Apache Maven

What are Build Tools?

-> Build tools are used to automate repetitive tasks involved in application build process

- Compile The Source Code
- Download Required Dependencies (Ex: Springboot, hibernate, Junit, log4j, Kafka...)
- Execute Unit Test Cases (JUnits)
- Package our application as jar / war

JAR ---> Java Archieve ---> It is package format for java standalone application

WAR ---> Web Archieve ---> It is package format for java web applications

-> Instead of we are doing the above build steps manually, we can take the help of Build Tools to automate that process.

-> We have below build tools for java applications

1. Ant (Outdated)
2. Maven
3. Gradle
4. MS Build

Maven

-> Maven is a free and open-source software given by Apache Organization

-> Maven s/w is developed using Java programming language

-> Maven is used to perform Build Automation for java projects

-> Maven is called as Java Build Automation Tool

Note: Maven is used as a build tool for only java projects.

What we can do using Maven?

- We can create initial project folder structure using maven
- We can download "project dependencies" using maven
  (ex : springboot, hibernate, kafka, redis, email, log4j, junit, security...)
- We can compile project source code using maven
- We can execute Unit Test cases (Junits) using maven
- We can package java project as jar or war file using maven

Maven's Objectives

- Making the build process easy
- Providing a uniform build system
- Providing quality project information
- Encouraging better development practices

Maven Installation in Windows

Step-1: Download and install Java software

Link To Download Java: [*https://www.oracle.com/in/java/technologies/downloads/#jdk17-windows*](https://www.oracle.com/in/java/technologies/downloads/#jdk17-windows)

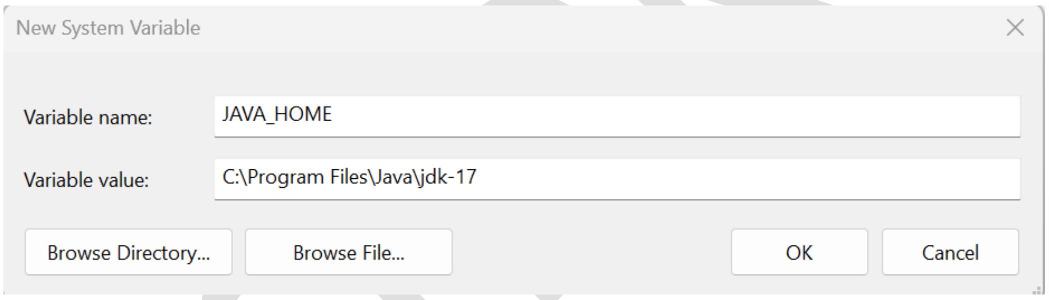
Step-2: Set JAVA_HOME in Environment Variables (System Env Variables)

User Environment Variables: Specific to particular account which logged in our PC

System Environment Variables: For All User Accounts

Note: Environment Variables will be used by operating system

JAVA_HOME = <path-upto-jdk-directory>



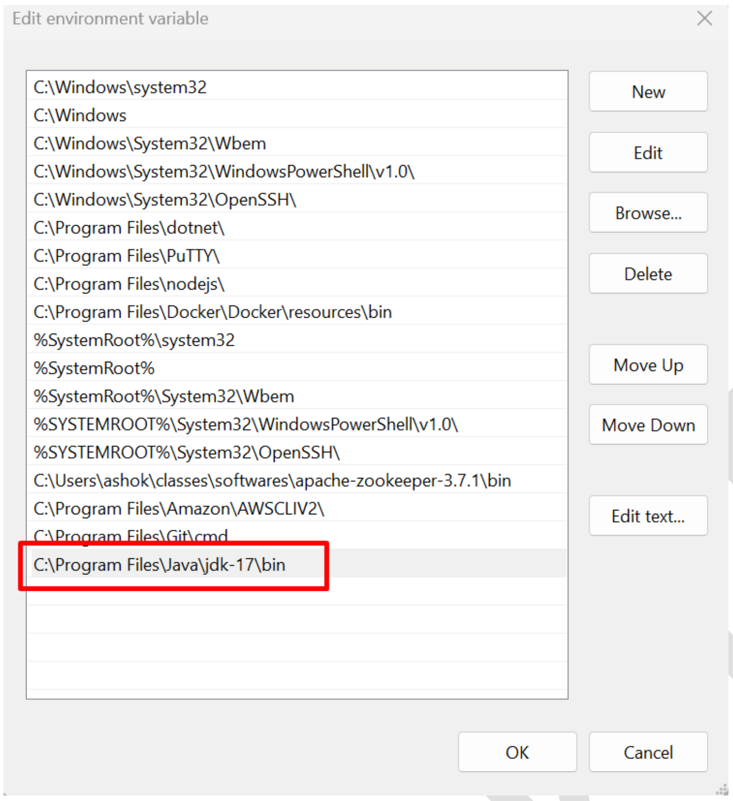
Variable name: JAVA_HOME

Variable value: C:\Program Files\Java\jdk-17

Step – 3: Set Path for JAVA

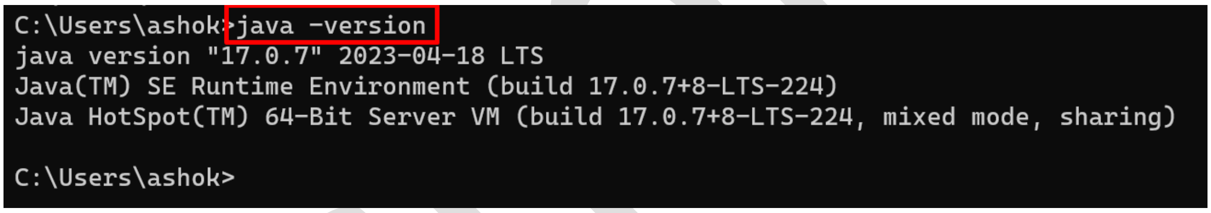
(Go to System Env Variables -> Env Variables -> System Variables -> Select Path and Click on Edit then add JDK path like below up to bin directory)

Path = <up to-bin-directory>



```
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
C:\Program Files\dotnet\
C:\Program Files\PuTTY\
C:\Program Files\nodejs\
C:\Program Files\Docker\Docker\resources\bin
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Users\ashok\classes\softwares\apache-zookeeper-3.7.1\bin
C:\Program Files\Amazon\AWSCLIV2\
C:\Program Files\Git\cmd
C:\Program Files\Java\jdk-17\bin
```

Step-4: Verify Java installation by executing below command in "Command Prompt"



```
C:\Users\ashok>java -version
java version "17.0.7" 2023-04-18 LTS
Java(TM) SE Runtime Environment (build 17.0.7+8-LTS-224)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.7+8-LTS-224, mixed mode, sharing)

C:\Users\ashok>
```

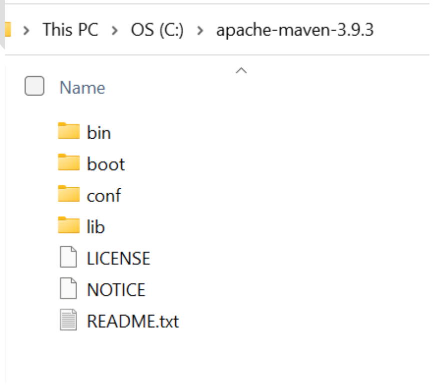
Note: It should display java version which we have installed like above

Step – 5: Download Maven software from Apache website

Link To download Maven: https://maven.apache.org/download.cgi

File Name: apache-maven-3.9.3-bin.zip (Binary Archive)

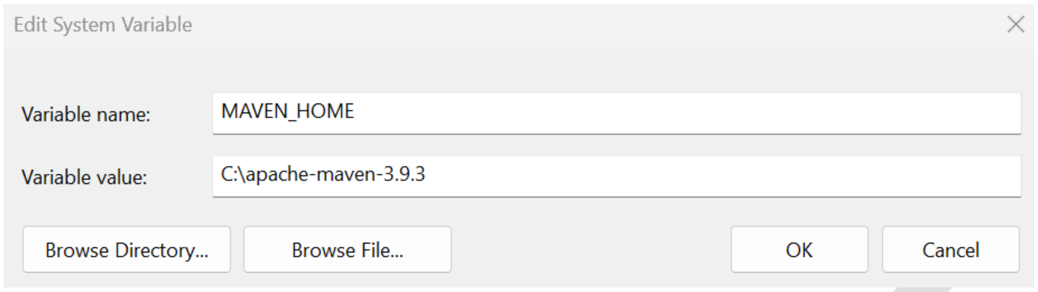
Step - 6: Extract Maven Zip file -> Copy extracted maven folder and paste it in "C" drive



```
This PC > OS (C:) > apache-maven-3.9.3
Name
bin
boot
conf
lib
LICENSE
NOTICE
README.txt
```

Step - 7) Set MAVEN_HOME in System Environment Variables

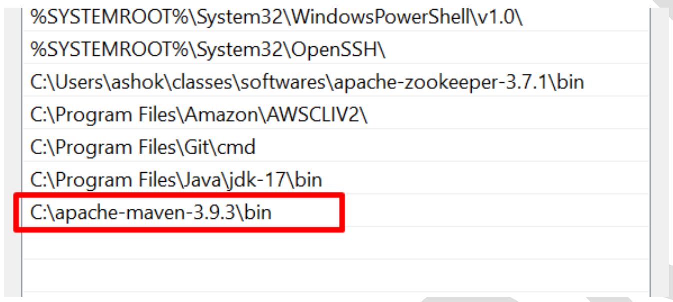
MAVEN_HOME = C:\apache-maven-3.9.3



Variable name: MAVEN_HOME
Variable value: C:\apache-maven-3.9.3

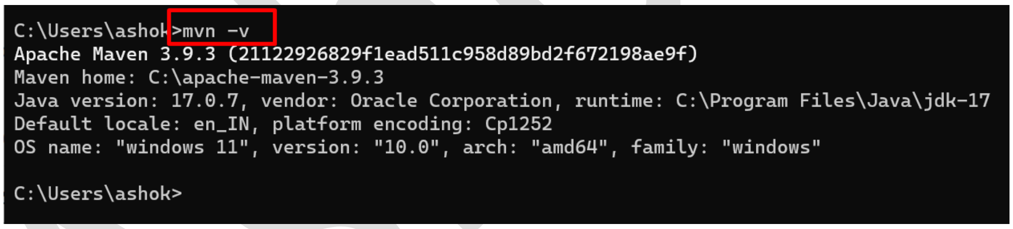
Step-8: Set Path for Maven in System Environment Variables

Path: C:\apache-maven-3.9.0\bin



```
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Users\ashok\classes\softwares\apache-zookeeper-3.7.1\bin
C:\Program Files\Amazon\AWSCLIV2\
C:\Program Files\Git\cmd
C:\Program Files\Java\jdk-17\bin
C:\apache-maven-3.9.3\bin
```

Step – 9: Open Command Prompt and verify Maven Installation using below command



```
C:\Users\ashok>mvn -v
Apache Maven 3.9.3 (21122926829f1ead511c958d89bd2f672198ae9f)
Maven home: C:\apache-maven-3.9.3
Java version: 17.0.7, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"

C:\Users\ashok>
```

Note: With this maven setup completed in Windows OS

<u>**Maven Terminology**</u>

1) archetype	6) maven goals
2) groupId	7) maven plugins
3) artifactId	8) maven repositories
4) version	9) maven dependencies
5) packaging	10) pom.xml

-> **Archetype** 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 name

Ex : com.ashokit

-> **artifactId** represents project name or project module name

Ex: Student-Portal

-> **Version** represents project release version number

Ex: 0.0.1-SNAPSHOT, 1.0-RELEASE

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

-> **Maven goals** are used to perform build process

Ex: clean, compile, test, package and install

-> **Maven plugins** will perform actual build process, for every goal associated plugin is available.

-> **Maven Repositories** will maintain maven dependencies

Ex: Local Repo, Remote Repo and Central Repo

-> **Maven Dependencies** are nothing but jars that required for our project development

Ex : hibernate, spring, kafka, redis, jdbc driver etc…