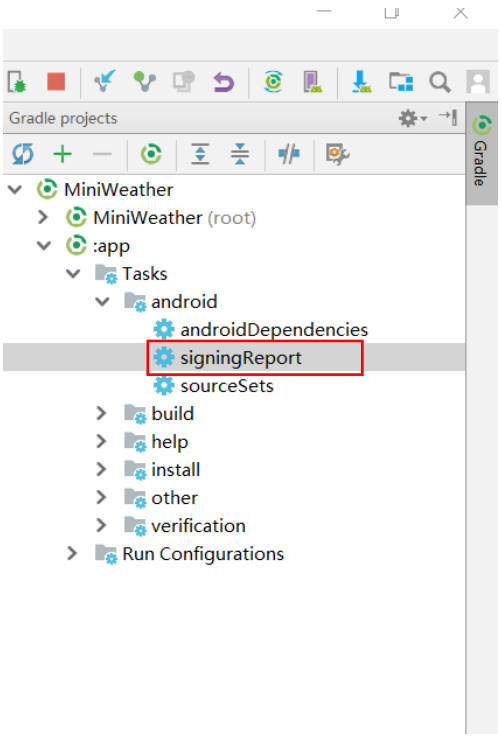
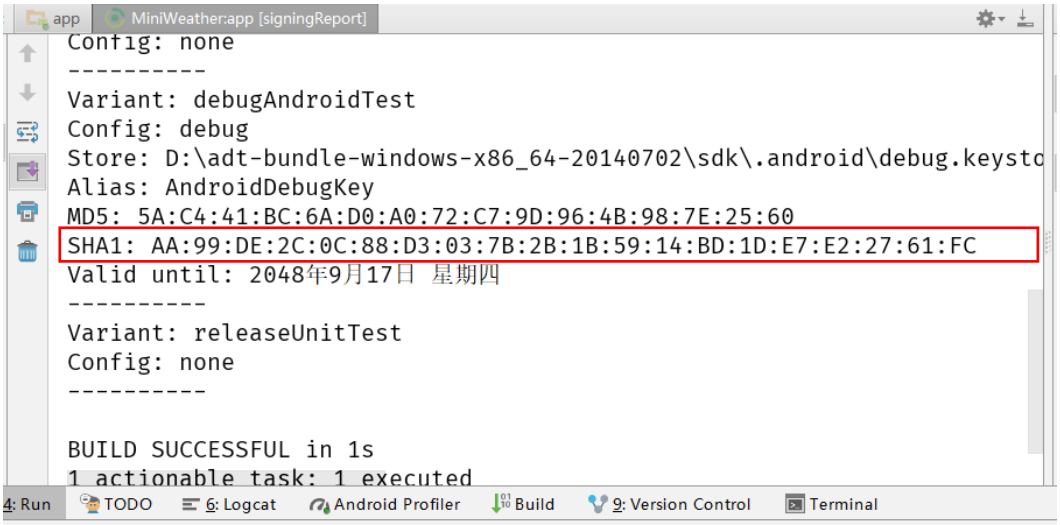
一、 前期准备

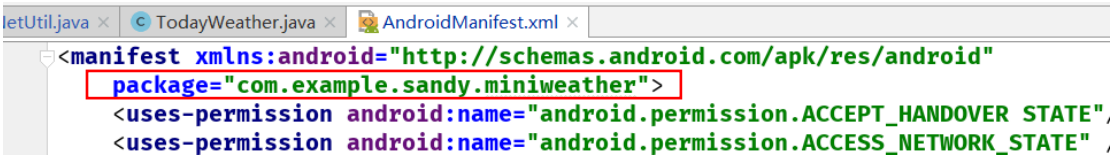
1. 注册一个百度开发者账号
2. http://lbsyun.baidu.com/apiconsole/key 进入控制台，创建应用

开发版 SHA1 可以在 Android Studio 里直接找到。
点击右边框的 Gradle，双击 signingReport





包名在 AndroidManifest.xml 文件中

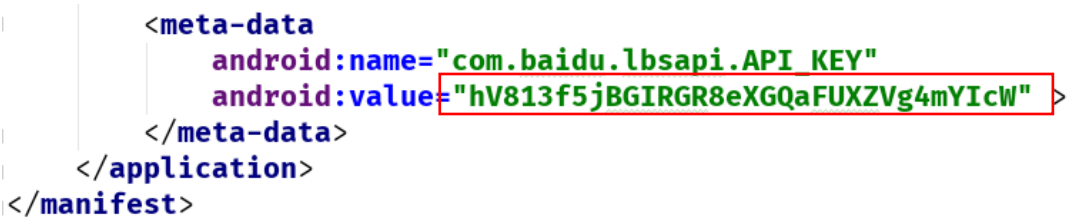


如果程序要发布，就需要生成发布版 SHA1　可以利用以下命令获取

keytool -list -v -keystore <jks 文件路径>

现在我们测试的话暂时不需要

点提交我们就得到了 AK，复制到 AndroidManifest.xml



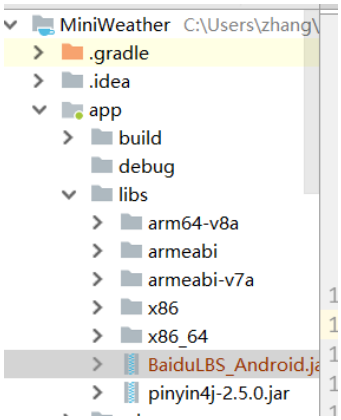

```
<meta-data
    android:name="com.baidu.lbsapi.API_KEY"
    android:value="AK 复制到这里" >
</meta-data>
```

3. **准备 LBS SDK**

   http://lbsyun.baidu.com/index.php?title=sdk/download&action#selected=location_all

   可以根据自己的需要下载 SDK

下载解压之后把文件全部 copy 到 app 下 libs 文件夹中

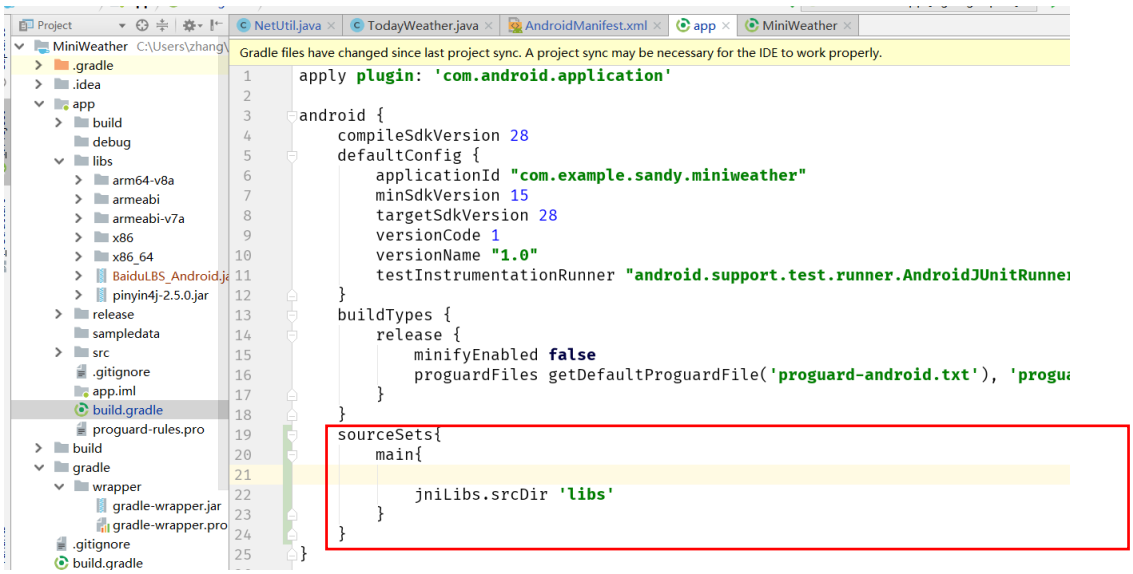


右键 jar 包，选择 add as library，我们就导入进去了，这时候检查 app/build.gradle

```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs
    implementation 'com.android.support:appcompat-v7:28.+'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:ru
    androidTestImplementation 'com.android.support.test.es
    implementation files('libs/pinyin4j-2.5.0.jar')
    implementation files('libs/BaiduLBS_Android.jar')
}
```

表示导入成功

然后导入.so 文件



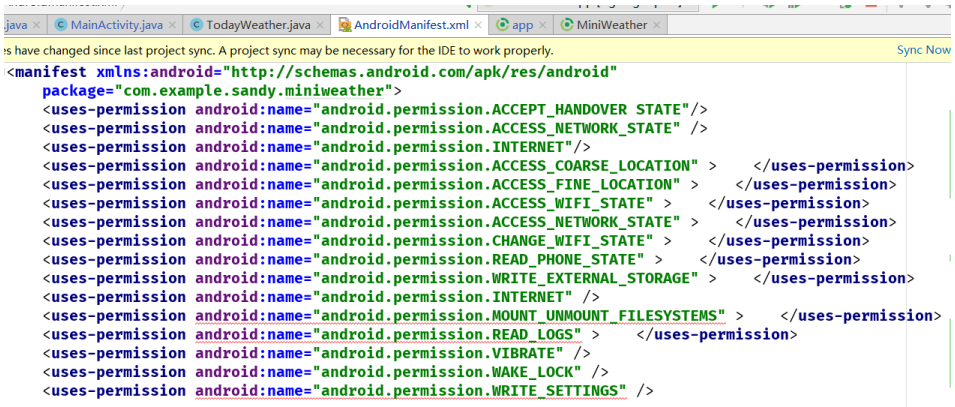

```
sourceSets{
    main{
        jniLibs.srcDir 'libs'
    }
}
```

到此为止准备工作就做完了。

二、 确定自己位置

我开始是按百度开发者文档做的，但是有点坑，有的地方写的不对

在 AndroidManifest.xml 中增加静态权限




```
<uses-permission android:name="android.permission.ACCEPT_HANDOVER STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" > </uses-permission>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" > </uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" > </uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" > </uses-permission>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" > </uses-permission>
<uses-permission android:name="android.permission.READ_PHONE_STATE" > </uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" > </uses-permission>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" > </uses-permission>
```

```
<uses-permission android:name="android.permission.READ_LOGS" >
</uses-permission>
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.WRITE_SETTINGS"
/>
```

然后增加定位权限

```
        <service
            android:name="com.baidu.location.f"
            android:enabled="true"
            android:process=":remote">
        </service>
        <meta-data
            android:name="com.baidu.lbsapi.API_KEY"
            android:value="hV813f5jBGIRGR8eXGQaFUXZVg4mYIcW" >
        </meta-data>
    </application>
</manifest>
```

```
<service
    android:name="com.baidu.location.f"
    android:enabled="true"
    android:process=":remote">
</service>
```

初始化：

```
private void initLocation() {
    LocationClientOption option = new LocationClientOption();
    //就是这个方法设置为true，才能获取当前的位置信息
    option.setIsNeedAddress(true);
    option.setOpenGps(true);
    option.setLocationMode(LocationClientOption.LocationMode.Hight_Accuracy
    );//可选，默认高精度，设置定位模式，高精度，低功耗，仅设备
    option.setCoorType("gcj02");//可选，默认gcj02，设置返回的定位结果坐标系
    //int span = 1000;
    //option.setScanSpan(span);//可选，默认0，即仅定位一次，设置发起定位请求的间隔需要大于等于1000
    mLocationClient.setLocOption(option);
}
```

```
private void initLocation() {
    LocationClientOption option = new LocationClientOption();
    //就是这个方法设置为true，才能获取当前的位置信息
    option.setIsNeedAddress(true);
    option.setOpenGps(true);

option.setLocationMode(LocationClientOption.LocationMode.Hight_Accuracy
    );//可选，默认高精度，设置定位模式，高精度，低功耗，仅设备
    option.setCoorType("gcj02");//可选，默认 gcj02，设置返回的定位结果坐
```

```
标系
        //int span = 1000;
        //option.setScanSpan(span);//可选，默认 0，即仅定位一次，设置发起定
位请求的间隔需要大于等于 1000ms 才是有效的
        mLocationClient.setLocOption(option);
}
```

然后是可以获取各种信息：

```
public class MyLocationListener extends BDAbstractLocationListener {
    @Override
    public void onReceiveLocation(BDLocation location){
        //此处的 BDLocation 为定位结果信息类，通过它的各种 get 方法可获
取定位相关的全部结果
        //以下只列举部分获取经纬度相关（常用）的结果信息
        //更多结果信息获取说明，请参照类参考中 BDLocation 类中的说明
        String getcity=location.getCity();
        String city=getcity.replace("市","");
        //打印出当前城市
        List<City> mCitylist;
        MyApplication myApplication;
        myApplication=MyApplication.getInstance();
        mCitylist=myApplication.getCityList();
        for(City city1:mCitylist){
            if(city1.getCity().equals(city))
            {
                cityCode=city1.getNumber();

            }
        }
       // int errorCode = location.getLocType();
        Log.i("TAG", "location.getLocType()=" + location.getLocType());
        //获取定位类型、定位错误返回码，具体信息可参照类参考中
BDLocation 类中的说明
    }
}
```

可以通过 **location.getLocType()**返回的错误码检验哪里出了问题

开发者文档的坑：

```
public class MyLocationListener extends BDAbstractLocationListener {
    @Override
    public void onReceiveLocation(BDLocation location){
        //此处的BDLocation为定位结果信息类，通过它的各种get方法可获取定位相关的全部结果
        //以下只列举部分获取经纬度相关（常用）的结果信息
        //更多结果信息获取说明，请参照类参考中BDLocation类中的说明
        String getcity=location.getCity();
```