WZLocation SDK for uni-app开发者指南

[WZLocation SDK for uni-app开发者指南](#)

- [获取位置信息](#)
  - [第一步、准备工作](#)
  - [第二步、SDK调用](#)
  - [第三步、打包不同平台](#)

概述

WZLocation SDK for uni-app 能够使得开发者准确快速的获取用户的位置信息。根据以下步骤，便可以得到用户的经纬度坐标，逆地理、正地理、poi 搜索等接口的快速使用。 uni-sdk 支持打包成H5,web,微信小程序, app。

第一步、准备工作

1. 从官方网站按照需求下载开发包并解压。

   解压后，将本文件导入到项目目录下，建议放到 utils 目录下。

2. 申请appkey：

   ```
   1. 通过官网至LOTBoard 注册账号：https://lothub.newayz.com/
   2. 注册账号后，并申请应用，获取 APP 的 accesske。
   ```

3. HBuilder X 请升级到 4.4 以上

第二步、SDK调用

1. 下载插件：uni-WiFi
2. 使用sdk

   根据您的业务需求和使用场景配置参数，参数配置如下：

字段	类型	是否必填	说明
appKey	String	是	从官网申请的应用的accesskey
deviceId	String	是	用户唯一ID
isLocateOnce	Boolean	是	是否单次定位，默认单次定位
interval	String	否	定位频率，默认 5 秒定位一位，单位毫秒

3. 在工程 manifest.json 的App权限配置中添加WIFI相关权限

```
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
```

4. 单次定位

在您的页面中导入定位包，并初始化数据

```
import * as wzLocation from '@/utils/wayz-sdk-uniApp.js';

onLoad(){
    // 参数1 官网申请应用的appKey, 参数2 用户设备唯一ID, 参数2 是否单次定位, 参数3 定位 频率(单次定位时此参数不生效)
    wzLocation.initOption("abcde","12345", true, 0)
    ...
}
```

开始定位，并设置定位回调

```
wzLocation.getLocation(
    res => {
        console.log("index success", res);
        res.timestamp = new Date(res.timestamp).toLocaleString()
        this.wzLocation = res;
    },
    fail => {
        console.log("fail" + fail.code + fail.msg);
    }
);
```

5. 连续定位

在您的页面中导入定位包，并初始化数据

```
import * as wzLocation from '@/utils/wayz-sdk-uniApp.js';

onLoad(){
    // 参数1 官网申请应用的appKey, 参数2 用户设备唯一ID, 参数2 是否单次定位, 参数3 定位 频率(单次定位时此参数不生效)
    wzLocation.initOption("abcde","12345", false, 0)
    ...
}
```

开始定位，并设置定位回调

```
wzLocation.getLocation(
    res => {
        console.log("index success", res);
        res.timestamp = new Date(res.timestamp).toLocaleString()
        this.wzLocation = res;
    },
    fail => {
        console.log("fail" + fail.code + fail.msg);
    }
);
```

6. 停止定位

调用 stopLocation 即可停止定位，此方法无需提供回调

```
wzLocation.stopLocation()
```

7. 逆地理接口调用

字段	类型	是否必填	说明
longitude	Number	是	经度
latitude	Number	是	纬度
spatialReference	String	是	坐标点所属的空间坐标系

```
getAddress() {
    // 设置官网申请应用的appKey
    wzLocation.setAk("")
    // 参数1 经度, 参数2 纬度, 参数3 坐标系
    wzLocation.getReverseCode(107.191693, 27.945061, 'gcj02').then(
        (res) => {
            console.log("address : " + res.address.name);
        },
        (error) => {
            console.log("error : ", error);
        });
}
```

- 返回成功

字段	类型	说明
source	String	定位源信息，如："Entity" 等
timestamp	String	定位结果时间戳
longitude	Number	经度
latitude	Number	纬度
accuracy	Number	定位精度
spatialReference	String	坐标系
address	String	地址信息
place	Place	定位 POI信息

Place字段如下

字段	类型	必须	说明
name	String	是	对应的 POI名称
type	String	是	当前场景的属性信息，如："Entity" 等
Distance{ line }	Number	否	定位结果坐标点到 poi 坐标点的直线距离

- 返回失败

字段	类型	说明
code	Number	错误码
msg	String	错误信息

8. 正地理接口调用

字段	类型	说明
name	String	搜索 KEY 关键词
city	String	城市

```
getGeoCode(){
    wzLocation.setAk("")
    wzLocation.getGeoCode("泛悦城", '武汉市').then(
        (res) => {
            let dataList =  res;
            console.log(dataList[0].geoPoint,dataList[0].address.name)
        },
        (error) => {
            console.log("error : ", error);
        });
}
```

9. poi 搜索接口调用

```
poiSearch(){
    wzLocation.setAk("")
    wzLocation.poiSearch("泛悦城T2", '武汉市').then(
        (res) => {
            let dataList =  res;
            console.log("poi搜
索: ",dataList[0].geoPoint,dataList[0].address.name)
        },
        (error) => {
            console.log("error : ", error);
        });
}
```

第三步、打包不同平台

1. 打包成浏览器

   不支持 chrome 浏览器（chrome 连接的是国外服务，故无法定位），浏览器只支持 GPS 或是 IP 定位。

2. 打包成应用 Android 与 IOS

   直接打包即可。

3. 打包成微信小程序

打包成小程序之后，需在 app.json 中添加相关权限，不支持连续后台定位，后台定位需申请小程序的 background 定位接口。

在 `app.json` 中声明位置权限：

- 添加必要的权限声明，例如：

```
{
  "requiredPrivateInfos": ["getLocation"],
  "permission": {
    "scope.userLocation": {
      "desc": "您的位置信息将用于小程序位置接口的效果展示"
    }
  }
}
```

- 错误码备注

错误码	参数说明
1001	定位权限未开
1002	WIFI 开关未开
1003	您尚未配置 appKey
1004	option 参数未配置
1005	option 参数非法, 需提供设备 ID
1006	WIFI 开启失败
10010	定位服务返回失败
2001	特殊异常