

Gradle for Android 第一篇(从 Gradle 和 AS 开始)

**neu**

3.1k • 1 • 5

发布于

2015-12-31

这一系列暂不更新，相关技术讨论，请移步微信群，谢谢，希望大家多多支持！

正如大家所见，这是本英文书，而由于国内的gradle翻译资料不全，所以特次开辟专栏，翻译 gradle for android这本书，同时添加自己的心得体会以及在实际工作上的实战，希望大家能够喜欢。

这会是一个系列，所以如果你看完这篇文章，请看下列文章：

[Gradle for Android 第二篇\(Build.gradle入门 \)](#)

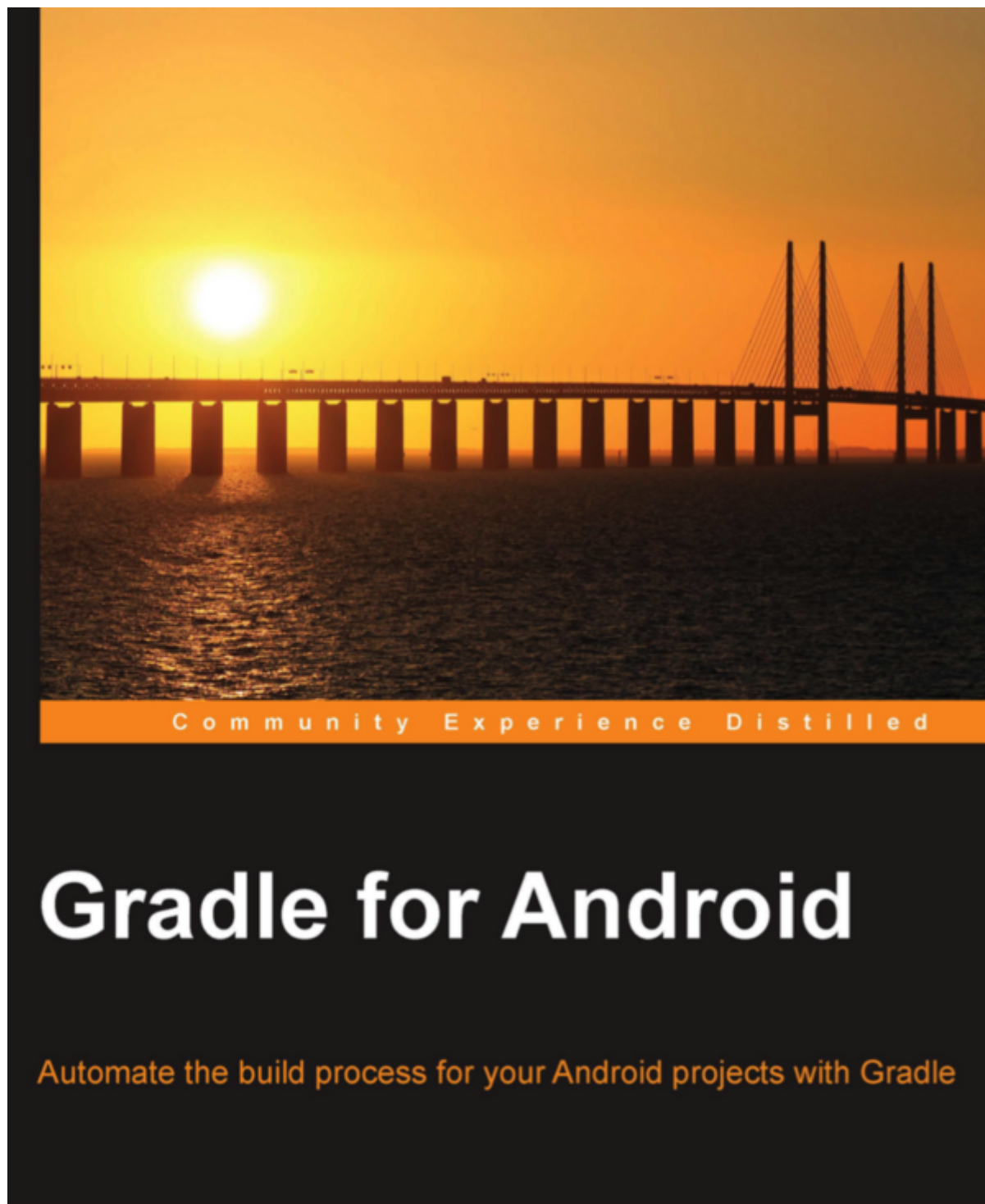
[Gradle for Android 第三篇\(依赖管理 \)](#)

[Gradle for Android 第四篇\(构建变体 \)](#)

[Gradle for Android 第五篇\(多模块构建 \)](#)

[Gradle for Android 第六篇\(测试 \)](#)

[Gradle for Android 第七篇\(Groovy入门 \)](#)



如果你是名Android开发新手，或者是名从eclipse切换到Android studio的新手，那么我强烈建议您follow我的文章，正如封面所见，利用gradle构建工具来自动构建你的Android项目。废话不多说，我们直接开始吧。

今天主要介绍Android studio工具的使用，以及cradle基础入门，使用cradle wrapper和如何从eclipse迁移到Android studio。

这篇文章记于2015.12.30，Android studio正式版本已经开发到1.5，而预览版已经到了2.0，所以转到Android studio吧。

当你第一次打开Android studio的时候，有一个视图显示你即将创建的环境以及确保你使用了最新的Android SDK和必要的google依赖包，同时会让你选择是否创建AVD，这样你就可以使用模拟器

了。在这儿多说几句：

1. 尽量使用Android studio 2.0, 因为它真的变快了, 而其模拟器的速度也提升了不少, 我使用至今, 也无发现任何bug,所以完全不用担心。
2. 如果使用模拟器开发Android, 尽量使用Genymotion模拟器, 尽管其现在的Android6.0仍然有很多bug,但是在其以下版本, 其速度还是非常快的, 利用模拟器开发, 为虚拟机安装文件夹浏览器, 是及时查看SQLite表文件利器, 具体操作办法, 可以google。
3. 尽量使用最新的23.0.0以上的构建版本。

理解基本的Gradle

如果你想创建一个Android project基于gradle,那么你必须写一个构建脚本, 这个文件通常称之为build.gradle,你可能已经觉察到了, 当我们查看这一脚本, gradle会为我们提供很多默认的配置以及通常的默认值, 而这极大的简化了我们的工作, 例如ant和maven,使用他们的时候, 我们需要编写大量的配置文件, 而这很恶心。而gradle得默认配置, 如果你需要使用自己的配置, 完全可以简单的去重写他们就好。

Gradle脚本不是像传统的xml文件那样, 而是一种基于Groovy的动态DSL, 而Groovy语言是一种基于jvm的动态语言。

你完全不用担心, 你在使用gradle的时候, 还需要去学习Groovy语言, 该语言很容易阅读, 并且如果你已经学习过java的话, 学习Groovy将不会是难事, 如果你想开始创建自己的tasks和插件, 那么你最好对Groovy有一个较深的理解, 然而由于其基于jvm,所以你完全可能通过纯正的java代码或者其他任何基于jvm的语言去开发你自己的插件, 关于插件开发, 我们后续将会有相关介绍。

Project和tasks

在grade中的两大重要的概念, 分别是project和tasks。每一次构建都是有至少一个project来完成, 所以Android studio中的project和Gradle中的project不是一个概念。每个project有至少一个tasks。每一个build.grade文件代表着一个project。tasks在build.gradle中定义。当初始化构建进程, gradle会基于build文件, 集合所有的project和tasks,一个tasks包含了一系列动作, 然后它们将会按照顺序执行, 一个动作就是一段被执行的代码, 很像Java中的方法。

构建的生命周期

一旦一个tasks被执行, 那么它不会再次执行了, 不包含依赖的Tasks总是优先执行, 一次构建将会经历下列三个阶段:

1. 初始化阶段：project实例在这儿创建，如果有多个模块，即有多个build.gradle文件，多个project将会被创建。
2. 配置阶段：在该阶段，build.gradle脚本将会执行，为每个project创建和配置所有的tasks。
3. 执行阶段：这一阶段，gradle会决定哪一个tasks会被执行，哪一个tasks会被执行完全依赖开始构建时传入的参数和当前所在的文件夹位置有关。

build.gradle的配置文件

基于grade构建的项目通常至少有一个build.gradle，那么我们来看看Android的build.gradle：

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:1.2.3'  
    }  
}
```

这个就是实际构建开始的地方，在仓库地址中，我们使用了JCenter，JCenter类似maven库，不需要任何额外的配置，gradle还支持其他几个仓库，不论是远程还是本地仓库。

构建脚本也定义了一个Android构建工具，这个就是Android plugin的来源之处。Android plugin提供了所有需要去构建和测试的应用。每个Android应用都需要这么一个插件：

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

插件用于扩展gradle脚本的能力，在一个项目中使用插件，这样该项目的构建脚本就可以定义该插件定义好的属性和使用它的tasks。

注意：当你在开发一个依赖库，那么你应该使用'com.android.library'，并且你不能同时使用他们2个，这将导致构建失败，一个模块要么使用Android application或者Android library插件，而不是二者。

当使用Android 插件的时候，Android标签将可以被使用，如下所示：

```
android {  
    compileSdkVersion 22  
    buildToolsVersion "22.0.1"  
}
```

更多的属性我们将在第二章中进行讨论。

项目结构

和eclipse对比来看，Android studio构建的结构有很大的不同：

```
MyApp
├── build.gradle
├── settings.gradle
└── app
    ├── build.gradle
    ├── build
    ├── libs
    └── src
        └── main
            ├── java
            │   └── com.package.myapp
            └── res
                ├── drawable
                ├── layout
                └── etc.
```

gradle项目通常在根文件夹中包含一个build.gradle，使用的代码在app这个文件夹中，这个文件夹也可以使用其他名字，而不必要定义为app,例如当你利用Android studio创建一个project针对一个手机应用和一个Android wear应用的时候，模块将被默认叫做application和wearable。

gradle使用了一个叫做source set的概念，官方解释：一个source set就是一系列资源文件，其将会被编译和执行。对于Android项目，main就是一个source set，其包含了所有的资源代码。当你开始编写测试用例的时候，你一般会把代码放在一个单独的source set，叫做androidTest，这个文件夹只包含测试。

开始使用Gradle Wrapper

gradle只是一个构建工具，而新版本总是在更迭，所以使用Gradle Wrapper将会是一个好的选择去避免由于gradle版本更新导致的问题。Gradle Wrapper提供了一个windows的batch文件和其他系统的shell文件，当你使用这些脚本的时候，当前gradle版本将会被下载，并且会被自动用在项目的构建，所以每个开发者在构建自己app的时候只需要使用Wrapper。所以开发者不需要为你的电脑安装任何gradle版本，在mac上你只需要运行gradlew，而在windows上你只需要运行gradlew.bat。

你也可以利用命令行./gradlew -v来查看当前gradle版本。下列是wrapper的文件夹：

```
myapp/
├─ gradlew
├─ gradlew.bat
├─ gradle/wrapper/
│   ├─ gradle-wrapper.jar
│   └─ gradle-wrapper.properties
```

可以看到一个bat文件针对windows系统，一个shell脚本针对mac系统，一个jar文件，一个配置文件。配置文件包含以下信息：

```
#Sat May 30 17:41:49 CEST 2015
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/
gradle-2.4-all.zip
```

你可以改变该url来改变你的gradle版本。

使用基本的构建命令

使用你的命令行，导航到你的项目，然后输入：

```
$ gradlew tasks
```

这一命令将会列出所以可运行的tasks,你也可以添加--all参数，来查看所有的task。当你在开发的时候，构建项目，你需要运行assemble task通过debug配置：

```
$ gradlew assembleDebug
```

该任务将会创建一个debug版本的app,同时Android插件会将其保存在MyApp/app/build/outputs/apk目录下。

除了assemble，还有三个基本的命令：

- check 运行所有的checks,这意味着运行所有的tests在已连的设备或模拟器上。
- build 是check和assemble的集合体。

- clean 清楚项目的output文件。

保持旧的eclipse文件结构

关于如何将eclipse项目导入Android studio本文不再介绍。

```
android {
    sourceSets {
        main {
            manifest.srcFile 'AndroidManifest.xml'
            java.srcDirs = ['src']
            resources.srcDirs = ['src']
            aidl.srcDirs = ['src']
            renderscript.srcDirs = ['src']
            res.srcDirs = ['res']
            assets.srcDirs = ['assets']
        }
        androidTest.setRoot('tests')
    }
}
```

在grade文件中配置，将会保存eclipse目录结构，当然，如果你有任何依赖的jar包，你需要告诉gradle它在哪儿，假设jar包会在一个叫做libs的文件夹内，那么你应该这么配置：

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
}
```

该行意为：将libs文件夹中所有的jar文件视为依赖包。

总结

通过本文，我们可以学习到gradle的优势，以及为什么要使用gradle,我们简单的看了看Android studio，以及其是如何帮助我们生成build文件。

同时我们学习了Gradle Wrapper，其让我们维护以及分享项目变得更加简单，我们知道了如何创建一个新的项目在Android studio中，以及如何从eclipse迁移到Android studio并且保持目录结构。

同时我们学习了最基本的gradle task和命令行命令。在下一篇文章中，我们将会定制自己的build文件。

gradle android-studio  android



neu

3.1k 声望 1.2k 粉丝

关注作者