Q. What is maven and why ?

Ans: It is project management tool .Maven is build tool .it will help to manage dependency .

Java build tool = Ant --- > Maven ----- > Gradle

C and C++ = make file

Maven is written in java ,

Maven will give the whole java structure in java application .

POM.xml includes :

Metadata : information of data , Kind of project , Kind of output(jar or war) , description

Q> maven Life cycle : 1. Generate resource (Dependencies) 2. Compile code 3. Unit test 4. Package (build) 5. Install(into local repo and artifactory) 6. Deploy( to server) 7. Clean ( delete all run time files)

Sequence order 1 to 6 .

How to run maven project in command line .

1st : change directory to project directory

2nd: mvn install

3rd: java -jar <building jar : coming from mvn install : target/applicationName.SNAPSHOT:100.jar>

4th : or we can use : mvn spring-boot:run

####### $$$$$$$$$$$$$$$$ %%%%%%%%%%%%%%%

How to run maven springboot application in Kubernetes or minikube cli

########################### $$$$$$$$$$$$$$$$$$$$$$$$$$$$

1st : clone the project into linux or Kubernetes server

2nd : mvn clean package -- > it will compile the whole code including “test cases “

3rd: mvn clean package -Dmaven.skitp.test=true -- > it will skip the test cases and compile

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

In Eclipse no need of prefix mvn , just write -> clean or install etc: run as - - > build.. - > clean install -X

mvn dependency:tree 🡪 it will give dependency list in tree structure

In pom.xml == >> put this properties file ( same version which we use in jdk and compiler)

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<build>

<plugins>

<plugin>

<!-- Build an executable JAR -->

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

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

<version>3.1.0</version>

<configuration>

<archive>

<manifest>

<mainClass>shambhu.Hello</mainClass>

</manifest>

</archive>

</configuration>

</plugin>

</plugins>

</build>

Note : in main class we need to provide actual main classs

Note : set the same jdk and compiler version from build path

**What is multimodule springboot application ?**

ANS : If we create a parent project and this parent project holds other project like web , service and dao layer as different project then we say this is multimodule spring project

While creating parent project we need to give packaging type : pom ( no jar or war)

* > parent project will have **packaging : pom** in pom.xml

Module project added automatically when we make module for parent - > find in pom.xml under <Module><module>

* > we can add common dependencies under
* **<dependencyManagement>**

**<dependencies>**

**<dependency><dependency>**

**<dependencies><dependencyManagement>**

needs for module or child project . Dependency usesd in parent project will be used for all the child project

* - > how this dependency will be used by child project ?

We need to give only group id and artifactId under

**<dependencies>**

**<dependency>**

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

**<artifactId>testing</artifactId>**

**<dependency>**

**<dependencies>**

Q . How to create web or service or dao layer as module project for parent project ?

While creating module : packaging type will be : jar ( not pom)

1st : we create independent project like : select parent project and go to -- > file -- > new -- > other - > maven module - >

Name: Dao , or web , or service

2nd : we can see in pom.xml file -- > parent project reference

* - > at the same time we can see module project is added in parent project’s pom.xml
* > in module project no packaging type given (it happens by default)