**Apache Maven | Duration: 1 Day**

## **Installing the Maven**

1. Download the latest version of Maven and Install by unzipping the war file
2. Create an environment variable MAVEN\_HOME
3. Add the path for Maven. Path is always set to bin
4. Check the maven is properly configured using the command mvn –version

## **Installing the M2Eclipse Plugin**

1. Select **Help | Install New Software** from Eclipse Menu
2. Add a new repository named M2Eclipse for the URL

URL:<http://download.eclipse.org/technology/m2e/releases>

1. Select the plugins and Install. Apache maven is integrated to the IDE
2. Check if maven has been installed successfully
3. Click Help-> About Eclipse and then check the presence of M2Eclipse icon
4. Get the details of the Maven installation by clicking on the M2Eclipse icon

## **Creating the Maven Project**

1. Create a new Project from File-> New->Other, then Select Maven -> Maven Project
2. Select the **Create a simple project while creating the Maven project**

**Group Id: com.jala.maven**

**Artifact id: MavenTest**

1. **Check the project structure**
2. **By default, the JRE System library is pointing to J2SE1.5. Change it default JRE on your system (latest JRE on your system)**
3. Select pom.xml from Project explorer
4. Add dependencies for TestNG, maven, Selenium and log4J with in the project node

Maven dependencies :-

maven-compiler-plugin

maven-surefire-plugin

1. Know the path where the dependency jars are downloaded to from central repository of maven(.m2 folder in users->username)
2. Create a TestNG class in src/test/java by right clicking on the folder with all the **annotations selected** and name it FirstTest

Add print statements for every annotation to check the order of the annotations execution

Ex: System.out.println("Before Suite");

System.out.println("After Suite");

1. Add a first test case with the annotation @Test and name it firstTestCase
2. Run the TestNG test case and observe the order of the annotations execution
3. Convert the Maven project to TestNG Project by right clicking on the project and then click TestNG->Convert to TestNG

testng.xml file is generated in the project folder

1. Running test from testing.xml

Right click on testing.xml, Run As -> Run configurations

Type testing.xml for Suite, So it runs testing.xml file when the test suite is run

Then continue running the test suite from testing.xml by right clicking on xml file

## **Running the test as Maven Test**

* 1. Add dependencies for

maven-compiler-plugin

maven-surefire-plugin

* 1. Run test as maven test

Right Click on the project and Run As->Maven test

(Run Maven clean, Maven Build and then Maven test if script fails)

## **Default Life Cycle**

The default life cycle is the build life cycle which generates, compiles, packages etc. your source code.

You cannot execute the default build life cycle directly, as is possible with the clean and site. Instead you have to execute a specific build phase within the default build life cycle.

The most commonly used build phases in the default build life cycle are:

|  |  |
| --- | --- |
| Build Phase | Description |
| validate | Validates that the project is correct and all necessary information is available. This also makes sure the dependencies are downloaded. |
| compile | Compiles the source code of the project. |
| test | Runs the tests against the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed. |
| package | Packs the compiled code in its distributable format, such as a JAR. |
| install | Install the package into the local repository, for use as a dependency in other projects locally. |
| deploy | Copies the final package to the remote repository for sharing with other developers and projects. |