**Which type of build tool we are using in previous framework? VVVVVIMPPPPPPPPPPPPPPP**

What is Maven? VVVVVIMPPPPPPPPPPPPPPP

What is POM.xml file? VVVVVIMPPPPPPPPPPPPPPP

diff between POM and pom.xml file? VVVVVIMPPPPPPPPPPPPPPP

What is Maven life cycle? VVVVVIMPPPPPPPPPPPPPPP

What are the different commands present inside the Maven life cycle? VVVVVIMPPPPPPPPPPPPPPP

how to run the maven project? VVVVVIMPPPPPPPPPPPPPPP

**What are the different build tools available in market for java platform? VVVVVIMPPPPPPPPPPPPPPP**

1) maven build tool

2) Gradle build tool

3) ant build tool

**What is Maven? VVVVVIMPPPPPPPPPPPPPPP**

- Maven is an build tool and it is developed by the apache group.

- Maven tool is used to build the project, publish the project and deploy to several project for better project management.

- Maven build tool is written in java language.

- Maven build is based on POM.xml concept

- Maven build tool has made java developer and java tester life easier while developing the scripts, generating the reports.

**What is pom.xml file? VVVVVIMPPPPPPPPPPPPPPP**

- pom.xml file means Project Object Model dot extensive markup language.

- this pom.xml file is created automatically when we create the maven project.

- in pom.xml file we mention different things as like

I) project name by using artifact id VVVVVIMPPPPPPPPPPPPPPP

2) package name by using group id VVVVVIMPPPPPPPPPPPPPPP

3) version name by using version name tags

4) packaging project in jar or war file

5) we mention java version name inside the properties tags

6) we add different dependency inside the dependencies tags

we use different dependency

I) selenium java dependency

2) testng maven dependency

3)common io dependency

4) apache POI dependency

5) extent reports dependency

6) AShot dependency

7) cucumber-java dependency

8) cucumber-core dependency

9) cucumber-junit dependency

10) cucumber-testng dependency

11) junit dependency

7) we add different plugins in build tags in project

- to run the project from window command line or GitBash command line or Jenkins tools

I) maven compiler file

2) maven surefire plugin

3) maven fail safe plugin

- pom.xml file is located in project home directory.

- When we run the project or test cases, maven first it searches pom.xml file in project home directory and then it will perform the task.

- When we add any dependency inside the pom.xml file, automatically maven will convert dependency into the jar files.,

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

What is Maven life cycle? VVVVIMMPPPPPPPPPPPPPP

or

what are the different phase in maven life cycle? VVVVIMMPPPPPPPPPPPPPP

1) maven install

- if we use maven install phase it will downloaded the latest jar files from official maven repository server and it will install in (.m2) local repository.

-how to run the maven install phase,

right click on project ---> go to run as options and click on maven install.

2) maven clean

- if we use maven clean phase, it delete all files and folders from target directory/folder.

how to use maven clean phase?

- Right click on project ---> go to run as Options and click on maven clean

3) maven validate phase

- if we use maven validate phase, it check project is correct or not>

how to use maven clean phase?

- Right click on project ---> go to run as Options and click on maven build and then maven goal as validate and click on apply

4) maven compile phase

- if we use maven compile phase , it will compile the source code and classes stored in source folder.

how to use maven compile phase?

- Right click on project ---> go to run as Options and click on maven build and then maven goal as compile and click on apply

5) maven package phase

once we use maven package phase, it will compiled source code and package into the jar or war file.

how to use maven package phase?

- Right click on project ---> go to run as Options and click on maven build and then maven goal as package and click on apply

6) maven verify phase

- once we use maven verify phase, it will verify the packages jar or war file is correct or not

how to use maven verify phase?

- Right click on project ---> go to run as Options and click on maven build and then maven goal as verify and click on apply

7) maven test phase

- once we use maven test phase, it will run the test cases

how to use maven verify phase?

- Right click on project ---> go to run as Options and click test

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

go to project home directory in window command line and run the maven commands

1) clean

mvn clean

2) install

mvn install

3) validate

mvn validate

4) compile

mvn compile

5) package

mvn package

6) verify

mvn verify

7) test

mvn test

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

we have to install maven in local machine and as well as

we have configure the java and maven in environment system variables

<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>OrangeHRM</groupId>

<artifactId>OrangeHRM</artifactId>

<version>2.7.0</version>

<packaging>jar</packaging>

<name>OrangeHRM</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

<dependencies>

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

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

<!--

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

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>4.16.1</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.3.0</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>testng.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Repository in maven**

**there are 3 types of repository in maven**

**1) Local Repository**

**2) Central Repository**

**3) Remote Repository**

1) Local Repository

- once we add any dependency or jar file in project it store in local repository in .m2 folder

e.g. local repository location

c://user/{PC name}/.m2/repository/....etc

2) Central Repository

- organization / company store all dependency or jar files in central server or company server.

- it is Central repository server which is private.

- and it is used by the organization employee.

e.g.

https://www.amdocs.com/repository

3) Remote Repository

- it is official maven repository server, which is public.

- anybody can use these remote repository server.

e.g.

https://mvnrepository.com/

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

if we want to use maven commands, then first we have to download the maven source files in local machine,

then we have to set the maven source file location in environment variables.

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

if we want to run the specific testng.xml file then we have to mention location of file in pom.xml file.

<project etc...>

<dependencies>

<dependency>

<groupId></groupId>

<artifactId></artifactId>

<version></version>

<scope> test </scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.3.0</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile> testng.xml </suiteXmlFile>

</suiteXmlFiles>

</configuration>

<plugin>

<plugins>

</build>

</project>

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

Maven goal / phases in maven life cycle

1) maven clean

mvn clean

2) maven install

mvn install

3) maven validate

mvn validate

4) maven compile

mvn compile

5) maven package

mvn package

6) maven verify

mvn verify

7) maven test

mvn test

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

mvn clean install test

mvn tree

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

how to run the test cases?

how you are running test cases in previous company?

- we use maven commands to run the test cases.

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

<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>com.orangehrm</groupId>

<artifactId>OrangeHRMApplication</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>OrangeHRMApplication</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>4.16.1</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.3.0</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>testng.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

</plugins>

</build>

</project>

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

<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>com.orangehrm</groupId>

<artifactId>OrangeHRMApplication</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>OrangeHRMApplication</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.4.0</version>

<scope>test</scope>

</dependency>

<dependency>

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

<artifactId>selenium-java</artifactId>

<version>4.16.1</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.3.0</version>

<configuration>

<includes>

<include>\*\*/\*Test.java</include>

</includes>

</configuration>

</plugin>

</plugins>

</build>

</project>