



河北师范大学软件学院  
Software College of Hebei Normal University

# 百度地图实战



Android教研室



# 自我介绍

丁盟

qq: 2622885094





## ● 百度地图简介



● SDK下载

● 申请秘钥

● 配置环境

● 第一个百度地图应用





## 百度地图简介



百度地图API是为开发者**免费**提供的一套**基于百度地图服务**的应用接口，包括JavaScript API、Web服务API、Android SDK、iOS SDK、定位SDK、车联网API、LBS云等多种开发工具与服务，提供基本地图展现、搜索、定位、逆/地理编码、路线规划、LBS云存储与检索等功能，适用于PC端、移动端、服务器等多种设备，多种操作系统下的地图应用开发。



# 百度地图简介



Android地图SDK

Android定位SDK

Android鹰眼Android SDK

Android导航SDK

Android HUD SDK

Android全景SDK



## 百度地图简介 - Android地图SDK

**百度地图 Android SDK**是一套基于  
**Android 2.1及以上版本**设备的应用程序接口。  
您可以使用该套 SDK开发适用于Android系  
统移动设备的地图应用，通过调用地图SDK  
接口，您可以轻松访问百度地图服务和数据，  
构建功能丰富、交互性强的地图类应用程序。





## 百度地图简介 - Android地图SDK

- 地图
- POI检索
- 地理编码
- 线路规划
- 地图覆盖物
- 定位
- 离线地图
- 调启百度地图
- 周边雷达
- LBS云
- 特色功能
- 个性化地图
- 室内图
- Android Wear





# 百度地图简介 - Android定位SDK

**百度地图Android定位SDK**是为Android移动端应用提供的一套简单易用的LBS定位服务接口，专注于为广大开发者提供最好的综合定位服务，通过使用百度定位SDK，开发者可以轻松为应用程序实现智能、精准、高效的定位功能。

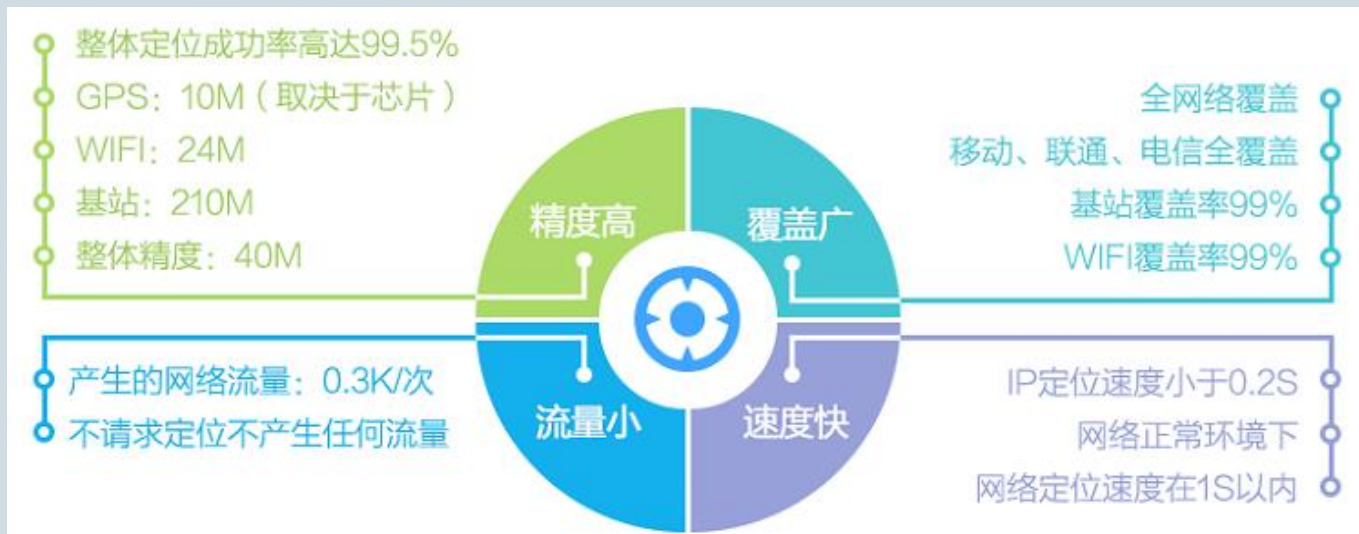






## 百度地图简介 - Android定位SDK

百度地图Android定位SDK提供GPS，基站，Wi-Fi等多种定位方式，适用于室、内外多种定位场景，具有出色的定位性能：定位精度高、覆盖率广、网络定位请求流量小、定位速度快。





# 百度地图简介

❖ LBS ( 全称Location Based Services ) 可以理解  
为移动定位服务。





● 百度地图简介

● SDK下载 

● 申请秘钥

● 配置环境

● 第一个百度地图应用
















## SDK下载

➤ 百度地图 Android SDK 开发资源下载平台

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

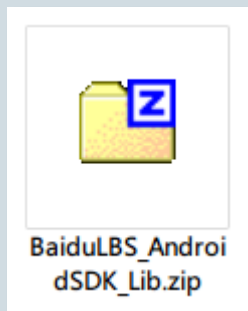


# SDK下载

 基础定位 ✓	 离线定位 ✓	 室内定位 ✓	 全量定位 ✓
 基础地图(含室内图) ✓	 检索功能 ✓	 LBS云检索 ✓	 计算工具 ✓
 周边雷达 ✓	 导航功能(有TTS) ✓	 全景图功能 ✓	
<div>↓ 开发包</div> <div>↓ 示例代码</div> <div>↓ 类参考</div>			



# SDK下载



libBaiduMapSDK\_base\_v4\_0\_0.so

类型: SO 文件



libBaiduMapSDK\_cloud\_v4\_0\_0.so

类型: SO 文件



libBaiduMapSDK\_map\_v4\_0\_0.so

类型: SO 文件



libBaiduMapSDK\_radar\_v4\_0\_0.so

类型: SO 文件



libBaiduMapSDK\_search\_v4\_0\_0.so

类型: SO 文件



libBaiduMapSDK\_util\_v4\_0\_0.so

类型: SO 文件



liblocSDK7.so

类型: SO 文件

## 地图SDK

## 定位SDK



● 百度地图简介

● SDK下载

● 申请秘钥 

● 配置环境

● 第一个百度地图应用





## 申请密钥

在使用百度地图SDK为您提供提供各种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>





# 申请秘钥 - 登录百度账号

## 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。

Baidu 百度  
百度通行证，玩转百度!

Go

登录百度帐号

lbsyun

密码

☒ 下次自动登录 [忘记密码!](#) [立即注册](#)

登录

可以使用以下方式登录

Icons for various login methods: WeChat, QQ, etc.



# 申请秘钥 - 开发者激活

## 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。

我的应用

查看应用

创建应用

回收站

我的服务

查看服务

我的数据

数据管理平台

开发者信息

我要认证

记录中心

认证资料

反馈

完善资料

认证状态：未认证

百度地图开放平台开发者激活

\* 姓名：

\* 手机：

\* 邮箱：

☒ 我已阅读并同意 [《百度地图开放平台服务条款》](#)



# 申请秘钥 - 登陆API控制台

## 3、登陆API控制台

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

我的应用

查看应用

创建应用

回收站

我的服务

查看服务

我的数据

数据管理

开发者信息

完善资料

➔ 应用列表

每页显示30条 ▼

应用编号	应用名称	访问应用 ( AK )	应用类别	备注信息 ( 双击更改 )	应用配置
3849961	Test for Geotrace		服务端		设置 删除
3352411	纸巾巾纸		服务端		设置 删除
2788477	yutt		服务端		设置 删除



# 申请秘钥 - 创建应用

## 4、创建应用

点击"创建应用"按钮。

我的应用

查看应用

创建应用

回收站

我的服务

查看服务

我的数据

数据管理

开发者信息

完善资料

应用列表

请输入AK

搜索

创建应用

回收站

每页显示30条

应用编号	应用名称	访问应用 ( AK )	应用类别	备注信息 ( 双击更改 )	应用配置
3849961	Test for Geotrace		服务端		设置 删除
3352411	纸巾巾纸		服务端		设置 删除



## 申请秘钥 - 创建应用

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

➔ 创建应用

应用名称：

应用类型：

Android SDK ▼

服务器  
**Android SDK**  
iOS SDK v2  
浏览器

启用服务：

☒ 路线交通API

☒ Android地图SDK

☒ 静态图API

☒ 坐标转换API

☒ Javascript API

☒ IP定位API

☒ Android地图SDK SDK

☒ 全景图API

☒ Place API v2

☒ 车联网API

☒ Android导航离线

安全码：

Android SDK安全码组成：数字签名+;+包名。(查看详细配置方法)

新申请的Mobile与Browser类型的ak不再支持云存储接口的访问，如要使用云存储，请申请Server类型ak。

提交



## 申请秘钥 - 创建应用

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）。

应用名称：

应用类型：

Android SDK

启用服务：

☒ 云检索API

☒ Javascript API

☒ Place API v2

☒ Geocoding API v2

☒ IP定位API

☒ 路线交通API

☒ Android地图SDK

☒ Android导航离线SDK

☒ Android导航SDK

☒ 静态图API

☒ 全景静态图API

☒ 坐标转换API

☒ 鹰眼API

☒ 全景URL API

☒ Android导航 HUD SDK

☒ 云逆地理编码API

☒ Routematrix API

\* 发布版SHA1：

开发版SHA1：

\* 包名：

安全码：

输入sha1和包名后自动生成

Android SDK安全码组成：SHA1+包名。(查看详情配置方法)

新申请的Mobile与Browser类型的ak不再支持云存储接口的访问，如要使用云存储，请申请Server类型ak。

提交



# 申请秘钥

在输入安全码后，点击“确定”完成应用的配置工作。

应用名称：

应用类型：

启用服务：

<input checked="" type="checkbox"/> 云检索API	<input checked="" type="checkbox"/> Javascript API	<input checked="" type="checkbox"/> Place API v2
<input checked="" type="checkbox"/> Geocoding API v2	<input checked="" type="checkbox"/> IP定位API	<input checked="" type="checkbox"/> 路线交通API
<input checked="" type="checkbox"/> Android地图SDK	<input checked="" type="checkbox"/> Android导航离线SDK	<input checked="" type="checkbox"/> Android导航SDK
<input checked="" type="checkbox"/> 静态图API	<input checked="" type="checkbox"/> 全景静态图API	<input checked="" type="checkbox"/> 坐标转换API
<input checked="" type="checkbox"/> 鹰眼API	<input checked="" type="checkbox"/> 全景URL API	<input checked="" type="checkbox"/> Android导航 HUD SDK
<input checked="" type="checkbox"/> 云逆地理编码API	<input checked="" type="checkbox"/> Routematrix API	

\* 发布版SHA1：

开发版SHA1： 输入正确

\* 包名：

安全码：

Android SDK安全码组成：SHA1+包名。(查看详细配置方法)

新申请的Mobile与Browser类型的ak不再支持云存储接口的访问，如要使用云存储，请申请Server类型ak。



## 申请秘钥

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

应用列表 认证状态：未认证

搜索

创建应用 回收站 每页显示30条 ▼

应用编号	应用名称	访问应用 ( AK )	应用类别	备注信息 ( 双击更改 )	应用配置
8535165	helloBaiduMap	Gwd5[REDACTED]xNnEtuc	Android端		<a href="#">设置</a> <a href="#">删除</a>

您当前创建了 1 个应用 < 1 >





## 申请秘钥 - 获取安全码

### 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



## 申请秘钥 - 获取包名

### 获取包名

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

 .gitignore	8	
 app.iml	9	
 build.gradle	10	applicationId "com.baidu.demo"
 proguard-rules.pro	11	minSdkVersion 15
	12	targetSdkVersion 22
		versionCode 1
		versionName "1.0"

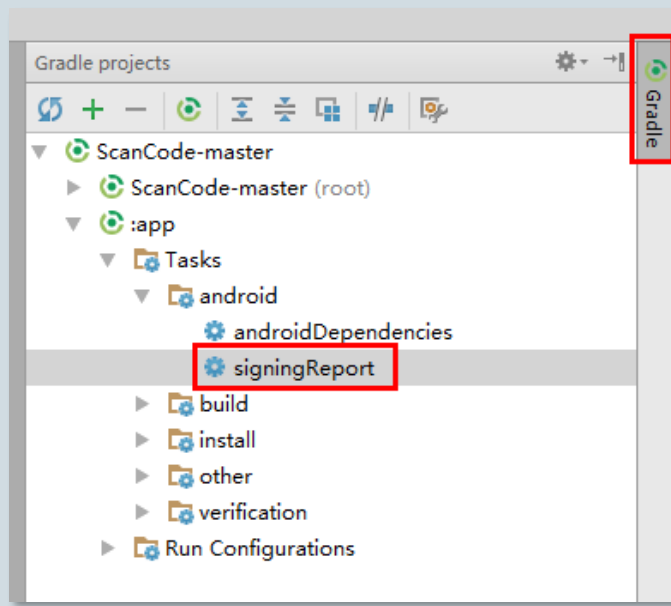


## 申请秘钥 - 获取开发版SHA1

### 获取开发版SHA1

#### 方法一：

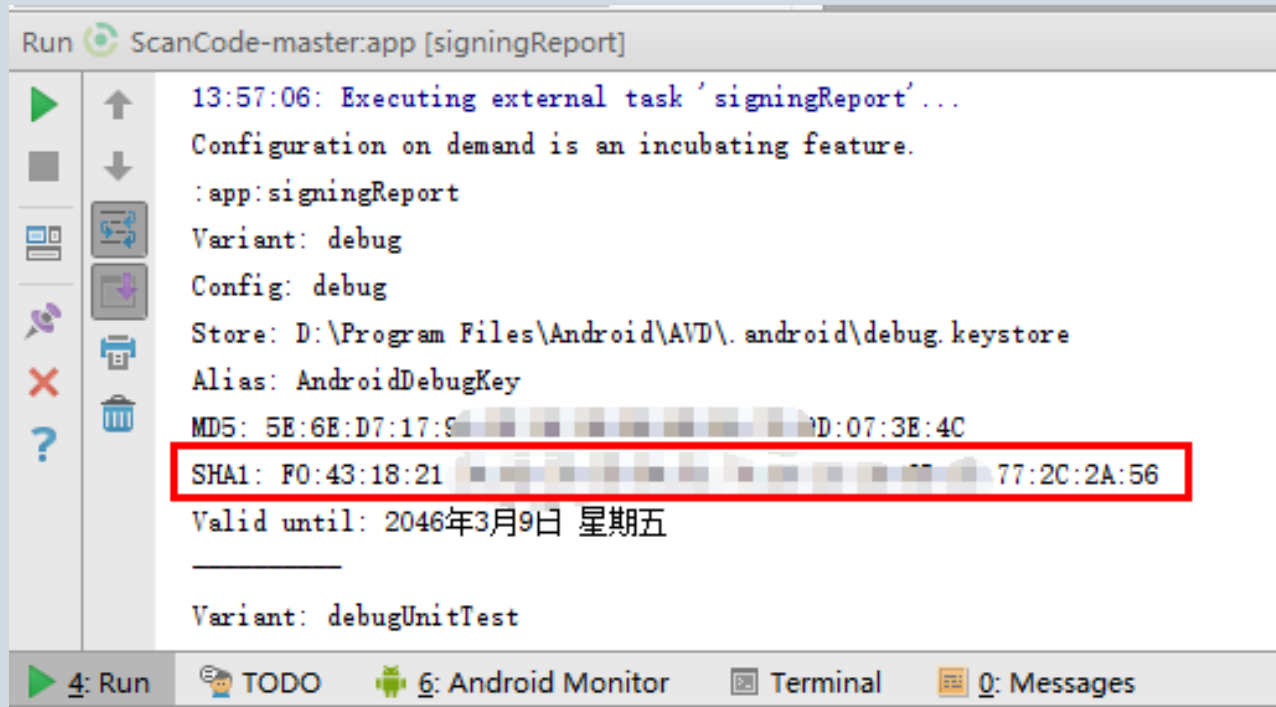
Android Studio 项目界面右侧点击 “Gradle” ,  
找到并双击 Tasks->android->signingReport





## 申请秘钥 - 获取开发版SHA1

在信息输出框中显示开发版的**SHA1**



```
Run ScanCode-master:app [signingReport]

13:57:06: Executing external task 'signingReport'...
Configuration on demand is an incubating feature.
:app:signingReport
Variant: debug
Config: debug
Store: D:\Program Files\Android\AVD\.android\debug.keystore
Alias: AndroidDebugKey
MD5: 5E:6E:D7:17:9...D:07:3E:4C
SHA1: F0:43:18:21...77:2C:2A:56
Valid until: 2046年3月9日 星期五

Variant: debugUnitTest
```



## 申请秘钥 - 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**

```
d:\\Program Files\\Android\\AVD\\.android>keytool -v -list -keystore debug.keystore
```

输入密钥库口令：

密钥库类型：JKS

密钥库提供方：SUN

您的密钥库包含 1 个条目

口令默认：android

别名：androiddebugkey

创建日期：2016-3-16

条目类型：PrivateKeyEntry

证书链长度：1

证书[1]：

所有者：C=US, O=Android, CN=Android Debug

发布者：C=US, O=Android, CN=Android Debug

序列号：1

有效期开始日期：Wed Mar 16 16:30:56 CST 2016, 截止日期：Fri Mar 09 16:30:56 CST 2046

证书指纹：

MD5: 5E:6E:D7:1

:07:3E:4C

SHA1: F0:43:18:21:

55:77:2C:2A:56

SHA256: 83:A9:A0:34:DC

签名算法名称：SHA1withRSA

版本：1

开发版SHA1



## 申请秘钥 - 获取发布版SHA1

### 获取发布版SHA1

1、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。

Generate Signed APK

Key store path:   

Key store password:

Key alias:  ...

Key password:

☐ Remember passwords



# 申请密钥 - 获取发布版SHA1

## 2、在APK创建页面输入Key信息。

**New Key Store**

**密钥存放路径**

Key store path: D:\Program Files\Android\key\keystore.jks

Password: ..... **库密码** Confirm: ..... **再次输入**

**Key**

Alias: Key **key名称**

Password: ..... **key密码** Confirm: ..... **再次输入**

Validity (years): 25

**Certificate**

First and Last Name: ding **颁发者姓名**

Organizational Unit: ruanjian **颁发者组织单位**

Organization: ruanjian **颁发者组织**

City or Locality: shijiazhuang **城市**

State or Province: hebei **省份**

Country Code (XX): 86 **国家代码**

OK Cancel



## 申请秘钥 - 获取发布版SHA1

3、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。

Generate Signed APK

Key store path: D:\Program Files\Android\key\keystore.jks

Create new... Choose existing...

Key store password: .....

Key alias: Key ...

Key password: .....

☐ Remember passwords

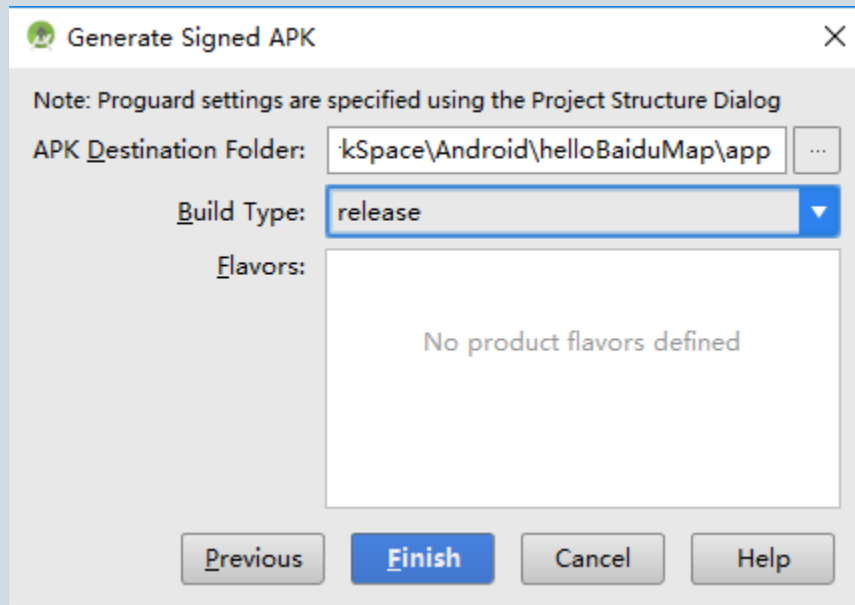
Previous Next Cancel Help





## 申请秘钥 - 获取发布版SHA1

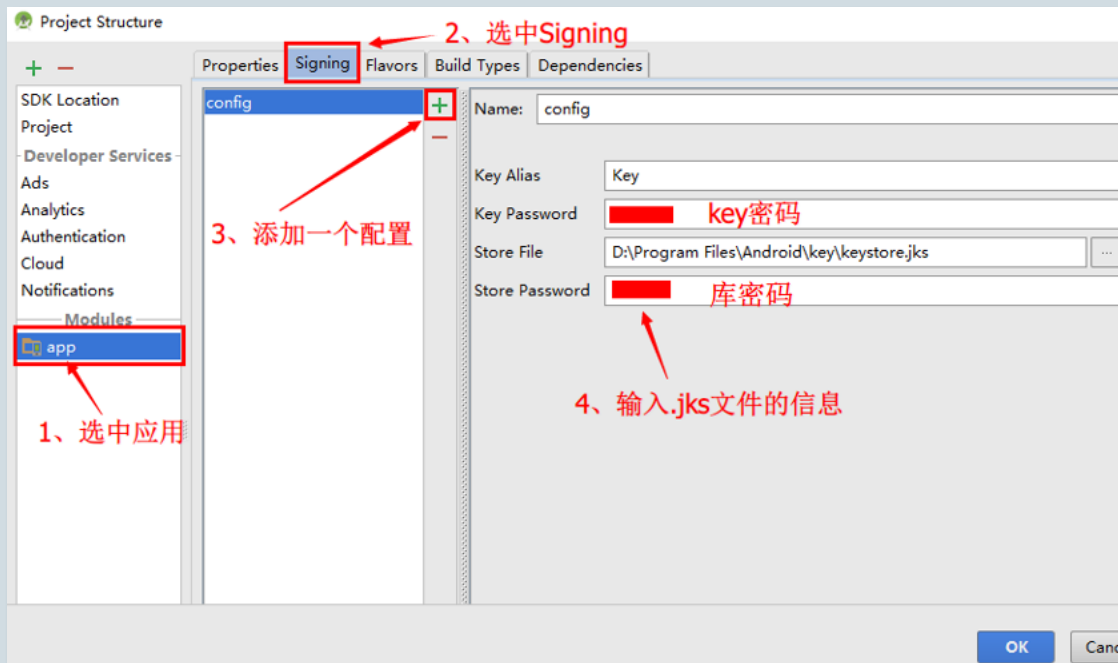
4、在Build Type中选择 release , 并点击Finish。





## 申请秘钥 - 获取发布版SHA1

5、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。





## 申请秘钥 - 获取发布版SHA1

6、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 '██████████'
            storeFile file('D:/Program Files/Android/key/keystore.jks')
            storePassword '██████████'
        }
    }
}
```



## 申请秘钥 - 获取发布版SHA1

7、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**

```
d:\Program Files\Android\key>keytool -list -v -keystore keystore.jks
```

输入密钥库口令:

库密码

密钥库类型: JKS

密钥库提供方: SUN

您的密钥库包含 1 个条目

别名: key

创建日期: 2016-8-23

条目类型: PrivateKeyEntry

证书链长度: 1

证书[1]:

所有者: CN=eetze, OU=ruanjian, O=ruanjian, L=shijiazhuang, ST=hebei, C=86

发布者: CN=eetze, OU=ruanjian, O=ruanjian, L=shijiazhuang, ST=hebei, C=86

序列号: 54fab5a

有效期开始日期: Tue Aug 23 08:41:38 CST 2016, 截止日期: Sat Aug 17 08:41:38 CST 2041

证书指纹:

MD5: 9B:73:28:1...D:7D:54

SHA1: FB:0E:A...89:CC:B4:D7

发布版SHA1

SHA256: 7B:07:1A:BF:05...

:CB:F8:B5:55

签名算法名称: SHA256withRSA

版本: 3



● 百度地图简介

● SDK下载

● 申请秘钥

● 配置环境 

● 第一个百度地图应用





## 配置环境

1、将之前下载的SDK开发库文件进行解压缩得到如下文档结构。



libs



readme.txt



arm64-v8a



armeabi



armeabi-v7a



x86



x86\_64

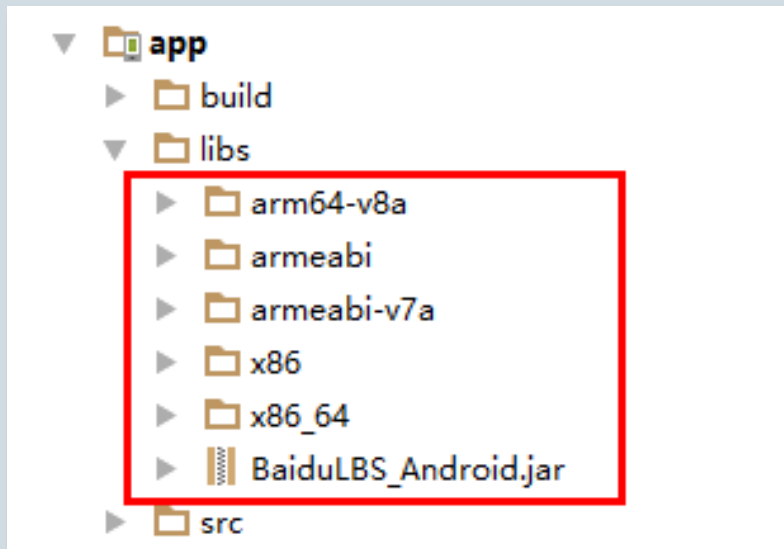


BaiduLBS\_Android.jar



## 配置环境

2、将libs文件夹里面的文件全部引入到项目工程中。





## 配置环境

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

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

```
1  apply plugin: 'com.android.application'  
2  
3  android {  
4      sourceSets {  
5          main {  
6              jniLibs.srcDirs = ['libs']  
7          }  
8      }  
}
```





## 配置环境

4、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build.gradle生成工程所依赖的jar文件说明。

此时，Android studio 百度地图工程配置完成。

```
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')  
}
```



● 百度地图简介

● SDK下载

● 申请秘钥

● 配置环境

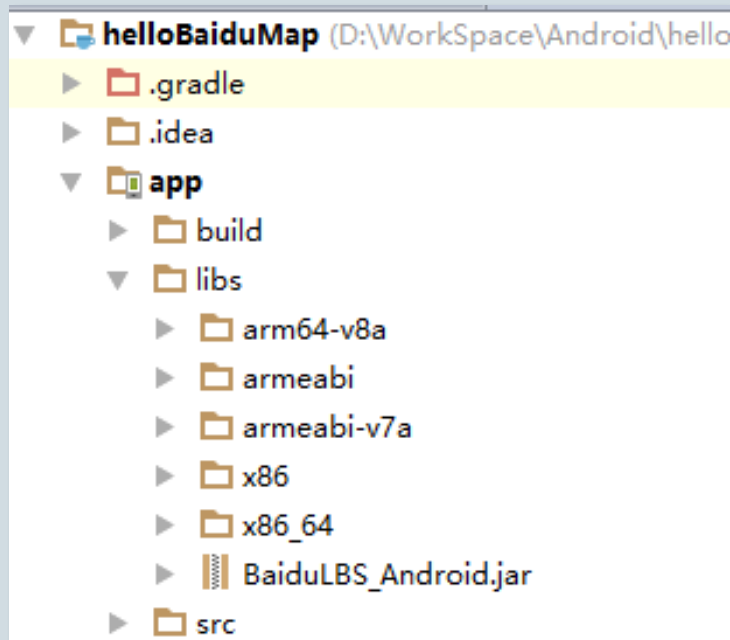
● 第一个百度地图应用





# 第一个百度地图应用

1、创建helloBaiduMap工程，并按前面方法完成配置。





## 第一个百度地图应用

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

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

```
<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_ Etuc" />
```



# 第一个百度地图应用

## 3、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.GET_TASKS" />  
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
```



# 第一个百度地图应用

```
<?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" />
```



# 第一个百度地图应用

## 4、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>
```



## 第一个百度地图应用

5、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);  
    }  
}
```





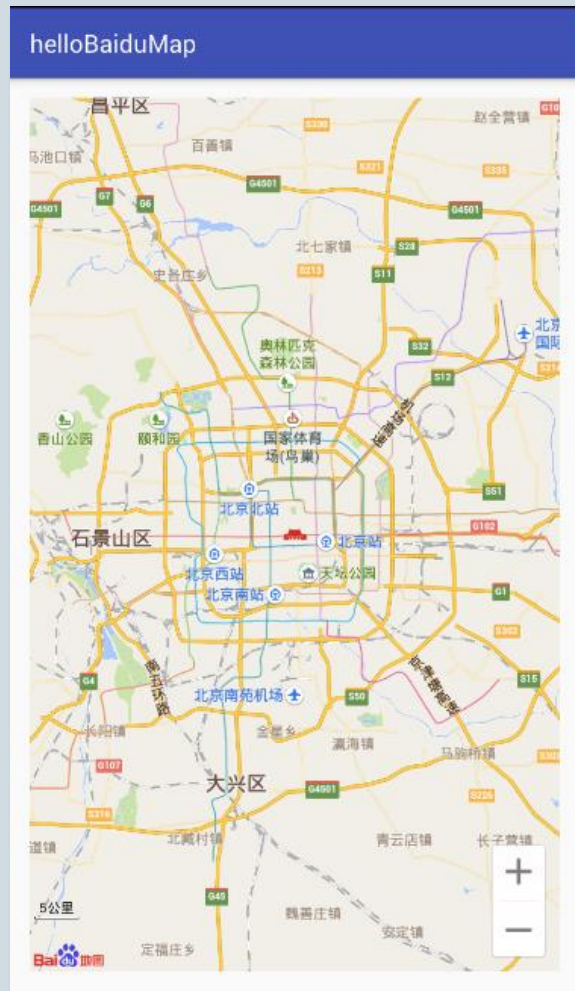
# 第一个百度地图应用

```
@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();
}
}
```



# 第一个百度地图应用

6、运行程序，会出现如图效果。





**Thank You!**

