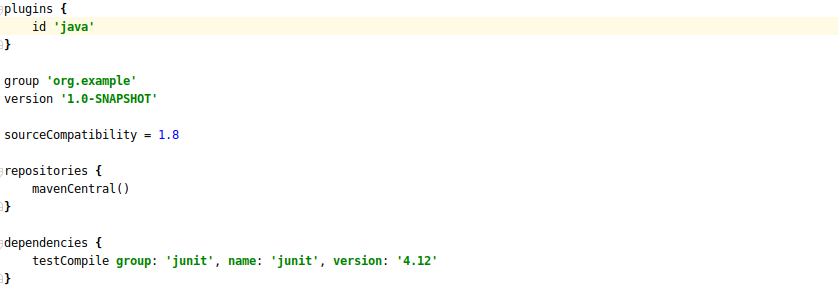
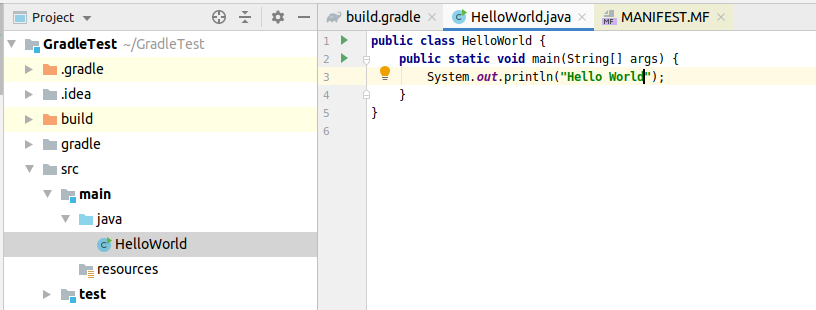
**Dependancy Management using Gradle**

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

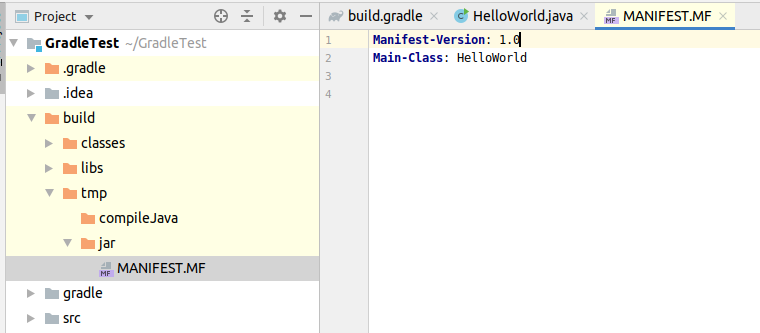


1. **Using java plugin, make changes in the manifest to make the jar executable. Using java -jar JAR\_NAME, the output should be printed as "Hello World"**

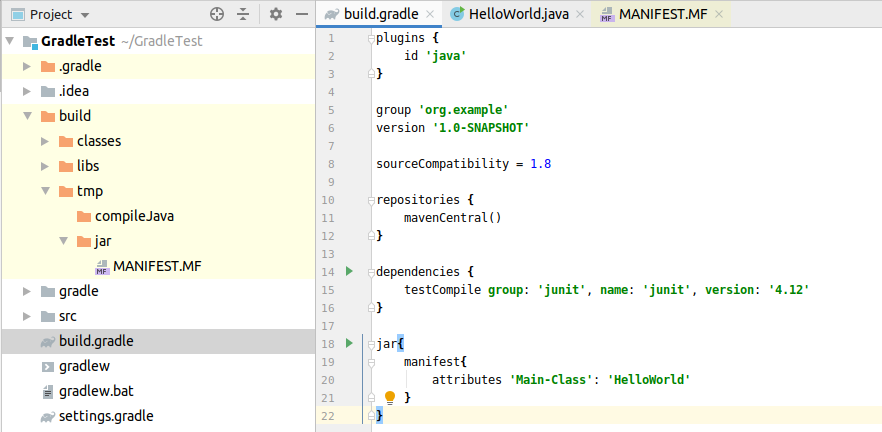
Step 1: Create a class and write the program to return “Hello World”



Step 2: Go to the MANIFEST.MF file and add the main class want to execute while execuring the jar file



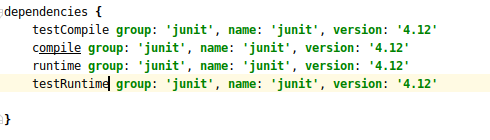
Step 3: Then we will add the jar attribute to build.gradle file



Output:



1. **Differentiate between the different dependency scopes: compile, runtime, testCompile, testRuntime using different dependencies being defined in your build.gradle.**



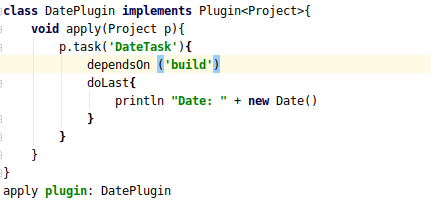
*compile:* dependencies we need during compile time

*testCompile:* dependencies which are required to compile the tests of our project

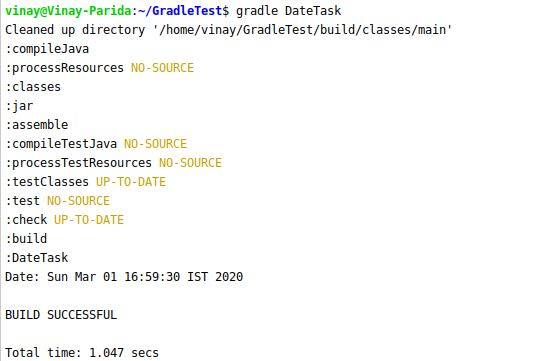
*runtime:* dependencies we need during run time

*testRuntime:* dependencies which are required during when our test are running

1. **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.**

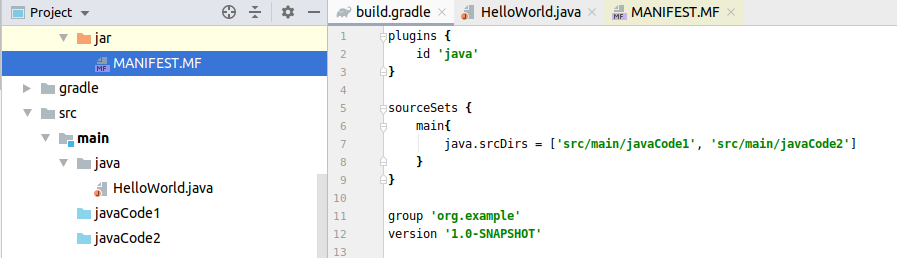


Now every time we run *gradle DateTask*, build will start first

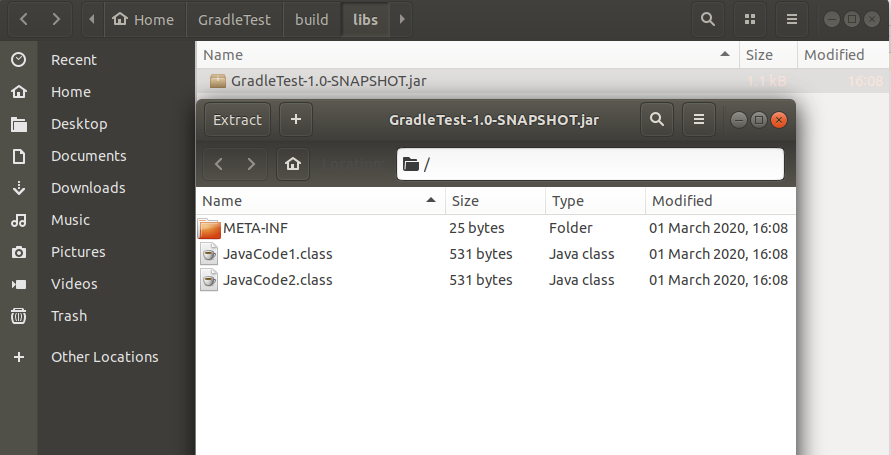


1. **Instead of using default source set, use src/main/javaCode1, src/main/javaCode2 to be taken as code source. Make sure that the JAR created contains files from both the directories and not from src/main/java.**

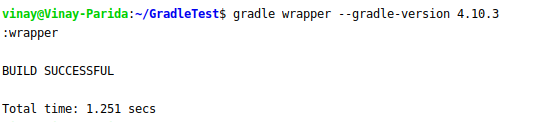
First we create 2 directories inside src/main and then changing the main sourceSets to these directories



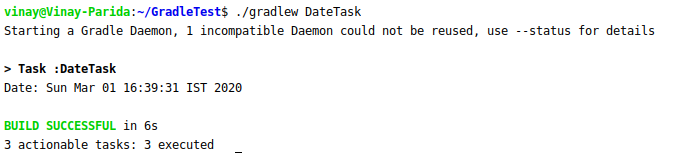
If we select jar file in ‘show in files’ we will get this:



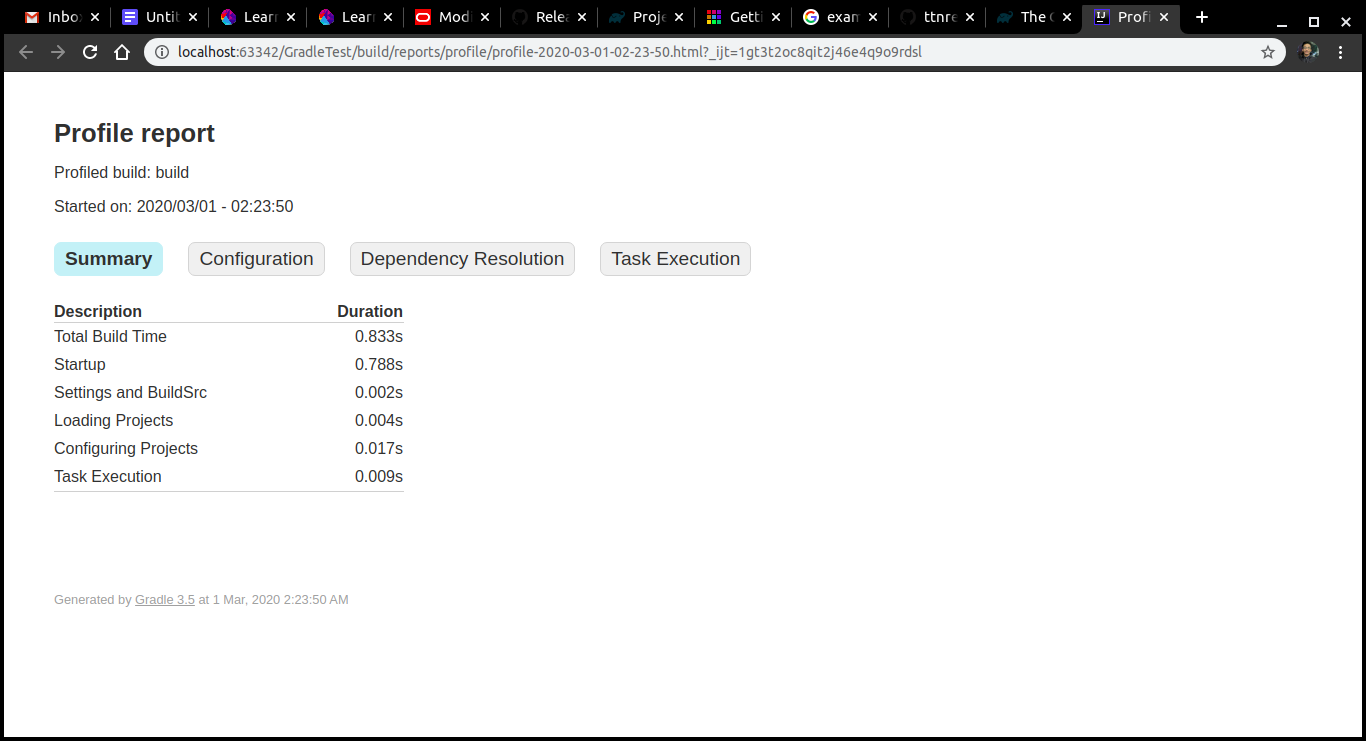
1. **Override the Gradle Wrapper task to install a different version of gradle. Make sure that the task written in Q4 also executes with it.**



Now to execute the DateTask



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



### 