

# 在Vue3中使用高德地图API入门篇

## 主要场景

在购物APP中，用户通过在地图上选择所在位置，作为送货地址

好处：无需使查询全国各地省市街道等数据库，即点即存

## 步骤

1. 注册并成为高德开放平台用户，并完成身份验证
2. 安装高德地图Vue3插件
3. 加载并创建地图库对象
4. 创建地图对象，在页面加载高德地图
5. 响应用在地图上的单击事件，取得地址并显示在页面

## 实现

1. 必须在“高德开发平台”网站注册并登录：<https://lbs.amap.com/>，可以使用QQ，或者支付宝“扫一扫”登录

2. 控制台：应用管理-->我的应用-->创建新应用

应用名称：随意，一般你的APP名

应用类型：随意选，一般选择与你应用相关的范围，例如社交、美食、教育等

3. 完成后，选择“添加key”

key名称：随意输入，例如mywebkey

服务平台：由于我们使用web，所以选择“web端（JS API）”

域名白名单：空，不填

提交即可

4. 在控制台你已经可以看到开发者key和安全key了。

**【官网参考】安全key的使用方法**

<https://lbs.amap.com/api/javascript-api-v2/guide/abc/load>

**【官网参考】在Vue中使用示例**

<https://lbs.amap.com/api/javascript-api-v2/guide/abc/amac-vue>

5. 必须注册成为开发者，进行实名认证：这样你才能使用

登录后，在控制台首页[\[https://console.amap.com/dev/index\]](https://console.amap.com/dev/index)，可以看到“开发者认证”链接

# 实例：在vue3中使用

需求：

- (1) 在页面加载地图
- (2) 单击地图获取单击点的地址信息，作为APP用户的送货地址

**地理编码GeoCoder: 只有两个方法getLocaltion(取得经纬度) /getAddress (取地址)**

步骤（必须认证成为个人开发者后，才能调用API）

- (1) 安装插件

```
npm i @amap/amap-jsapi-loader --save
```

- (2) 创建Vue组件

```
<template>
  <h3>鼠标选点，取得该点地址信息</h3>
  <div class="c">
    <div class="map" id="map" style="width: 80vw; height: 80vh"></div>
    <div>
      <p>当前选择的地址</p>
      <p>{{ curAddress }}</p>
    </div>
  </div>
</template>
<script setup>
// 导入高德地图加载器
import AMapLoader from '@amap/amap-jsapi-loader'
import { onMounted, onUnmounted, reactive, ref } from 'vue'

//保存当前在地图上单击时取得的完整地址
const curAddress = ref(null)
//详细的地址信息：省、市、区、街道、路、门牌号
const addressDetails = reactive({})

//不需要ref包装，因为不需要对他们作出响应式变化
//四个对象，基本常用到
let AMap = null //地图类库
let map = null //地图对象
let geocoder = null //地理编码对象
let marker = null //标注点

const initMap = async () => {
  // 1.必须有安全密钥，否则无法查询地址
  window._AMapSecurityConfig = {
    securityJsCode: '填写你自己的安全密钥', // "「你申请的安全密钥」",
  }

  //2.加载地图，获取map对象
  AMap = await AMapLoader.load({
    key: '填写你的KEY', // 填写你在高德地图中申请获取的开发者KEY
    version: '2.0', //当前使用的版本
  })
}
```

```

    plugins: ['AMap.Geocoder', 'AMap.Marker', 'AMap.Scale', 'AMap.ToolBar'], //
    需要加载的插件（类名），例如标尺：AMap.scale，缩放工具栏：AMap.ToolBar
  })

  //3.创建地图对象：容器id,选项参数（经纬度坐标[lng,lat]，缩放级别3-20）
  map = new AMap.Map('map', {
    // viewMode: '3D', // 是否为3D地图模式
    // center: [116.397428, 39.90923], //初始中心点，默认北京,如果不设置，将查询离你最近
    位置
    zoom: 11, //初始缩放级别：11-市级，18-建筑物级（小区级）
    resizeEnable: true, //是否可以缩放
  })

  //添加可见插件对象，添加到地图
  map.addControl(new AMap.ToolBar())
  map.addControl(new AMap.Scale())

  //创建不可见插件对象待用：地理编码：地址->坐标;坐标->地址
  geocoder = new AMap.Geocoder()//地理编码
  marker = new AMap.Marker()//标准点对象

  //关联单击事件，单击时，取到经纬度
  map.on('click', clickHandler)
}

//单击地图点，获取地址信息
const clickHandler = (e) => {
  let point = e.lnglat //经纬度对象
  // 通过经纬度取得地址
  getAddress(e.lnglat)
}

const getAddress = (point) => {
  geocoder.getAddress(point, (status, result) => {
    //完成且存在区域编码
    if (status === 'complete' && result.regeocode) {
      // 直接取得格式化的地址 regeo-code
      curAddress.value = result.regeocode.formattedAddress
      //详细地址信息 在控制台输出看每一个属性值
      const addressDetails = result.regeocode.addressComponnet

      //添加该位置标注
      marker.setPosition(point)
      marker.setTitle(curAddress.value)
      map.add(marker)

      map.setFitView()
    } else {
      curAddress.value = '无法取得当前地址，请配置安全秘钥'
    }
  })
}

onMounted(() => {
  initMap()
})

onUnmounted(() => {

```

```

map?.destroy()
//新语法
//等价于:   if(map) map.destroy
})
</script>

```

## 扩展: 根据城市（或更详细地址），找经纬度，并在地图上标注

```

const getLocation = (address) => {
  geocoder.getLocation(
    //关键字: address, 例如: 广州市
    address,
    //回调函数
    (status, result) => {
      if (status === 'complete' && result.geocodes.length > 0) {
        //存在则取第一个
        console.log(result.geocodes) //输出看结构
        const geo = result.geocodes[0]
        //geo 还包含formattedAddress和addressComponnet属性
        const point = geo.location //经纬度
        map.setZoomAndCenter(16, point) // 一步飞过去，以新的点为中心

        //标注
        marker.setPosition(point)
        marker.setTitle(geo.formattedAddress)
        map.add(marker)
      }
    }
  )
  // 可选项: 指定城市
  // { city: '广州市' }
}

```

## 完整代码=取得地址信息，查询地址

已给出项目代码

注意: 填写你自己的安全KEY和开发者KEY

```

<template>
  <h3>鼠标选点，取得该点地址信息</h3>
  <div class="c">
    <div class="map" id="map" style="width: 80vw; height: 80vh"></div>
    <div>
      <p>当前选择的地址</p>
      <p>{{ curAddress }}</p>
    </div>
  </div>
</template>

```

```

<script setup>
// 导入高德地图加载器
import AMapLoader from '@amap/amap-jsapi-loader'
import { onMounted, onUnmounted, reactive, ref, shallowRef } from 'vue'

//保存当前在地图上单击时取得的完整地址
const curAddress = ref(null)
//详细的地址信息：省、市、区、街道、路、门牌号
const addressDetails = reactive({
  province: '', //坐标点所在省名称 例如：北京市
  city: '', //坐标点所在城市名称,请注意：当城市是省直辖市时返回为空，以及城市为北京、上海、天津、重庆四个直辖市时，该字段返回为空；省直辖市列表
  citycode: '', //城市编码,例如：010
  district: '', //坐标点所在区,例如：海淀区
  adcode: '', // 行政区编码,例如：110108
  township: '', //坐标点所在乡镇/街道（此街道为社区街道，不是道路信息）例如：燕园街道
  towncode: '', //乡镇街道编码,例如：110101001000
  streetNumber: '', //门牌号
  // building: '', //楼信息列表:空值
})

//删：地图对象:对象太大，使用浅引用shallowRef（不要监听所有属性的变化），使用时与ref一样
let AMap = null //地图类库
let map = null //地图对象
let geocoder = null //地理编码对象
let marker = null //标注点

const initMap = async () => {
  // 必须有安全密钥，否则无法查询地址
  window._AMapSecurityConfig = {
    securityJsCode: '', // "「你申请的安全密钥」",
  }

  AMap = await AMapLoader.load({
    key: '', // 你在高德地图中申请获取的开发者KEY
    version: '2.0', //当前使用的版本
    plugins: ['AMap.Geocoder', 'AMap.Marker', 'AMap.Scale', 'AMap.ToolBar'], //
    需要加载的插件（类名），例如标尺：AMap.scale ,缩放工具栏:AMap.ToolBar
  })

  //创建不可见插件对象待用：地理编码：地址->坐标;坐标->地址
  geocoder = new AMap.Geocoder()
  marker = new AMap.Marker()

  //创建地图对象：容器id,选项参数（经纬度坐标[lng,lat]，缩放级别3-20）
  map = new AMap.Map('map', {
    // viewMode: '3D', // 是否为3D地图模式
    // center: [116.397428, 39.90923], //初始中心点，默认北京,如果不设置，将查询离你最近位置
    zoom: 11, //初始缩放级别：11-市级，18-建筑物级（小区级）
    resizeEnable: true, //是否可以缩放
  })

  //添加可见插件对象
  map.addControl(new AMap.ToolBar())
  map.addControl(new AMap.Scale())

  //关联单击事件，单击时，取到经纬度

```

```

    map.on('click', clickHandler)
  }

  const clickHandler = (e) => {
    let point = e.lnglat //经纬度对象
    // 通过经纬度取得地址
    getAddress(e.lnglat)
  }

  //根据经纬度查找地址信息
  const getAddress = (point) => {
    geocoder.getAddress(point, (status, result) => {
      //完成且存在区域编码
      //status (string) 当status为complete时, result为GeocodeResult;
      //当status为error时, result为错误信息info; 当status为no_data时, 代表检索返回0结果
      if (status === 'complete' && result.regeocode) {
        // 直接取得格式化的地址 regeo-code
        curAddress.value = result.regeocode.formattedAddress
        const comp = result.regeocode.addressComponent

        console.log(curAddress.value)
        // console.log(comp)

        for (let props in addressDetails) {
          addressDetails[props] = comp[props]
        }

        console.log(addressDetails)
        //添加该位置标注

        marker.setPosition(point)
        marker.setTitle(curAddress.value)
        map.add(marker)

        map.setFitView()
      } else {
        curAddress.value = '无法取得当前地址, 请配置安全秘钥'
      }
    })
  }

  const getLocation = (address) => {
    geocoder.getLocation(
      //关键字: address,例如: 广州市
      address,
      //回调函数
      (status, result) => {
        if (status === 'complete' && result.geocodes.length > 0) {
          //存在则取第一个
          console.log(result.geocodes) //输出看结构
          const geo = result.geocodes[0]
          //geo 还包含formattedAddress和addressComponent属性
          const point = geo.location //经纬度
          map.setZoomAndCenter(16, point) // 一步飞过去, 以新的点为中心

          //标注
          marker.setPosition(point)
          marker.setTitle(geo.formattedAddress)
        }
      }
    )
  }

```

```
        map.add(marker)
    }
}
// 可选项: 指定城市
// { city: '广州市' }
)
}
onMounted(() => {
    initMap()
})

onUnmounted(() => {
    map?.destroy()
    //新语法
    //等价于:    if(map) map.destroy
})
</script>
```

**注意：安全密码和开发者KEY必填！**

## 参考资料

---

名称	含义	规则说明	
status	返回结果状态值	返回值为 0 或 1，0 表示请求失败；1 表示请求成功。	
count	返回结果数目	返回结果的个数。	
info	返回状态说明	当 status 为 0 时，info 会返回具体错误原因，否则返回“OK”。详情可以参阅 <a href="#">info 状态表</a>	
geocodes	地理编码信息列表	结果对象列表，包括下述字段：	
	country	国家	国内地址默认返回中国
province	地址所在的省份名	例如：北京市。此处需要注意的是，中国的四大直辖市也算作省级单位。	
city	地址所在的城市名	例如：北京市	
citycode	城市编码	例如：010	
district	地址所在的区	例如：朝阳区	
street	街道	例如：阜通东大街	
number	门牌	例如：6号	
adcode	区域编码	例如：110101	
location	坐标点	经度，纬度	
level	匹配级别	参见下方的地理编码匹配级别列表	

## result.regeocode对象的格式

名称	含义	规则说明	
status	返回结果状态值	返回值为 0 或 1，0 表示请求失败；1 表示请求成功。	
info	返回状态说明	当 status 为 0 时，info 会返回具体错误原因，否则返回“OK”。详情可以参考 <a href="#">info 状态表</a>	
regeocode	逆地理编码列表	返回 regeocode 对象；regeocode 对象包含的数据如下：	
	addressComponent	地址元素列表	

<b>regeocode</b>	<b>逆地理编码列表</b>	<b>返回 regeocode 对象；regeocode 对象包含的数据如下：</b>
------------------	----------------	---



regeocode	addressComponent	返回 regeocode 对象; regeocode 对象包含的数据如下:
<p>province //坐标点所在省名称 例如: 北京市</p> <p>city //坐标点所在城市名称, 请注意: 当城市是省直辖县时返回为空, 以及城市为北京、上海、天津、重庆四个直辖市时, 该字段返回为空; 省直辖县列表</p> <p>citycode //城市编码, 例如: 010</p> <p>district 坐标点所在区, 例如: 海淀区</p> <p>//adcode 行政区编码, 例如: 110108</p> <p>township //坐标点所在乡镇/街道 (此街道为社区街道, 不是道路信息) 例如: 燕园街道</p> <p>// towncode //乡镇街道编码, 例如: 110101001000</p> <p>streetNumber//门牌号</p>		

streetNumber	门牌信息列表		
	street	街道名称	例如: 中关村北二条
number	门牌号	例如: 3号	
location	坐标点	经纬度坐标点: 经度, 纬度	
direction	方向	坐标点所处街道方位	
distance	门牌地址到请求坐标的距离	单位: 米	可以做离你最近距离

# 有疑问? -官网

查看官网帮助文档: 主页-->产品介绍

```
https://lbs.amap.com/api/webservice/guide/api/georegeo
```

# JS-API帮助

```
全部JS API文档: https://lbs.amap.com/api/javascript-api-v2/summary
地理编码API: https://lbs.amap.com/api/javascript-api-v2/documentation#%e5%9c%b0%e7%90%86%e7%bc%96%e7%a0%81
```

# 补充: 搜索并定位: 示例 (Vue3 / 普通JS 都通用)

在高德 JSAPI 中, `AMap.Geocoder.getLocation(address, callback)` 拿到 **经纬度** 后, 只需要 **三步** 就能把地图“飞”过去:

1. 把返回的 `lnglat` 设成地图中心
2. 适当放大层级 ( `setZoom` 或 `setZoomAndCenter` )
3. 如有需要再挂一个 `Marker` 标出来

下面给出**最小可运行示例**, 你直接粘进项目即可看到效果。

```
<template>
  <div>
    <input v-model="address" placeholder="请输入地址" @keyup.enter="locate" />
    <button @click="locate">定位</button>
    <div id="map" style="width:100%;height:90vh;"></div>
  </div>
</template>

<script setup>
import { ref, onMounted } from 'vue'
import AMapLoader from '@amap/amap-jsapi-loader'

const address = ref('北京市朝阳区望京SOHO')
let map, geocoder

onMounted(async () => {
  const AMap = await AMapLoader.load({
    key: '你的KEY',
    version: '2.0',
    plugins: ['AMap.Geocoder', 'AMap.Marker']
  })

  map = new AMap.Map('map', {
    zoom: 11,
    center: [116.397428, 39.90923]
  })

  geocoder = new AMap.Geocoder({ city: '北京' })
})

function locate () {
  geocoder.getLocation(address.value, (status, result) => {
    if (status === 'complete' && result.geocodes.length) {
      const lnglat = result.geocodes[0].location // LngLat 对象
      map.setZoomAndCenter(16, lnglat)           // 一步飞过去
      new AMap.Marker({ position: lnglat, map }) // 画个标记
    } else {
      alert('地址解析失败!')
    }
  })
}
</script>
```

## 关键 API 速记

步骤	代码	说明
初始化	<code>new AMap.Geocoder({city:'北京'})</code>	创建地理编码实例
解析	<code>geocoder.getLocation(addr, cb)</code>	地址 → 经纬度
定位	<code>map.setZoomAndCenter(zoom, lnglat)</code>	一步把地图中心/缩放设好
标记	<code>new AMap.Marker({position:lnglat, map})</code>	在目标点画标记

## 常见错误排查

- 没结果：检查 `city` 参数；设为 `'全国'` 试试。
- 坐标不对：确认返回的是 `AMap.LngLat` 对象，可直接喂给地图。
- 没显示标记：忘记把 `map` 传给 `Marker` 的构造函数。