Android Studio + Gradle的组合用起来非常方便，很多第三方开源项目也早都迁移到了Studio，为此今天就来介绍下查看、编译并导入第三方开源项目的方法。

Sublime + Terminal编译并查看源码

首先来给大家介绍一种简便并且个人最喜欢的一种办法。很多时候我们在GitHub上看到一个不错的开源项目，一般有两种需求，阅读源码和查看运行效果，如果是单纯的查看源码我更喜欢用一些轻量级编辑器，如vim，sublime等，vim不是很熟练，所以个人一种都习惯用sublime来查看并阅读源码（实际开发还是Android Studio、Eclipse等IDE更有效率）；至于查看运行效果首先得需要apk，对于一些开源项目已提供apk下载，那就好办，直接安装到手机即可。 而对于一些没有提供apk下载的，则需要自己手动编译打包。

下面以9GAG为例来讲解下命令行Gradle编译的过程。

•1、切换到9GAG项目的根目录，执行 ./gradlew -v 来查看下项目所用的Gradle版本

如果你是第一次执行会去下载Gradle，这个过程如果不翻墙非常慢，建议翻墙



紧接着下载成功会看到如下信息：

------------------------------------------------------------

Gradle 2.2.1

------------------------------------------------------------

Build time: 2014-11-24 09:45:35 UTC

Build number: none

Revision: 6fcb59c06f43a4e6b1bcb401f7686a8601a1fb4a

Groovy: 2.3.6

Ant: Apache Ant(TM) version 1.9.3 compiled on December 23 2013

JVM: 1.7.0\_60 (Oracle Corporation 24.60-b09)

OS: Mac OS X 10.9.5 x86\_64

•2、接着执行 ./gradlew clean

执行这个命令会去下载Gradle的一些依赖，下载成功并编译通过时会看到如下信息：

:app:clean UP-TO-DATE

:extras:ShimmerAndroid:clean UP-TO-DATE

BUILD SUCCESSFUL

•3、最后执行 ./gradlew build

这个命令会直接编译并生成相应的apk文件，如果看到如下字样就代表build成功了

BUILD SUCCESSFUL

Total time: 31.456 secs

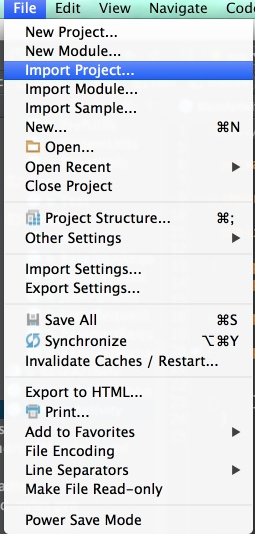
紧接着在 9GAG/app/build/outputs/apk 目录下会看到类似于app-debug-unaligned.apk, app-release-unsigned.apk等，看名字应该能理解意思，unaligned代表没有进行zip优化的，unsigned代表没有签名的。然后就可以直接安装apk查看运行效果了。

以上是我个人习惯的一种阅读第三方源码的习惯，关于上面提到的一些gradle命令大家应该还不理解，后面会一一进行说明的。

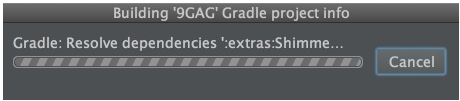
导入Android Studio

但是如果你还是想导入Android Studio的话，下面就来简单介绍下导入Studio的方法以及一些注意事项。

•1、选择File->Import Project, 选择本地9GAG项目的目录

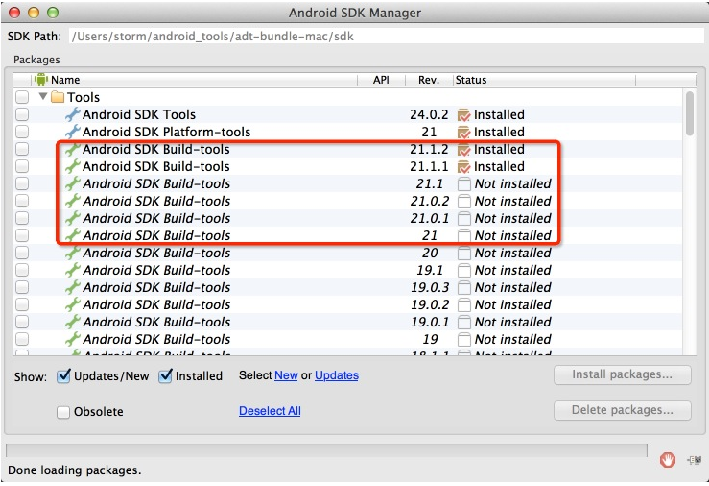


•2、第一次依然会下载Gradle，其实自己在命令行已经下载过了，但是这次依然还要下载一次（依然要翻墙），可能是个bug



•3、导入之后你需要注意以下几个地方

1. 每个Module下的 build.gradle 下的buildToolsVersion，即9GAG/app/build.gradle 和 9GAG/extras/ShimmerAndroid/build.gradle , 可以打开 SDK Manager 查看本地你安装的 SDK Build-tools, 如下图，如果相应版本没有安装请先下载。



1.项目根目录下的 build.gradle 下gradle插件的版本，如 9GAG/build.gradle 的内容：

classpath 'com.android.tools.build:gradle:1.0.0'

Android Studio 1.0必须指定gradle插件1.0的版本

1.Gradle Wrapper的版本，在 9GAG/gradle/wrapper/gralde-wrapper.properties 文件里有如下内容：

Thu Dec 18 16:02:24 CST 2014

distributionBase=GRADLE\_USER\_HOME

distributionPath=wrapper/dists

zipStoreBase=GRADLE\_USER\_HOME

zipStorePath=wrapper/dists

distributionUrl=https://services.gradle.org/distributions/gradle-2.2.1-all.zip

上述内容制定了gradle的版本使用2.2.1，在Studio中如果你手动改了gradle插件的版本会有提示要不要更新Gradle Wrapper的版本。

如果以上几个地方确定版本是ok的，那么导入Android Studio应该没有问题，有些老的项目可能gradle版本比较老，仍然在使用一些老的语法，如果编译还通不过参考Android Studio系列教程四--Gradle基础更改成最新的语法。

Gradle常用命令

上面大家接触了一些命令如 ./gradlew -v ./gradlew clean ./gradlew build, 这里注意是./gradlew, ./代表当前目录，gradlew代表 gradle wrapper，意思是gradle的一层包装，大家可以理解为在这个项目本地就封装了gradle，即gradle wrapper， 在9GAG/gradle/wrapper/gralde-wrapper.properties文件中声明了它指向的目录和版本。只要下载成功即可用grdlew wrapper的命令代替全局的gradle命令。

理解了gradle wrapper的概念，下面一些常用命令也就容易理解了。

•./gradlew -v 版本号

•./gradlew clean 清除9GAG/app目录下的build文件夹

•./gradlew build 检查依赖并编译打包

这里注意的是 ./gradlew build 命令把debug、release环境的包都打出来，如果正式发布只需要打Release的包，该怎么办呢，下面介绍一个很有用的命令 assemble, 如

•./gradlew assembleDebug 编译并打Debug包

•./gradlew assembleRelease 编译并打Release的包

除此之外，assemble还可以和productFlavors结合使用，具体在下一篇多渠道打包进一步解释。

•./gradlew installRelease Release模式打包并安装

•./gradlew uninstallRelease 卸载Release模式包

下一篇来介绍用Gradle进行方便的多渠道打包，以及项目中完整的Gradle配置。有问题或建议欢迎大家直接博客留言。