

定位与搜索功能集成文档

一、环境要求

1. 支持的平台

- 浏览器环境：Chrome、Firefox、Edge 等支持现代 JavaScript 特性（ES6+、Fetch API、地理定位 API）的浏览器。
- Node.js 环境：v14.0+（如需在服务端使用，需额外处理浏览器特定 API，如 `navigator.geolocation`）。

2. 依赖

- 无第三方依赖，使用浏览器原生 API 和 Fetch 进行网络请求。

二、集成步骤

1. 引入代码文件

将以下 JavaScript 文件引入项目：

```
import { loadData, getLocation, reverseGeocode, geocode, poiSearch, nearbySearch } from './wz_location.min.js';
```

2. 初始化配置

```
// 加载访问密钥（必填，从服务端获取有效密钥）
loadData('your-access-key');
```

三、功能使用说明

1. 获取设备定位（含逆地理编码）

```
// 调用定位函数
getLocation()
  .then((data) => {
    console.log('定位返回信息结果:', data);
    // 此处可进行地址展示或后续业务逻辑
  })
  .catch((error) => {
    console.error('定位失败:', error);
  });
```

2. 正地理

```
geocode(address, city)
  .then((data) => {
    showResult(`正地理编码请求成功, 返回数据: ${JSON.stringify(data)}`);
  })
  .catch((error) => {
    showResult(`正地理编码请求失败: ${error}`);
  });
```

3. 逆地理

```
reverseGeocode(location)
  .then((data) => {
    showResult(`请求成功, 返回数据: ${JSON.stringify(data)}`);
  })
  .catch((error) => {
    showResult(`请求失败: ${error}`);
  });
```

4. POI 关键词搜索

```
// 搜索 "襄阳五中", 指定城市为 "襄阳市"
poiSearch('襄阳五中', '襄阳市')
  .then((result) => {
    console.log('搜索结果:', result);
  })
  .catch((error) => {
    console.error('搜索失败:', error);
  });
```

5. 附近搜索（基于定位）

```
// 假设已获取定位: location = "116.397,39.908" (经度, 纬度)
nearbySearch('咖啡馆', '', '116.397,39.908')
  .then((result) => {
    console.log('附近咖啡馆列表:', result);
  })
  .catch((error) => {
    console.error('附近搜索失败:', error);
  });
```

四、错误处理

常见错误码与解决方案

错误类型	错误信息示例	原因分析	解决建议
PERMISSION_DENIED	"用户拒绝了获取地理位置的请求。"	用户未授权浏览器获取定位	提示用户允许定位权限
TIMEOUT	"获取地理位置信息超时。"	浏览器定位超时 (默认 5 秒)	检查网络环境，或回退到 IP 定位
HTTP error! status: 403	"HTTP error! status: 403"	accessKey 无效或未配置	重新获取有效密钥并调用 loadData

接口文档

一、核心函数列表

loadData(key)

- **功能：**加载访问密钥，用于后续接口认证。
- **参数：**

名称	类型	是否必填	说明
key	string	是	有效的 accessKey

- **返回值：**无。

getLocation()

- **功能：**获取设备定位信息（优先使用浏览器定位，失败时通过 IP 定位）
- **返回值：**
 - **成功：**解析为逆地理编码结果（地址信息 JSON）。
 - **失败：**返回错误信息字符串（如权限拒绝、超时等）。

geocode(address, city)

- **功能：**将地址信息转换为经纬度（正地理编码）。
- **参数：**

名称	类型	是否必填	说明
address	object	是	湖北省武汉市洪山区泛悦城，需标准化地址信息
city	string	否	武汉市

- **返回值：**
 - **成功：**返回地址信息 JSON
 - **失败：**返回错误信息

reverseGeocode(locationP)

- **功能：**将经纬度转换为地址信息（逆地理编码）。

• 参数:

名称	类型	是否必填	说明
locationP	object	是	经纬度对象, 结构: { latitude: number, longitude: number, accuracy?: number }

- 返回值:
 - **成功**: 解析为地址信息 JSON (包含国家、城市、详细地址等)。
 - **失败**: 返回错误信息。

poiSearch(keyword, city, city_code, area_code, page_index, page_size)

- **功能**: 基于关键词搜索 POI 信息 (支持城市 / 区域筛选)。
- **参数**: 城市或是 city_code 或是 area_code 必填一个

名称	类型	是否必填	默认值	说明
keyword	string	是	-	搜索关键词 (如 "学校")
city	string	否	""	城市名称 (如 "北京市")
city_code	string	否	""	城市编码 (如 "110000")
area_code	string	否	""	区域编码 (如 "110101")
page_index	number	否	1	页码默认值 1
page_size	number	否	20	每页数量默认值 20

- 返回值:
 - **成功**: 解析为搜索结果 JSON (包含 POI 列表、总页数等)。
 - **失败**: 拒绝, 返回 "获取失败" 或 HTTP 错误信息。

nearbySearch(keyword, category, location, radius, orderby, page_index, page_size)

- **功能**: 搜索指定位置附近的 POI (需先获取定位)。
- **参数**:

名称	类型	是否必填	默认值	说明
keyword	string	是	-	搜索关键词（如 "餐厅"）
category	string	否	""	搜索类别（如 "food"）
location	string	是	""	中心位置（格式："经度，纬度"）
radius	number	否	5000	搜索半径（米）
orderby	string	否	"distance_score"	排序方式（如距离、评分）
page_index	number	否	1	页码
page_size	number	否	20	每页数量

- **返回值：**同 `poiSearch`。

三、接口限制

1. **精度说明：**浏览器定位精度受设备支持影响（GPS 精度最高，网络定位精度较低）。
2. **密钥安全：**`accessKey` 需通过安全通道获取，禁止硬编码在客户端代码中（推荐使用环境变量或服务端转发）。