**APACHE** MAVEN

* Maven is a free and open source software given by Apache Organization.
* Maven is called as Java Build Tool.
* Maven is used to perform Build Automation for java projects.
* Note: Maven is used as a build tool for only java projects.

**Why Maven?**

1) We can create initial project folder structure using maven.

2) We can download "project dependencies" using maven

(ex : spring boot, hibernate, kafka, redis, email, log4j, junit, security...)

* To develop one java project we will use several frameworks like spring, hibernate.
* We need to download those frameworks and we should add to our java project.
* These frameworks we are using in our project are called as our project dependencies
* Instead of we are downloading dependencies, we can tell to maven software to download dependencies.
* Note: Required dependencies we will add in "pom.xml" file then maven will download them.
* **pom** stands for project object model.
* When we create maven project then pom.xml file will be created automatically
* pom.xml will act as input file for maven software

3) We can compile project source code using maven.

4) We can package java project as jar or war file using maven.

* Java program files will have .java as extension
  + Ex: Demo.java, Hello.java, Driver.java, Calculator.java etc.
* Java Programs (.java file) contains source code, we can't execute .java files directly
* We need to compile java source code into byte code using java compiler (javac)
  + Ex: javac Demo.java
* When we compile java code it will create .class file
* We need to execute .class file to run the java program
  + Ex: java Demo
* When we run java program using java command, JVM will start and it will execute java program
* Note: JVM stands for Java Virtual Machine
* When we compile project source code we will get .class files
* To deploy java project, we will package all .class files as JAR file or WAR file
  + JAR : Java Archive
  + WAR : Web Archive
* Standalone java projects will be packaged as a JAR file
* Note: Standalone applications means the project which runs only in one computer
  + Ex: notepad, calculator....
* Web Applications will be packaged as WAR file
* Note: Multiple users can access web applications at a time
  + Ex: gmail, facebook, flipkart, naukri etc....

**MAVEN INSTALLATION: (Using RHEL/CentOS)**

Step#1 Check Java and Maven are installed

$ java -version

$ mvn --version

Step#2 Install wget if not exist

$ sudo yum install wget -y

Step#3 Install Java 11 Software (JRE)

$ sudo yum install wget java-11-openjdk -y

$ java -version

Step#4 Download Maven Software

$ cd /tmp

$ sudo wget https://dlcdn.apache.org/maven/maven-3/3.9.1/binaries/apache-maven-3.9.1-bin.tar.gz

Step#5 Extract to a folder

$ sudo tar xvzf apache-maven-3.9.1-bin.tar.gz -C /opt/

$ cd /opt

# rename to a short folder name

$ sudo mv apache-maven-3.9.1 maven

Step#6 Set Maven Home Directory

**# Optional if execute permissions are given**

$ sudo chmod +x /etc/profile.d/maven.sh

$ sudo vi /etc/profile.d/maven.sh

# insert below lines

export M2\_HOME=/opt/maven

export PATH=${M2\_HOME}/bin:${PATH}

(ESC :wq)

Step#7 Reload Profiles for updating path

$ source /etc/profile.d/maven.sh

Step#8 Check Maven Installation

$ mvn -version

Step#9 Remove TAR ZIP File

$ sudo rm -f /tmp/apache-maven-3.9.1-bin.tar.gz

**[JAVA FOR DIFFERENT OS]**

**For Ubuntu:**

**$ sudo apt install openjdk-11-jdk openjdk-11-jre -y**

**For Amazon Linux:**

**$ sudo yum install java-11-amazon-corretto-devel -y**

**For Redhat Linux:**

**$ sudo yum install java-11-openjdk -y**

**MAVEN TERMINOLOGY:**

archetype, groupId, artifactId, packaging, version

**Archteype** represents what type of project we want to create

maven-archetype-quickstart : It represents java standalone application

maven-archetype-webapp : It represents java web application

Note: Maven providing 1500+ archetypes

**groupId** represents company name or project provider name

**artifactId** represents project name or project module name

**packaging** represents how we want to package our java application (jar or war)

**version** represents project version

**MAVEN REPOSITORY:**

Maven will download dependencies using repository

In Maven we have 3 types of repositories

1) Central Repository: central repository is maintaining by apache organization

2) Remote Repository: Every organization will maintain their own remote repository

3) Local Repository: Local repository will be created in our system (C:/users/< username >/.m2 (or) /home/<username>/.m2)

* When we add dependency in pom.xml, maven will search for that dependency in local repository. If it is available it will add to project build path.
* If dependency not available in local repository then maven will connect to Central Repository or Remote Repository based on our configuration.

Note: By default maven will connect with central repository. If we want to use remote repository then we need to configure remote repository details.

Note: Every software company will maintain their own remote repository (Ex: JFrog, Nexus)

**MAVEN GOALS:**

To perform project build activities maven provided several goals for us.

**clean**: clean goal is used to delete target folder

**compile**: compile goal is used to compile project source code. Compiled code will be stored in target folder ( convert .java to .class)

**test**: test goal is used to execute unit test code of our application (JUnit code)

**package**: package goal is used to generate jar or war file for our application based on packaging type available in pom.xml file. Jar or War file will be created in target folder.

**install**: install goal is used to install our project as a dependency in maven local repository.

Note: Every maven goal is associated with maven plugin. When we execute maven goal then respective maven plugin will execute to perform the operation.

**Syntax : mvn <goal-name>**

Note: We need to execute maven goals from project folder (where pom.xml exist)