## Android高德地图SDK的使用

1. 开这者平台Android Studio
2. 在Android Studio新建项目SDSYu

## 外部配置

1. 获取高德key

|  |
| --- |
| 先在这个网址注册高德账号：  <https://lbs.amap.com/api/android-location-sdk/guide/create-project/get-key/>  注册完成后进行登录  登录进去后点击应用管理，新建一个应用，这里的应用名称和类型可以随便填    点击创建    然后要为这个项目添加新key    出现这个界面 |

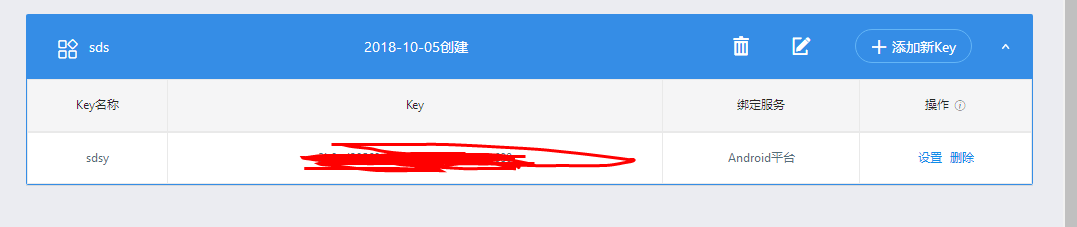
1. SHA1的获取:

|  |
| --- |
| 打开控制台：    然后输入cd .android 回车  在输入：keytool -list -v -keystore debug.keystore 在回车可以看到下面的情况    这里就能找到你要的SHA1了 |

1. PackName的获取

|  |
| --- |
| 打开    这句话就是： |

1. 点击提交：



1. 样子就给咱们的SDSYu的配置好了高德的外部配置

## SDSYu工程内部配置了

1. 添加高德的定位架包：

|  |  |  |
| --- | --- | --- |
| <https://lbs.amap.com/api/android-location-sdk/download/在这里下载架包>  架包下载好后复制这个文件夹下的D:\高德地图的架包\AMapLocation    然后将jar包拷贝到libs目录下：   |  | | --- | | libs目录的查找： | | 拷贝进去后配置 build.gradle 文件(app) 在 build.gradle 文件的 dependencies 中配置 compile fileTree(include: ['\*.jar'], dir: 'libs')。  IMG_256 | |

9.在AndroidManifest.xml的application标签中配置Key：

[XML](https://lbs.amap.com/api/android-location-sdk/javascript:void(0);)

<meta-data android:name="com.amap.api.v2.apikey" android:value="您的Key"></meta-data>

这里你的key就是刚才在网站生成的key值

在application标签中声明service组件：

[XML](https://lbs.amap.com/api/android-location-sdk/javascript:void(0);)

<service android:name="com.amap.api.location.APSService"></service>

在AndroidManifest.xml中配置权限:

[XML](https://lbs.amap.com/api/android-location-sdk/javascript:void(0);)

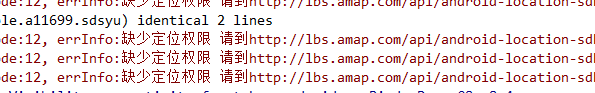
<!--用于进行网络定位--><uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"></uses-permission><!--用于访问GPS定位--><uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"></uses-permission><!--获取运营商信息，用于支持提供运营商信息相关的接口--><uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"></uses-permission><!--用于访问wifi网络信息，wifi信息会用于进行网络定位--><uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"></uses-permission><!--这个权限用于获取wifi的获取权限，wifi信息会用来进行网络定位--><uses-permission android:name="android.permission.CHANGE\_WIFI\_STATE"></uses-permission><!--用于访问网络，网络定位需要上网--><uses-permission android:name="android.permission.INTERNET"></uses-permission><!--用于读取手机当前的状态--><uses-permission android:name="android.permission.READ\_PHONE\_STATE"></uses-permission><!--写入扩展存储，向扩展卡写入数据，用于写入缓存定位数据--><uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"></uses-permission><!--用于申请调用A-GPS模块--><uses-permission android:name="android.permission.ACCESS\_LOCATION\_EXTRA\_COMMANDS"></uses-permission><!--用于申请获取蓝牙信息进行室内定位--><uses-permission android:name="android.permission.BLUETOOTH"></uses-permission><uses-permission android:name="android.permission.BLUETOOTH\_ADMIN"></uses-permission>

## finished

然后测试：

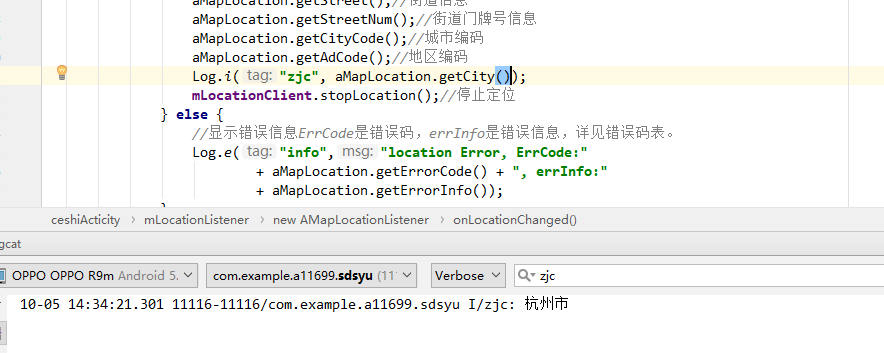
|  |
| --- |
| **package** com.example.a11699.sdsyu.activities;  **import** android.provider.ContactsContract; **import** android.support.v7.app.AppCompatActivity; **import** android.os.Bundle; **import** android.util.Log;  **import** com.amap.api.location.AMapLocation; **import** com.amap.api.location.AMapLocationClient; **import** com.amap.api.location.AMapLocationClientOption; **import** com.amap.api.location.AMapLocationListener; **import** com.example.a11699.sdsyu.R;  **import static** android.provider.ContactsContract.\*;  **public class** ceshiActicity **extends** AppCompatActivity {  **public** AMapLocationClient **mLocationClient**=**null**;  *//声明定位回调监听器* **public** AMapLocationClientOption **mLocationOption**=**null**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_ceshi\_acticity***);  initMap();  }  **private void** initMap(){  *//初始化定位* **mLocationClient**=**new** AMapLocationClient(ceshiActicity.**this**);  *//设置定位回调监听* **mLocationClient**.setLocationListener(**mLocationListener**);  **mLocationOption** = **new** AMapLocationClientOption();  *//设置定位模式为高精度模式，AMapLocationMode.Battery\_Saving为低功耗模式，AMapLocationMode.Device\_Sensors是仅设备模式* **mLocationOption**.setLocationMode(AMapLocationClientOption.AMapLocationMode.***Hight\_Accuracy***);  *//设置是否返回地址信息（默认返回地址信息）* **mLocationOption**.setNeedAddress(**true**);  *//设置是否只定位一次,默认为false* **mLocationOption**.setOnceLocation(**false**);  *//设置是否强制刷新WIFI，默认为强制刷新* **mLocationOption**.setWifiActiveScan(**true**);  *//设置是否允许模拟位置,默认为false，不允许模拟位置* **mLocationOption**.setMockEnable(**false**);  *//设置定位间隔,单位毫秒,默认为2000ms* **mLocationOption**.setInterval(2000);  *//给定位客户端对象设置定位参数* **mLocationClient**.setLocationOption(**mLocationOption**);  *//启动定位* **mLocationClient**.startLocation();  }  **public** AMapLocationListener **mLocationListener**=**new** AMapLocationListener() {  @Override  **public void** onLocationChanged(AMapLocation aMapLocation) {  **if** (aMapLocation != **null**) {  **if** (aMapLocation.getErrorCode() == 0) {  *//定位成功回调信息，设置相关消息* aMapLocation.getLocationType();*//获取当前定位结果来源，如网络定位结果，详见定位类型表* aMapLocation.getLatitude();*//获取纬度* aMapLocation.getLongitude();*//获取经度* aMapLocation.getAccuracy();*//获取精度信息* java.text.SimpleDateFormat df = **new** java.text.SimpleDateFormat(**"yyyy-MM-dd HH:mm:ss"**);   aMapLocation.getAddress();*//地址，如果option中设置isNeedAddress为false，则没有此结果，网络定位结果中会有地址信息，GPS定位不返回地址信息。* aMapLocation.getCountry();*//国家信息* aMapLocation.getProvince();*//省信息* aMapLocation.getCity();*//城市信息* aMapLocation.getDistrict();*//城区信息* aMapLocation.getStreet();*//街道信息* aMapLocation.getStreetNum();*//街道门牌号信息* aMapLocation.getCityCode();*//城市编码* aMapLocation.getAdCode();*//地区编码* Log.*i*(**"zjc"**, aMapLocation.getCity());  **mLocationClient**.stopLocation();*//停止定位* } **else** {  *//显示错误信息ErrCode是错误码，errInfo是错误信息，详见错误码表。* Log.*e*(**"info"**,**"location Error, ErrCode:"** + aMapLocation.getErrorCode() + **", errInfo:"** + aMapLocation.getErrorInfo());  }  }   }  }; } |

如果log打印输出：



这个缺少的权限是你的手机里的定位权限，而不是代码配置的权限，代码里给了权限，而你在手机禁用了

手机给了权限后：



可以看到定位到了：地名 杭州