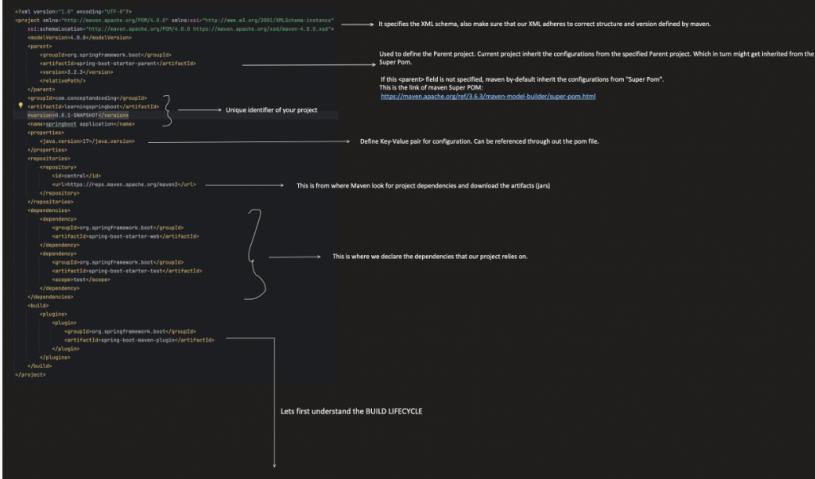
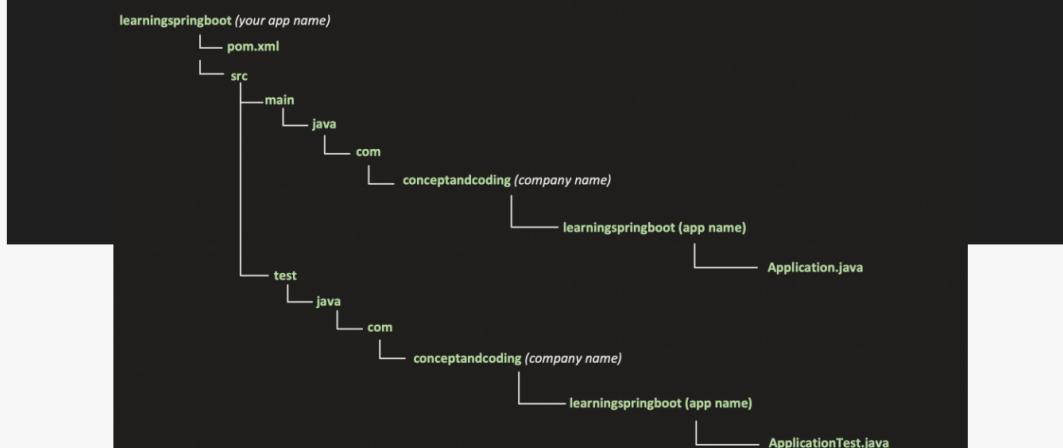


SpringBoot: Maven

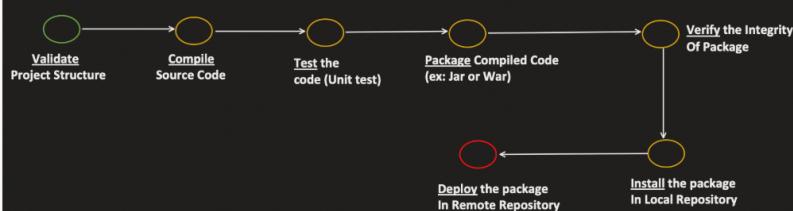
"Concept && Coding" YT Video Notes

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

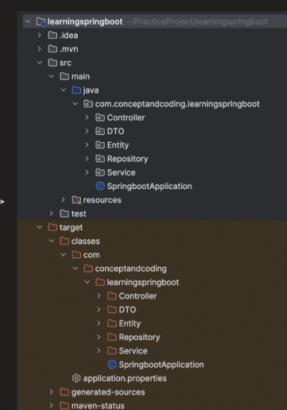
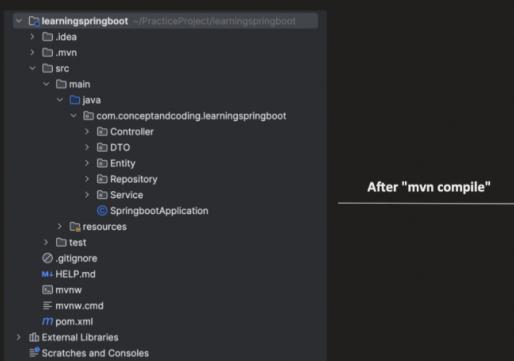
```
<plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-checkstyle-plugin</artifactId>
<version>3.1.2</version>
<executions>
<execution>
<id>validate-checkstyle</id>
<phase>validate</phase>
<goals>
<goal>check</goal>
</goals>
</execution>
</executions>
<configuration>
<configLocation>myCodeStyle.xml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

Compile: Run the command: mvn compile



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

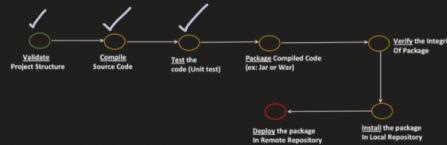
```
[INFO] Scanning for projects...
[INFO]
[INFO] -----> com.conceptandcoding:learningspringboot <-----
[INFO] Building springboot application 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----
[INFO] [ jar ] -----
[INFO] --- resources:3.3.1:resources (default-resources) @ learningspringboot ---
[INFO] 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] -----
```



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>
<pathellement location="src/main/java"/>
</src>
//some other properties here
</javac>
</target>
</project>
```

Test: mvn test



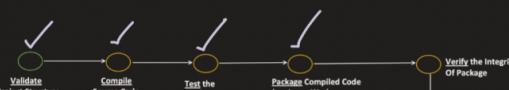
It will validate, compile and then run the TEST cases in your project.

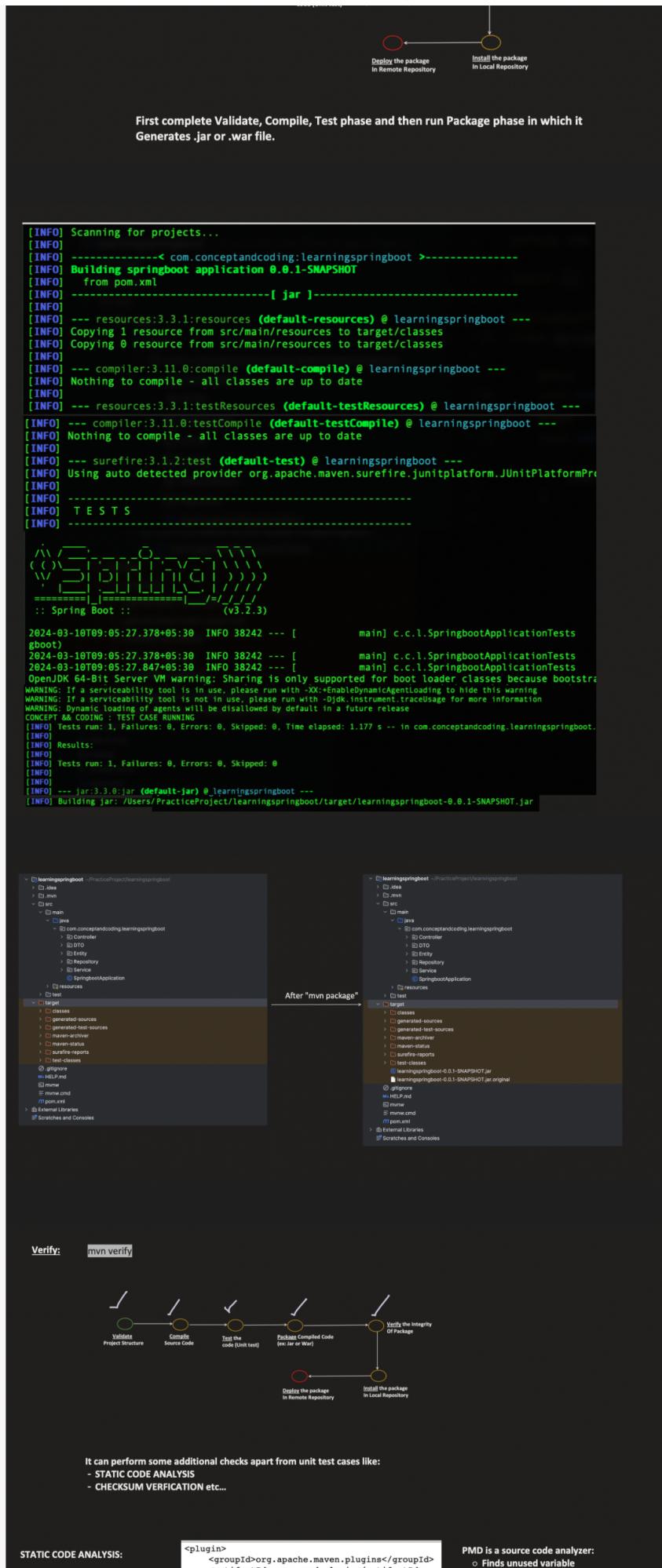
```
learningsspringboot ~/PracticeProject/learningsspringboot
> .idea
> .mvn
> src
  > main
    > java
      > com.conceptandcoding.learningspringboot
        > Controller
        > DTO
        > Entity
        > Repository
        > Service
        > SpringbootApplication
      > resources
    > test
      > java
        > com.conceptandcoding.learningspringboot
          > SpringbootApplicationTests
  > target
```

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

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.conceptandcoding:learningspringboot >-----
[INFO] Building springboot application 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO]
[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] [ E S T S ]
[INFO] -----
[INFO] Running com.conceptandcoding.learningspringboot.SpringbootApplicationTests
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO] Total time:  1.235 s
[INFO] Finished at: 2024-03-10T08:50:22.866+05:30
[INFO] -----
[INFO] Spring Boot :: (v3.2.3)
[INFO] -----
[INFO] 2024-03-10T08:50:22.866+05:30 INFO 37695 --- [           main] c.c.l.SpringbootApplicationTests
[INFO] 2024-03-10T08:50:22.866+05:30 INFO 37695 --- [           main] c.c.l.SpringbootApplicationTests
[INFO] 2024-03-10T08:50:23.292+05:30 INFO 37695 --- [           main] c.c.l.SpringbootApplicationTests
[INFO]
[INFO] WARNING: If a serviceability tool is not in use, please run with -Djde.instrument.traceUsage for more information
[INFO] WARNING: Dynamic loading of agents will be disabled by default in a future release
[INFO] CONCEPT & CODING : TEST CASE RUNNING
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO] Results:
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[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





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

- STATIC CODE ANALYSIS
- CHECKSUM VERIFICATION etc....

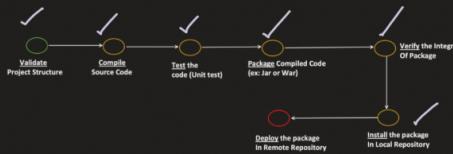
STATIC CODE ANALYSIS:

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-pmd-plugin</artifactId>
```

- o PMD is a source code analyzer

```
<version>3.1.1</version>
<executions>
    <execution>
        <id>pmd-analysis</id>
        <phase>verify</phase>
        <goals>
            <goal>pmd</goal>
        </goals>
    </execution>
</executions>
</plugin>
```

Install: mvn install



It will install the .jar package in local Maven Repository, which is typically located in your user home directory (~/.m2/repository).

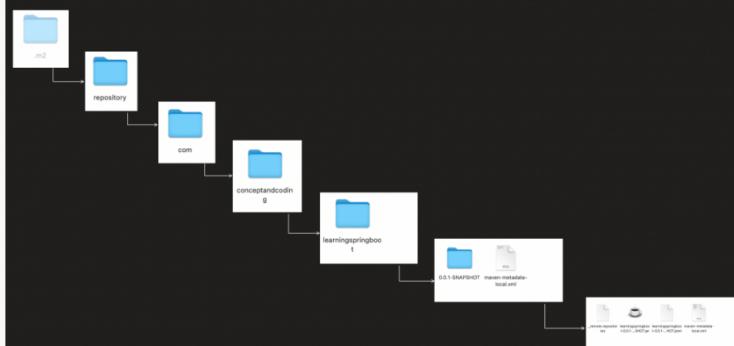
settings.xml (in .m2 folder)

```
<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>
```



Deploy: mvn deploy

