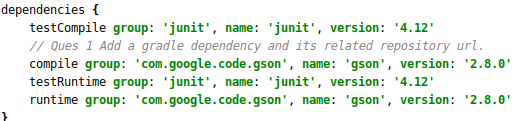
Dependency Management using Gradle

Exercise

1. Add a gradle dependency and its related repository url.



1. Using java plugin, make changes in the manifest to make the jar executable. Using java -jar JAR\_NAME, the output should be printed as "Hello World"



Code: In the gradle.build file.

jar **{**

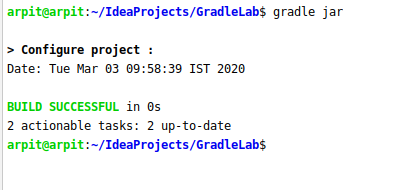
manifest **{**

attributes **'Main-Class'**: **'App'**

**}**

**}**

1. Differentiate between the different dependency scopes: compile, runtime, testCompile, testRuntime using different dependencies being defined in your build.gradle.
2. Compile: It is the dependency used to compile the source code and build the application.It is mainly used for production and for our Main classes.
3. Runtime: It is the dependency used to run the application and it is exposed to the consumers.It is in runtime and test classpath.
4. testCompile: It is a group of dependencies that you need only for testing.A junit is the testing framework just required for testing.Since it's not needed at runtime, it's not going to be included in the released package.It is used for testing classes.
5. testRuntime: It is used to declare test runtime dependencies.This type of dependencies are required to run the test cases at run time.By default, it includes runtime and test compile dependencies.
6. Create a custom plugin which contains a custom task which prints the current date-time. Using that plugin in your project, execute that task after the jar task executes.



Code: In the gradle.build file.

**class** DatePlugin **implements** Plugin<Project> {

**void** apply(Project project) {

project.task(**'ShowDate'**) **{**

dependsOn(**"build"**)

println **"Date: "** + **new** Date()

**}**

}

}

apply **plugin**: DatePlugin

1. Instead of using default source set, use src/main/javaCode1, src/main/javaCode2 to be taken as code source. Make sure that the JAR created contains files from both the directories and not from src/main/java.



Code: In the gradle.build file.

sourceSets **{**

main **{**

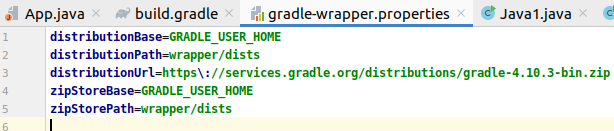
java.srcDirs = [**'src/main/javaCode1'**, **'src/main/javaCode2'**]

**}**

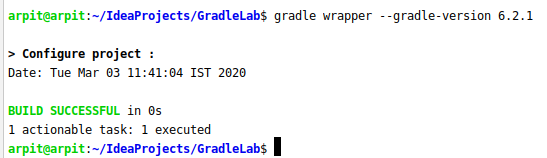
**}**

1. Override the Gradle Wrapper task to install a different version of gradle. Make sure that the task written in Q4 also executes with it.

Before Running the Gradle Wrapper command:



After running the command:



Command: gradle wrapper --gradle-version 6.2.1

Code: to run it after the wrapper task.

**class** DatePlugin **implements** Plugin<Project> {

**void** apply(Project project) {

project.task(**'ShowDate'**) **{**

dependsOn(**"build"**)

println **"Date: "** + **new** Date()

**}**

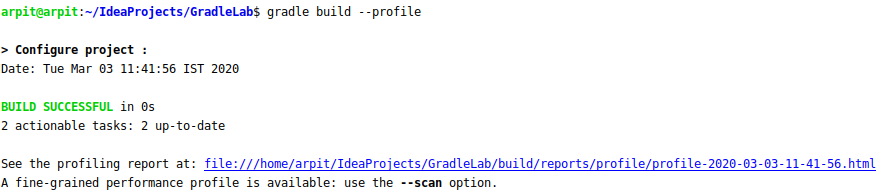
}

}

In this as the wrapper command internally calls the build command therefore just put the dependsOn to build.

1. Run the gradle profile command and attach the resulting files.

After running the command gradle build --profile:



The following is the resulting html page:

