**Gradle - Build and Deploy Tool**

* Ant 🡪 Maven 🡪 Apache Ivy [ XML Based ]
* Gradle – [ Groovy Script ]

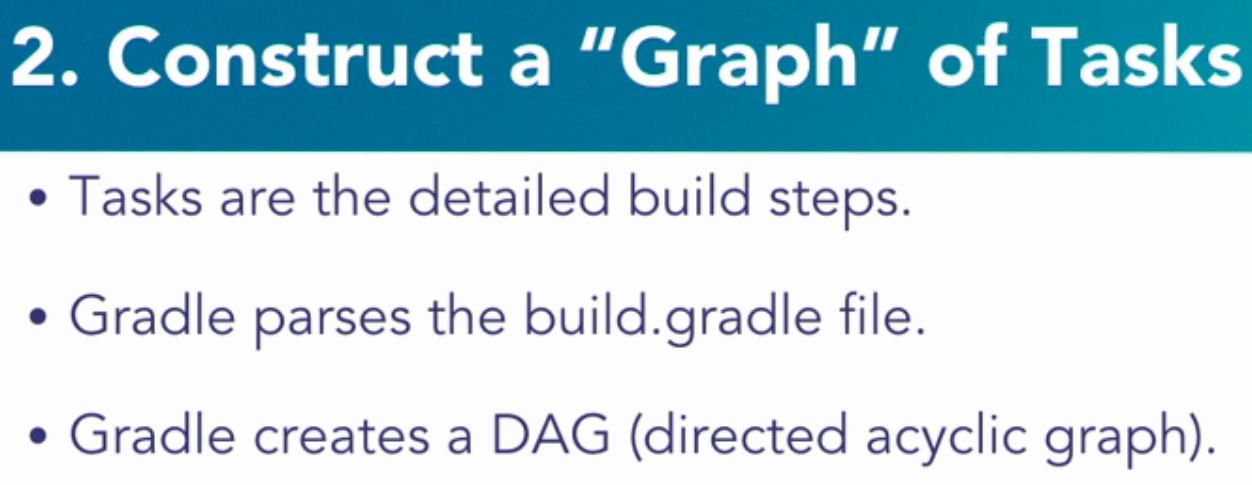
**Gradle alternatives**

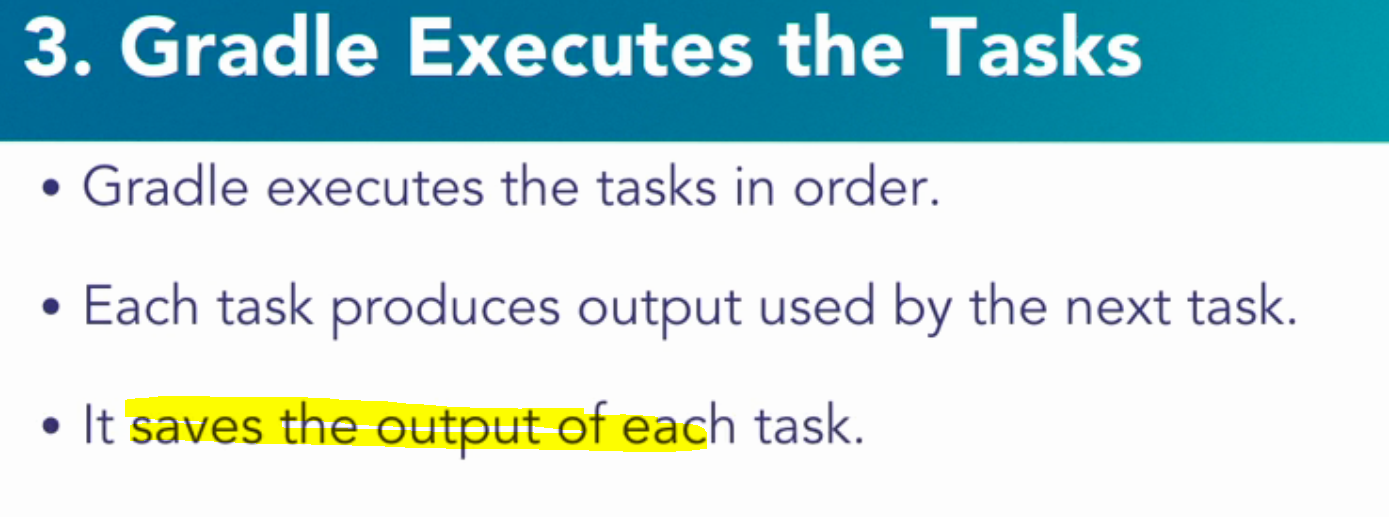
* AWS Code Build
* Maven

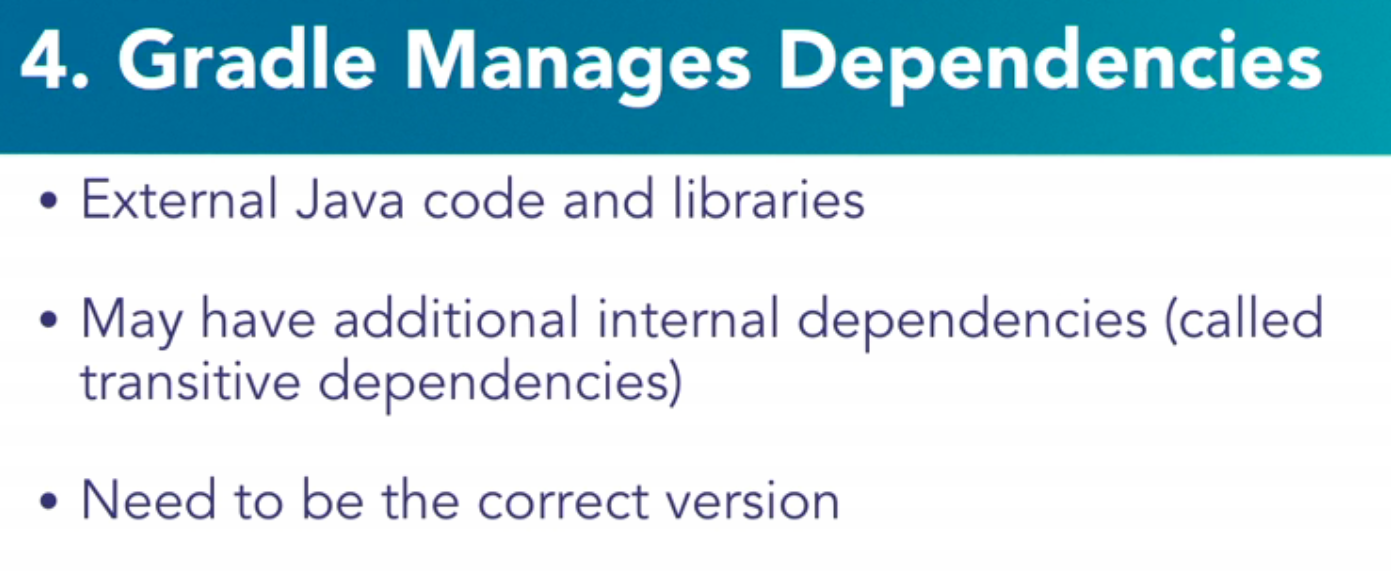
**Build File**

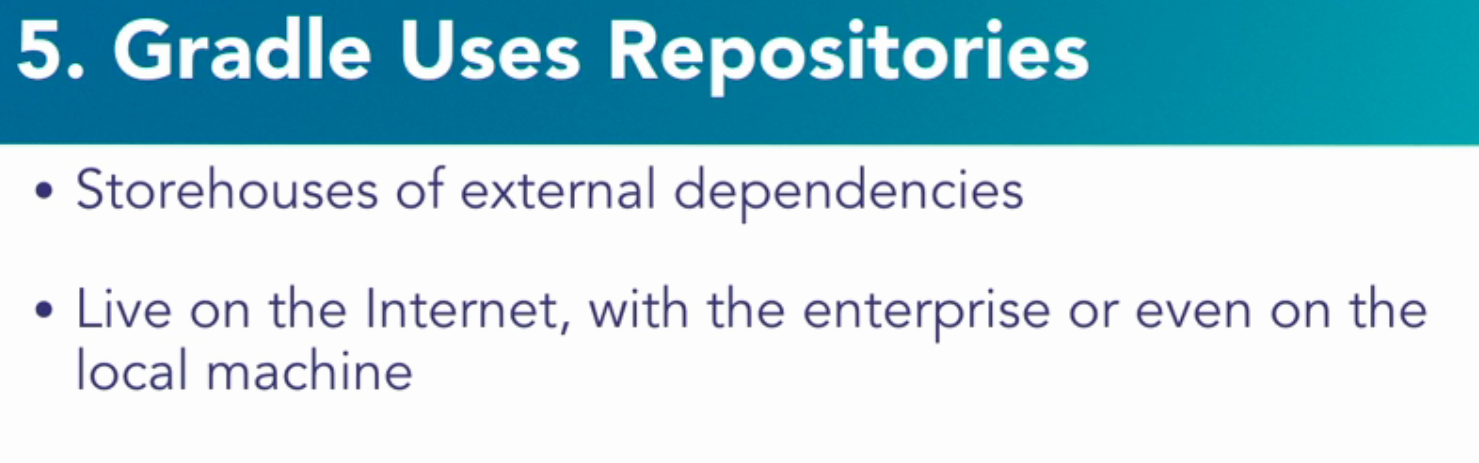
* build.gradle

**Properties**









Without maven etc one has to manually add libraries

**Important things**

Gradle can build different types of source code.

1. If it has to be for Java, then Java plugin needs to be added for Gradle. This statement will ensure gradle is going to working with Java build



1. Different types of task can be performed . Examples
2. Clean – Build clean
3. Compile – Class
4. Assemble – Create Jar
5. Test – Executing tests
6. Adding dependencies

Here the task is to compile, organization path, name of the library and the version

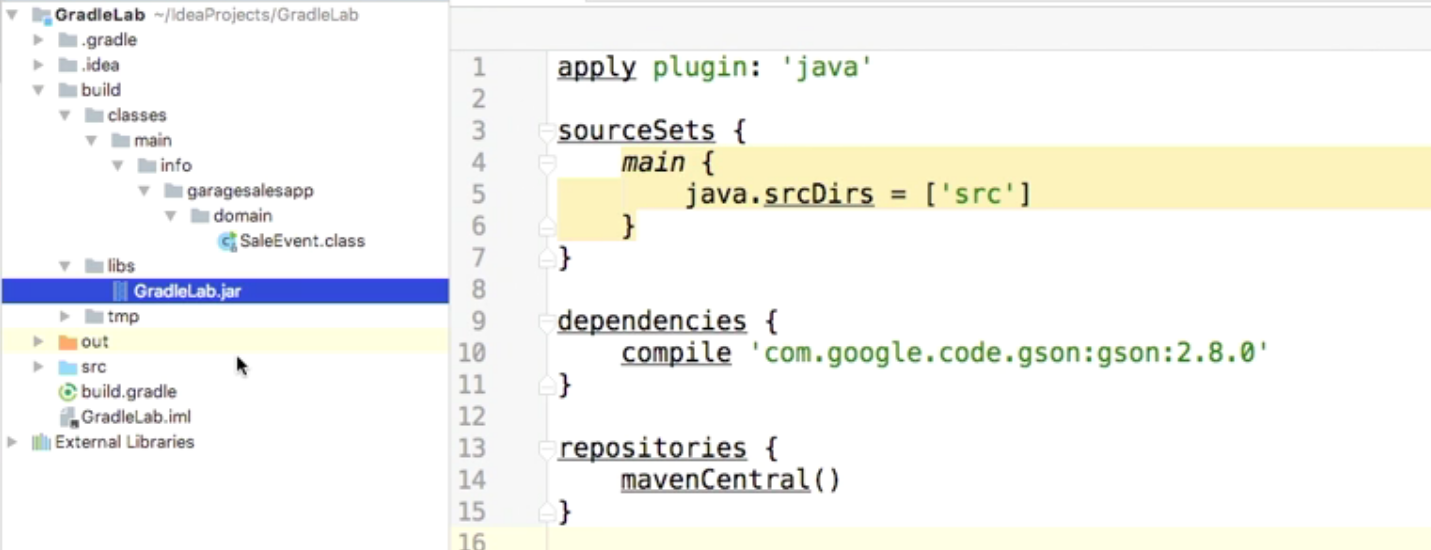


1. Where to look for the dependencies . One such repository is maven central . There can be many like jcentre, maven local etc.



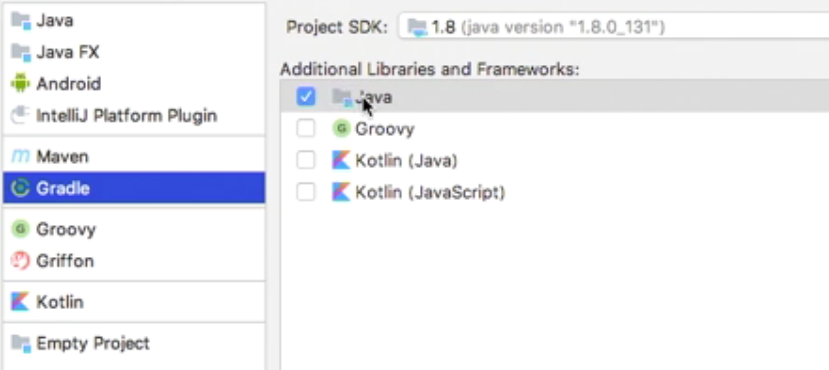
**Steps to use Gradle in Java Project**

1. Configure build.gradle file
   1. Apply plugin
   2. Point the Source code section
   3. Add dependency
   4. Add repository



* Classes are built and present – Look at the hierarchy
* Jar is output in libs folder

**Creating a Gradle Java Project**



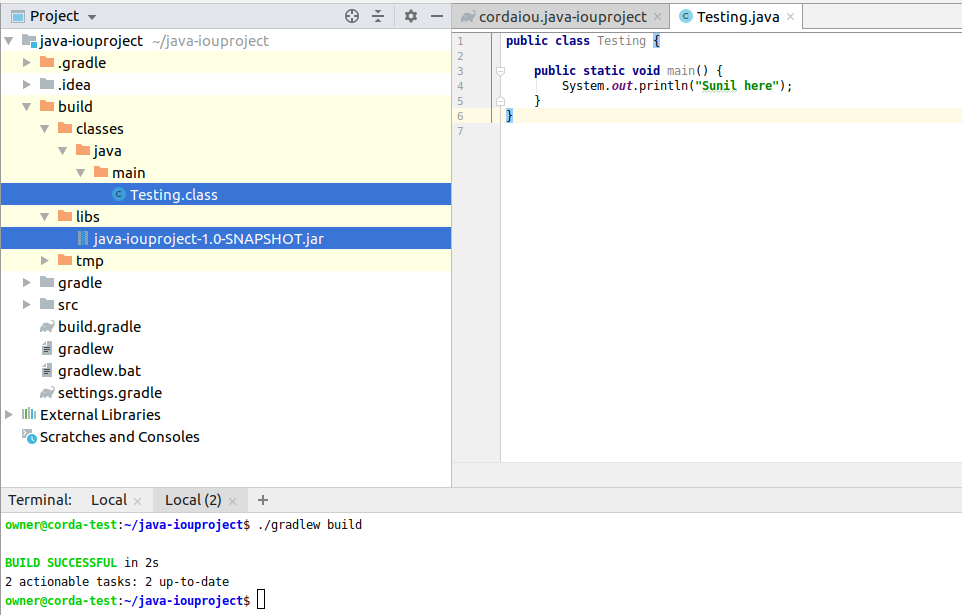
* Empty project creation. Notice build.grade and other files are automatically created and in repositories section -🡪 mavenrepository is referred
* Source code structure – maven style is followed by gradle



* Dependencies refer to junit 🡪 this is because by default unit cases are written using junit framework

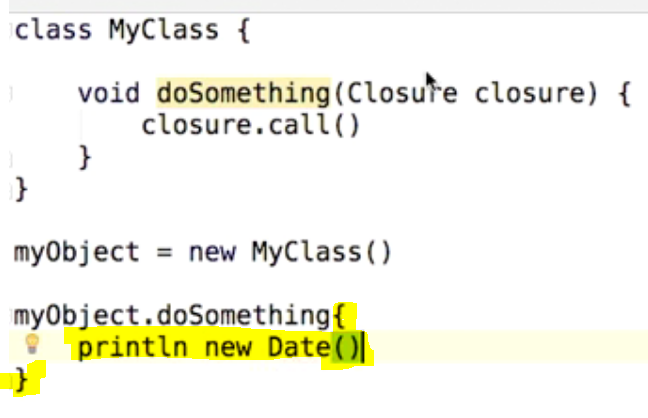
Below is a sample program which is built using gradle. After build process

* Build folder is created



**Simple Groovy commands – just enough for build**

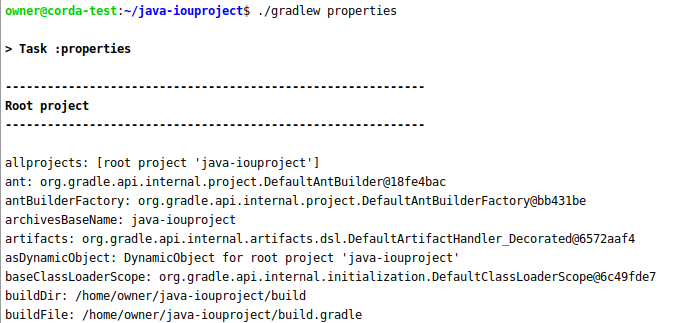
* Passing a function to closure

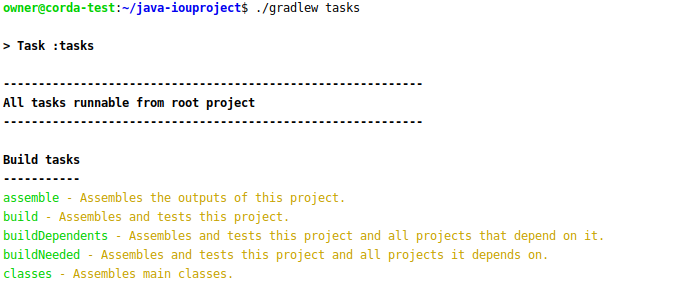


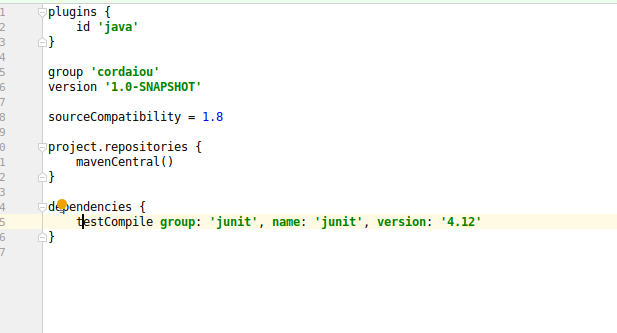
**Gradle Project Model**

First thing, gradle looks while building is “ It will look for project model”

* Projects
* Tasks







Repositories etc are nothing but the properties of project. There are a list of properties and tasks available to explore.

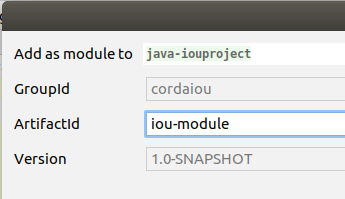
**Find out the Dependencies used in the project**

./gradlew dependencies

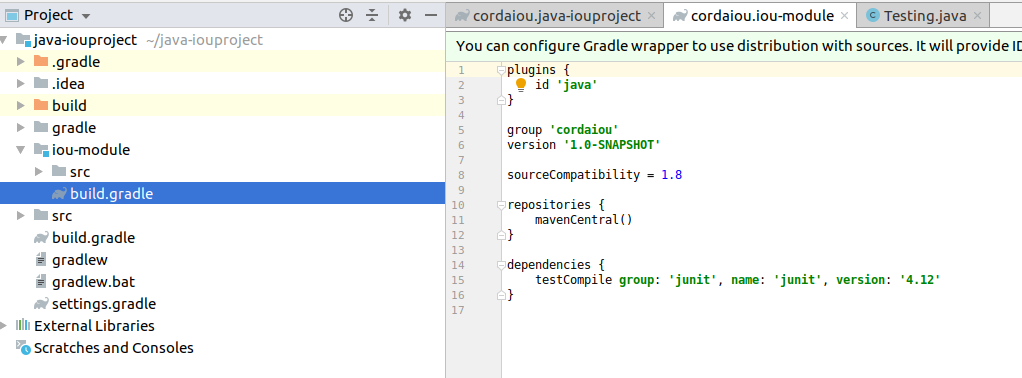
**Modules**

* There may be multiple modules which can be used in a projects folders. These modules can be interlinked in the same project
* Each module will be built separately and JAR is created

**Creating a Java Module**



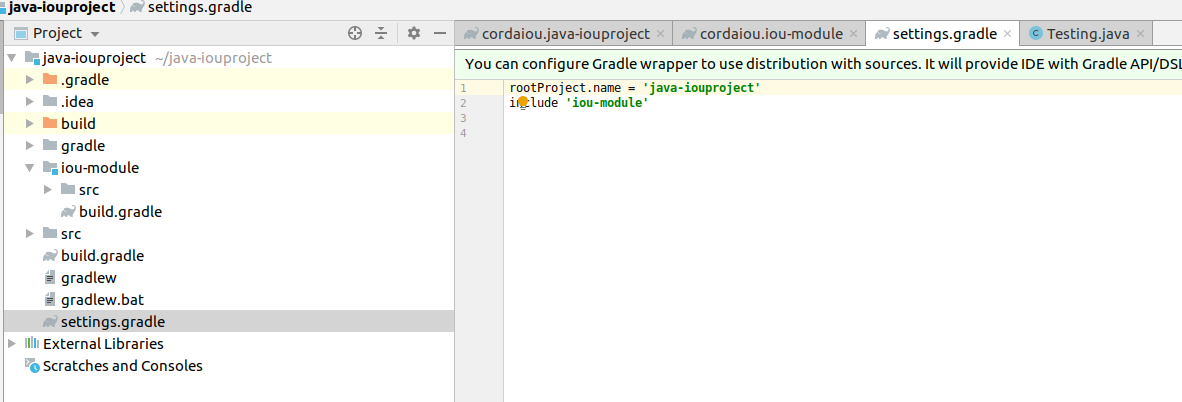
**IOU Module listed below**



Settings.gradle file will provide the details about this workspace

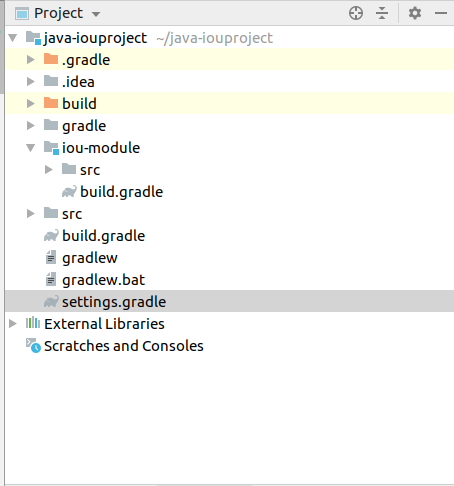
Project Name : java-iouproject

Modules : iou-modules



Each module will also have a build.gradle file

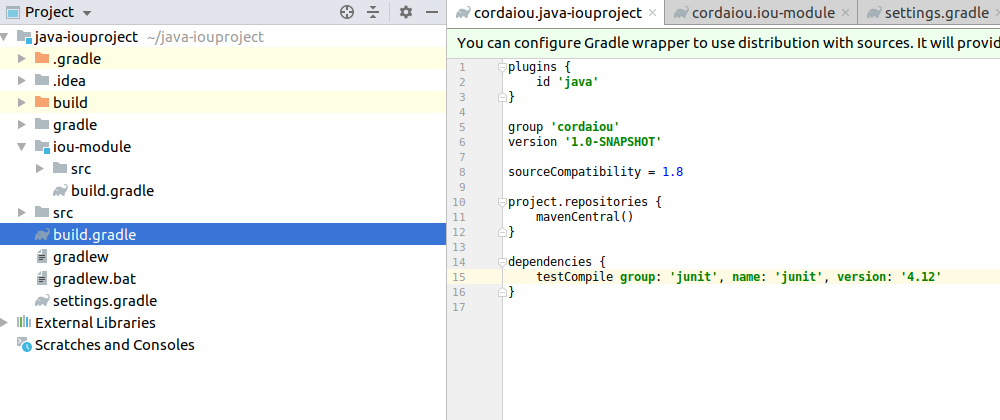
**Gradle Project Structure**



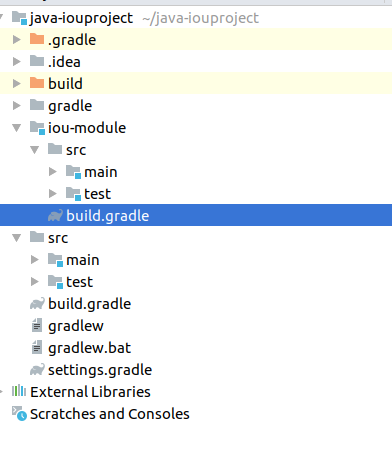
Contents of the settings.gradle 🡪 Mentions the project and the name of the module



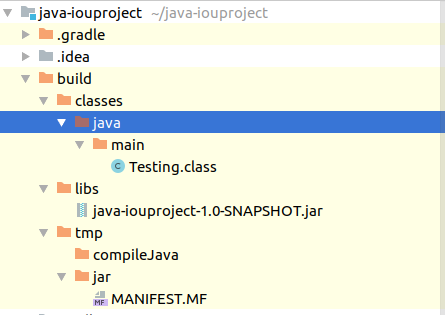
Build.gradle



src directory for both module and the root project



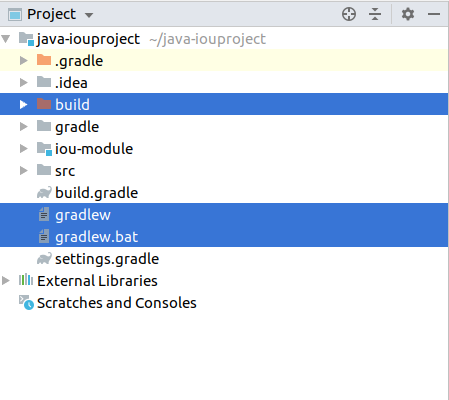
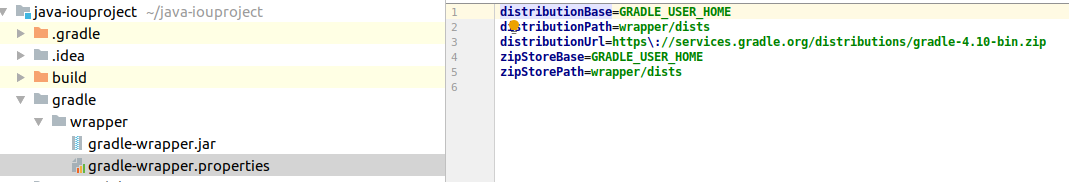
Build directory contains all the build artifacts



./gradlew clean 🡪 cleans the build artificats

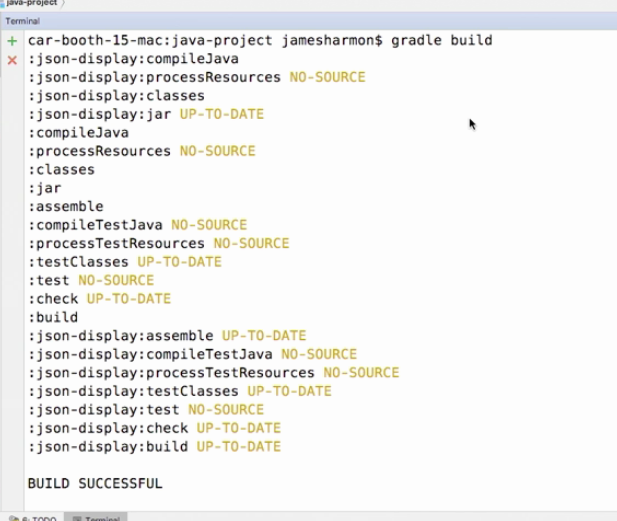
Gradlew , gradlew.bat and gradle folder is present to make sure the right version of gradle is used to build.

From the properties, we know version 4.10 is used. All the team members have to use the same version for build. This is made sure by checking the code in source code repository. Also this is an important information for the Jenkins automated build ( so there is no mismatch of versions)



**Gradle Tasks**

Each one is a task.

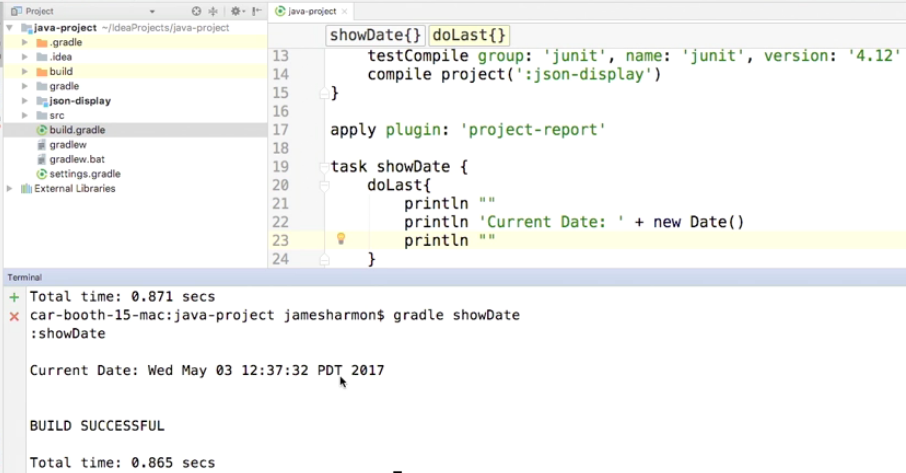


List of tasks available

<https://docs.gradle.org/current/userguide/java_plugin.html#sec:java_tasks>

Creating a new task – Groovy script is used , created and updated in build.gradle

**Simple task – show Date**

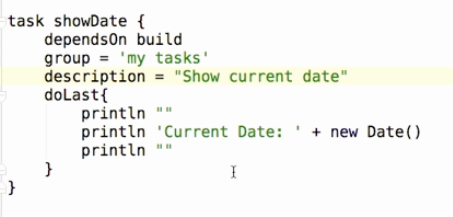


**Simple task – show Date depends on Build**

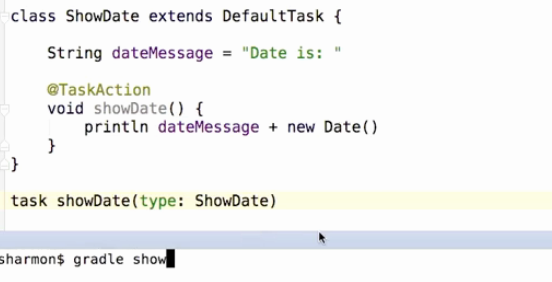
Build task will be completed and then showDate task executes



Task is grouped under “My tasks”



**Java style task creation**



**Create a task as a separate module and building it as Plugin**

