MAVEN CONFIGURATION

===================

Download Maven

Open URL: https://maven.apache.org/download.cgi

Click on "apache-maven-3.6.3-bin" under "File" section

Extract zip file

Windows logo + Pausebreak ---> Click on Advanced system settings

Set System Variable Home & Path

MAVEN\_HOME

C:\Users\Ehtesham\Downloads\apache-maven-3.6.3

PATH

C:\Users\Ehtesham\Downloads\apache-maven-3.6.3\bin

Download Maven

Open URL: https://maven.apache.org/download.cgi

Click on "apache-maven-3.6.3-bin" under "File" section

Extract zip file

--------------------------------------------------------

Step 2: Specify maven path to System Environment variable

MAVEN\_HOME: C:\Users\Ehtesham\apache-maven-3.6.3

path: C:\Users\Ehtesham\apache-maven-3.6.3\bin

--------------------------------------------------------

Note: To check maven version

Go to command prompt

Type: mvn -version

Click on enter

----------------------------------------------------------

WORKING WITH MAVEN IN SELENIUM

------------------------------

Step 1: Create Maven project in Eclipse

Navigation

----------

File menu in Eclipse

New

Select "Others"

Maven

Select "Maven Project"

Select "create simple project"

Enter group id (package name)

Enter Artifact id (project name)

Click on "Finish"

=====================================================================================================================================================================

Note: MAVEN PROJECT STRUCTURE

=======================

--> src/test/java

In this folder we can

-- create packages

-- create automation programs for testing

---> src/main/resources

In this folder we maintain

-- create testng.xml file

-- Testdata excel files

-- properties files.....etc

=====================================================================================================================================================================

Step 2: Work with POM .xml file (Project Object Model)

We can add all the dependency and Plugin details in pom .xml

File dependencies: used to download jars like Webdriver jars, TestNg jars, apache poi,....etc

plugins: used to run your Maven project

maven - compiler - plugin : To compile java programs

maven - surefire - plugin : To execute testNg .xml files // To run TestNG xml files

=====================================================================================================================================================================

Navigation

----------

Go to mvn repository website

Search for Selenium jars, POI jars, testng jars, extent reports jars

Those are copy and paste in the pom .xml file

or you can Copy and paste the following code in pom .xml file

=====================================================================================================================================================================

Note: First time when we run Maven project we will be getting error

In order to overcome the error we need to change from jre compiler to jdk compiler files

As Maven needs jdk compiler

=====================================================================================================================================================================

POM FILE SETTINGS

=================

<properties>

<suiteXmlFile>src/main/resources/testng.xml</suiteXmlFile>

</properties>

<dependencies>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.9.9</version>

</dependency>

<!-- Adding Selenium dependency -->

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.141.59</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>maven-compiler-plugin</artifactId>

<version>3.0</version>

<configuration>

<compilerVersion>1.8</compilerVersion>

<source>1.6</source>

<target>1.6</target>

</configuration>

</plugin>

<plugin>

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

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

<version>3.0.0-M4</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>${suiteXmlFile}</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

</plugins>

</build>