# Integration of maven project with Jenkins pipeline(How to create maven project and integrate maven project with Jenkins pipeline?)

CREATE MAVEN PROJECT:

1. Add maven project.
2. Open pom.xml
3. Enter following reference in pom.xml

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>2.45.0</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.8.8</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.google.inject</groupId>

<artifactId>guice</artifactId>

<version>4.1.0</version>

<classifier>no\_aop</classifier>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>2.3.2</version>

<configuration>

<source>1.7</source>

<target>1.7</target>

</configuration>

</plugin>

<plugin>

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

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

<version>2.12</version>

<inherited>true</inherited>

<configuration>

<suiteXMLFiles>

<suiteXMLFile>testing.xml</suiteXMLFile>

</suiteXMLFiles>

</configuration>

</plugin>

</plugins>

</build>

1. Select maven project and convert into TestNG project.testing.xml file will add into project
2. Configure build path. Add TestNG library
3. Configure build path. Add reference library of selenium web driver.
4. Navigate to MavenProject/src/test/java.
5. Add TestNG class.
6. Write code in class

**package** Package;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.Assert;

**import** org.testng.Reporter;

**import** org.testng.annotations.Test;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.AfterTest;

**public** **class** NewTest {

**private** WebDriver driver;

@Test

**public** **void** test() {

//driver.get("http://demo.guru99.com/test/guru99home/");

driver.navigate().to("http://demo.guru99.com/test/guru99home/");

Reporter.*log*("Demo Guru99 website open");

String title = driver.getTitle();

Assert.*assertTrue*(title.contains("Demo Guru99 Page"));

Reporter.*log*("Verify title");

}

@BeforeTest

**public** **void** beforeTest() {

System.*setProperty*("webdriver.chrome.driver", "D:\\Selenium web driver java\\Setups files\\chromedriver\_win32\\chromedriver.exe");

driver = **new** ChromeDriver();

}

@AfterTest

**public** **void** afterTest() {

driver.quit(); }

}

1. Right click ‘JRE System Library’. Select Properties. Select radio button ‘Workspace default JRE (jdk 13.0.2).Click Apply & Close.
2. Execute test. Test should run successfully.
3. Open Jenkins.
4. Create new maven pipeline.
5. Add pom.xml full path ‘C:\Users\fgul\eclipse-workspace\MavenProject\pom.xml’.
6. Click SAVE.
7. Click Build now.
8. Maven project should be successful.

Pre-requisite:

1) Install Maven and set maven environment variable. PATH= C:\Users\ffff\Downloads\apache-maven-3.6.3-bin\apache-maven-3.6.3\bin

2) Install JDK and set maven environment variable. PATH=C:\Program Files\Java\jdk-13.0.2\bin CLASSPATH=C:\Program Files\Java\jdk-13.0.2\lib

3) Install JRE and set environment variable. PATH = D:\Program Files\Java\jre1.8.0\_231\bin

4) Install Jenkins.

5) Install Maven integration plugin into Jenkins.

6) Set maven and JDK path into Global Configuration Tools JDK C:\Program Files\Java\jdk-13.0.2 Maven C:\Users\fff\Downloads\apache-maven-3.6.3-bin\apache-maven-3.6.3