

# 高德地图 JS API 使用

高德地图 JS API 是一套 JavaScript 语言开发的的地图应用编程接口

此篇文档仅为实现点击地图图标显示相应经纬度坐标功能，[代码地址](#)。详细API见[官网文档](#)

## 准备

### 注册账号并申请Key

1. 首先，注册[开发者账号](#)，成为高德开放平台开发者
2. 登陆之后，在进入「应用管理」 页面「创建新应用」
3. 为应用添加 Key（key值会作为调用API是的识别参数），「服务平台」一项选择「 Web 端 ( JSAPI ) 」

### 加载 JS API(异步加载)

```
let option = { // 初始配置参数
  resizeEnable: true, // 监控地图容器尺寸变化
  zoom: 15, // 级别，地图放大缩小的程度
}
let url = 'https://webapi.amap.com/maps?v=1.4.15&key=' + 添加的key值 + '&callback=createMap';
let jsapi = document.createElement('script');
jsapi.charset = 'utf-8';
jsapi.src = url;
document.head.appendChild(jsapi);
createMap = function () { // 回调函数
  let map = new AMap.Map('container', option); // id为container的地图容器
}
```

到此，页面就会生成地图的标准底图

## API使用

### 功能

## 定位

定位功能需要用到提供的插件 `AMap.Geolocation` ,插件的使用

1. 引入插件，支持按需异步加载和同步加载，可同时引入多个插件；
2. 创建插件实例；
3. 调用实例的方法，使用相关功能。

```
AMap.plugin('AMap.Geolocation',function(){//异步加载插件
    let geolocation = new AMap.Geolocation({ // 创建插件实例，配置插件
        enableHighAccuracy: true, //是否使用高精度定位，默认:true
        timeout: 10000, //超过10秒后停止定位，默认: 5s
        buttonPosition: 'RB', //定位按钮的停靠位置
        buttonOffset: new AMap.Pixel(10, 20), //定位按钮与设置的停靠位置的偏移量，默认: Pixel(10, 20)
        zoomToAccuracy: true, //定位成功后是否自动调整地图视野到定位点
        showMarker: true
    });
    map.addControl(geolocation);
    geolocation.getCurrentPosition(function (status, result) { // 调用插件实例方法实现定位功能
        if (status == 'complete') {
            // addMarker(result.position)
        } else {
        }
    });
});
```

## 获取地图点击处的位置信息

要实现此功能需要调用AMap的事件

## 事件

事件的两种方式

1. 针对Map、覆盖物等常用类，我们推荐直接使用这些类的对象的on、off成员方法来实现事件的简单绑定和移除，具有使用方便、代码简洁的特点

```
// 注册事件，给Map或者覆盖物对象注册事件，
// eventName: 事件名称（必填）；
// handler: 事件回调函数（必填）；
// context可选参数
on( eventName, handler, context)
```

## 对应的移除事件

```
on( eventName, handler)
```

## 2. 使用AMap.event命名空间

```
// 注册DOM对象事件：给DOM对象注册事件，并返回eventListener，
// instance：需注册事件的DOM对象（必填）
// eventName：事件名称（必填）
// handler：事件功能函数（必填）
addDomListener( instance, eventName, handler, context)

// 注册对象事件：给对象注册事件，并返回eventListener，
// instance：需注册事件的DOM对象（必填）
// eventName：事件名称（必填）
// handler：事件功能函数（必填）
addListener( instance, eventName, handler, context)

// 删除event.addDomListener 和 event.addListener 传回的指定侦听器。
removeListener( listener)
```

## 实现

```
let mapClick = function (e) {
  map.clearMap() // 清除地图上的所有覆盖层
  addMarker(e.lnglat)
}
let addMarker = function (position) { // 创建marker
  let marker = new AMap.Marker({
    position: position
  })
  map.add(marker)
  showLocation(position)
}
let showLocation = function (pos) { // 显示经纬度
  console.log(pos)
  let locationWrap = document.querySelector('.location-wrap')
  locationWrap.innerHTML = `

纬度: ${pos.lat}</p><p>经度: ${pos.lng}</p>`
}
AMap.event.addListener(map, 'click', mapClick)


```

## vue引入高德地图

### demo代码地址

### 1. 主文件引入地图

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <script type="text/javascript" src="http://webapi.amap.com/maps?v=1.3&key='key值'"></script>
    <title>vue-amap</title>
  </head>
  <body>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

## 2. webpack.base.conf.js文件的最后添加代码

```
externals: { // 新增的配置代码
  'AMap': 'AMap'
}
```

## 3. 在需要引入地图的组件中引入

```
import AMap from 'AMap'
```

现在组件中就可以使用AMap了, 注意执行生成地图的方法不能在created钩子中, 要在mounted钩子执行

```

methods: {
  createMap () {
    let map = new AMap.Map('container', {
      resizeEnable: true,
      zoom: 15
    })
    AMap.plugin('AMap.Geolocation', function () {
      // 异步加载插件
      let geolocation = new AMap.Geolocation({
        // 定位插件
        enableHighAccuracy: true,
        timeout: 10000,
        buttonPosition: 'RB',
        buttonOffset: new AMap.Pixel(10, 20),
        zoomToAccuracy: true,
        showMarker: true
      })
      map.addControl(geolocation)
      geolocation.getCurrentPosition(function (status, result) {
        if (status === 'complete') {
          console.log(result)
          // addMarker(result.position)
        } else {
          // onError(result)
        }
      })
    })
  }
  let addMarker = function (position) {
    // 创建marker
    let marker = new AMap.Marker({
      position: position
    })
    map.add(marker)
    showLocation(position)
  }
  let showLocation = function (pos) {
    // 显示经纬度
    console.log(pos)
    let locationWrap = document.querySelector('.location-wrap')
    locationWrap.innerHTML = `

纬度: ${pos.lat}</p><p>经度: ${pos.lng}</p>`
  }
  let mapClick = function (e) {
    // map.remove('marker')
    map.clearMap() // 清除地图上的所有覆盖层
    addMarker(e.lnglat)
    console.log(map.getAllOverlays('marker')) // 获取所有marker覆盖层
  }
  AMap.event.addListener(map, 'click', mapClick)
},
mounted () {


```

```
    this.createMap()  
}
```

未完待续