**-----------------------------------------------------------Maven Setup ---------------------------------------------------------**

**Required\***

(Download Maven Apache)

1. Create MAVEN\_Home by clicking New -- Paste the Maven Path

Example - (C:\Users\NEW-PC\OneDrive\Documents\apache-maven-3.8.1-bin\apache-maven-3.8.1)

1. Under User variable -- Path -- Paste the Maven Path Till Bin Example - (C:\Users\NEW-PC\OneDrive\Documents\apache-maven-3.8.1-bin\apache-maven-3.8.1\bin)

To check installed or not -- Go to CMD and hit **"mvn -version"**

1. Correct your pom.xml(Maven Project should have been created)

2. Refer this video (https://www.youtube.com/watch?v=nasdb9QhFqg) **Mukesh Otwani**

(https://www.youtube.com/watch?v=lPxrPqSyCo8&list=PLFGoYjJG\_fqq6cHeqfsDes3pdVh3kpl74&index=6) **Naveen AutomationLabs**

3. Or Refer this site **(https://learn-automation.com/maven-integration-with-selenium/)**

**(https://learn-automation.com/selenium-maven-build-and-usage/)**

4.Add this under pom.xml along with required Dependencies.

<properties>

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

</properties>

<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>2.17</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>${suiteXmlFile}</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

</plugins>

</build>

5. Now go to Eclipse -- Right Click Project -- Copy Project Path Example - (C:\Users\NEW-PC\eclipse-workspace\VzAutomationTest)

6. Run -- Paste the Path -- Check if the location is correct.

7. Open CMD -- Paste the Path -- cd C:\Users\NEW-PC\eclipse-workspace\VzAutomationTest

8. Enter **"mvn clean" on CMD**

9. **BUILD SUCCESS** message will be shown.

10. Now Enter **"mvn install"on CMD** -- If issue resolved automatically cases will triggered ..

11. Enter **"mvn test" on CMD** to trigger from CMD :)

12. Completed

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

**-----------------------------Jenkins Setup and Integration of Maven Project in Jenkins ---------------------------**

**Required\***

1. Download Java 8 or 11

2. Set Environment Variable **Ex: C:\Program Files\Java\jdk-11.0.11\bin**

3. Windows -- Settings -- System Settings --Search Env variables -- Edit Environment Variable

-- Under System variable -- Path -- Paste your Java Location Till Bin (C:\Program Files\Java\jdk-11.0.11\bin)

-- Create JAVA\_HOME by clicking New -- Paste your Java Location (C:\Program Files\Java\jdk-11.0.11)

4. Download **Jenkins.war file** and save under any folder

5. Open Jenkins.war under CMD

6. Open the Jenkins.war file in CMD Example - cd C:\Users\NEW-PC\OneDrive\Documents\Jenkins>**java -jar jenkins.war**

7. Jenkins Password would be created on CMD.

Example -

Jenkins initial setup is required. An admin user has been created and a password generated.

Please use the following password to proceed to installation:

b8aed81fdca1442db878ac7b114917c8

8.Enter http://localhost:8080/ in your PC (Jenkins URL)

9.Copy Paste the password the generated Password on CMD (Don't Close CMD) (Recommended to change password Later)

10. Click **Install suggested Jenkins** on the first page.

(It will take roughly 2 hours) based on internet speed

11. Create Admin User Password

12. Jenkins is Ready Now !!

13. Click on New Item

14. Have to download **Maven Plugin** --> Manage Plugins --> Search for Maven --> Select all Maven related plugins and Click Install now and restart later

15. Once All Successfully installed then again need to login on Jenkins with the username and password

16. http://localhost:8080/safeRestart --- To Restart

17. If you forgetfully close CMD then your local-host won't work. Need Jenkins Up and Running in CMD.

18. After restart Got to New Item --> we can see Maven Project added.

19. Give any name on the Enter the Item name and select Maven --> Click OK

20. General --> Enter Project Name --> Come to bottom of the page and enter the pom.xml file from eclipse.

Click on tool configuration --> Click on Add Maven --> Give any name Example - "TestMaven" and Click Save and Apply. (Close the Newly Opened Tab)

21. Copy Paste the Pom.XML file path under Root POM Example -

C:\Users\NEW-PC\eclipse-workspace\VzAutomationTest\pom.xml

21.Goals and Options -- **“clean install”**

22.Post Steps -- Run regardless of built result

23.Post-build Actions -- Editable Email Notification (Have to download testng plugin) Apply and Save.

24.Jenkins -- Manage Jenkins -- Manage Plugins -- **Search for TestNG** -- Select and Install.

25.Restart Jenkins

26. Go to created project --> configure --> click Add post-build action --> Select publish TestNG Report --> Apply and Save.

27.Now click Build and see the executions.

-----------------------------------------------------------COMPLETED------------------------------------------------------------

**---------------- Check in Code on GIT Hub and Run Test cases from GIT Hub Using Jenkins -----------**

1. Create GitHub Account
2. Start a Project = Create a Repository **Ex:”PageObjectModel”** and give Description if needed.
3. Click on Create Repository
4. Now HTTPS link will be generated. Ex - **https://github.com/mohammedferozali/PageObjectModel.git**
5. Start Eclipse
6. Go to Perspective (Top Right Corner Next to Search) - Click on Git or Search for **“Git Repositories”**
7. Click on (**“Create a New Local Git Repository”**)
8. Next --> Set the Directory Path (Keep it Default) --> Finish
9. Right Click the Selenium Project --> Team --> Share --> Select the added Repository Path on the drop down of Repository under Configure Git Repository Pop UP and Click Finish.
10. Observe the Project added icon of ? - means it is now Git Project.
11. Do a Right Click on Project --> Team --> Commit --> It will show all the changes of the Project.
12. Drag all the file and move to Staged Changes.
13. Enter “First Commit” or any Comment on the Commit Message Box.
14. Click on Commit or Commit and Push.
15. Click Preview on Push to branch in remote Pop Up. (Observe “branch name shown on the pop up)
16. Click Push.
17. Enter your Username and Password of GitHub.
18. Make sure “Finish” button is shown. If Configure link is shown then your project is not created properly. Practice with dummy project and do for real project :) ..