**Build and Configure CI/CD Pipeline with Maven Project**

**Step 1:** Setting up a Pipeline

* Open Jenkins- locahhost:8080.
* Click on **Manage Jenkins.**
* Select **Global Tool Configuration.**
* Set the path of **Java and Maven.**
* Then, **Apply and Save.**
* Select a **New Item.**
* Name the job: **Jenkins-maven-build-pipeline.**
* Select the **Pipeline.**
* Now, click on **OK.**

**Step 2:** Writing a Jenkins file

* Create a job that needs to be configured.
* Click on **Pipeline.**
* Select the definition as- **Pipeline script from SCM.**
* Pass the GitHub URL.
* Now, we need to write the Jenkins file inside that git file.
* Now, click on ‘Apply’ and ‘Save.’
* Click on **Build Now.**

**Source Code**

**pom.xml**

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>NinjaDemo-POMFramework</groupId>

<artifactId>NinjaDemo-POMFramework</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>4.12.1</version>

</dependency>

<dependency>

<groupId>io.github.bonigarcia</groupId>

<artifactId>webdrivermanager</artifactId>

<version>5.4.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.testng/testng -->

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.7.1</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/com.aventstack/extentreports -->

<dependency>

<groupId>com.aventstack</groupId>

<artifactId>extentreports</artifactId>

<version>5.0.9</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>4.1.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>4.1.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml-schemas -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml-schemas</artifactId>

<version>4.1.1</version>

</dependency>

</dependencies>

<build>

<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

<plugins>

<!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

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

<version>3.8.0</version>

</plugin>

<plugin>

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

<version>2.22.1</version>

<configuration>

<suiteXmlFiles>

<!-- <suiteXmlFile>src/main/resources/TestRunner3.xml</suiteXmlFile> -->

<suiteXmlFile>src/test/resources/TestNgRunner.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

<plugin>

<artifactId>maven-jar-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

<!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

<plugin>

<artifactId>maven-site-plugin</artifactId>

<version>3.7.1</version>

</plugin>

<plugin>

<artifactId>maven-project-info-reports-plugin</artifactId>

<version>3.0.0</version>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>

**TestNgRunner.xml**

<suite name=*"test1"* verbose=*"1"*>

<test name=*"POMframework"*>

<classes>

<class name=*"com.ninjademo.qa.testpages.HomePageTest"*></class>

</classes>

</test>

</suite>

**Jenkin Maven Pipeline Script**

pipeline{

tools{

maven 'mymaven'

}

agent any // node where we have to run pipeline

stages{

stage('Clone the repo')

{

steps{

git ‘https://github.com/PruthvishreeSK/LMSProjects/tree/main/NinjaDemo-POMFramework’

}

}

stage('CompileCode'){

steps{

sh 'mvn compile'

}

}

stage('TestCode'){

steps{

sh 'mvn test'

}

}

stage('PackgeCode'){

steps{

sh 'mvn package'

}

}

}

}