**Maven**

Maven is a dependency management tool

In a project v use soo many different JAR Files and PULGINs

These will be getting updated nd new versions will b generated soo thn every time u need to uninstall/upgrade the older version to NEW VERSION

Soo to make that updation automatically v use MAVEN

Every JAR Files have dependencies TAG and PULGINs will have built TAG if u specify thm in POM.xml file thn maven only automatically update thm when new version is released

MAVEN PROJECT CREATION

Eclipse-file-new-project-maven-maven project-next-create a sample project-next-group id-artifact id(project name)-finish

Project-(src/main/java)Package, pom.xml

POM.XML

Project object module it will cm with maven by default all the project configurations with maven are specified here

Pom.xml(here v need to specify the dependency TAGs)

Selenium dependency maven

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<dependencies> // v need add dis

<dependency> //copy paste this frm google after VERSION TAG

<groupId>org.seleniumhq.selenium</groupId> <artifactId>selenium-java</artifactId>

<version>3.4.0</version>

</dependency>

</dependencies>

<groupId>123</groupId>

<artifactId>MavenProject</artifactId>

<version>0.0.1-SNAPSHOT</version>

</project>

When v add dis <dependencies> a new dependencies folder will be created automatically with all selenium jar files

Where all these jar files will b downloaded in our system (whn v download eclipse a default maven repository will b created)

C:\Users\harshitha\.m2\repository <groupId>org.seleniumhq.selenium</groupId>

//in repository folder names(C:\Users\harshitha\.m2\repository-org-seleniumhq-selenium)

If the dependencies are removed or deleted frm the repository also np it will automatically download nd use in the program

If new version is released thn u should just change the version name in POM.xml file that’s it

SAME WITH other things also

TestNG dependency maven

|  |
| --- |
| <dependencies>  <dependency> |

|  |
| --- |
| <groupId>org.testng</groupId> |

|  |
| --- |
| <artifactId>testng</artifactId> |

|  |
| --- |
| <version>6.8</version> |

|  |
| --- |
| <scope>test</scope> |

|  |
| --- |
| </dependency> |

</dependencies>

Jxl(excel) dependency maven

<dependencies>

<!-- https://mvnrepository.com/artifact/net.sourceforge.jexcelapi/jxl -->

<dependency>

<groupId>net.sourceforge.jexcelapi</groupId>

<artifactId>jxl</artifactId>

<version>2.6</version>

</dependency>

</dependencies>

**RUNNING TESTCASES IN MAVEN(Eclipse commandpromt )**

**In Eclipse--- run pom.xml**

**POM.xml by default it will b in java1.5 so 1.5 will not support every thing so v need to use**

**MAVEN COMPILER PULGIN JAVA // copy nd paste that in pom.xml**

1. <build>
2. <plugins>
3. <plugin>
4. <groupId>org.apache.maven.plugins</groupId>
5. <artifactId>maven-compiler-plugin</artifactId>
6. <version>3.6.1</version>
7. <configuration>
8. <source>1.8</source>
9. <target>1.8</target>
10. </configuration>
11. </plugin>
12. </plugins>
13. </build>

**To run the TestNG.XML file in maven use**

**MAVEN SUREFIRE PLUGIN FOR TestNG**

<plugin>

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

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

<version>2.20</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>testng.xml</suiteXmlFile> //name of the xml.file in the project

</suiteXmlFiles>

</configuration>

</plugin>

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>123</groupId>

<artifactId>MavenProject</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

--------------- //all <dependency>

-----------

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

--------- //al <plugin>

----------

</plugin>

</plugins>

<build>

**EXECUTION maven commands**

**mvn clean--- it will remove if there r any unwanted file in the project/program**

**mvn validate--- will check for the syntax errors**

**mvn compile (validate + compile)---compile the project**

**mvn test (validate + compile + test)--- test/run the testcases**

**POM.Xml—runas—maven clean(**BUILD SUCCESS)

**----maven test( )**

**REPORTS(project-properties-copy-run-D:\eclipse workspace\MavenProject\target\surefire-reports)**

**Run from COMMAND PROMT**

**Download maven software from google**

<https://maven.apache.org/download.cgi>

|  |  |
| --- | --- |
| Source zip archive  Set system properties  MAVEN\_HOME JAVA\_HOME PATH  MAVEN\_HOME---- D:\drivers\apache-maven-3.5.0 //// whr u place the maven in ur pc  Control Panel\System and Security\System---advance system setting-environmental variables-new  Variable name----- MAVEN\_HOME variavle value---- D:\drivers\apache-maven-3.5.0  New-- Variable name----- JAVA\_HOME variavle value---- C:\Program Files\Java\jdk1.8.0\_71  PATH---edit—new-D:\drivers\apache-maven-3.5.0\bin  Command prompt—cd (project path)  >mvn clean(built success)  >mvn compile  >mvn test  **BASH FILES.bat(text files)**  **Here v write execution commands(command prompt) soo without opening the program in eclipse**  **also v can run by dis** | [apache-maven-3.5.0-src.zip](http://www-us.apache.org/dist/maven/maven-3/3.5.0/source/apache-maven-3.5.0-src.zip) |