

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

目录视图

摘要视图



从创业到再就业，浅谈对程序员职业生涯的看法 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

Android百度定位API的使用

标签： android 百度地图定位

2016-05-15 19:05 168人阅读 评论(0) 收藏 举报

分类： 移动开发 ( 38 ) 百度地图 ( 2 )

目录(?) [+]

导入库文件

在下载页面下载最新的库文件。将liblocSDK2.4.so文件拷贝到libs/armeabi目录下。将locSDK2.4.jar文件拷贝到工程根目录下，并在工程属性->Java Build Path->Libraries中选择“Add JARs”，选定locSDK2.4.jar，确定后返回。这样您就可以在程序中使用百度定位API了。

设置AndroidManifest.xml

为区分2.3版本service，需要将manifest file中的 intent filter声明为com.baidu.location.service\_v2.4 在application标签中声明service组件

```
1. <service android:name="com.baidu.location.f" android:enabled="true" android:process=":remote"
2. android:permission="android.permission.BAIDU_LOCATION_SERVICE">
3.     <intent-filter>
4.         <action android:name="com.baidu.location.service_v2.4"></action>
5.     </intent-filter>
6. </service>
```

声明使用权限

```
1. <permission android:name="android.permission.BAIDU_LOCATION_SERVICE"></permission>
2. <uses-permission android:name="android.permission.BAIDU_LOCATION_SERVICE"></uses-permission>
3. <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-permission>
4. <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
5. <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
6. <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
7. <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
8. <uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-permission>
9. <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
10. <uses-permission android:name="android.permission.INTERNET" />
11. <uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"></uses-permission>
12. <uses-permission android:name="android.permission.READ_LOGS"></uses-permission>
```

import相关类

```
1. import com.baidu.location.BDLocation;
2. import com.baidu.location.BDLocationListener;
3. import com.baidu.location.LocationClient;
4. import com.baidu.location.LocationClientOption;
5. import com.baidu.location.BDNotifyListener; //假如用到位置提醒功能，需要import该类
```

功能类的使用

初始化LocationClient类

此处需要注意：LocationClient类必须在主线程中声明。需要Context类型的参数。

```
1. public LocationClient mLocationClient = null;
2. public BDLocationListener myListener = new MyLocationListener();
3.
4. public void onCreate() {
5.     mLocationClient = new LocationClient(this);    //声明LocationClient类
6.     mLocationClient.registerLocationListener( myListener );    //注册监听函数
7. }
```

### 实现BDLocationListener接口

BDLocationListener接口有2个方法需要实现：

- 1.接收异步返回的定位结果，参数是BDLocation类型参数。
- 2.接收异步返回的POI查询结果，参数是BDLocation类型参数。

```
1. public class MyLocationListener implements BDLocationListener {
2.     @Override
3.     public void onReceiveLocation(BDLocation location) {
4.         if (location == null)
5.             return ;
6.         StringBuffer sb = new StringBuffer(256);
7.         sb.append("time : ");
8.         sb.append(location.getTime());
9.         sb.append("\nerror code : ");
10.        sb.append(location.getLocType());
11.        sb.append("\nlatitude : ");
12.        sb.append(location.getLatitude());
13.        sb.append("\nlonitude : ");
14.        sb.append(location.getLongitude());
15.        sb.append("\nradius : ");
16.        sb.append(location.getRadius());
17.        if (location.getLocType() == BDLocation.TypeGpsLocation){
18.            sb.append("\nspeed : ");
19.            sb.append(location.getSpeed());
20.            sb.append("\nsatellite : ");
21.            sb.append(location.getSatelliteNumber());
22.        } else if (location.getLocType() == BDLocation.TypeNetWorkLocation){
23.            sb.append("\naddr : ");
24.            sb.append(location.getAddrStr());
25.        }
26.
27.        logMsg(sb.toString());
28.    }
29.    public void onReceivePoi(BDLocation poiLocation) {
30.        if (poiLocation == null){
31.            return ;
32.        }
33.        StringBuffer sb = new StringBuffer(256);
34.        sb.append("Poi time : ");
35.        sb.append(poiLocation.getTime());
36.        sb.append("\nerror code : ");
37.        sb.append(poiLocation.getLocType());
38.        sb.append("\nlatitude : ");
39.        sb.append(poiLocation.getLatitude());
40.        sb.append("\nlonitude : ");
41.        sb.append(poiLocation.getLongitude());
42.        sb.append("\nradius : ");
43.        sb.append(poiLocation.getRadius());
44.        if (poiLocation.getLocType() == BDLocation.TypeNetWorkLocation){
45.            sb.append("\naddr : ");
46.            sb.append(poiLocation.getAddrStr());
47.        }
48.        if(poiLocation.hasPoi()){
49.            sb.append("\nPoi:");
50.            sb.append(poiLocation.getPoi());
51.        }
```

```
51.     }else{
52.         sb.append("noPoi information");
53.     }
54.     logMsg(sb.toString());
55. }
56. }
```

### 设置参数

设置定位参数包括：定位模式（单次定位，定时定位），返回坐标类型，是否打开GPS等等。eg：

```
1. LocationClientOption option = new LocationClientOption();
2. option.setOpenGps(true);
3. option.setAddrType("detail");
4. option.setCoorType("gcj02");
5. option.setScanSpan(5000);
6. option.disableCache(true); //禁止启用缓存定位
7. option.setPoiNumber(5); //最多返回POI个数
8. option.setPoiDistance(1000); //poi查询距离
9. option.setPoiExtraInfo(true); //是否需要POI的电话和地址等详细信息
10. mLocClient.setLocOption(option);
```

### 发起定位请求

发起定位请求。请求过程是异步的，定位结果在上面的监听函数onReceiveLocation中获取。

```
1. if (mLocClient != null && mLocClient.isStarted())
2.     mLocClient.requestLocation();
3. else
4.     Log.d("LocSDK_2.0_Demo1", "locClient is null or not started");
```

### 发起POI查询请求

发起POI查询请求。请求过程是异步的，定位结果在上面的监听函数onReceivePoi中获取。

```
1. if (mLocClient != null && mLocClient.isStarted())
2.     mLocClient.requestPoi();
```

### 位置提醒使用

位置提醒最多提醒3次，3次过后将不再提醒。假如需要再次提醒，或者要修改提醒点坐标，都可通过函数SetNotifyLocation()来实现。

```
1. //位置提醒相关代码
2. mNotifier = new NotifyListner();
3. mNotifier.SetNotifyLocation(42.03249652949337, 113.3129895882556, 3000, "gps"); //4个参数代表要位置提醒的点的坐标，具体含义依次为：纬度，经度，距离范围，坐标系类型(gcj02, gps, bd09, bd0911)
4. mLocationClient.registerNotify(mNotifier);
5. //注册位置提醒监听事件后，可以通过SetNotifyLocation 来修改位置提醒设置，修改后立刻生效。
```

```
1. //BDNotifyListner实现
2. public class NotifyListner extends BDNotifyListener{
3.     public void onNotify(BDLocation mlocation, float distance){
4.         mVibrator01.vibrate(1000); //振动提醒已到设定位置附近
5.     }
6. }
```

```
1. //取消位置提醒
2. mLocationClient.removeNotifyEvent(mNotifier);
```