# Android项目目录介绍

## 简介

一个Android项目是由很多目录或文件构成，每个目录或文件都有相应的作用，开发过程中创建的各个目录或文件，只有严格按照规定放在相应的目录下才能起作用，项目才能正常运行。Android的开发工具很多，不同的开发工具，项目目录结构不一样，本节介绍的是在本课程所用的开发工具Android Studio里这些目录的作用及其使用。

## 二、目标

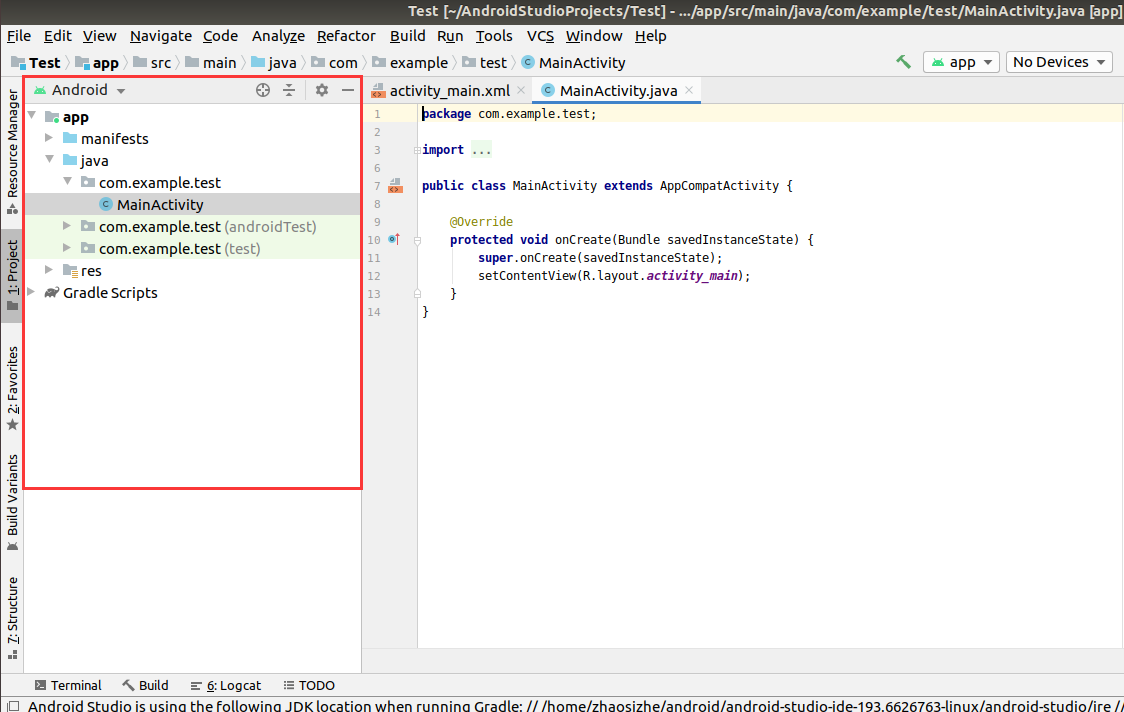
* 熟练掌握4个核心目录
* 掌握2个常用文件的使用
* 了解所有目录或文件的作用

## 三、详细介绍

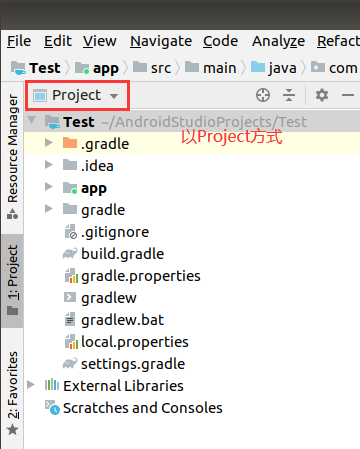
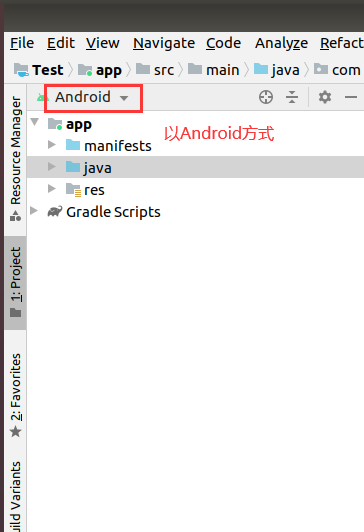
### 1.认识项目目录

#### 1.1 项目目录结构

当用Android Studio工具打开或创建好一个Android项目后，可以看到在界面的左边有展开的目录结构列表，如下图。



这就是Android项目目录结构，这个目录其结构可以选择不同的方式呈现，点击【▼】会弹出选项。默认情况下是以Android方式呈现的，还可以其他方式呈现，比如Project方式，如下图。

这些方式中，Project方式呈现了一个Android项目的所有目录和文件结构，Android方式只呈现了开发人员常用的目录和文件结构。

#### 1.2 检验标准

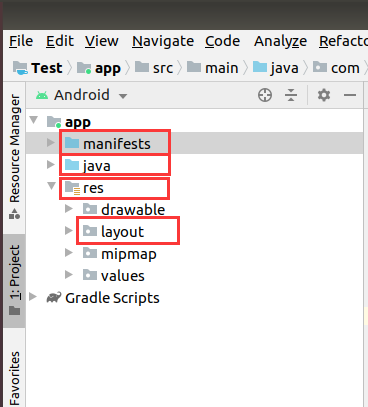
能知道怎么以不同方式打开项目目录，尤其以Android方式和Project方式。

### 4个核心目录

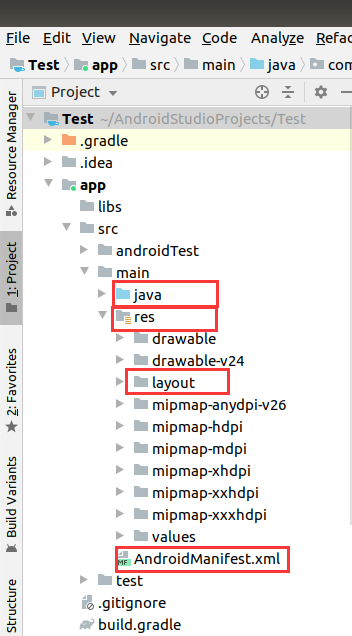
Android项目里的原始目录和文件非常多，作为一个开发人员，必须要掌握的是4个核心目录。

#### 2.1 认识4个核心目录

这个4个核心目录分别是：layout、java、res、manifests。以Android方式呈现目录结构时，他们的位置如下图。



以Project方式呈现目录结构时，他们的位置如下图。



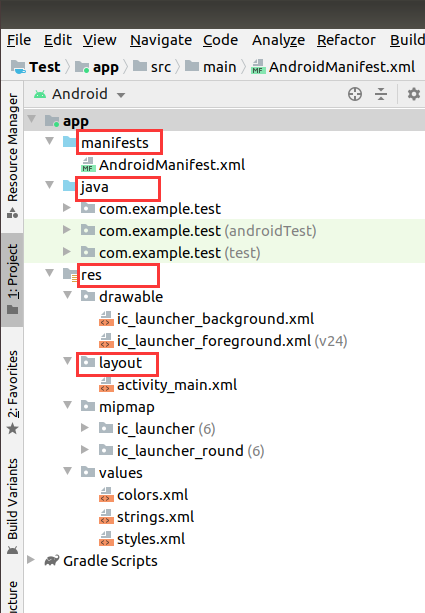
不过，和前面不同的是，以Project方式呈现时，没有manifests目录，只有AndroidManifest.xml文件，这个文件就是manifests目录下的唯一文件，只不过这里没有显示目录名而已。

#### 2.2 4个核心目录的作用

这4个目录或文件是开发时的核心，会频繁使用，他们的作用如下表。

|  |  |
| --- | --- |
| layout | 用于放置App的一些界面设计文件，在Android里，我们一般叫布局文件，以 **.xml**为文件后缀。 |
| java | 用于放置App的java文件，也就是以 **.java**为后缀的文件。 |
| res | 用于放置项目需要的资源文件，比如图片、音频、视频、颜色、文本等等文件,下面的所有文件或目录命名只能小写字母，可以中间有下划线，数字不能在开头。 |
| Androidmanifest.xml | 主要用于配置设置整个项目的标识、权限、软件图标、软件名称和四大核心组件，比如定位权限和蓝牙权限。该文件只能有一个。 |

展开这个4个目录后，他们的结构如下图。



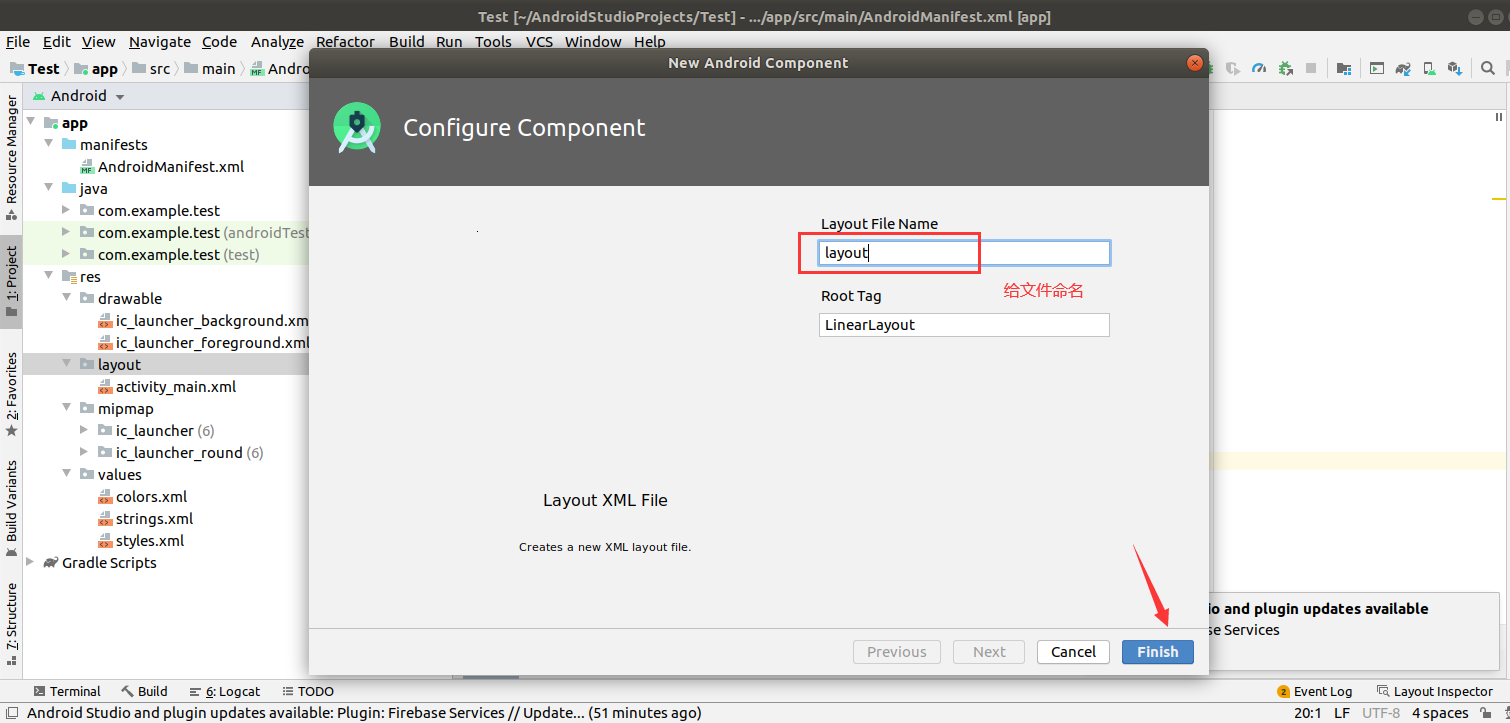
我们可以看到，java目录下面还有子目录，layout目录被包含在res目录下，res目录除了有layout目录外，还有很多子目录，下面详细介绍他们的作用。首先java下面有3个子目录，目录后面括号中有“test”字样的，是放测试时的java文件，没有“test”字样的，放我们平常开发时的java文件，如类、接口。res目录下有4个下级目录:drawable、layout、mipmap、values。另外3个目录根据项目需要往其下面添加相应文件，layout目录下至少要有一个布局文件。drawable目录一般用于放样式文件和图片文件；mipmap目录用于放图片文件；values目录一般用于放值文件，比如颜色值文件、文本值文件、风格文件。音频视频文件，我们一般在res目录下新建一个raw目录来存放。

#### 2.3 4个核心目录的使用

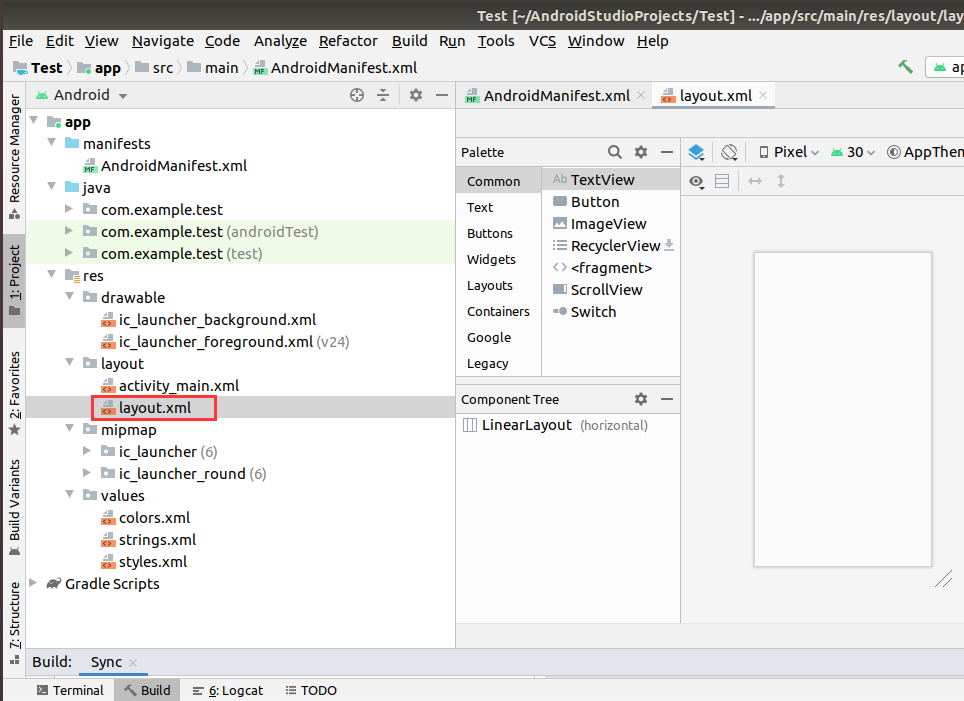
知道这个4个目录的作用后，我们要能精准的把创建的文件放在相应的目录下，下面演示整个过程。

**添加布局文件**

布局文件用于App的界面设计，已有的布局文件都要放在layout目录下面，或者直接新建布局文件：右键点击layout目录>【New】>【XML】>【Layout XML File】,在弹出的界面中给文件命名，点击【Finish】即可，如下图。

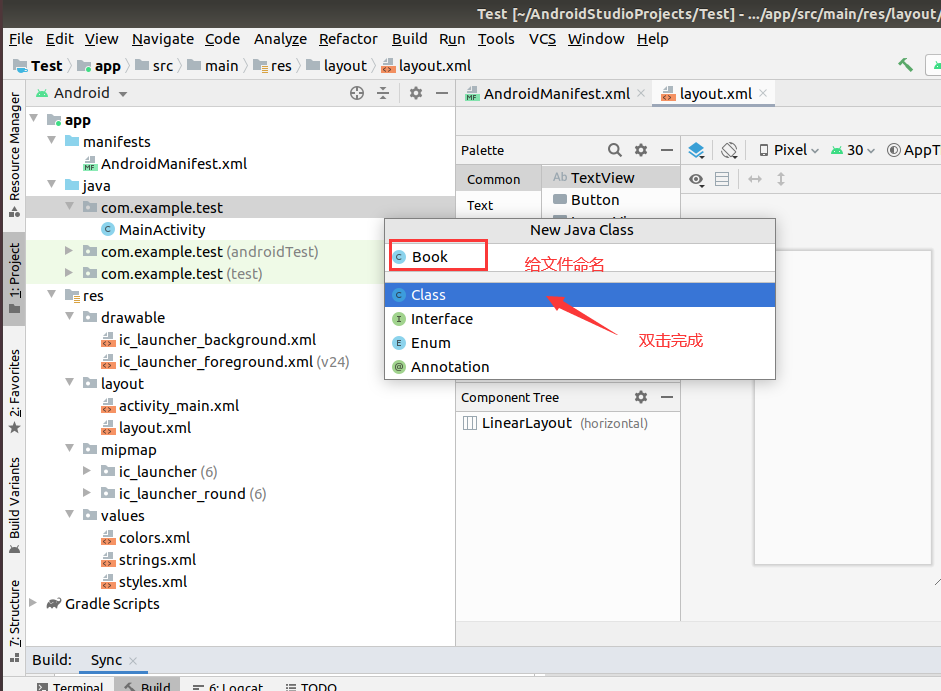


这里把新建的布局文件命名为“layout”，完成后会在layout目录下多出一个**layout.xml**文件，如下图。

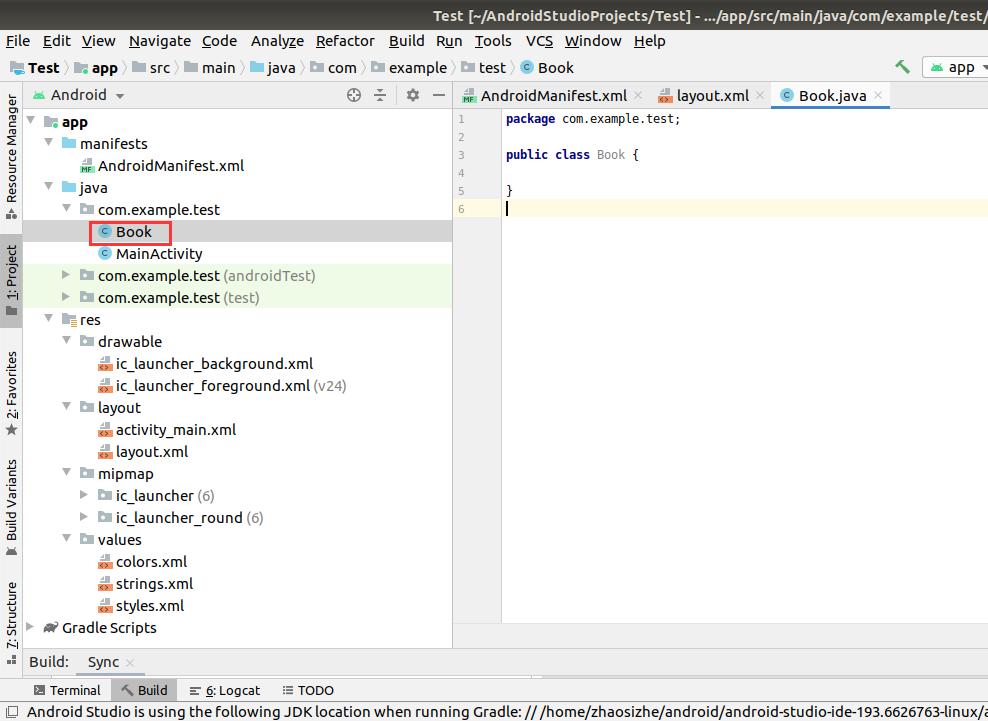


**添加java文件**

java文件用于处理App的功能逻辑，已有的布局文件都要放在java目录下，或直接java文件，比如新建一个类：右键点击java目录下括号里没有”test”字样的子目录>【New】>【Java Class】>在弹出的窗口中给文件命名，双击【Class】即可，如下图。

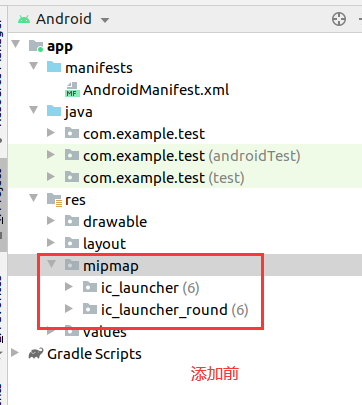
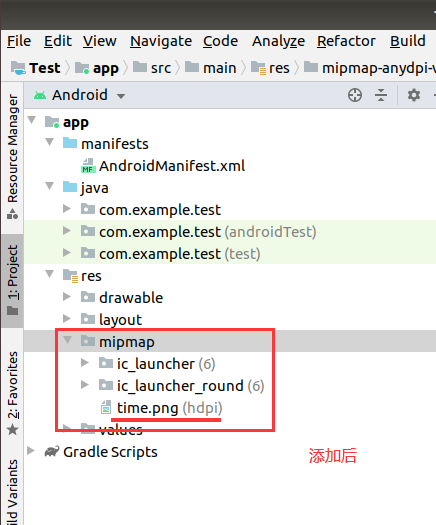


这里把新建的java类命名为Book，完成后，可一看到java目录的子目录下（也就是包）多了一个Book文件，如下图。

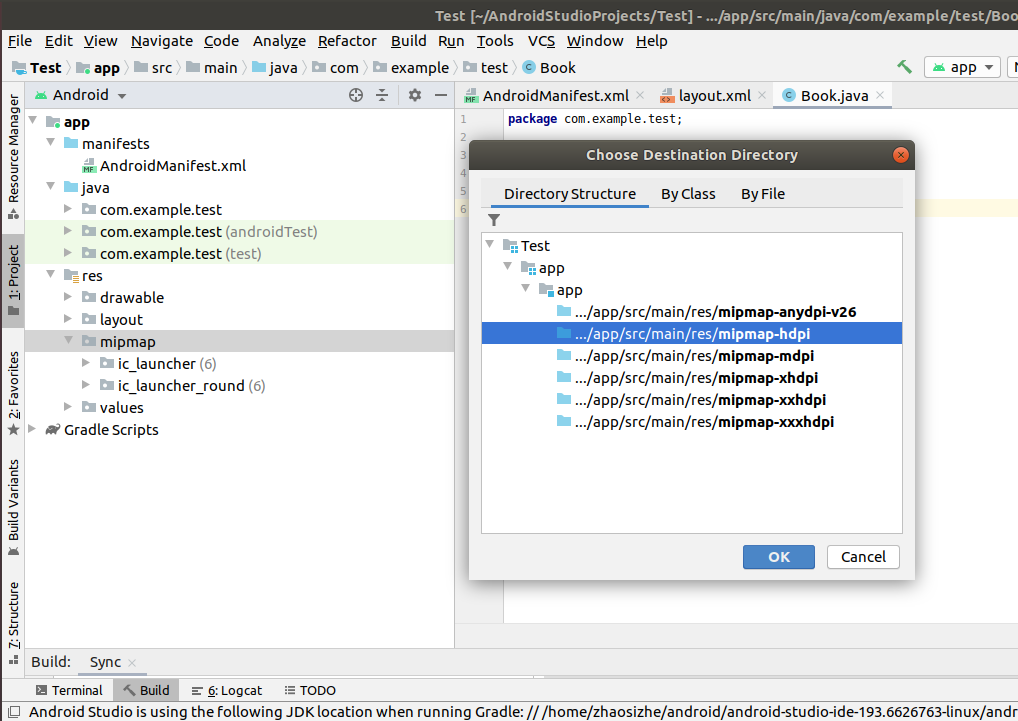


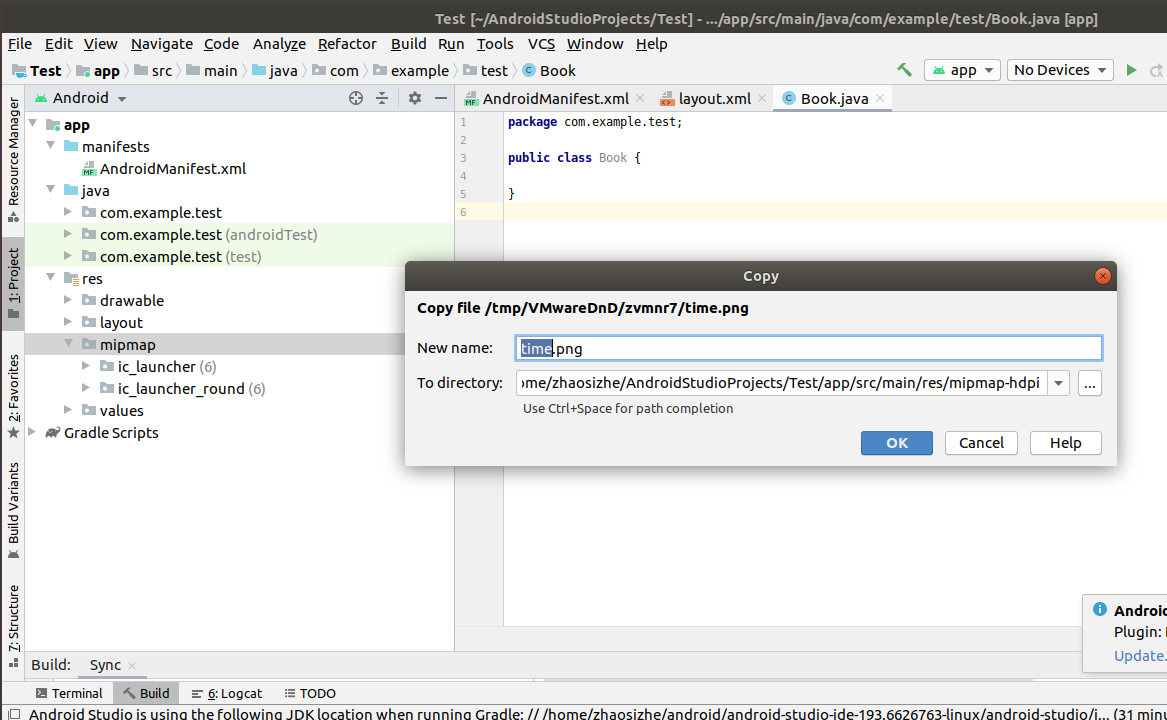
**添加图片文件**

图片几乎是我们App里必不可少的，图片文件可以放在res\drawable目录下，也可以放在res\mipmap目录下，一般我们是放在res\mipmap目录下，只需要把图片文件复制粘贴到下面即可，如下图。

在外部图片粘贴到mipmap目录下时，会弹出一个选择窗口，里面是一些不同大小尺寸的目录，不同大小的图片放在相应尺寸的目录下，一般手机App优先在mipmap-hdpl目录下放置，如果要多屏适配的话，每一个大小尺寸的目录下都要放置一张相应大小的图片。然后点击【ok】确定即可，如下图。





#### 2.4 检验标准

不用参考文档资料就能把布局文件、java文件、图片文件准确的添加到对应目录下，能背出配置文件Androidmanifest.xml的作用，并找到其位置。

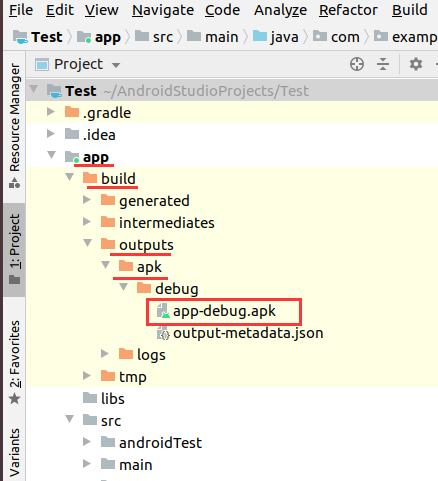
### 2个常用文件

#### 3.1 介绍

Android项目除了上面频繁使用的4个核心目录或文件外，还有2个文件有时也会常用，第1个是**.apk**文件，第2个是**build.gradle**文件。

**.apk文件**

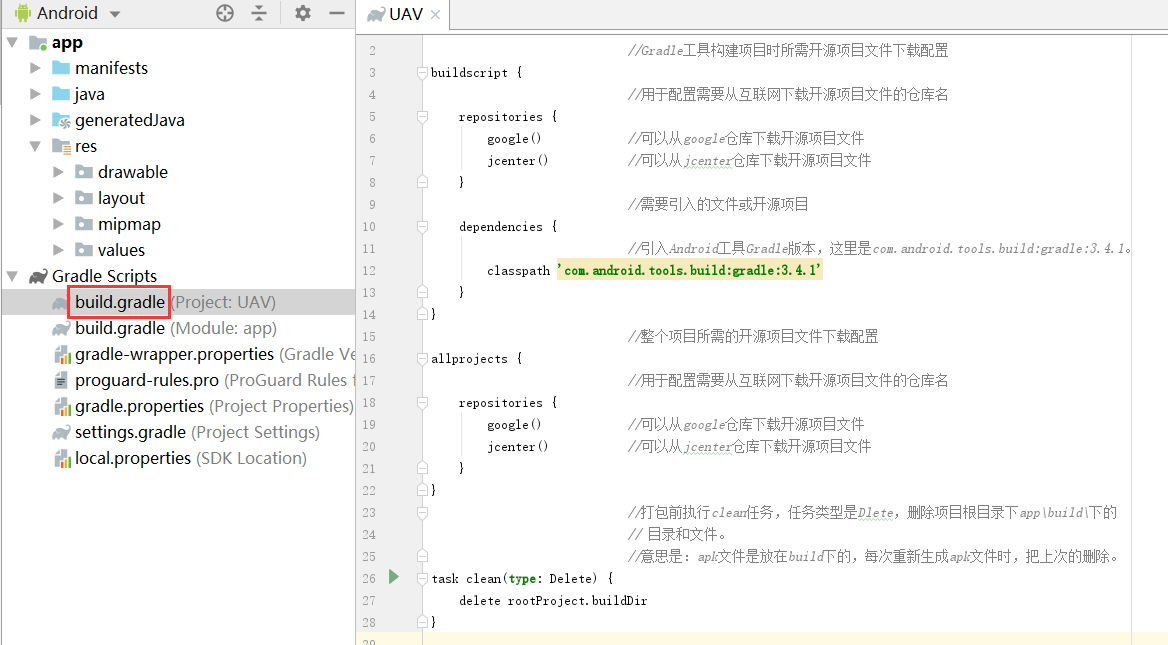
该文件是Android App的安装包，当我们生成后，就可以直接把它通过QQ、微信、邮件、数据线等等方式发给手机安装。如果我们不想重新生成该文件，使用已有的，可以到文件所在目录下去找，该文件对应的的路径为**app\build\outputs\apk\debug\**，如下图。



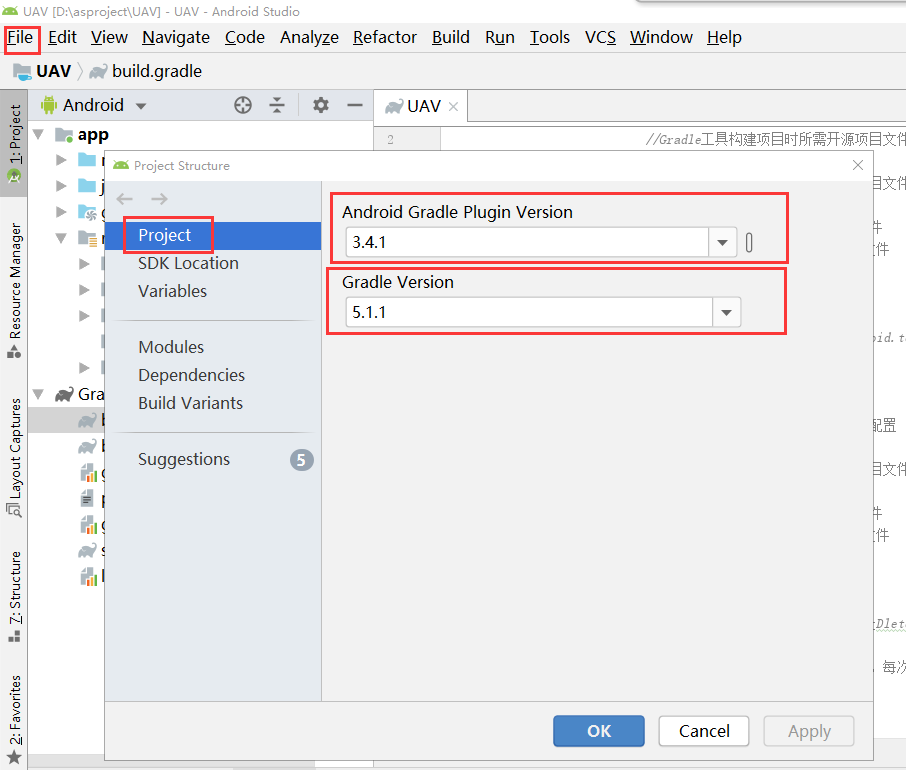
**build.gradle文件**

该文件是给Gradle工具创建编译Android Studio项目用的。Gradle是一种构建项目用的工具，它使用了基于Groovy语言来声明项目设置，抛弃了基于XML，如Ant和Mavend的各种配置。该文件在Android Studio工具里有2个，以Android方式打开项目时都是在Gradle Scripts目录下；以Project方式打开项目时分别是在app目录下和gradle目录下。这2个build.gradle文件，一个是Project形式，一个Module形式，下面分别详细介绍。

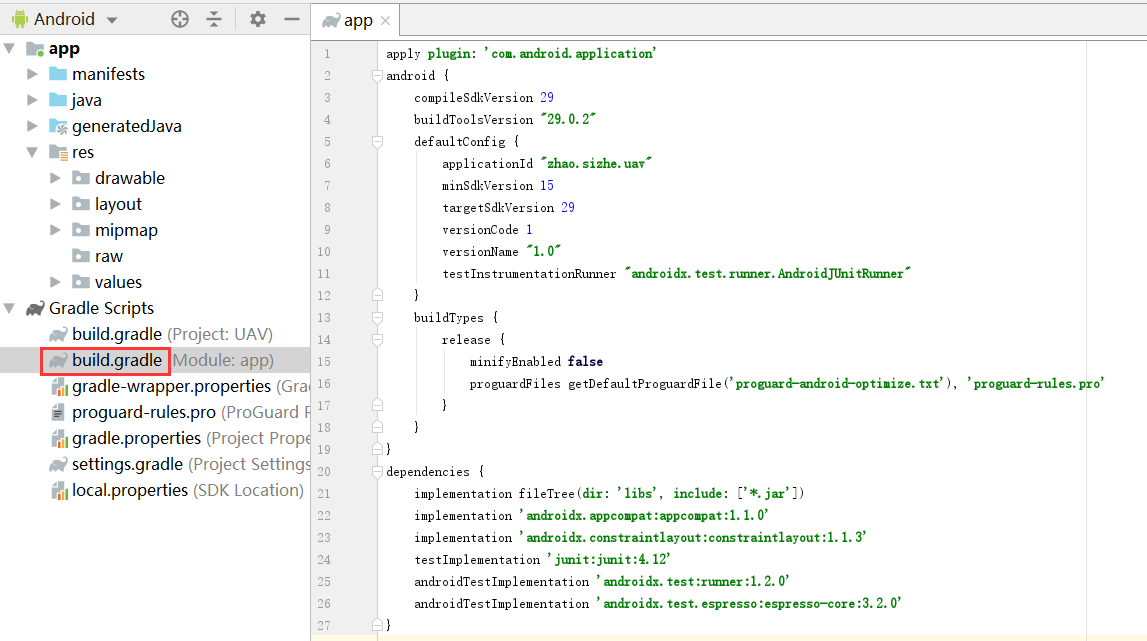
Project形式的build.gradle文件



这个文件除了有时可能需要修改Gradle工具的版本外，其他的都不需要动，因为不同版本的工具构建的安卓项目可能不兼容，其详细的注释说明见图片代码中的注释。这个工具其实有2个版本选择，一是原始Gradle工具的版本，一是Android Gradle工具插件的版本，可以通过Android Studio工具顶部菜单【File】>【Project Structure】>【Project】选项查看或设置，如下图：



Module形式的build.gradle文件



这个文件里分为3部分：apply plugin、android和dependencies，下面详细介绍。

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

*//如果当前安卓项目是作为开发的APP,值用com.android.application表示。  
 //如果当前安卓项目作为APP的一个被调用的库，值用com.android.library表示。  
 //它的值在创建项目或创建库项目时，会自动生成。*

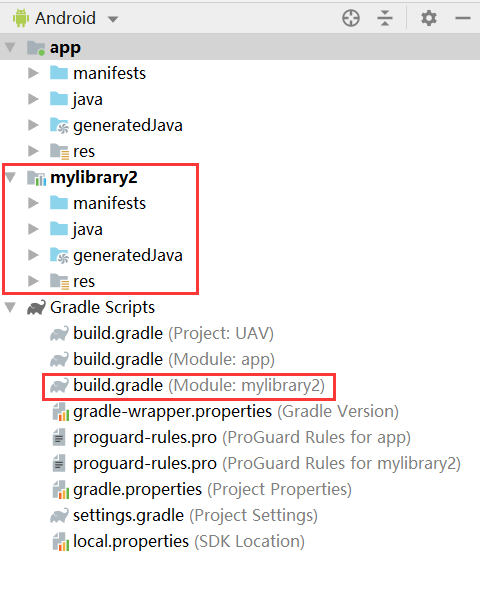
*//安卓项目构建时所需要的配置*android {  
 compileSdkVersion 29 *//编译阶段的SDK安卓版本* buildToolsVersion **"29.0.2"** *//构建SDK工具的版本，和编译时SDK的版本最好保持一致。  
 //默认配置* defaultConfig {

*//APP的标识，或者说APP的包名，这个要是唯一的。*  
 applicationId **"zhao.sizhe.uav"** minSdkVersion 15 *//APP适用的最小安卓SDK版本* targetSdkVersion 29 *//运行阶段的SDK版本，不能小于最小SDK的版本，不能大于  
 //编译阶段的SDK版本。若设备版本大于运行版本，则按运行  
 //版本的SDK执行。* versionCode 1 *//APP版本编号* versionName **"1.0"** *//APP版本名称  
 //单元测试时需要引入的文件* testInstrumentationRunner **"androidx.test.runner.AndroidJUnitRunner"** }  
 *//构建类型，用于配置debug版本APP或release版本APP的参数。  
 //debug是测试版本，release是正式发布版本，前者一般不配置。* buildTypes {  
 *//发布版本配置* release {  
 minifyEnabled **false** *//是否开启代码混淆，false表示否，true表示是。  
 //指定代码混淆规则配置文件，前者是通用规则，后者是自定义规则。* proguardFiles getDefaultProguardFile(**'proguard-android-optimize.txt'**), **'proguard-rules.pro'** }  
 }  
}

*//引入所需文件或开源项目，根据开源项目的路径可分为3种：  
 //本地项目（或本地依赖）、库项目（或库依赖）和远程项目（远程依赖），如下。  
 //fileTree(dir: 'libs', include: ['\*.jar'])表示可调用libs目录下的所有.jar  
 //implementation '...'表示可调用远程项目  
 //implementation project(': 库名')表示可调用本地的库项目或库模块*dependencies {  
 implementation fileTree(**dir**: **'libs'**, **include**: [**'\*.jar'**])  
 implementation **'androidx.appcompat:appcompat:1.1.0'** implementation **'androidx.constraintlayout:constraintlayout:1.1.3'** testImplementation **'junit:junit:4.12'** androidTestImplementation **'androidx.test:runner:1.2.0'** androidTestImplementation **'androidx.test.espresso:espresso-core:3.2.0'**}

#### **3.2 关于库（**Module**）**

Android Studio 工具除了可以新建平常的项目外，还可以新建库或库项目或库模块，以供本项目调用。库项目是一个安卓项目，作为所开发的安卓项目的功能补充，可以创建或调用多个库项目，就像jar包一样。创建库项目步骤：顶部菜单【File】>【New】>【New Moudl】>【Android Library】>【Next】>......一直点击【Next】，创建好后，如下图。

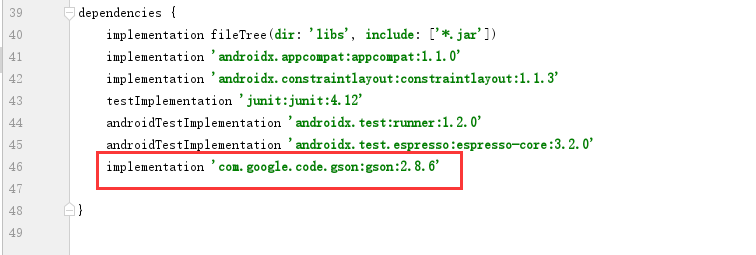


#### 3.3 引入网上的库或开源项目

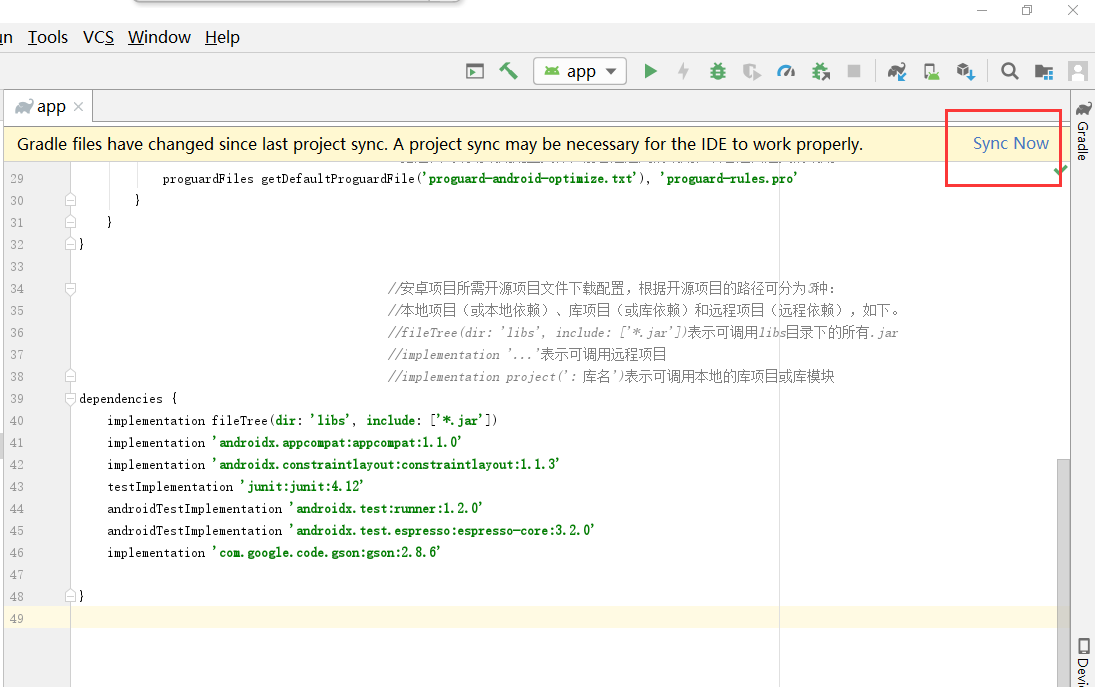
可以通过手动的dependencies方式和图形界面的dependencies方式调用下载，这里以调用网上的gson库为例分别介绍。

手动的dependencies方式分2步：

第一步：在build.gradle文件的dependencies里添加要调用文件库地址。

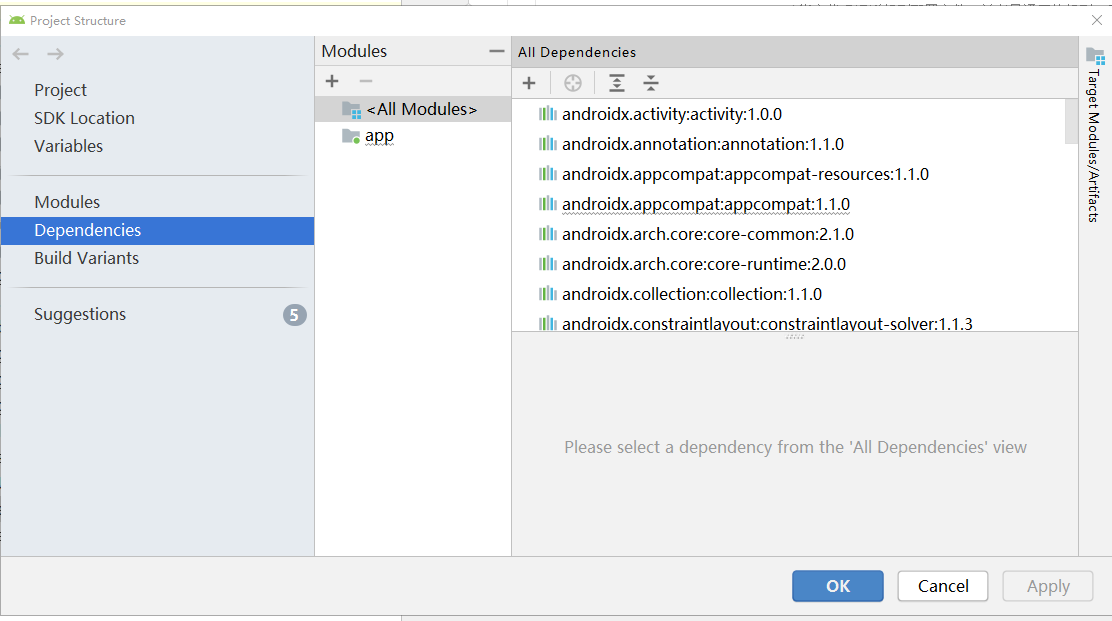


第二步：点击Android Studio工具右上角的Sync Now。点击后将会从网上下载相应的文件到External Libraries目录下。

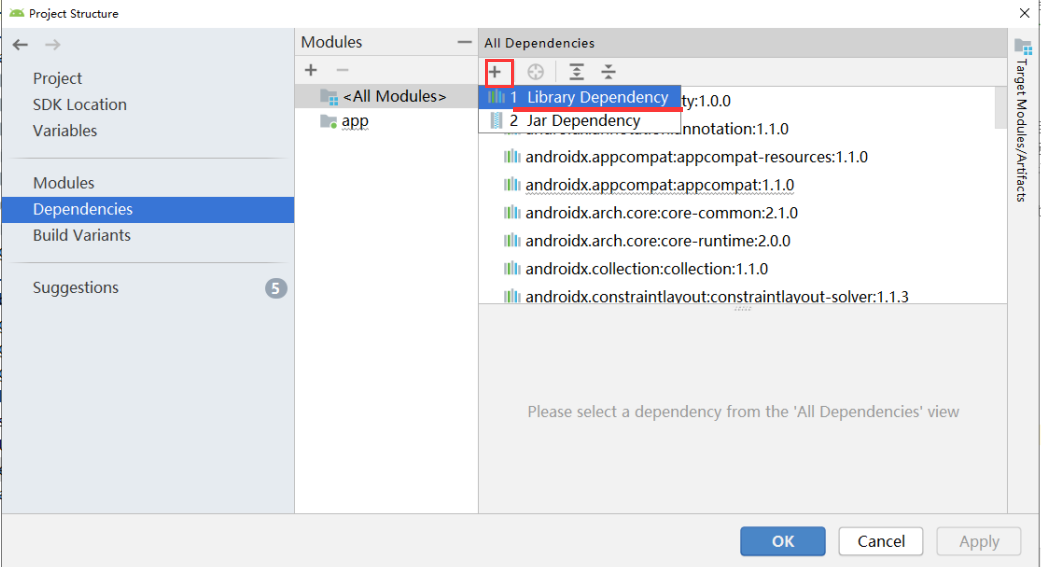


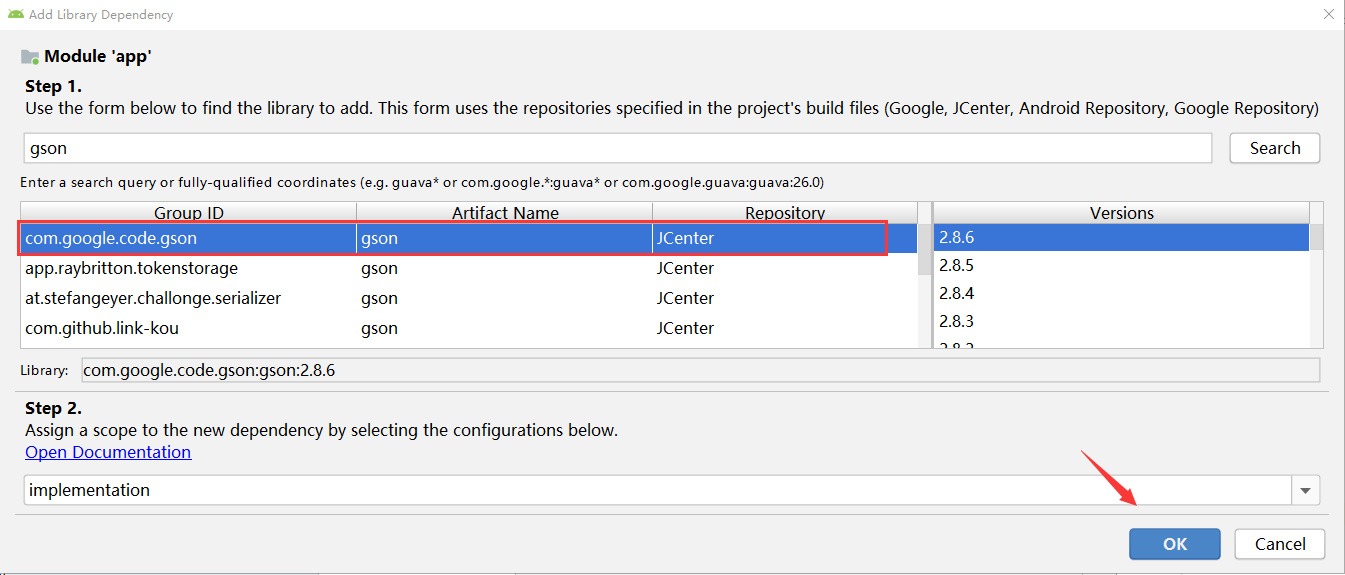
图形界面的dependencies方式分2步：

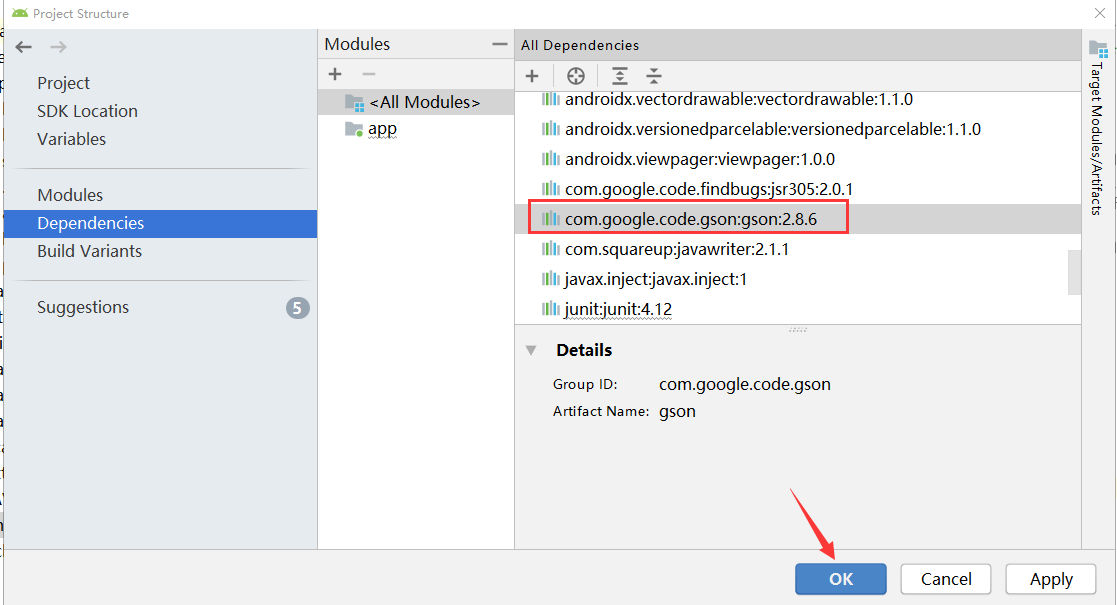
第一步：进入到Android Studio工具的顶部菜单【File】>【Project Stuctrue】，会弹出界面，选择【Dependencies】。



第二步：点击【All Dependency】里的【+】,选择【Library Dependency】，在输入框输入要添加的库或项目的关键字搜索，搜索到后选择需要的，点击【OK】，将会从网上下载相应的文件到External Libraries目录下。







#### 2.4 检验标准

知道怎么把apk文件安装到安卓设备里，比如手机或模拟器里；能熟练的引入网上的库或开源项目。

### 其他目录或文件介绍

#### 4.1 介绍

##### .gradle

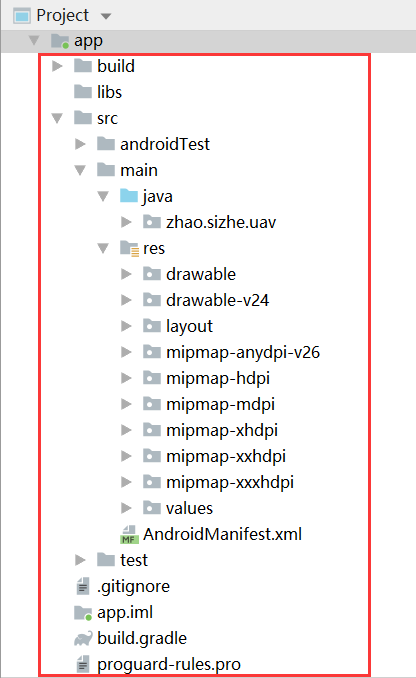
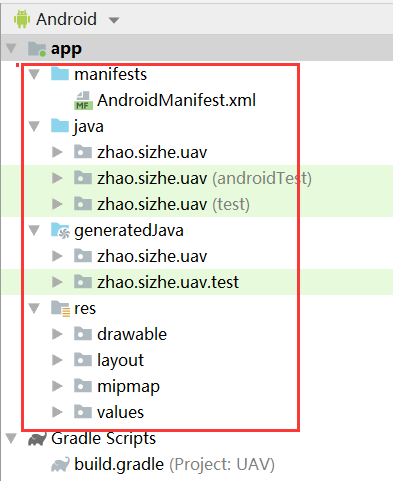
这个目录下放的是自动生成的Gradle工具相关的文件,比如Gradle版本。

##### .idea

这个目录下放的是自动生动的Android Studio工具相关的文件.

##### app

这个目录下放的是主安卓项目的所有文件。Android Studio工具的里的安卓项目分为2种：一种是我们要开发的APP或者说主安卓项目；一种库项目或库模块，这种项目只是作为功能被主项目调用，是主安卓项目的附属。这个目录下的子目录结构以Project方式呈现和以Android方式呈现稍微有点区别（如下图），这里我们以前者介绍它下面的目录。

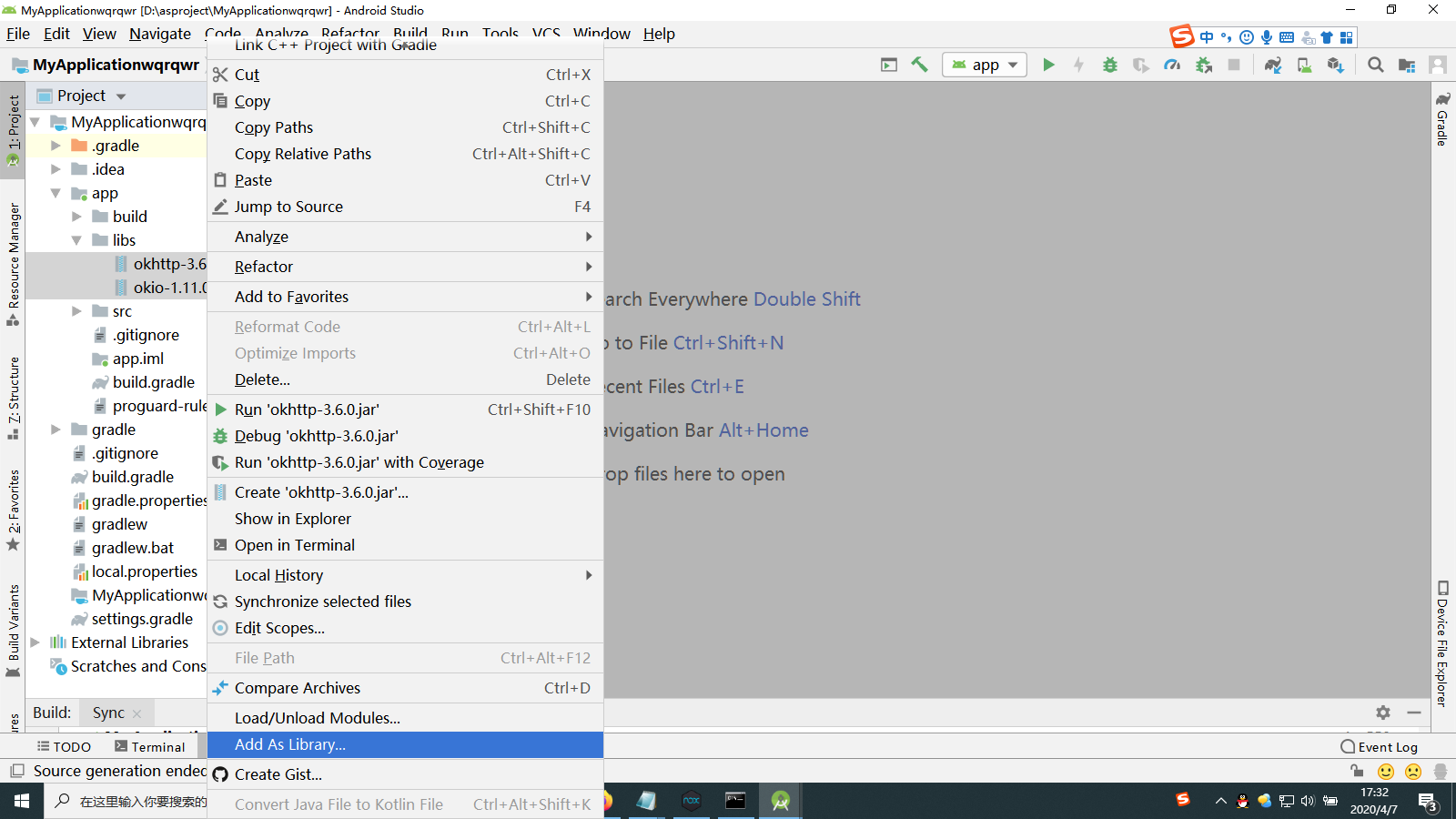
以Project方式打开 以Android方式打开

##### app\build

这个目录下放的是编译运行后自动生成的一些文件，其中就包含软件的安装文件apk，在build\outputs\apk\下。

##### app\libs

这个目录下放的是可以供调用的外部压缩包或jar库，如jar文件，把文件复制进来后，选中，点击右键，选择【Add As Library】即可。



##### app\src

这个目录下放的是java文件。

##### app\src\androidTest

这个目录下放的是主安卓项目做Android测试的一些文件。

##### app\src\main

这个目录下放的是主安卓项目的核心文件：java文件、资源文件（res）、配置文件(AndroidManifest.xml)。

##### app\src\test

这个目录下放的是主安卓项目做java测试的一些文件。

##### app\src\.gitgnore

这个文件是版本控制系统软件git使用的ignore文件，可指定app目录下的哪些目录或文件排除在版本控制之外。

##### app\app.iml

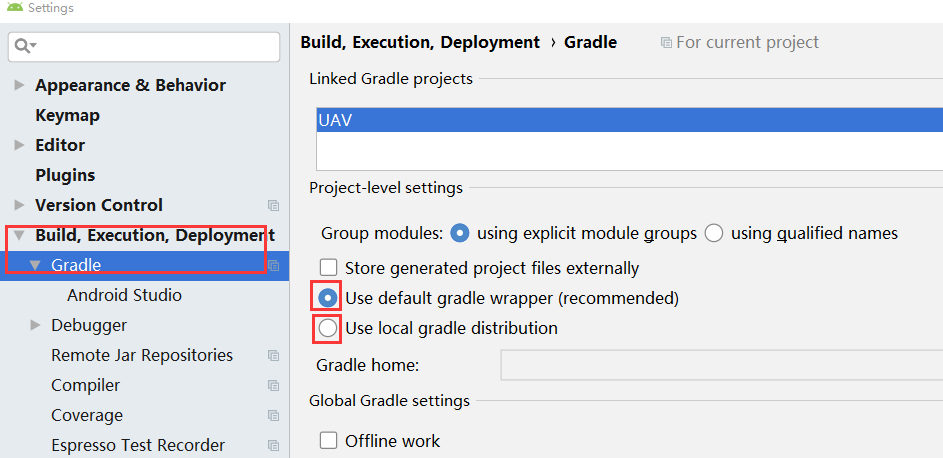
这个文件是保存Android Studio工具信息的文件，该文件自动生成。

##### app\proguard-rules.pro

这个文件是指定主安卓项目代码混淆规则的文件。

##### gradle

这个目录下放的是Gradle Wrapper的配置文件。Gradle Wrapper封装了Gradle，可以用它在没有安装Gradle工具的系统上，来构建项目。 gradle-wrapper.properties文件里配置了要下载的Gradle版本，这个是自动随Android Studio工具设置版本，可以在顶部菜单【File】>【Project Structure】>【Project】界面中设置可用的Gradle版本。如果当前没有Gradle，就从网上下载。不过有些版本的Android Studio工具默认没有启动，需要到顶部菜单【File】>【Settings】>【Build】>【Gradle】界面设置，如下图。



##### .gitgnore

这个文件是版本控制系统软件git使用的ignore文件，可指定哪些目录或文件排除在版本控制之外。

##### build.gradle

这个文件是Gradle工具的配置文件，一般保持默认的配置，该文件里的具体说明如下：

*//Gradle工具构建项目时所需开源项目文件下载配置*buildscript {  
 *//用于配置需要从互联网下载开源项目文件的仓库名* repositories {  
 google() *//可以从google仓库下载开源项目文件* jcenter() *//可以从jcenter仓库下载开源项目文件* }  
 *//需要引入的文件或开源项目* dependencies {  
 *//引入Android工具Gradle版本，这里是com.android.tools.build:gradle:3.4.1。* classpath **'com.android.tools.build:gradle:3.4.1'** }  
}  
 *//整个项目所需的开源项目文件下载配置* allprojects {  
 *//用于配置需要从互联网下载开源项目文件的仓库名* repositories {  
 google() *//可以从google仓库下载开源项目文件* jcenter() *//可以从jcenter仓库下载开源项目文件* }

}  
 *//打包前执行clean任务，任务类型是Dlete，删除项目根目录下app\build\下的  
 //目录和文件。  
 //意思是：apk文件是放在build下的，每次重新生成apk文件时，把上次的删除。*task clean(**type**: Delete) {  
 delete rootProject.buildDir  
}

##### gradlew

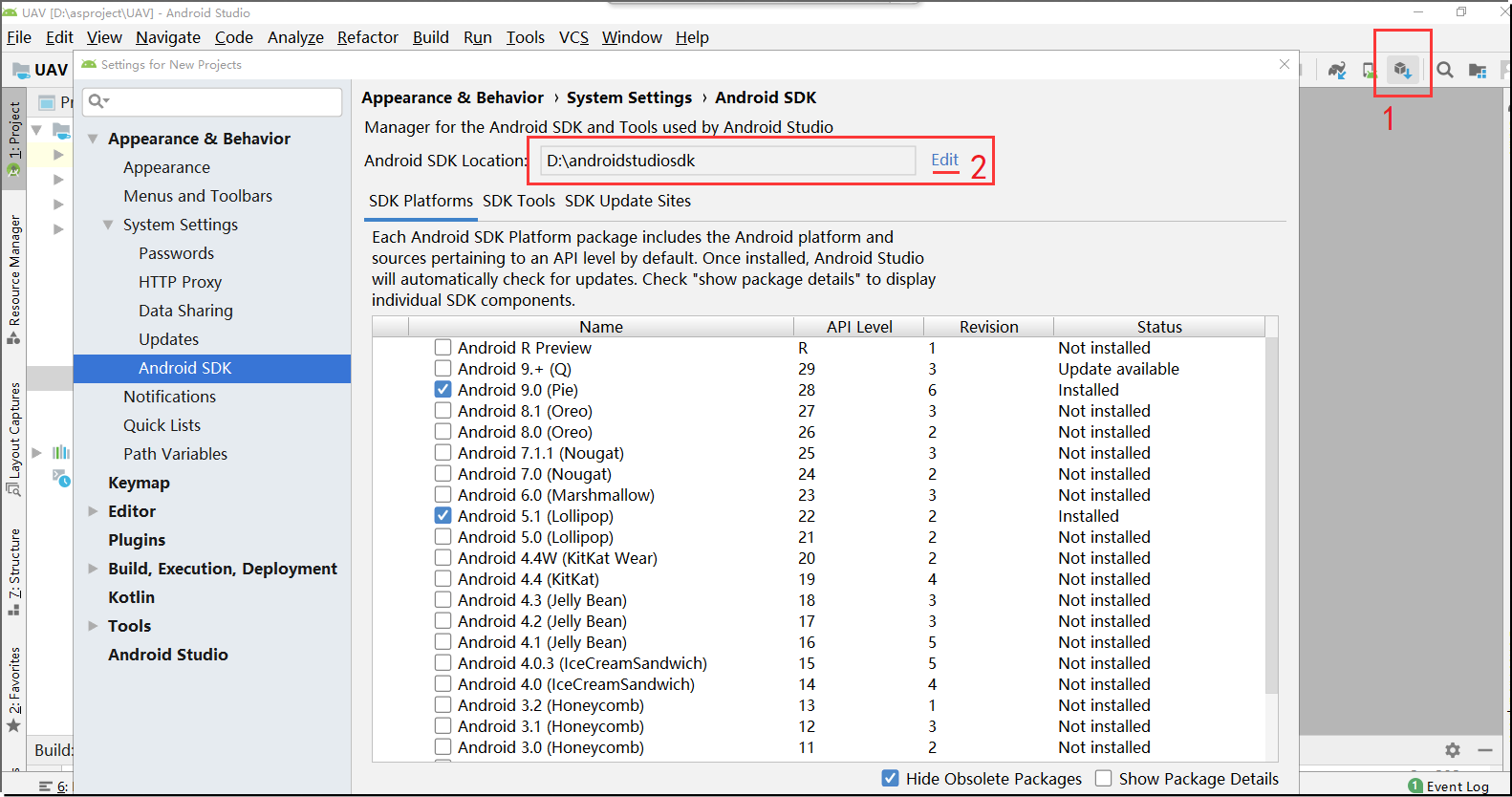
这个文件是\*nix下的gradle wrapper可执行文件。

##### gradlew.bat

这个文件是windows下的gradle wrapper可执行文件。

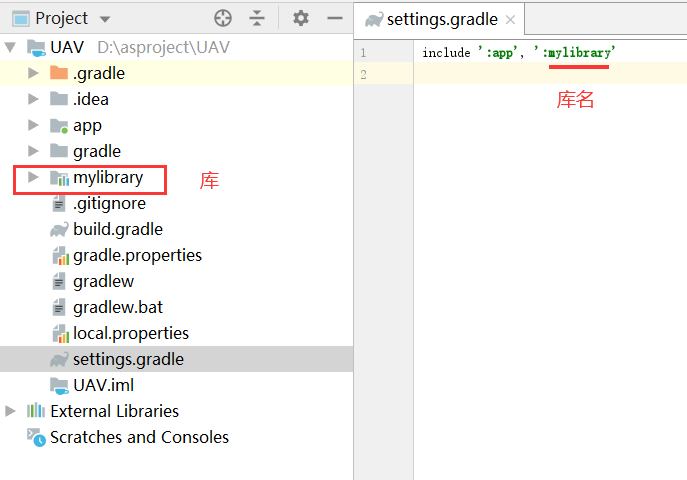
##### local.properties

这个文件是本地SDK路径的配置文件，如果本地计算机SDK的路径发生了变化可以修改，一般不在这里修改路径，在图形界面里修改，如下图：



##### settings.gradle

这个文件是用来配置主安卓项目的库项目或库模块，当创建库项目的时候会自动生成。比如下面创建了一个名字叫mylibrary的库模块，该文件里会自动配置。

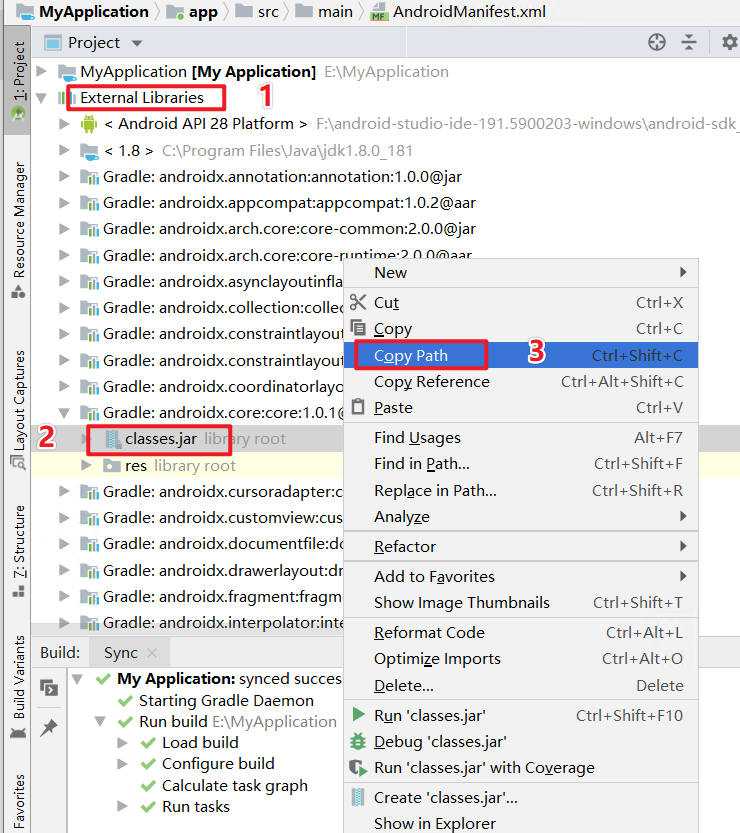


##### 项目名.iml

这个文件是Android Studio工具自动生成的保存项目信息的文件。

##### External Libraries

这个目录下放的是项目所调用的外部文件或库，比如jdk等。build.gradle文件dependencies里引入的文件或开源项目也会下载到这里，默认存放路径为/home/用户名/.gradle/caches/下。如果要查看详细的路径或复制文件，可以按下面步骤操作。



#### 4.2 检验标准

知道怎么查找任意目录介绍，并能理解其意思，能按照其介绍过程操作一遍。