Migrate Maven to Gradle — Springboot Web Application

Romil Jain Follow
Nov 16, 2017 · 3 min read

Migrate from Maven to Gradle

If you have Maven Project and thinking to move to Gradle then don't think much **Just do it**.

It's easy and simple with some gradle commands. Later on you can enhance your build.gradle as per the project need.

If you don't have gradle and need help in installation and setup, Please refer here.

Once gradle is setup, then goto project directory of maven spring-boot application where you have pom.xml. In my case, I have one spring-boot maven application with name spring-boot (Application built on Spring-Boot and Hibernate). Let's run —

```
$ gradle init

> Task :init

Maven to Gradle conversion is an incubating feature.

BUILD SUCCESSFUL in 1s

2 actionable tasks: 2 executed
```

This will create build.gradle from pom.xml and other gradle related files and folder like Gradle Wrapper.

Once done you can delete pom.xml and can start polishing your gradle application :)

Now import this project in eclipse (if you don't have gradle plugin setup in eclipse, please refer here) and check build.gradle. You may see something like this depending upon your java version used for the application

build.gradle

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

group = 'com.springboot.poc'
version = '1.0'

description = """spring-boot Maven Webapp"""

sourceCompatibility = 1.5
targetCompatibility = 1.5
tasks.withType(JavaCompile) {
    options.encoding = 'UTF-8'
}



repositories {

     maven { url "http://repo.maven.apache.org/maven2" }
}
dependencies {
    compile group: 'org.springframework.boot', name: 'spring-boot-starter-web', version:'1.5.7.RELEASE'
    compile group: 'org.springframework.boot', name: 'spring-boot-starter-data-jpa', version:'1.5.7.RELEASE'
    compile group: 'mysql', name: 'mysql-connector-java', version:'5.1.44'
    compile group: 'commons-dbcp', name: 'commons-dbcp', version:'1.4'
    compile group: 'org.apache.commons', name: 'commons-collections4', version:'4.1'
    compile group: 'org.springframework.boot', name: 'spring-boot-legacy', version:'1.1.0.RELEASE'
    providedCompile group: 'org.springframework.boot', name: 'spring-boot-starter-tomcat', version:'1.5.7.RELEASE'
}

```

Default build.gradle created via gradle init

Lets update this file to have latest version of gradle spring-boot plugin.

build.gradle

```
buildscript {
    repositories {
        mavenCentral()
    }
    dependencies {
        classpath("org.springframework.boot:spring-boot-gradle-plugin:1.5.8.RELEASE")
    }
}

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

description = "gradle-springboot"

war {
    baseName = 'gradle-springboot'
    version =  '1.0'
}

def javaVersion = JavaVersion.VERSION_1_8

sourceCompatibility = javaVersion
targetCompatibility = javaVersion

tasks.withType(JavaCompile) {
    options.encoding = 'UTF-8'
}

repositories {
     mavenCentral()
}

dependencies {
    compile group: 'org.springframework.boot', name: 'spring-boot-starter-web', version:'1.5.8.RELEASE'
    compile group: 'org.springframework.boot', name: 'spring-boot-starter-data-jpa', version:'1.5.8.RELEASE'
    compile group: 'mysql', name: 'mysql-connector-java', version:'5.1.44'
    compile group: 'commons-dbcp', name: 'commons-dbcp', version:'1.4'
    compile group: 'org.apache.commons', name: 'commons-collections4', version:'4.1'
    compile group: 'org.springframework.boot', name: 'spring-boot-legacy', version:'1.1.0.RELEASE'
    providedCompile 'org.springframework.boot:spring-boot-starter-tomcat:1.5.8.RELEASE'
}

```

Now to go terminal and run

```
$ gradle build
Download
https://repo1.maven.org/maven2/org/springframework/boot/...
.
.....................................................
.
.....................................................
.
```

```
BUILD SUCCESSFUL in 9s
```

```
5 actionable tasks: 5 executed
```

Update gradle wrapper (To know more about gradle wrapper please refer here)

```
$ gradle wrapper --gradle-version 4.3.1
```

```
BUILD SUCCESSFUL in 0s
```

```
1 actionable task: 1 up-to-date
```

You can see war (in my case gradle-springboot-1.0.war) created under build/libs

Now we can deploy this WAR on tomcat server in eclipse or run ‘./gradlew bootRun’ or ‘gradle bootRun’ to start the application. I will prefer gradle wrapper to run it.

```
$ ./gradlew bootRun
```

```
> Task :bootRun
```

```
.   ____          _            __ _ _
/\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
\\/  ___)| |_)| | | | | || (_| |  ) ) ) )
'  |____| .__|_| |_|_| |_\__, | / / / /
```

```
=========|_|==============|___/=/_/_/_/

:: Spring Boot ::        (v1.5.8.RELEASE)
.....................................................
.....................................................
.....................................: Tomcat
started on port(s): 8080 (http)
```

Now you can test your application either in browser or using CURL command.

. . .

In case you want to dockerize this application click here. If you want to install and setup docker and dockerize a basic gradle spring boot application then click here.