

# **EV\_JavaScriptServiceAPI**

## **服务用户手册**

编 制：北京易伟航科技有限公司

修订日期：2015-2-6

## 说明

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

服务 API 提供了：本地搜索，位置偏移，路线计算，地理编码，逆地理编码，IP 定位、行政区划查询等服务，通过使用服务 API 结合地图 API，您可以将本地搜索，位置偏移，路线计算，地理编码，逆地理编码，IP 定位、行政区划查询等操作的查询结果显示在地图上，为您的网站提供更有针对性而且方便的应用。

用户在阅读下面的文档时，可以根据需求对文档进行查询或跳跃式阅读。从而更好地应用 API 类服务于自己的领域。

## 目录

<b>EV_JavaScriptServiceAPI</b> .....	<b>1</b>
<b>服务用户手册</b> .....	<b>1</b>
<b>1 EV. ServiceSH 类</b> .....	<b>1</b>
1.1 描述 .....	1
1.2 构造函数 .....	1
1.3 成员方法 .....	1
1.4 SHRequest 对象规范 .....	1
1.5 SHResult 对象规范 .....	2
<b>2 EV. ServiceLS 类</b> .....	<b>2</b>
2.1 描述 .....	2
2.2 构造函数 .....	2
2.3 全局变量 .....	3
2.4 成员变量 .....	3
2.5 成员方法 .....	3
2.6 LSRequest 对象规范 .....	4
2.7 LSArea 对象规范 .....	4
2.8 LSClass 对象规范 .....	4
2.9 LSPageinfo 对象规范 .....	5
2.10 LSResult 对象规范 .....	5
2.11 LSResultData 对象规范 .....	5
<b>3 EV. ServiceGC 类</b> .....	<b>5</b>
3.1 描述 .....	5
3.2 构造函数 .....	6
3.3 成员方法 .....	6
3.4 GCRequest 对象规范 .....	6
3.5 RGCCRequest 对象规范 .....	6
3.6 GCResult 对象规范 .....	7
3.7 RGCCResult 对象规范 .....	8

---

4	EV. ServiceRC 类.....	9
4.1	描述.....	9
4.2	构造函数.....	9
4.3	成员方法.....	9
4.4	成员变量.....	10
4.5	RCRequest 对象规范.....	10
4.6	RCResult 对象规范.....	11
4.7	RouterRCData 对象规范.....	11
5	EV. ServiceIP 类.....	12
5.1	描述.....	12
5.2	构造函数.....	12
5.3	成员方法.....	12
5.4	IPRequest 对象规范.....	13
5.5	IPResult 对象规范.....	13
6	EV. ServiceRAC 类.....	13
6.1	描述.....	13
6.2	构造函数.....	13
6.3	成员方法.....	14
6.4	RACRequest 对象规范.....	14
6.5	RACResult 对象规范.....	14
7	EV. Admin 类.....	15
7.1	描述.....	15
7.2	构造函数.....	15
7.3	成员方法.....	15
7.4	PositionRequest 对象规范.....	15
7.5	PositionResult 对象规范.....	16
7.6	ListsubRequest 对象规范.....	16
7.7	ListsubResult 对象规范.....	16
8	EV.Weather 类.....	17
8.1	描述.....	17

---

8.2	构造函数.....	17
8.3	成员方法.....	17
8.4	PositionRequest 对象规范 .....	18
8.5	WeatherResult 对象规范 .....	18
8.5.1	WeatherResult 对象 .....	18
8.5.2	WeathObject 对象 .....	19
8.5.3	WeathIndex 对象 .....	19
8.6	AdminRequest 对象规范 .....	19
8.7	RegionRequest 对象规范 .....	20
8.8	RegionResult 对象规范 .....	20
8.8.1	RegionWeather 对象 .....	20
8.9	天气相关信息表.....	21
8.9.1	天气编码表.....	21
8.9.2	风向编码表.....	22
8.9.3	风力编码表.....	22
8.10	天气指数类型.....	22
8.10.1	洗车指数.....	23
8.10.2	出行指数.....	23
8.10.3	穿衣指数.....	23
8.10.4	空气指数.....	23
8.10.5	舒适指数.....	24

# 1 EV. ServiceSH 类

## 1.1 描述

EV.ServiceSH 是处理经纬度坐标偏移的服务类。

## 1.2 构造函数

构造函数	返回值	说明
EV.ServiceSH(map:EV.Map)	EV.ServiceSH 对象	构造一个处理经纬度坐标偏移的服务类。

## 1.3 成员方法

方法	返回值	说明
shift(request:SHRequest,callback:function(result:SHResult))	None	处理经纬度坐标之间的转换，参数 request 为封装了请求参数的 SHRequest 类，callback 为客户端的回调函数。

## 1.4 SHRequest 对象规范

属性	类型	说明
request	Object	<p>request 是一个对象，包含三个属性：tid、points 和 show</p> <ul style="list-style-type: none"><li>➤ points: String 或 EV.LngLat[] 必选。需要转换的坐标串或坐标对象数组。 可以为一个或多个坐标点转换。如果是坐标串，坐标点用经度坐标+逗号+纬度坐标来表示，多个点用分号隔开。 GET 请求字符串，只支持 4096 字节长度。 POST 请求字符串，支持长度视浏览器而定。</li><li>➤ tid: String 可选。回调参数。</li></ul>

		<p>➤ <b>show: Number</b></p> <p>可选。是否显示原坐标，取值：</p> <p><b>show=0</b> 不显示</p> <p><b>show=1</b> 显示</p> <p>默认为：show=0</p>
--	--	---

## 1.5 SHResult 对象规范

属性	类型	说明
status	string	<p>ok 或者 error。</p> <p>ok 说明有结果。可以从 <b>result</b> 属性中得到数据。</p> <p>error 说明检索发生错误。</p>
result	object	<p>返回的结果信息。</p> <p>➤ <b>tid</b>: 回调参数</p> <p>➤ <b>points</b> 返回来的坐标数组，元素属性如下：</p> <p><b>lng</b>: 经度</p> <p><b>lat</b>: 纬度</p> <p><b>original_lng</b>: 原始经度（只有在 <b>show=1</b> 时包含此值）</p> <p><b>original_lat</b>: 原始纬度（只有在 <b>show=1</b> 时包含此值）</p>

## 2 EV. ServiceLS 类

### 2.1 描述

用于与地图相关的 POI 信息的搜索，不仅支持指定关键字或者 POI 类别的搜索，更提供了点线面在指定半径范围内的关键字或者类别的搜索。

### 2.2 构造函数

构造函数	返回值	说明
------	-----	----

EV. ServiceLS(map:EV.Map)	EV. ServiceLS 对象	创建一个本地搜索类实例。
---------------------------	------------------	--------------

## 2.3 全局变量

名称	描述
WTK_TYPE_STRING	区域搜索的 WTK 参数类型——字符串
WTK_TYPE_POINT	区域搜索的 WTK 参数类型——点
WTK_TYPE_CIRCLE	区域搜索的 WTK 参数类型——圆
WTK_TYPE_RECT	区域搜索的 WTK 参数类型——矩形
WTK_TYPE_LINE	区域搜索的 WTK 参数类型——线
WTK_TYPE_MULTI_LINE	区域搜索的 WTK 参数类型——多线
WTK_TYPE_PLOYGON	区域搜索的 WTK 参数类型——多边形
WTK_TYPE_MULTI_PLOYGON	区域搜索的 WTK 参数类型——多多边形
WTK_TYPE_COLLECTION	区域搜索的 WTK 参数类型——集合

## 2.4 成员变量

名称	描述
_lsdata	保存 LS 服务时的返回的数据

## 2.5 成员方法

方法	返回值	说明
search(request:LSRequest, callback?:function(result:LSResult))	None	<p>将请求发送给服务器，并将查询结果返回给回调函数。</p> <ul style="list-style-type: none"> <li>➤ request 为请求参数；</li> <li>➤ callback 为回调函数，可以不传 callback，如果不传 callback，可以调用默认的渲染方法 renderer 进行渲染。</li> </ul>
renderer(map:EV.Map,resultContainer,data?:LSResult)	none	<p>此方法为默认的渲染方法，如果用户不提供回调函数，可以通过此方法将搜索结果渲染到地图 map 上。</p> <ul style="list-style-type: none"> <li>➤ resultContainer 是显示文本信息的 DIV 对象或 ID 号；</li> <li>➤ data 为渲染的数据，一般为搜索的结果（可选）。</li> </ul>



		如果不传入此参数，则渲染数据为本次搜索的结果。
--	--	-------------------------

## 2.6 LSRequest 对象规范

要使用 EV.ServiceLS 进行检索的请求的字符串，以及各个字段说明。

属性	类型	说明
city	string	必选。城市参数，城市名称。
keyword	string	可选。搜索关键词，最长 128 个字符。 支持智能数字匹配。 智能全角半角匹配。
lsclass	LSClass	可选。分类搜索。
area	LSArea	可选。区域搜索。
pageinfo	LSPageinfo	可选。页码信息。

## 2.7 LSArea 对象规范

属性	类型	说明
range	object	参考 WTK 字符串说明。

## 2.8 LSClass 对象规范

属性	类型	说明
classid	string	可选。POI 类别，具体参考 class_id_name.xls。采用四维的类别编码 支持： <ul style="list-style-type: none"> <li>➤ classid=1080 编号为 1080</li> <li>➤ classid=2070:2F80 2070 到 2F80 的分类编号</li> <li>➤ classid=1080 2070:2F80 2070 到 2F80 的分类编号以及 1080</li> </ul>

## 2.9 LSPageinfo 对象规范

属性	类型	说明
page	Number	第几页,，默认第一页。
number	Number	每页显示的条数，默认 10 条。

## 2.10 LSResult 对象规范

属性	类型	说明
status	String	ok 或者 error 。如果为 error。会在 error 属性中留有错误信息。
result	LSResultData	返回的结果信息。

## 2.11 LSResultData 对象规范

属性	类型	说明
points	➤ object	➤ 返回来的结果说明： ➤ name: 名称 ➤ telephone: 电话号码 ➤ address: 地址 ➤ lng: 经度 ➤ lat: 纬度 ➤ zipcode: 邮政编码

# 3 EV. ServiceGC 类

## 3.1 描述

EV. ServiceGC 包含两个功能，地理编码功能：用于根据数据点的空间地址(如省市、街道、楼层、房间等)描述，对地址描述进行解析，返回该地址描述对应

的详细地理信息。逆地理编码功能：根据给出的纬度坐标，可以得出省、市、区、街、门牌号等信息的功能。

## 3.2 构造函数

构造函数	返回值	说明
EV.ServiceGC(map:EV.Map)	EV.ServiceGC 对象	创建新 EV.ServiceGC 实例。

## 3.3 成员方法

方法	返回值	说明
gcEncoding(request:GCRrequest,callback::function(result:GCRresult))	None	地理编码,根据数据点的空间地址(如省市、街道、楼层、房间等)描述,对地址描述进行解析,返回该地址描述对应的详细地理信息。 ➤ request: 请求参数。 ➤ callback: 为回调函数。
rgcEncoding(request:RGCRrequest,callback::function(result:RGCRresult))	None	逆地理编码: 根据给出的纬度坐标,可以得出省、市、区、街、门牌号等信息的功能。 ➤ request: 请求参数。 ➤ callback: 为回调函数。

## 3.4 GCRrequest 对象规范

属性	类型	说明
address	object	为封装了请求数据的对象。 ➤ addr: 地址, 类型为 String。越详细越好, 最好是 “**省**市**区**路**号”; ➤ city: 城市, 类型为 String。

## 3.5 RGCRrequest 对象规范

属性	类型	说明
location	EV.LngLat	查询的经纬度对象(必需)。

course	integer	<p>行驶方向与正北方向的夹角，方向值以正北向为 0 度，顺时针旋转，单位：1/10 度。</p> <p>取值范围：0-3599</p> <ul style="list-style-type: none"> <li>➤ 0：自南向北；</li> <li>➤ 900：自西向东；</li> <li>➤ 1800：自北向南；</li> <li>➤ 2700：自东向西。</li> </ul> <p>默认不考虑此参数。</p> <p>如果没有该方向的道路返回对条件模糊匹配的道路信息。</p>
type	integer	<p>查询类型。</p> <ul style="list-style-type: none"> <li>➤ 1：地标查询；</li> <li>➤ 10：道路查询；</li> <li>➤ 11：地标和道路查询。</li> </ul> <p>将多个值相加来支持多种查询，如 11 表示地标和道路查询。</p> <p>默认为 1。</p>

### 3.6 GCRResult 对象规范

属性	类型	说明
status	string	<p>ok 或者 error 。</p> <p>ok 说明有结果。可以从 result 属性中得到数据。</p> <p>error 说明发生错误。</p>
result	object	<p>返回的结果信息。</p> <ul style="list-style-type: none"> <li>➤ time: 耗时</li> <li>➤ bound: 区域</li> <li>➤ total: 查询到点的个数</li> <li>➤ cutwordinfo: 分词信息</li> <li>➤ addressinfo <ul style="list-style-type: none"> <li>■ addresstype: 地址类型</li> <li>■ accuracy: 精确度</li> </ul> </li> <li>➤ points: 返回来的地址数组 <ul style="list-style-type: none"> <li>■ id: 编号</li> <li>■ name: 名称</li> </ul> </li> </ul>

		<ul style="list-style-type: none"> <li>■ address: 地址</li> <li>■ lng: 经度</li> <li>■ lat: 纬度</li> </ul>
--	--	---

### 3.7 RGCResult 对象规范

属性	类型	说明
status	string	<p>ok 或者 error 。</p> <p>ok 说明有结果。可以从 result 属性中得到数据。</p> <p>error 说明发生错误，在 error 属性中留有错误信息</p>
result	object	<p>返回的结果信息。</p> <ul style="list-style-type: none"> <li>➤ address: 查询点最近的 POI 地址</li> <li>➤ road_address: 查询道路地址</li> <li>➤ district: 行政区域编码</li> <li>➤ district_text: 行政区域</li> <li>➤ point: 经纬度坐标 <ul style="list-style-type: none"> <li>■ name: 查询点最近的 POI 名称</li> <li>■ lng: POI 经度</li> <li>■ lat: POI 纬度</li> </ul> </li> <li>➤ road: 道路 <ul style="list-style-type: none"> <li>■ name: 道路名称</li> <li>■ road_level: 道路等级</li> <li>■ limit_speed: 道路限速</li> <li>■ lng: 道路最近的 POI 的经度</li> <li>■ lat: 道路最近的 POI 的纬度</li> </ul> </li> </ul>

#### 附录：道路等级说明

属性类型名称	属性值名称
道路等级	高速路
	都市高速路
	国道
	省道
	县道

	乡镇村道
	其它道路
	九级路
	轮渡
	行人道路

## 4 EV. ServiceRC 类

### 4.1 描述

根据给定的两点或者多点的位置检索各点间的导航路线。

### 4.2 构造函数

构造函数	返回值	说明
EV. ServiceRC(map:EV.Map)	EV. ServiceRC 对象	创建 EV. ServiceRC 对象。

### 4.3 成员方法

方法	返回值	说明
route(request:RCRequest, callback?:function(result:RCResult))	None	<p>将请求发送给服务器，并将查询结果返回给回调函数。回调函数可以没有，如果没有回调函数，可以调用默认的渲染方法 <code>renderRC</code> 对查询结果进行渲染。</p> <ul style="list-style-type: none"> <li>➤ <b>request:</b> 请求参数 <code>request</code> 可以有两种结构(请参考 <code>RCRequest</code> 对象规范)，此方法将根据 <code>request</code> 的不同自动进行处理。</li> <li>➤ <b>callback:</b> 为回调函数，可以不传 <code>callback</code>，如果不传 <code>callback</code>，可以调用默认的渲染方法 <code>renderRC</code> 进行渲染。</li> </ul>

renderRC(map:EV.Map, resultContainer, data?:RCResult)	None	<p>RC 默认的渲染方法。用户可以调用此方法将之后会将结果渲染到地图 map 上，或指定的结果区 &lt;div&gt;resultContainer 上。</p> <ul style="list-style-type: none"> <li>➤ map: 是 EV.Map 对象。</li> <li>➤ resultContainer: 是 ID 号或 DIV 对象。</li> <li>➤ data: 为渲染的数据，一般为 route() 的返回结果（可选）。如果不传入此参数，则渲染数据为本次搜索的结果。</li> </ul>
---	------	---

## 4.4 成员变量

名称	描述
_rcdata	保存 RC 服务时的返回数据
_start	导航起始点
_end	导航结束点
_start_data	data LS 搜索导航起始点的数据
_end_data	data LS 搜索导航结束点的数据

## 4.5 RCRequest 对象规范

要使用 EV.ServiceRC 类 route 进行检索的请求参数规范。

属性	类型	说明
end	EV.LngLat 或对象	<p>必选。查询的目的地。</p> <p>如果为 EV.LngLat，则只能传递给 EV.ServiceRC 对象的 route 方法。</p> <p>如果为对象(例如{city：北京,keyword：北京大学})则可以传递给 EV.ServiceRC 对象的 route 和 searchLS 方法，route 方法直接导航，而 searchLS 方法查询出一个列表。</p>
start	EV.LngLat 或对象	<p>必选。查询的起始点。</p> <p>如果为 EV.LngLat，则只能传递给 EV.ServiceRC 对象的 route 方法。</p>

		如果为对象(例如{city : 北京,keyword : 北京大学})则可以传递给 EV.ServiceRC 对象的 route 和 searchLS 方法, route 方法直接导航, 而 searchLS 方法查询出一个列表。
waypoints	EV.LngLat[]	可选。途经点, 最多支持 16 个。
type	Number	可选策略 。 <ul style="list-style-type: none"> <li>➤ 0: 最短</li> <li>➤ 1: 最快</li> <li>➤ 2: 最经济</li> </ul> 默认: 1

## 4.6 RCRestult 对象规范

属性	类型	说明
status	String	服务端返回的请求状态。 ok 说明有结果。可以从 result 属性中得到数据。 error 说明检索发生错误。
result	RouteRCData	返回的结果信息。

## 4.7 RouteRCData 对象规范

属性	类型	说明
distance	Number	距离
time	Number	时间
events	Object	返回来的结果串: <ul style="list-style-type: none"> <li>➤ roadname: 线路名称</li> <li>➤ lng: 经度</li> <li>➤ lat: 纬度</li> <li>➤ nextdistance: 到下一事件的距离(单位: 米)</li> <li>➤ reachtype: 到达类型</li> <li>➤ limitspeed: 限速</li> <li>➤ turntype: 转向类型</li> <li>➤ roadlevel: 道路等级</li> </ul>



		<ul style="list-style-type: none"> <li>➤ <b>changename:</b> 改变类型</li> <li>➤ <b>eventprompt:</b> 系统生成的事件提示</li> <li>➤ <b>realtraffic:</b> 实时路况 <ul style="list-style-type: none"> <li>■ 缓慢</li> <li>■ 畅通</li> <li>■ 拥堵</li> <li>■ 无数据</li> </ul> </li> </ul>
route	Object	<ul style="list-style-type: none"> <li>➤ <b>pointnum:</b> 轨迹的关键点的经纬度坐标</li> <li>➤ <b>points:</b> 经纬度坐标 <ul style="list-style-type: none"> <li>■ lng: 经度</li> <li>■ lat: 纬度</li> </ul> </li> </ul>

## 5 EV.ServiceIP 类

### 5.1 描述

EV.ServiceIP 可以根据 IP 地址判断该 IP 来自于哪里，定位到城市。

### 5.2 构造函数

构造函数	返回值	说明
EV.ServiceIP (map:EV.Map)	EV.ServiceIP 对象	构造一个 IP 定位的服务类。

### 5.3 成员方法

方法	返回值	说明
locate(request:IPRequest,callback:function(result:IPRequest))	None	<p>针对指定 IP 进行定位, 参数 request 为封装了请求参数的 IPRequest 类, callback 为客户端的回调函数。</p> <p>如果 IP 地址为空或格式不正确, 该方法将抛出异常。</p>

## 5.4 IPRequest 对象规范

属性	类型	说明
request	Object	request 是一个对象，包含一个属性：ip ➤ ip: String 必选。需要定位的 IP 地址。用点分十进制表示，如：127.0.0.1

## 5.5 IPResult 对象规范

属性	类型	说明
status	string	ok 或者 error 。 ok 说明有结果。可以从 result 属性中得到数据。 error 说明检索发生错误。
result	object	返回的结果信息。 ■ district: 行政区划编码 ■ district_text: 行政区划名称 ■ error: 错误信息（当 status 为 error 时）

# 6 EV. ServiceRAC 类

## 6.1 描述

EV.ServiceRAC 可以根据行政区划码获取该地区的边界线。

## 6.2 构造函数

构造函数	返回值	说明
EV. ServiceRAC (map:EV.Map)	EV. ServiceRAC 对象	构造一个行政区划的服务类。

## 6.3 成员方法

方法	返回值	说明
getBoundary(request:RACRequest,callback:function(result:RACResult))	None	<p>根据行政区划码获取边界线，参数 request 为封装了请求参数的 RACRequest 类，callback 为客户端的回调函数。</p> <p>如果行政区划码为空或格式不正确，该方法将抛出异常。</p>

## 6.4 RACRequest 对象规范

属性	类型	说明
request	Object	<p>request 是一个对象，包含两个属性：admincode 和 zoom。</p> <ul style="list-style-type: none"> <li>➤ admincode: String 必选。需要获取边界线的行政区划码，如：110000。</li> <li>➤ adminname: String 需要获取边界线的行政区划名称。</li> </ul>

## 6.5 RACResult 对象规范

属性	类型	说明
status	String	<p>ok 或者 error 。</p> <ul style="list-style-type: none"> <li>➤ ok 说明有结果。可以从 result 属性中得到数据。</li> <li>➤ error 说明检索发生错误，在 error 属性中留有错误信息。</li> </ul>
result	Object	<ul style="list-style-type: none"> <li>➤ name 行政区域文字描述</li> <li>➤ geo 行政区域的 WKT 几何边界</li> </ul>

## 7 EV. Admin 类

### 7.1 描述

EV.Admin 查询行政区划的地图中心点，最佳显示级别的方法。

### 7.2 构造函数

构造函数	返回值	说明
EV. Admin (map:EV.Map)	EV. Admin 对象	构造一个行政区划关系的服务类。

### 7.3 成员方法

方法	返回值	说明
position (request:PositionRequest,c allback:function(result:Pos itionResult))	None	根据经纬度位置获取该位置的省，市，县三级行政区划。
listsub (request:ListsubRequest,c allback:function(result: ListsubResult))	None	根据行政区划编码获取该行政区划的下级行政区划列表。

### 7.4 PositionRequest 对象规范

属性	类型	说明
request	Object	request 是一个对象，包含两个属性：lng 和 lat。 ➤ lng: Number 必选。查询位置的经度。

		➤ lat: Number 必选。查询位置的纬度。
--	--	------------------------------

## 7.5 PositionResult 对象规范

属性	类型	说明
status	String	ok 或者 error 。 ➤ ok 说明有结果。可以从 result 属性中得到数据。 ➤ error 说明检索发生错误，在 error 属性中留有错误信息。
Result	Object	返回的结果信息。 ➤ pcode: 省行政区划代码 ➤ ccode: 地市行政区划代码 ➤ code: 区县行政区划代码 ➤ province: 省名称 ➤ city: 地市名称 ➤ dis: 区县名称

## 7.6 ListsubRequest 对象规范

属性	类型	说明
request	Object	request 是一个对象，只有一个属性。 ➤ id: 必选。 所要查询的行政区划编码。

## 7.7 ListsubResult 对象规范

属性	类型	说明
status	String	ok 或者 error 。 ➤ ok 说明有结果。可以从 result 属性中得到数据。 ➤ error 说明检索发生错误，在 error 属性中留有错误信息。

Result	Object	<p>返回的结果信息。</p> <ul style="list-style-type: none"> <li>➤ id: 行政区划代码</li> <li>➤ subs: 子行政区划列表, Array&lt;Listsub&gt;</li> <li>➤ Listsub <ul style="list-style-type: none"> <li>■ id: 行政区划代码</li> <li>■ lon: 中心位置经度</li> <li>■ lat: 中心位置纬度</li> <li>■ level: 最佳显示级别</li> <li>■ name: 行政区划名称</li> </ul> </li> </ul>
--------	--------	---

## 8 EV.Weather 类

### 8.1 描述

根据经纬度点位置、行政区划、矩形范围查询天气预报数据。

### 8.2 构造函数

构造函数	返回值	说明
EV. Weather (map:EV.Map)	EV. Weather 对象	构造一个天气预报查询的服务类。

### 8.3 成员方法

方法	返回值	说明
getposition (request:PositionRequest,c allback:function(result: WeatherResult))	None	根据经纬度位置获取该位置天气预报数据。
getadmin (request:AdminRequest,ca	None	根据行政区划获取该位置天气预报数据。返回结果对象结构和 getposition 相同

llback:function(result: WeatherResult))		
getregion (request:RegionRequest,callback:function(result: RegionResult))	None	根据经纬度矩形范围获取该位置天气预报数据。
windString(code:Number)	String	根据风向编码返回风向字符串。
weatherString (code:Number)	String	根据天气编码返回天气字符串。

## 8.4 PositionRequest 对象规范

属性	类型	说明
request	Object	<p>request 是一个对象，包含两个属性：lng 和 lat。</p> <ul style="list-style-type: none"> <li>➤ lng: Number 必选。查询位置的经度。</li> <li>➤ lat: Number 必选。查询位置的纬度。</li> <li>➤ periodtype:String 可选。预报周期，取值： daily：今天实时天气和后 3 天预报天气，默认值，返回 WeatherResult 对象 hourly: 60 小时内天气预报（目前未开放，需要请联系）</li> </ul>

## 8.5 WeatherResult 对象规范

### 8.5.1 WeatherResult 对象

属性		类型	说明
status		String	ok 表示正常返回
publishDate		Date	发布时间 如："2015-02-06T09:10:10"

location		Object	行政区划
	admincode	Number	市行政区划编码
	adminname	String	市名
currentWeather		WeathObject	当前天气
forecast		Array< WeathObject>	预报的三天天气
	0	WeathObject	第一天预报天气
	1	WeathObject	第二天预报天气
	2	WeathObject	第三天预报天气

### 8.5.2 WeathObject 对象

属性	类型	说明
temperature	String	温度
time	Date	时间
weather	String	天气编码，参看天气编码表
windDirection	String	风向编码，参看风向编码表
windPower	String	风力编码，参看风力编码表
indices	Array< WeathIndex>	天气指数（只有第一天的天气预报数据包含此数据）

### 8.5.3 WeathIndex 对象

属性	类型	说明
level	String	指数等级
type	String	指数类型，请参看指数类型表

## 8.6 AdminRequest 对象规范

属性	类型	说明
request	Object	request 是一个对象，包含两个属性：lng 和 lat。 ➤ admincode: Number 必选。行政区划代码。



## 8.7 RegionRequest 对象规范

属性	类型	说明
request	Object	<p>request 是一个对象，包含两个属性：lng 和 lat。</p> <ul style="list-style-type: none"> <li>➤ box: EV.LngLatBounds 必选。经纬度范围。</li> <li>➤ level: Number 必选。显示级别，一般和地图级别保持一致。</li> </ul>

## 8.8 RegionResult 对象规范

属性	类型	说明
status	String	ok 表示正常返回
data	Array<RegionWeather>	返回范围内天气信息点

### 8.8.1 RegionWeather 对象

属性	类型	说明
lon	String	位置经度
lat	String	位置纬度
adminName	String	行政区划名称
temperature	String	温度
weather	String	天气编码，参看天气编码表
windDirection	String	风向编码，参看风向编码表
windPower	String	风力编码，参看风力编码表

## 8.9 天气相关信息表

### 8.9.1 天气编码表

编码	天气
1	晴
2	多云
3	阴
4	阵雨
5	雷阵雨
6	雷阵雨伴有冰雹
7	雨夹雪
8	小雨
9	中雨
10	大雨
11	暴雨
12	大暴雨
13	特大暴雨
14	阵雪
15	小雪
16	小到中雪
17	中雪
18	中到大雪
19	大雪
20	大到暴雪
21	暴雪
22	雾
23	冻雨
24	沙尘暴
25	扬沙
26	浮尘
28	强沙尘暴
29	飏线
30	龙卷风
31	吹雪
32	轻雾
33	小雨-中雨
34	中雨-大雨
35	大雨-暴雨
36	暴雨-大暴雨
37	大暴雨-特大暴雨
38	霾

## 8.9.2 风向编码表

编码	说明
0	静风
7	北风
8	东北风
1	东风
2	东南风
3	南风
4	西南风
5	西风
6	西北风
-1	旋转风

## 8.9.3 风力编码表

编码	等级	说明
0	Level 0	无风
1	Level 1	软风
2	Level 2	轻风
3	Level 3	微风
4	Level 4	和风
5	Level 5	清风
6	Level 6	强风
7	Level 7	疾风
8	Level 8	大风
9	Level 9	烈风
10	Level 10	狂风
11	Level 11	暴风
12	Level 12-17	飓风

## 8.10 天气指数类型

编码	说明
1	洗车指数
2	旅游指数
3	穿衣指数

4	舒适度指数
5	空气污染扩散指数

### 8.10.1 洗车指数

编码	说明
1	适宜
2	较适宜
3	较不宜
4	不宜

### 8.10.2 出行指数

编码	说明
1	适宜
2	较适宜
3	一般
4	较不宜
5	不适宜

### 8.10.3 穿衣指数

编码	说明
1	炎热
2	热舒适
3	舒适
4	凉舒适
5	温凉
6	凉
7	冷
8	寒冷

### 8.10.4 空气指数

编码	说明
0	舒适
1	较舒适
2	较不舒适
3	很不舒适

4	极不舒适
6	较舒适
7	较不舒适
8	很不舒适
9	极不舒适

### 8.10.5 舒适指数

编码	说明
1	优
2	良
3	中
4	较差
5	差