

AndroidStudio使用教程(第四弹)

本文为转载文章：原文地址：<https://github.com/CharonChui/AndroidNote>

Gradle

讲解到这里我感觉有必要说明一下 Gradle。

Gradle 是一个基于 Apache Ant 和 Apache Maven 概念的项目自动化建构工具。它使用一种基于 Groovy 的特定领域语言来声明项目设置，而不是传统的 XML。更多介绍请直接参考[Gradle](#)或 Google 搜索。

以下是为什么Android Studio选择Gradle的主要原因：

- 使用领域专用语言（Domain Specific Language）来描述和处理构建逻辑。（以下简称DSL）- 基于Groovy。DSL可以混合各种声明元素，用代码操控这些DSL元素达到逻辑自定义。- 支持已有的Maven或者Ivy仓库基础建设 - 非常灵活，允许使用best practices，并不强制让你遵照它的原则来。- 其它插件时可以暴露自己的DSL和API来让Gradle构建文件使用。- 允许IDE集成，是很好的API工具

Overview

AndroidStudio build 系统是一个你可以用来 build, test, run, package your apps 的工具。 build 系统与 Android Studio 之间是独立的，所以你可以在 Android Studio 中或者 command line 中去执行。写完自己的应用程序之后，你可以用 build 系统来做以下事情：- 自定义、配置和扩展 build 过程。
- 用同一个工程根据不同的特性来创建多个 APK。
- 重复利用代码和资源。
Android Studio 灵活的 build 系统能让你不用修改项目和核心文件而完成上面所有的工作。

Overview of the Build System

Android Studio 的构建系统包含一个 Gradle 的 Android 插件， Gradle 是一个管理依赖关系并且允许自定义构建逻辑的高级构建工具。许多软件都用 Gradle 来进行构建。 Gradle 的 Android 插件并不依赖 Android Studio，虽然 Android Studio 全部集成了 Gradle，这就意味着：
- 你可以用命令行方式去构建 Android 应用或者是在一些没有安装 Android Studio 的机器上。- 你可以用命令行构建时的配置和逻辑来在 Android Studio 中进行构建 Android 项目。不管你是通过命令行还是远程机器或者是 Android Studio 来进行构建的产出都是一致的。

Build configuration

项目的构建配置都在 Gradle build files 中，都是些符合 Gradle 要求的选项和语法， Android 插件并不依赖 Android 通过 build 文件来配置一下几个方面：
- Build variants 构建系统能对同一个项目通过不同的配置生成多个 APK，当你想构建多个不同版本时这是非常有用的，因为不用为他们建立多个不同的项目。
- Dependencies 构建系统管理项目的依赖关系，并且支持本地已经远程 entries 构建系统能够通过构建配置去指定清单文件中某些元素的值。 当你想生成一些包名、最低 SDK 版本或目标 SDK 版本不同的 APK 时是非常有用的。
- Signing 构建系统能在 build 配置中设置指定的签名，在科一些特定元素通信的语言。

Build by convention

Android Studio 构建系统对项目结构和一些其他的构建选项做了一些很好的默认规范声明，如果你的项目符合这些条件， Gradle 的构建文件就非常简单了。如果你的项目不符合其中的一些要求，灵活的构建系统也允许你去配置几乎所有的构建选项。例如你项目的源码没有放在默认的文件夹中，你可以通过 build 文件去配置它的位置。

Projects and modules

Android Studio 中的 Project 代表了一个完整的 Android 应用，每个 Project 中可以有一个或多个 Module。Module 是应用中可以单独 build, test 或 debug 的组件。Module 中包含应用中的源码和资源文件， Android Studio 中的 Project 包含以下三种 Module：
- 包含可复用代码的 Java library modules。构建系统对 Java library module 会生成一个 JAR 包。
- 有可复用 Android 代码和资源的 Android library modules。对该 library modules 构建系统会生成一个 AAR(Android ARchive) 包。
- 有应用代码或者也可能是依赖其他 library modules 的 Android application modules，虽然很多 Android 应用都只包含一个 application module。对于 application modules 构建系统会生成一个 APK 包。
Android Studio projects 在 project 的最外城都包含一个列出所有 modules 的 Gradle build file，每个 module 也都包含自己的 Gradle build file。

Dependencies

Android Studio 的构建系统管理着依赖项目并且支持 module 依赖，本地二进制文件依赖和远程二进制文件的依赖。

- Module Dependencies 一个项目的 module 可以在构建未见中包含一系列所依赖的其他 modules，在你构建这个 module 的时候，系统回去组装这些所包含的 modules。
- Local Dependencies 如果本地文件系统中有 module 所依赖的二进制包如 JAR 包，你可以在该 module 中的构建文件中声明这些依赖关系。
- Remote Dependencies 当你的依赖是在远程仓库中，你不需要去下载他们然后拷贝到自己的工程中。 Android Studio 支持远程 Maven 依赖。 Maven 是一个流行的项目管理工具，它可以使用仓库帮助组织项目依赖。

许多优秀的软件类库和工具都在公共的 Maven 仓库中，对于这些依赖只需按照远程仓库中不同元素的定义来指定他们的 Maven 位置即可。构建系统使用的 Maven 位置格式是 group:name:version。例如 Google Guava 16.0.1版本类库的 Maven 坐标是 com.google.guava:guava:16.0.1。
Maven Central Repository 现在被广泛用于分发许多类库和工具。

下面分别为以上三种依赖关系的配置：

```
dependencies {
    compile project(":name")
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.google.guava:guava:16.0.1'
}
```

Build tasks

Android Studio 构建系统定义了一系列的构建任务，高级别的任务调用一些产出必要输出的任务。构建系统提供了 project tasks 来构建 app 和 module tasks 来独立的构建 modules。

可以通过 Andorid Studio 或者命令行看到当前可用任务的列表，并且执行里面的任务。

The Gradle wrapper

Android Studio 项目包含了 Gradle wrapper，包括：
- A properties file
- A shell script for Windows platforms
- A shell script for Mac and Linux platforms
声明：需要把这些文件提交到代码控制系统。

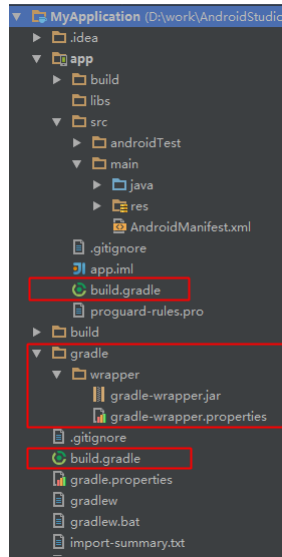
使用 Gradle wrapper (而不用本地安装的 Gradle)能确保经常运行配置文件中配置的 Gradle 版本。通过在配置文件中定义最新的版本来确保你的工程一直使用最新版的 Gradle。

Android Studio 从你项目中的 Gradle wrapper 目录读取配置文件，并且在该目录运行 wrapper，这样在处理多个需要不同 Gradle 版本的项目时就会游刃有余。声明：Android Studio 不使用 shell 脚

本，所以对于他们的任何改变在 IDE 构建时都不会生效，你应该在 Gradle build files 中去设置自定义的逻辑。

你可以在开发及其或者是一些没有安装 Android Studio 的及其上使用命令行运行 shell 脚本来构建项目。

直接上图：



- 邮箱：charon.chui@gmail.com
- Good Luck!