Maven

To execute the scripts using Terminal

1. Add sure fire plugin in pom.xml and give the testng.xml file path in it

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.1.2</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>testng.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

1. How to handle multiple testng.xml files in pom.xml

Using profiles tag. Below example used 3 xml files

<profiles>

<profile>

**<id> Regression</id>**

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.1.2</version>

<configuration>

<suiteXmlFiles>

**<suiteXmlFile>testng.xml</suiteXmlFile>**

</suiteXmlFiles>

</configuration>

</plugin>

<plugins>

</pluginManagement>

</build>

</profile>

<profile>

**<id> Order Delivery</id>**

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.1.2</version>

<configuration>

<suiteXmlFiles>

**<suiteXmlFile>orderDelivery.xml</suiteXmlFile>**

</suiteXmlFiles>

</configuration>

</plugin>

<plugins>

</pluginManagement>

</build>

</profile>

<profile>

**<id> Error Validation</id>**

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.1.2</version>

<configuration>

<suiteXmlFiles>

**<suiteXmlFile>purchase.xml</suiteXmlFile>**

</suiteXmlFiles>

</configuration>

</plugin>

<plugins>

</pluginManagement>

</build>

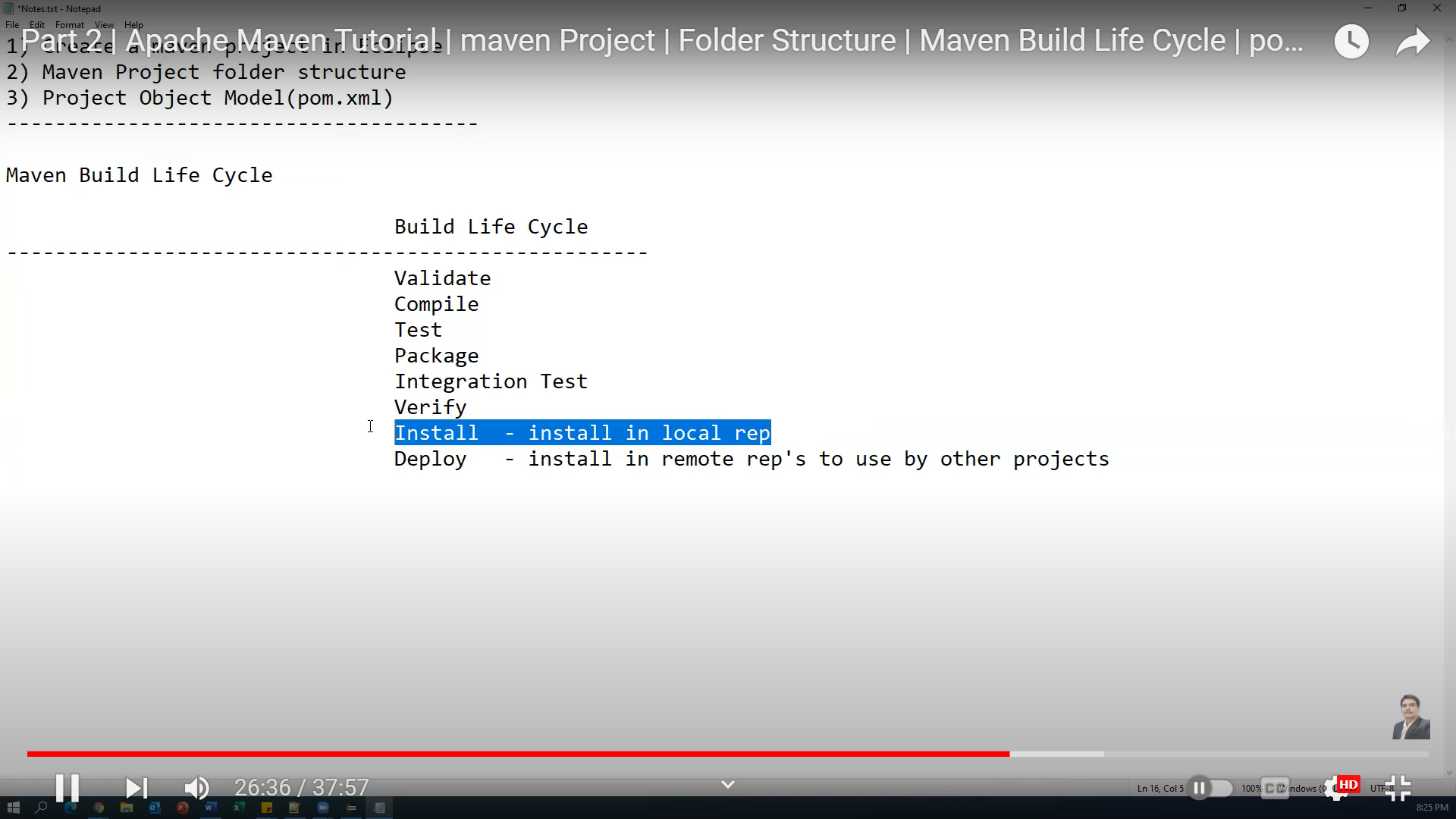
</profile>

While running in the terminal run as below-> goto the path where pom.xml is present

mvn test -PRegression

p stands for profile

1. Maven build life cycle



1. Maven clean command : Will clean the unnecessary files(files in target folder)
2. Maven install: will perform all the commands above it as shown in picture

It will run the code and generate jar file in target folder. If you want to share source code to others.. in that case you can share the jar created

1. \*\*Maven Install\*\*:

- Think of this as a "build and package" command.

- When you run `maven install`, it compiles your code, runs tests (if any), and then packages your project into a distributable format (like a JAR or WAR file).

- It's like creating a final, complete version of your project that can be shared or deployed.

- It ensures that your code not only compiles but also can be used by other parts of your project or other projects.

\*\*Maven Test\*\*:

- This is all about testing your code.

- When you run `maven test`, it compiles your code, runs all the tests you've written, and reports the test results.

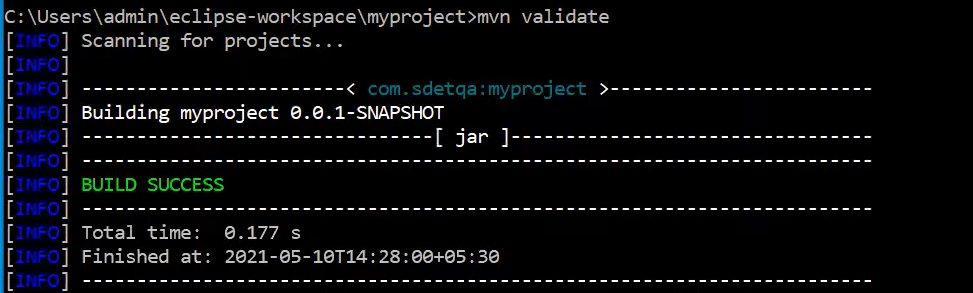
- It's like a quality check to make sure your code is working as expected.

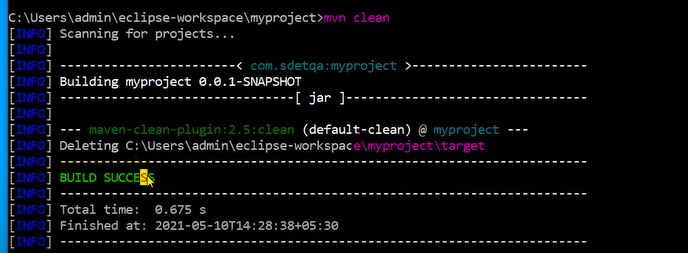
- It doesn't package your code for distribution; it's just making sure your code does what you think it should.

So, in simple terms, `maven install` prepares your code for distribution, while `maven test` checks if your code works correctly by running tests. They serve different purposes in the development and build process.

Open terminal and go to project location

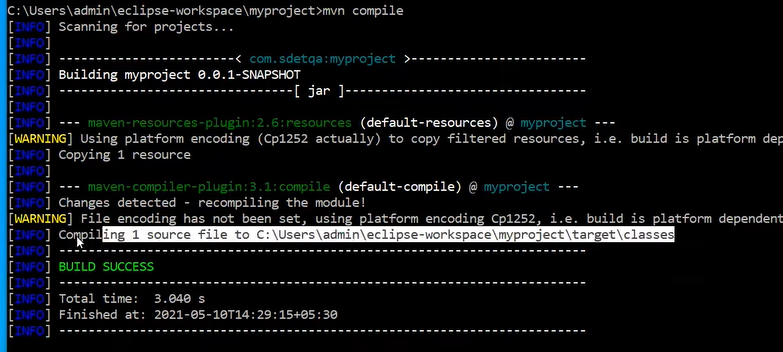
mvn validate – checks if all the files required for project are present or not





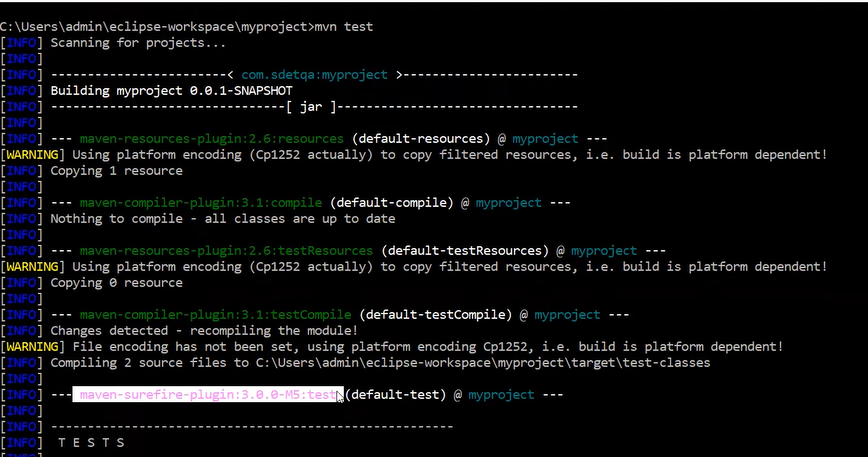
1. Mvn compile

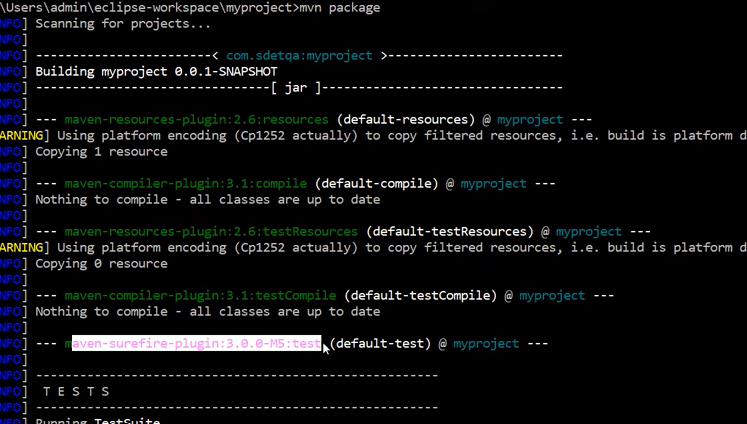
Will compile the source code

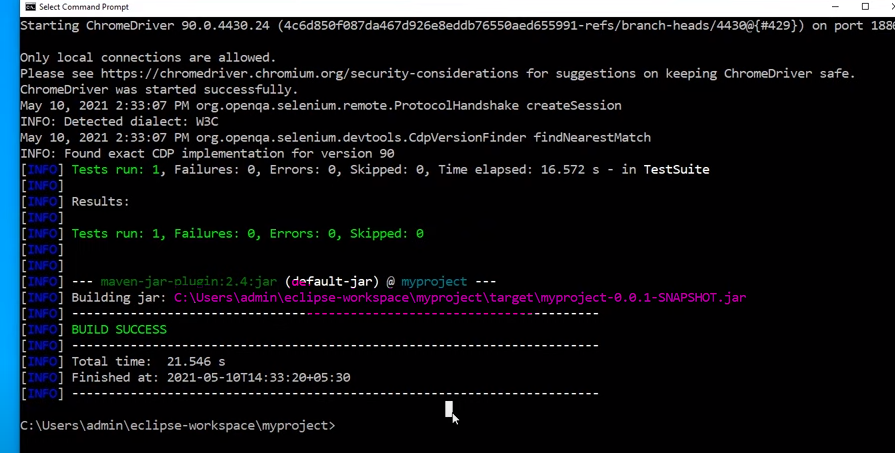


1. Mvn test

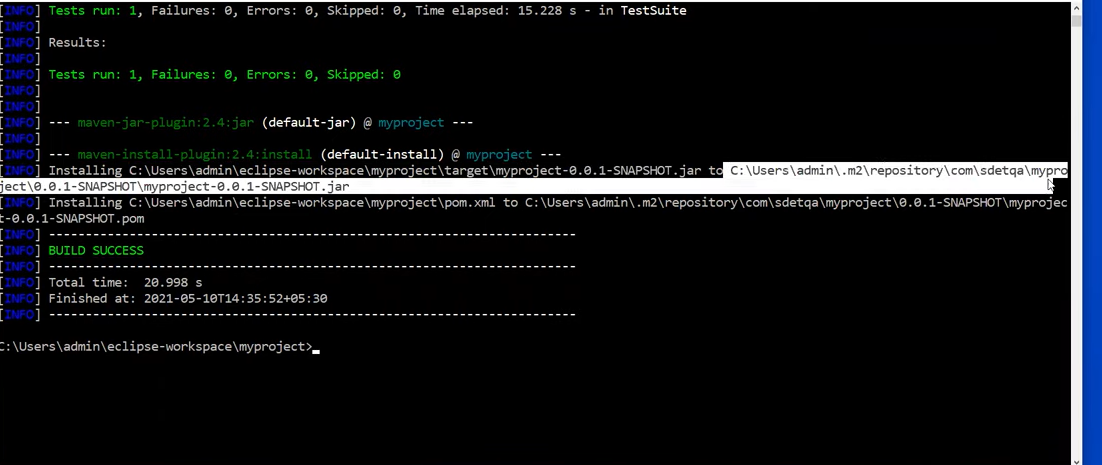
Will run the test cases



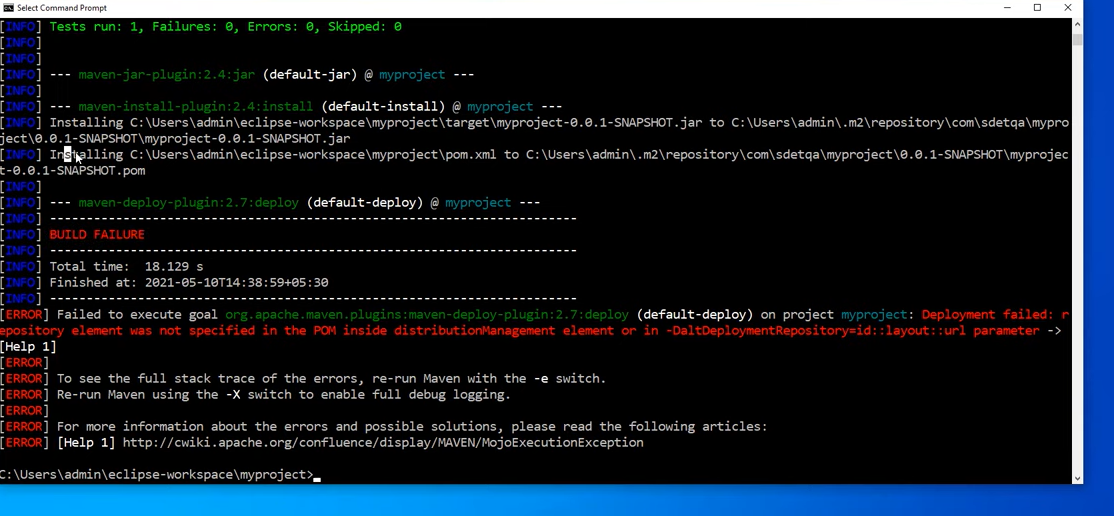
1. 



1. Mvn install ->will copy jar created above to local repo



1. Mvn deploy – will deploy the jar copied to Local to external repo like Jenkins, git etc



Error highlighted in red as we just gave mvn deploy. We need to specify the configurations where you want to deploy the jar to external sites.

1. Although each and every command in Maven lifecycle executes its prior commands as well We can give multiple commands together

Ex: mvn clean test