

JavaScript Map API 用户手册

编制: 北京易伟航科技有限公司

修订日期:2016 03 30

版本:1.4

版本记录

版本	更新说明	日期	修订人
1.1	文档第一版	20140610	康佩锋
1.2	1) 增加 EV.Size 类 2) 增加 EV.ContextMenu 和 EV.Menuitem 支持右键菜单操作	20140801	康佩锋
1.3	修改服务器地址	20141029	康佩锋
1.4	修改服务器地址 工具栏增加行政区划栏创建方法	20140116	康佩锋

概要

本文档是地图 API 的用户使用参考手册，详细阐述了 API 提供用户的类和方法，以及可以实现的功能介绍。

地图 API 让您可以使用 JavaScript 将地图嵌入您自己的网页中，也同时支持移动设备的浏览器应用。API 提供了大量实用且易用的方法与地图进行交互，拥有一系列向地图添加内容的服务，更极大程度地从移动设备用户的需求和体验出发，实现功能丰富、交互性强的地图应用程序的构建，从而使您可以创建稳定且功能全面的地图应用程序。

本文档分为基础类、地图类、控件类、叠加物、工具类、服务类、实用类等章节。

每个类基本按照：描述、属性、构造函数、方法、示例 分小章节进行详细介绍。

目录

JavaScript Map API 用户手册	1
概述.....	11
一 基础类.....	12
1 EV.Event 类.....	12
1.1 描述.....	12
1.2 构造函数.....	12
1.3 属性.....	12
1.4 方法.....	12
1.5 示例.....	13
2 EV.LngLat 类	13
2.1 描述.....	13
2.2 构造函数.....	13
2.3 属性.....	13
2.4 方法.....	13
2.5 示例.....	14
3 EV.Point 类	14
3.1 描述.....	14
3.2 构造函数.....	14
3.3 属性.....	14
3.4 方法.....	14
3.5 示例.....	14
4 EV.LngLatBounds 类	14
4.1 描述.....	14
4.2 构造函数.....	15
4.3 属性.....	15
4.4 方法.....	15
4.5 示例.....	15

5 EV.Size 类.....	15
5.1 描述.....	15
5.2 构造函数.....	16
5.3 属性.....	16
5.4 方法.....	16
5.5 示例.....	16
二 地图类.....	16
1 EV.Map	16
1.1 描述.....	16
1.2 构造函数.....	17
1.3 属性.....	17
1.4 方法.....	17
1.5 示例.....	19
2 EV.InfoWindow	19
2.1 描述.....	19
2.2 构造函数.....	19
2.3 属性.....	20
2.4 方法.....	20
2.5 示例.....	20
三 控件类.....	21
1 EV.CopyrightControl	21
1.1 描述.....	21
1.2 构造函数.....	21
1.3 属性.....	21
1.4 方法.....	21
1.5 示例.....	22
2 EV.MapControl	22
2.1 描述.....	22
2.2 构造函数.....	22

2.3 属性.....	22
2.4 方法.....	23
2.5 示例.....	23
3 EV.MapTypeControl.....	23
3.1 描述.....	23
3.2 构造函数.....	23
3.3 属性.....	23
3.4 方法.....	23
3.5 示例.....	23
4 EV.OverviewMapControl	24
4.1 描述.....	24
4.2 构造函数.....	24
4.3 属性.....	24
4.4 方法.....	24
4.5 示例.....	24
5 EV.ScaleControl.....	25
5.1 描述.....	25
5.2 构造函数.....	25
5.3 属性.....	25
5.4 方法.....	25
5.5 示例.....	25
四 叠加物.....	25
1 EV.Overlay.....	25
1.1 描述.....	25
1.2 构造函数.....	26
1.3 属性.....	26
1.4 方法.....	26
1.5 示例.....	29
2 EV.Circle.....	29

2.1 描述.....	29
2.2 构造函数.....	29
2.3 属性.....	30
2.4 方法.....	30
2.5 示例.....	30
3 EV.Ellipse.....	31
3.1 描述.....	31
3.2 构造函数.....	31
3.3 属性.....	31
3.4 方法.....	31
3.5 示例.....	32
4 EV.Marker	32
4.1 描述.....	32
4.2 构造函数.....	33
4.3 属性.....	33
4.4 方法.....	33
4.5 示例.....	33
5 EV.PolyLine	34
5.1 描述.....	34
5.2 构造函数.....	34
5.3 属性.....	35
5.4 方法.....	35
5.5 示例.....	35
6 EV.Polygon	36
6.1 描述.....	36
6.2 构造函数.....	36
6.3 属性.....	36
6.4 方法.....	36
6.5 示例.....	36

7 EV.Rect	37
7.1 描述	37
7.2 构造函数	37
7.3 属性	37
7.4 方法	37
7.5 示例	38
8 对象编辑事件	38
8.1 描述	38
8.2 示例	39
五 菜单类	40
1 EV.ContextMenu 类	40
1.1 描述	40
1.2 构造函数	40
1.3 方法	40
1.4 示例	41
2 EV.MenuItem 类	42
2.1 描述	42
2.2 构造函数	42
2.3 属性	42
2.4 示例	43
六 工具类	43
1 EV.Toolbar	43
1.1 描述	43
1.2 构造函数	43
1.3 属性	43
1.4 方法	43
1.5 示例	44
2 EV.Tool 类	46
2.1 描述	46

2.2 方法.....	46
2.3 示例.....	46
3 漫游工具.....	47
3.1 描述.....	47
3.2 示例.....	47
4 全图工具.....	47
4.1 描述.....	47
4.2 示例.....	47
5 拉框放大工具.....	48
5.1 描述.....	48
5.2 示例.....	48
6 拉框缩小工具.....	48
6.1 描述.....	48
6.2 示例.....	49
7 标点工具.....	49
7.1 描述.....	49
7.2 示例.....	49
8 折线工具.....	50
8.1 描述.....	50
8.2 示例.....	50
9 多边形工具.....	51
9.1 描述.....	51
9.2 示例.....	51
10 矩形工具.....	52
10.1 描述.....	52
10.2 示例.....	52
11 椭圆工具.....	53
11.1 描述.....	53
11.2 示例.....	53

12	圆形工具.....	53
12.1	描述.....	53
12.2	示例.....	54
13	测量长度工具.....	54
13.1	描述.....	54
13.2	示例.....	54
14	测量面积工具.....	55
14.1	描述.....	55
14.2	示例.....	56
15	编辑工具.....	56
15.1	描述.....	56
15.2	示例.....	56
七	服务类.....	56
八	实用类.....	57
1	EV.MarkerClusterer 类	57
1.1	描述.....	57
1.2	构造函数.....	57
1.3	方法.....	57
1.4	示例.....	58

概述

当前服务器有三台，地址分别为

`http://a.map.icctic.cn:81`

`http://b.map.icctic.cn:81`

`http://c.map.icctic.cn:81`

客户可以任选一个使用。

要使用本 API 必须在 html 的 head 中添加引用，格式如下：

```
<script type="text/javascript" language="javascript" src="
http://a.map.icctic.cn:81/SE_JSAPI?uid=您的 UID"></script>
```

请使用发送给你的 UID（或者用户名）替换红色部分

有三个在线的示例页面，可以参考调用方法

SE_SH 等其他服务接口的 示例页面：

`http://map.icctic.cn/api/api.jsp`

JAVAScript 地图控件的示例页面 1：

`http://map.icctic.cn/api/mapapi.jsp`

JAVAScript 地图控件的示例页面 2：

`http://map.icctic.cn/api/demo.jsp`

一 基础类

1 EV.Event 类

1.1描述

用于注册事件处理程序和触发自定义事件，它的所有方法都是静态方法，不需要构造 EV.Event 对象。

1.2构造函数

无

1.3属性

无

1.4方法

名称	返回值	说明
EV.Event.addListener(source:Object, event:String, handler:Function)	EV.EventListener	为源对象（source）的自定义事件（event）注册事件处理程序（handler），返回一个可用于最终注销处理程序的句柄。事件处理程序的 this 指向源对象(source)
EV.Event.removeListener(handle:EV.EventListener)	None	删除使用 addListener() 注册的事件处理程序。
SE.Event.clearListeners(source:Object, event:String)	None	删除使用 addListener() 在指定对象（source）上为指定事件（event）注册的所有事件处理程序。
EV.Event.trigger(source:Object, event:String, args?:Array)	None	在源对象（source）上触发自定义事件（event），可选参数作为自定义事件调用函数的参数传递。
EV.Event.bind(EV.EventListener	将指定对象（object）的方法

source:Object, event:String, object:Object, method:Function)		(method) 调用注册为源对象 (source) 自定义事件 (event) 的 事件处理程序，返回一个可用于最 终注销处理程序的句柄。
---	--	---

1.5 示例

2 EV.LngLat 类

2.1 描述

经纬度坐标点类

2.2 构造函数

EV.LngLat(lon:Number,lat:Number) 使用经纬度值构造一个坐标点

2.3 属性

2.4 方法

名称	返回值	说明
EV.LngLat.WM(wlon:Number, wlat:Number)	EV.LngLat	此方法是静态方法，返回一个经纬 度坐标对象，wlon, wlat 代表了 web 魔卡托坐标
getLon()	Number	返回坐标点的经度值
getLat()	Number	返回坐标点的纬度值
distanceFrom(other:EV.LngLa t)	Number	返回距离另一个经纬度点 other 的 长度，单位米。

2.5 示例

```
var p = new EV.LngLat (119.212,36.023);
```

3 EV.Point 类

3.1 描述

地图的 DIV 元素的坐标点

3.2 构造函数

EV.Point(x:Number,y:Number) 构造一个坐标点

3.3 属性

x : Number 水平坐标

y: Number 垂直坐标

3.4 方法

3.5 示例

```
var p = new EV.Point(100,100);
```

4 EV.LngLatBounds 类

4.1 描述

表示一个经纬度范围对象

4.2构造函数

- `EV.LngLatBounds(Inglats: Array<EV.LngLat>)` 根据一组坐标点，构造一个矩形范围。

4.3属性

4.4方法

名称	返回值	说明
<code>containsLonLat(Inglat:EV.LngLat)</code>	Boolean	此范围是否包含某个点
<code>containsBounds(bounds:EV.LngLatBounds)</code>	Boolean	此范围是否包含另一范围
<code>getWS()</code>	EV.LngLat	返回此范围西南角点
<code>getWN()</code>	EV.LngLat	返回此范围西北角点
<code>getES()</code>	EV.LngLat	返回此范围东南角点
<code>getEN()</code>	EV.LngLat	返回此范围东北角点

4.5示例

```
var p1 = new EV.LngLat(119.01,30.01);  
var p2 = new EV.LngLat(119.02,30.02);  
var Inglats= [p1,p2];  
var bounds = new EV.LngLatBounds(Inglats);
```

5 EV.Size 类

5.1描述

表示一个控件的像素大小

5.2构造函数

EV.Size(w: Number, h: Number)

5.3属性

w: 宽度

h: 高度

5.4方法

名称	返回值	说明
<code>equals(sz:EV.Size)</code>	Boolean	判断两个是否一样大

5.5示例

```
var infow = new EV.InfoWindow("w01", new EV.LngLat(110,39), "标题", "这里是信息框内容", new EV.Size(300,300), function() {
    this.destroy();
    infow = null;
});
map.addPopup(infow);
```

二 地图类

1 EV.Map

1.1描述

地图控件对象，在一个 html div 元素上创建一个地图对象。

1.2构造函数

`EV.Map(div:String)` `div` 代表了 html 中一个 `div` 元素的 ID

1.3属性

`toolbar: EV.Toolbar` 详细请阅读 `EV.Toolbar`

1.4方法

名称	返回值	说明
<code>setMapType(maptype:Number)</code>	None	设置当前的地图类型 EV.Map.NORMAL_MAP 普通地图 EV.Map.SATELLITE_MAP 影像地图 EV.Map.HYBRID_MAP 混合地图
<code>zoomToChina()</code>	None	定位到中国范围
<code>setZoom(zoom:Number)</code>	None	设置当前地图缩放级别
<code>setLonLatZoom(lon:Number, lat:Number, zoom?:Number)</code>	None	设置地图的中心点和缩放级别
<code>setCenter(lnglat:EV.LngLat, zoom?:Number)</code>	None	设置地图中心点和缩放级别
<code>getCenterPoint()</code>	EV.LngLat	返回地图中心坐标点
<code>getLngLatBounds()</code>	EV.LngLatBounds	返回地图显示的经纬度范围
<code>getZoom()</code>	Number	返回地图当前的缩放级别
<code>getBestMap(lnglats:Array<EV.LngLat>)</code>	None	使当前地图定位到刚好能够显示 array 中所有 Ev.lnglat;
<code>fromContainerPixelToLngLat(pint:EV.Point)</code>	EV.LngLat	把地图 Div 的像素坐标转为经纬度坐标点
<code>fromLngLatToContainerPixel(lnglat:EV.LngLat)</code>	EV.Point	把经纬度坐标点转为地图的像素坐标点
<code>panTo(lnglat:EV.LngLat, zoom?:Number)</code>	None	类似 <code>setCenter</code> , 内部判定是否执行动画平移操作
<code>handleKeyboard()</code>	None	使地图可以用键盘平移
<code>disableHandleKeyboard()</code>	None	禁用键盘操作
<code>panDirection(dx:number, dy:number)</code>	None	动画平移地图, <code>dx</code> 和 <code>dy</code> 是地图窗口的倍率, 比如 -1, -1 表示地图向左向下平移一个窗口的距离
<code>zoomIn</code>	None	地图放大一级

技术支持：北京易伟航科技有限公司

zoomOut	None	地图缩小一级
addOverlay(overlay:EV.Overlay)	None	增加地图叠加层
removeOverlay(overlay:EV.Overlay)	None	删除地图叠加层
clearOverlays()	None	清空所有叠加层
addEventListener(event:String, handler:Function)	None	添加一个地图事件的回调响应 支持的事件: "mouseover", "mousemove", "mouseout", "move", "movestart", "moveend", "zoomend", "updatesize"
removeEventListener(event:String, handler:Function)	None	删除一个地图事件的回调响应
getEventPos(e:MouseEvent)	EV. Point	从一个鼠标事件中返回当前的鼠标位置
getEventBtn(e:MouseEvent)	Number 1 左键 2 右键	判断一个鼠标事件的左右键
addPopup(popup:EV. InfoWindow)	None	地图上增加一个信息框
removePopup(popup:EV. InfoWindow)	None	删除一个地图信息框
t_addcircle()	None	启用增加圆的工具
t_addellipse()	None	启用增加椭圆的工具
t_addline()	None	启用绘制折线的工具
t_addpoint()	None	启用标注点的工具
t_addpoly()	None	启用绘制多边形的工具
t_addrect()	None	启用绘制矩形的工具
t_edit()	None	启用编辑工具
t_global()	None	执行全图操作
t_marea()	None	启用测量面积的工具
T_mlength()	None	启用测量距离的工具
t_nav()	None	启用鼠标漫游工具, 这是默认工具
t_select()	None	启用对象选择工具
t_zoomin()	None	启用拉框放大工具
t_zoomout()	None	启用拉框缩小工具
t_print()	None	执行打印功能
t_screenshot()	None	启用截图工具
addContextMenu(menu:EV. ContextMenu)	None	增加一个右键菜单
removeContextMenu(menu:EV. C	None	移除一个右键菜单

ontextMenu)		
-------------	--	--

1.5 示例

```
//新建一个地图 它创建在 id = "mapdiv" 的 div元素上
map = new EV.Map("mapdiv");
map.addEventListener("mousemove", function(e) {
    //获取鼠标位置
    var point = map.getEventPos(e);
    //获取地理坐标
    var lonlat = map.fromContainerPixelToLngLat(point);
    //结果输出到一个 div 元素上显示
    var element = document.getElementById('mousepos');
    var out = "经度: " + lonlat.getLon().toFixed(5) + "  纬度: " +
lonlat.getLat().toFixed(5);
    element.innerHTML = out;
});
```

2 EV.InfoWindow

2.1 描述

地图窗口上添加的信息框

2.2 构造函数

EV.InfoWindow(id:String, lnglat:EV.LngLat, title:HtmlString, content: HtmlString, size:EV.Size, closecallback:Function)

lnglat: 信息框所指的地理范围

title: 信息框的标题支持 html 字符串

content:信息框的内容 支持 html 字符串

size: 信息框大小, 此值如果为 null, 那么会自动根据内容计算大小

closecallback: 关闭之后的回调函数

2.3 属性

2.4 方法

名称	返回值	说明
<code>updatePosition()</code>	None	强制刷新位置
<code>setLngLat(lnglat:EV.LngLat)</code>	None	设置信息框的指向位置
<code>visible()</code>	Boolean	信息框是否可见
<code>setVisible(v:Boolean, anim:Boolean)</code>	None	设置信息框是否可见 v: true 可见 false 不可见 anim: 是否动画显示 true 表示动画显示，不传或者其他值那么立即显示。
<code>toggle()</code>	None	切换信息框的可见性
<code>show()</code>	None	显示
<code>hide()</code>	None	隐藏
<code>setTitle(title:String)</code>	None	设置标题
<code>setContent(content:String)</code>	None	设置内容
<code>destroy()</code>	None	删除这个信息框

2.5 示例

//下面这段代码创建一个一直跟随鼠标位置的信息框

```
var infow = null;
map.addListener("mousemove", function(e) {
    //获取鼠标位置
    var point = map.getEventPos(e);
    //获取地理坐标
    var lonlat = map.fromContainerPixelToLngLat(point);
    //如果信息框不存在，那么新建，否则更改它的位置
    if(!infow) {
        infow = new EV.InfoWindow("w01", lonlat, "标题", "这里是信息框内容",
```

```

null, function() {
    this.destroy();
    infow = null;
});
map.addPopup(infow);
}
else
    infow.setLngLat(lonlat);
});

```

三 控件类

1 EV.CopyrihtControl

1.1描述

此控件一般用来在地图上显示一行文字或者其他静态内容



1.2构造函数

EV.CopyrihtControl (left?: Number, right?: Number, top?: Number, bottom?: Number, width?: Number, height?: Number, innerhtml: String)

left, right, top, bottom, width, height 这些是控件的定位参数，不需要全部设置，根据需要可以设置其中的几项

innerhtml: 这是控件的显示内容

1.3属性

1.4方法

名称	返回值	说明
----	-----	----

setContent (content:String)	None	更改控件内容
--------------------------------	------	--------

1.5 示例

```
var cp = new EV.CopyrightControl(105,null,null,2,null,null,"<span style='font:11px arial,simsun'> © <a style='color:#0781d3;' href='http://www.evalab.com' target='_blank'>www.evalab.com</a></span>");
map.addControl(cp);
```

2 EV.MapControl

2.1 描述



此控件构造一个漫游控件，支持各种地图漫游操作。

2.2 构造函数

```
EV.MapControl()
```

2.3 属性

无

2.4 方法

无

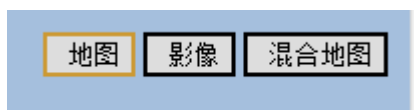
2.5 示例

```
map.addControl(new EV.MapControl());
```

3 EV.MapTypeControl

3.1 描述

此控件构造一个图层切换控件；



3.2 构造函数

```
EV.MapTypeControl()
```

3.3 属性

无

3.4 方法

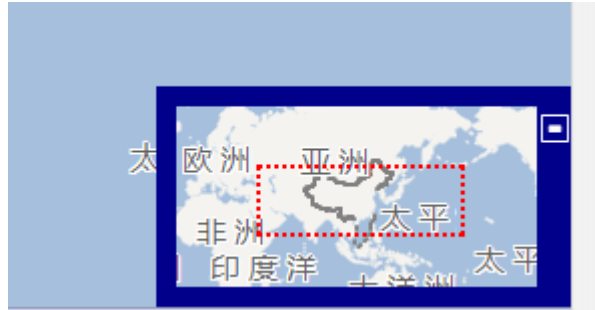
无

3.5 示例

```
map.addControl(new EV.MapTypeControl());
```

4 EV.OverviewMapControl

4.1描述



此控件构造一个鹰眼预览图控件；

4.2构造函数

EV. OverviewMapControl()

4.3属性

无

4.4方法

无

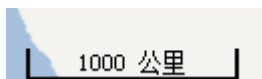
4.5示例

```
map.addControl(new EV. OverviewMapControl ());
```

5 EV.ScaleControl

5.1描述

此控件构造一个比例尺控件：



5.2构造函数

EV.ScaleControl()

5.3属性

无

5.4方法

无

5.5示例

```
map.addControl(new EV. ScaleControl ());
```

四 叠加物

1 EV.Overlay

1.1描述

地图上动态添加的各种元素，标注，折线，多边形，圆，矩形，椭圆等对象都派生这个

类。

1.2 构造函数

无，此类不直接构造，其他派生子类构造。

1.3 属性

`name: String` 覆盖物名称，它是该覆盖物弹出信息框的默认标题

`popupContent:String` 信息框内容

1.4 方法

名称	返回值	说明
<code>remove()</code>	None	移除这个覆盖物
<code>hide()</code>	None	隐藏
<code>show()</code>	None	显示
<code>isHidden()</code>	None	是否被隐藏
<code>refresh()</code>	None	强制刷新线的显示
<code>openInfoWinHtml(info:String)</code>	None	打开绑定到这个标注上的信息框。 info 里填的信息框信息。
<code>showPopup()</code>	None	显示一个默认的信息框，信息框标题使用 name 属性，内容使用 popupContent 属性
<code>setPopupContent()</code>		
<code>closeInfoWindow()</code>	None	关闭这个标注的信息框
<code>getBounds()</code>	EV. LngLatBounds	获取这个覆盖物的范围
<code>addEventListener(event:string, handler:Function)</code>	None	添加一个事件回调，支持的事件："click", "dblclick", "mouseout", "mouseover"
<code>removeEventListener(event:string, handler:Function)</code>	None	删除事件回调
<code>getCenterPoint()</code>	EV. LngLat	获取这个覆盖物的中心经纬度点
<code>setExternalGraphic(href:String)</code>	None	设置样式 externalGraphic, 该样式对应了标注的图标路径
<code>setFill(f:Boolean)</code>	None	是否填充（对多边形，圆，椭圆，矩形等面状元素有效）
<code>setFillColor(color:String)</code>	None	设置填充颜色，颜色值符合 css 的

		颜色标准
setFillOpacity(opacity: Number)	None	设置填充透明度 0~1
setFontColor(color:String)	None	设置文字颜色
setFontFamily(font:String)	None	设置文字字体
setFontOpacity(opacity: Number)	None	设置字体透明度
setFontSize(size: Number)	None	对应 CSS font-size
setFontWeight(weight:String)	None	对应 CSS font-weight
setFontStyle(style:String)	None	对应 CSS font-style
setGraphic(graphic:Boolean)	None	改元素是否有图形表示
setGraphicHeight(height: Number)	None	图标的像素高度
setGraphicName(name:String)	None	设置内置的图形样式, 有 "circle", "square", "star", "x", "cross", "triangle"。如果设置了 externalGraphic 此属性无效
setGraphicOpacity(opacity: Number)	None	设置图形透明度
setGraphicWidth(width: Number)	None	图标的像素高度
setGraphicXOffset(x: Number)	None	图形在 x 方向的像素偏移
setGraphicYOffset(y: Number)	None	图形在 y 方向的像素偏移
setGraphicZIndex(z: Number)	None	图形的 Z 值, Z 大者覆盖 z 小者
setLabel(label:String)	None	设置文字标注内容
setLabelAlign(align:String)	None	设置文字对齐方式 横向: "l"=left, "c"=center, "r"=right 垂直: "t"=top, "m"=middle, "b"=bottom 比如 "lt" 代表左上对齐
setLabelOutlineColor(color:String)	None	设置文字外边界颜色
setLabelOutlineOpacity(opacity: Number)	None	设置文字外边界透明度
setLabelOutlineWidth(width: Number)	None	设置外边界像素宽度
setLabelXOffset(x: Number)	None	设置文字的 x 方向的像素偏移
setLabelYOffset(y: Number)	None	设置文字的 y 方向的像素偏移
setLine(line:boolean)	None	对于面状对象 (多边形、圆) 是否绘制边线

setLineColor(color:String)	None	线颜色
setLineDashstyle(style:String)	None	线型, 可选: dot dash <u>dashdot</u> <u>longdash</u> <u>longdashdot</u> solid
setLineLinecap(style:String)	None	线连接样式可选: butt round square
setLineOpacity(opacity:Number)	None	线的透明度
setLineWidth(width:Number)	None	线宽
setPointRadius(radius:Number)	None	标注点的图形大小
setRotation(rotation:Number)	None	标注的旋转角度 (IE7 及以下版本不支持)
setStroke(stroke:boolean)	None	等同 setLine
setStrokeColor	None	等同 setLineColor
setStrokeDashstyle	None	等同 setLineDashstyle
setStrokeLinecap	None	等同 setLineLinecap
setStrokeOpacity	None	等同 setLineOpacity
setStrokeWidth	None	等同 setLineWidth
getExternalGraphic()	String	参看 set 方法
getFill()	Boolean	参看 set 方法
getFillColor()	String	参看 set 方法
getFillOpacity()	Number	参看 set 方法
getFontColor()	String	参看 set 方法
getFontFamily()	String	参看 set 方法
getFontOpacity()	Number	参看 set 方法
getFontSize()	Number	参看 set 方法
getFontWeight()	String	参看 set 方法
getFontStyle()	String	参看 set 方法
getGraphic()	String	参看 set 方法
getGraphicHeight()	Number	参看 set 方法
getGraphicName()	String	参看 set 方法
getGraphicOpacity()	Number	参看 set 方法
getGraphicWidth()	Number	参看 set 方法
getGraphicXOffset()	Number	参看 set 方法
getGraphicYOffset()	Number	参看 set 方法
getGraphicZIndex()	Number	参看 set 方法
getLabel()	String	参看 set 方法
getLabelAlign()	String	参看 set 方法
getLabelOutlineColor()	String	参看 set 方法
getLabelOutlineOpacity()	Number	参看 set 方法
getLabelOutlineWidth()	Number	参看 set 方法

getLabelXOffset()	Number	参看 set 方法
getLabelYOffset()	Number	参看 set 方法
getLine()	boolean	参看 set 方法
getLineColor()	String	参看 set 方法
getLineDashstyle()	String	参看 set 方法
getLineLinecap()	String	参看 set 方法
getLineOpacity()	Number	参看 set 方法
getLineWidth()	Number	参看 set 方法
getPointRadius()	Number	参看 set 方法
getRotation()	Number	参看 set 方法
getStroke()	boolean	参看 set 方法
getStrokeColor()	String	参看 set 方法
getStrokeDashstyle()	String	参看 set 方法
getStrokeLinecap()	String	参看 set 方法
getStrokeOpacity()	Number	参看 set 方法
getStrokeWidth()	Number	参看 set 方法

1.5 示例

2 EV.Circle

2.1 描述

派生自 EV.Overlay ， 描述了一个圆形覆盖物

2.2 构造函数

EV.Circle(center:EV.LngLat,radius:Number,style:StyleOption)

center: 圆心经纬度

radius: 圆半径 单位米

style: 这个是一个配置对象，直接使用 javascript 的对象配置

可以配置的样式，参看 EV.Overlay 的 set 和 get 方法

```
var style = {
```

```
strokeColor:"red"
```

```
strokeWidth:4
```

```
};
```

2.3 属性

2.4 方法

名称	返回值	说明
<code>getType()</code>	<code>EV.OVERLAY_CIRCLE</code>	返回覆盖物的类型，以下是定义的几个类型常量，其他类型返回类似 <code>EV.OVERLAY_POINT</code> <code>EV.OVERLAY_MARKER</code> <code>EV.OVERLAY_INFOWIN</code> <code>EV.OVERLAY_POLYLINE</code> <code>EV.OVERLAY_POLYGON</code> <code>EV.OVERLAY_RECT</code> <code>EV.OVERLAY_ELLIPSE</code> <code>EV.OVERLAY_CIRCLE</code>
<code>getArea()</code>	Number	返回圆面积，单位平方米
<code>getCenter()</code>	<code>EV.LngLat</code>	获取圆的中心经纬度点
<code>setCenter(center:EV.LngLat)</code>	None	设置圆的中心点
<code>getRadius()</code>	Number	获取圆半径 单位米
<code>setRadius(r:Number)</code>	None	设置圆半径

2.5 示例

```
var circle = new EV.Circle(new EV.LngLat(119,30), 10000, {
    strokeColor:"red",
    fillColor:"yellow",
    fillOpacity:0.5
});
circle.addEventListener("click", function(e) {
    alert("点击了圆");
});
map.addOverlay(circle);
```

在地图上的效果



3 EV.Ellipse

3.1描述

派生自 EV.Overlay ， 描述了一个椭圆形覆盖物

3.2构造函数

EV.Ellipse(bounds:EV.LngLatBounds, style)

bounds: 覆盖物的地理范围

style: 这个是一个配置对象，直接使用 javascript 的对象配置

可以配置的样式，参看 EV.Overlay 的 set 和 get 方法

3.3属性

3.4方法

名称	返回值	说明
getType()	EV. OVERLAY_ELLIPSE	返回覆盖物的类型

3.5 示例

```
var p1 = new EV.LngLat(119,30);
var p2 = new EV.LngLat(120,31);
var bounds = new EV.LngLatBounds([p1,p2]);

var ellipse = new EV.Ellipse(bounds, {
    strokeColor:"red",
    fillColor:"yellow",
    fillOpacity:0.5
});

ellipse.name = "椭圆01";
ellipse.setPopupContent("点击了椭圆");
ellipse.addEventListener("click", function(e) {

    this.showPopup();
});

map.addOverlay(ellipse);
```

运行效果（点击椭圆后弹出信息框）如图：



4 EV.Marker

4.1 描述

派生自 EV.Overlay，描述了一个点状覆盖物。

4.2构造函数

`EV.Marker(lonlat:EV.LngLat, style)`

`lonlat`：点装覆盖物的点位置

`style`: 这个是一个配置对象，直接使用 javascript 的对象配置

可以配置的样式，参看 `EV.Overlay` 的 `set` 和 `get` 方法

4.3属性

4.4方法

名称	返回值	说明
<code>getType()</code>	<code>EV. OVERLAY_MARKER</code>	返回覆盖物的类型
<code>setLngLat(lnglat:EV.LngLat)</code>	<code>None</code>	设置点位置
<code>getLngLat()</code>	<code>EV.LngLat</code>	获取标注的点位置
<code>setIcon(icon:String)</code>	<code>None</code>	设置标注的图标路径
<code>getIcon()</code>	<code>String</code>	获得标注的图标路径
<code>distanceFrom(other:EV.Marker)</code>	<code>Number</code>	计算和另一个位置点的距离
<code>directionFrom(other:EV.Maker)</code>	<code>Number</code>	计算和另一位置点的偏差角度

4.5示例

//默认标注

```
var m1 = new EV.Marker(new EV.LngLat(119,30));
```

//更换了图标

```
var m2 = new EV.Marker(new EV.LngLat(120,31), {
    externalGraphic:
"http://a.map.iccttic.cn:81/img/marker-blue.png"
});
```

//添加了文字

```
var m3 = new EV.Marker(new EV.LngLat(119,31), {
    label:"带有文字",
    fontColor : "#0000ff",
    fontSize : "12px",
```

技术支持：北京易伟航科技有限公司

```
    fontFamily : "Courier New, monospace",  
    fontWeight : "bold",  
    labelYOffset : -15,  
    labelOutlineColor : "white",  
    labelOutlineWidth : 2  
  });  
map.addOverlay(m1);  
map.addOverlay(m2);  
map.addOverlay(m3);
```

运行效果如下：添加了 3 个标注



5 EV.PolyLine

5.1 描述

派生自 EV.Overlay，描述了一个折线状覆盖物。

5.2 构造函数

EV.PolyLine(Inglats:Array<EV.LngLat>, style) 有若干经纬度点构造一条折线。

style: 这个是一个配置对象，直接使用 javascript 的对象配置

可以配置的样式，参看 EV.Overlay 的 set 和 get 方法

5.3属性

5.4方法

名称	返回值	说明
getType()	EV. OVERLAY_POLYLINE	返回覆盖物的类型
setLngLats(lnglats:Array<EV.LngLat>)	None	设置折线点
getLngLats()	Array<EV.LngLat>	返回折线点

5.5示例

```
var p1 = new EV.LngLat(119,30);
var p2 = new EV.LngLat(120,31);
var p3 = new EV.LngLat(119, 31);
var line = new EV.PolyLine([p1,p2,p3], {
    strokeWidth:3,
    strokeColor:"green",
    strokeDashstyle:"dash"
});
map.addOverlay(line);
```

运行效果:



6 EV.Polygon

6.1描述

派生自 EV.Overlay ， 描述了一个多边形覆盖物。

6.2构造函数

EV. Polygon (lnglats:Array<EV.LngLat>, style) 至少 3 个点的坐标数组构造一个面。

style: 这个是一个配置对象，直接使用 javascript 的对象配置

可以配置的样式，参看 EV.Overlay 的 set 和 get 方法

6.3属性

6.4方法

名称	返回值	说明
getType()	EV. OVERLAY_POLYGON	返回覆盖物的类型
setLngLats (lnglats:Array<EV.LngLat>)	None	设置多边形外边界点
getLngLats ()	Array<EV.LngLat>	返回多边形外边界点

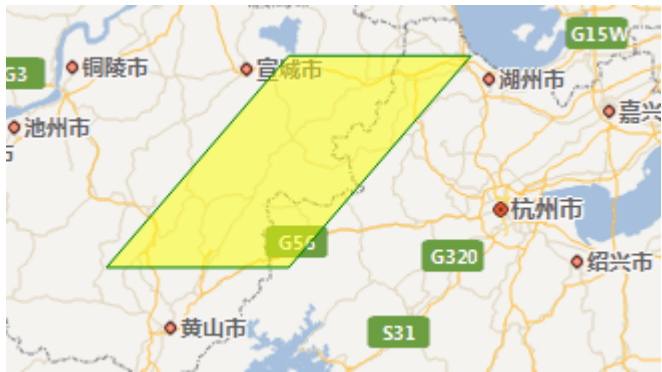
6.5示例

```
var p1 = new EV.LngLat(119,30);
var p2 = new EV.LngLat(120,31);
var p3 = new EV.LngLat(119, 31);
var p4 = new EV.LngLat(118, 30);

var polygon = new EV.Polygon([p1,p2,p3,p4], {
  strokeWidth:1,
  strokeColor:"green",
  fillColor:"yellow",
```

```
fillOpacity:0.5
});
map.addOverlay(polygon);
```

运行效果如下：



7 EV.Rect

7.1描述

派生自 EV.Overlay ， 描述了一个矩形覆盖物。

7.2构造函数

EV.Rect(bounds:EV.LngLatBounds, style)

bounds: 覆盖物的地理范围

style: 这个是一个配置对象，直接使用 javascript 的对象配置

可以配置的样式，参看 EV.Overlay 的 set 和 get 方法

7.3属性

7.4方法

名称	返回值	说明
getType()	EV. OVERLAY_POLYGON	返回覆盖物的类型

运行效果如下:

beforefeaturemodified: 在开始编辑对象之前触发。

afterfeaturemodified: 在编辑对象后触发，对象失去了焦点。

featurevertexclick: 对象被选中后，点击对象的顶点时触发。只适用于折线和多边形

8.2 示例

//在开始编辑之前就触发的事件: **beforefeaturemodified**

var map = new EV.Map("mapdiv", {defaultTools : false}); //创建地图对象

```
EV.Event.addListener(map.vectors, "beforefeaturemodified", function(e){
    //当前编辑的 overlay 类型
    var t = e.feature.getType();
    alert(t);
});
```

//在编辑对象后，失去焦点前触发的事件: **featuremodified**

```
EV.Event.addListener(map.vectors, "featuremodified", function(e) {
    //弹出编辑对象类型
    alert(e.feature.id);
});
```

//在编辑对象后，失去焦点后触发的事件: **afterfeaturemodified**

```
EV.Event.addListener(map.vectors, "afterfeaturemodified", function(e) {
    //弹出编辑对象类型
    alert("afterfeaturemodified" + e.feature.id);
});
```

//对象被选中后，点击对象的顶点时触发的事件: **featurevertexclick**

//单击对象的第几个点

```
EV.Event.addListener(map.vectors, "featurevertexclick", function(e) {
    points=e.feature.getLngLats();//将获得的点保存进数组
    num=e.index;
    tmp=points[num];//设置显示信息窗口的点
    if(!infow){
        infow = new EV.InfoWindow("w01", tmp, "", "你点击了第
```

```

"+(num+1)+"个点", new EV.Size(88,100), function(){
    this.destroy();
    infow = null;
});
map.addPopup(infow);
}
else
    infow.setLngLat(tmp);

});

```

五 菜单类

1 EV.ContextMenu 类

1.1 描述

地图中的右键菜单类

1.2 构造函数

EV.ContextMenu()

1.3 方法

名称	返回值	说明
addItem(item:EV.MenuItem)	None	添加一个菜单项
getItem(index:Number)	EV.MenuItem	获取第 index 个菜单项
removeItem(item:EV.MenuItem)	None	删除一个菜单项
addSeparator()	None	增加一个分隔符
removeSeparator(index:Number)	None	删除一个分隔符

<code>getItems()</code>	<code>Array<EV.MenuItem></code>	返回所有菜单项
<code>getAllSeparator()</code>	<code>Array<Number></code>	返回所有分隔符，元素为每个分隔符在菜单项中位置
<code>addEventListener(event:string, handler:Function)</code>	None	添加一个事件回调，支持的事件："beforeshow"
<code>removeEventListener(event:string, handler:Function)</code>	None	删除事件回调

1.4 示例

```

var menu = new EV.ContextMenu();
menu.addItem(new EV.MenuItem("放大", function() { map.zoomIn(); }));
menu.addItem(new EV.MenuItem("缩小", function() { map.zoomOut(); }));
menu.addItem(new EV.MenuItem("全图", function() {
    map.zoomToChina(); }));
menu.addSeparator();
menu.addItem(new EV.MenuItem("禁用下一个菜单", function() {
    var meneitem = menu.getItem(4);
    meneitem.enabled = !meneitem.enabled;
    this.text = meneitem.enabled? "禁用下一个菜单" : "启用下一个菜单";
}));
menu.addItem(new EV.MenuItem("弹出本菜单文本", function() {
    alert(this.text);
}));
menu.addItem(new EV.MenuItem("只对汽车1启用", function() {
    alert(this.menu.feature.style.label);
}));
map.addContextMenu(menu);
menu.addEventListener("beforeshow", function(evt) {
    //在这里可以根据菜单点击的坐标和位置 对菜单项进行修改
    this.getItem(5).enabled = (this.feature == car);
});

```

运行效果如下：



2 EV.MenuItem 类

2.1 描述

构造一个菜单项

2.2 构造函数

`EV.MenuItem(text :String, callback:Func ,opts:{})`

`text`: 菜单项文字

`callback`: 菜单项点击后回调

`opts`: 菜单项其他配置

2.3 属性

`text`: 菜单项文字

`enabled`: 启用或者禁用菜单点击

`menu`: 所属的菜单

2.4示例

参考 EV.ContextMenu 示例

六 工具类

1 EV.Toolbar

1.1描述

地图上工具条，如下图所示



1.2构造函数

EV.Toolbar()

1.3属性

1.4方法

名称	返回值	说明
addDeafultTools ()	None	创建默认的工具栏按钮，包括： "nav", "global", "zoomin", "zoomout", "addpoint", "addline", "addpoly", "ad dcircle", "addrect", "addellipse", "m length", "marea", "select", "edit"
addTool (type:String, option, style}	EV.Tool	在工具条上添加一个工具。 type : 工具类型，可选 "nav", "global", "zoomin", "zoomout", "addpoint", "addline", "addpoly", "ad dcircle", "addrect", "addellipse", "m length", "marea", "select", "edit"

		option: 工具的其他配置项 style: 工具操作过程中的样式, 此样式和上述 Overlay 的样式配置一致 返回值是一个工具的对象
activateTool (tool:EV.Tool)	None	使某个工具处于使能状态
activateToolByName (name:String)	None	根据工具名称去使能工具
getToolsByName()	Array<EV.Tool>	根据工具 name 属性获取工具集
initAdminAndWeather()	None	创建地图中心点行政区划和天气查询
removeTool (type:EV.Tool)	None	删除某个工具

1.5 示例

```
//map 里已经创建一个toolbar 并且添加了如下工具
"nav","global","zoomin","zoomout",
"addpoint","addline","addpoly","addcircle","mlength","mareas","select",
"edit"
var toolbar = map.toolbar;
// 它默认是不可见的 可以让它可见
toolbar.setVisible(true);
//添加的时候可以指定工具的样式
var tpoint = toolbar.addTool("addpoint",{ //这是一个标点的工具
//工具栏的文字是 "标注"
text:"标注",
//设置工具名称 可以用在activateToolByName 和getToolsByName使用的名称
name:"iaddpoint"
},{
//这个标点工具的样式和默认的不一样
externalGraphic: server + "images/marker-blue.png",
graphicWidth:17,
graphicHeight:20,
graphicYOffset: -19
});
tpoint工具的效果如下图左边 右边是默认的标注工具效果
```



//添加一个自定义样式的折线绘制工具

```
var tline = toolbar.addTool("addline",{
    text:"红色线",    //工具栏显示的名称
    name:"iaddline" //工具的名称
```

//样式配置

```
},{
    pointRadius: 4,
    graphicName: "star",
    strokeWidth: 2,
    strokeOpacity: 1,
    strokeColor: "#ff0000"    //红色线颜色
});
```

tline 的效果如左图，右图是默认折线工具效果



toolbar.initAdminAndWeather(); // 工具栏显示地图中心点行政区划和天气



2 EV.Tool 类

2.1 描述

目前工具类无直接的构造函数,都是从 EV.Toolbar 的 addTool 获得对象,或者通过 Toolbar 的 getToolsByName 获得对象

2.2 方法

名称	返回值	说明
addEventListener(event:String, handler:Function)	None	添加一个地图事件的回调响应 支持的事件: "activate "," deactivate"
removeEventListener(event:String, handler:Function)	None	删除一个地图事件的回调响应
deactivate()	None	取消工具使能状态
activate()	None	使工具使能

2.3 示例

```
//获得漫游工具
var navtool = toolbar.getToolsByName("nav")[0];
//添加事件响应
navtool.addEventListener("activate",function(){

    alert("工具: " + this.name + "被使能");
});

navtool.addEventListener("deactivate",function(){
    alert("工具: " + this.name + "去使能");
});
```

3 漫游工具

3.1 描述

派生自 EV.Tool 使能后地图执行默认的漫游工具。

type = "nav"

使用 EV.Toolbar.addTool("nav"...) 添加

工具栏上效果



map.t_nav() 方法可使默认的漫游工具使能

3.2 示例

参看 EV.Tool 示例

4 全图工具

4.1 描述

派生自 EV.Tool 每次点击地图定位到全中国。

type = "global"

使用 EV.Toolbar.addTool("global"...) 添加

工具栏上效果



全图工具是按钮型工具，不会产生"activate","deactivate"等事件，会产生"click"事件

map.t_global() 方法可使默认的全图工具执行一次

4.2 示例

```
var globaltool = toolbar.getToolsByName("global")[0];
globaltool.addEventListener("click", function() {

    alert("工具: " + this.name + "被点击");
});
```

技术支持：北京易伟航科技有限公司

```
});
```

5 拉框放大工具

5.1 描述

派生自 EV.Tool 使能后地图执行拉框放大操作。

`type = "zoomin"`

使用 `EV.Toolbar.addTool("zoomin "...)` 添加

工具栏上效果  放大

`map.t_zoomin()` 方法可使默认的拉框放大工具使能

5.2 示例

6 拉框缩小工具

6.1 描述

派生自 EV.Tool 使能后地图执行拉框缩小操作。

`type = "zoomout"`

使用 `EV.Toolbar.addTool("zoomout "...)` 添加

工具栏上效果  缩小

`map.t_zoomout()` 方法可使默认的拉框缩小工具使能

6.2 示例

7 标点工具

7.1 描述

派生自 `EV.Tool` 使能后地图执行标点操作，每次操作完毕，在 `map` 上新建一个 `EV.Marker` 类型的覆盖物。

```
type = "addpoint"
```

使用 `EV.Toolbar.addTool("addpoint "...)` 添加

工具栏上效果  标点

`map.t_addpoint()` 方法可使默认的标点工具使能。

具有一个额外的 `"done"` 事件，每次标点完毕触发。

7.2 示例

下面的示例演示如何添加一个 自定义样式的标点工具，并且当每次标点结束后，不继续标点，也不添加到 `Map` 的 `Overlay` 图层中去。

```
function noContinueAndNoAdd(e) {
    alert("绘制了一个:" + e.overlay.getType() + "    此元素不会添加到地图上，而且不继续绘制");
    //停止此工具
    this.deactivate();
    //不添加到地图上
    return false;
}

//添加的时候可以指定工具的样式

var tpoint = toolbar.addTool("addpoint",{    //这是一个标点的工具
//工具栏的文字是 "标注"
    text:"标注",
//设置工具名称 可以用在activateToolByName 和getToolsByName使用的名称
    name:"iaddpoint"
},{
    //这个标点工具的样式和默认的不一样
```

技术支持：北京易伟航科技有限公司

```

        externalGraphic: server + "images/marker-blue.png",
        graphicWidth:17,
        graphicHeight:20,
        graphicYOffset: -19
    });
    //当绘制结束之后, 不添加到地图上, 而且不继续绘制
    tpoint.addEventListener("done", noContinueAndNoAdd);

```

8 折线工具

8.1描述

派生自 `EV.Tool` 使能后地图执行绘制折线操作, 每次操作完毕, 在 `map` 上新建一个 `EV.PolyLiine` 类型的覆盖物。

`type = "addline"`

使用 `EV.Toolbar.addTool("addline "...)` 添加

工具栏上效果 

`map.t_addline()` 方法可使默认的折线工具使能。

具有一个额外的 `"done"` 事件, 每次标点完毕触发。

8.2示例

下面的示例演示如何添加一个 自定义样式的折线工具, 并且当每次双击结束折线绘制后, 不继续绘制, 也不把当前绘制结果添加到 `map` 上去。

```

var tline = toolbar.addTool("addline",{
    text:"红色线",
    name:"iaddline"
},{
    pointRadius: 4,
    graphicName: "star",
    strokeWidth: 2,
    strokeOpacity: 1,
    strokeColor: "#ff0000"
});

```

```

tline.addEventListener("done", noContinueAndNoAdd);

function noContinueAndNoAdd(e) {
    alert("绘制了一个:"+e.overlay.getType() + "    此元素不会添加到地图上, 而且不继续绘制");
    //停止此工具
    this.deactivate();
    //不添加到地图上
    return false;
}

```


9 多边形工具

9.1描述

派生自 `EV.Tool` 使能后地图执行绘制多边形操作, 每次操作完毕, 在 `map` 上新建一个 `EV.Polygon` 类型的覆盖物。

`type = "addpoly"`

使用 `EV.Toolbar.addTool("addpoly"...)` 添加

工具栏上效果 

`map.t_addpoly()` 方法可使默认的多边形工具使能。

具有一个额外的 "done" 事件, 每次标点完毕触发。

9.2示例

下例演示了, 如何使默认的多边形工具 绘制一个多边形后就恢复漫游状态。

```

var addpolytool = toolbar.getToolsByName("addpoly")[0];
addpolytool.addEventListener("done", function(e) {

    this.deactivate();//当前工具去使能, 恢复漫游状态
    return true; //添加到map中

});

```


10 矩形工具

10.1 描述

派生自 `EV.Tool` 使能后地图执行绘制矩形操作，每次操作完毕，在 `map` 上新建一个 `EV.Rect` 类型的覆盖物。

`type = "addrect"`

使用 `EV.Toolbar.addTool("addrect "...)` 添加

工具栏上效果 

`map.t_addrect()` 方法可使默认的多边形工具使能。

具有一个额外的 `"done"` 事件，每次标点完毕触发。

10.2 示例

下例演示如何添加一个自定义的空心矩形工具，并且当绘制结束后，返回漫游状态，并且不添加绘制结果。

```
toolbar.addTool("addrect",{
    text:"空心矩形",
    name:"iaddrect"
},{
    fill : false
}).addEventListener("done", noContinueAndNoAdd);
function noContinueAndNoAdd(e) {
    alert("绘制了一个:"+e.overlay.getType() + "    此元素不会添加到地图上，而且不继续绘制");
    //停止此工具
    this.deactivate();
    //不添加到地图上
    return false;
}
```


11 椭圆工具

11.1 描述

派生自 `EV.Tool` 使能后地图执行绘制椭圆操作，每次操作完毕，在 `map` 上新建一个 `EV.Ellipse` 类型的覆盖物。

`type = "addellipse"`

使用 `EV.Toolbar.addTool("addellipse "...)` 添加

工具栏上效果  椭圆

`map.t_addellipse()` 方法可使默认的椭圆形工具使能。

具有一个额外的 `"done"` 事件，每次标点完毕触发。

11.2 示例

12 圆形工具

12.1 描述

派生自 `EV.Tool` 使能后地图执行绘制圆操作，每次操作完毕，在 `map` 上新建一个 `EV.Circle` 类型的覆盖物。

`type = "addecircle"`

使用 `EV.Toolbar.addTool("addecircle "...)` 添加

工具栏上效果  圆形

`map.t_addecircle()` 方法可使默认的圆形工具使能。

具有一个额外的 `"done"` 事件，每次标点完毕触发。

12.2 示例

13 测量长度工具

13.1 描述

派生自 `EV.Tool` 使能后地图执行测量长度工具。

`type = "mlength"`

使用 `EV.Toolbar.addTool("mlength "...)` 添加

工具栏上效果 

`map.t_mlength()` 方法可使默认的测量长度使能。

具有 额外的 `"measure"` 事件, 每次鼠标点击增加了测量关键点后, 计算长度。

具有 额外的 `"measurepartial"` 事件, 每次鼠标移动, 计算鼠标所在位置的, 计算长度。

13.2 示例

下例显示了一个自定义的 测量工具, 测量过程是 黑色 虚线, 并且绑定测量结果事件, 测量完成后, 弹出测量结果, 切换到漫游工具。

//添加一个自定义的测量长度工具

```
var mlen = toolbar.addTool("mlength", {
  text: "自定义测量"
}, {
  strokeWidth: 2,
  strokeDashstyle: "dot",
  strokeColor: "#000000"
});
mlen.addEventListener("measure", function(event) {
  var geometry = event.geometry; //测量的几何体对象
  var units = event.units; //测量结果单位
  var order = event.order; //1 长度 2面积
  var measure = event.measure; //测量结果

  var out = "";
```

技术支持: 北京易伟航科技有限公司

```

//测量长度
if (order == 1) {
    out += "测量结果: " + measure.toFixed(3) + " " + units;
}
//测量面积
else {
    out += "测量结果: " + measure.toFixed(3) + " " + units +
"<sup>2</sup>" + "sup>";
}
alert(out);

this.deactivate();
});

```

运行效果:



14 测量面积工具


14.1 描述

派生自 EV.Tool 使能后地图执行测量面积工具。

type = "marea"

使用 EV.Toolbar.addTool("marea "...) 添加

技术支持: 北京易伟航科技有限公司

工具栏上效果 

`map.t_marea()` 方法可使默认的测量面积使能。

具有 额外的 "measure" 事件, 每次鼠标点击增加了测量关键点后, 计算面积。

具有 额外的 "measurepartial" 事件, 每次鼠标移动, 计算鼠标所在位置的, 计算面积。

14.2 示例


15 编辑工具

15.1 描述

派生自 `EV.Tool` 使能后地图执行 `Overlay` 编辑工具。

`type = "edit"`

使用 `EV.Toolbar.addTool("edit "...)` 添加

工具栏上效果 

`map.t_edit()` 方法可使默认的编辑工具使能。

15.2 示例

七 服务类

此部分请参考单独的 `EV_JavaScriptServiceAPI` 文档。

八 实用类

1 EV.MarkerClusterer 类

1.1 描述

聚类效果实现类

1.2 构造函数

`EV.MarkerClusterer(map:EV.Map, config:Object)`

`map` : EV.Map 对象

`config`: 是一个 JS Object 可配置的选项如下

`config.radius` : Number 类型, 表示聚类的像素半径, 默认 40, 任意两标注的距离小于 40×2 才可能被聚合。

`config.one`: String 类型, 当聚合个数为 1 的时候, 使用的图标。

`config.clusterStyle` : Array 类型, 它包含多个层层聚合显示效果, 每个层次有如下属性

`url`: String 类型, 表示这个级别的使用的图标

`size`: Number 类型 表示这个级别的图标大小

`number`: Number 类型 表示这个级别最大的聚合个数。

1.3 方法

名称	返回值	说明
<code>cluster(lnglats:Array<EV.LngLat>)</code>	None	对传入的坐标点数组进行聚合显示
<code>setRadius(radius:Number)</code>	None	设置聚合的像素半径, 默认 40
<code>setColor(color:Color)</code>	None	设置聚合文字的颜色
<code>clearClusters()</code>	None	清空聚合显示

1.4 示例

//先请求测试数据

```
EV.ajax({
  url:"demo/clusterdata.js",
  success:function(request){

    eval(request.responseText);

    onDataLoaded();
  }
});
```

```
function onDataLoaded(){

  //配置聚合参数
  var config = {};
  config.clusterStyle=[];
  var c_u = 'img/cluster/m';
  for(var i=0;i<5;i++){
    config.clusterStyle.push({url:c_u+i+".png",number:100*(i+1)});
  }

  //构造聚合坐标点
  var _cluster_data=[];
  for(var i=0;i<2000;i++){
    var lng = parseFloat(cluster_data[i][0]);
    var lat = parseFloat(cluster_data[i][1]);
    if(lng&&lat){
      _cluster_data.push(new EV.LngLat(lng,lat));
    }
  }
  var markerClusterer = new EV.MarkerClusterer(map,config);
  // markerClusterer.setColor('#00FFCC');//设置字体颜色
  markerClusterer.cluster(_cluster_data);
}
```

运行效果如图:

