

Kodstuga: Gradle

Levon Saldamli

levon.saldamli@idainfront.se

Build automation

- Compile source code
 - Manage and download dependencies
- Generate or process resources
- Run checks
 - Unit tests, integration tests
 - Run code quality tools (checkstyle, findbugs)
 - Generate reports
- Running and debugging during development
- Create packages (jar, zip, war)
- Create documentation (javadoc)
- Upload to repositories

Build tools

- Ant
 - XML
 - Java
 - Classpath, local files
- Maven
 - XML
 - Java
 - Dependency management
- Gradle
 - Domain Specific Language (DSL)
 - Java/Groovy
 - Dependency management
- JavaScript: Grunt + Bower

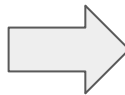
Maven vs Gradle

```
<dependencies>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.12</version>
  </dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <artifactId>maven-checkstyle-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

← Maven

Gradle



```
apply plugin: 'checkstyle'

dependencies {
    compile "org.slf4j:slf4j-api:1.7.12"
}
```

Groovy

- JVM
- Mostly Java compatible
- Additional classes/methods
- Enhanced java classes
 - <http://groovy-lang.org/gdk.html>

```
new File("notes.txt").eachLine { line, index -> println "${index}: ${line}" }
```

- Script

```
println "In gradle script"
task hello {
    ...
}
```

Groovy

- Closures

```
def fn(Closure closure) {  
    closure("FN calling")  
}
```

```
fn { input ->  
    println "Closure called with: ${input}"  
}
```

Gradle

Gradle

- Since 2007
- JVM based
- Implemented in Java, Groovy outer layer (DSL)
- 100% Free Open Source - Apache Standard License 2.0
- Current version 2.13
 - New software model in 3.0
- Incremental builds
- More than Java
 - C/C++, Android
 - <http://gradle.org/getting-started-native/>
- Gradleware
 - Company behind Gradle
 - Development, consulting, support, training

Gradle commands

- **gradle -version**
- **gradle tasks**
- **gradle build**
Default build directory: build/
- **gradle clean**
Cleans build output
- **gradle wrapper**

Tasks

Tasks

```
task hello
```

```
task hello {  
    description = 'Greeting'  
    doFirst {  
        println "Hello!"  
    }  
}
```

```
tasks["hello"].doFirst { println "Hello" }
```

```
hello << {  
    println " world"  
}
```

Task dependencies

```
task hello {  
    dependsOn prepare  
    doFirst {  
        println "Hello!"  
    }  
}
```

```
hello.dependsOn prepare
```

```
hello {  
    dependsOn prepare  
}
```

Incremental builds

- Inputs/Outputs
 - Files, directories, properties

```
task hello {  
    inputs.property "userName", userName  
    doFirst {  
        println "Hello, ${userName}"  
    }  
}
```

- upToDateWhen { ... }
- Override
 - gradle clean
 - --rerun-tasks

Tasks types

```
task hello(type: Exec) {  
    group = 'Publishing'  
    description = 'Greeting'  
    workingDir "${buildDir}"  
    commandLine 'echo', '"Hello world"'  
}
```

Custom tasks

- Groovy class
- Extend from existing task, e.g. `DefaultTask`
- Groovy properties
 - `String property = 'defaultValue'`
 - `property = 'newValue' // setProperty('newValue')`
 - `println property // println getProperty()`

```
class MyTask extends DefaultTask {  
    String property = 'defaultValue'  
  
    @TaskAction  
    def perform() {  
        }  
}
```

Applying plugins

- From other scripts
 - `apply from: 'path_to_script.gradle'`
- From binary distributions
 - With plugin id:
`apply plugin: 'java'`
 - With binary type:
`apply plugin: PluginClass`
- <https://plugins.gradle.org/>

Dependencies

Dependencies

```
apply plugin: 'java'

repositories {
    maven { url "http://repo.idainfront.se/content/groups/internal" }
    mavenCentral()
}

dependencies {
    compile group: 'org.slf4j', name: 'slf4j-api', version: '1.7.21'
    compile 'org.slf4j:slf4j-api:1.7.21'

    runtime 'org.slf4j:slf4j-log4j12:1.7.21'

    testCompile 'junit:junit:4.12'
}
```

- gradle dependencies

Configurations

```
apply plugin: 'java'

configurations {
    serverConf
    myCompile.extendsFrom compile
}

dependencies {
    serverConf 'org.slf4j:slf4j-api:1.7.21'
    myCompile('org.springframework:spring-core:4.2.6.RELEASE') {
        exclude module: 'unwanted-module'
    }
}

configurations.serverConf.each { file ->
    println file
}
```

- ConfigurationContainer
 - <https://docs.gradle.org/current/dsl/org.gradle.api.artifacts.ConfigurationContainer.html>

Processing java resources

```
apply plugin: 'java'
```

```
processResources { // Copy task, from src/main/resources -> build/resources/main
    filesMatching("**/*.properties") {
        expand(...)
        filter(...)
    }
}
```

- Copy task:
 - <https://docs.gradle.org/current/dsl/org.gradle.api.tasks.Copy.html>

Multi project builds

- **settings.gradle**
 - Searched for in parent folders
- Configuration injection
 - All projects or subprojects from parent **build.gradle**
- Project dependencies
- Partial builds
- **gradle projects**

Multi project builds

- `settings.gradle`

```
include "projectA"  
include "projectB"  
include "projectC" // default directory name == project name
```

```
settings.gradle  
build.gradle  
projectA/  
    build.gradle  
    src/  
projectB/  
    build.gradle  
    src/  
projectC/  
    build.gradle  
    src/
```

Multi project builds

- `build.gradle`

```
allprojects {  
}  
  
subprojects {  
    apply plugin: 'java'  
    apply plugin: 'eclipse'  
    sourceCompatibility = 1.8  
    targetCompatibility = 1.8  
}  
  
project(':projectB') {  
    dependencies {  
        compile project(':projectA')  
    }  
}
```

Multi project builds

- `build.gradle`

```
allprojects {  
}  
  
subprojects {  
    apply plugin: 'java'  
    apply plugin: 'eclipse'  
    sourceCompatibility = 1.8  
    targetCompatibility = 1.8  
}
```

- `projectB/build.gradle`

```
dependencies {  
    compile project(':projectA')  
}
```


Multi project builds

- `gradle :projectA:build`
- `gradle build`
- `gradle check`
- `gradle buildNeeded`
- `gradle buildDependents`

Thank you

Labs:

<https://github.com/levsa/gradle-workshop-ida>


```
apply plugin: 'java'

sourceCompatibility=1.6
targetCompatibility=1.6

ext {
    junitVersion = '4.11'
}

dependencies {
    compile 'org.slf4j:slf4j-api:1.7.12'

    testCompile "junit:junit:${junitVersion}"
}

jar {
    manifest {
        attributes "Built-By": System.getProperty("user.name")
    }
}

tasks.withType(Test) {
    maxParallelForks = 4
}

tasks.withType(JavaCompile) {
    doFirst {
        logger.info "${project.name}: sourceCompatibility: ${sourceCompatibility}"
        logger.info "${project.name}: targetCompatibility: ${targetCompatibility}"
    }
}
```

Applying plugins

- Buildscript dependencies:

```
buildscript {  
    repositories {  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.jfrog.bintray.gradle:gradle-bintray-plugin:0.4.1'  
    }  
}  
  
apply plugin: 'com.jfrog.bintray'
```