**Devops:**

* **Devops is a process to automate build and deploy of our application**
* **Devops Process is used to simplify application delivery process**
* **Devops culture will help us to speed up application delivery process**
* **Devops nothing but set of best practices.**

**Devops => Development + Operations**

* **Devops is used to collaborate Development team activities and Operation team Activities.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Build Tool:**

**Apache Maven:**

* **It is used to build java projects**
* **Apache maven is free & opensource**
* **Maven Develop by using Java only.**
* **To Automate or simplify java projects build process we are using Apache maven.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Java:**

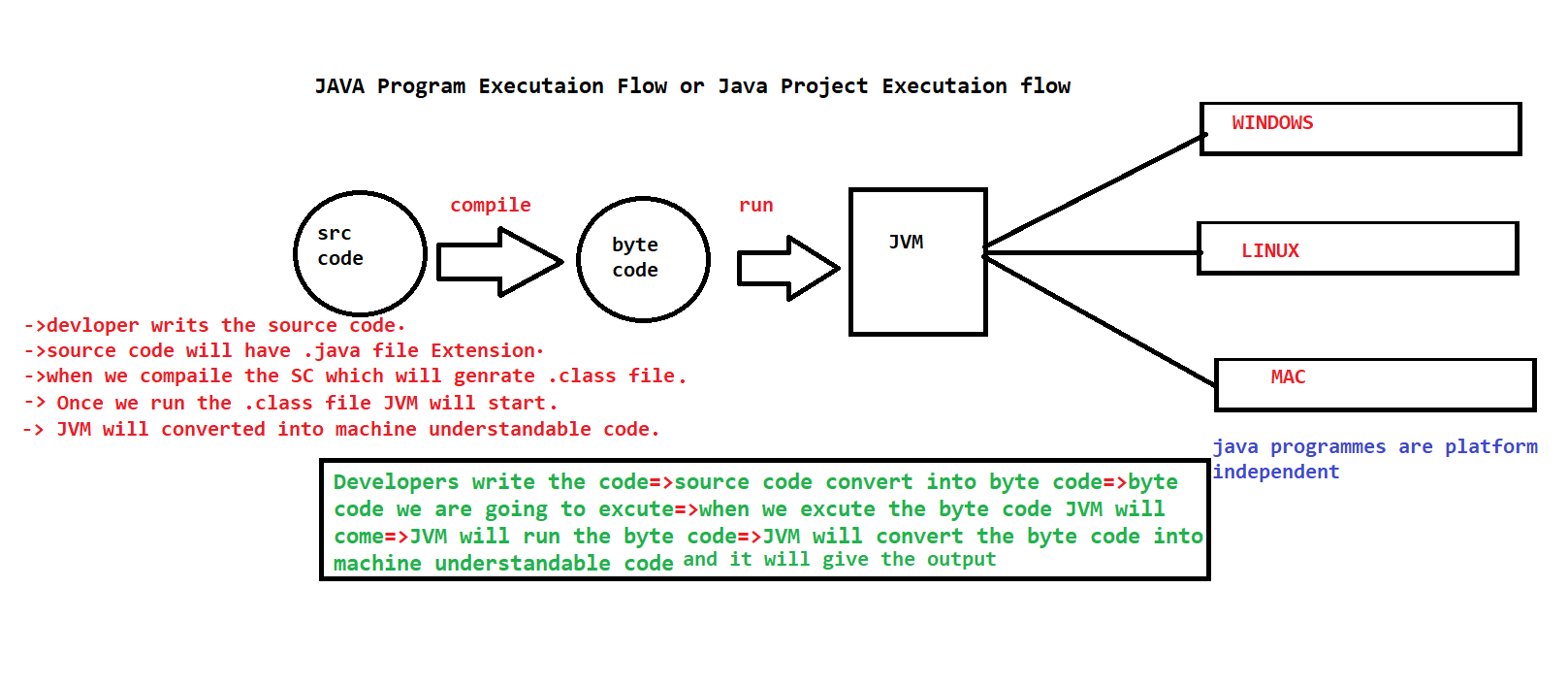
**Before seeing about Maven. We need to know about java?**

* **Java is Programing Language.**
* **Java developed by sun micro system company in 1991.**
* **Oracle company acquired Sun Microsystem in 2010.**
* **Java is under license of oracle corp.**
* **Java it is also free and open source.**
* **Java it is high level program language.(Means human understandable but it does not machine understandable).**
* **Java is Platform independent.(one java program we can execute any computers).**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**How this java program has been executed?**

* **Developers are going to write the program, that program will contents source code.**
* **Source Code cannot be executed directly, that source code we need to compile.**
* **When the compile the source code , it is giving byte code, this is called compilation.**
* **Once we executed byte code . JVM will come( Java virtual machine) .**
* **Once JVM will start , we can execute the program any machines.( window, Linux, mac)**
* **Java program we can be execute any machines. That’s why java is called as platform independent Program.**
* **We can run our program( window, Linux , mac) machines. Because Java Program is Platform independent.**



**Package the files . Why we need to Package file?**

* **Coming to the real team Project.**
* **Single .java files and .class file we can run directly**
* **But what about project , because project will have several java files.**
* **For that we are going use the concept called packaging.**
* **To deploy java project, we will package all .class files as JAR or WAR file**

**JAR => JAVA ARCHIEVE**

**WAR => WEB ARCHIEVE**

**Where we use JAR file and WAR file?**

**Standalone java projects will be packaged as “JAR file”**

**Web Application java Project will be packaged as “WAR file”**

**Note: Standalone Application means the project which runs only in one computer.**

**JAR file will be used for Standalone Application.**

**Ex: notepad , Calculator.**

**Note: Web application means Multiple users can access our web applications at the time over the internet.**

**WAR file will be used for web application.**

**Ex: Gmail, Facebook , flip kart , Naukri etc …**

**Conclusion:**

* **Source code should be compiled into Byte code.**
* **Then we need to execute the byte code , JVM will come and help to convert byte code into binary code.**
* **In Real time Project several Class files will be available , So once we compile the code . then we need to package the code as JAR file or WAR file.**

**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 Tool.**

**Note: Maven is used for building only java project.** **We cannot use for other programming language.(like python, Dotnet, NodeJS ….)**

**What is Build Automation?**

**Project Compilation , Project Packaging .**

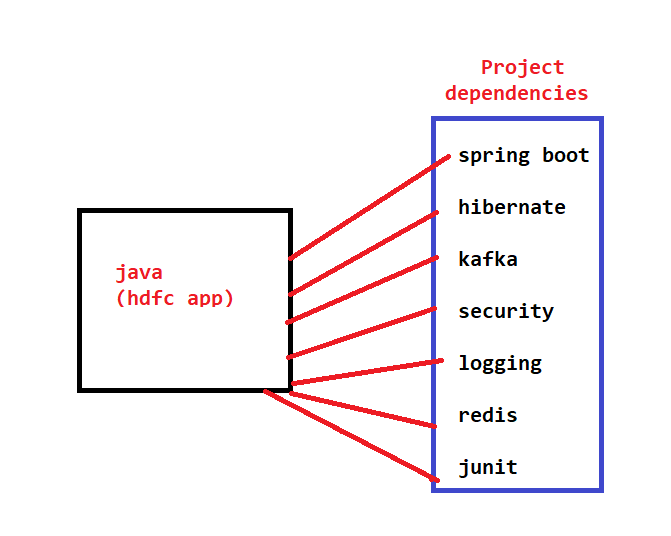
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**What can we do by using the Maven?**

* **We can create initial project folder structure using maven.**
* **We can download “Project Dependencies” by using maven.**

**Ex: springboot,hibernate,Kafka,redis,email,log4J,Junit,Security…**

* **To develop one java project, we will use several frameworks like spring, hibernate etc along with java.**
* **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.**
* **We don’t need to download the dependencies manually.**
* **Instead of we are downloading the dependencies ,we can tell maven S/W to download dependencies.**

****

**NOTE: Required dependencies we will add in “POM.XML” file than maven s/w 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.**
* **In POM developers are going to configure which dependencies this maven has to download**
* **We can compile project source code using maven**
* **We can Package project as jar or war file using maven**

**Project creation, dependencies downloading, Project Compilation , Project Packaging everything will be done by maven.**

**1.Create Project Folder structure.**

**2.Download required dependencies based on POM.XML configuration.**

**3.Source Code Compilation (.java files Program logic)**

**4.Project Packaging. ( jar or war )**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Maven Installation:**

**First, we need to install java . when we install java , which will give JDK and JRE**

**JDK : java Development Kit**

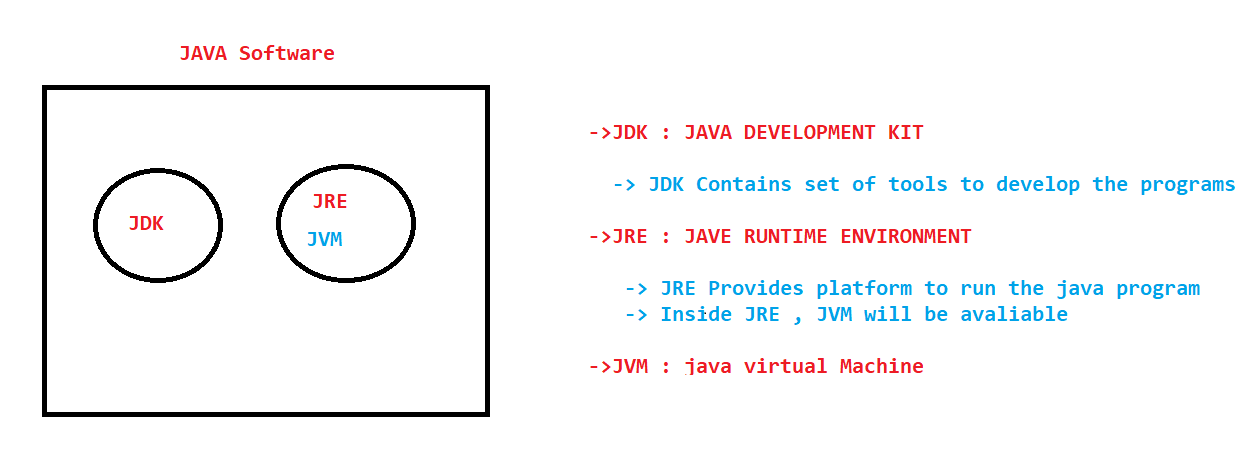
* **JDK Contains set of tools to develop java programs**

**JRE: Java Runtime Environment**

* **JRE Provides Platform to run java program**
* **Inside JRE, JVM will be available**

**JVM : Java Virtual Machines**

* **When we want to develop the java Program, we need to install JDK.**
* **When we want to run the program, we need to install JRE.**



**We need to setup the Environmental variable for OS. We need to inform the OS where we install java and Maven**

**JAVA\_HOME Setup**

**JAVA Path Setup**

**MAVEN\_HOME Setup**

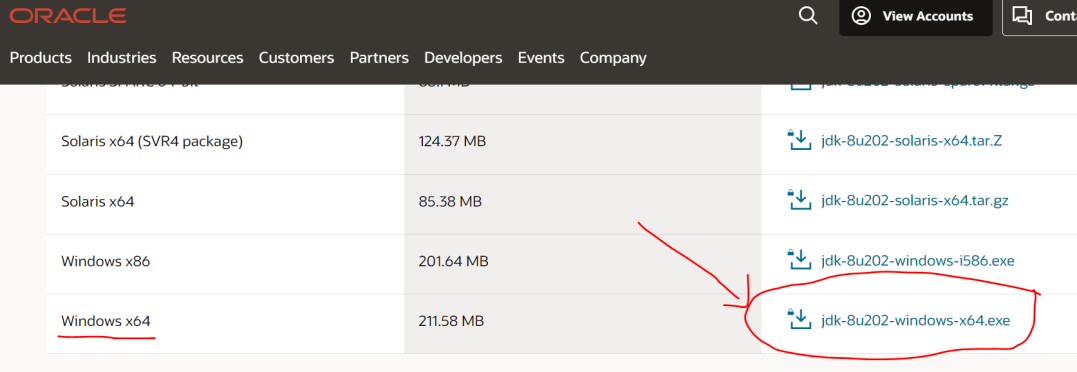
**MAVEN Path Setup**

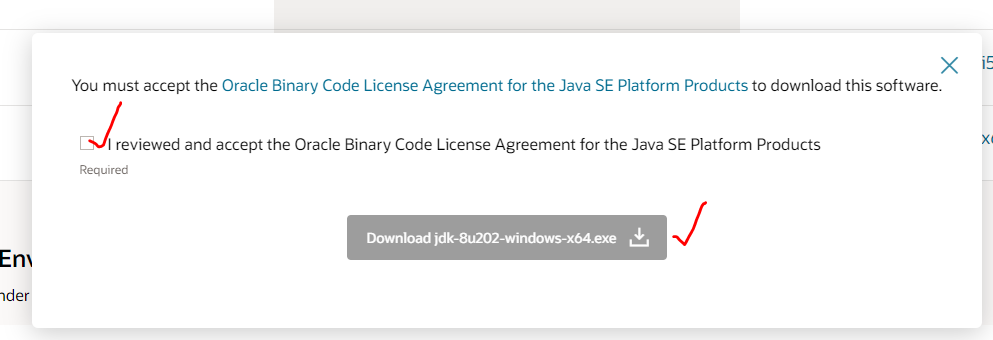
**Kindly follow the below mentioned Setups**

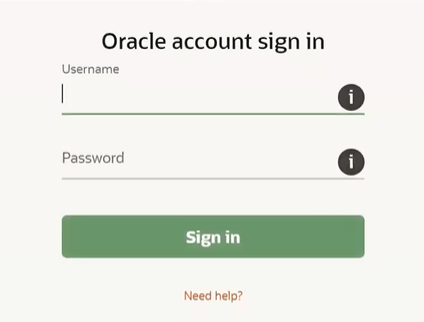
**Download JAVA:**

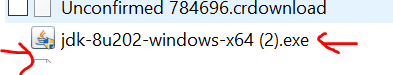
**STEP:1**

**<https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html>**







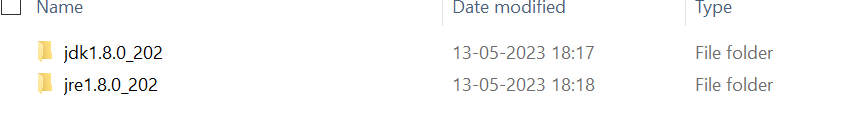


**🡺login the oracle account 🡺then it will automatically come .exe file**

**🡺double .exe file 🡺install it 🡺Next (4 times)**

**🡺Once you install JDK and JRE both will install.**

**Then go to C drive 🡺 Program files 🡺 java folder🡺 we able to see both JDK and JRE folder.**

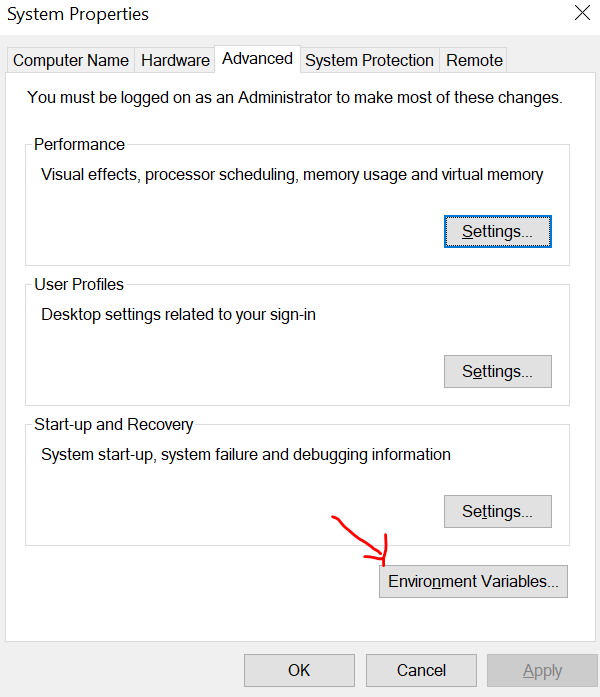


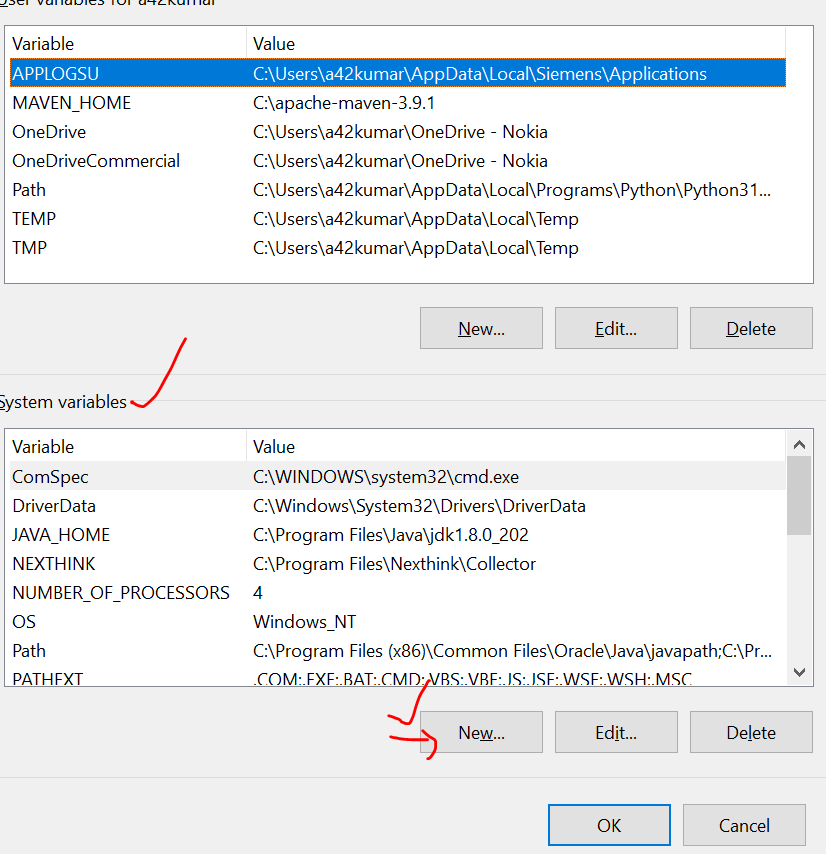
**JAVA by default it will install C drive Folder.**

**STEP:2**

**Set System Environment variable:**

**Go to windows search option🡺Edit the system environment variables🡺system properties**

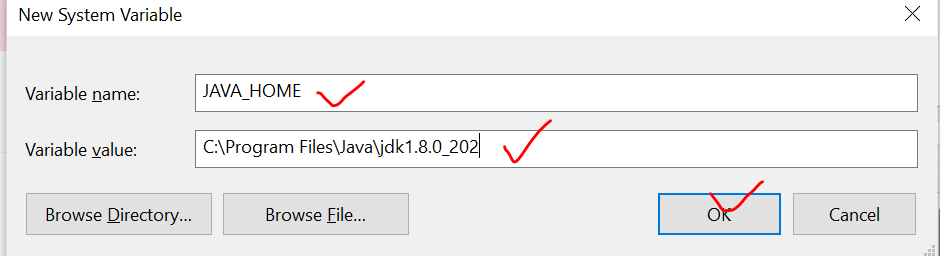


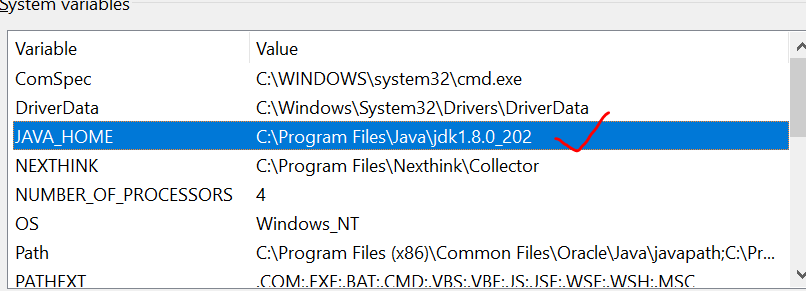


**Click New🡺SET VARIABLE NAME (JAVA\_HOME) 🡺 SET VARIABLE PATH (LOCATION OF JDK PATH IN YOUR SYSTEM)**

**JAVA\_HOME=C:\Program Files\Java\jdk1.8.0\_202**

**Example:**



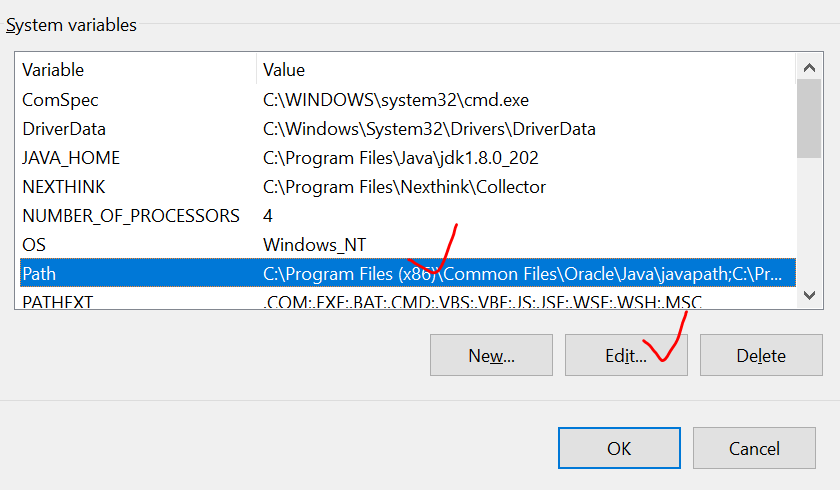


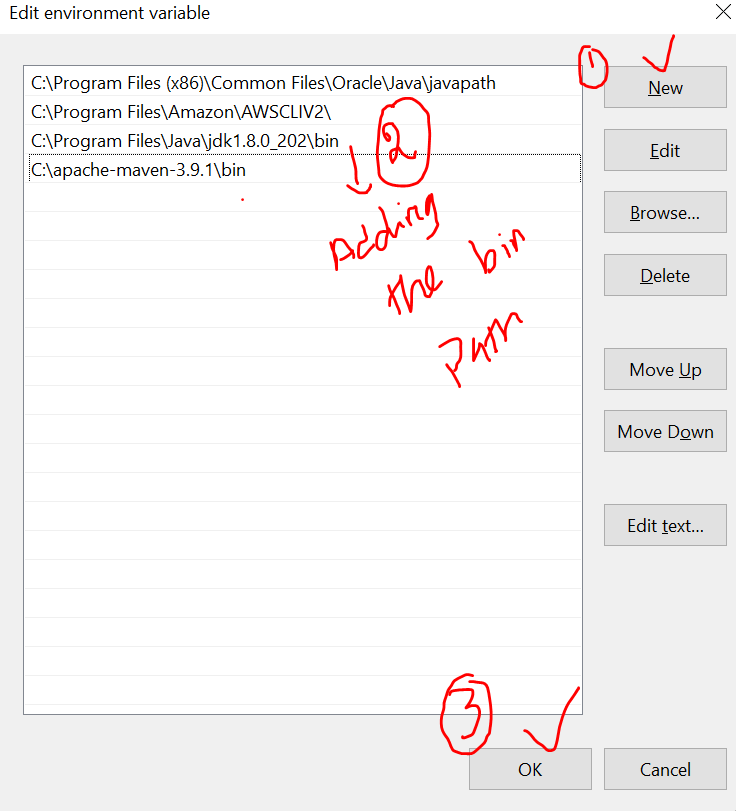
**STEP:3**

**After that need to add the path of JDK bin.**

**C:\Program Files\Java\jdk1.8.0\_202\bin**

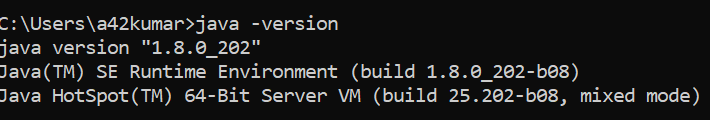
**Go to environment system variable 🡺 Click Path 🡺 Edit Button 🡺 NEW🡺COPY BIN PATH🡺OK**





**Once installation and setup completed , we can check and confirm the run command in windows.**

**$ mvn -version**

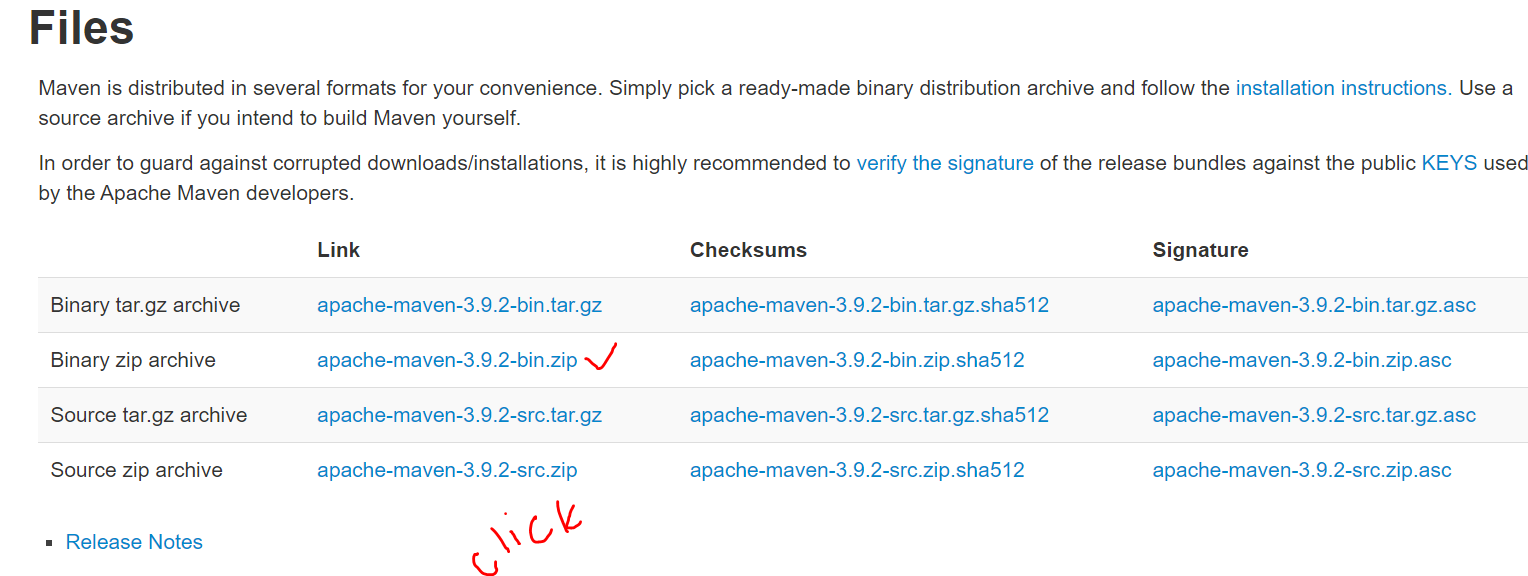


**Maven Installation:**

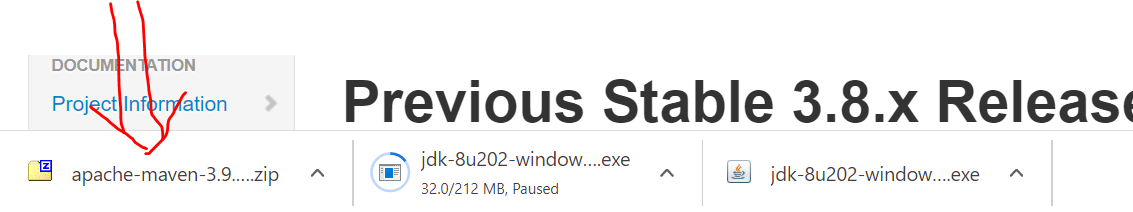
**Step-1**

**Link below:**

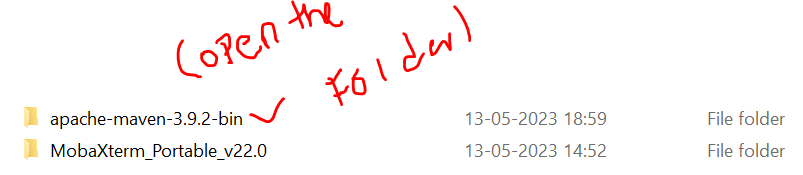
**<https://maven.apache.org/download.cgi>**



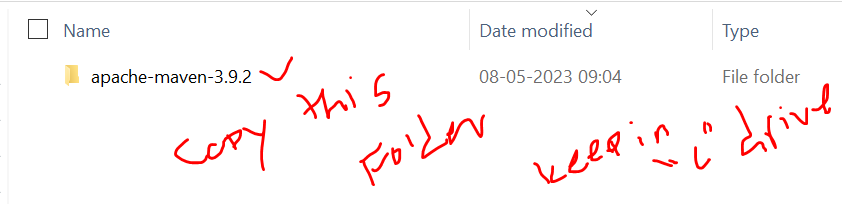
**Step-2**



**Then Unzip the folder=> right click and extract it=>open this folder and copy inside the folder**



**Copy the folder and paste into “C” drive**



**Similarly, we need to setup MAVEN\_HOME AND MAVEN Path as same as JAVA setup.**

**Set up the MAVEN\_HOME in system Environmental Variable .**

**Copy the Path of maven in C drive.**

1. **MAVEN\_HOME = C:\apache-maven-3.8.8**

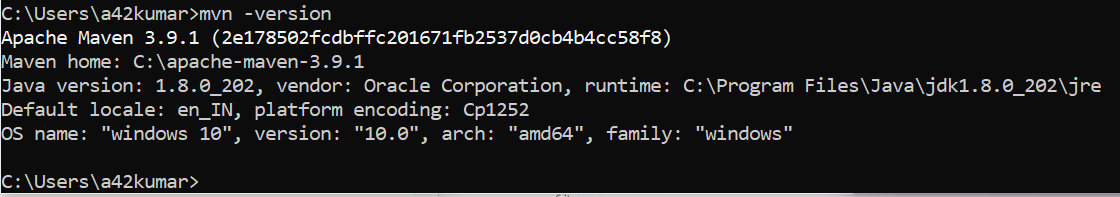
**Setup the path for maven**

**System Environmental 🡺 Click Path 🡺 Edit 🡺 copy the maven bin path**

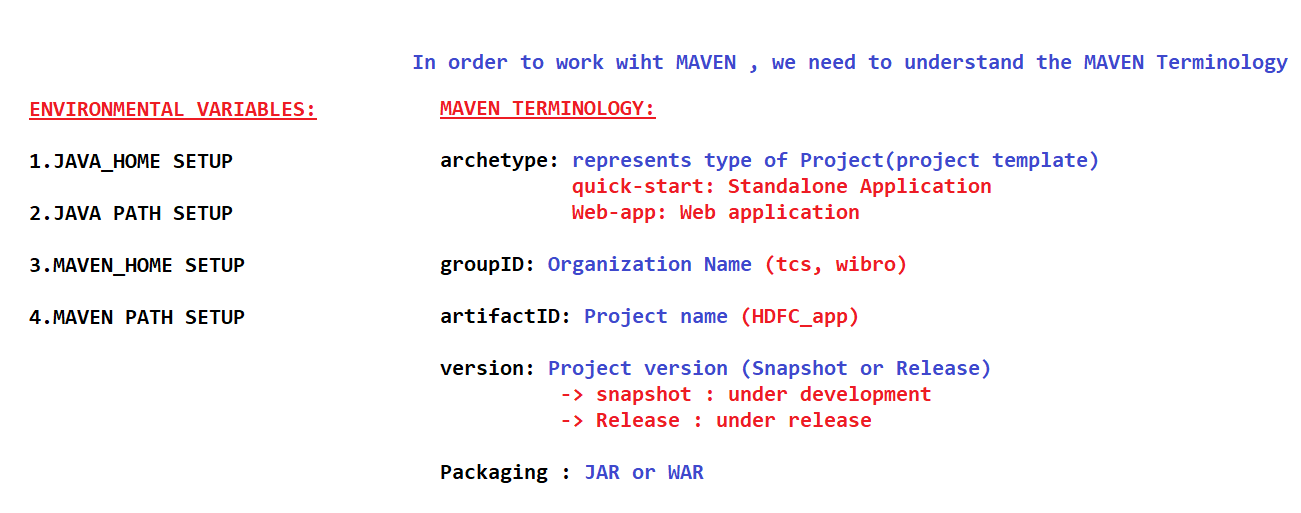
1. **C:\apache-maven-3.8.8\bin**

**Go to CMD Prompt and confirm the Maven Version.**

**$ mvn -version**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

****

**In order to Work with MAVEN, we need to understand Maven Terminology**

**Maven Terminology:**

**Archetype: Represent type of the project (project template)**

**Ex: Standalone , Web application**

**Quick-Start : standalone**

**Web-app: web application**

**Group Id: Organization Name (which company that company name we will mention)**

**Ex: greens it**

**Artifact id: (Project name)**

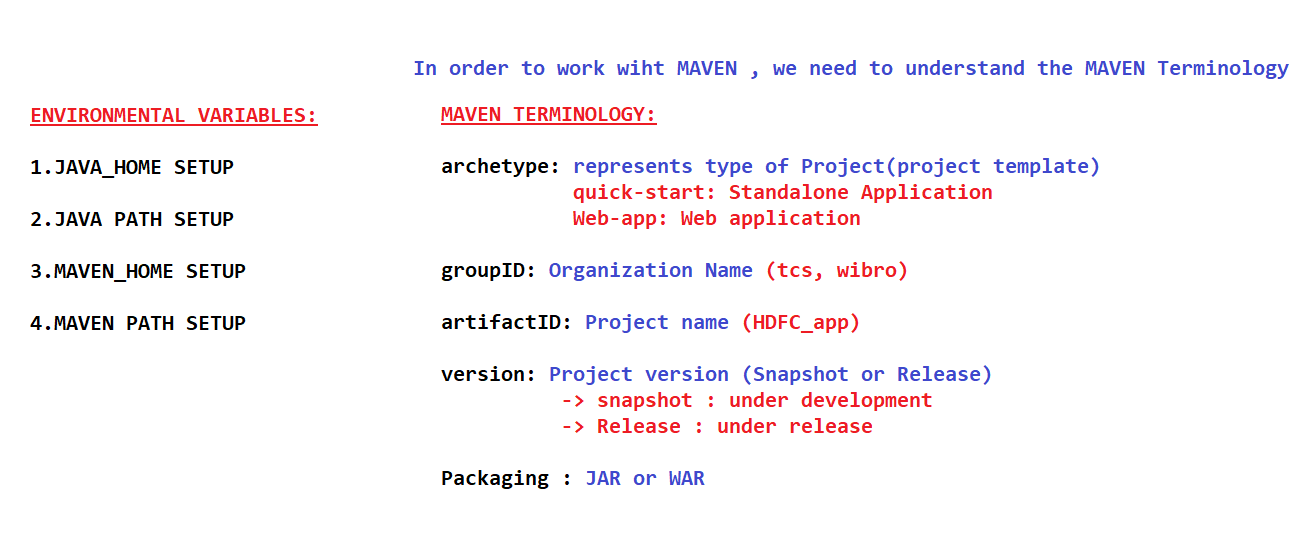
**Ex: sbi.app**

**Version: Project Version**

**(snapshot : under development)**

**(Release: under release)**

**Packaging: jar or war (how we want to package our application)**



**Demo For Maven:**

1. **Create one folder.**
2. **Open command prompt from that folder.**
3. **Execute below command to create Maven Project.**

**$mvn archetype:generate -DgroupId=in.greensit -DartifactId=01-Maven-App -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false**

**(-D system properties which is input for the command.)**

**Go to folder => click on the path and cmd=> command prompt will open**

**$ mvn archetype:generate**

**Then it will ask details one by one , that’s interactive mode**

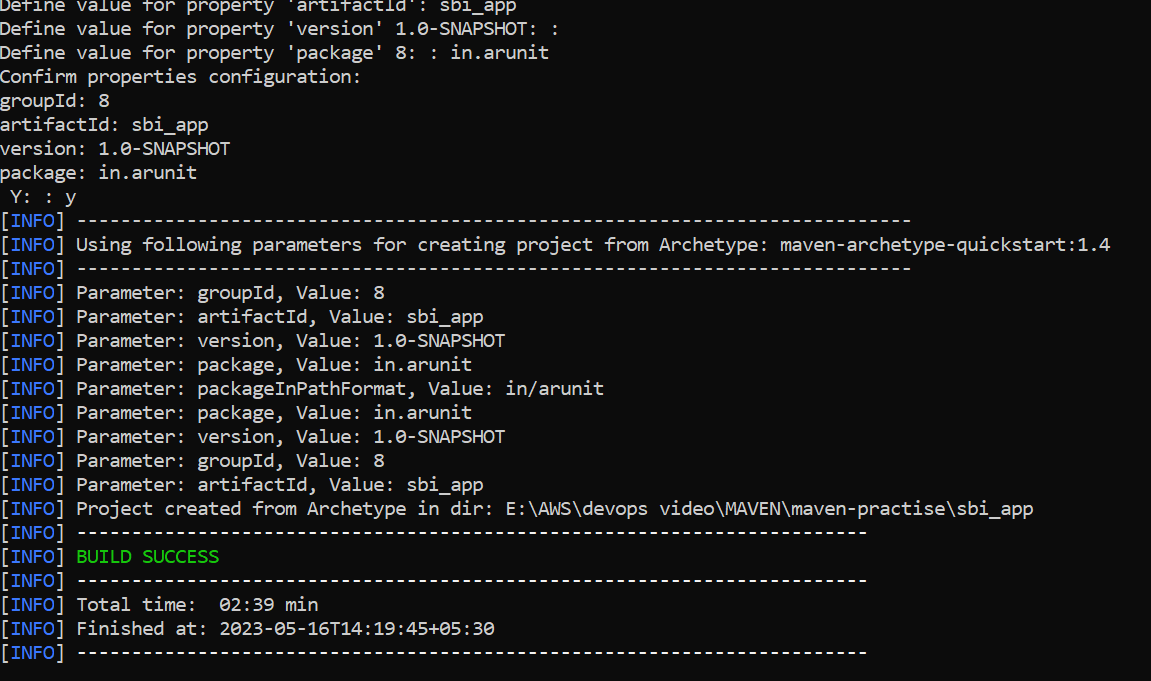
**Then if you see the folder project will create inside the folder.**

**Instead of creating the interactive mode we are going to create the project in batch mode**

**Batch mode: all the required details in the command itself**

**Execute the below command in CMD**

**$mvn archetype:generate -DgroupId=in.greensit -DartifactId=01-Maven-App -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false**



**4.Once Project created Verify Project folder structure:**

**-01.Maven-App**

**- src**

**- main**

**-java**

**- test**

**- java**

**- pom.xml**

**src/main/java: Application source code (.java files)**

**src/test/java: Application unit test (.java files)**

**pom.xml: Project object Model (Maven configuration files)**

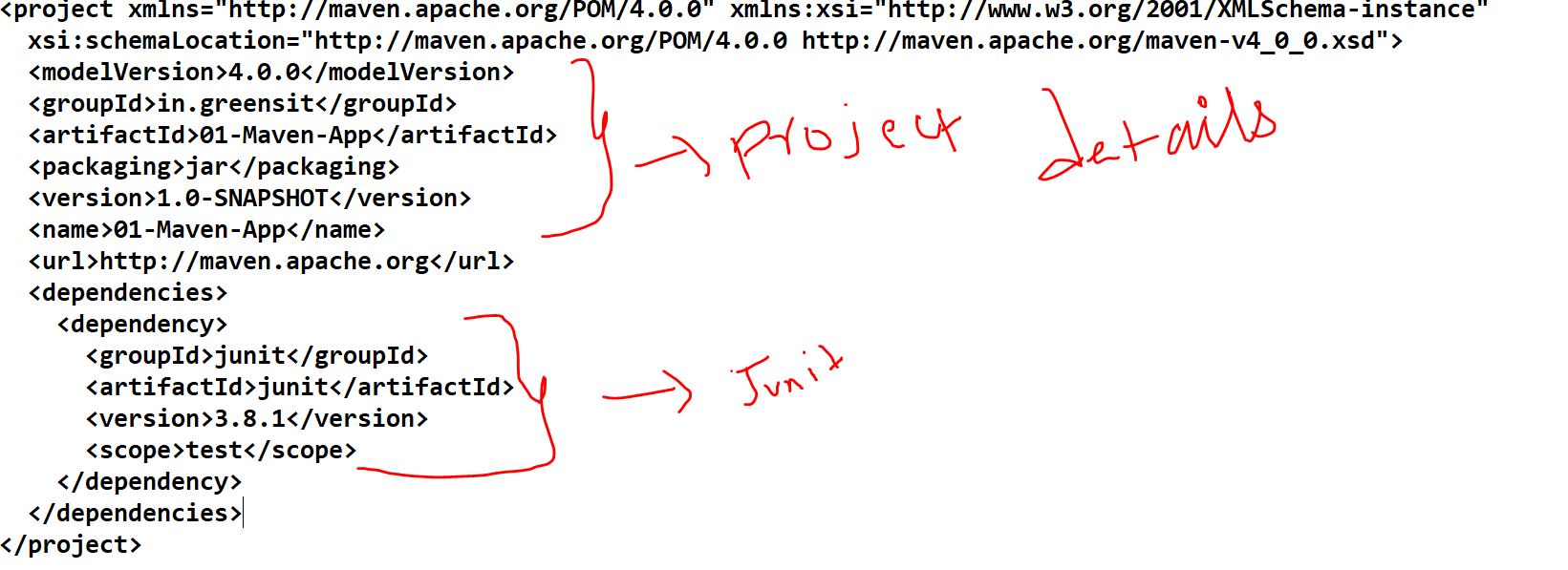
**Inside => main file => hello world java file is created**

**Inside => test file => testing file is created**

**POM.xml=All project details are available**

**By default, it configure dependencies also , that is called Junit. Whatever maven project will create Junit as a dependency.**

**Junit is one framework which is used for unit testing for java applications.  
 Java project will use junit for unit test  
 developers are going to add other dependencies also in POM.XML**



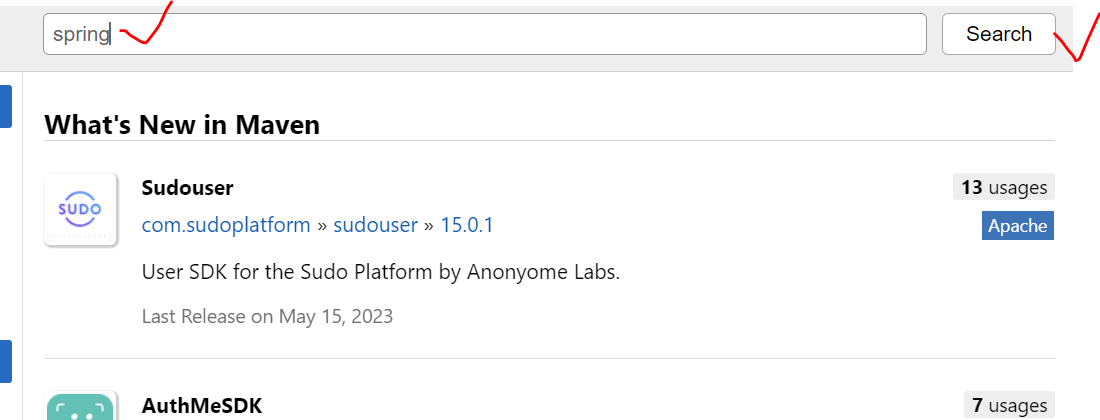
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**5.Adding Dependencies in POM.xml.**

**6.Go to MVN repository.com**

**<https://mvnrepository.com/> (We can find the dependencies)**

**Example: search spring and get dependencies**



**7.Adding Dependencies like this below**

**<dependency>**

**<groupId>org.springframework</groupId>**

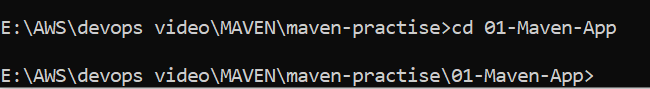
**<artifactId>spring-context</artifactId>**

**<version>6.0.9</version>**

**</dependency>**

**After adding dependencies . we can compile the project.**

**By cmd prompt we have to go inside the project folder .**



**Then execute the command (maven goals)**

**$ mvn compile ( it will compile src code into byte code/ .java files into .class file)**

**Then target folder will be created . inside the folder .class file will be available.**

**$ mvn clean ( when we execute this mvn goal target folder will be deleted)**

**Note: In real time in src code folder multiple files .java files are available , then maven it will compile , it will convert the byte code multiple .class files will generate.**

**After that we need to package the .class files.**

**$ mvn package (In order to do the package, there is goal called mvn package)**

**Then go inside the target folder we able to see .jar file .**

**We can take this file for deployment .**

**(Maven is compiling our project and maven is package our project)**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**MVN Goals:**

**What is purpose of MVN goals?**

**-> Maven goals are used to performs build operations**

**Goals are, Clean, Compile , test , Package , install**

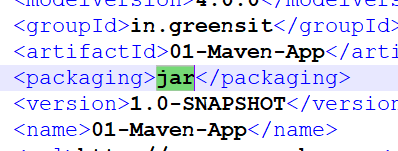
**->For every goal there is plugin which performs actual build operations.**

**Clean: delete the target folder**

**Compile: compile source code and store into the target folder.**

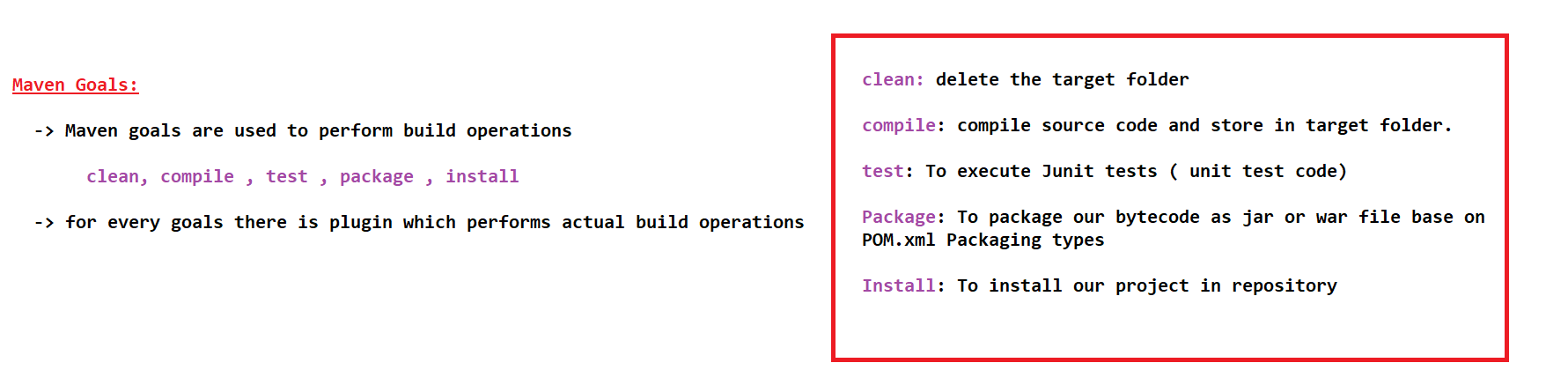
**Test: To execute Junit test (unit test code will be executed) (developers are writing the test code).**

**Package: it is used to package our byte code as JAR or WAR file based on POM.xml packaging type. (based an archetype it will understand standalone or web application)**



**Note: When we execute the Package goal it will internally COMPILE+TEST+PACKAGE**

**Install: To install our project in repository.**

****

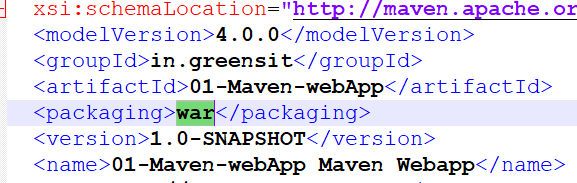
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Once again, we are going to create the project as .war file**

**Note: The folder Structure will be differed as standalone and web-application project.**

**$ mvn archetype:generate -DgroupId=in.greensit -DartifactId=01-Maven-webApp -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false**

**And open the pom.xml . we able to view war file**



**Inside the project folder**

**$ mvn compile ( target folder will come)**

**$ mvn package (.war will execute inside the target folder)**

**When we execute package goals , which will execute internally three goals**

**Package = compile + test + package**

**$ mvn clean package ( this command will execute delete , compile , test , package)**

**Everything will be deleted and freshly it will compile and package the code.**

**It will be deleted previous package and re-create the new package ( compile + package)**

**Maven Repositories:**

* **Repositories means collection of libraries**
* **Maven having three types of Repositories**

**1.Local Repositories**

**2.Central Repositories**

**3.Remote Repositories**

* **Local Repositories will be created in our system. Whenever we create the maven project , the maven will generate one local repositories in our system.**

**Local Repositories can connect with Central Repositories as well as Remote Repositories.**

**Our Project will always connect with the local repositories.(Maven APP)**

**Within that Maven App , POM.xml also available**

* **Central repositories maintaining by Apache**
* **Remote Repo we can create using NEXUX/JFROG**

**NOTE: IT companies definitely will use REMOT REPOSITORIES**

**What is meaning of compiling?**

**Source code will be converted to byte code**

**Target folder will be created🡺 inside the target folder 🡺 ./class file is created**

**$ mvn clean**

**target folder will be deleted.**

**What is purpose by using mvn clean?**

**Whatever source code available in SRC folder that source code will be converted into byte code.**

**What is there inside the target?**

**Class file is available.**

**Maven is compiling our project and packaging our project.**

**We can create folder structure by using maven**

**Download Required dependencies based on Porm.xml configuration**

**Source code compilation**

**Project Packaging ( jar or war)**

**In order to download the dependencies, we need to configure dependencies details in the pom.xml**

**In order to compiling the project . mvn compile will use.**

**Maven goals are available to perform mvn operations.**

**After this compilation is completed, we able to package.**

**===============================================================================**

**Maven Goals:**

* **Maven Goals are used to perform build operations**

**Clean , Compile , test , package , install …….**

* **For every goal there is a plugin which performs actual operations.**

**Clean: To delete the target Folder**

**Compile: Compile Source Code and store in the target Folder**

**Compile code is store in the target folder**

**Test: To execute Junit tests ( unit test code)**

**(developers are going to write code and they will write the code for testing also)**

**Package: To package is used to execute our byte code as jar or war file.**

**How maven will understand it stand alone or web-app, it refer the archetype**

**Install: To install our project in maven repository.**

**$ mvn clean package ( if we use this goal, it will delete existing and freshly compile the code).**