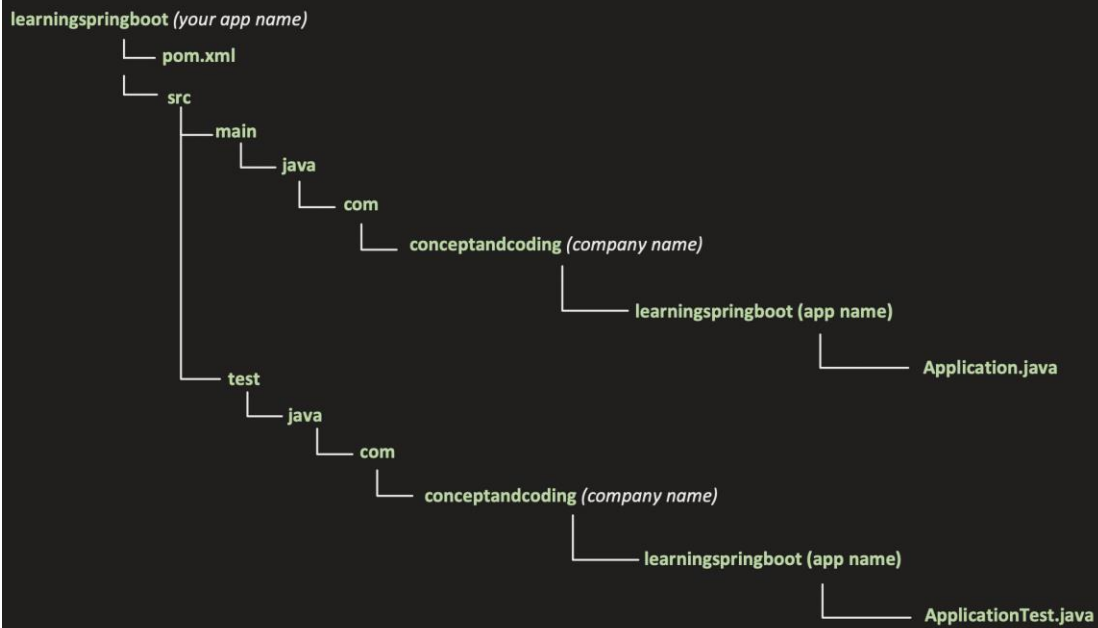


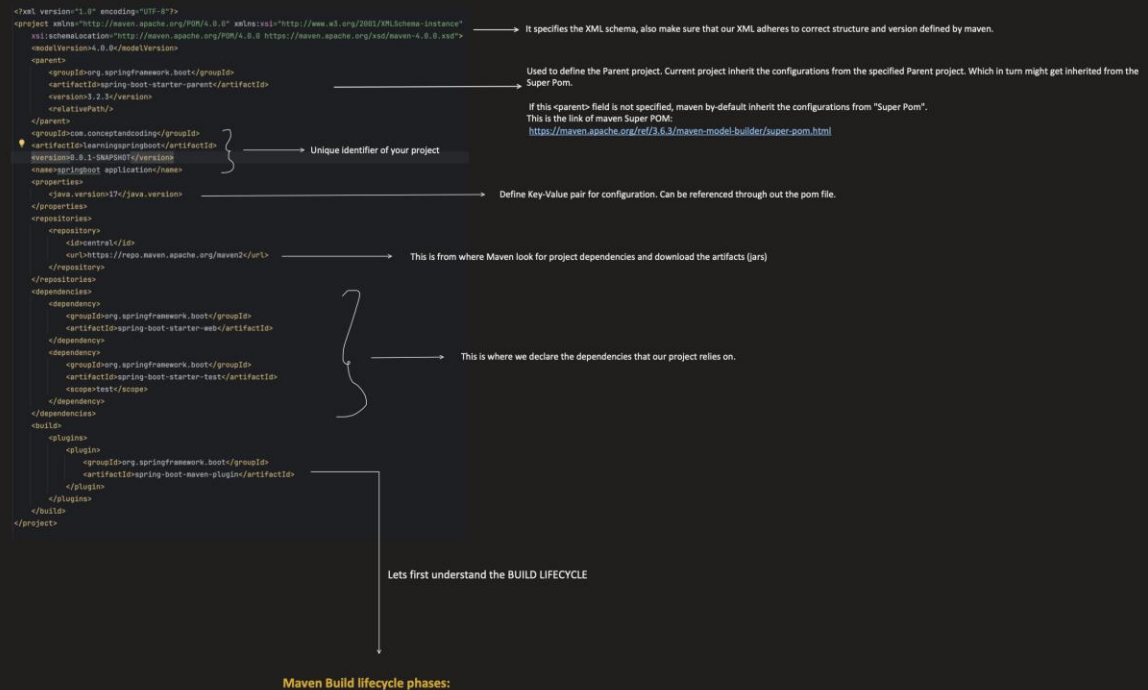
## SpringBoot: Maven

### "Concept & Coding" YT Video Notes

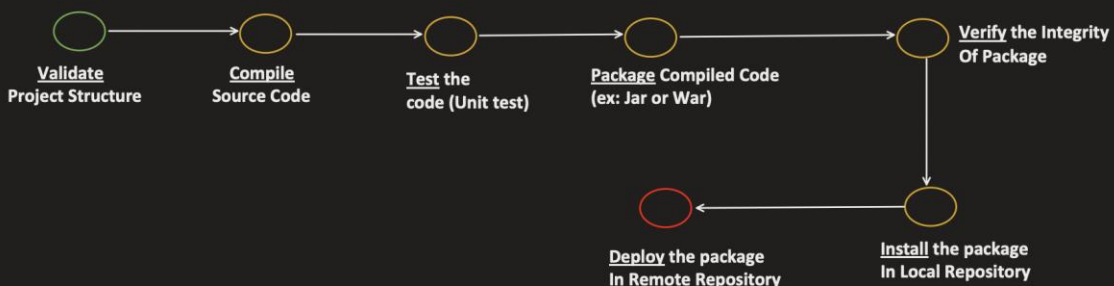
#### What is Maven:

- It's a project management tool. Helps developers with:
  - Build generation
  - Dependency resolution
  - Documentation etc.
- Maven uses POM (Project Object Model) to achieve this.
- When "maven" command is given, it looks for "pom.xml" in the current directory & get needed configuration.





- If you want to run "package" phase, all its previous phase will get executed first.
- And if you want to run specific goal of a particular phase, then all the goals of previous phases + current phase goals before the one you defined will get run.



Maven already has Validate phase defined and its goal, but if we want to add more goals or task, then we can use <build> element. And add the goal to specific phase.

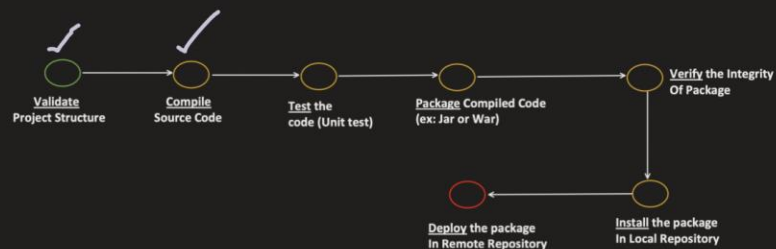
Validate :

**mvn validate**

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-checkstyle-plugin</artifactId>
  <version>3.1.2</version>
  <executions>
    <execution>
      <id>validate-checkstyle</id>
      <phase>validate</phase>
      <goals>
        <goal>check</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <configLocation>myCodeStyle.xml</configLocation>
  </configuration>
</plugin>
```

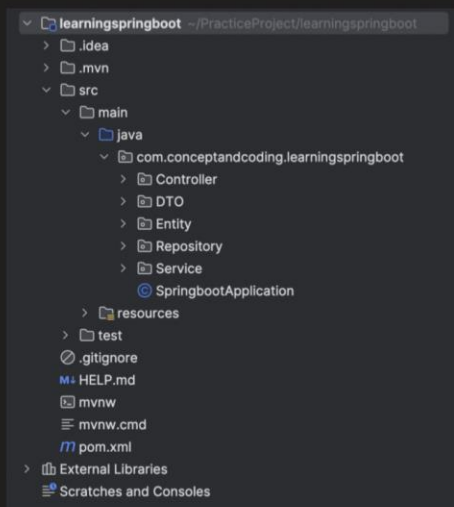
Compile:

Run the command: **mvn compile**

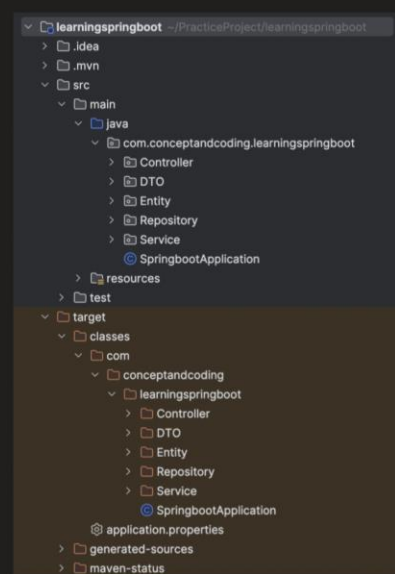


It will validate and compile your code and put it under **`${project.basedir}/target/classes`**

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.conceptandcoding:learningspringboot >-----
[INFO] Building springboot application 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ learningspringboot ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ learningspringboot ---
[INFO] Changes detected - recompiling the module! ;source
[INFO] Compiling 7 source files with javac [debug release 17] to target/classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 0.812 s
[INFO] Finished at: 2024-03-09T16:35:24+05:30
[INFO] -----
```



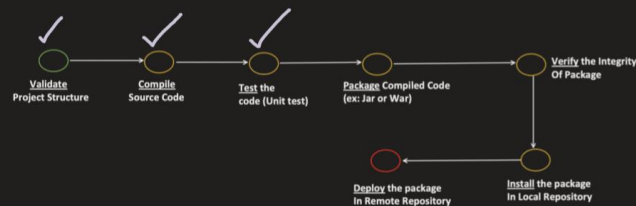
After "mvn compile"



Previously, "**Ant**" was popular, there we have to tell what to do and also how to do.

```
<project default="compile">
<target name="compile">
//some other properties here
<javac destdir="{provide your destination directory}">
<src>
<path element location="src/main/java"/>
</src>
//some other properties here
</javac>
</target>
</project>
```

**Test:** `mvn test`



It will validate, compile and then run the TEST cases in your project.

The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a 'test' directory with a 'SpringbootApplicationTests' class. The code editor shows the following Java code:

```
1 package com.conceptandcoding.learningspringboot;
2
3 > import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
4
5
6 @SpringBootTest
7 class SpringbootApplicationTests {
8
9     @Test
10     void contextLoads() {
11         System.out.println("CONCEPT && CODING : TEST CASE RUNNING");
12     }
13 }
14
15
16
```

```

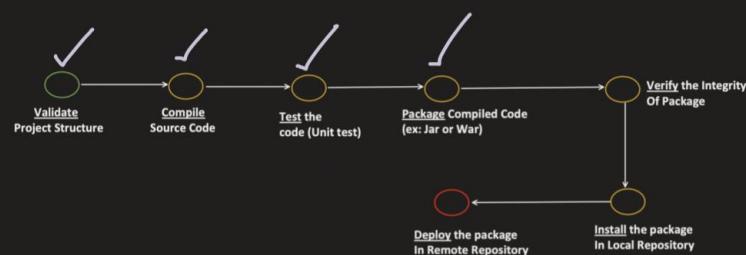
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.conceptandcoding:learningspringboot >-----
[INFO] Building springboot application 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ learningspringboot ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ learningspringboot ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- surefire:3.1.2:test (default-test) @ learningspringboot ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.conceptandcoding.learningspringboot.SpringbootApplicationTests
08:50:22.606 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils --
tApplicationTests does not declare any static, non-private, non-final, nested classes annotated with @Cc
08:50:22.655 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper -- Four
ot.SpringbootApplicationTests

:: Spring Boot ::
(v3.2.3)

2024-03-10T08:50:22.806+05:30 INFO 37695 --- [          main] c.c.l.SpringbootApplicationTests
gboot)
2024-03-10T08:50:22.806+05:30 INFO 37695 --- [          main] c.c.l.SpringbootApplicationTests
2024-03-10T08:50:23.292+05:30 INFO 37695 --- [          main] c.c.l.SpringbootApplicationTests
WARNING: If a serviceability tool is not in use, please run with -Djdk.instrument.traceUsage for more information
WARNING: Dynamic loading of agents will be disallowed by default in a future release
CONCEPT && CODING : TEST CASE RUNNING
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.235 s -- in com.conceptandcoding.learningspringboot.SpringbootApplicationTests
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.824 s
[INFO] Finished at: 2024-03-10T08:50:23+05:30
[INFO] -----

```

**Package:** `mvn package`



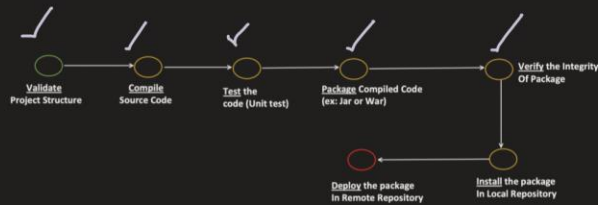
First complete Validate, Compile, Test phase and then run Package phase in which it Generates .jar or .war file.





**Verify:**

`mvn verify`



It can perform some additional checks apart from unit test cases like:

- STATIC CODE ANALYSIS
- CHECKSUM VERIFICATION etc...

#### STATIC CODE ANALYSIS:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-pmd-plugin</artifactId>
  <version>3.21.2</version>
  <executions>
    <execution>
      <id>pmd-analysis</id>
      <phase>verify</phase>
      <goals>
        <goal>pmd</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

PMD is a source code analyzer:

- Finds unused variable
- Finds unused imports
- Empty Catch block
- No usage of object
- Finds duplicate code etc...



```
mvn install
```



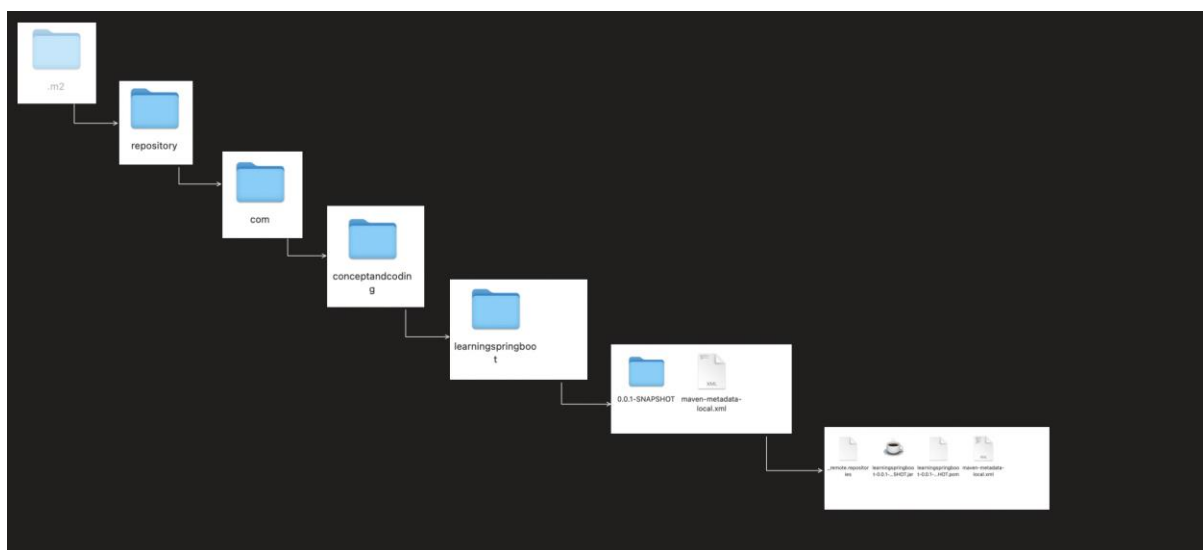
```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
    http://maven.apache.org/xsd/settings-1.0.0.xsd">

  <!-- Local Repository: Location where Maven stores downloaded artifacts -->
  <localRepository>${user.home}/.m2/repository</localRepository>

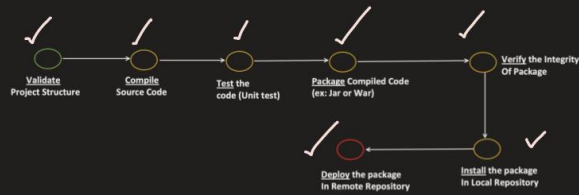
  <!-- some other configurations goes here -->

</settings>
```

[illegible]



**Deploy:** `mvn deploy`



It will deploy the .jar to REMOTE Repository

**pom.xml**

```
<project>
  <!-- ... other project configurations ... -->

  <distributionManagement>
    <repository>
      <id>remote repository id</id>
      <url>https://remote-repository-url</url>
    </repository>
  </distributionManagement>

  <!-- ... other pom.xml configurations ... -->
</project>
```

**settings.xml**

```
<settings>
  <!-- other settings configurations -->

  <servers>
    <server>
      <id>remote repository id</id>
      <username>remote-repository-username</username>
      <password>remote-repository-password</password>
    </server>
  </servers>

  <!-- other settings configurations -->
</settings>
```

If we do not define the remote repository, then MAVEN during "mvn deploy" command we will face below error

```
[INFO] --- deploy:3.1.1:deploy (default-deploy) @ learningspringboot ---
[INFO]
[INFO] BUILD FAILURE
[INFO]
[INFO] Total time: 4:58 s
[INFO] Finished at: 2024-03-10T13:00:11+05:30
[INFO]
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-deploy-plugin:3.1.1:deploy (default-deploy) on project learningspringboot: Deployment failed: repository element was not specified in the POM inside distributionManagement element or in <altDeploymentRepository>id: url parameter -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/RepositoryException
```

If we want to deploy the manifest to Maven central repository : <https://repo.maven.apache.org/maven2>.  
Since its public, we do not need username and password in settings.xml