

## uexBaiduMap

提供百度地图的相关相关功能。用户可以使用该插件在网页上嵌入一张地图，并实现对地图的基本操作，例如放大缩小，移动，点击，隐藏，显示。另外还提供了定位功能，搜索功能，路线规划功能，地理编码功能。

### 方法

`open` 打开地图  
`clean` 关闭地图  
`hide` 隐藏地图  
`show` 显示地图  
`setCenter` 设置地图的中心点  
`setZoomLevel` 设置地图的比例级别  
`setType` 设置地图的类型  
`zoomIn` 放大地图  
`zoomOut` 缩小地图  
`zoomToSpan` 将地图缩放指定的经纬跨度  
`getCurrentLocation` 获取当前位置  
`showUserLocation` 显示当前位置  
`hideUserLocation` 隐藏当前位置，并且关闭持续定位  
`fastGetLocation` 快速获取当前位置  
`geocode` 地理编码  
`reverseGeocode` 逆地理编码  
`getBaiduFromGoogle` Google 地图经纬度转换为百度地图经纬度  
`showBubbleView1` 弹出样式 1 的气泡框  
`showBubbleView2` 弹出样式 2 的气泡框  
`showBubbleView3` 弹出样式 3 的气泡框  
`showBubbleView4` 弹出样式 4 的气泡框  
`addMark` 添加标注到地图  
`addMarkNewStyle` 添加标注到地图  
`updateMark` 更新标注  
`clearMarks` 移除标注  
`addAreaMark` 添加区域标注

updateAreaMark 更新区域标注  
clearAreaMarks 移除区域标注  
addLineOverLayer 添加线形图层到地图  
addCircleOverLayer 添加圆形图层到地图  
addPolygonOverLayer 添加多边形图层到地图  
clearOverLayers 清除图层  
addOneTrack 添加一条轨迹到地图  
setTrackBeginImage 设置轨迹起点图片  
setTrackEndImage 设置轨迹终点图片  
setTrackColor 设置轨迹颜色  
clearAllTrack 清除所有轨迹  
suggestionSearch 联想猜词  
setPoiPageCapacity 设置搜索 POI 单页数据量  
poiSearchInCity 城市范围内搜索  
poiSearchNearBy 周边搜索  
poiMultiSearchNearBy 周边多关键字搜索  
poiSearchInBounds 区域内搜索  
poiMultiSearchInBounds 区域内多关键字搜索  
busLineSearch 公交线路搜索  
showRoutePlan 规划最优路线  
clearRoutePlan 清除路线

## 回调方法

cbGetCurrentLocation 获取到当前位置时的回调方法  
cbShowUserLocation 显示当前位置时的回调方法  
cbFastGetLocation 获取到快速定位时的回调方法  
cbBaiduFromGoogle 谷歌地图经纬度转百度地图经纬度的回调方法  
cbSuggestionSearch 联想猜词的回调方法  
cbPoiSearchInCity 城市内搜索的回调方法  
cbPoiSearchArea 周边搜索和区域内搜索的回调方法  
cbPoiMultiSearchArea 周边多关键字搜索和区域内多关键字搜索的回调方法  
cbGeocode 地理编码的回调方法  
cbReverseGeocode 逆地理编码的回调方法  
cbBusLineSearch 公交线路搜索的回调方法  
cbShowRoutePlan 规划路线的回调方法

## 监听方法

onTouchMark 点击标注的监听方法

onTouchBubbleView 点击气泡框监听方法

onMapDrag 拖拽地图时的监听方法

onNetworkError 网络连接失败的监听方法

onPermissionDenied key 非法时的监听方法

## open

打开地图

uexBaiduMap.open(key , x, y,width, height, longitude, latitude)

说明：

在界面的指定位置显示地图。如果传入的 key 不对，将获取不到地图数据。

参数：

key: (*String* 类型)必选 百度 api 的密钥，用户自己参考官网申请

x: (*Number* 类型)必选 x 坐标

y: (*Number* 类型)必选 y 坐标

width: (*Number* 类型)必选 地图宽度

inHeight: (*Number* 类型)必选 地图高度

inLongitude: (*Number* 类型)必选 中心点经度

inLatitude: (*Number* 类型)必选 中心点纬度

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 1

## clean

关闭地图

uexBaiduMap.clean()

说明：

关闭地图

参数：

无

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 1

## hide

隐藏地图

`uexBaiduMap.hide()`

说明:

隐藏地图，不会清除地图上的状态信息

参数:

无

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 1

## show

显示地图

`uexBaiduMap.uexBaiduMap.show()`

说明:

显示地图

参数:

无

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 1

## setType

设置地图的类型

`uexBaiduMap.uexBaiduMap.setType(type)`

说明:

设置地图的类型

参数:

**type:** (*Number* 类型) 必选 地图的类型, 0-普通模式; 1-卫星模式; 2-开启实时路况;3-关闭实时路况,其他值当做 0 处理

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 1

## setCenter

设置地图的中心点

`uexBaiduMap.setCenter(longitude, latitude)`

说明:

设置地图的中心点经纬度, 地图会移动使此点成为地图的中心点

参数:

**longitude:** (*Number* 类型) 必选 中心点的经度

**latitude:** (*Number* 类型) 必选 中心点的纬度

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 1

## **setZoomLevel**

设置地图的比例级别

`uexBaiduMap.setZoomLevel(zoomLevel)`

说明：

设置地图的比例级别

参数：

`zoomLevel`: (*Number* 类型) 必选      地图的缩放率，范围 3---18

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 1

## **zoomIn**

放大一个地图级别

`uexBaiduMap.zoomIn()`

说明：

放大一个地图级别

参数：

无

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 1

## **zoomOut()**

缩小一个地图级别

`uexBaiduMap.zoomOut()`

说明：

缩小一个地图级别

参数:

无

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 1

## **getCurrentLocation**

获得当前位置的经纬度

`uexBaiduMap.getCurrentLocation()`

参数:

无

说明:

获得当前位置的经纬度

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 2

## **showUserLocation**

显示用户位置并添加用户位置的标注

`uexBaiduMap.showUserLocation()`

说明:

显示位置后会持续定位用户位置，但不会移动地图以显示此标注。

参数:

无

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 2

## hideUserLocation

隐藏用户位置的标注，关闭持续定位

`uexBaiduMap.hideUserLocation()`

说明:

隐藏用户位置的标注，关闭持续定位

参数:

无

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 2

## addMark

向地图添加标注

`uexBaiduMap.addMark(markInfo)`

说明:

向地图添加标注

参数:

**markInfo:** (*String 类型*) 必选 添加到地图上的标注信息的集合。该字符串为 Json 格式，如下:

```
{“markInfo”:[{“id”:”111”,”longitude”:”116.232323”,”latitude”:”39.021514”,”imageUrl”:”http:www.xxx.xxx.jpg”,”imageWidth”:”30”,”imageHeight”:”30”,”message”:”这里是新中关村大厦”,”title”:”我的位置”},
{“id”:”112”,”longitude”:”114.232323”,”latitude”:”38.021514”,”imageUrl”:”http:www.xxx.xxx.jpg”,”imageWidth”:”30”,”imageHeight”:”30”,”message”:”这里是随机定的位置”,”title”:”我的位置”}], “canShowCallout”:”0”}
```



各字段含义如下：

参数	是否必须	说明
id	是	唯一标识符
longitude	是	经度
latitude	是	纬度
imageUrl	是	图片路径，路径协议详见 CONSTANT 中 Path Types
imageWidth	是	图片显示在地图上的宽度
imageHeight	是	图片显示在地图上的高度
message	是	点击标注后提示的内容
title	是	点击标注后提示的标题
canShowCallout	是	标注样式, 0-系统默认, 1-左边带图片浅灰色, 2-浅灰色, 3-黑色

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 3

## addMarkNewStyle

向地图添加标注

uexBaiduMap.addMarkNewStyle(markInfo)

说明：

向地图添加标注

参数：

inMarkInfo: (*String* 类型) 必选 添加到地图上的标注信息的集合。该字符串为 Json 格式,如下：

```
{“background”:””,“fontcolor”:”#a2313122”,“markInfo”:[{“id”:”111”,“longitude”:”116.232323”,
”latitude”:”39.021514”,“imageWidth”:”30”,“imageHeight”:”30”,title”测试 1”, ”star”:”1”},
{“id”:”112”,“longitude”:”116.232321”,“latitude”:”39.021514”,“imageWidth”:”30”,“imageHeight”:”30”,title”测试 2”, ”star”:”4”}]}}
```

各字段含义如下：

参数	是否必须	说明
id	是	唯一标识符
longitude	是	经度
latitude	是	纬度
imageWidth	是	标注显示在地图上的宽度（建议 iOS 宽度 84）
imageHeight	是	标注显示在地图上的高度（建议 iOS 高度 64）
title	iOS(否)/Android(是)	标题

star	是	星星数量，取值 0-4
background	是	背景图片路径，支持网络和本地图片，Android 需要 9path 格式的图片
fontcolor	是	字体颜色值

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 3

## updateMark

### 更新标注

uexBaiduMap.updateMark(updateInfo)

说明：

更新标注

参数：

updateInfo: (*String 类型*)必选 标注点的更新信息。该字符串为 Json 格式，如下：

Android:

```
{"id":"111","longitude":"116.404","latitude":"39.415","imageUrl":"http://img.cnbeta.com/topics/apple.png","imageWidth":"100","imageHeight":"100","message":"this is beijing"};
```

iOS:

```
{"id":"112","imageUrl":"res://kafjld.png","timer":"3","imageWidth":"80","imageHeight":"80","message":"这里是新中关村大厦","title":"我的位置","canShowCallout":"0","property":[{"longitude":"121.467803","latitude":"31.249531"}, {"longitude":"121.357419","latitude":"31.375897"}]};
```

各字段含义如下：

参数	是否必须	说明
id	是	唯一标识符
longitude	iOS(否)/Android(是)	经度
latitude	iOS(否)/Android(是)	纬度
imageUrl	是	图片路径，路径协议详见 CONSTANT 中 Path Types
timer	是	更新频率，单位是秒
imageWidth	是	宽度
imageHeight	是	高度
message	是	点击标注后提示的内容
title	iOS(是)/Android(否)	点击标注后提示的标题

---

canShowCallout	iOS(是)/Android(否)	是否使用系统的气泡，0-使用， 1-不使用
property	iOS(是)/Android(否)	标注更新时动画效果从旧的位置到新的位置途经的点集合

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 3

## clearMarks

清除添加的标注

uexBaiduMap.clearMarks(markIdSet)

说明：

清除添加的标注

参数：

markIdSet: (*Array 类型*) 必选 要清除的标注的 id 数组，当该参数缺省时，默认全部清除。

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 3

## addLineOverLayer

添加线形覆盖物

uexBaiduMap.addLineOverLayer(jsonStr)

说明：

添加线形覆盖物

参数：

jsonStr: (*String 类型*) 必选 线形覆盖物的数据，JSON 格式，如下：

```
{ "id": "150", "fillColor": "#FF0000", "strokeColor": "#990033", "lineWidth": "4", "property": [ { "longitude": "116", "latitude": "39" }, { "longitude": "115", "latitude": "38" }, { "longitude": "117", "latitude": "38" }, { "longitude": "120", "latitude": "39" } ] }
```

各字段含义如下:

参数	是否必须	说明
id	是	唯一标识符
property	是	线的连接点的经纬度集合
longitude	是	经度
latitude	是	纬度
fillColor	是	填充颜色
strokeColor	是	画笔颜色
lineWidth	是	画笔宽度, 单位为像素

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 3

## addCircleOverLayer

添加圆形覆盖物

uexBaiduMap.addCircleOverLayer(jsonStr)

说明:

添加圆形覆盖物

参数:

jsonStr: (*String* 类型) 必选 圆形覆盖物的数据, JSON 格式如下:

```
{ "id": "151", "longitude": "116.404", "latitude": "39.915", "radius": "400",  
  "fillColor": "#4169E1", "strokeColor": "#990033", "lineWidth": "4" }
```

各字段含义如下:

参数	是否必须	说明
id	是	唯一标识符
longitude	是	经度
latitude	是	纬度
fillColor	是	填充颜色
strokeColor	是	画笔颜色
lineWidth	是	画笔宽度, 单位为像素
radius	是	圆的半径范围, 单位为米

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 3

## addPolygonOverLayer

添加多边形覆盖物

uexBaiduMap.addPolygonOverLayer (jsonStr)

说明:

添加多边形覆盖物

参数:

jsonStr: (*String* 类型)必选 多边形形覆盖物的数据, JSON 格式如下:

```
{"id":"152","fillColor":"#990033","strokeColor":"#990033","lineWidth":"2.0","property":[{"long
itude":"116","latitude":"39"}, {"longitude":"115","latitude":"38"}, {"longitude":"117","latitude":"3
8"}, {"longitude":"120","latitude":"39"}]}
```

各字段含义如下:

参数	是否必须	说明
id	是	唯一标识符
fillColor	是	填充颜色
strokeColor	是	画笔颜色
lineWidth	是	画笔宽度, 单位为像素
property	是	多边形顶点的经纬度集合

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 3

## clearOverLayers

清除指定的几何覆盖物

uexBaiduMap.clearOverLayers(ids)

说明:

清除指定的几何覆盖物

参数:

**ids:** (*Array 类型*) 可选 被删除的 **OverLayer** 的 Id 数组，不传参数将删除全部覆盖物

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 3

## **addAreaMark**

添加覆盖一块指定区域的标注

`uexBaiduMap.addAreaMark(markInfo)`

说明:

添加覆盖一块指定区域的标注

参数:

**markInfo:** (*String 类型*) 必选 区域标注的信息，JSON 格式为:  
{“markInfo”:[{“id”:”111”,”ltLongitude”:”116.232323”,”ltLatitude”:”39.021514”,  
”rbLongtiude”:”118.345”,”rbLatitude”:”40.567”,”imageUrl”:”http:www.xxx.xxx.jpg” }]}  
各字段含义如下:

参数	是否必须	说明
id	是	唯一标识符
ltLongitude	是	区域左上角的经度
ltLatitude	是	区域左上角的纬度
rbLongitude	是	区域右下角的经度
rbLatitude	是	区域右下角的纬度
imageUrl	是	标注的图片地址，支持网络和本地图片

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 3

## updateAreaMark

刷新指定 ID 的区域标注数据

uexBaiduMap.updateAreaMark(markInfo)

说明：

刷新指定 ID 的区域标注数据

参数：

markInfo: (*String* 类型) 必选 区域标注的更新信息。JSON 格式为：

iOS 格式：

```
'{"id": "111", "timer": "5", "property": [{"ltLongitude": "116.3333", "ltLatitude": "39.6666",  
"rbLongitude": "118.1111", "rbLatitude": "40.00", "imageUrl": "res//:0.png"}, .....]}'
```

各字段含义如下：

参数	是否必须	说明
id	是	唯一标识符
property	是	执行更新时，动画轨迹的区域顶点集
ltLongitude	是	区域左上角的经度
ltLatitude	是	区域左上角的纬度
rbLongitude	是	区域右下角的经度
rbLatitude	是	区域右下角的纬度
imageUrl	是	标注的图片地址，支持网络和本地图片
timer	是	更新的频率，单位为 s

Android 格式：

```
{“id”:”111”,”ltLongitude”:”116.232323”,”ltLatitude”:”39.021514”,”rbLongtiude”:”118.345”,  
”rbLatitude”:”40.567”,”imageUrl”:”http:www.xxx.xxx.jpg” }
```

各字段含义如下：

参数	是否必须	说明
id	是	唯一标识符
ltLongitude	是	区域左上角的经度
ltLatitude	是	区域左上角的纬度
rbLongitude	是	区域右下角的经度
rbLatitude	是	区域右下角的纬度
imageUrl	是	标注的图片地址，支持网络和本地图片

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例:

见附录 3

## showBubbleView1

弹出样式 1 的气泡框

`uexBaiduMap.showBubbleView1 (id, imgUrl, title, content)`

说明:

弹出悬浮窗口，左边带图片，右边两行文字，背景为浅灰色

参数:

**id:** *(String 类型)必选* 唯一标识符,必须与 `addMark` 中的 `id` 相同

**imgUrl :** *(String 类型)必选* 图片链接地址，支持网络和本地图片

**Title:** *(String 类型)必选* 第一栏文字，标题

**Content:** *(String 类型)必选* 第二栏文字，内容

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 4

## showBubbleView2

弹出样式 2 的气泡框

`uexBaiduMap.showBubbleView2 (id, title, content)`

说明:

弹出悬浮窗口，两行文字，背景为浅灰色

参数:

**id:** *(String 类型)必选* 唯一标识符,必须与 `addMark` 中的 `id` 相同

**title:** *(String 类型)必选* 第一栏文字，标题。

**content:** *(String 类型)必选* 第二栏文字，内容

平台支持:

Android 2.2+

iOS 4.3+

版本支持:



3.0.0+

示例:

见附录 4

### showBubbleView3

弹出样式 3 的气泡框

`uexBaiduMap.showBubbleView3(id, title, content)`

说明:

弹出悬浮窗口，两行文字，背景为深黑色

参数:

**id:** *(String 类型)*必选 唯一标识符,必须与 addMark 中的 id 相同

**title:** *(String 类型)*必选 第一栏文字，标题

**content:** *(String 类型)*必选 第二栏文字，内容

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 4

### showBubbleView4

弹出样式 4 的气泡框

`uexBaiduMap.showBubbleView4(id,URL,title,content,BackgroundNor, backgroundClicked, titlefontcolor, contentfontcolor)`

说明:

弹出悬浮窗口，两行文字，背景为自定义

参数:

**id:** *(String 类型)*必选 唯一标识符,必须与 addMark 中的 id 相同

**URL:** *(String 类型)*必选 弹出气泡框中的头像

**title:** *(String 类型)*必选 第一栏文字，标题

**content:** *(String 类型)*必选 第二栏文字，内容

**backgroundNor:** *(String 类型)*必选 背景图片地址，Android 使用的 9patch 样式图片，iOS 使用的悬浮左窗口图片

**backgroundClicked:** (*String 类型*)必选 点击时背景图片地址，Android 使用的 9patch 样式图片，iOS 使用的悬浮左窗口图片

**titlefontcolor:** (*String 类型*)必选 标题字体的颜色

**contentfontcolor:** (*String 类型*)必选 内容字体的颜色

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 4

## **poiSearchInCity**

根据单个关键字在指定城市内搜索兴趣点

`uexBaiduMap.poiSearchInCity(city, key, pageIndex)`

说明:

根据单个关键字在指定城市内搜索兴趣点

参数:

**city:** (*String 类型*)必选 城市名称，比如“北京”

**key :** (*String 类型*)必选 兴趣点关键字，比如“餐厅”

**pageIndex :** (*Number 类型*)必选 兴趣点集合的页索引

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 5

## **poiSearchNearBy**

根据单个关键字在指定的中心点和半径范围内搜索兴趣点

`uexBaiduMap.poiSearchNearBy(key, longitude, latitude, radius, pageIndex)`

说明:

根据单个关键字在指定的中心点和半径范围内搜索兴趣点

参数:

key : (*String 类型*)必选 兴趣点关键字, 比如”餐厅”, ”酒店”

longitude : (*String 类型*)必选 中心点经度

latitude : (*String 类型*)必选 中心点纬度

radius : (*Number 类型*)必选 搜索范围半径, 单位米

pageIndex : (*Number 类型*)必选 兴趣点集合的页索引

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 5

## poiMultiSearchNearBy

根据多个关键字在指定的中心点和半径范围内搜索兴趣点

uexBaiduMap.poiMultiSearchNearBy(key, longitude, latitude, radius, pageIndex)

说明:

根据多个关键字在指定的中心点和半径范围内搜索兴趣点

参数:

Android 平台:

key: (*Array 类型*)必选 兴趣点关键字集合, 比如”餐厅”, ”酒店”, 必须大于 1 个且小于 10 个。

longitude : (*Number 类型*)必选 中心点经度

latitude: (*Number 类型*)必选 中心点纬度

radius : (*Number 类型*)必选 搜索范围半径, 单位米

pageIndex : (*Number 类型*)必选 兴趣点集合的页索引

iOS 平台:

使用 JSON 字符串来表示参数, var jssonstr =

'{"pageIndex": "0", "longitude": "116.404", "latitude": "39.915", "radius": "1000", "list": ["酒店", "餐厅"]}' ;各 key 的值参见 Android 的参数解释。

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 5

## poiSearchInBounds

根据单个关键字在指定的矩形区域内搜索兴趣点

uexBaiduMap.poiSearchInBounds(key, ltLongitude, ltLatitude, rbLongitude, rbLatitude, pageIndex)

说明:

根据单个关键字在指定的矩形区域内搜索兴趣点

参数:

Android 平台:

key : (*String 类型*)必选 兴趣点关键字, 比如”餐厅”, ”酒店”

ltLongitude : (*Number 类型*)必选 矩形区域左上角经度

ltLatitude: (*Number 类型*)必选 矩形区域左上角纬度

rbLongitude: (*Number 类型*)必选 矩形区域右下角经度

rbLatitude: (*Number 类型*)必选 矩形区域右下角纬度

pageIndex : (*Number 类型*)必选 兴趣点集合的页索引

iOS 平台:

使用 JSON 字符串来表示参数,格式如下:

```
var jssonstr = '{"pageIndex": "0", "ltLongitude": "116.404", "ltLatitude": "39.915",  
                "rbLongitude": "116.408", "rbLatitude": "39.920", "key": "餐厅"}';
```

JSON 中各 key 的值参见 Android 平台的参数解释。

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 5

## poiMultiSearchInBounds

根据多个关键字在指定的经纬度起始点内搜索兴趣点

uexBaiduMap.poiMultiSearchInBounds(key, ltLongitude, ltLatitude, rbLongitude, rbLatitude, pageIndex)

说明：

根据多个关键字在指定的经纬度起始点内搜索兴趣点

参数：

Android 平台：

key: (*Array* 类型) 必选 兴趣点关键字集合，比如”餐厅”，”酒店”，必须大于 1 个且小于 10 个。

ltLongitude: (*Number* 类型) 必选 矩形区域左上角经度

ltLatitude: (*Number* 类型) 必选 矩形区域左上角纬度

rbLongitude: (*Number* 类型) 必选 矩形区域右下角经度

rbLatitude: (*Number* 类型) 必选 矩形区域右下角纬度

pageIndex: (*Number* 类型) 必选 兴趣点集合的页索引

iOS 平台：

使用 JSON 字符串来表示参数, var jssonstr =

```
'{"pageIndex": "0", "ltLongitude": "116.404", "ltLatitude": "39.915", "rbLongitude": "116.408",  
"rbLatitude": "39.920", "list": ["餐厅", "电影院"]}';
```

JSON 中各 key 的解释 Android 平台参数的解释。

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 5

## suggestionSearch

### 联想词检索

uexBaiduMap.suggestionSearch(key)

说明：

联想词检索，一般用于根据已输入字进行实时检索，实现输入关键字时的自动提示

参数：

key: (*String* 类型) 必选 搜索关键字，比如”餐厅”，”酒店”

平台支持：

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 5

## busLineSearch

公交线路搜索

uexBaiduMap.busLineSearch(city, lineUid, id)

说明:

根据公交线路编号在指定城市范围内搜索公交线路详情

参数:

city : (*String* 类型) 必选 城市名称, 比如: “北京”

lineUid: (*String* 类型) 必选 公交线路编号, 比如: ”715”

id: (*String* 类型) *iOS* 必选, *Android* 无 用户自定义的搜索出来路线的唯一标识

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 5

## showRoutePlan

规划并显示路线

uexBaiduMap.showRoutePlan(json)

说明:

由于需要根据参数搜索和规划线路, 显示会有延迟, 有结果之后会显示路线 然后回调 `callback` 方法。注意在传递搜索的参数时, 一个地点的信息可以用 `city` 和 `name` 结合模糊描述, 也可以用 `longitude` 和 `latitude` 结合精确描述, 在 `start` 和 `end` 里面可以出现任意一种结合方式, 但是两种位置不能同时出现 `city` 和 `name` 结合的描述方式。

参数:

json: (*String* 类型) 必选 规划路线所需要的信息。JSON 格式为:

```
{“id”:”rp345”, “type”:”1”, “start”: {“city”:”北京”, “name”:”中关村”, “longitude”:”39.123456”, “latitude”:”112.564325”}, “end”: {“longitude”:”39.123456”, “latitude”:”112.564325”}}
```

各字段含义如下:

---

参数	是否必须	说明
id	是	唯一标识符
type	是	路线规划的类型：0-驾车；1-公交；2-步行
start	是	起始点的相关信息，JSON 格式，如上示例
end	是	终点的相关信息，JSON 格式，如上示例
city	是	指定点所在的城市，必须与 name 合用，比如“北京”
name	是	指定点的详细地址 比如“中关村大街 XX 号”
longitude	是	指定点的经度
latitude	是	指定点的纬度

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 6

## clearRoutePlan

清除路线规划覆盖物

uexBaiduMap.clearRoutePlan(ids)

说明：

清除路线规划覆盖物

参数：

ids: (*Array 类型*) 可选      需要被删除路线的 id 集合，缺省将删除全部路线

平台支持：

Android 2.2+

iOS 4.3+

版本支持：

3.0.0+

示例：

见附录 6

## geocode

根据地名获取地址经纬度信息

uexBaiduMap.geocode(city, address)

说明：

根据地名获取地址经纬度信息

参数:

city: (*String* 类型) 必选 所在城市, 比如“北京”

address: (*String* 类型) 必选 详细地址, 比如“海淀区中关村大街”

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 7

## reverseGeocode

根据经纬度获取地址详细信息

uexBaiduMap.reverseGeocode(longitude, latitude)

说明:

根据经纬度获取地址详细信息

参数:

longitude: (*Number* 类型) 必选 地址经度

latitude: (*Number* 类型) 必选 地址纬度

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 7

## getBaiduFromGoogle

Google 地图经纬度转换为百度地图经纬度

uexBaiduMap.getBaiduFromGoogle(longitude, latitude)

说明:

Google 地图经纬度转换为百度地图经纬度

参数:

longitude: (*Number* 类型) 必选 经度



latitude: (*Number 类型*) 必选      纬度

平台支持:

Android 2.2+

iOS 4.3+

版本支持:

3.0.0+

示例:

见附录 7

## cbGetCurrentLocation

获取到当前位置时的回调方法

uexBaiduMap.cbGetCurrentLocation(opId, dataType, data)

参数:

opId: (*Number 类型*) 必选    操作 ID, 在此方法中不起作用, 可忽略

dataType: (*Number 类型*) 必选    返回 uex.cJson 类型, 详见 CONSTANT 中 Callback Data Type

data: (*String 类型*) 必选    当前位置的经纬度, 数据为 JSON 格式:

```
{“longitude”:” 33.567”,“latitude”:”67.976”}
```

各字段含义如下:

参数	是否必须	说明
longitude	是	经度
latitude	是	纬度

版本支持:

3.0.0+

## cbShowUserLocation

showUserLocation 的回调方法, 成功显示当前位置时被调用。

uexBaiduMap.cbShowUserLocation(opId, dataType, data)

参数:

opId: (*Number 类型*) 必选    操作 ID, 在此方法中不起作用, 可忽略

dataType: (*Number 类型*) 必选    返回 uex.cJson 类型, 详见 CONSTANT 中 Callback Data Type

data: (*String 类型*) 必选    当前位置的经纬度, 数据为 JSON 格式:

```
{“longitude”:” 33.567”,“latitude”:”67.976”}
```

各字段含义如下:

---

参数	是否必须	说明
longitude	是	经度
latitude	是	纬度

版本支持:

3.0.0+

## cbFastGetLocation

**fastGetLocation** 的回调方法，获取到快速定位时被回调

uexBaiduMap.cbFastGetLocation(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 操作 ID，在此方法中不起作用，可忽略

dataType: (*Number 类型*)必选 返回 uex.cJson 类型，详见 CONSTANT 中 Callback Data Type

data: (*String 类型*)必选 当前位置的经纬度，数据为 JSON 格式:

```
{“longitude”:” 33.567”,”latitude”:”67.976”}
```

各字段含义如下:

参数	是否必须	说明
longitude	是	经度
latitude	是	纬度

版本支持:

3.0.0+

## cbBaiduFromGoogle

**getBaiduFromGoogle** 的回调方法，谷歌经纬度转换为百度经纬度时回调

uexBaiduMap.cbBaiduFromGoogle(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 操作 ID，在此方法中不起作用，可忽略

dataType: (*Number 类型*)必选 返回 uex.cJson 类型，详见 CONSTANT 中 Callback Data Type

data: (*String 类型*)必选 转换之后的经纬度，数据为 JSON 格式:

```
{“longitude”:” 33.567”,”latitude”:”67.976”}
```

各字段含义如下:

参数	是否必须	说明
longitude	是	经度
latitude	是	纬度

版本支持:

3.0.0+

## cbSuggestionSearch

**suggestionSearch** 的回调函数，搜索到联想的关键字列表时被调用

uexBaiduMap.cbSuggestionSearch(opId, dataType, data)

参数：

opId: (*Number 类型*)必选 操作 ID，在此方法中不起作用，可忽略

dataType: (*Number 类型*)必选 返回 uex.cJson 类型, 详见 CONSTANT 中 Callback Data

Type

data: (*String 类型*)必选 联想的关键字列表，数据为 JSON 格式：

```
{"list":[{"city":"北京市","key":"酒店用品批发部"}]}
```

各字段含义如下：

参数	是否必须	说明
list	是	联想的关键字列表
city	是	搜索结果城市
key	是	搜索结果关键字

版本支持：

3.0.0+

## cbPoiSearchInCity

**poiSearchInCity** 的回调函数，搜索到兴趣点时被调用

uexBaiduMap.cbPoiSearchInCity(opId, dataType, data)

参数：

opId: (*Number 类型*)必选 操作 ID，在此方法中不起作用，可忽略

dataType: (*Number 类型*)必选 返回 uex.cJson 类型, 详见 CONSTANT 中 Callback Data

Type

data: (*String 类型*)必选 搜索出来的 POI 列表信息，数据为 JSON 格式：

```
{"list":[{"longitude":"116.403030","latitude":"40.005403","name":"CGV 星星国际影城(北京奥体店)", "uid":"****", "adress":"地址", "city":"*****", "phone":"*****", "postCode":"*****", "ePoiType":"***"}],
```

```
"totalPoiNum":"302","pageNum":"31","currPoiNum":"10","pageIndex":"0"}
```

各字段含义如下：

参数	是否必须	说明
list	是	搜索出来的 POI 信息的集合
longitude	是	经度
latitude	是	纬度
name	是	名称
uid	是	唯一标识符
address	是	地址

---

city	是	所在城市
phone	是	电话
postcode	是	邮编
epoitype	是	类型 (///POI 类型, 0:普通点 1:公交站 2:公交线路 3:地铁站 4:地铁线路 )
totalPoiNum	是	本次 POI 搜索的总结果数
pageNum	是	本次 POI 搜索的总页数
currPoiNum	是	当前页的 poi 结果数
pageIndex	是	当前页

版本支持:

3.0.0+

## cbPoiSearchArea

**poiSearchNearby** 和 **poiSearchInBounds** 的回调，搜索到兴趣点时被调用  
 uexBaiduMap.cbPoiSearchArea(opId, dataType, data)

参数:

同 cbPoiSearchInCity。

版本支持:

3.0.0+

## cbPoiMultiSearchArea

**poiMultiSearchNearBy** 和 **poiMultiSearchInBounds** 的回调函数，搜索到兴趣点时被调用

uexBaiduMap.cbPoiMultiSearchArea(opId, dataType, data)

参数:

同 cbPoiSearchInCity。

版本支持:

3.0.0+

## cbGeocode

**geocode** 的回调，编码成功时被调用

uexBaiduMap.cbGeocode(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 操作 ID，在此方法中不起作用，可忽略

dataType: (*Number 类型*)必选 返回 uex.cJson 类型，详见 CONSTANT 中 Callback Data Type

data: (*String 类型*)必选 编码出来的点经纬度，数据为 JSON 格式:

```
{“longitude”:” 33.567”,”latitude”:”67.976”}
```

各字段含义如下:

参数	是否必须	说明
longitude	是	编码出来的经度
latitude	是	编码出来的的纬度

版本支持:

3.0.0+

## cbReverseGeocode

**reverseGeocode** 的回调，逆地理编码时被调用

uexBaiduMap.cbReverseGeocode(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 操作 ID，在此方法中不起作用，可忽略

dataType: (*Number 类型*)必选 返回 uex.cJson 类型，详见 CONSTANT 中 Callback Data

Type

data: (*String 类型*)必选 逆编码出来点的信息，数据为 JSON 格式:

```
{"streetNumber": "甲 10 号", "address": "北京市东城区中华路甲 10 号", "district": "东城区", "city": "北京市", "longitude": "116.403982", "latitude": "39.915099", "province": "北京市", "streetName": "344\270\255 华路"}
```

各字段含义如下:

参数	是否必须	说明
streetNumber	是	反编译出的地点的街道编号
address	是	反编译出的地点的地址
district	是	反编译出的地点的区县名称
city	是	反编译出的地点的城市名称
longitude	是	反编译出的地点的经度
latitude	是	反编译出的地点的纬度
province	是	反编译出的地点的省份名称
streetName	是	反编译出的地点的街道名称

版本支持:

3.0.0+

## cbBusLineSearch

**busLineSearch** 的回调，搜索到公交线路时被调用

uexBaiduMap.cbBusLineSearch(opId, dataType, data)

参数:

opId: (*Number 类型*)必选 操作 ID，在此方法中不起作用，可忽略

**dataType:** (*Number 类型*)必选 返回 uex.cJson 类型, 详见 CONSTANT 中 Callback Data Type

**data:** (*String 类型*)必选 搜索到的公交路线的详细信息, 数据为 JSON 格式:

```
{"busName": "717 路(菜户营桥东-城铁西二旗站)", "startTime": "05:30", "company": "", "endTime": "22:00", "stepInfo": [{"longitude": "116.310792", "latitude": "40.059805", "title": "城铁西二旗站"}, ...]}
```

各字段含义如下:

参数	是否必须	说明
busName	是	线路名称
company	是	公交线路所属公司
startTime	是	首班时间
endTime	是	末班时间
stepInfo	是	站点信息列表
longitude	是	站点的经度
latitude	是	站点的纬度
title	是	站点的名称

版本支持:

3.0.0+

## cbShowRoutePlan

**showRoutePlan** 的回调, 规划出路线并显示时被调用

uexBaiduMap.cbShowRoutePlan(opId, dataType, data)

参数:

**opId:** (*Number 类型*)必选 操作 ID, 在此方法中不起作用, 可忽略

**dataType:** (*Number 类型*)必选 返回 uex.cJson 类型, 详见 CONSTANT 中 Callback Data Type

**data:** (*String 类型*)必选 路线显示的信息, 数据为 JSON 格式:

```
{"id": "2345", "result": "0"}
```

各字段含义如下:

参数	是否必须	说明
id	是	showRoutePlan 时传入的 id, 用来标识此 RoutePlan
result	是	显示结果, 0-成功, 1-失败

版本支持:

3.0.0+

## onTouchMark

点击地图上的标注的监听方法

uexBaiduMap.onTouchMark(overlayId, lat, lon)

参数:

overlayId: (*String* 类型) 必选    被点击的标注的 ID  
lat: (*Number* 类型) 必选    被点击的标注的纬度  
lon: (*Number* 类型 ) 必选    被点击的标注的经度

版本支持:

3.0.0+

## onTouchBubbleView

点击气泡框的监听方法

uexBaiduMap.onTouchBubbleView(id)

参数:

id: (*String* 类型) 必选    被点击气泡框所属的标注的 ID

版本支持:

3.0.0+

## onMapDrag

拖动地图时的监听方法，从拖动开始到结束一直会被调用

uexBaiduMap.onMapDrag(state, lat, lon)

参数:

state: (*Number* 类型) 必选    地图当前状态，0-开始拖动，2-停止拖动，1-正在拖动  
lat: (*Number* 类型) 必选    地图当前中心点的纬度  
lon: (*Number* 类型 ) 必选    地图当前中心点的经度

版本支持:

3.0.0+

## onNetworkError

初始化地图时，网络错误的监听方法

uexBaiduMap.onNetworkError()

参数:

无

版本支持:

3.0.0+

## onPermissionDenied

初始化地图时，key 非法时的监听方法

uexBaiduMap.onPermissionDenied()

参数：

无

版本支持：

3.0.0+

## 附录 1

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width,
initial-scale=1.0, user-scalable=no" />
```

```
    <link rel="stylesheet" type="text/css" href="../css/index.css">
```

```
    <title>百度地图功能</title>
```

```
    <style>
```

```
      input {
        color: black;
      }
```

```
    </style>
```

```
    <script type="text/javascript">
```

```
      打开百度地图
```

```
      function openmap(){
```

```
        uexBaiduMap.open('6643147DF56BD40F6140AB3ABA5A3FFBE12C439F',
                          0, 400, 800, 800, '116.401', '39.925');
```

```
      }
```

```
      关闭百度地图
```

```
      function clean(){
```

```
        uexBaiduMap.clean();
```



```
}
```

隐藏地图

```
function goBack(){  
    uexBaiduMap.hide();  
}
```

显示地图

```
function goFront(){  
    uexBaiduMap.show();  
}
```

设置地图的中心点经纬度

```
function setCenter(){  
    uexBaiduMap.setCenter("116.401", "39.925");'116.401','39.925'是天安门的  
    实际经纬度  
}
```

设置地图的缩放率(范围 3---18)

```
function setZoomLevel(){  
    var t = document.getElementById("mapZoom").value;  
    uexBaiduMap.setZoomLevel("3");  
}
```

获得地图的显示类型

```
function setType(){  
    /* type 为 0（标准地图），1（卫星地图），2（打开实时路况），3（关闭实  
    时路况）*/  
    var t = document.getElementById("type").value;  
    uexBaiduMap.setType(t);  
}
```

缩放指定经纬度跨度

```
function zoomToSpan(){  
    uexBaiduMap.zoomToSpan("0.05", "0.05");  
}
```

放大一个地图级别

```
function zoomIn(){  
    uexBaiduMap.zoomIn();
```

```
}
```

缩小一个地图级别

```
function zoomOut(){
```

```
    uexBaiduMap.zoomOut();
```

```
}
```

```
window.uexOnload = function(){
```

```
    uexWidgetOne.cbError = function(opid, type, data){
```

```
        alert(data);
```

```
    }
```

当手动移动百度地图时的通知

```
uexBaiduMap.onMapDrag = function(state){
```

```
    document.getElementById('dragState').innerHTML = "onMapDrag State: " + state;
```

```
};
```

apiKey 授权失败，回调方法

```
uexBaiduMap.onPermissionDenied = function(){
```

```
    alert(onPermissionDenied);
```

```
}
```

网络出问题时的回调

```
uexBaiduMap.onNetworkError = function(){
```

```
    alert(onNetworkError);
```

```
}
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<div class="tit">
```

百度地图功能

```
</div>
```

```
<div class="conbor">
```

```
<div class="consj">
```

```
<br/>
```

```
<input class="btn" type="button" value="打开地图" onclick="openmap();">
```

```
<input class="btn" type="button" value="关闭" onclick="clean();">
```

```
<input class="btn" type="button" value="隐藏 map" onclick="goBack();">
```

```
<input class="btn" type="button" value="显示 map" onclick ="goFront();">
<input class="btn" type="button" value="设置地图类型为: " onclick="setType();">
<input type="text" id="type" value="1"/>
<br/>
type:type 为 0（标准地图）,1（卫星地图），2（打开实时路况），3（关闭实时路况）
<br/>
<input class="btn" type="button" value="放大一级别" onclick ="zoomIn();">
<input class="btn" type="button" value="缩小一级别" onclick ="zoomOut();">
<br/>
<input class="btn" type="button" value="设置缩放级别为" onclick ="setZoomLevel();">
<input type="text" id="mapZoom" value="10">
<br/>
<input class="btn" type="button" value="设置地图中心点为天安门"
      onclick ="setCenter();">
<br/>
<input class="btn" type="button" value="缩放到指定经纬跨度"
      onclick ="zoomToSpan();">
</div>
</div>
</body>
</html>
```

## 附录 2

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0,
user-scalable=no" />
    <link rel="stylesheet" type="text/css" href="../css/index.css">
    <title>百度地图功能</title>
    <style>
      input {
        color: black
      }
    </style>
    <script type="text/javascript">
```

获得当前位置的经纬度

```
function getLocation(){  
    uexBaiduMap.getCurrentLocation();  
}
```

显示用户位置并添加代表用户位置的标注，并会持续定位用户位置

```
function showUserLocation(){  
    uexBaiduMap.showUserLocation();  
}
```

隐藏标识用户位置信息的标注，关闭持续定位用户位置

```
function hideUserLocation(){  
    uexBaiduMap.hideUserLocation();  
}
```

```
window.uexOnload = function(){  
    uexWidgetOne.cbError = function(opid, type, data){  
        alert(data);  
    }  
    将获取的当前位置数据返回  
    uexBaiduMap.cbGetCurrentLocation = function(i, t, data){  
        alert(data);  
    };  
    openmap();  
}  
function openmap(){  
    uexBaiduMap.open('6643147DF56BD40F6140AB3ABA5A3FFBE12C439F',  
        0,300,800,800, '116.401', '39.925');  
}
```

</script>

</head>

<body>

<div class="tit">

百度地图位置功能

</div>

<div class="conbor">

<input class="btn" type="button" value="获得当前位置" onclick="getLocation();">

<input class="btn" type="button" value="显示用户当前位置的经纬度" onclick="showUserLocation();">

<input class="btn" type="button" value="隐藏用户当前位置的经纬度" onclick="hideUserLocation();">

</div>

</body>

</html>

## 附录 3

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0, user-scalable=no" />
```

```
    <link rel="stylesheet" type="text/css" href="../css/index.css">
```

```
    <title>百度地图功能</title>
```

```
    <style>
```

```
      input {
```

```
        color: black
```

```
      }
```

```
    </style>
```

```
    <script type="text/javascript">
```

```
      添加标注
```

```
      function addMark(){
```

```
        var jssonstr =
```

```
'{"markInfo":[{"id":"111","longitude":"116.404","latitude":"39.915","imageUrl":"http://www.cmkjz.com/uploads/allimg/100116/0134423.jpg","imageWidth":"60.0","imageHeight":"60.0","message":"海淀区中关村大街 15 号中关村 15 号中关村 15 号中关村","title":"我的
```

```
"},{id":"112","longitude":"116.450","latitude":"39.989","imageUrl":"http://82238.com/uploads/allimg/110519/2-110519130404.jpg","imageWidth":"60.0","imageHeight":"60.0","message":"this is beijing","title":"我的位置
```

```
"},{id":"113","longitude":"116.351","latitude":"39.854","imageUrl":"http://www.qqhead.com/UploadFiles/2010-04/2010458740103062.gif","imageWidth":"30.0","imageHeight":"30.0","message":"this is shanghai","title":"我的位置
```



```
        uexBaiduMap.addMark(jssonstr);
```

```
      }
```

```
      清除标注
```

```
      function clearMarks(){
```

```
        var idarr = ["111", "112", "113", "114"];
```

```
        uexBaiduMap.clearMarks(idarr);
```

```
      }
```

添加覆盖物，直线

```
function addLineOverLayer(){
    var jssonstr =
'{"id":"150","fillColor":"#990033","strokeColor":"#990033","lineWidth":"2.0","property":[{"longitude":"116.23232
3","latitude":"39.021514"},{"longitude":"118.345","latitude":"40.567"}]}';, {"longitude":"117","latitude":"38"}, {"lo
ngitude":"120","latitude":"39"}

    uexBaiduMap.addLineOverLayer(jssonstr);
}
```

向百度地图添加覆盖物，圆形

```
function addCircleOverLayer(){
    var jssonstr =
'{"id":"151","longitude":"116.404","latitude":"39.915","radius":"5000","fillColor":"#4169E1","strokeColor":"#9900
33","lineWidth":"2.0"}';

    uexBaiduMap.addCircleOverLayer(jssonstr);
}
```

```
function addMultiOverLayer(){
    var jssonstr =
'{"id":"152","fillColor":"#990033","strokeColor":"#990033","lineWidth":"2.0","property":[{"longitude":"116","latit
ude":"39"}, {"longitude":"115","latitude":"38"}, {"longitude":"117","latitude":"38"}, {"longitude":"120","latitude":"3
9"}]}';
```

添加多边形覆盖物

```
uexBaiduMap.addPolygonOverLayer(jssonstr);
}
```

清除指定的几何覆盖物

```
function clearOverLayer(type){
    var idarr = [type];

    uexBaiduMap.clearOverLayers(idarr);
}
```

清除所有几何覆盖物

```
function clearOverLayers(){
    uexBaiduMap.clearOverLayers();
}

function openmap(){
```

```
uexBaiduMap.open('6643147DF56BD40F6140AB3ABA5A3FFBE12C439F',  
0,300,800,800, '116.401', '39.925');  
}
```

```
window.uexOnload = function() {  
    uexWidgetOne.cbError = function(opid, type, data) {  
        alert(data);  
    }  
    openmap();  
}
```

```
function updateMark1() {  
    var jstr =  
    '{"id":"'111','longitude":"'116.404','latitude":"'39.415','imageUrl":"'http://img.cnbeta.com/topics/apple.png  
','imageWidth":"'100','imageHeight":"'100','message":"'this is beij'"}';  
    uexBaiduMap.updateMark(jstr);  
}  
function updateMark2() {  
    var lonlat =  
    '{"id":"'112','imageUrl':'res:kafjld.png','timer':'3','imageWidth':'80','imageHeight':'80','message':'这里是新  
    中关大厦','title':'我的位置','canShowCallout':'1',  
    'property':[{ 'longitude':'121.467803','latitude':'31.249531'},  
    { 'longitude':'121.357419','latitude':'31.375897'},  
    { 'longitude':'116.905273','latitude':'39.264714'},  
    { 'longitude':'116.886876','latitude':'39.350472'},  
    { 'longitude':'116.776492','latitude':'39.535919'},  
    { 'longitude':'116.666108','latitude':'39.706658'},  
    { 'longitude':'116.482135','latitude':'39.848617'}]}'';  
    uexBaiduMap.updateMark(lonlat);  
}  
添加覆盖一块指定区域的标注  
function addAreaMark() {  
    alert();
```

---

```
var ltLo = "-191.993965";
var ltla = "84.109426";
var rbLo = "201.266653";
var rbLa = "-46.314504";
var json = '{"markInfo":[{"id":"114","ltLongitude":"' + ltLo + "','ltLatitude":"' + ltla +
"',rbLongitude":"' + rbLo + "','rbLatitude":"' + rbLa + "','imageUrl":"res:pano_b.jpg"}]}'
uexBaiduMap.addAreaMark(json);
}
```

添加覆盖一块指定区域的标注（for iOS）

```
function addAreaMarkiOS(){
    var ltLo = "-191.993965";
    var ltla = "84.109426";
    var rbLo = "201.266653";
    var rbLa = "-46.314504";
    var json = '{"markInfo":[{"id":"111","ltLongitude":"' + ltLo + "','ltLatitude":"' + ltla +
"',rbLongitude":"' + rbLo + "','rbLatitude":"' + rbLa + "','imageUrl":"res:0.png"}]}'
    uexBaiduMap.addAreaMark(json);
    var jsons = [];
    for (var i = 0; i < 9; i++) {
        var item = {
            "ltLongitude": ltLo,
            "ltLatitude": ltla,
            "rbLongitude": rbLo,
            "rbLatitude": rbLa,
            "imageUrl": "res:" + i + ".png"
        }
        jsons.push(item);
    }
    var json = {
        "id": "111",
        "timer": "3",
        "property": jsons
    }
    var temjson = JSON.stringify(json);
```



```

        uexBaiduMap.updateAreaMark(temjson);
    }
    刷新一个指定 ID 表示的标注的数据
    function updateAreaMark(){
        alert();
        var ltLo = "-191.993965";
        var ltla = "84.109426";
        var rbLo = "201.266653";
        var rbLa = "-46.314504";
        var json = '{"id":"114","ltLongitude":"' + ltLo + "','ltLatitude":"' + ltla + "','rbLongitude":"' +
rbLo + "','rbLatitude":"' + rbLa + "','imageUrl":"res:uexTimeMachine_cunhuo.png"}';
        uexBaiduMap.updateAreaMark(json);
    }
    function updateAreaMarkiOS(){
        var json =
'{"markInfo":[{"id":"112","longitude":"121.467803","latitude":"31.249531","imageUrl":"res:kafjld.png","imageWid
th":"80","imageHeight":"80","message":"这里是新中关村大厦","title":"我的位置"}],"canShowCallout":"0"}';
        uexBaiduMap.addMark(json);
        var lonlat =
'{"id":"112","imageUrl":"res:kafjld.png","timer":"3","imageWidth":"80","imageHeight":"80","message":"这里是新
中关村大厦","title":"我的位置
","canShowCallout":"0","property":[{"longitude":"121.467803","latitude":"31.249531"}, {"longitude":"121.357419",
"latitude":"31.375897"}, {"longitude":"121.302227","latitude":"31.502092"}, {"longitude":"116.776492","latitude":"
39.535919"}, {"longitude":"116.666108","latitude":"39.706658"}, {"longitude":"116.482135","latitude":"39.848617"
}]}';
        uexBaiduMap.updateMark(lonlat);
    }
    清除指定范围内的标注
    function clearAreaMarks(){
        uexBaiduMap.clearAreaMarks();
    }

</script>
</head>

```

```
<body>
  <div class="tit">
    百度地图覆盖物相关功能
  </div>
  <div class="conbor">
    <div class="consj">
      <input class="btn" type="button" value="添加线型覆盖物"
        onclick="addLineOverLayer();">
      <input class="btn" type="button" value="清除线型覆盖物"
        onclick="clearOverLayer('150');">
      <input class="btn" type="button" value="添加圆型覆盖物"
        onclick="addCircleOverLayer();">
      <input class="btn" type="button" value="清除圆型覆盖物"
        onclick="clearOverLayer('151');">
      <input class="btn" type="button" value="添加多边形覆盖物"
        onclick="addMultiOverLayer();">
      <input class="btn" type="button" value="清除多边形覆盖物"
        onclick="clearOverLayer('152');">
      <input class="btn" type="button" value="清除 overlays"
        onclick="clearOverLayers();">
      <input class="btn" type="button" value="添加标注" onclick="addMark();">
      <input class="btn" type="button" value="android 更新标注"
        onclick="updateMark1();">
      <input class="btn" type="button" value="android 添加覆盖指定区域的标注"
        onclick="addAreaMark();">
      <input class="btn" type="button" value="android 更新覆盖指定区域的标注"
        onclick="updateAreaMark();">
      <input class="btn" type="button" value="iOS 更新标注" onclick="updateMark2();">
      <input class="btn" type="button" value="iOS 添加覆盖指定区域的标注"
        onclick="addAreaMarkiOS();">
      <input class="btn" type="button" value="iOS 更新覆盖指定区域的标注"
        onclick="updateAreaMarkiOS();">
      <input class="btn" type="button" value="清除覆盖指定区域的标注"
        onclick="clearAreaMarks();">
```

## 附录 4

## &lt;!DOCTYPE HTML&gt;

```
<html>

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0,
user-scalable=no" />

    <link rel="stylesheet" type="text/css" href="../css/index.css">

    <title>百度地图功能</title>

    <style>

        input {

            color: black

        }

    </style>

    <script type="text/javascript">

        window.uxOnload = function(){

            uexWidgetOne.cbError = function(opid, type, data){

                alert(data);

            }

            openmap();

            uexBaiduMap.onTouchMark = function(mid, longitude, latitude){

                if(id=="111"){

                    uexBaiduMap.showBubbleView1 (id, "res://bubbleicon.png", "我是样式 1 的气泡框",

                                                    "我是标注 111");

                }else if(id=="112"){

                    uexBaiduMap.showBubbleView2 (id, "我是样式 2 的气泡框", "我是标注 112");

                }else if(id=="113"){

                    uexBaiduMap.showBubbleView3(id, "我是样式 3 的气泡框", "我是标注 113");

                }

            }

        }

    </script>

</html>
```

```

    }else if(id=="114"){
        uexBaiduMap.showBubbleView4(id,"res://bubbleicon.png","我是样式 4 的气泡框",
        "我是标注 114","res://backgroundNor" , "res://backgroundClicked", "#001111", "#eeeeee");
    }
};

uexBaiduMap.onTouchBubbleView = function(strId){
    alert(strId);
}
}

function addMarks(){
    var jssonstr = '{"markInfo":[
{"id":"111","longitude":"116.404","latitude":"39.915","imageUrl":"http://www.cmkjz.com/uploads/allimg/100116/0134423.jpg","imageWidth":"60.0","imageHeight":"60.0","message":"中关村海淀中路 15 号","title":"我的位置"},
{"id":"112","longitude":"116.450","latitude":"39.989","imageUrl":"http://82238.com/uploads/allimg/110519/2-110519130404.jpg","imageWidth":"60.0","imageHeight":"60.0","message":"this is beij","title":"我的位置"},
{"id":"113","longitude":"116.351","latitude":"39.854","imageUrl":"http://www.qqhead.com/UploadFiles/2010-04/2010458740103062.gif","imageWidth":"30.0","imageHeight":"30.0","message":"this is shangh","title":"我的位置"}],
{"id":"114","longitude":"115.351","latitude":"38.854","imageUrl":"http://www.qqhead.com/UploadFiles/2010-04/2010458740103062.gif","imageWidth":"30.0","imageHeight":"30.0","message":"随机取的位置","title":"我的位置"}],
"canShowCallout":"0","leftCalloutAccessoryImageUrl":"http://www.cmkjz.com/uploads/allimg/100116/0134423.jpg"}';

    uexBaiduMap.addMark(jssonstr);
}

function openmap(){
    uexBaiduMap.open('6643147DF56BD40F6140AB3ABA5A3FFBE12C439F', 0, 300, 800,800,
'116.401', '39.925');
}
</script>
</head>
<body>
<div class="tit">
    气泡框样式示例
</div>
<div class="conbor">

```

```
<div class="consj">
  <input class="btn" type="button" value="添加四个标注" onclick="addMarks();">
</div>
</div>
</body>
</html>
```

## 附录 5

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0,
user-scalable=no" />
    <link rel="stylesheet" type="text/css" href="../css/index.css">
    <title>百度地图功能</title>
    <style>
      input {
        color: black;
      }
    </style>
    <script type="text/javascript">

      //根据单个关键字在指定城市内搜索兴趣点
      function poiSearchInCity(){
        uexBaiduMap.poiSearchInCity("北京", "电影院", "0");
      }
      //根据单个关键字在指定的中心点和半径范围内搜索兴趣点
      function poiSearchNearBy(){
        uexBaiduMap.poiSearchNearBy("餐厅", "116.404", "39.915", "500", "0");
      }
      //根据多个关键字在指定的中心点和半径范围内搜索兴趣点
      function poiMultiSearchNearBy(){
        uexBaiduMap.poiMultiSearchNearBy(["酒店", "餐厅"], '116.404', '39.915', '1000', '0');
```

```
}

//根据多个关键字在指定的中心点和半径范围内搜索兴趣点（for iOS）
function poiMultiSearchNearByiOS(){
    var jssonstr =
'{"pageIndex":"0","longitude":"116.404","latitude":"39.915","radius":"1000","list":["酒店","餐厅"]};
    uexBaiduMap.poiMultiSearchNearBy(jssonstr);
}

//根据单个关键字在指定的经纬度起始点内搜索兴趣点
function poiSearchInBoundsiOS(){
    var jssonstr =
'{"pageIndex":"0","ltLongitude":"116.404","ltLatitude":"39.915","rbLongitude":"116.408","rbLatitude":"39.920","key":"餐厅"}';
    uexBaiduMap.poiSearchInBounds(jssonstr);
}

//根据多个关键字在指定的经纬度起始点内搜索兴趣点
function poiMultiSearchInBoundsiOS(){
    var jssonstr =
'{"pageIndex":"0","ltLongitude":"116.404","ltLatitude":"39.915","rbLongitude":"116.408","rbLatitude":"39.920","list":["餐厅","电影院"]};
    uexBaiduMap.poiMultiSearchInBounds(jssonstr);
}

//根据单个关键字在指定的经纬度起始点内搜索兴趣点
function poiSearchInBounds(){
    uexBaiduMap.poiSearchInBounds('餐厅', '116.404', '39.915', '116.408', '39.920', '0');
}

//根据多个关键字在指定的经纬度起始点内搜索兴趣点
function poiMultiSearchInBounds(){
    var jssonstr =
'{"pageIndex":"0","ltLongitude":"116.404","ltLatitude":"39.915","rbLongitude":"116.408","rbLatitude":"39.920","list":["餐厅","电影院"]};
    uexBaiduMap.poiMultiSearchInBounds(jssonstr);
}

//联想词检索
function suggestionSearch(){
```

```
        uexBaiduMap.suggestionSearch("酒店");
    }
    //根据公交线路编号在指定城市范围内搜索公交线路详情
    function showBusLineSearch(){
        uexBaiduMap.busLineSearch("北京", "717");
    }
    //根据公交线路编号在指定城市范围内搜索公交线路详情(for iOS)
    function showBusLineSearchiOS(){
        uexBaiduMap.busLineSearch("北京", "717", "111");
    }

    window.uexOnload = function(){
        uexWidgetOne.cbError = function(opid, type, data){
            alert(data);
        }

        uexBaiduMap.cbPoiSearchInCity = function(mid, type, data){
            alert("城市范围单关键字内搜索返回的数据: " + data);
        }
        uexBaiduMap.cbPoiSearchArea = function(mid, type, data){
            alert("区域内单关键字搜索返回的数据" + data);
        }
        uexBaiduMap.cbPoiMultiSearchArea = function(mid, type, data){
            alert("区域内多关键字搜索返回的数据" + data);
        }

        uexBaiduMap.cbSuggestionSearch = function(mid, type, data){
            alert("联想猜词返回的数据: "+data);
        }
        uexBaiduMap.cbBusLineSearch = function(mid, type, data){
            alert(data);
        }
    }
    openmap();
```

```
}

//打开百度地图
function openmap(){
    uexBaiduMap.open('6643147DF56BD40F6140AB3ABA5A3FFBE12C439F', 0,400,800, 800,
'116.401', '39.925');
}
</script>
</head>
<body>
    <div class="tit">
        百度地图搜索功能
    </div>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type = "button" value="联想词搜索" onclick ="suggestionSearch();">
            <input class="btn" type = "button" value="指定城市内搜索兴趣点（单）"
                onclick ="poiSearchInCity();">
            <input class="btn" type = "button" value="周边（圆形区域）单关键字搜索兴趣点"
                onclick ="poiSearchNearBy();">
        </div>
        以下是 android 使用
    </div>
    <input class="btn" type = "button" value="Android 周边（圆形区域）多关键字搜索兴趣点"
        onclick ="poiMultiSearchNearBy();">
    <input class="btn" type = "button" value="Android 范围内（矩形区域）单关键字搜索兴趣点"
        onclick ="poiSearchInBounds();">
    <input class="btn" type = "button" value="Android 范围内（矩形区域）多关键字搜索兴趣点"
        onclick ="poiMultiSearchInBounds();">
    <input class="btn" type = "button" value="Android 公交线路搜索"
        onclick ="showBusLineSearch();">
    </div>
    以下是 iOS 使用
</div>
```



```

<input class="btn" type = "button" value="iOS 周边（圆形区域）多关键字搜索兴趣点"
        onclick ="poiMultiSearchNearByiOS();">
<input class="btn" type = "button" value="iOS 范围内（矩形区域）单关键字搜索兴趣点"
        onclick ="poiSearchInBoundsiOS();">
<input class="btn" type = "button" value="iOS 范围内（矩形区域）多关键字搜索兴趣点"
        onclick ="poiMultiSearchInBoundsiOS();">
<input class="btn" type = "button" value="iOS 公交线路搜索"
        onclick ="showBusLineSearchiOS();">
</div>
</div>
</body>
</html>

```

## 附录 6

```

<!DOCTYPE HTML>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0,
user-scalable=no" />
        <link rel="stylesheet" type="text/css" href="../css/index.css">
        <title>百度地图功能</title>
        <style>
            input {
                color: black
            }
        </style>
        <script type="text/javascript">
            window.uexOnload = function(){
                uexWidgetOne.cbError = function(opid, type, data){
                    alert(data);
                }
                uexBaiduMap.cbShowRoutePlan = function(mid, type, data){
                    var obj = eval('(' + data + ')');
                    alert("id:=" + obj.id + "\nresult:=" + obj.result);
                }
                openmap();
            }
            //根据传入参数信息在地图上显示规划的线路
            function showRoutePlan(){
                var type = document.getElementById("routetype").value;

```

```

        var jssonstr = '{"id":"rp345","type":"' + type + "','"start":{"city":"北京","name":"天安门"},
end":{"city":"北京","name":"百度大厦","longitude":"116.307827","latitude":"40.056957"}}';
        alert(jssonstr);
        uexBaiduMap.showRoutePlan(jssonstr);
    }
    //清除路线规划覆盖物
    function cleanRoutePlan(){
        uexBaiduMap.clearRoutePlan();
    }
    //打开百度地图
    function openmap(){
        uexBaiduMap.open('6643147DF56BD40F6140AB3ABA5A3FFBE12C439F', 0,400,800, 800,
'116.401', '39.925');
    }
</script>
</head>
<body>
    <div class="tit">
        百度地图功能
    </div>
    <div class="conbor">
        <div class="consj">
            <input type="text" value="2" id="routetype">0 驾车，1 公交，2 步行
            <input class="btn" type="button" value="显示路线" onclick="showRoutePlan();">
            <input class="btn" type="button" value="清除路线" onclick="cleanRoutePlan();">
        </div>
    </div>
</body>
</html>

```

## 附录 7

```

<!DOCTYPE HTML>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <meta name="viewport" content="target-densitydpi=device-dpi, width=device-width, initial-scale=1.0,
user-scalable=no" />
        <link rel="stylesheet" type="text/css" href="../css/index.css">
        <title>百度地图功能</title>
        <style>
            input {
                color: black
            }
        </style>

```

```
<script type="text/javascript">
    //根据地名获取地址经纬度信息
    function geocode(){
        uexBaiduMap.geocode("北京", "东长安街 33 号");
    }
    //根据经纬度获取地址详细信息
    function reverseGeocode(){
        uexBaiduMap.reverseGeocode("116.403981", "39.915101");
    }
    function getBaiduFromGoogle(){
        //谷歌百度经纬度转换
        uexBaiduMap.getBaiduFromGoogle(116.401, 39.925);
    }
    window.uexOnload = function(){
        uexWidgetOne.cbError = function(opid, type, data){
            alert(data);
        }
        uexBaiduMap.cbBaiduFromGoogle = function(longitude, latitude){
            alert("经度:=" + longitude + "\n 纬度=" + latitude);
        };
        uexBaiduMap.cbGeocode = function(mid, type, data){
            alert(data);
        }
        uexBaiduMap.cbReverseGeocode = function(mid, type, data){
            alert(data);
        }
        openmap();
    }
    //打开百度地图
    function openmap(){
        uexBaiduMap.open('6643147DF56BD40F6140AB3ABA5A3FFBE12C439F', 0,400,800, 800,
'116.401', '39.925');
    }
</script>
</head>
<body>
    <div class="tit">
        百度地理编码功能示例
    </div>
    <div class="conbor">
        <div class="consj">
            <input class="btn" type="button" value="谷歌百度经纬度转换"
onclick="getBaiduFromGoogle();">
            <input class="btn" type="button" value="地理编码" onclick="geocode();">
        </div>
    </div>
</body>
</html>
```

[键入文字]

---

```
        <input class="btn" type = "button" value="反地理编码" onclick ="reverseGeocode();">
      </div>
    </div>
  </body>
</html>
```