

第六讲 Gradle项目管理

初版时间: 2015年5月1日

初版制作者: 林焕彬

教材版本号: szlanyou-V1.1

修订时间:

修订者:

教材版本号: szlanyou-V1.1

目录



Gradle简介

Gradle关键知识点

Gradle常用命令

Gradle实践

总结

一、Gradle介绍 1-1

- Gradle是一个基于Apache Ant和Apache Maven概念的项目自动化构建工具。它使用一种基于Groovy的特定领域语言(DSL)来声明项目设置，抛弃了基于XML的各种繁琐配置。
- 面向Java应用为主。当前其支持的语言限于Java、Groovy、Kotlin和Scala，计划未来将支持更多的语言。



一、Gradle的安装配置 1-2

- 机器已安装JDK，并配置好环境变量。
- 下载**Gradle**，解压，配置环境变量：
 - 增加GRADLE_HOME指向gradle-4.1
 - 把%GRADLE_HOME%\bin变量加入Path变量
 - 查看Gradle是否安装成功（cmd下）：

```
C:\Users\ly-linhuanbin>gradle -v
```

```
-----  
Gradle 4.1  
-----
```

```
Build time: 2017-08-07 14:38:48 UTC
```

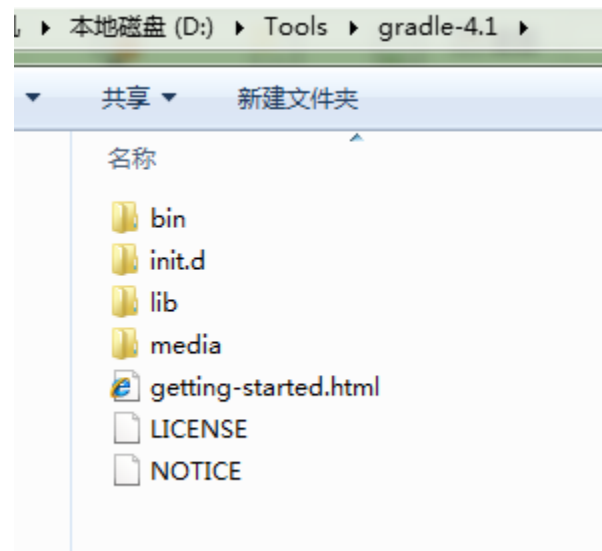
```
Revision: 941559e020f6c357ebb08d5c67acdb858a3defc2
```

```
Groovy: 2.4.11
```

```
Ant: Apache Ant(TM) version 1.9.6 compiled on June 29 2015
```

```
JVM: 1.8.0_60 (Oracle Corporation 25.60-b23)
```

```
OS: Windows 7 6.1 amd64
```



一、与Maven的异同 1-3

- **依赖管理系统：** Maven为Java世界引入了一个依赖管理系统，可以用groupId、artifactId、version组成的Coordination（坐标）唯一标识一个依赖。Gradle在设计的时候基本沿用了Maven的这套依赖管理体系，不同的是：
 - 引用依赖方面变得非常简洁
 - Maven和Gradle对依赖项的scope有所不同
 - Gradle支持动态的版本依赖
 - 在解决依赖冲突方面Gradle的实现机制更加明确
- **一致的项目结构：** Maven和Gradle都沿用同一套标准的Java项目结构，不过Gradle作为JVM上的构建工具，也同时支持groovy、scala等源代码的构建，甚至支持Java、groovy、scala语言的混合构建。

一、与Maven的异同 1-4

- **一致的构建模型：**Maven特意设置了标准的项目构建周期，而Gradle在构建模型上则非常灵活。在Gradle世界里可以轻松创建一个task，并随时通过depends语法建立与已有task的依赖关系。对于Java项目的构建来说，Gradle是通过名为java的插件来包含了一个对Java项目的构建周期，这等于Gradle本身直接与项目构建周期是解耦的。
- **插件机制：**Maven和Gradle设计时都采用了插件机制，Gradle插件设计更简单。

一句话总结：**Maven**更加标准，**Gradle**更加简洁灵活

目录



Gradle简介

Gradle关键知识点

Gradle常用命令

Gradle实践

总结

二、Gradle关键知识点 2-1

- 现在我们都在谈领域驱动设计，Gradle本身的领域对象主要有Project和Task。Project中可以添加不同的Task，Project为Task提供了执行上下文。
- 一个Task表示一个逻辑上较为独立的执行过程，比如编译Java源代码，拷贝文件，打包Jar文件，甚至可以是执行一个系统命令或者调用Ant。另外，一个Task可以读取和设置Project的Property以完成特定的操作。

```
task hello {  
    doLast {  
        println 'Hello Gradle'  
    }  
}
```


二、Gradle构建 2-2

- **声明式构建：**Gradle是一种声明式的构建工具。在执行时，Gradle并不会一开始便顺序执行build.gradle文件中的内容，而是分为两个阶段，第一个阶段是配置阶段，然后才是实际的执行阶段。在配置阶段，Gradle将读取所有build.gradle文件的所有内容来配置Project和Task等，比如设置Project和Task的Property，处理Task之间的依赖关系等。
- **增量式构建：**在增量式构建中，我们为每个Task定义输入（inputs）和输出（outputs），如果在执行一个Task时，如果它的输入和输出与前一次执行时没有发生变化，那么Gradle便会认为该Task是最新的（UP-TO-DATE），因此Gradle将不予执行。

二、Gradle依赖管理 2-3

- 一个Java项目总会依赖于第三方类库，比如Apache commons；通常来说，这种依赖的表示形式都是将第三方的Jar文件放在自己项目的classpath下。在声明对第三方类库的依赖时，我们需要先告诉Gradle在什么地方去获取这些依赖，即配置Gradle的Repository。

```
repositories {  
    mavenLocal()  
    mavenCentral()  
    maven{ url 'http://maven.oschina.net/content/groups/public/' }  
}
```

本地.m2目录

Maven中央仓库

二、Gradle依赖管理 2-4

- 我们可以通过dependencies()方法加入实际的依赖项：

```
dependencies {  
    compile 'org.slf4j:slf4j-log4j12:1.7.2'  
    testCompile 'junit:junit:4.8.2'  
}
```

声明依赖的**groupId**、**artifactId**、**version**

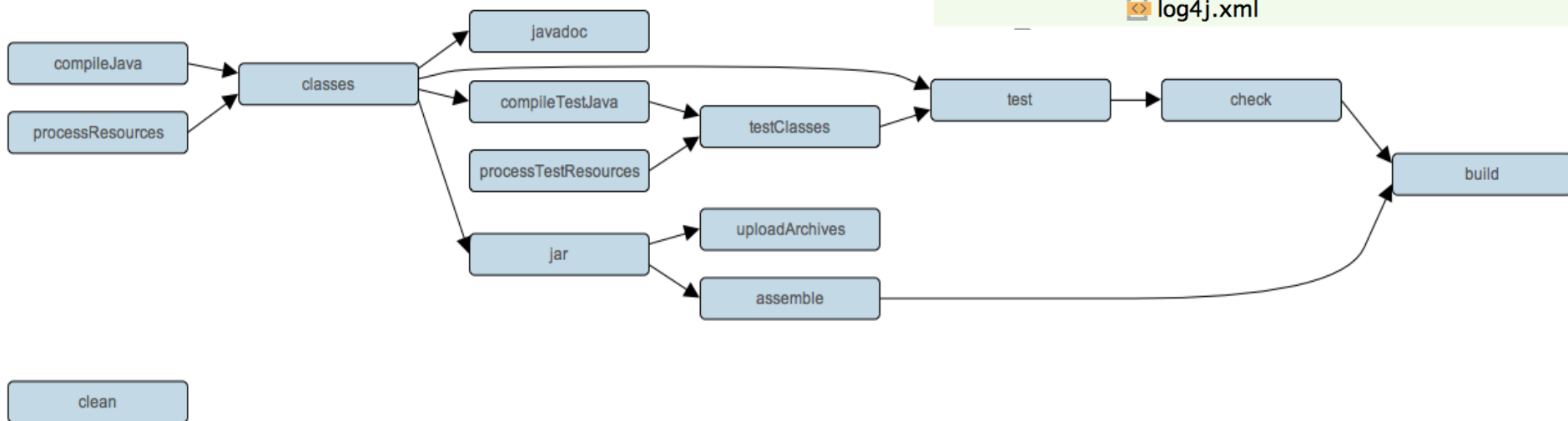
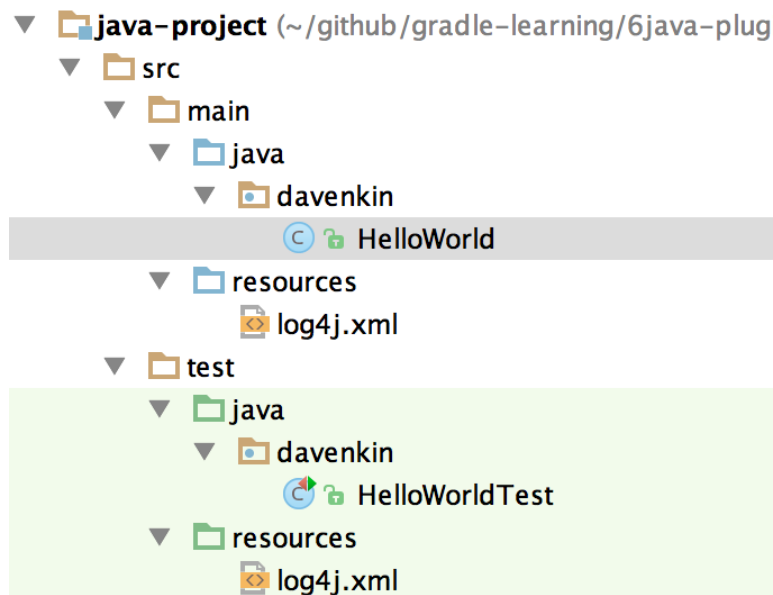
- Java Plugin会自动定义compile和testCompile，分别用于编译Java源文件和编译Java测试源文件。另外，java Plugin还定义了runtime和testRuntime这两个Configuration，分别用于在程序运行和测试运行时加入所配置的依赖。

二、Gradle Plugin介绍 2-5

- 和Maven一样，Gradle只是提供了构建项目的一个框架，真正起作用的是Plugin。Gradle在默认情况下为我们提供了许多常用的Plugin，其中包括有构建Java项目的Plugin，还有War，Ear等。与Maven不同的是，Gradle不提供内建的项目生命周期管理，只是java Plugin向Project中添加了许多Task，这些Task依次执行，为我们营造了一种如同Maven般项目构建周期。
- Gradle在采用了Maven一致的目录结构的同时，还融入了自己的一些概念，即source set。我们完全可以在build.gradle文件中重新配置这些source set所对应的目录结构，同时，我们还可以创建新的source set。

二、Gradle Java Plugin 2-6

- Gradle Java Plugin定义了与Maven完全一致的项目布局
- 也定义了构建生命周期，包括编译主代码、处理资源、编译测试代码、执行测试、上传归档等等任务：



目录



Gradle简介

Gradle关键知识点

Gradle常用命令

Gradle实践

总结

三、Gradle常用命令 3-1

检测Gradle版本: `gradle -v` 或者 `gradle -version`

获取帮助选项: `gradle -h` 或者 `gradle -help`

显示可用的任务列表: `gradle tasks`

创建Java项目:

```
task create {  
    sourceSets*.java.srcDirs*.each { it.mkdirs() }  
    sourceSets*.resources.srcDirs*.each { it.mkdirs() }  
}
```

三、Gradle常用命令 3-2

加入Eclipse插件: `apply plugin: 'eclipse'`

创建Eclipse项目: `gradle eclipse`

加入Java插件: `apply plugin: 'java'`

执行项目构建: `gradle build`

三、Gradle常用操作 3-3

- 指定默认任务

```
defaultTasks 'bootRun'
```

- 为build task增加uploadArchives任务

```
build.dependsOn 'uploadArchives'
```

- build task忽略test任务

```
加参数 -x test
```

目录



Gradle简介

Gradle关键知识点

Gradle常用命令

Gradle实践

总结

四、Gradle实践 4-1

项目实践

- ▣ 编写build.gradle文件，创建一个可以运行的spring boot项目。
- ▣ 熟悉build.gradle文件的各项组成元素的作用
- ▣ 实践Gradle常用命令

完成时间：10分钟

共性问题集中讲解

四、Gradle实践 4-2

```
buildscript {
    repositories {
        maven{ url 'http://172.26.161.125:8888/nexus/content/repositories/e4s/' }
        mavenLocal()
        mavenCentral()
        jcenter()
    }
    dependencies {
        classpath("org.springframework.boot:spring-boot-gradle-plugin:1.4.5.RELEASE")
    }
}

apply plugin: 'java'
apply plugin: 'org.springframework.boot'

repositories {
    maven{ url 'http://172.26.161.125:8888/nexus/content/repositories/e4s/' }
    jcenter()
    mavenLocal()
    mavenCentral()
}

dependencies {
    compile ("org.springframework.boot:spring-boot-starter-web")
    compile ("org.springframework.boot:spring-boot-starter-data-jpa")
    compile ("io.springfox:springfox-swagger2:2.2.2")
    compile ("io.springfox:springfox-swagger-ui:2.2.2")
    compile ("mysql:mysql-connector-java:5.1.21")
    compile ("javax.el:el-api:2.2.1-b04")
    testCompile ("org.springframework.boot:spring-boot-starter-test")
}
```

四、Gradle实践 4-3

● 执行gradle命令：

```
E:\Documents\课件\workspace\gradle-prj>gradle build
:compileJava
:processResources UP-TO-DATE
:classes
:findMainClass
:jar
:bootRepackage
:assemble
:compileTestJava NO-SOURCE
:processTestResources NO-SOURCE
:testClasses UP-TO-DATE
:test NO-SOURCE
:check UP-TO-DATE
:uploadArchives
:build

BUILD SUCCESSFUL in 3s
6 actionable tasks: 5 executed, 1 up-to-date
```

四、Gradle实践 4-4

- ## 执行gradle命令，默认执行bootRun任务:

```
E:\Documents\课件\workspace\gradle-prj>gradle
:compileJava UP-TO-DATE
:processResources UP-TO-DATE
:classes UP-TO-DATE
:findMainClass
:bootRun
```

[illegible]

```
2017-09-04 16:42:37.401 INFO 8464 --- [main] com.qhong.SpringBootDemoApplication : Starting SpringBootDemoApplication on B011685 with PID 8464 (started by ly-linhuanbin in E:\Documents\璇玑欢\workspace\gradle-prj)
2017-09-04 16:42:37.403 INFO 8464 --- [main] com.qhong.SpringBootDemoApplication : No active profile set, falling back to default profiles: default
```

目录



Gradle简介

Gradle关键知识点

Gradle常用命令

Gradle实践

总结

总结



提问

Gradle是一个基于Apache Ant和Apache Maven概念的项目
自动化构建工具

Gradle关键的几个知识点

Gradle几个重要过程的命令

build.gradle是Gradle项目的核心配置文件

Gradle实践，正常运行一下Spring boot项目

Thank you

Q&A

