

# 1 项目初始化

“Android 样本项目”的 GitHub 资源里面包含有项目的需求文档、原型切图和数据库 API 等。

这一章节的内容是介绍项目的初始化，主要是规范代码和图片资源的组织形式。

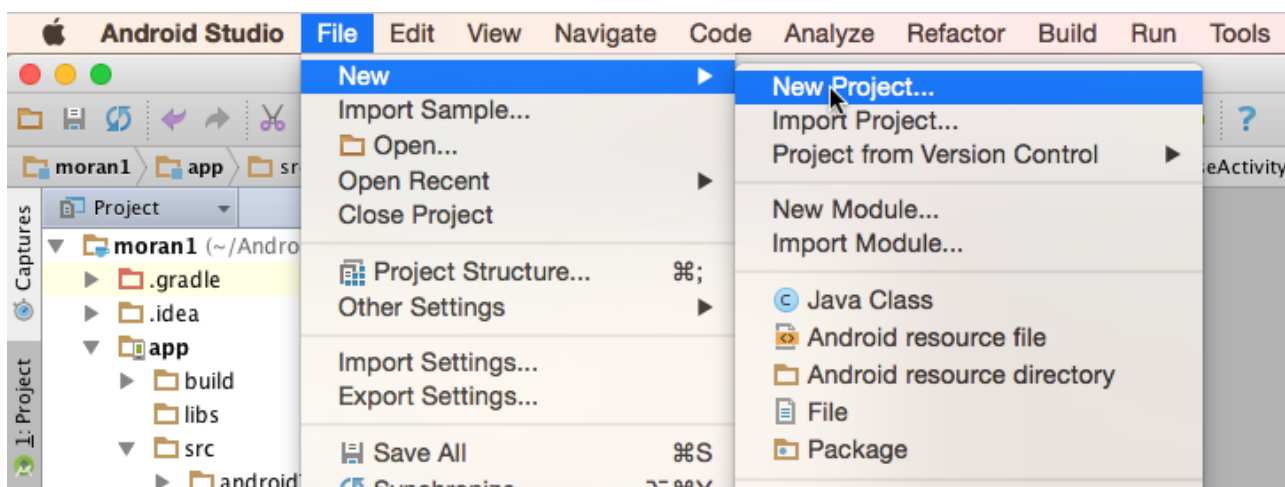
## 开发环境

Android Studio 1.3.1 / SDK Version 23

## 操作步骤

### 1.1 新建 App

(1) 运行 Android Studio，依次选择“File | New | New Project...”，打开新建项目对话框；



(2) 在对话框中将应用名称改为：Moran，并设置合适的组织域名、项目存放路径，点击 Next 按钮；

Create New Project

## New Project

Android Studio

### Configure your new project

Application name:

Company Domain:

Package name:  [Edit](#)

Project location:  [...](#)

( 3 ) 选择最低兼容的 Android 版本为 4.0 ( 覆盖了 94%的设备 , 版本越低 , 可用的功能越少 ) , 点击 Next 按钮 ;

Create New Project

## Target Android Devices

### Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK:  [Help me choose](#)

☐ Wear

Minimum SDK:

☐ TV

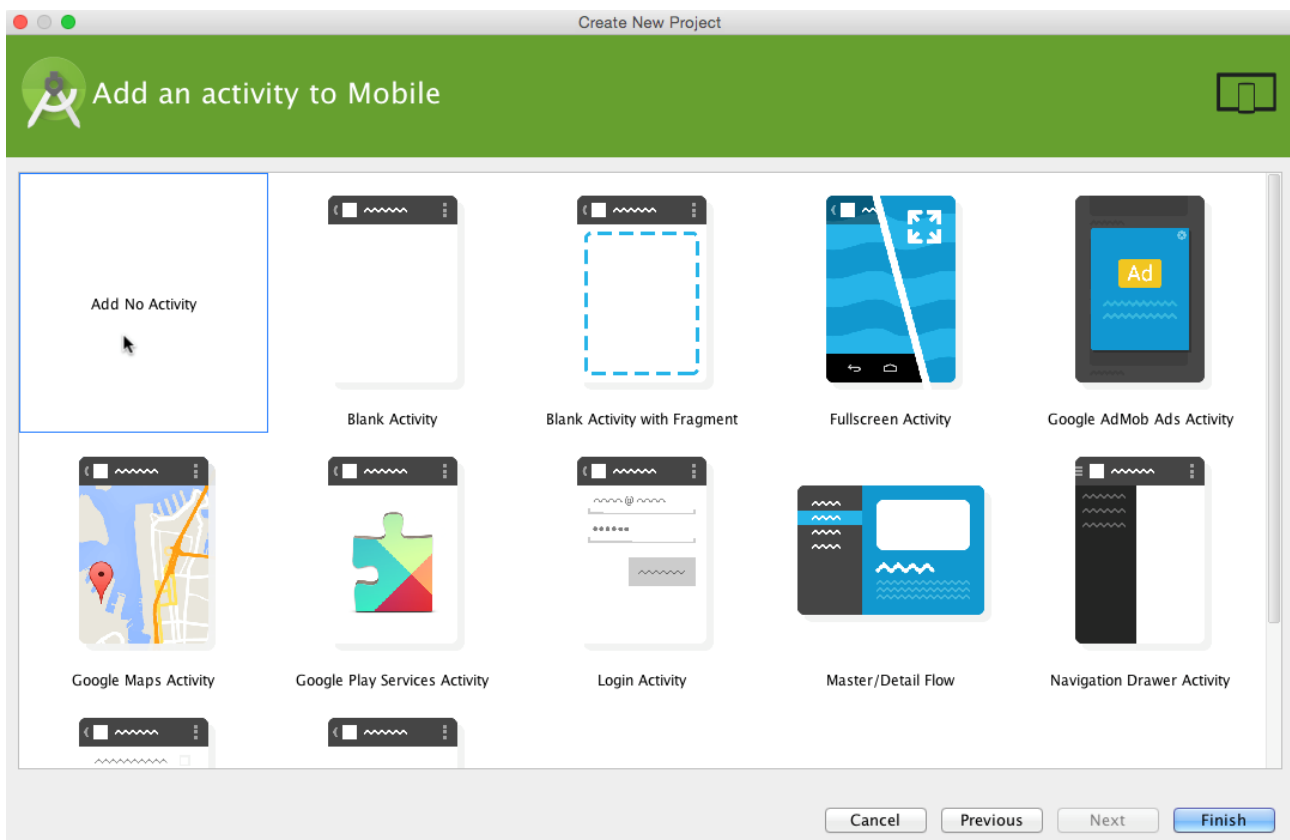
Minimum SDK:

☐ Android Auto

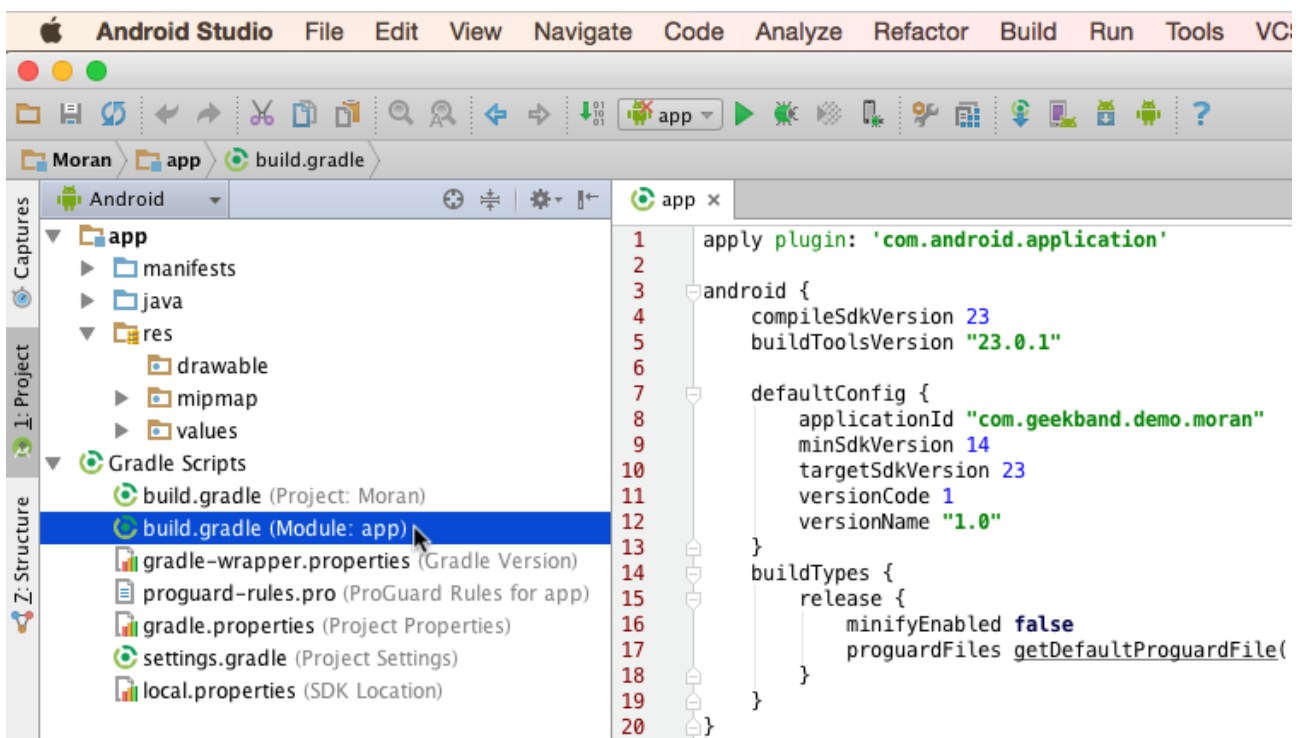
☐ Glass (Not Installed)

Minimum SDK:

(4) 向应用添加一个 Activity，可选择不同的模版套用，此处我们选择第一个，不添加（待项目创建完成再添加），点击完成。

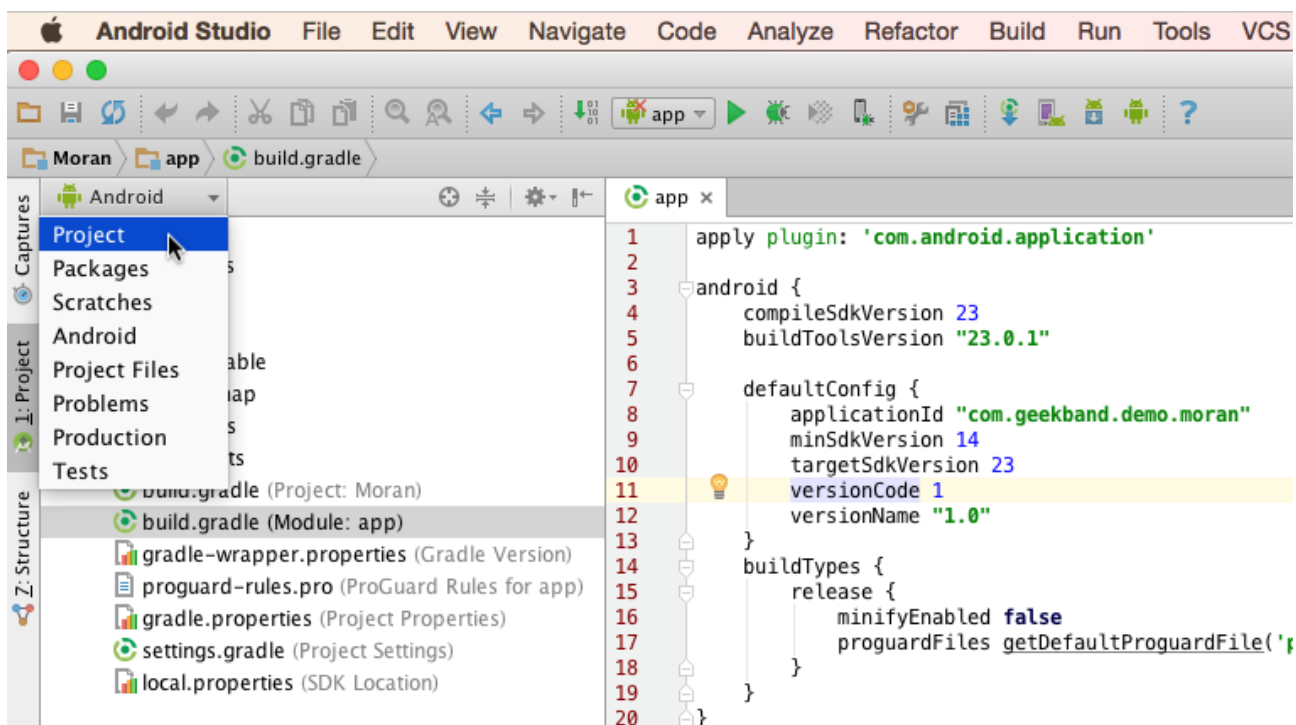


(5) 完成上述步骤后，新的应用项目就创建好了，等待开发环境准备完成，就可以着手开发应用了。如图点击左侧导航，可以在 Gradle 中查看到应用的配置信息。

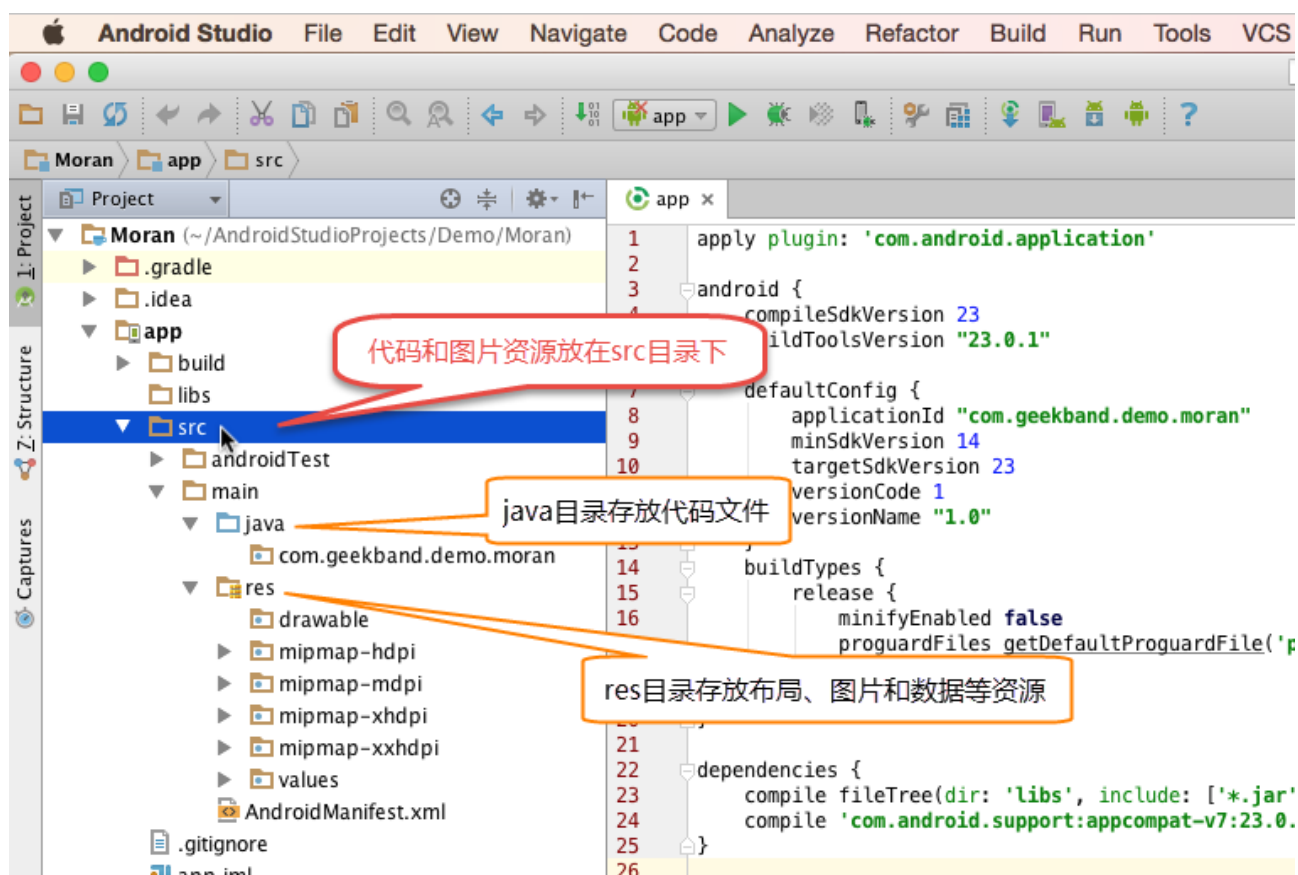


## 1.2 添加代码资源

(1) 如图点击左侧导航菜单，从左上角到视图切换列表中选择 Project 视图；

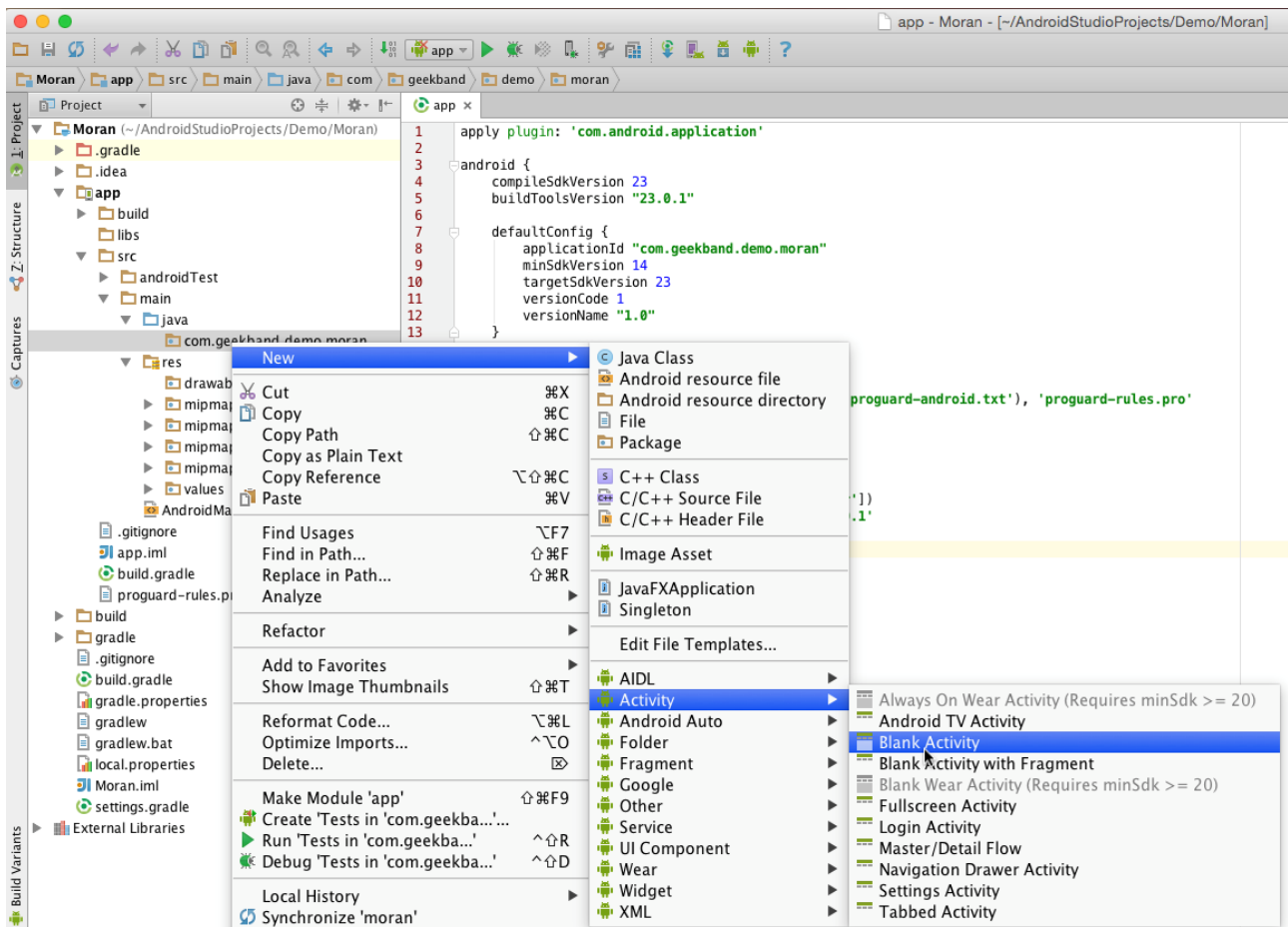


(2) 展开到 src 目录，该目录存放了 Android 应用的代码和资源文件，其中 java 目录存放代码，res 目录存放图片、动画等资源；

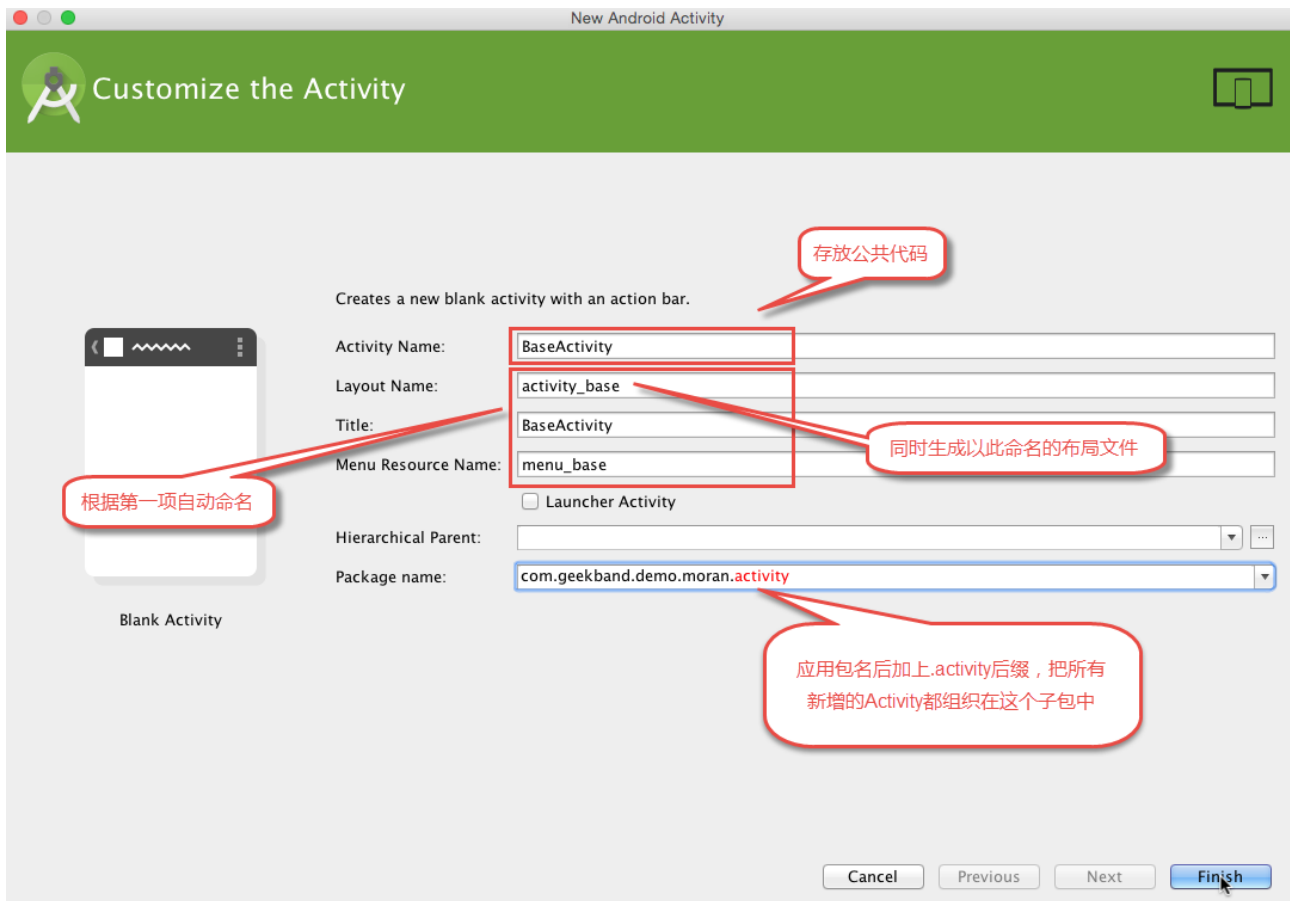


(3) 现在我们对代码资源进行组织。新建一个 Activity (Activity 的作用是为用户提供一个用于任务交互的画面)，如图定位到 java 目录下的包命名空间 (即本文 1.1 节第二步中的 Package name，默认会根据该步骤前两项自动命名，这里为

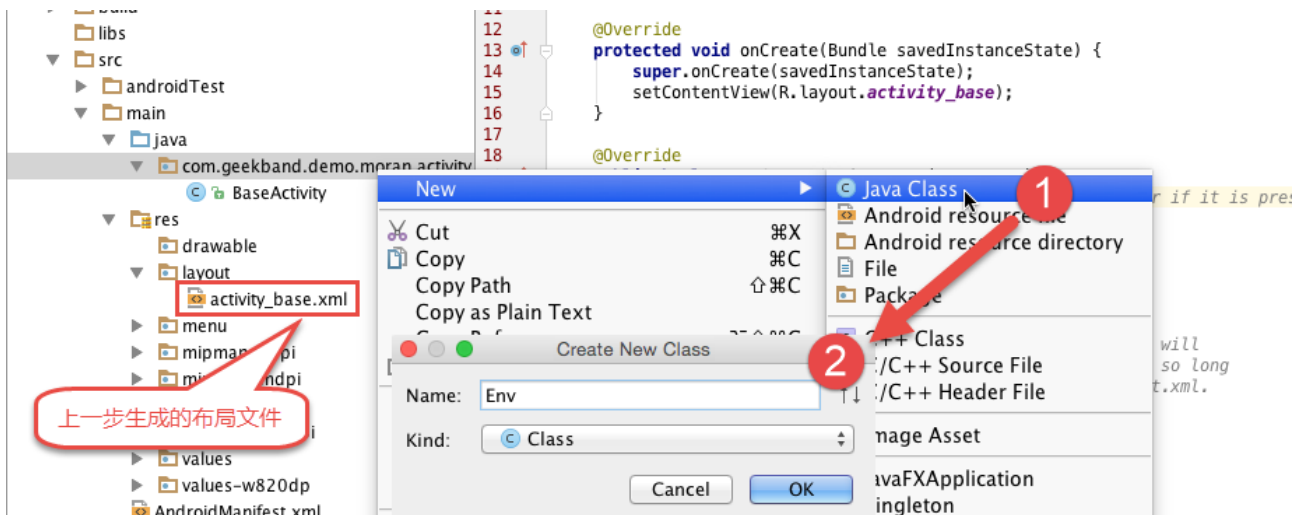
“com.geekband.demo.moran”，以下简称包名，它唯一标识了我们的应用)，右键点击 “New | Activity | Blank Activity”，打开新建 Activity 对话框；



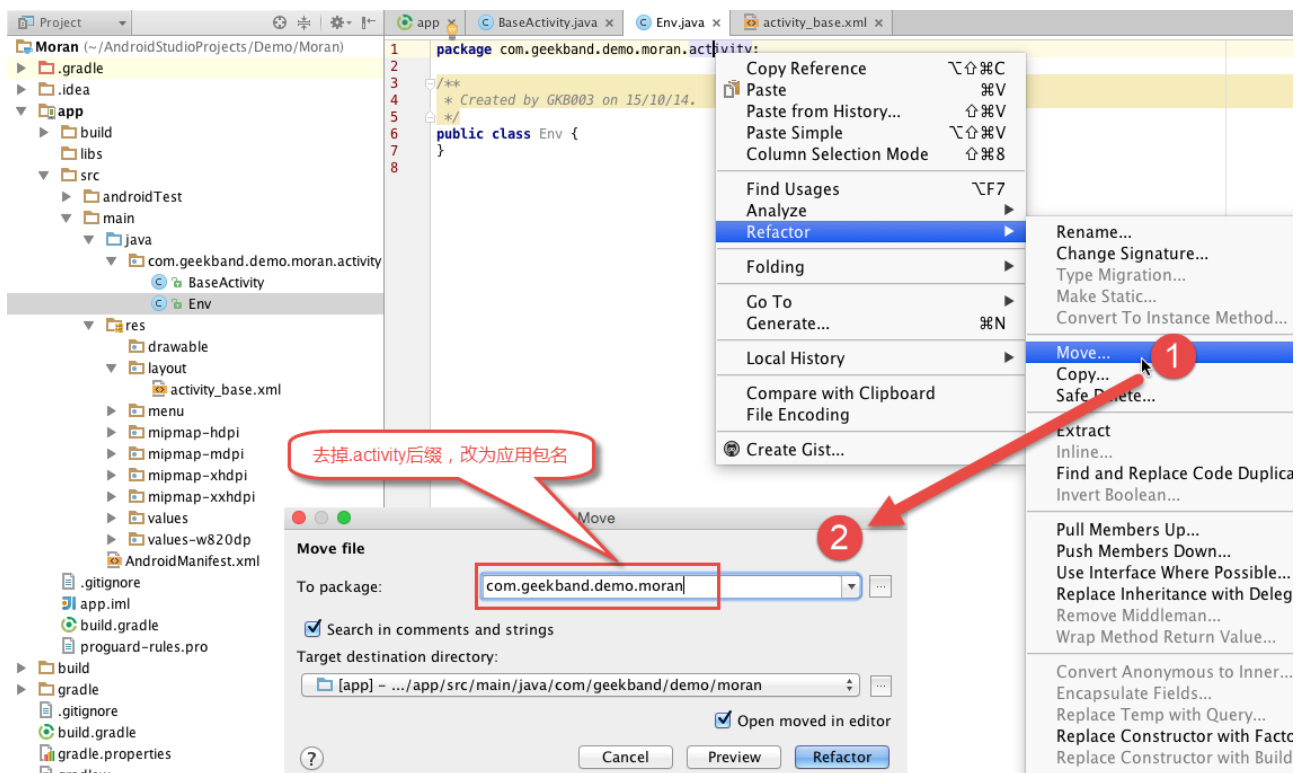
(4) 首先修改 Activity Name 为 BaseActivity (存放一些公共代码)，下面几项会自动重命名，然后需要修改最后一项 Package Name，给它加上 ".activity" 后缀 (作为应用包的子节点，以后所有新增的 Activity 都放在 activity 子包中)，点击完成；



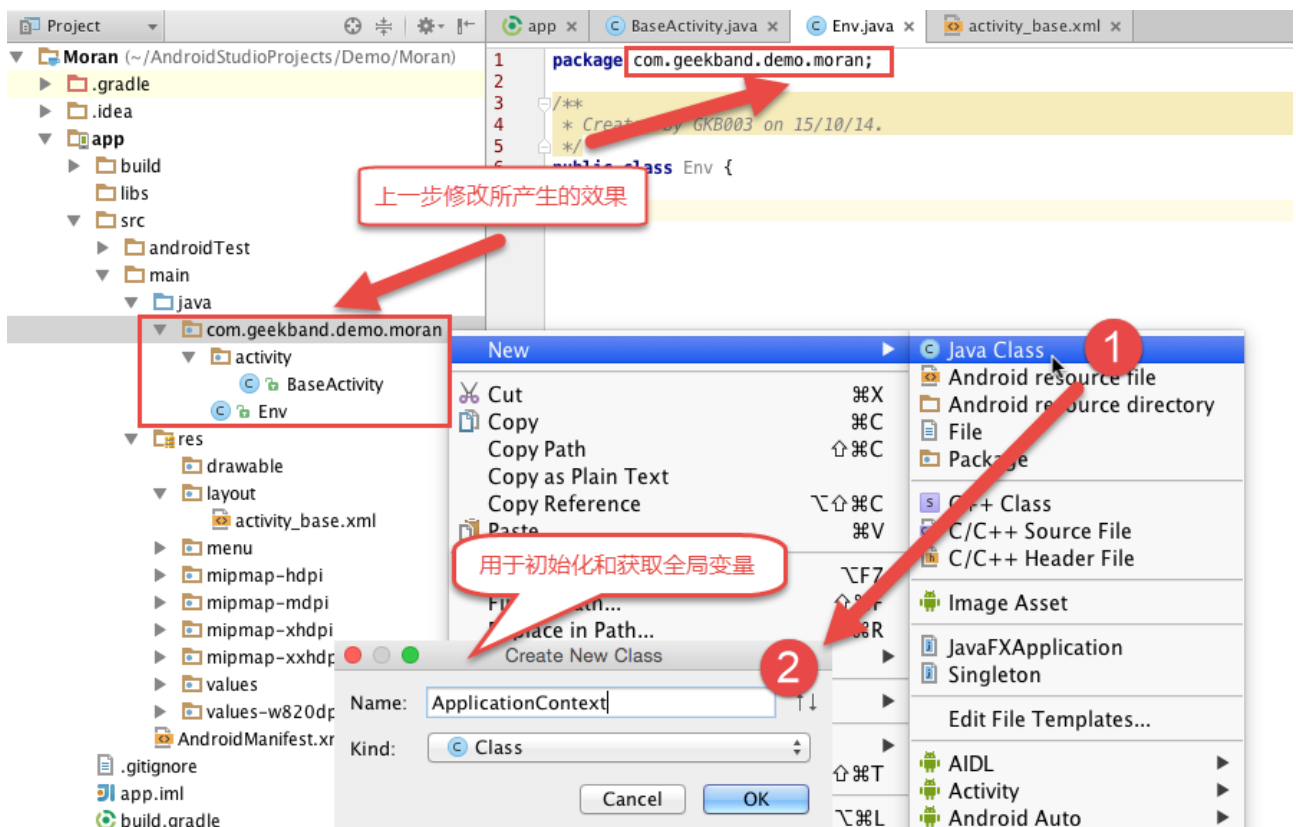
(5) 如图在包名上右键 “New | Java Class” , 在弹出的窗口中输入 Env , 新建一个名为 Env 的类存储静态变量 ;



(6) 打开新建的 Env 类 , 可以看到其所属包名 , 我们其实不想把它放在 activity 子包中 , 所以这里重新组织一下 , 如图在代码中的包名上右键 “Refactor | Move” , 在弹出对话框中去掉包名中的 “.activity” 后缀 , 注意观察更改效果 ;

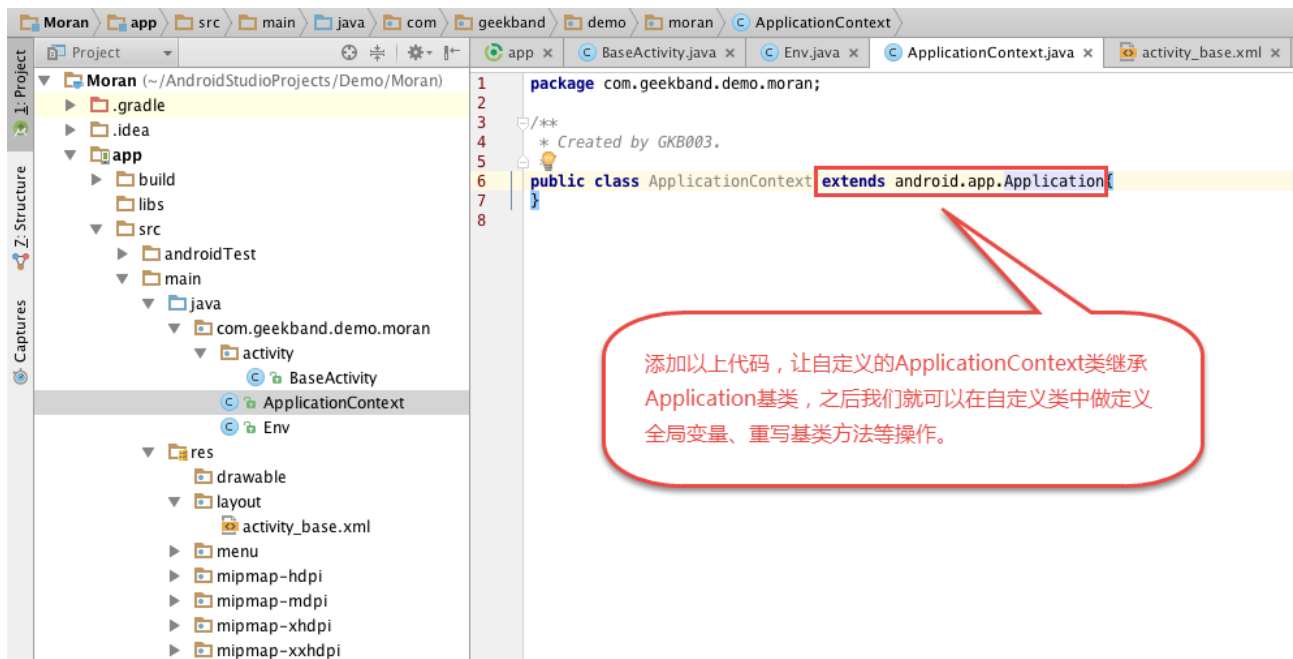


(7) 再新建一个类, 取名 ApplicationContext, 用来定义全局变量;

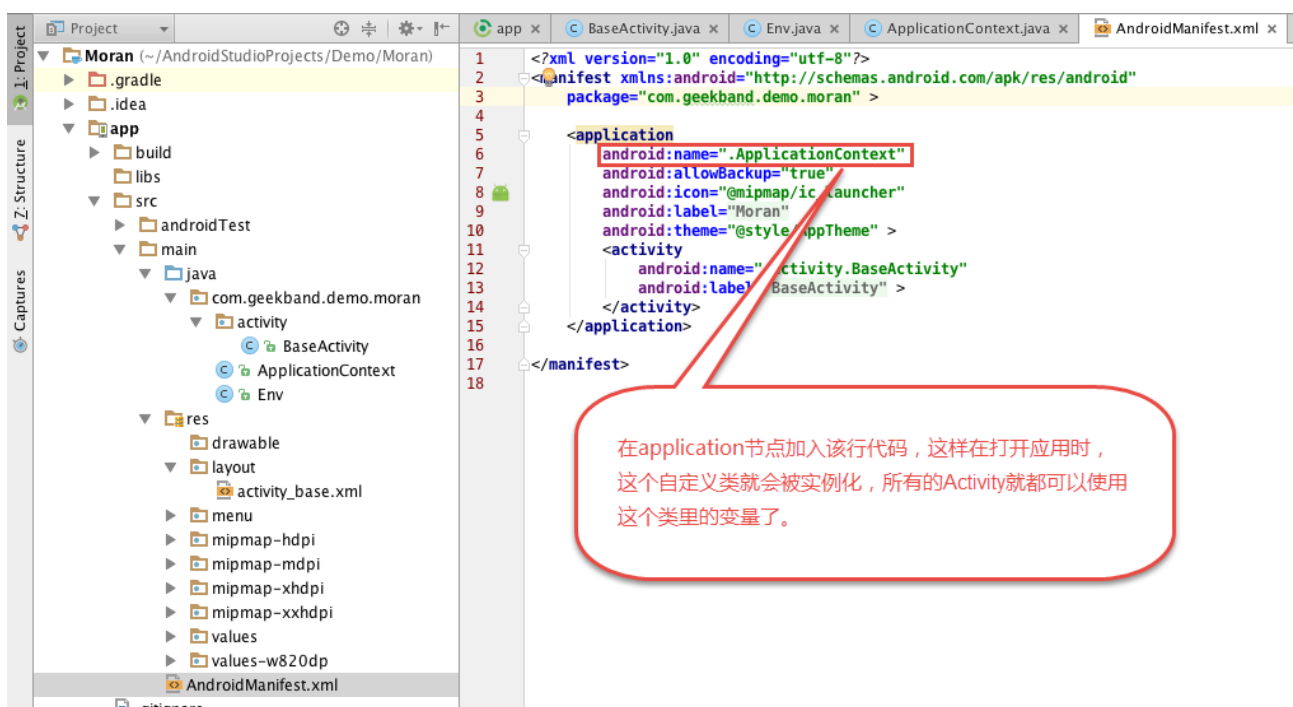


(8) 打开上一步创建的自定义类, 修改如下代码, 让其继承 Application 基类 (这个基类的作用是用来获取整个 App 的状态);





( 9 ) 打开 AndroidManifest.xml 文件，如图在 application 节点加入一行代码，这样 App 进程被创建时，这个类就会被实例化；

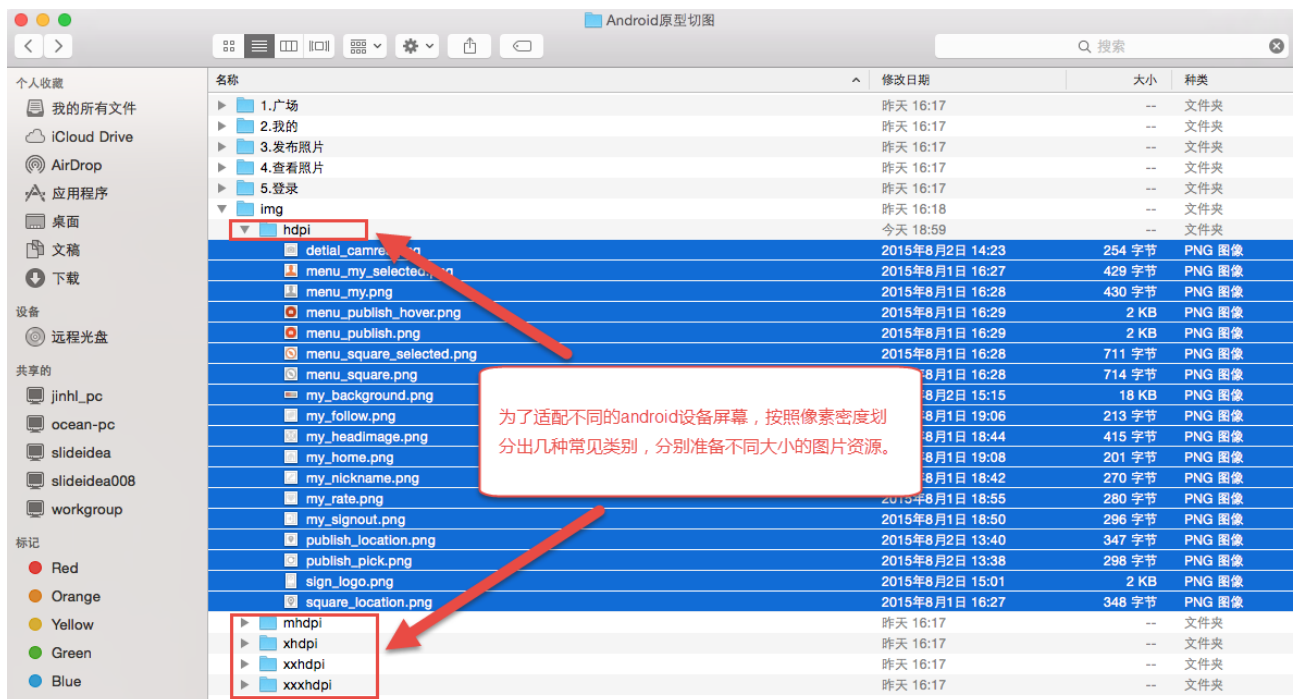


( 10 ) 在应用包名上右键选择 "New | Package"，输入包名 "model"，完成后在 model 节点上新建一个 modelplaceholder 类，如图所示，基本的代码结构就组织完成了。

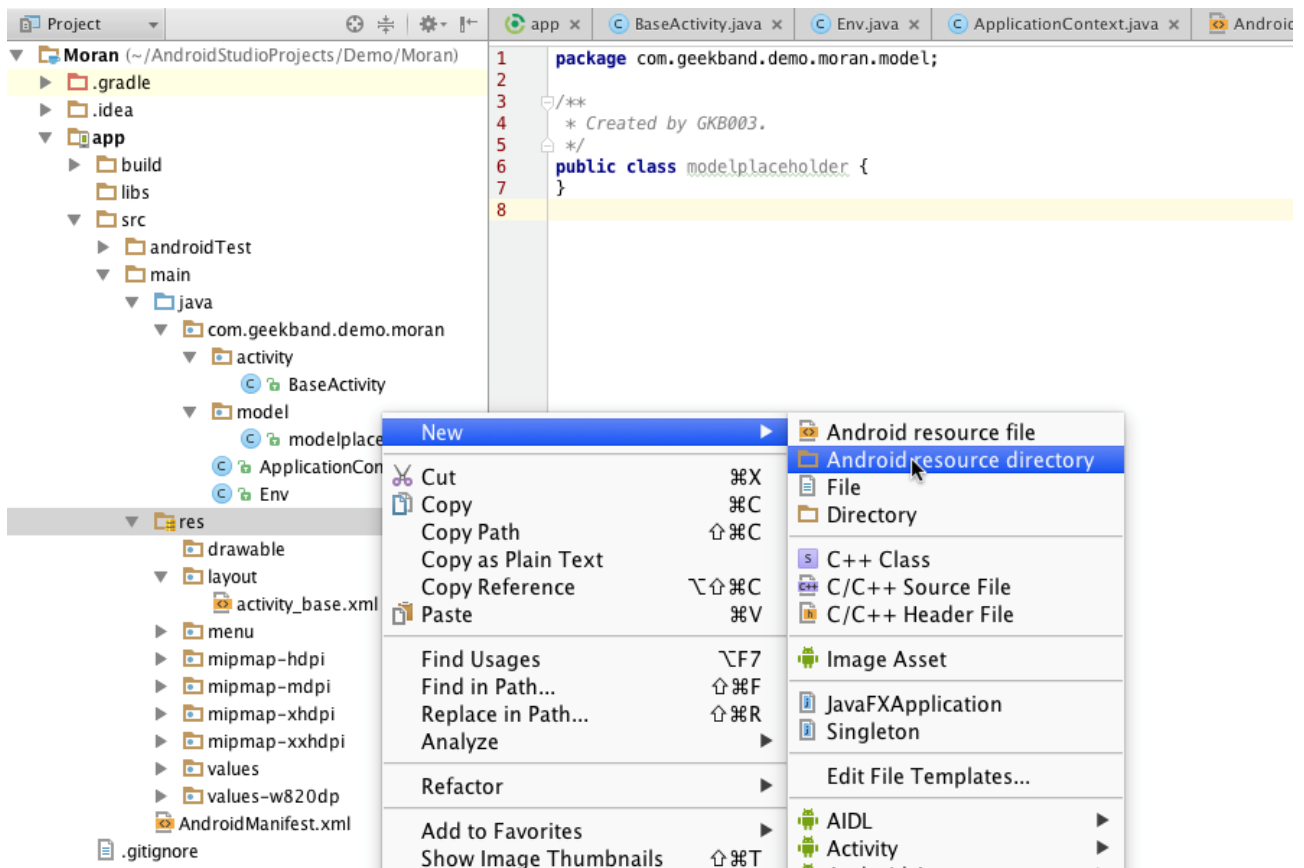


### 1.3 导入图片资源

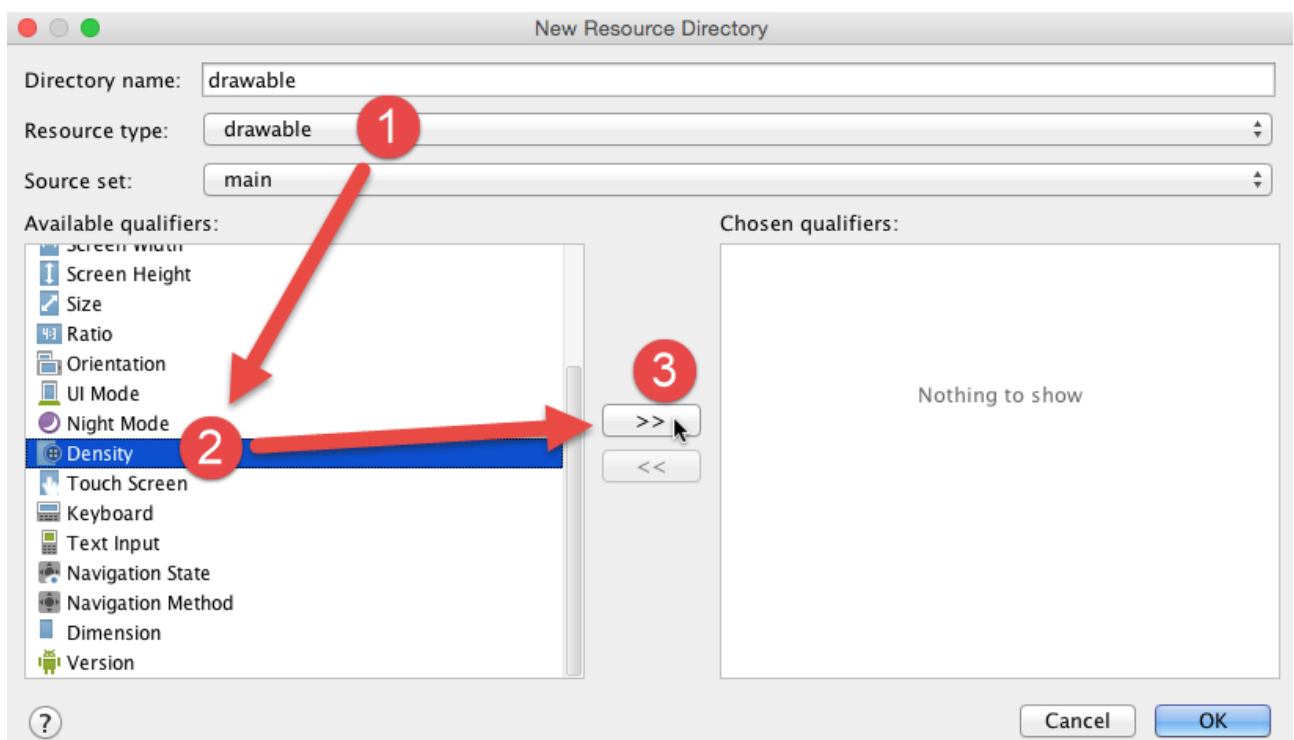
(1) 接下来进行图片资源的整理。将事先切好的原型准备好，如图拷贝原型切图里“img/hdpi”目录下的图片资源；



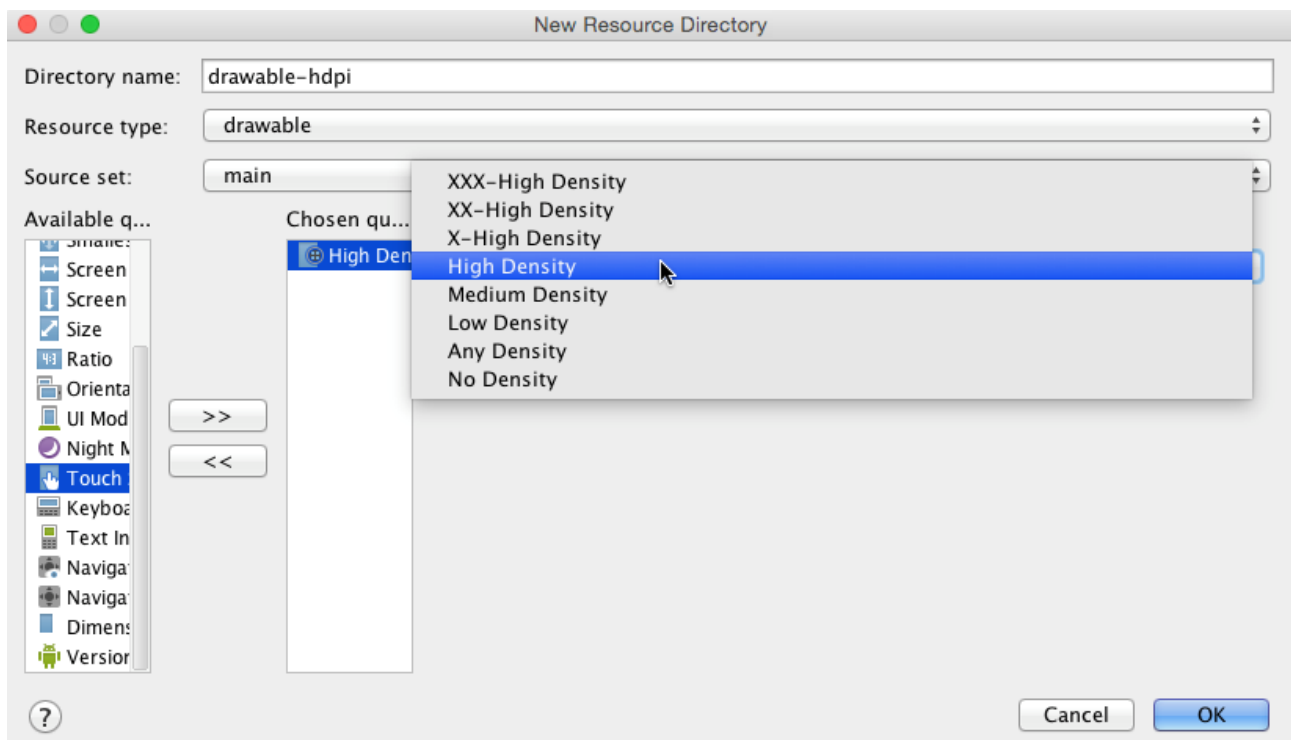
(2) 定位到 Android Studio 中 res 目录，右键选择 “New | Android resource directory” ；



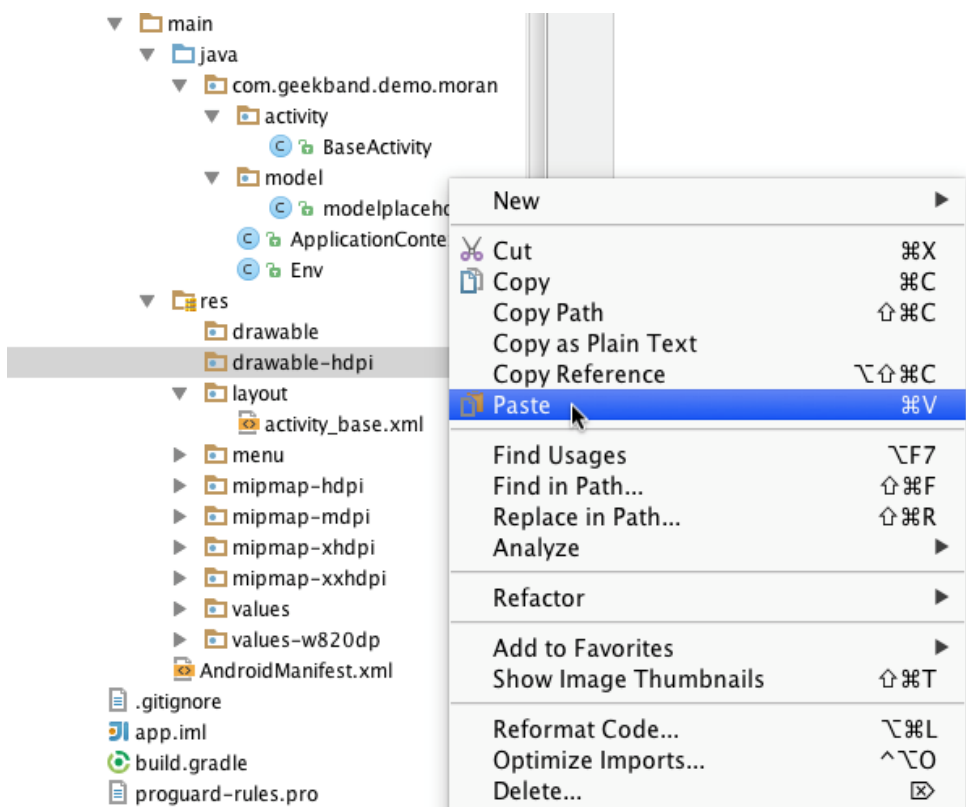
(3) 如图，在新增资源目录对话框中，Resource type 选择 drawable，Available qualifiers 选择 Density；



(4) 紧接着密度下拉列表框中选择 “High Density”，点击 OK 完成；



( 5 ) 如图，将上面拷贝的图片资源粘贴到新增的 drawable-hdpi 资源目录中；



( 6 ) 类似地，将适配其他设备屏幕的图片按照密度分类导入到对应资源目录，至此实验手册第一部分“项目初始化”结束。

