

4-14 把定位坐标转换成真实地址

140.143.132.225:8000/project-1/doc-62

一、获取定位坐标

微信小程序提供了定位接口，只需要我们调用方法即可。**uni-app**框架的**uni**对象里面也封装了地理定位的方法，我们来看一下。

uni.getLocation(OBJECT)

获取当前的地理位置和速度。在微信小程序中，当用户离开应用后，此接口无法调用，除非申请后台持续定位权限；当用户点击“显示在聊天顶部”时，此接口可继续调用。

OBJECT 参数说明

参数名	类型	必填	说明	平台差异说明
type	String	否	默认为 wgs84 返回 gps 坐标	
altitude	Boolean	否	传入 true 会返回高度信息，由于获取高度需要较高精确度，会减慢接口返回速度	App和字节跳动小程序不支持
geocode	Boolean	否	默认false，是否解析地址信息	仅App平台支持
success	Function	是	接口调用成功的回调函数，返回内容详见返回参数说明。	
fail	Function	否	接口调用失败的回调函数	
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）	

success 返回参数说明

参数	说明
latitude	纬度，浮点数，范围为-90~90，负数表示南纬
longitude	经度，浮点数，范围为-180~180，负数表示西经
speed	速度，浮点数，单位m/s
accuracy	位置的精确度
altitude	高度，单位 m
verticalAccuracy	垂直精度，单位 m（Android 无法获取，返回 0）

参数	说明
horizontalAccuracy	水平精度，单位 m
address	地址信息（仅App端支持，需配置geocode为true）

address 地址信息说明

属性	类型	描述	说明
country	String	国家	如“中国”，如果无法获取此信息则返回undefined
province	String	省份名称	如“北京市”，如果无法获取此信息则返回undefined
city	String	城市名称	如“北京市”，如果无法获取此信息则返回undefined
district	String	区（县）名称	如“朝阳区”，如果无法获取此信息则返回undefined
street	String	街道信息	如“酒仙桥路”，如果无法获取此信息则返回undefined
streetNum	String	获取街道门牌号信息	如“3号”，如果无法获取此信息则返回undefined
poiName	String	POI信息	如“电子城. 国际电子总部”，如果无法获取此信息则返回undefined
postalCode	String	邮政编码	如“100016”，如果无法获取此信息则返回undefined
cityCode	String	城市代码	如“010”，如果无法获取此信息则返回undefined

示例

```
1. uni.getLocation({
2.   type: 'wgs84',
3.   success: function (res) {
4.     console.log('当前位置的经度: ' + res.longitude);
5.     console.log('当前位置的纬度: ' + res.latitude);
6.   }
7. });
```

二、编辑签到页面

我们首先要获取用户签到时的地理定位

```
1. uni.showLoading({
2.   title: '签到中请稍后'
3. });
4. setTimeout(function() {
5.   uni.hideLoading();
6. }, 30000);
7. //获取地理定位
8. uni.getLocation({
9.   type: 'wgs84',
10.  success: function(resp) {
11.    let latitude = resp.latitude;
12.    let longitude = resp.longitude;
13.  }
14. })
```

接下来我们根据定位坐标，换算成真实地址，先引用腾讯位置SDK文件

```
1. var QQMapWX = require('../../lib/qqmap-wx-jssdk.min.js');
2. var qqmapsdk;
```

然后在 `onLoad()` 生命周期函数中，初始化 `qqmapsdk` 对象

```
1. onLoad: function() {
2.   qqmapsdk = new QQMapWX({
3.     key: 'KSFBZ-####-####-####-37KUE-W3FLZ'
4.   });
5. },
```

编写JS代码把GPS坐标转换成地址

```
1. qqmapsdk.reverseGeocoder({
2.   location: {
3.     latitude: latitude,
4.     longitude: longitude
5.   },
6.   success: function(resp) {
7.     // console.log(resp.result);
8.     let address = resp.result.address;
9.     let addressComponent = resp.result.address_component;
10.    let nation = addressComponent.nation;
11.    let province = addressComponent.province;
12.    let city = addressComponent.city;
13.    let district = addressComponent.district;
14.  }
15. })
```