Android Gradle 构建系统·初探

kvh 何畅彬

About Me

- 主要从事 Android 和 Backend Service 开发
- 目前创业，bugtags.com, 帮助开发团队快速迭代
- 对技术和创业感兴趣
- 博客：kvh.io
- 公众号：mobdev
- 《拥抱 Android Studio》

Outline

- About This Topic
- Prerequisites
- Android New Build System
- Pearls in Code
- Gradle Plugin in Action
- Summary

About This Topic

- The very beginning
- Turn out to be huge
- Step by step
- Overview & action

The Very Beginning

- Users' problem in Bugtags SDK integration
- FAQ of beginners
- 《拥抱 Android Studio 》系列博客解决了部分用户的疑惑
- We should go further

Huge Topic

- Source code: core lib > **2.5GB**, whole system **25GB**
- More than **50** sub-projects
- Android team is working on it since **2013**(even earlier)
- Under **heavy development**(instant-run,jack jill,data-binding, native development support, etc.)

Overview & Action

- Project overview
- Pearls found in code
- Learn from code
- Plugin in action

Prerequisites

- Basics of Groovy and Gradle
  ref: http://kvh.io/cn/embrace-android-studio-groovy-gradle.html

- Basics of Android Gradle build tool and process
  ref:http://kvh.io/cn/embrace-android-studio-indepth.html

Warmup: Groovy

- Dynamic language on JVM
- Improve development productivity
- Functional programming
- DSL support

What's DSL

- Domain Specific Language 领域限定的语言
  一系列处理问题的规则
  如何处理这个问题的语言描述
  语言描述的解析器

- 我个人认为是：
  用配置的方式来写码

Warmup: Gradle

- Modern build tools for Java and other languages
- Build on Groovy
- Build file: every project has a build file
- Easy to learn and use
- DSL, not XML

```
bugtags {
    mappingUploadEnabled false
}
```

Warmup: Gradle Cont'

- Project[rootProject, subProject]
- Plugin
- Task
- Action

Gradle Cli

- gradle & gradlew

  use specific version of gradle as you wish

  https://docs.gradle.org/current/userguide/gradle_wrapper.html

- wrapper structure

  gradlew (UN*X Shell script)
  gradlew.bat (Windows batch file)
  gradle/wrapper/gradle-wrapper.jar (Wrapper JAR)
  gradle/wrapper/gradle-wrapper.properties (Wrapper properties)

  distributionUrl=https\://services.gradle.org/distributions/gradle-2.10-all.zip

Gradle Cli Cont'

- Execute cli, UN*X & Windows
  ./gradlew (UN*X)
  gradlew.bat (Windows)

- Some useful commands
  clean build --info
  projects
  tasks
  assemble
  build -x test

- Ref
  http://kvh.io/cn/gradle-indepth-cmd.html

Android Gradle Build Tool

- One plugin package

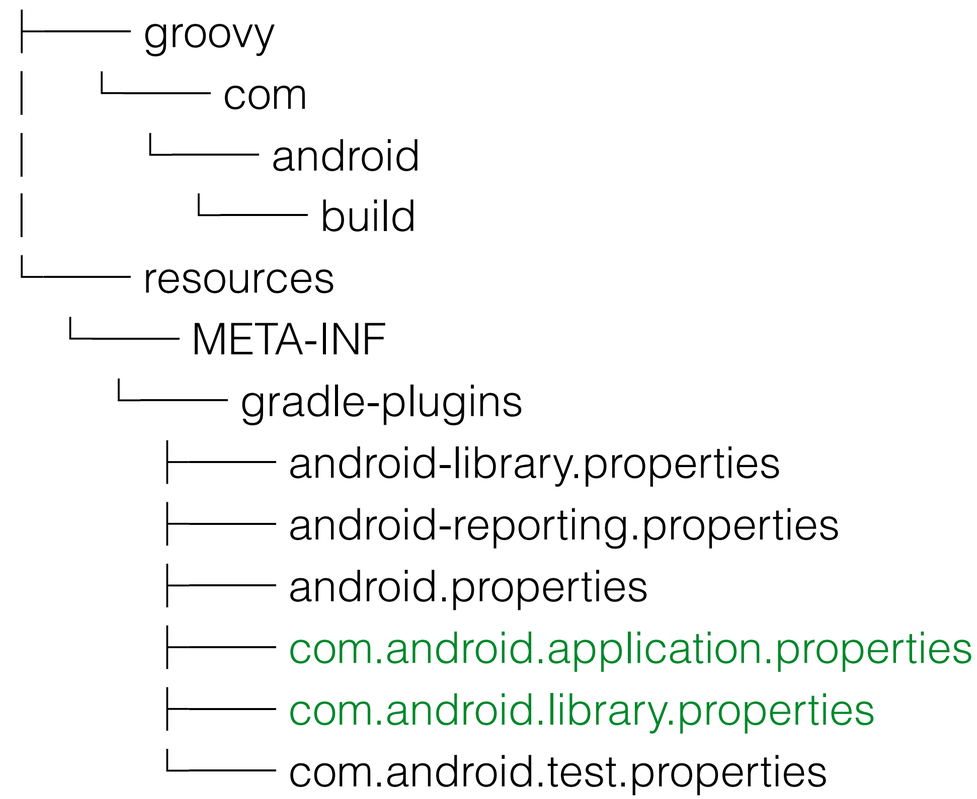
```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:2.1.0'
    }
}
```

- Two plugins：

```
apply plugin: 'com.android.application'：apk
apply plugin: 'com.android.library'：aar
```

Android Gradle Plugin Package Directory Structure

src/main



Android Gradle Build Output

- apk:

```
.
├── AndroidManifest.xml
├── META-INF
├── classes.dex
├── res
└── resources.arsc
```

- aar：
  - `/AndroidManifest.xml` (mandatory)
  - `/classes.jar` (mandatory)
  - `/res/` (mandatory)
  - `/R.txt` (mandatory)
  - `/assets/` (optional)
  - `/libs/*.jar` (optional)
  - `/jni/<abi>/*.so` (optional)
  - `/proguard.txt` (optional)
  - `/lint.jar` (optional)

Android New Build System

- Website
- Bintray Repository & Dependencies
- Source Code
- Project Structure

Website

- http://tools.android.com/
- http://tools.android.com/tech-docs
  My favorite site
- http://tools.android.com/build
  how to build the [build system]

Android Gradle Plugin DSL for Users

- What we are familiar with

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

android {
        defaultConfig{}
        buildTypes{}
        lintOptions{}
        packageOptions{}
        //...
}
```

- Explain what you can config inside

  http://google.github.io/android-gradle-dsl/current/index.html

Bintray Repository

- Android Gradle Plugin: version, developer https://bintray.com/android/android-tools/com.android.tools.build.gradle
- Whole system: 40 projects https://bintray.com/android/android-tools
- Sometimes our library can depends on some android-tool projects

Android Gradle Plugin's Dependencies

com.android.tools.build:gradle Gradle Plug-in for Android

> com.android.tools.build:gradle-core Core Library for Android Gradle Plug-in

->

com.android.tools.build:builder Android Builder library

com.android.tools.build:lint

com.android.tools.build:transform-api

com.android.tools.build:gradle-api Android Gradle API

com.android.databinding:compilerCommon Data Binding Compiler Common

...

Source Code

- Git repository
  https://android.googlesource.com/platform/tools/base/+/gradle_2.0.0

- Use USTC source instead
  https://lug.ustc.edu.cn/wiki/mirrors/help/aosp

Source Code Cont'

- Download repo tool
  mkdir ~/bin
  PATH=~/bin:$PATH
  curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
  ## if the above is not working, try the following
  ## curl https://storage-googleapis.lug.ustc.edu.cn/git-repo-downloads/repo > ~/bin/repo
  chmod a+x ~/bin/repo

- Init: put source code in **~/android/gradle**
  mkdir -p ~/android/gradle
  cd ~/android/gradle
  repo init -u git://mirrors.ustc.edu.cn/aosp/platform/manifest **-b gradle_2.0.0**

- Sync
  repo sync

- Wait
  about 8 hours

Project Structure

- Directories

  cd ~/android/gradle

  ```
  ├── external
  ├── frameworks
  ├── prebuilts
  ├── sdk
  └── tools //main project
  ```

- Find out projects

  cd ~/android/gradle/tools

  ./gradlew projects

Part of the Project List

Root project 'tools'

+--- Project ':base'

| +--- Project ':base:annotations'

| +--- Project ':base:ant-tasks'

| +--- Project ':base:api-generator'

| +--- Project ':base:archquery'

| +--- Project ':base:asset-studio'

| +--- Project ':base:builder'

| +--- Project ':base:builder-model'

| +--- Project ':base:builder-test-api'

| +--- Project ':base:chartlib'

| +--- Project ':base:common'

| +--- Project ':base:ddmlib'

| +--- Project ':base:docs'

Learn from Subproject's build.gradle File

- Every sub-project is a gradle project
- Read build.gradle file, find out what is it.

eg. **com.android.tools.build:builder**

cd ~/android/gradle/tools/base/build-system/builder

```
group = 'com.android.tools.build'
archivesBaseName = 'builder'
version = rootProject.ext.buildVersion
project.ext.pomName = 'Android Builder library'
project.ext.pomDesc = 'Library to build Android applications.'
```

Try to Build

- Gradle build guideline
  http://tools.android.com/build#TOC-Building-the-Android-Gradle-Plugin

- Create output directory

  cd ~/android/gradle
  mkdir -p out/dist

- Assemble

  cd ~/android/gradle/tools
  ./gradlew clean assemble

Try to Build Cont'

- May fail for JDK version check(i don't have JDK 1.6 in my mac)

```
vim ~/android/gradle/tools/buid.gradle

if (!jvmVersion.startsWith(requiredVersion)) {
        //throw new RuntimeException("Tools need to be compiled with Java $requiredVersion,
you are using Java $jvmVersion.")
}
```

- build result in out/dist

Pearls in Code

- Learn from the project structure
- Learn how Android team members code

buildSrc Project

- ~/android/gradle/tools/buildSrc
- Actually **a plugin for sub-project**
- Will run **before** any other sub-project

Gradle Plugin Types

- Three types of gradle plugin:

  build script:simple
  buildSrc project
  standalone project: share

- Ref:

  http://kvh.io/cn/embrace-android-studio-gradle-plugin.html

buildSrc Project Cont'

- internal plugins

```
.
└── META-INF
    └── gradle-plugins
        ├── clone-artifacts.properties
        ├── jsoup.properties
        ├── license-report.properties
        ├── native-setup.properties
        ├── offline-repo.properties
        ├── pegdown.properties
        ├── presubmit-runner.properties
        ├── sdk-files.properties
        ├── sdk-java-lib.properties
        ├── sdk-tools.properties
        └── windows-setup.properties
```

buildSrc Project Cont'

- Gradle build script collections, sub-project reference them ~/android/gradle/tools/buildSrc/base

- List

```
.
├── base.gradle
├── baseJava.gradle
├── bintray.gradle
├── build.gradle
├── gradle.properties
├── gradlew
├── gradlew.bat
├── javadoc.gradle
├── publish.gradle
├── release.gradle
├── settings.gradle
├── update_from_root.sh
└── version.gradle
```

android.jar as classpath

- Android library with pure Java code
- You will **never** need an android library module
- eg. ~/android/gradle/tools/base/instant-run/instant-run-server

android.jar as classpath Cont'

- How it works

```
apply plugin: 'java'

File androidJar = new File(System.env.ANDROID_HOME + '/platforms/android-23/
android.jar');
if (!androidJar.exists()) {

throw new RuntimeException("android-23 android.jar not found at " + androidJar.absolu
tePath)
}


configurations {
    provided
}

dependencies {
    compile project(':base:instant-run:instant-run-runtime')
    provided files(androidJar)
}

sourceSets {
    main { compileClasspath += configurations.provided }
}
```

Dependency Configurations

- dependencies {
  compile|provided('a:b:1.0.0')
  }
- Customization for dependency
- ~/android/gradle/tools/base/build-system/gradle-core

Dependency Configurations Cont'

- includeInJar

```
configurations {
    includeInJar
}

dependencies{
  includeInJar(':insta-run:server')
}

jar {
    into('instant-run') {
        from configurations.includeInJar
    }
}
```

Dependency Transitive

- A -> B -> C
- compile(A) will have B, C downloaded
- compile(A){
  transitive = false
  }
- will stop download B, C

Dependency Transitive Cont'

- When depends on an aar package
- eg.
- io.kvh.public:library:1.0.0 is an aar package
- **depends on** com.mcxiaoke.volley:library:1.0.19

Dependency Transitive Cont'

using

- compile('io.kvh.public:library:1.0.0@aar')

or

```
compile('io.kvh.public:library:1.0.0') {
    transitive = false
}
```

- **volley** library **won't be** downloaded

Code Analysis Tool

- findbug: static code analysis tool
- jacoco: code coverage analysis tool

Gradle Plugin in Action

- Understand task dependency
- Learn about Android Gradle tasks
- Write a simple plugin
- Ref
  http://kvh.io/cn/embrace-android-studio-gradle-plugin.html

Task Dependency

- Gradle task dependencies
  https://docs.gradle.org/current/userguide/more_about_tasks.html

- Eg.

```
task hello << {
        println 'Hello,'
}

task world << {
    println 'World!'
}
```

Task Dependency Cont'

```
task intro(dependsOn: hello) << {
    println 'intro'
}

world.finalizedBy hello
```

- ./gradlew intro
  hello intro
- ./gradlew world
  world hello

Task Dependency Cont'

- Do things in right time
- Find the right task to dependsOn or finalizedBy

Android Gradle Plugin Tasks

- ./gradlew -p app tasks
- or
- ./gradlew -p app clean assembleRelease -x test -x lint
- -x: remove tasks of little importance

Critical Tasks

- **merge**Release**Resources: merge resources**
- **process**Release**Manifest: process manifest**
- **process**Release**Resources: process & compile resources**
- **compile**Release**JavaWithJavac: compile Java**
- **transformClassesAndResourcesWithProguardFor**Release**: proguard**
- **transformClassesWithDexFor**Release**: dex java class**
- **package**Release**: package aar or apk**

Our Plugin Requirements

- Upload proguard mapping files to server
- Run before packaging finish
- Run after proguard, when mapping file is generated

dependsOn

- task uploadMapping
- def variantName = 'release|debug'
- "package${variantName}".**dependsOn** uploadMapping
- uploadMapping.**dependsOn** "transformClassesAndResourcesWithProguardFor${variantName}"

Summary

- Groovy & Gradle knowledge speedup your development: know what, how, why
- Gradle Plugin: automation, copy output files to some dir
- SDK development: split module, and merge
- Next step: go further

Ads

- bugtags.com 帮助 APP 开发者快速迭代
- 上线一年，上万企业，近十万的 APP
- 优秀的 SDK(Android | iOS) 工程师
- 后端工程师(PHP)、大数据工程师、产品/用户运营
- 简历邮箱 he.changbin@bugtags.com

Q&A

mobdev 公众号

Thanks

- Good day