

HarmonyOS NEXT 定位 SDK 接口文档

一、概述

本文档详细描述了定位 SDK 提供的一系列接口，包括接口名称、调用方式、请求参数、返回参数、调用示例以及错误码等信息。

二、接口详情

1. initOption 接口

接口名称

```
initOption(key: string, deviceId: string, interval: number): void
```

接口说明

初始化配置接口

接口参数

参数名	类型	描述
key	string	开发者密钥，用于验证身份。
deviceId	string	设备 ID，用于标识使用 SDK 的设备。
interval	number	定位间隔时间，单位为长秒，用于设置定位的频率。

返回参数

无

调用示例

```
const key = 'your_key';
const deviceId = 'your_device_id';
const interval = 5; // 5 秒
initOption(key, deviceId, interval);
```

2. openWifiMode接口

接口名称

```
openWifiMode(canScanWifi: boolean)
```

接口说明

该接口定义打开 WIFI 模式，是否允许采用扫描 WIFI 的 MAC 地址定位接口参数

参数名	类型	描述
canSanWifi	boolean	是否允许扫描 WIFI 的 Mac 地址

调用示例

```
this.wzLocation.openWifiMode = true;
```

3. LocationCallback 接口

接口名称

```
LocationCallback
```

接口说明

该接口定义了位置相关操作的回调函数，包含成功和失败的处理逻辑。

接口参数

参数名	类型	描述
success	(location?: LocationRes) => void	操作成功时的回调函数，接收一个可选的 <code>LocationRes</code> 类型的参数。
fail	(error: ErrorInfo) => void	操作失败时的回调函数，接收一个 <code>ErrorInfo</code> 类型的参数。

返回参数

无

4. setLocationCall 接口

接口名称

```
setLocationCall(locationCall: LocationCallback): void
```

接口说明

设置接口回调接口

接口参数

参数名	类型	描述
locationCall	LocationCallback	要设置的位置回调函数，包含成功和失败的处理逻辑。

返回参数

无

调用示例

```
const locationCall: LocationCallback = {
  success: (location) => {
    console.log('位置操作成功:', location);
  },
  fail: (error) => {
    console.error('位置操作失败:', error);
  }
};

setLocationCall(locationCall);
```

错误码

错误码	描述
10000	未同意隐私合规政策接口
10003	未配置 KEY
10007	未开启定位权限，去设置打开位置权限。
10001	扫描 Wi-Fi 出现错误
10006	context 获取失败
10005	startBackgroundRunning failed

5. startLocation 接口

接口名称

startLocation

接口说明

开启单次定位

调用方式

异步调用，使用 `await` 关键字或 `.then()` 方法处理回调

接口参数

无

返回参数

无

调用示例

```
startLocation().then(() => {
  console.log('单次定位已启动');
});
```

6. startContinuousLocation 接口

接口名称

```
async startLocation(): Promise<void>
```

接口说明

开启持续定位

调用方式

异步调用，使用 `await` 关键字或 `.then()` 方法处理返回结果。

接口参数

无

返回参数

无

调用示例

```
startContinuousLocation().then(() => {  
  console.log('单次定位已启动');  
});
```

7. stopLocation 接口

接口名称

```
stopLocation()
```

接口说明

停止定位

接口参数

无

返回参数

无

调用示例

```
stopLocation();  
console.log('位置服务已停止');
```

8. setAgreePrivacy 接口

接口名称

```
setAgreePrivacy(isAgree: boolean)
```

接口说明

是否同意合规政策

调用方式

同步调用

接口参数

参数名	类型	描述
isAgree	boolean	是否同意隐私合规协议，true 表示同意，false 表示不同意。

返回参数

无

调用示例

```
const isAgree = true;
setAgreePrivacy(isAgree);
```

9. convertCoordinate 接口

接口名称

```
convertCoordinate(locationFrom: string, originSprf: string, targetSprf: string = "gcj02", callBack: { success: (result: any) => void }): void
```

- **功能：**将指定的坐标从一个坐标系统转换到另一个坐标系统。
- **参数：**
 - locationFrom：待转换的坐标，格式为 "经度, 纬度 | 经度, 纬度 |..."
 - originSprf：原始坐标系统，例如 "bd09"
 - targetSprf：目标坐标系统，默认为 "gcj02"
 - callBack：回调函数，包含 success 方法用于处理转换成功的结果
- **返回值：**无，通过回调函数返回结果

10. poiKeywordSearch 接口

接口名称

```
poiKeywordSearch(cityCode: string, keywords: string, area_code?: string, pageIndex: number = 1, pageSize: number = 10): void
```

- **功能：**根据关键字在指定城市和区域进行 POI 搜索。
- **参数：**
 - cityCode：城市代码
 - keywords：搜索关键字
 - area_code：区域代码，可选参数
 - pageIndex：页码，默认为 1
 - pageSize：每页显示数量，默认为 10

- **返回值**：无，通过 `mLocationCall.success` 回调返回搜索结果

11. poiNearbySearch

接口名称

```
poiNearbySearch(location: string, keywords: string, radius: number = 5000,
pageIndex: number = 1, pageSize: number = 20): void
```

- **功能**：在指定中心点的周边进行 POI 搜索。
- **参数**：
 - `location`：中心点坐标，格式为 "经度, 纬度"
 - `keywords`：搜索关键字
 - `radius`：搜索半径，单位为米，默认为 5000 米
 - `pageIndex`：页码，默认为 1
 - `pageSize`：每页显示数量，默认为 20
- **返回值**：无，通过 `mLocationCall.success` 回调返回搜索结果

12. getGeocode

接口名称

```
getGeocode(city?: string, address: string): void
```

- **功能**：根据地址信息获取对应的地理编码。
- **参数**：
 - `city`：城市名称，可选参数
 - `address`：地址信息。需按照标准规范填写例：湖北省武汉市洪山区华中科技大学
- **返回值**：无，通过 `mLocationCall.success` 回调返回地理编码结果

13. getAddressFromWZ

接口名称

```
getAddressFromWZ(location: { longitude: number, latitude: number }): void
```

- **功能**：根据坐标点获取对应的地址信息。
- **参数**：
 - `location`：坐标点，包含经度和纬度
- **返回值**：无，目前该方法调用 `reqBodyByServer` 但未实现，可根据实际情况补充。