen }	鹰眼视图控件的开合状态变化时触发事件。 参数说明: isOpen: 状态变化后鹰眼地图是否在打开的状态中。

4 覆盖物类

4.1 Overlay 类

叠加层类都直接或间接继承于此基类。如果希望在地图上显示自定义的叠加层对象类型,可以继承这一基类。

4.1.1 OverlayOptions 属性

属性	类型	默认值	说明
type	number	0	叠加层类型，已定义的类型有： Label 类型为 1; Marker 类型为 2; InfoWindow 类型为 3; Polyline 类型为 4; Polygon 类型为 5; Rectangle 类型为 6; Circle 类型为 8; 对于自定义叠加层，也可以设定一个 Number 型的叠加层类型。

4.1.2 方法

方法	返回值	说明
initialize(opt:JSON)	none	构造函数时传递参数，对 OverlayOptions 属性值进行赋值。
onAdd(map:Map)	none	向地图上添加叠加物。当调用 map.addOverLay 时，API 将调用此方法。自定义叠加物时需要实现此方法。自定义叠加物时需要将覆盖物对应的 HTML 元素返回。 参数说明： map: Map 对象。
onRemove()	none	移除叠加物，释放覆盖物对象所占用的内存。自定义叠加物时需要实现此方法。
update()	none	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。
show()	none	在地图上显示叠加层。
hide()	none	在地图上隐藏叠加层。
isHidden()	boolean	判断叠加层是否隐藏，返回 true 表示当前叠加层是隐藏的，否则是处于显示状态。
setOptions(opt:JSON)	JSON	对 OverlayOptions 属性值赋值。
getElement	HTMLElement	返回叠加物所在的容器的标签。

4.2 Label 类

此类表示地图上的文本标注。

4.2.1 LabelOptions 属性

属性	类型	默认值	说明
text	string	""	文本标注的内容。
offset	Point	Point(0,0)	文本标注的位置偏移值。
position	LngLat	LngLat (0,0)	文本标注的地理位置。

4.2.2 构造函数

构造函数	说明
Label(opts:LabelOptions)	根据给定参数对象创建文本标注。 参数说明： opts: 文本标注的属性对象，请参考 LabelOptions。

4.2.3 方法

方法	返回值	说明
getType()	number	叠加层类型。
getLngLat ()	LngLat	返回文本所在的地理位置坐标。
setLngLat (lnglat: LngLat)	none	设置文本所在的地理位置坐标。 参数说明： lnglat: 地理位置坐标。
setOffset(offset: Point)	none	设置文本的偏移值。 参数说明： offset: 偏移量（像素）。
getOffset()	Point	返回文本的偏移值。
setLabel(content:String)	none	设置文本的内容。 参数说明： content: 文本的内容。
getLabel()	string	返回文本的内容。
setTitle(title:String)	none	设置文本的提示内容。 参数说明： title: 文本的提示内容。
getTitle()	string	返回文本的提示内容。
setZindex()	none	设置 z-index。
setFontSize (size: number)	none	设置文本的文本内容字体大小。 参数说明： size: 字体大小。
getFontSize ()	number	返回文本内容字体大小。
setFontcolor(color:String)	none	设置文本的文本的字体颜色。

		参数说明： color: 字体颜色。
getFontColor()	string	返回文本的字体颜色。
setBackgroundColor(color:String)	none	设置文本的背景色。 参数说明： color: 背景颜色。
getBackgroundColor()	string	返回文本的背景色。
setBorderLine (width: number)	none	设置文本的边框线宽。 参数说明： width: 边框线宽。
getBorderLine ()	number	返回文本的边框线宽。
setBorderColor(color:String)	none	设置文本的边框颜色。 参数说明： color: 边框颜色。
getBorderColor()	string	返回文本的边框颜色。
setOpacity(opacity:Number)	none	设置文本的显示不透明度。 参数说明： opacity: 不透明度（范围 0-1 之间）。
getOpacity()	number	返回文本的显示透明度。
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。

4.2.4 事件

事件	事件传参类型	说明
click	{type,target,lnglat,containerPoint }	点击多边形后会触发此事件。
dblclick	{type,target,lnglat,containerPoint }	双击多边形后会触发此事件。
mousedown	{type,target,lnglat,containerPoint }	鼠标在多边形上按下触发此事件。
mouseup	{type,target,lnglat,containerPoint }	鼠标在多边形释放触发此事件。
mouseout	{type,target,lnglat,containerPoint }	鼠标离开多边形时触发此事件。

4.3 Marker 类

此类表示地图上一个图像标注。

4.3.1 MarkerOptions 属性

属性	类型	默认值	说明
icon	Icon	*	图标类用来表达注记。默认设置为 new

			L.Icon.Default()。
draggable	boolean	false	决定注记是否可被鼠标或触摸拖动。
title	string	"	默认情况下，注记图片的叠置顺序由纬度自动设置。如果你想将某一注记放置于其他之上可用这个选项，设置一个较大的值即可，比如 1000（或是相反地设置一个较大的负值）。
zIndexOffset	number	0	设置 z-index。
opacity	number	1.0	设置透明度。

4.3.2 构造函数

构造函数	说明
Marker lnglat:LngLat[,opts:MarkerOptions])	<p>创建一个图像标注实例。lnglat 参数指定了图像标注所在的地理位置。</p> <p>参数说明：</p> <p>lnglat: 标注点的地理坐标。</p> <p>opts: 标注的属性对象，请参考 MarkerOptions。</p>

4.3.3 方法

方法	返回值	说明
getType()	number	叠加层类型。
getMap()	Map	返回标注所在的 map 对象。
setIcon(icon:Icon)	none	设置标注所用的图标对象。
getIcon()	Icon	返回标记显示时所使用的图标对象。
setLngLat (lnglat:LngLat)	none	<p>设置标注所在的地理位置坐标。</p> <p>参数说明：</p> <p>lnglat: 地理位置坐标。</p>
getLngLat ()	LngLat	返回标注所在的地理位置坐标。
setZindex(num:Number)	none	<p>设置 z-index。</p> <p>参数说明：</p> <p>num: 大于 1 的整数。</p>
enableDragging()	none	开启标注拖拽功能。
disableDragging()	none	关闭标注拖拽功能。
setOpacity(opacity:Number)	none	<p>设置标注透明度。</p> <p>参数说明：</p> <p>opacity: 取值范围 0-1 之间。</p>
openInfoWindow(infowin:InfoWindow)	none	<p>打开信息窗。</p> <p>参数说明：</p> <p>infowin: InfoWindow 对象。</p>

closeInfoWindow()	none	关闭信息窗。
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。

4.3.4 事件

事件	参数	描述
click	{ type,target,lnglat,containerPoint }	点击标注图标后会触发此事件。
dblclick	{ type,target,lnglat,containerPoint }	双击标注图标后会触发此事件。
mousedown	{ type,target,lnglat,containerPoint }	鼠标在标注图标上按下触发此事件。
mouseup	{ type,target,lnglat,containerPoint }	鼠标在标注图标释放触发此事件。
mouseout	{ type,target,lnglat,containerPoint }	鼠标离开标注图标时触发此事件。
mouseover	{ type,target,lnglat,containerPoint }	当鼠标进入标注图标区域时会触发此事件。
dragstart	{ type,target }	当用户拖动标注图标时触发。
drag	{ type,target,lnglat }	当用户拖动标注图标时不断触发。
dragend	{ type,target,lnglat }	当用户停止拖动标注图标时触发。
remove	{ type,target }	移除标注图标时触发。

4.4 Icon 类

此类表示标注覆盖物所使用的图标。

4.4.1 IconOptions 属性

属性	类型	默认值	说明
iconUrl	String	""	请求图标图片的 URL（脚本中的绝对或相对路径）。
iconSize	Point	Point(25, 41)	图标可视区域的大小。
iconAnchor	Point	Point(12, 41)	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。

4.4.2 构造函数

构造函数	说明
------	----

Icon(opts:IconOptions)	以给定的图像地址和大小创建图标对象实例。 参数说明： opts: 图标的属性对象，请参考 TIconOptions。
------------------------	--

4.4.3 方法

方法	返回值	说明
setIconUrl(iconUrl:String)	none	设置图片资源的地址。 参数说明： iconUrl: 图标地址。
getIconUrl()	string	获取图标使用的图片 URL。
setIconSize(size:Point)	none	设置图标可视区域的大小。 参数说明： size: 图标大小。
getIconSize()	Point	获取图标可视区域的大小。
setIconAnchor(anchor:Point)	none	设置相对于图片左上角的像素距离。 参数说明： anchor: 对于图片左上角的像素距离。
getIconAnchor()	Point	获取相对于图片左上角的像素距离。

4.5 InfoWindow 类

此类表示地图上包含信息的窗口。

4.5.1 InfoWindowOptions 属性

属性	类型	默认值	说明
minWidth	number	50	弹出框的最小宽度。
maxWidth	number	300	弹出框的最大宽度。
maxHeight	number	null	设置后，如果内容超过弹出窗口的给定高度则产生一个可以滚动的容器。
autoPan	boolean	false	是否开启信息窗口打开时地图自动移动（默认关闭）。
closeButton	boolean	true	控制弹出窗口中出现的关闭按钮。
offset	Point	Point(0, 7)	弹出窗口位置的补偿值。在同一图层中打开弹出窗口时对于控制锚点比较有用。
autoPanPadding	Point	Point(5, 5)	在地图视图自动平移产生后弹出窗口和地图视图之间的边缘。
closeOnClick	boolean	false	是否开启点击地图关闭信息窗口（默认关闭）。

4.5.2 构造函数

构造函数	说明
InfoWindow(content:String HTML Element[,opts:TInfoWindowOptions])	<p>创建一个信息窗实例。</p> <p>参数说明：</p> <p>content: 信息窗口的内容。</p> <p>opts: 信息窗口的属性对象，请参考 InfoWindowOptions。</p>

4.5.3 方法

方法	返回值	说明
getType()	number	叠加层类型。
setLngLat(lnglat:LngLat)	none	<p>设置或改变信息浮窗所指向的地理位置坐标。</p> <p>参数说明：</p> <p>lnglat: 地理位置坐标。</p>
getLngLat()	LngLat	返回信息浮窗所指向的地理位置坐标。
setOffset(point:Point)	none	<p>设置信息浮窗显示时向右下角偏移量（像素）。</p> <p>参数说明：</p> <p>point: 偏移量（像素）。</p>
getOffset()	Point	返回信息浮窗显示时向右下角偏移量。
isOpen()	boolean	返回信息窗口的打开状态。
setContent(content:String HTMLElement)	none	<p>设置信息浮窗的显示 HTML 内容。</p> <p>参数说明：</p> <p>content: html 内容。</p> <p>该参数有二种类型：1, HTML 对象；2, HTML 字符串。</p>
getContent()	String HTMLElement	返回信息浮窗的显示 HTML 内容。
update()	none	重绘信息窗口，当信息窗口内容发生变化时进行调用。
closeInfoWindow()	none	关闭信息浮窗。
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。

4.5.4 事件

事件	参数	描述
close	{type, target, lnglat}	信息窗口被关闭时触发此事件。
open	{type, target, lnglat}	信息窗口被打开时触发此事件。
clickclose	{type, target}	点击信息窗口的关闭按钮时触发此事件。

4.6 Polyline 类

使用浏览器的矢量制图工具在地图上绘制折线的地图叠加层。

4.6.1 PolylineOptions 属性

属性	类型	默认值	说明
color	string	"#0000FF"	折线颜色。
weight	number	3	折线的宽度，以像素为单位。
opacity	number	0.5	折线的透明度（范围 0-1 之间）。
lineStyle	string	"solid"	折线的样式（solid 或 dashed）。

4.6.2 构造函数

构造函数	说明
Polyline(points:Array<LngLat>[,opts:PolylineOptions])	<p>创建折线覆盖物对象。</p> <p>参数说明：</p> <p>points：坐标数组。</p> <p>opts：线形的属性对象，请参考 PolylineOptions。</p>

4.6.3 方法

方法	返回值	说明
getType()	number	叠加层类型。
setLngLats(lnglats:Array<LngLat>)	none	<p>设置折线的点数组。</p> <p>参数说明：</p> <p>lnglats：顶点数组。</p>
getLngLats()	Array<LngLat>	返回折线的点数组。
setColor(color:String)	none	<p>设置折线的颜色。</p> <p>参数说明：</p> <p>color：合法的 CSS 颜色值。</p>

getColor()	string	返回折线的颜色。
setOpacity(opacity:Number)	none	设置折线的透明度。 参数说明： opacity: 范围 0-1 之间。
getOpacity()	number	返回折线的透明度。
setWeight(weight:Number)	none	设置折线的宽度。 参数说明： weight: 范围为大于等于 1 的整数。
getWeight()	number	返回线的宽度。
setLineStyle(style:String)	none	设置折线是为实线或虚线。 参数说明： style: solid 或 dashed。
getLineStyle()	string	返回当前折线样式状态，实线或者虚线。
getBounds()	LngLatBounds	返回折线的地理区域范围。
getMap()	Map	返回折线所在的 map 对象。
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。
enableEdit()	none	启用线编辑功能
disableEdit()	none	禁用线编辑功能
isEditable()	boolean	是否启用线编辑功能，true 表示启用，false 表示禁止。

4.6.4 事件

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

4.7 Polygon 类

此类表示一个多边形覆盖物。

4.7.1 PolygonOptions 属性

属性	类型	默认值	说明
color	string	"#0000FF"	多边形边线颜色。
weight	number	3	多边形边线的宽度，以像素为单位。
opacity	number	0.5	多边形边线的透明度（范围 0-1 之间）。
fillColor	String	"#0000FF"	多边形填充颜色。当参数为空时，折线覆盖物将没有填充效果。
fillOpacity	number	0.2	多边形填充的透明度（范围 0-1 之间）。
lineStyle	string	"solid"	多边形边线的样式（solid 或 dashed）。

4.7.2 构造函数

构造函数	说明
Polygon(points:Array<LngLat>[,opts:PolygonOptions])	<p>创建多边形覆盖物。</p> <p>参数说明：</p> <p>points: 坐标数组。</p> <p>opts: 多边形的属性对象，请参考 PolygonOptions。</p>

4.7.3 方法

方法	返回值	说明
getType()	number	叠加层类型。
setLngLats(lnglats:Array<LngLat>)	none	<p>设置多边形的点数组。</p> <p>参数说明：</p> <p>lnglats: 顶点数组。</p>
getLngLats()	Array<LngLat>	返回多边形的点数组。
setColor(color:String)	none	<p>设置多边形边线的颜色。</p> <p>参数说明：</p> <p>color: 合法的 CSS 颜色值。</p>
getColor()	string	返回多边形边线的颜色。
setOpacity(opacity:Number)	none	<p>设置多边形边线的透明度。</p> <p>参数说明：</p> <p>opacity: 范围 0-1 之间。</p>
getOpacity()	number	返回多边形边线的透明度。
setWeight(weight:Number)	none	<p>设置多边形边线的宽度。</p> <p>参数说明：</p> <p>weight: 范围为大于等于 1 的整数。</p>
getWeight()	number	返回多边形边线的宽度。
setLineStyle(style:String)	none	设置多边形边线是为实线或虚线。

		参数说明： style: solid 或 dashed。
getLineStyle()	string	返回当前多边形边线样式状态，实线或者虚线。
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，折线覆盖物填充颜色与边线颜色相同。 参数说明： color: 合法的 CSS 颜色值。
getFillColor()	string	返回多边形的填充颜色。
setFillOpacity(opacity:Number)	none	设置多边形的填充透明度。当参数为 0 时，折线覆盖物将没有填充效果。 参数说明： opacity: 取值范围 0-1 之间。
getFillOpacity()	number	返回多边形的填充透明度。
getBounds()	LngLatBounds	返回多边形的地理区域范围。
getMap()	Map	返回多边形所在的 map 对象。
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。
enableEdit()	none	启用线编辑功能
disableEdit()	none	禁用线编辑功能
isEditable()	boolean	是否启用线编辑功能，true 表示启用，false 表示禁止。

4.7.4 事件

事件	参数	描述
click	{ type,target,lnglat,containerPoint }	点击多边形后会触发此事件。
dblclick	{ type,target,lnglat,containerPoint }	双击多边形后会触发此事件。
mousedown	{ type,target,lnglat,containerPoint }	鼠标在多边形上按下触发此事件。
mouseup	{ type,target,lnglat,containerPoint }	鼠标在多边形释放触发此事件。
mouseout	{ type,target,lnglat,containerPoint }	鼠标离开多边形时触发此事件。
mouseover	{ type,target,lnglat,containerPoint }	当鼠标进入多边形区域时会触发此事件。
remove	{ type,target }	移除多边形时触发。

4.8 Rectangle 类

此类用来表示地图上的一个矩形图形对象。

4.8.1 RectangleOptions 属性

属性	类型	默认值	说明
color	string	"#0000FF"	矩形边线颜色。
weight	number	3	矩形边线的宽度，以像素为单位。
opacity	number	0.5	矩形边线的透明度（范围 0-1 之间）。
fillColor	String	"#0000FF"	矩形填充颜色。当参数为空时，折线覆盖物将没有填充效果。
fillOpacity	number	0.2	矩形填充的透明度（范围 0-1 之间）。
lineStyle	string	"solid"	矩形边线的样式（solid 或 dashed）。

4.8.2 构造函数

构造函数	说明
Rectangle(bounds:LngLatBounds[, opts:RectangleOptions])	<p>创建矩形覆盖物。</p> <p>参数说明：</p> <p>bounds：矩形范围。</p> <p>opts：矩形的属性对象，请参考 RectangleOptions。</p>

4.8.3 方法

方法	返回值	说明
getType()	number	叠加层类型。
setBounds(bounds:LngLatBounds)	none	<p>设置矩形的显示范围。</p> <p>参数说明：</p> <p>bounds：矩形的坐标范围。</p>
getBounds()	LngLatBounds	返回矩形的地理区域范围。
setColor(color:String)	none	<p>设置矩形边线的颜色。</p> <p>参数说明：</p> <p>color：合法的 CSS 颜色值。</p>
getColor()	string	返回矩形边线的颜色。
setOpacity(opacity:Number)	none	<p>设置矩形边线的透明度。</p> <p>参数说明：</p> <p>opacity：范围 0-1 之间。</p>
getOpacity()	number	返回矩形边线的透明度。
setWeight(weight:Number)	none	<p>设置矩形边线的宽度。</p> <p>参数说明：</p> <p>weight：范围为大于等于 1 的整数。</p>
getWeight()	number	返回矩形边线的宽度。
setLineStyle(style:String)	none	设置矩形边线是为实线或虚线。

		参数说明： style: solid 或 dashed。
getLineStyle()	string	返回当前矩形边线样式状态，实线或者虚线。
setFillColor(color:String)	none	设置矩形的填充颜色，参数为合法的 CSS 颜色值。当参数为空字符串时，矩形覆盖物填充颜色与边线颜色相同。 参数说明： color: 合法的 CSS 颜色值。
getFillColor()	string	返回矩形的填充颜色。
setFillOpacity(opacity:Number)	none	设置矩形的填充透明度。当参数为 0 时，矩形覆盖物将没有填充效果。 参数说明： opacity: 取值范围 0-1 之间。
getFillOpacity()	number	返回矩形的填充透明度。
getMap()	Map	返回矩形所在的 map 对象。
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。
enableEdit()	none	启用线编辑功能
disableEdit()	none	禁用线编辑功能
isEditable()	boolean	是否启用线编辑功能，true 表示启用，false 表示禁止。

4.8.4 事件

事件	参数	描述
click	{ type,target,lnglat,containerPoint }	点击矩形后会触发此事件。
dblclick	{ type,target,lnglat,containerPoint }	双击矩形后会触发此事件。
mousedown	{ type,target,lnglat,containerPoint }	鼠标在矩形上按下触发此事件。
mouseup	{ type,target,lnglat,containerPoint }	鼠标在矩形释放触发此事件。
mouseout	{ type,target,lnglat,containerPoint }	鼠标离开矩形时触发此事件。
mouseover	{ type,target,lnglat,containerPoint }	当鼠标进入矩形区域时会触发此事件。
remove	{ type,target }	移除矩形时触发。

4.9 Circle 类

此类表示地图上的圆覆盖物。

4.9.1 CircleOptions 属性

属性	类型	默认值	说明
color	string	"#0000FF"	圆边线颜色。
weight	number	3	圆边线的宽度，以像素为单位。
opacity	number	0.5	圆边线的透明度（范围 0-1 之间）。
fillColor	String	"#0000FF"	圆填充颜色。当参数为空时，折线覆盖物将没有填充效果。
fillOpacity	number	0.2	圆填充的透明度（范围 0-1 之间）。
lineStyle	string	"solid"	圆边线的样式（solid 或 dashed）。

4.9.2 构造函数

构造函数	说明
Circle(center:LngLat,radius:Number[,opts:CircleOptions])	<p>创建圆覆盖物。</p> <p>参数说明：</p> <p>center: 圆心经纬度坐标。</p> <p>radius: 圆的半径，以米点单位。</p> <p>opts: 圆形的属性对象，请参考 CircleOptions。</p>

4.9.3 方法

方法	返回值	说明
getType()	number	叠加层类型。
setCenter(lnglat:LngLat)	none	<p>设置圆的中心点。</p> <p>参数说明：</p> <p>lnglat: 中心点对象。</p>
getCenter()	LngLat	返回圆的中心点。
setRadius(radius:Number)	none	<p>设置圆的半径。</p> <p>参数说明：</p> <p>radius: 半径（单位米）。</p>
getRadius()	number	返回圆的半径。
getBounds()	LngLatBounds	返回矩形的地理区域范围。
setColor(color:String)	none	<p>设置圆边线的颜色。</p> <p>参数说明：</p> <p>color: 合法的 CSS 颜色值。</p>
getColor()	string	返回圆边线的颜色。
setOpacity(opacity:Number)	none	<p>设置圆边线的透明度。</p> <p>参数说明：</p> <p>opacity: 范围 0-1 之间。</p>
getOpacity()	number	返回圆边线的透明度。

setWeight(weight:Number)	none	设置圆边线的宽度。 参数说明： weight: 范围为大于等于 1 的整数。
getWeight()	number	返回圆边线的宽度。
setLineStyle(style:String)	none	设置圆边线是为实线或虚线。 参数说明： style: solid 或 dashed。
getLineStyle()	string	返回当前圆边线样式状态，实线或者虚线。
setFillColor(color:String)	none	设置圆的填充颜色，参数为合法的 CSS 颜色值。当参数为空字符串时，矩形覆盖物填充颜色与边线颜色相同。 参数说明： color: 合法的 CSS 颜色值。
getFillColor()	string	返回圆的填充颜色。
setFillOpacity(opacity:Number)	none	设置圆的填充透明度。当参数为 0 时，矩形覆盖物将没有填充效果。 参数说明： opacity: 取值范围 0-1 之间。
getFillOpacity()	number	返回圆的填充透明度。
getMap()	Map	返回圆所在的 map 对象。
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。
enableEdit()	none	启用线编辑功能
disableEdit()	none	禁用线编辑功能
isEditable()	boolean	是否启用线编辑功能，true 表示启用，false 表示禁止。

4.9.1 事件

事件	参数	描述
click	{ type,target,lnglat,containerPoint }	点击圆后会触发此事件。
dblclick	{ type,target,lnglat,containerPoint }	双击圆后会触发此事件。
mousedown	{ type,target,lnglat,containerPoint }	鼠标在圆上按下触发此事件。
mouseup	{ type,target,lnglat,containerPoint }	鼠标在圆释放触发此事件。
mouseout	{ type,target,lnglat,containerPoint }	鼠标离开圆时触发此事件。
mouseover	{ type,target,lnglat,containerPoint }	当鼠标进入圆区域时会触发此事件。
remove	{ type,target }	移除圆时触发。

5 工具类

5.1 MarkTool 类

标注工具，用来让用户在地图上标注一个点，可以通过该工具获得用户标点的经纬度位置。

5.1.1 MarkToolOptions 属性

属性	类型	默认值	说明
icon	Icon	*	图标类用来表达注记。默认设置为 <code>new T.Icon.Default()</code> 。
follow	number	false	标记图标是否跟随鼠标。

5.1.2 构造函数

构造函数	说明
<code>MarkTool(map:Map[,opts:MarkToolOptions])</code>	创建一个 MarkTool 工具。 参数说明： <code>map</code> : 地图对象。 <code>opts</code> : 详见 MarkToolOptions 类。

5.1.3 方法

方法	返回值	说明
<code>setPointImage(url:String)</code>	none	设置标注工具显示的标注图标路径的路径。 参数说明： <code>url</code> : 标注图标的路径 URL。
<code>getMarkControlPoint()</code>	LngLat	获取用户标注点的坐标，如果用户尚未操作，则返回 <code>null</code> 。
<code>open()</code>	boolean	开启标注工具。
<code>close()</code>	none	关闭标注工具。
<code>clear()</code>	none	清除工具绘制的所有标注图标。
<code>getMarkers()</code>	Array<Marker>	获取所有工具绘制的标注图标。
<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。

5.1.4 事件

事件	参数	说明
mouseup	{type,target,currentLnglat,currentMarker,allMarkers}	在用户每完成一次标注时触发事件。 参数说明： currentLnglat: 用户在地图上标的坐标。 currentMarker: 用户当前的标注点对象。 allMarkers: 用户使用工具所有的标注点对象。

5.2 PolygonTool 类

多边形工具，可以通过事件来获取用户绘制的多边形，包含测面积功能。

5.2.1 PolygonToolOptions 属性

属性	类型	默认值	说明
color	string	"#0000FF"	多边形边线颜色。
weight	number	3	多边形边线的宽度，以像素为单位。
opacity	number	0.5	多边形边线的透明度（范围 0-1 之间）。
fillColor	String	"#0000FF"	多边形填充颜色。当参数为空时，折线覆盖物将没有填充效果。
fillOpacity	number	0.2	多边形填充的透明度（范围 0-1 之间）。
lineStyle	string	"solid"	多边形边线的样式（solid 或 dashed）。
showLabel	boolean	false	是否显示面积，如果不显示面积，则可以作为画面控件使用，默认值为 true。

5.2.2 构造函数

构造函数	说明
PolygonTool(map:Map[,opts:PolygonToolOptions])	创建一个 PolygonTool 控件对象。 参数说明： map: 地图对象。 opts: 详见 PolygonToolOptions 类。

5.2.3 方法

方法	返回值	说明
setTips(text:String)	none	跟随鼠标移动的说明文字。

		参数说明： text: 文字内容。
getArea(points:LngLat[])	number	计算由地理点组成的多边形的面积。 参数说明： points: 点坐标 LngLat 数组。
endDraw()	none	完成一个多边形的绘制，运行此方法相当于用户点击双击结束当前多边形的绘制。
getPolygons()	Array<Polygon>	获得所有绘制的多边形。
clear()	none	清除所有绘制的多边形。
open()	boolean	开启测面积工具。
close()	none	关闭侧面积工具。
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。

5.2.4 事件

事件	参数	说明
draw	{type,target,currentLnglats,currentArea,currentPolygon,allPolygons}	用户双击完成一次折线绘制时触发事件。 参数说明： currentLnglats: 用户当前绘制的多边形的点坐标数组。 currentArea: 用户最后绘制的多边形的地理面积。 currentPolygon: 当前所画多边形的对象。 allPolygons: 获取工具所有绘制的多边形
addpoint	{type,target,currentLnglats,currentArea,currentPolygon,allPolygons}	用户多边形绘制过程中,每次点击底图添加节点时触发事件。 参数说明： currentLnglats: 最后绘制的多边形的顶点坐标数组。 currentArea: 用户最后绘制的折线的地理面积。 currentPolygon: 当前临时绘制的多边形。 allPolygons: 获取工具所有绘制的多边形

5.3 PolylineTool 类

折线工具，可以通过事件来获取用户绘制的折线，包含测距功能。

5.3.1 PolylineToolOptions 属性

属性	类型	默认值	说明
color	string	"#0000FF"	折线颜色。
weight	number	3	折线的宽度，以像素为单位。
opacity	number	0.5	折线的透明度（范围 0-1 之间）。
lineStyle	string	"solid"	拆线的样式（solid 或 dashed）。
showLabel	boolean	false	是否显示距离，如果不显示距离，则可以作为画线控件使用，默认值为 true（可选）。

5.3.2 构造函数

构造函数	说明
PolylineTool(map:Map[,opts:PolylineToolOptions])	创建一个 PolylineTool 控件。 参数说明： map: 地图对象。 opts: 详见 PolylineToolOptions 类。

5.3.3 方法

方法	返回值	说明
setTips(text:String)	none	跟随鼠标移动的说明文字。 参数说明： text: 文本内容。
getDistance(LngLats:LngLat[])	number	计算一系列地理坐标点的距离总和。 参数说明： LngLats: 点坐标 LngLat 数组。
endDraw()	none	完成一个折线的绘制，运行此方法相当于用户点击双击结束当前折线的绘制。
clear()	none	清空工具绘制所有线。
getPolylines()	Array<Polyline>	获取所有编辑完成的线。
open()	boolean	开启测距工具。
close()	none	关闭测距工具。
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。

5.3.4 事件

事件	参数	说明
draw	{ type,target,currentLnglats,currentDistance,currentPolyline,allPolylines }	用户每次完成拉框操作时触发事件。 参数说明： currentLnglats: 用户当前绘制的折线的点坐标数组。 currentDistance: 用户当前绘制的折线的地理长度。 currentPolyline: 当前测距所画线的对象。 allPolylines: 返回所有工具绘制的线对象。
addpoint	{ type,target,currentLnglats,currentDistance,currentPolyline,allPolylines }	用户测距过程中，每次点击底图添加节点时触发事件。 参数说明： currentLnglats: 用户当前绘制的折线的点坐标数组。 currentDistance: 用户当前绘制的折线的地理长度。 currentPolyline: 当前测距所画线的对象。 allPolylines: 返回所有工具绘制的线对象。

5.4 RectangleTool 类

矩形工具，用来实现在地图上选择一个矩形区域或绘制矩形的功能。

5.4.1 RectangleToolOptions 属性

属性	类型	默认值	说明
color	string	"#0000FF"	矩形边线颜色。
weight	number	3	矩形边线的宽度，以像素为单位。
opacity	number	0.5	矩形边线的透明度（范围 0-1 之间）。
fillColor	String	"#0000FF"	矩形填充颜色。当参数为空时，折线覆盖物将没有填充效果。
fillOpacity	number	0.2	矩形填充的透明度（范围 0-1 之间）。
lineStyle	string	"solid"	矩形边线的样式（solid 或 dashed）。

5.4.2 构造函数

构造函数	说明
RectangleTool(map:Map[,opts:Re	创建一个 RectangleTool 控件。

ctToolOptions])	参数说明： map：地图对象 opts：详见 RectangleToolOptions 类
-----------------	---

5.4.3 方法

方法	返回值	说明
clear()	none	清除所有该控件在地图上绘制的矩形图形。
getRectangles()	Array<Rectangle>	返回工具左右绘制的矩形。
open()	boolean	开启画矩形工具。
close()	none	关闭画矩形工具。
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。

5.4.4 事件

事件	事件传参类型	说明
draw	{type,target,currentBounds,currentRectangle,allRectangles}	用户每次完成拉框操作时触发事件。 参数说明： currentBounds：用户拉框选择的地理范围。 currentRectangle：用户绘制的矩形图形对象。 allRectangles：所有绘制的矩形对象。

5.5 CircleTool 类

画圆工具，用来实现在地图上画圆的功能。

5.4.5 CircleToolOptions 属性

属性	类型	默认值	说明
color	string	"#0000FF"	圆边线颜色。
weight	number	3	圆边线的宽度，以像素为单位。
opacity	number	0.5	圆边线的透明度（范围 0-1 之间）。
fillColor	String	"#0000FF"	圆填充颜色。当参数为空时，折线覆盖物将没有填充效果。
fillOpacity	number	0.2	圆填充的透明度（范围 0-1 之间）。
lineStyle	string	"solid"	圆边线的样式（solid 或 dashed）。

5.4.6 构造函数

构造函数	说明
CircleTool(map:Map[,opts:CircleToolOptions])	<p>创建一个 CircleTool 控件。</p> <p>参数说明：</p> <p>map：地图对象。</p> <p>opts：详见 CircleToolOptions 类。</p>

5.4.7 方法

方法	返回值	说明
open()	boolean	开启画圆工具。
close()	none	关闭画圆工具。
getCircles()	Array<Circle>	获取工具中所有绘制的圆
Clear()	none	清空工具中所有绘制的圆。
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。

5.4.8 事件

事件	参数	说明
draw	{type,target,currentCenter,currentRadius,currentCircle,allCircles}	用户拖动绘制圆时触发。 参数说明： currentCenter: 中心点的地理坐标。 currentRadius: 半径，单位为米。 currentCircle: 用户最后绘制的圆对象。 allCircles: 用户所有绘制的圆对象。
drawend	{type,target,currentCenter,currentRadius,currentCircle,allCircles}	用户完成绘制圆时触发。 参数说明： currentCenter: 中心点的地理坐标。 currentRadius: 半径，单位为米。 currentCircle: 用户最后绘制的圆对象。 allCircles: 用户所有绘制的圆对象。

6 组件类

6.1 MarkerClusterer 类

用来解决加载大量标注点到地图上产生覆盖现象的问题，并提高性能。

6.1.1 MarkerClustererOptions 属性

属性	类型	默认值	说明
markers	array	[]	要聚合的标注点数组。
styles	array		自定义聚合后的图标风格，格式如下： [{ //图上 url url: 'http://api.tianditu.com/img/map/cluster/heart30.png', size: [30, 26], //图片大小 offset: new T.Point(-15, -13), //显示图片的偏移量 textColor: '#000000', //显示数字的颜色 textSize: 8, //显示文字的大小

			<pre> range: [0, 50], }, { url: 'http://api.tianditu.com/img/map/cluster/heart40.png', size: [42, 36], offset: new T.Point(-20, -17), textColor: '#ff0000', textSize: 10, range: [50, 400], }, { url: 'http://api.tianditu.com/img/map/cluster/heart50.png', size: [52, 46], soffset: new T.Point(-10, -22), textColor: 'white', textSize: 12, range: [400, 500], }] </pre>
girdSize	number	60	聚合计算时网格的像素大小，默认 60。
maxZoom	number	无穷大	最大的聚合级别，大于该级别就不进行聚合。

6.1.2 构造函数

构造函数	说明
MarkerClusterer(map:Map,opts:MarkerClustererOptions)	<p>创建一个 MarkerClusterer 组件。</p> <p>参数说明：</p> <p>map：地图对象</p> <p>opts：详见 MarkerClustererOptions 类</p>

6.1.3 方法

方法	返回值	说明
addMarker(marker:TMarker)	none	添加一个聚合的标记。
addMarkers(markers:Array<TMarker>)	none	添加要聚合的标记数组。
getClustersCount()	none	获取聚合的总数量。
clearMarkers()	number	从地图上彻底清除所有的标记。
getGridSize()	number	获取网格大小。
getMarkers()	array	获取所有的标记数组。

getMaxZoom()	number	获取聚合的最大缩放级别。
getMinClusterSize()	number	获取单个聚合的最小数量。
getStyles()	array	获取聚合的样式风格集合。
removeMarker(marker:Marker)	boolean	删除单个标记。
removeMarkers(markers:Array)	boolean	删除一组标记。
setGridSize(number)	none	设置网格大小。
setMaxZoom(maxZoom:Number)	none	设置聚合的最大缩放级别。
setStyles(styles:JSON)	none	设置聚合的样式风格集合。 参数说明： styles: 为 JSON 对象，具体内容请参考 MarkerClustererOptions 类的 styles 属性。

6.2 CoordinatePickup 类

用来实现在地图上用鼠标获取地理坐标的功能。

6.2.1 CoordinatePickupOptions 属性

属性	类型	默认值	说明
callback	function	null	设置回调函数。

6.2.2 构造函数

构造函数	说明
CoordinatePickup(map:Map,opts:CoordinatePickupOptions)	创建一个 CoordinatePickup 组件。 参数说明： map : 地图对象。 opts: 详见 CoordinatePickupOptions 类。

6.2.3 方法

方法	返回值	说明
addEvent ()	none	开启鼠标点击地图时获取地理坐标的事件。
removeEvent ()	none	关闭鼠标点击地图时获取地理坐标的事件。

7 右键菜单

7.1 MenuItem 类

此类表示一个菜单项。

7.1.1 构造函数

构造函数	说明
<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> 。 参数说明： <code>text</code> ：菜单项显示的文字。 <code>callback</code> ：点击菜单时的回调函数。 <code>opts</code> ：菜单项配置参数。

7.1.2 方法

方法	返回值	说明
<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。

7.2 ContextMenu 类

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。

7.2.1 ContextMenuOptions 属性

属性	类型	默认值	说明
<code>width</code>	number	120	指定此菜单项的宽度，菜单以最长的菜单项宽度为准。

7.2.2 构造函数

构造函数	说明
------	----

ContextMenu	创建一个右键菜单实例。
-------------	-------------

7.2.3 方法

方法	返回值	说明
addItem(item:MenuItem)	none	添加菜单项。
getItem(index:Number)	TMenuItem	返回指定索引位置的菜单项，第一个菜单项的索引为 0。
removeItem(item:MenuItem)	none	移除菜单项。
addSeparator()	none	添加分隔符。
removeSeparator(index:Number)	none	移除指定索引位置的分隔符，第一个分隔符的索引为 0。
getItems()	array	返回所有的 TMenuItem，是一个数组。
getAllSeparato()	array	返回所有分割线，是一个数组。
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。

7.2.4 事件

事件	参数	说明
open	{point,type,target}	右键菜单打开时触发事件， 参数说明： point 表示菜单开启时的像素坐标点。
close	{type,target}	右键菜单关闭时触发事件。

8 实体类

8.1 LngLat 类

此类以经度和纬度表示的地理位置坐标点。

8.1.1 LngLatOptions 属性

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

8.1.2 构造函数

构造函数	说明
LngLat (lng:Number,lat:Number)	根据给定经度和纬度创建地理位置坐标点。 参数说明： lng：地理经度。 lat：地理纬度。

8.1.3 方法

方法	返回值	说明
getLng()	number	获取地理坐标点的经度。
getLat()	number	获取地理坐标点的纬度。
distanceTo(other:LngLat)	number	计算当前地理坐标点与给定坐标点之间的距离。 参数说明： other：经纬度坐标。
equals(other:LngLat)	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。 参数说明： other：地理坐标。

8.2 LngLatBounds 类

此类可以指定的坐标范围建立一个矩形区域。

8.2.1 构造函数

构造函数	说明
LngLatBounds(sw:LngLat,ne:LngLat)	根据给定的多个地理位置坐标点创建矩形。 参数说明： sw：矩形区域的西南角。 ne：矩形区域的东北角。

8.2.2 方法

方法	返回值	说明
getSouthWest()	LngLat	返回西南点坐标。

getNorthEast()	LngLat	返回东北点坐标。
getCenter()	LngLat	返回边界的中心点。
extend(lnglat:LngLat)	this	放大此矩形，使其包含给定的点。 参数说明： lnglat: 点的地理坐标。
contains(other:LngLat LngLatBounds)	boolean	如果点的地理坐标位于此矩形内或者传入的矩形区域完全包含于此矩形区域中，则返回 true。 参数说明： other: 点的地理坐标或者地理范围坐标。
intersects(other:LngLatBounds)	boolean	如果矩形与给定边界相交则返回 true。 参数说明： other: 地理范围坐标。

8.3 Point 类

此类是以像素坐标表示的地图上的一个点。

8.3.1 PointOptions 属性

属性	类型	说明
x	number	像素坐标中 x 坐标，x 坐标向右增大。
y	number	像素坐标中 y 坐标，y 坐标向下增大。

8.3.2 构造函数

构造函数	说明
Point (x:Number, y:Number)	根据给定 x 和 y 坐标创建地图上的一个像素坐标点。

8.3.3 方法

方法	返回值	说明
equals(other:Point)	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。 参数说明： other: 点的像素坐标。
distanceTo(other:Point)	number	获得当前点与给定的距离。 参数说明： other: 点的像素坐标。

9 服务类

9.1 搜索

9.1.1 LocalSearch 类

此类用于位置检索、周边检索和范围检索。

9.1.1.1 LocalSearchOptions 属性

属性	类型	说明
pageCapacity	number	每页容量。
onSearchComplete	Function	检索结束后的回调函数。

9.1.1.2 构造函数

构造函数	说明
LocalSearch(map:Map ,opts:LocalSearchOptions)	创建一个搜索类实例。 参数说明： map：地图对象。 opts：详见 LocalSearchOptions 类。

9.1.1.3 方法

方法	返回值	说明
search(keyword:String,type:Number)	none	根据检索词发起检索。 参数说明： keyword：关键字。 type：搜索类型,1 表示普通搜索;2 表示视野内搜索;4 表示普通建议词搜索;5 表示公交规划建议词搜索;7 表示 纯 POI 搜索(不搜公交线) ;10 表示拉框搜索
searchInBounds(keyword:String, bounds:Bounds)	none	根据范围和检索词发起范围检索。 参数说明： keyword：关键字。 bounds：搜索范围
searchNearby(keyword:String,center:LngLat,radius:Number)	none	根据中心点、半径与检索词发起周边检索。 参数说明： keyword：关键字。 center：中心点。 Radius：搜索半径。
getResults()	LocalSearch	返回最近一次检索的结果。

	Result	
clearResults()	none	清除最近一次检索的结果。
gotoPage(page:Number)	none	检索特定页面的结果。 参数说明： page: 检索特定页面数值。
setSpecifyAdminCode(code:Number)	none	设置检索范围。 参数说明： code: 行政区的国标码。
setQueryType(type:Number)	none	设置搜索类型。 参数说明： type: 搜索类型,1 表示普通搜索;2 表示视野内搜索;4 表示普通建议词搜索;5 表示公交规划建议词搜索;7 表示 纯 POI 搜索(不搜公交线)。
getQueryType()	number	返回搜索类型。
setPageCapacity(count:Number)	none	设置每页容量。 参数说明： count: 每页容量。
getPageCapacity()	number	返回每页容量。
setSearchCompleteCallback(fun:Function)	none	设置检索结束后的回调函数。 参数说明： fun: 回调函数。回调函数参数为 LocalSearchResult, 具体内容详见 TLocalSearchResult 类。
firstPage()	none	检索第一页。
nextPage()	none	检索下一页。
previousPage()	number	检索上一页。
lastPage()	none	检索最后一页。
getCountNumber()	number	返回总记录数。
getCountPage()	none	返回共分总页数。
getPageIndex()	string	返回当前页。

9.1.2 LocalSearchResult 类

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

9.1.2.1 方法

方法	返回值	说明
getResultType()	number	返回搜索类型，1 表示普通搜索;2 表示视野内搜索;4 表示普通建议词搜索;5 表示公交规划建议词搜索;7 表示 纯 POI 搜索(不搜公交线)。
getCount()	number	返回检索总条数。

getKeyword()	string	返回检索关键词。
getPois()	JSON	<p>返回点信息，当 getResultType 值为 1 时返回。 数据结构如下：</p> <pre>[{ "phone": "", //电话 "lonlat": "116.313488 39.979416", //坐标 "address": "北京市海淀区", //地址 "name": "中关村", //Poi点名称 "poiType": "102", //poi类型（102 表示公交站，普通poi 该参数可以不返回） }, ...]</pre>
getStatistics()	none	<p>返回统计信息，当 getResultType 值为 2 时返回。 数据结构如下：</p> <pre>{ "priorityCitys": [//推荐显示城市 { "count": "38341", //统计数量 "name": "上海市", //城市名称 "adminCode": 156310000 //城市 国标码 }, { "count": "37346", "name": "北京市", "adminCode": 156110000 }, ...], "keyword": "饭店", //搜索的关键字 "countryCount": 1, //搜索的国家数量 "citysCount": 383, //搜索的城市数量 "allAdmins": [//各省包含信息集合 { "count": "82739", //统计数量 "name": "广东省", //省名称 "adminCode": 156440000, //省国 标码 "childAdmins": [//包括各市级 集合 }] }</pre>

		<pre> { "count": "19038", ///统 计数量 "name": "广州市", //城 市名称 "adminCode": "156440100" //城市国标码 }, { "count": "16578", "name": "深圳市", "adminCode": "156440300" }, ...] }, { "count": "62891", "name": "四川省", "adminCode": 156510000, "childAdmins": [{ "count": "30673", "name": "成都市", "adminCode": "156510100" }, { "count": "4907", "name": "绵阳市", "adminCode": "156510700" }, ...] }, {...}, ...], "provinceCount": 34 //搜索的省份数量 }</pre>
getArea()	none	返回行政区省信息,当 getResultType 值为 3 时返回。

		<p>数据结构如下：</p> <pre> { "level": "11", //显示级别 "lonlat": "116.38094,39.923615", //定位中心点坐标 "name": "北京市", //名称 "points": [//行政区边界坐标 { "region": "116.625875 41.058853,116.68501 41.040147,116.692716 41.019134,116.677056 40.998562,..." }], "type": "1" //1 表示正常区域，2 表示特殊区域 } </pre>
getSuggests()	none	<p>返回建议词信息，当 getResultType 值为 4 时返回。数据结构如下：</p> <pre> [{ "address": "", //地址 "name": "北京市", //名称 "gbCode": "000" //国标码 }, { "address": "北京市海淀区", "name": "北京北京", "gbCode": "156110108" }, ...] </pre>
getPrompt()	none	<p>返回提示信息。数据结构如下：</p> <pre> [{ "keyword": "123456", //关键字 "admins": [{ "name": "中国", //搜索的行政区范围 "adminCode": 156000000 //行政区码 }], },] </pre>

		<pre> "type": 2 /* 当 Type=1 时, 会给出一个 admin 一个 keyword 还有 admincode, 此时一般的提示为 “是否在 XXX 搜索名称含 XXX 的结果” 当 Type = 2 时, 会给出一个 admin 一个 keyword 还有 admincode, 此时一般提示为 “在 XXX 没有搜索到相关的结果” 当 Type =3 时, 会给出多个 admin 及对应的 admincode , 没有 keyword, 此时一般的提示为 “您是否要找: 广东省汕尾市城区 山西省晋城市城区 山西省大同市城区 山西省长治市城区 山西省阳泉市城区 ” 只列出名称即可, 点击后, 直接用这些关键字搜索即可完成行政区跳转 */ }, { "admins": [{ "name": "中国", "adminCode": 156000000 }], "type": 4 }] </pre>
getLineData()	number	<p>返回线路信息, 当 getResultType 值为 5 时返回。数据结构如下:</p> <pre> [{ "poiType": "103", // poi 点的 poi 类型 (poiType =102 表示公交站, 其它值表示普通 poi) "stationNum": "24", //站数 "name": "1 路(四惠枢纽站-靛厂新村)", //线路名称 "uuid": "23089" //线路的 id }, { </pre>

		<pre> "poiType": "103", "stationNum": "24", "name": "1 路(靛厂新村-四惠枢纽站)", "uuid": "23088" }, ...]</pre>
--	--	---

9.2 逆地理编码

9.2.1 Geocoder 类

此类用于获取用户的地址解析。

9.2.1.1 构造函数

构造函数	说明
Geocoder()	创建一个地址解析器的实例。

9.2.1.2 方法

方法	返回值	说明
getLocation(point:LngLat,callback:function)	none	对指定的坐标点进行反地址解析。如果解析成功，则回调函数的参数为 GeocoderResult 对象。

9.2.2 GeocoderResult 类

9.2.2.1 构造函数

构造函数	说明
GeocoderResult ()	创建一个地址解析器结果的实例。

9.2.2.2 方法

方法	返回值	说明
getStatus ()	numer	对指定的坐标点进行反地址解析。如果解析成功，则回调函数的参数为 LocationOptions 对象，否则回调函数的参数为 null。

getMsg()	string	返回响应信息。
getLocationPoint()	LngLat	获取此点坐标。
getAddress()	string	获取详细地址。
getAddressComponent()	AddressComponent	获取此点的详细信息。

9.2.3 AddressComponent 类

9.2.3.1 属性

属性	类型	说明
address	string	此点最近地点信息
address_distance	number	此点距离最近地点信息距离
address_position	string	此点在最近地点信息方向
city	string	此点所在国家或城市或区县
poi	string	距离此点最近 poi 点
poi_distance	number	距离此点最近 poi 点的距离
poi_position	string	此点在最近 poi 点的方向
road	string	距离此点最近的路
road_distance	number	此点距离此路的距离

9.3 公交

9.3.1 TransitRoute

用于获取公交线路规划方案。

9.3.1.1 TransitRouteOptions 属性

属性	类型	说明
policy	number	公交导航的策略参数。 策略常量。常量如下： TMAP_TRANSIT_POLICY_LEAST_TIME = 1 表示最少时间 TMAP_TRANSIT_POLICY_LEAST_TRANSFER = 2 表示最少换乘 TMAP_TRANSIT_POLICY_LEAST_WALKING = 4 表示最少步行 TMAP_TRANSIT_POLICY_AVOID_SUBWAYS = 8

		表示不乘地铁
onSearchComplete	Function	检索结束后的回调函数。

9.3.1.2 构造函数

构造函数	说明
TransitRoute(map:Map,opts:TransitRouteOptions)	创建一个公交导航实例。 参数说明： map: 地图对象。 opts: 详见 TransitRouteOptions 类。

9.3.1.3 方法

方法	返回值	说明
search(start:LngLat,end:LngLat)	none	发起检索。 参数说明： start: 起点坐标。 end: 终点坐标。
getResults()	TransitRouteResult	返回最近一次检索的结果。
clearResults()	none	清除最近一次检索的结果。
setPolicy(policy:Number)	none	设置路线规划策略，参数为策略常量。 参数说明： policy: 策略常量。常量如下： TMAP_TRANSIT_POLICY_LEAST_TIME = 1 表示最少时间 TMAP_TRANSIT_POLICY_LEAST_TRANSFER = 2 表示最少换乘 TMAP_TRANSIT_POLICY_LEAST_WALKING = 4 表示最少步行 TMAP_TRANSIT_POLICY_AVOID_SUBWAYS = 8 表示不乘地铁
setSearchCompleteCallback(fun:Function)	none	设置检索结束后的回调函数。 参数说明： fun: 回调函数。回调函数参数为 TransitRouteResult，具体内容详见 TransitRouteResult 类。
getStatus()	none	返回状态码。 状态码如下： 0 表示正常返回线路； 1 表示找不到起点；

		2 表示找不到终点; 3 表示规划不出线路; 4 表示起终点距离 200 米以内, 不返回线路; 5 表示起终点距离 500 米内, 返回线路; 6 表示输入参数错误;
--	--	--

9.3.2 TransitRouteResult

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

9.3.2.1 方法

方法	返回值	说明
getNumPlans()	number	返回方案个数。
getPlan(i:Number)	TransitRoutePlan	返回索引指定的方案。索引 0 表示第一条方案。

9.3.3 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。检索。

9.3.3.1 方法

方法	返回值	说明
getNumSegments()	number	返回单个方案的详细信息段数。
getDetails(i:Number)	TransitRouteLine	返回索引指定的详细信息。索引 0 表示第一条方案。
getLineName()	array	返回方案描述文本。
getDistance()	number	返回方案总距离。
getDuration()	number	返回方案总时间。
getLineType()	Number	返回公交导航的策略参数。

9.3.4 TransitRouteLine

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

9.3.4.1 方法

方法	返回值	说明
getSegmentType()	number	返回线路类型。 1 表示步行; 2 表示公交; 3 表示地铁; 4 表示地铁站内换乘。
getStationStart()	JSON	返回起站点信息。 数据结构如下: { "lonlat": "116.331796,39.991591", //起站点坐标 "name": "五道口站", //起站点名称 "uuid": "133010" //起站点 id }
getStationEnd()	JSON	返回起站点信息。 数据结构如下: { "lonlat": "116.331796,39.991591", //起站点坐标 "name": "五道口站", //起站点名称 "uuid": "133010" //起站点 id }
getSegmentLine()	number	返回线路内容。 数据结构如下: [{ "segmentStationCount": 3, //此段线路需要经过的 站点数 "segmentTime": 8, //此段线路需要的时间 "segmentDistance": 5933.84980962115, //此段线 路的距离 "direction": "地铁 1 3 号线", //此段线路的完整线 路名 "linePoint": "116.331796,39.991591;116.331993,39.987717;11 6.332141,39.98644;...", //此段线路的坐标 "lineName": "地铁 1 3 号线", //线路名称 "byuuid": "23200" //此段线路 id }]

9.3.5 BusLineSearch

公交线路搜索类。

9.3.5.1 BusLineSearchOptions 属性

属性	类型	说明
onGetBusListComplete	Function	设置公交列表查询后的回调函数。
onGetBusLineComplete	Function	设置公交线路查询后的回调函数。

9.3.5.2 构造函数

构造函数	说明
BusLineSearch(map:Map ,opts: BusLineSearchOptions)	公交线路搜索类。 参数说明： map: 地图对象。 opts: 详见 BusLineSearchOptions 类。

9.3.5.3 方法

方法	返回值	说明
getBusList(keyword: String)	none	检索公交信息。
getBusLine(busLstItem:Bus ListItem)	none	检索公交线路。
setGetBusListCompleteCall back(callback:Function)	none	设置公交列表查询后的回调函数。
setGetBusLineCompleteCall back(callback:Function)	none	设置公交线路查询后的回调函数。

9.3.6 BusListResult

此类表示 T.BusLineSearch 的检索结果，没有构造函数，通过 T.BusLineSearch 的 setGetBusListCompleteCallback 回调函数的参数得到。

9.3.6.1 方法

方法	返回值	说明
getNumBusList()	number	公交列表个数。
getBusListItem(i: Number)	JSON	获取某一个具体的公交列表中的对象。 数据结构如下： <pre>{ "stationNum": "42", //站数 "name": "850 路(东直门外-马坡站)", //线路名称 "uuid": "23084" //线路的 id }</pre>

9.3.7 BusLine

此类表示 T.BusLineSearch 的检索结果，没有构造函数，通过 T.BusLineSearch 的 setGetBusLineCompleteCallback 回调函数的参数得到。

9.3.7.1 BusLineOptions 属性

属性	类型	说明
lineName	string	线路名称，如：68 路。
lineType	number	线路类型，1 表示公交；2 表示地铁；3 表示磁悬浮。
length	number	线路的长度，单位米。
linePoint	string	线路的详细点信息，格式：x,y;x,y;
startTime	string	始发车时间格式为：hh:mm 24 小时制。
endTime	string	末班车时间格式为：hh:mm 24 小时制。
totalTime	number	公交线路的全程运营总时间，单位分钟。
stationCount	number	该线路所具有的站点总数量。
interval	number	发车间隔，单位秒。
ticketcal	number	计费模式，0 表示单一；1 表示按距离；2 表示按站。
totalPrice	number	全程票价，单位分。
startPrice	number	起步票价，单位分。
increasedPrice	number	递增距离票价，单位千米。
increasedStep	number	车站递增票价，按站。

ismonTicket	number	是否支持月票，0 表示不支持；1 表示支持。
isBidirectional	number	是否双向行驶，0 表示单项；1 表示双向。
isManual	number	是否人工售票，0 表示有人；1 表示无人。
status	number	状态，0 表示使用中；1 表示非使用中
company	string	所属公交公司，公交线路所属公司。

9.3.7.2 方法

方法	返回值	说明
getNumBusStations()	number	返回公交站点个数。
getBusStation(i: Number)	JSON	返回某一个具体的公交站信息。 数据结构如下： { "lonlat": "116.430522,39.939852", //站点坐标 "name": "东直门外", //站点名称 "uuid": "122805" //站的 id 信息 }
getPath()	array	返回公交线地理坐标点数组。

9.4 驾车

9.4.1 DrivingRoute

此类用于获取驾车路线规划方案。

9.4.1.1 DrivingRouteOptions 属性

属性	类型	说明
policy	number	驾车策略参数。 参数说明： policy: 策略常量。常量如下： TMAP_DRIVING_POLICY_LEAST_TIME = 0 最少时间 TMAP_DRIVING_POLICY_LEAST_DISTANCE = 1 最短距离 TMAP_DRIVING_POLICY_AVOID_HIGHWAYS = 2 避开高速 TMAP_DRIVING_POLICY_WALK = 3 步行

onSearchComplete	Function	检索完成后的回调函数。参数：results: DrivingRouteResult。
------------------	----------	--

9.4.1.2 构造函数

构造函数	说明
DrivingRoute(map:Map,opts:DrivingRouteOptions)	创建一个驾车导航实例。 参数说明： map: 地图对象。 opts: 详见 DrivingRouteOptions 类。

9.4.1.3 方法

方法	返回值	说明
search(start:LngLat,end:LngLat)	none	发起检索。 参数说明： start: 起点坐标。 end: 终点坐标。
getResults()	DrivingRouteResult	返回最近一次检索的结果。
clearResults()	none	清除最近一次检索的结果。
setPolicy(policy:Number)	none	设置路线规划策略，参数为策略常量。 参数说明： policy: 策略常量。常量如下： TMAP_DRIVING_POLICY_LEAST_TIME = 0 最少时间 TMAP_DRIVING_POLICY_LEAST_DISTANCE = 1 最短距离 TMAP_DRIVING_POLICY_AVOID_HIGHWAYS = 2 避开高速 TMAP_DRIVING_POLICY_WALK = 3 步行
setSearchCompleteCallback(fun:Function)	none	设置检索结束后的回调函数。 参数说明： fun : 回调函数。回调函数参数为 DrivingRouteResult，具体内容详见 DrivingRouteResult 类。

9.4.2 DrivingRouteResult

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

9.4.2.1 方法

方法	返回值	说明
getStart()	LngLat	返回起点。
getEnd()	LngLat	返回终点。
getNumPlans()	number	返回方案个数。
getPlan(i:Number)	TransitRoutePlan	返回索引指定的方案。索引 0 表示第一条方案。

9.4.3 RoutePlan

此类表示一条驾车出行方案。没有构造函数，通过 DrivingRouteResult.getPlan()方法获得。

9.4.3.1 方法

方法	返回值	说明
getNumRoutes()	number	返回该方案包含的线路的个数。
getRoute(i:Number)	Route	返回方案中索引指定的线路信息。索引 0 表示第一条线路。
getDistance()	number	返回方案总距离，单位为公里。
getDuration()	number	返回方案总时间，单位为秒。
getWayPois()	Array<LngLat>	返回途径点。
getPath()	Array<LngLat>	返回路线的经纬度。

9.4.4 Route

此类表示一条驾车路线。没有构造函数，通过 RoutePlan.getRoute()方法获得。

9.4.4.1 方法

方法	返回值	说明
getNumSteps()	number	返回该段路线包含的关键点个数。

getStep(i:Number)	Step	返回索引指定的关键点。索引 0 表示第一个关键点。
getDistance()	number	返回该段路线距离。
getIndex()	number	返回该段路线在方案中的索引位置。
getPath()	Array<Lng Lat>	返回该段路线的坐标。

9.4.5 Step

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

9.4.5.1 方法

方法	返回值	说明
getPosition()	lngLat	返回关键点地理坐标。
getIndex()	Number	返回本关键点在路线中的位置索引
getDescription()	string	返回关键点描述文本。

