Android提供了丰富的应用程序框架，它允许你建立创新的应用和游戏在Java语言环境中的移动设备。

Android应用提供多个入口点。App由多个可以单独调用的组件组成。比如：一个单独的Activity提供一个UserInterface，一个service在后台运行。在一个组件中通过intent启动另一个组件（可以是不同应用程序的组件）。这种模式对于单一的应用程序提供多个入口点，并默认允许其他app调用。

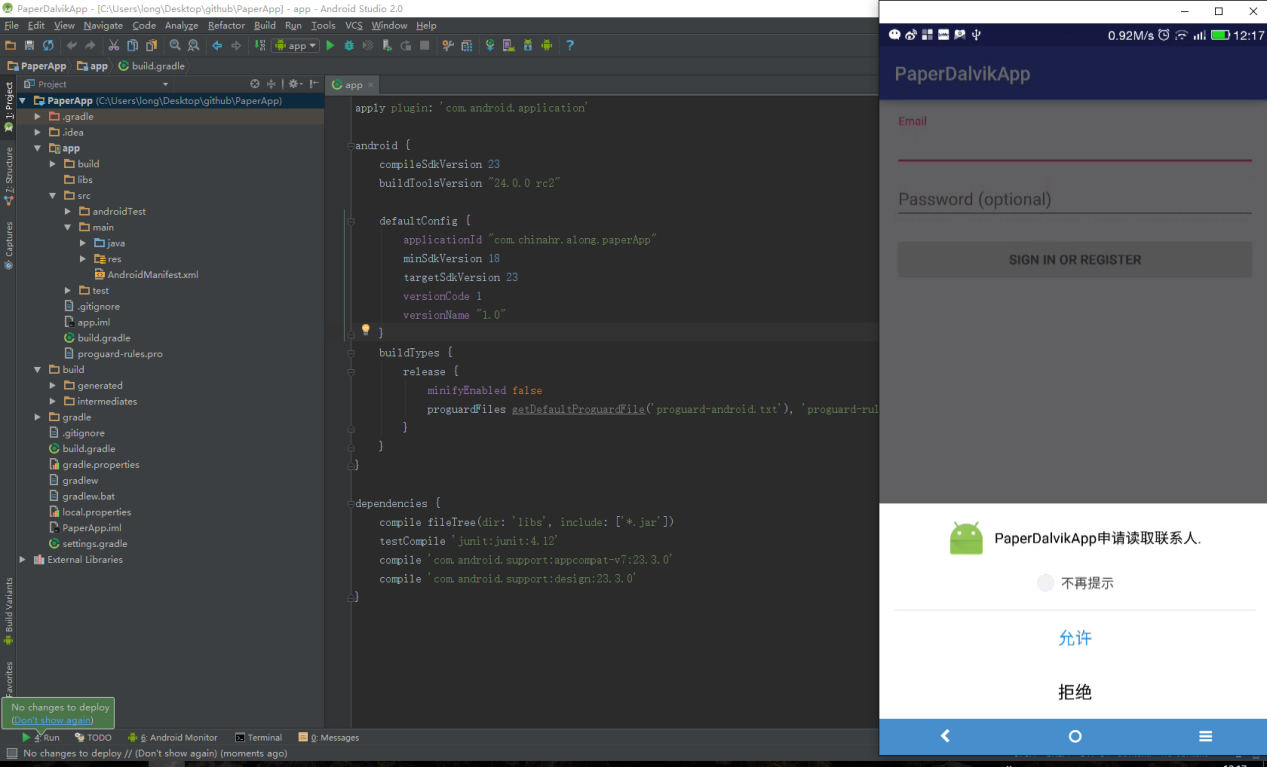
Apps可以适配不同的设备。Android提供了一个自适应的应用程​​序框架，它允许您为不同的设备配置提供独特的资源。例如，您可以创建不同的屏幕大小不同的XML布局文件和系统确定应用基于当前设备的屏幕尺寸，其布局。您可以在运行时查询设备功能的可用性，如果任何应用功能需要特定的硬件，如相机。如果需要，您还可以声明功能，您的应用程序需要这样的应用市场，如谷歌Play商店不允许在不支持该功能的设备的安装。

正常的工作流是这样的：创建，编译，测试，打包。

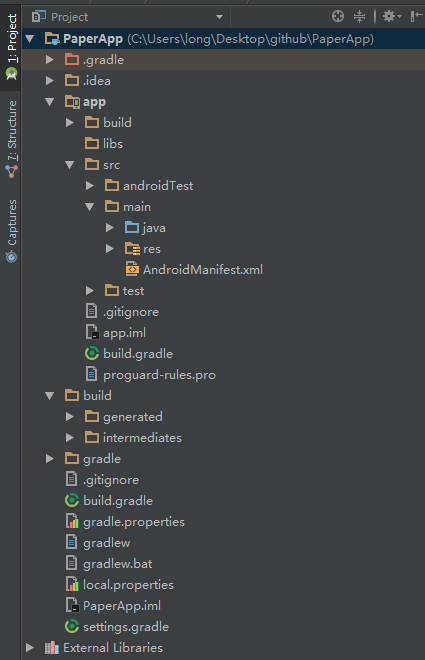
# 开始工作

## 创建应用

我们先创建一个app如下：



项目结构如下：



分为3块：

1. 编译系统：

.gradle 版本由gradle中的wrapper控制,通过gradlew执行

1. 配置文件：

Gradle.properties 与gradle相关属性的配置

Build.gradle 与gradle编译相关的配置

Local.properties 与本地环境配置相关

1. 应用模块：

build 编译后的文件存在的位置（包括最终生成的apk也在这里面）

libs 依赖的库所在的位置（jar和aar)

src 源代码所在的目录

src/main 主要代码所在位置（src/androidTest)就是测试代码所在位置了

src/main/assets android中附带的一些文件

src/main/java 最最重要的，我们的java代码所在的位置

src/main/jniLibs jni的一些动态库所在的默认位置(.so文件)

src/main/res android资源文件所在位置

src/main/AndroidManifest.xml AndroidManifest不用介绍了吧~

build.gradle 和这个项目有关的gradle配置，相当于这个项目的Makefile，一些项目的依赖就写在这里面

proguard.pro 代码混淆配置文件

## 编译，打包应用

构建过程使用许多工具，生成很多中间文件。在Android Studio中，运行gradle的编译任务会完成这一切的构建。构建过程非常灵活可配置。下图描述了参与编译的工具和流程：

标准的构建系统会合并配置在product flavors, build types, dependencies上的资源。过程如下：

1. Android Asset打包工具（AAPT）编译应用程序资源文件，如 AndroidManifest.xml文件，activity的布局XML文件。之后生成R.java，这样就可以从Java代码中引用资源。
2. aidl工具将.aidl文件转化成java的接口。
3. 编译java源码生成class文件。
4. Dex工具将第三方library和cllass文件转化成dex 文件。
5. Apkbuilder将编译的资源，没有编译的资源，dex文件打包成apk文件。
6. Apk文件创建成功后需要签名才能安装到设备上面。最后对齐apk，使应用程序运行时占用更少的内存。如果应用程序达到方法个数65k的限制，需要进行分包处理。
7. 编译后生成的apk输出在build文件中。

## 安装应用