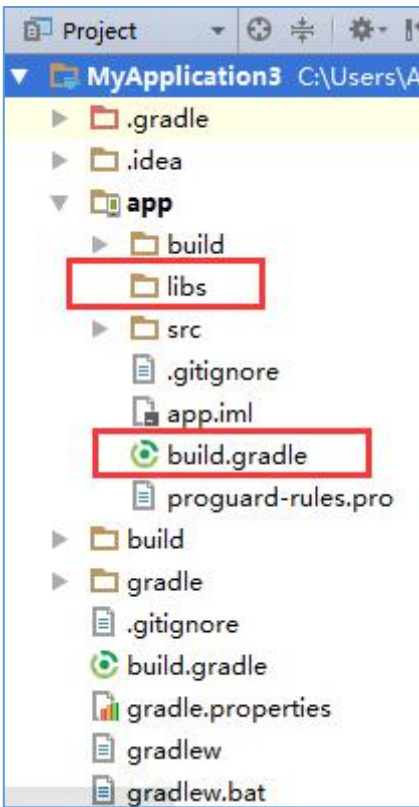


# 1 加密 SDK 集成指南（AAR 版）

## 1.1 AndroidStudio 项目结构目录简介



（项目结构）

如上图其中 app 文件夹是我们的主要 module:

- libs 下存放外部的 library（AAR）;
- build.gradle 是配置 App module，对第三方配置 libs 进行配置。

## 1.2 加固前后的 AAR 说明

### 1.2.1 加固前

名称	类型	大小
LHQPAY_v0102.aar	AAR 文件	1,160 KB

## 1.2.2 加固后

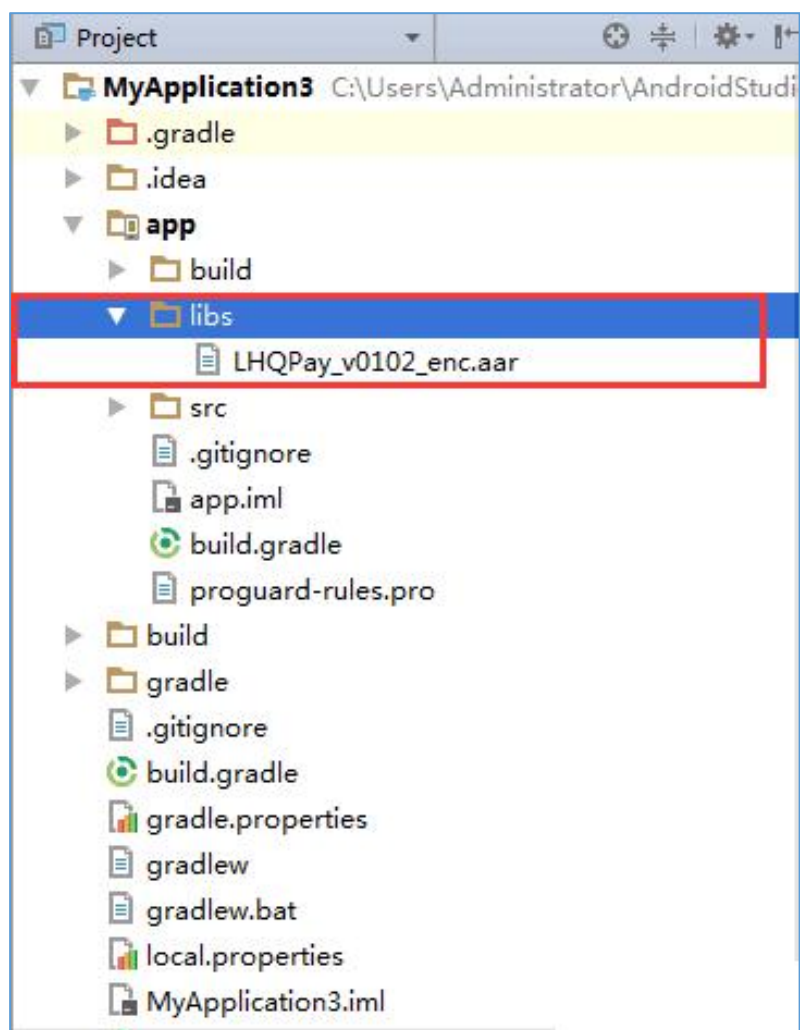
名称	类型	大小
sdknec	文件夹	
LHQPAY_v0102_enc.aar	AAR 文件	2,792 KB

AAR 在加固后会生成 aar 文件与 sdknec/classes.jar 文件，classes.jar 文件为被加固后的框架文件。

## 1.3 引入 SDK

### 1.3.1 拷贝.aar 文件

将加固得到的.aar 文件复制到 app/libs 下面，具体如下图所示：

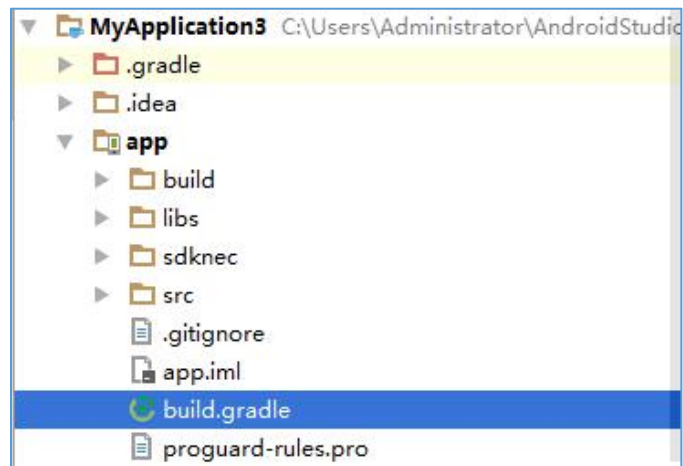


### 1.3.2 配置引用信息

配置 app/build.gradle 文件，加入如下配置信息：

```
repositories{  
    flatDir {  
        dirs 'libs'  
    }  
}  
  
dependencies {  
    compile(name: 'LHQPAY_v0102_enc', ext: 'aar')  
}
```

具体如下图：



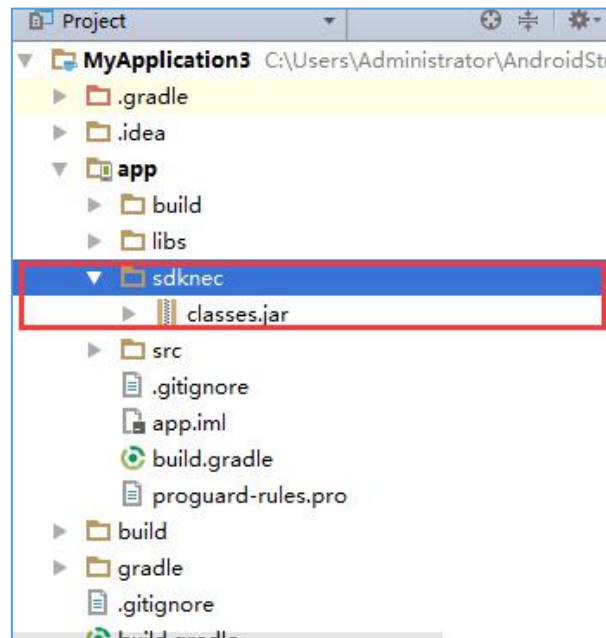
（文件目录地址）

```
repositories {  
    flatDir {  
        dirs 'libs'  
    }  
}  
  
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {  
        exclude group: 'com.android.support', module: 'support-annotations'  
    })  
    compile 'com.android.support:appcompat-v7:25.3.1'  
    compile 'com.android.support.constraint:constraint-layout:1.0.2'  
    testCompile 'junit:junit:4.12'  
    compile(name: 'LHQPAY_v0102_enc', ext: 'aar')  
}
```

(文件配置信息摘要)

### 1.3.3 拷贝 classes.jar 文件

将加固得到的 sdknec/classes.jar 文件复制到 app 下面， 具体如下图所示：

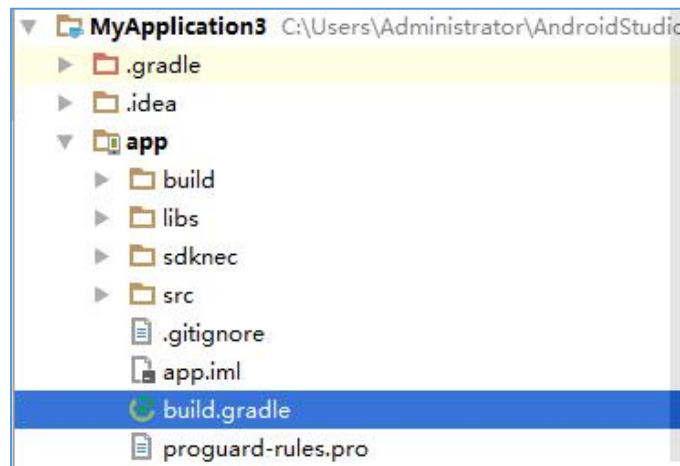


### 1.3.4 设置 classes.jar 不对外导出

在 build.gradle 中设置如下代码：

```
dependencies {  
    provided files('sdknec/classes.jar');  
}
```

具体如下图：



(文件目录结构摘要)

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    testCompile 'junit:junit:4.12'
    compile(name: 'LHQPAY_v0102_enc', ext: 'aar')
    provided files('sdknec/classes.jar')
}
```

(文件配置信息摘要)

## 1.4 应用程序加载 SDK 接口

### 1.4.1 方法一：自定义全局 Application

开发者在集成 SDK 时，务必先自定义全局 Application。Application 里面的代码，必须包含下行代码：`SdkManager.initSdkManager(this);`

示例代码如下：

```
package com.youmi.android.sample;

import android.app.Application;
import android.content.Context;
import com.shell.SdkManager;
```

```
public class MyApplication extends Application {
    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(base);
        SdkManager.initSdkManager(this);
    }
}
```

AndroidManifest.xml 中配置:

```
<application
    android:name="com.youmi.android.sample.MyApplication"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
.....
```

### 1.4.2 方法二：引用加密 Application

在 AndroidManifest.xml 中配置代理 Application，  
 "*com.shell.SdkManagerApplication*"。由代理 Application，自动完成 SDK 的  
 加载。

AndroidManifest.xml 中配置:

```
<application
    android:name="com.shell.SdkManagerApplication"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
```

如果用户应用已经自定义了 Application，请在 AndroidManifest.xml 中新增一

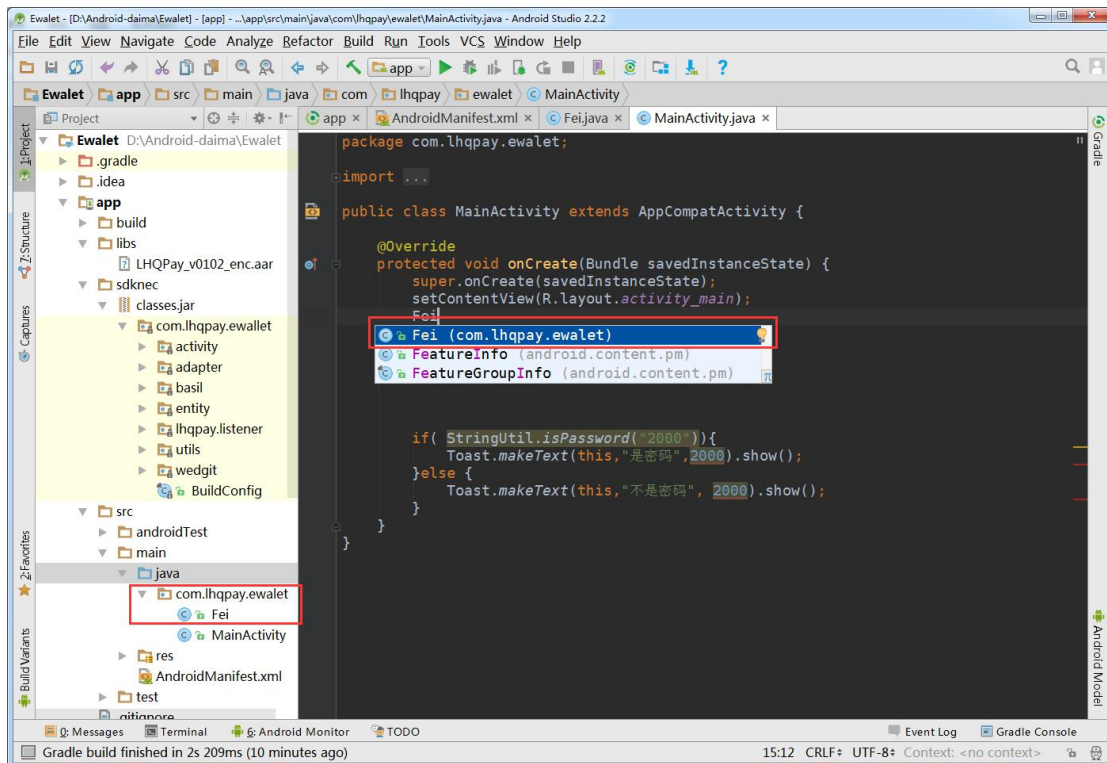
个 name 为 `"com.shell.ApplicationName"` 的 meta-data 节点，记录下应用原 Application 名称。

AndroidManifest.xml 中配置：

```
<application
    android:name="com.shell.SdkManagerApplication"
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <meta-data
        android:name="com.shell.ApplicationName"
        android:value="com.youmi.android.sample.MyApplication" />
    .....
```

## 2 验证加密完成的 aar 文件是否能正常

### 2.1 在工程项目里选择 AAP > src > main > java 下创建一个类。





## 2.2 第二个方式可以写一个判断语句验证加密效果



## 2.3 使用 java Decompiler 工具查看原包 aar 和加密包 aar 做对比

