《百度地图实战 实验手册 01》

Android 课程组

版 本 1.0

文档提供：Java 课程组 丁盟

目录

第1章　百度地图基础使用

1.1 实验目的

目的一： 了解百度地图 SDK 所包含的功能。

目的二： 掌握百度地图 SDK 在 Android 项目中的使用方法。

目的三： 掌握百度地图开发者秘钥的申请过程。

1.2 准备工作

准备一： 下载百度地图 SDK。

登录百度地图 Android SDK 开发资源下载平台选择相应开发包进行下载。

http://lbsyun.baidu.com/sdk/download

图 1.2.1

下载后将压缩文件进行解压可得到如图文件。

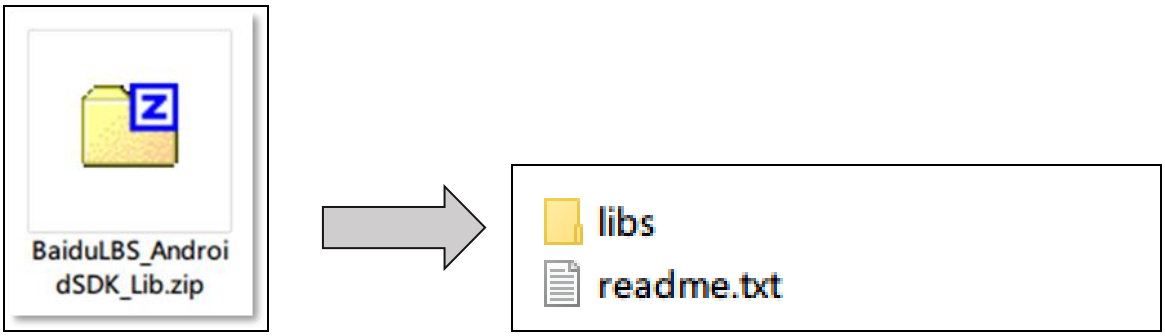

图 1.2.2

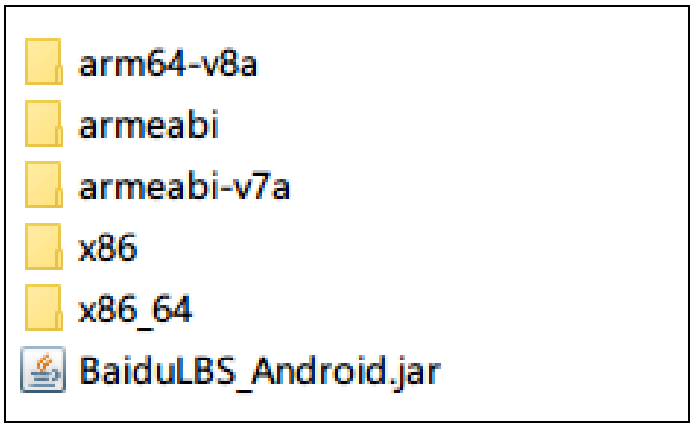

图 1.2.3

如上图所示便是百度地图 SDK 开发包的相关文件，同样的方法可以下载百度地图官方提供的帮助说明文档及官方 Demo 文件。

准备二： 申请秘钥。

在使用百度地图 SDK 为您提供的各种 LBS 能力之前，您需要获取百度地图移动版的开发密钥，该密钥与您的百度账户相关联。因此，您必须先有百度帐户，才能获得开发密钥。并且，该密钥与您创建的过程名称有关。

AK(API Key)的申请地址为：http://lbsyun.baidu.com/apiconsole/key

1) 登录百度账号

访问 API 控制台页面，若您未登录百度账号，将会进入百度账号登录页面。

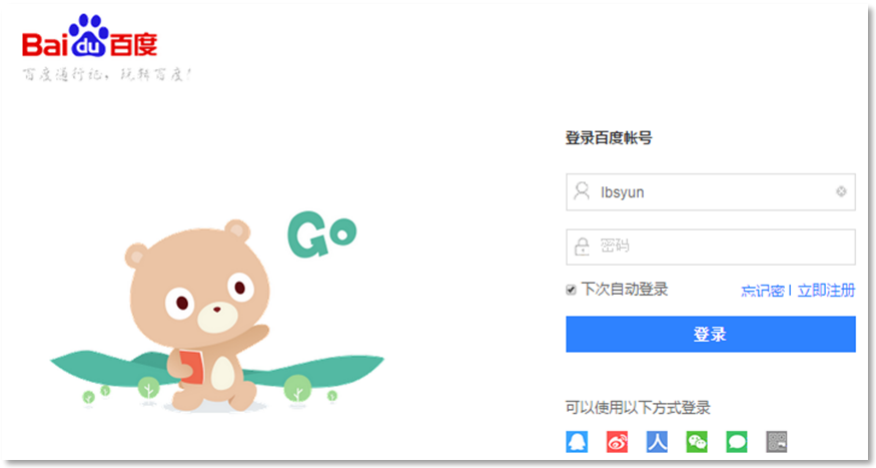

图 1.2.4

2） 开发者激活

如果该百度账号之前没有激活过，则需要进行百度地图开放平台开发者激活。

图 1.2.5

3） 登陆 API 控制台

访问 API 控制台页面 ：http://lbsyun.baidu.com/apiconsole/key

图 1.2.6

4） 创建应用

点击"创建应用"按钮。

图 1.2.7

进入创建 AK 页面，输入应用名称，将应用类型改为："Android SDK"。

图 1.2.8

在应用类型选为“Android SDK”后，需要配置应用的安全码，配置安全码需要获取发布版 SHA1、开发版 SHA1、包名三部分（获取方法见后）。

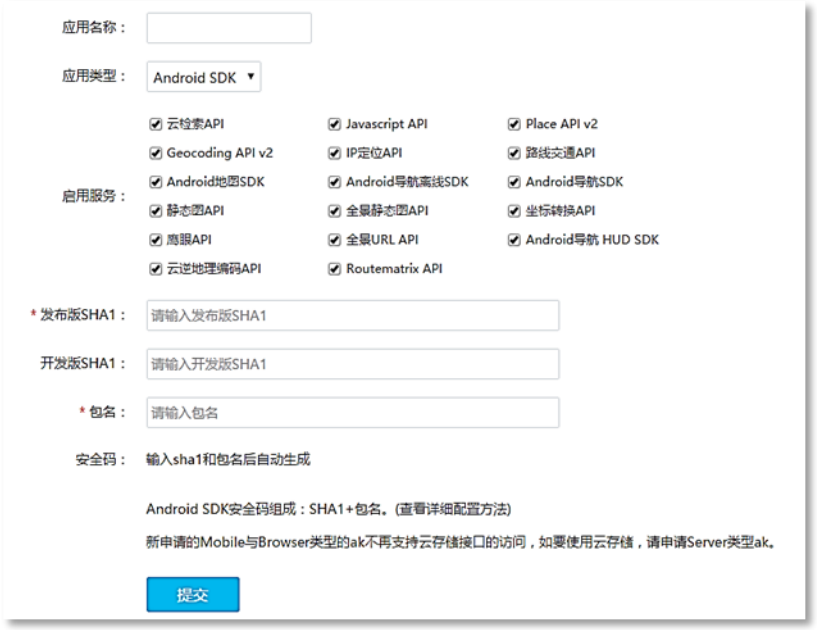

图 1.2.9

在输入安全码后，点击“确定”完成应用的配置工作。

图 1.2.10

此时将会得到一个创建的 AK(API Key)，请妥善保管所申请的 Key。到这就可以使用新 Key 来完成开发。

图 1.2.11

5) 获取安全码

安全码的组成规则为： Android 签名证书的 sha1 值 + packagename(包名)

例如：

SHA1：(以实际值为准)

BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75

包名：(以实际值为准)

com.baidumap.demo

获取包名：

使用 Android Studio 开发时，包名需要在文件 build.gradle 中查询 applictionId。

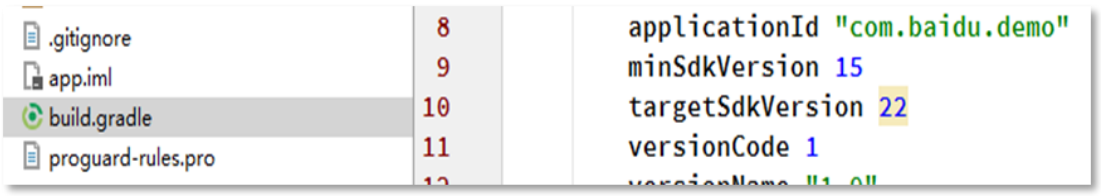

图 1.2.12

获取开发版 SHA1 方法一：

Android Studio 项目界面右侧点击“Gradle”，找到并双击

Tasks->android->signingReport。

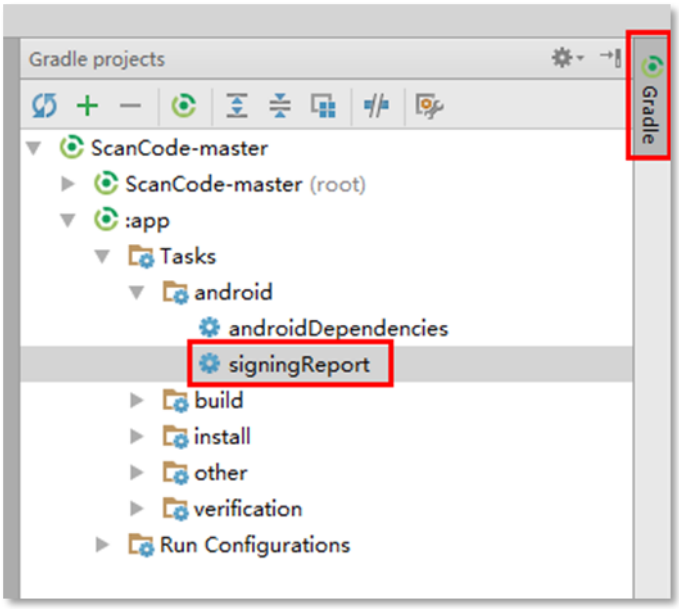

图 1.2.13

在信息输出框中显示开发版的 SHA1。

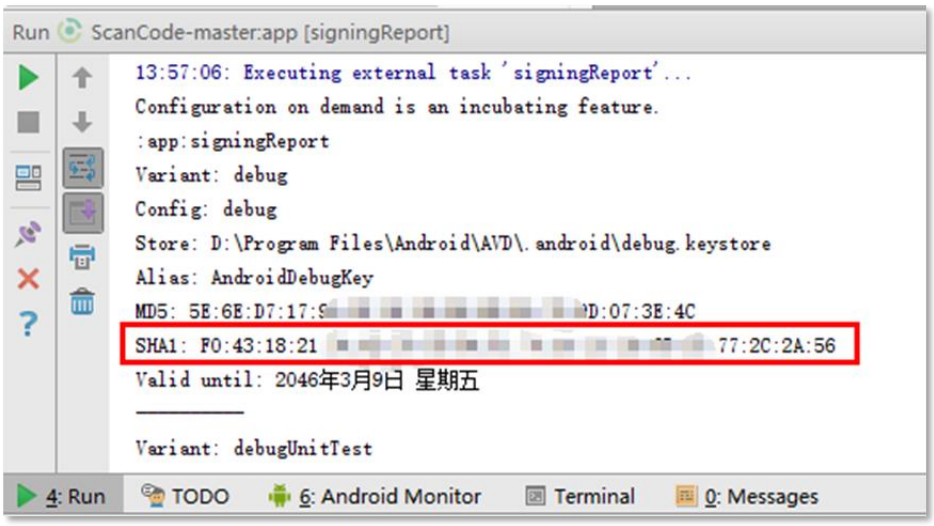

图 1.2.14

获取开发版 SHA1 方法二：

如果在系统环境变量中设置了 ANDROID_SDK_HOME 路径，则打开终端进入到该路径下的\.android 目录下，否则进入 C:\Users\你的用户名\.android 目录下，输入如下命令并回车：

keytool -v -list -keystore debug.keystore

图 1.2.15

获取发布版 SHA1：

在 Android Studio 界面选择 Build -> Generate Signed APK，打开创建 APK 窗口，如果已经创建过则直接 next 既可，如果没有则点击 Create New 弹出窗口进行创建。

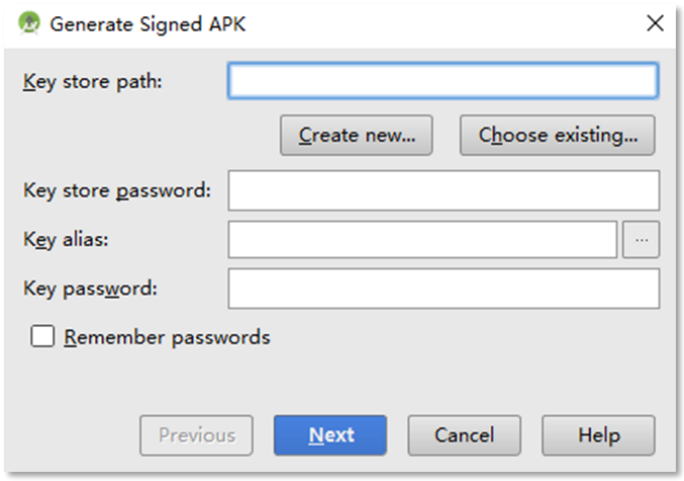

图 1.2.16

在 APK 创建页面输入 Key 信息。

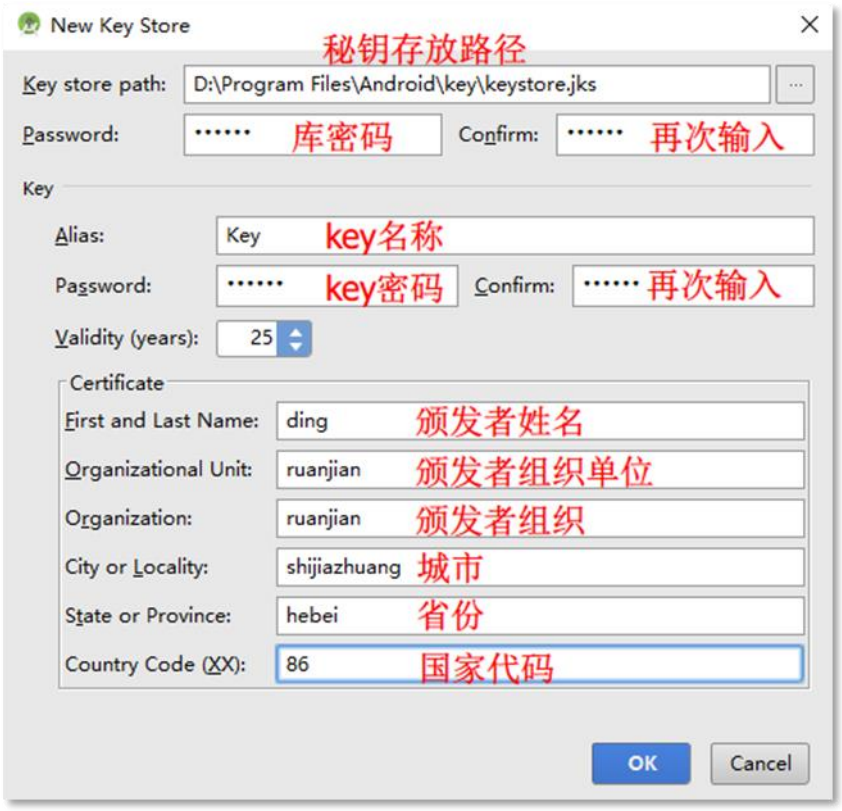

图 1.2.17

选择刚刚创建的 Key，并输入库密码，Key 名称，Key 密码，然后点击 Next。

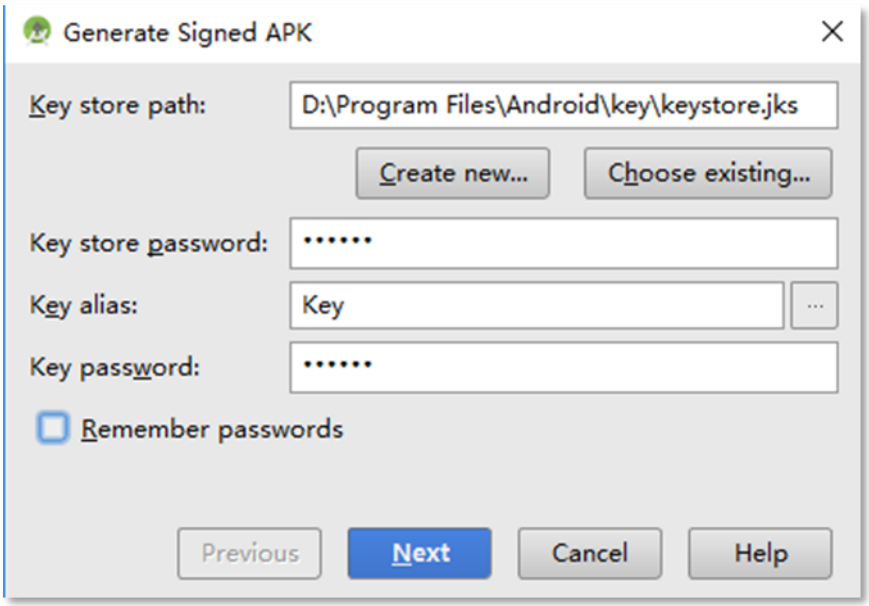

图 1.2.18

在 Build Type 中选择 release，并点击 Finish。

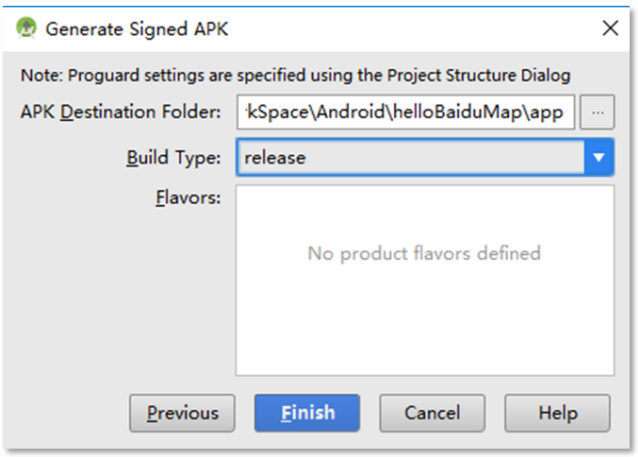

图 1.2.19

在 Android Studio 界面中依次点击 File -> Project Structure，然后按下图将创建好的.jks 文件添加到 build.gradle 中。

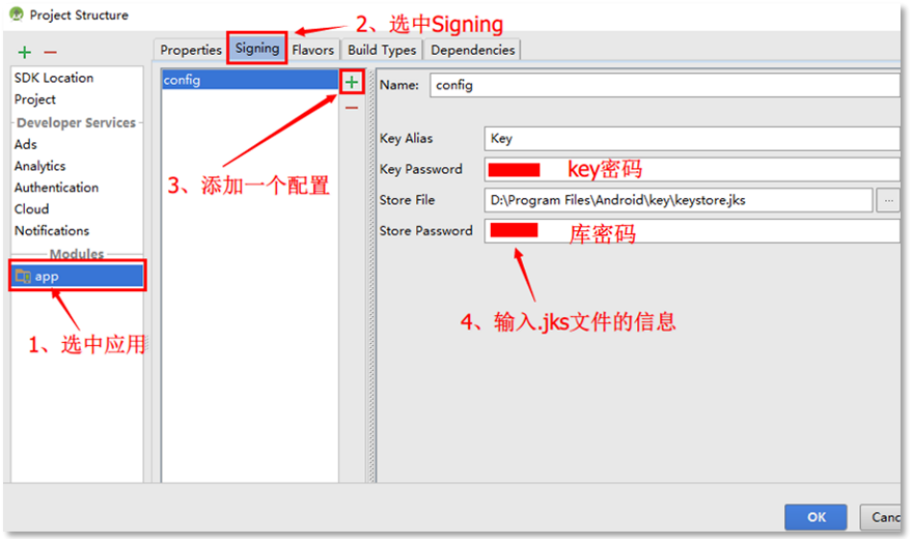

图 1.2.20

此时在 build.gradle 中会出现刚才创建的签名文件的相关信息。

```
apply plugin: 'com.android.application'

android {
    signingConfigs {
        config {
            keyAlias 'Key'
            keyPassword '[redacted]'
            storeFile file('D:/Program Files/Android/key/keystore.jks')
            storePassword '[redacted]'
        }
    }
```

图 1.2.21

在终端中移动到保存之前创建好的签名文件所在路径，然后输入如下指令即可获取发布版的 SHA1。

keytool -list -v -keystore keystore.jks

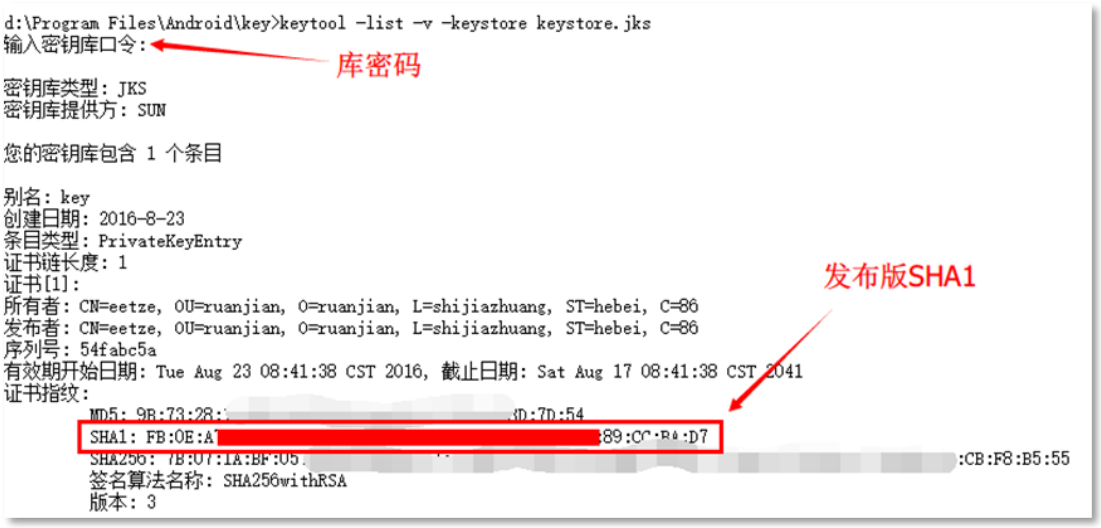

图 1.2.22

准备三： 配置环境

1） 将之前下载的SDK开发库文件进行解压缩得到的libs文件夹里面的文件全部引入到项目工程中。

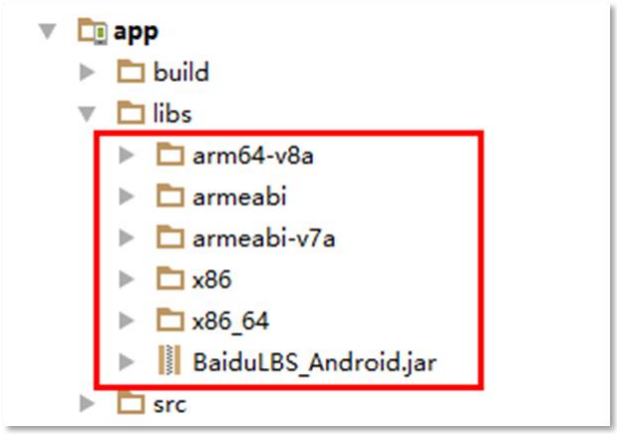

图 1.2.23

2） 在 build.gradle 中设置so文件路径，在android节点下添加如下内容。

```
apply plugin: 'com.android.application'

android {
    sourceSets {
        main {
            jniLibs.srcDirs = ['libs']
        }
    }
}
```

图 1.2.24

3) 加载刚才引入的 jar 文件，在项目结构列表中右键点击 BaiduLBS_Android.jar 文件，选择 Add As Library，导入到工程中。对应在 build.gradle 生成工程所依赖的 jar 文件说明。

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.2.1'
    compile files('libs/BaiduLBS_Android.jar')
}
```

图 1.2.25

4) 此时，Android studio 百度地图工程配置完成。

1.3 实验步骤

步骤一 创建 hellloBaiduMap 工程，并按前面方法完成配置。

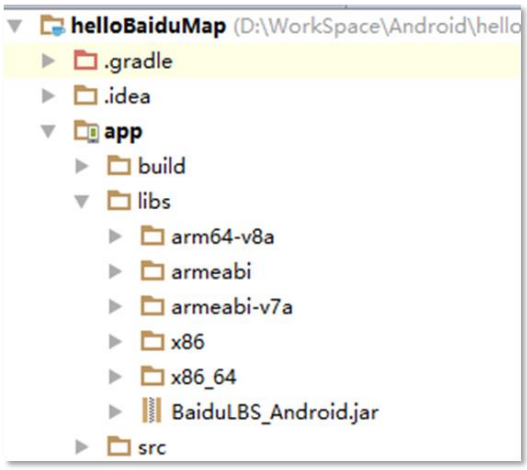

图 1.3.1

步骤二 申请 AK，并在 AndroidManifest.xml 中 application 节点中配置 AK 的信息。

```
<meta-data
   android:name="com.baidu.lbsapi.API_KEY"
   android:value="开发者 key" />
```

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="helloBaiduMap"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <meta-data
        android:name="com.baidu.lbsapi.API_KEY"
        android:value="Gwd5c3y[redacted]Etuc" />
```

图 1.3.2

步骤三 在 AndroidManifest.xml 中添加所需权限信息。

```
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission
android:name="com.android.launcher.permission.READ_SETTINGS" />
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission  android:name="android.permission.CHANGE_WIFI_STATE"
/>
<uses-permission  android:name="android.permission.ACCESS_WIFI_STATE"
/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ruanjian.hellobaidumap">

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="com.android.launcher.permission.READ_SETTINGS" />
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.GET_TASKS" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS" />
```

图 1.3.3

步骤四 在布局 xml 文件中添加地图控件。

```
<com.baidu.mapapi.map.MapView
```

```
    android:id="@+id/bmapView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clickable="true" />
```

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.ruanjian.hellobaidumap.MainActivity">

    <com.baidu.mapapi.map.MapView
        android:id="@+id/bmapView"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:clickable="true" />
</RelativeLayout>
```

图 1.3.4

步骤五 在应用程序创建时初始化 SDK 引用的 Context 全局变量，创建地图 Activity，管理地图生命周期。

```
public class MainActivity extends AppCompatActivity {
    MapView mMapView = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // 在使用 SDK 各组件之前初始化 context 信息，传入
ApplicationContext
        // 注意该方法要再 setContentView 方法之前实现
        SDKInitializer.initialize(getApplicationContext());
        setContentView(R.layout.activity_main);
        // 获取地图控件引用
        mMapView = (MapView) findViewById(R.id.bmapView);
}
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // 在 activity 执行 onDestroy 时执行 mMapView.onDestroy()，
实现地图生命周期管理
        mMapView.onDestroy();
    }
    @Override
    protected void onResume() {
        super.onResume();
```

```
        // 在 activity 执行 onResume 时执行 mMapView. onResume (),
实现地图生命周期管理
        mMapView.onResume();
    }
    @Override
    protected void onPause() {
        super.onPause();
        // 在 activity 执行 onPause 时执行 mMapView. onPause (), 实现
地图生命周期管理
        mMapView.onPause();
    }
}
```

1.4 实验结论

当编码工作完成后在模拟器或真机中运行项目，效果如下：

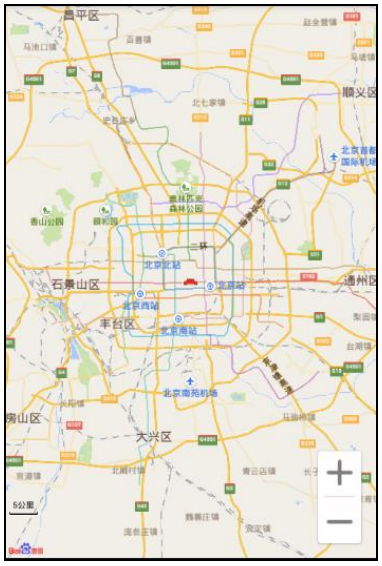

图 1.4.1