Maven:

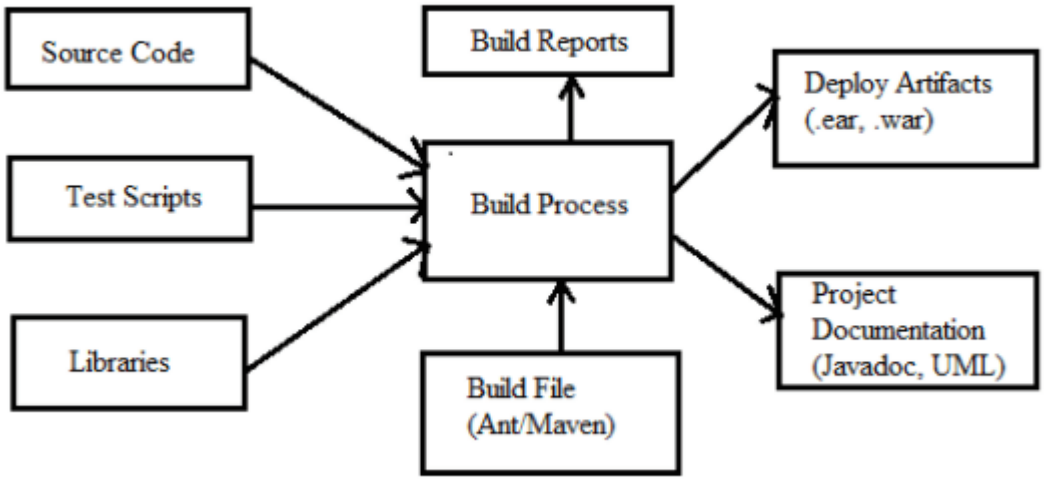
What is maven:

Maven is an open-source build tool that Software Developers use to build, publish, and deploy several projects at once. It's written in Java and is mainly used for Java-based projects, but you can also use it to build and manage projects in other languages like C#, Ruby, and Scala.

To better understand Maven and what it does, it's important to understand two key terms: build tools and Maven can also build any number of projects into desired output such as jar, war, metadata.

It was initially released on 13 July 2004. In the Yiddish language, the meaning of Maven is "accumulator of knowledge."

Maven Architecture:



What is a build tool?

A build tool compiles code and packages it into a form that can be executed. Dev teams rely on building tools to automate the process of turning source code into an actual application. A build tool typically does the following things:

- Create source code
- Create documentation from the source code
- Compile source code and bundle it into a package file format such as JAR.

What does Maven do?

Maven helps developers manage multiple projects at once by simplifying and automati
the project lifecycle.

The default Maven lifecycle phases are:

- Validate that the project information is correct and available.
- Compile source code.
- Test using an appropriate testing framework.
- Package code into a JAR file.
- Verify that quality assurance criteria are met.
- Install packages into a local repository that other projects can use.
- Deploy copies into a remote repository for access by other developers and projects.

Maven used

We can easily build a project using maven.

We can add jars and other dependencies of the project easily using the help of maven.

Maven provides project information (log document, dependency list, unit test reports, etc.)

Maven is very helpful for a project while updating the central repository of JARs and other dependencies.

With the help of Maven, we can build any number of projects into output types like the JAR, WAR, etc without doing any scripting.

Using maven we can easily integrate our project with a source control systems (such as Subversion or Git).

Maven also helps in managing the project's build lifecycle, including tasks like compiling, testin packaging, and deploying the code. (log document, dependency list, unit test reports, etc.)

How to Install Maven?

The installation of Maven includes the following Steps:

Verify that your system has java installed or not. if not then install java (Link for Java Installation)

Check java Environmental variable is set or not. if not then set java environmental variable.(link to install java and setting environmental variable)

Download maven (Link)

Unpack your maven zip at any place in your system.

Add the bin directory of the created directory apache-maven-3.5.3(it depends upon your installation version) to the PATH environment variable and system variable.

open cmd and run mvn -v command. If it print following lines of code then installation completed

First u check it in your system java is there or not

Download java and install :

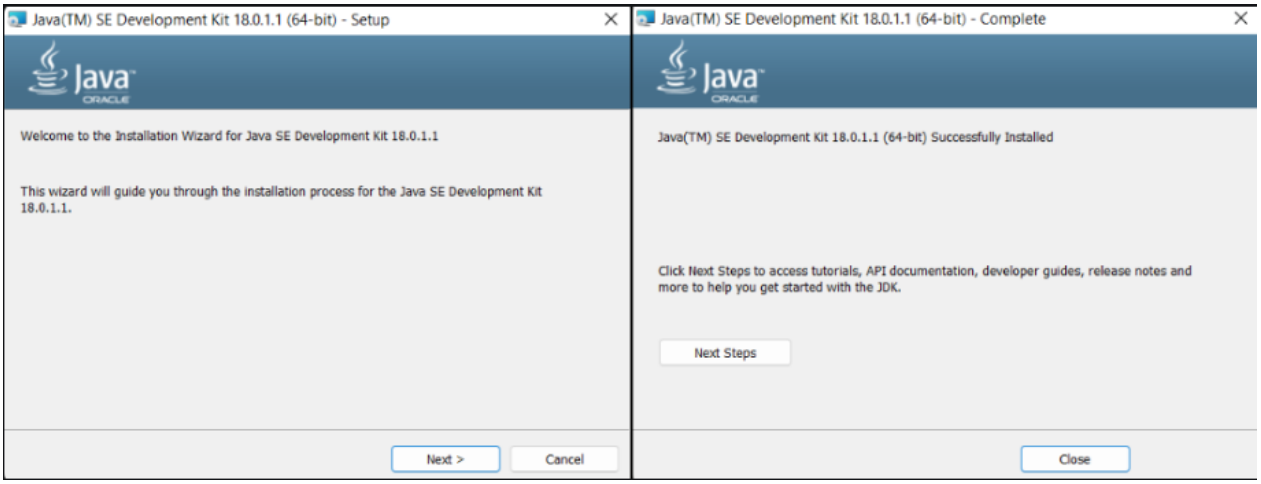
Create oracle account https://profile.oracle.com/myprofile/account/create-account.jspx

Jdk path : https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html

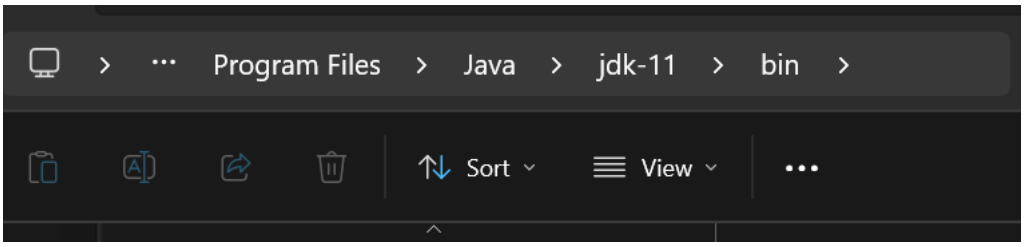
Then u choose which version u needs and download it (windows or mac)

Windows x86	201.64 MB	jdk-8u202-windows-i586.exe
Windows x64	211.58 MB	jdk-8u202-windows-x64.exe

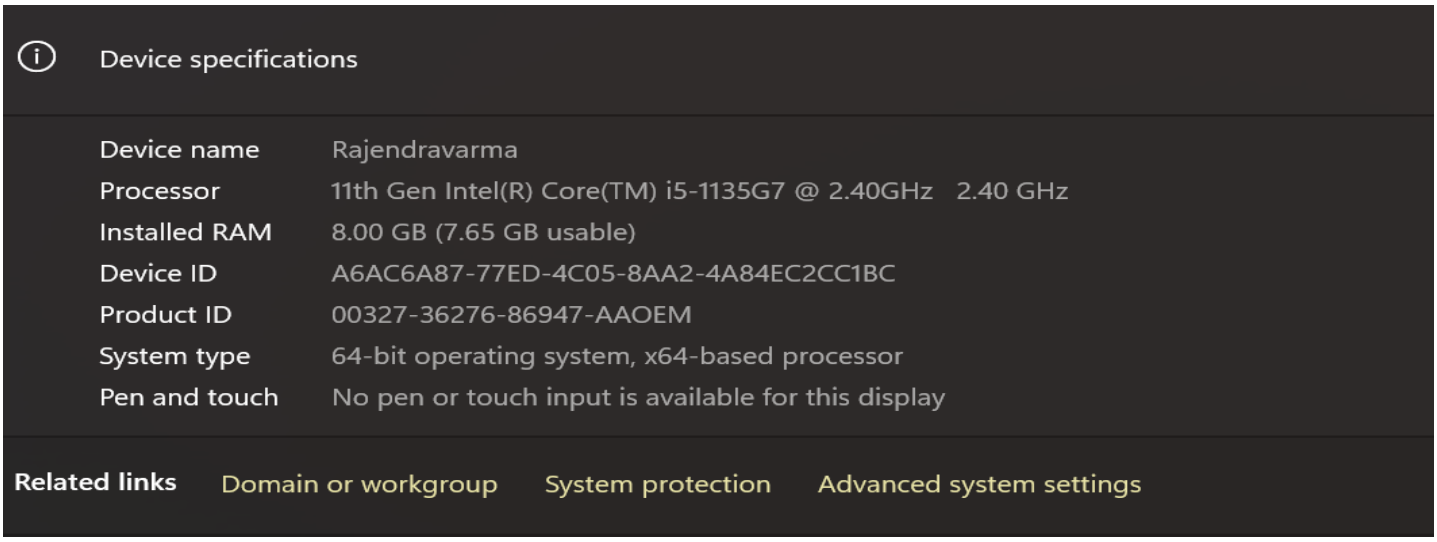
then download u install :



After installation Goto c drive then click on program files -go to java file open it -click on bin-copy that path



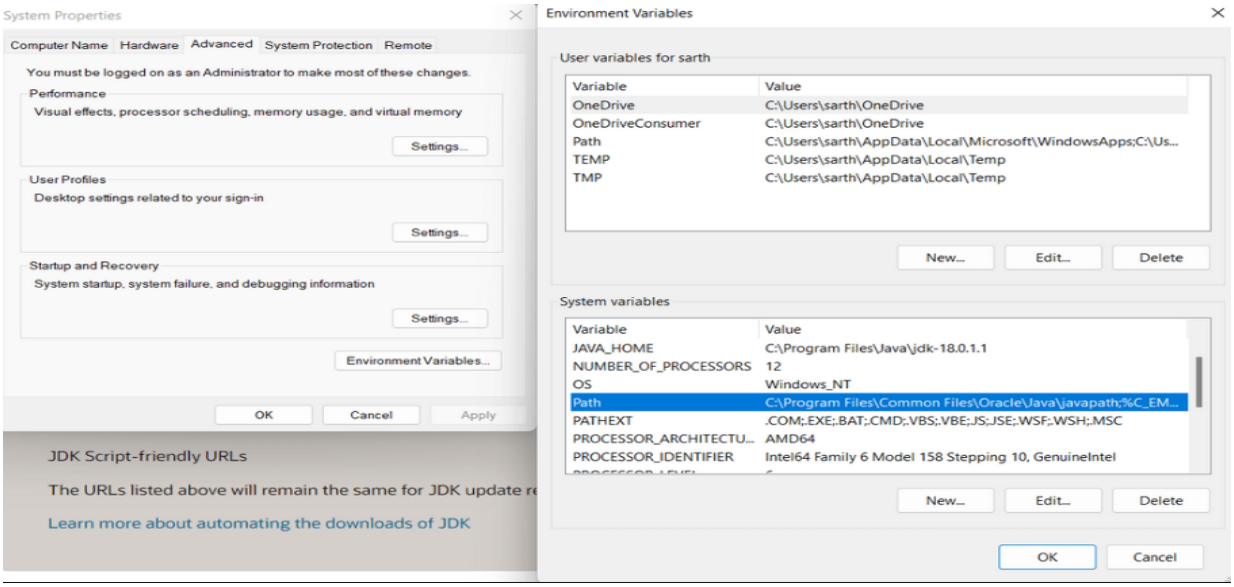
Goto System variables settings --- click on about pc -click on advance system settings



Click on advanced option – click on environment variables -ok

U see different paths tab open

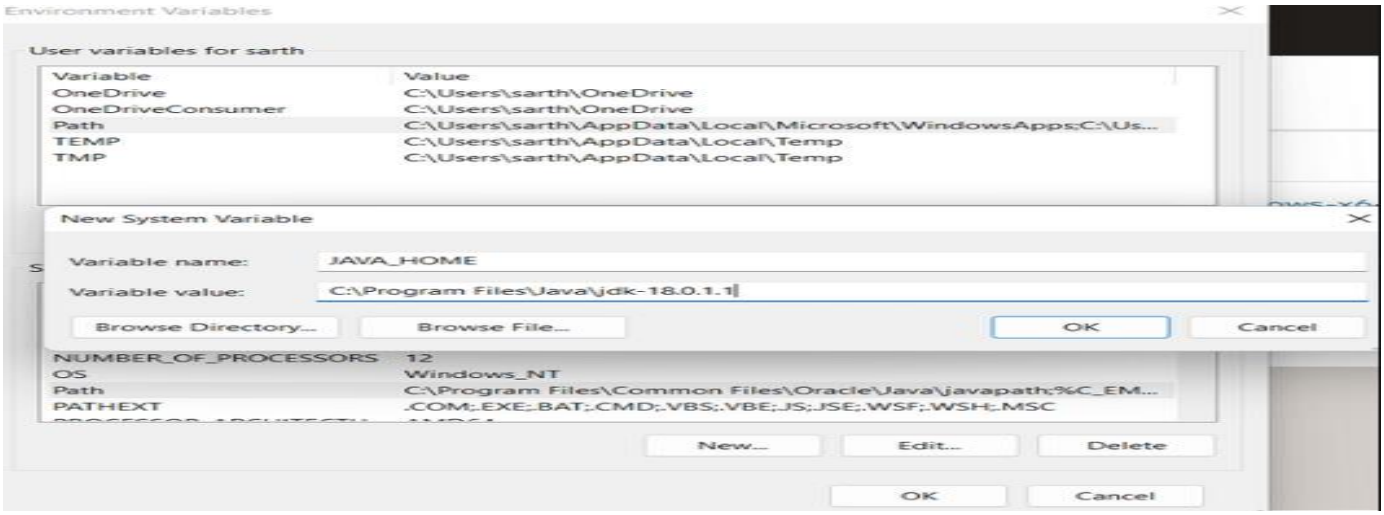
Goto system variable -path click - click on new option



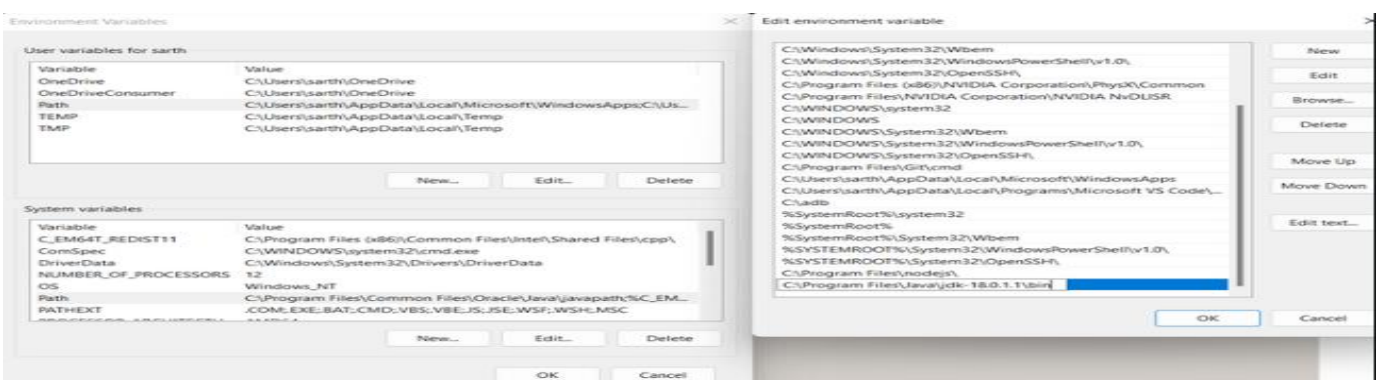
Set the path Test and variable value

The value u already copied here paste it

Ex: C:\Program Files\Java\jdk-11\bin ---- remove \bin and click ok



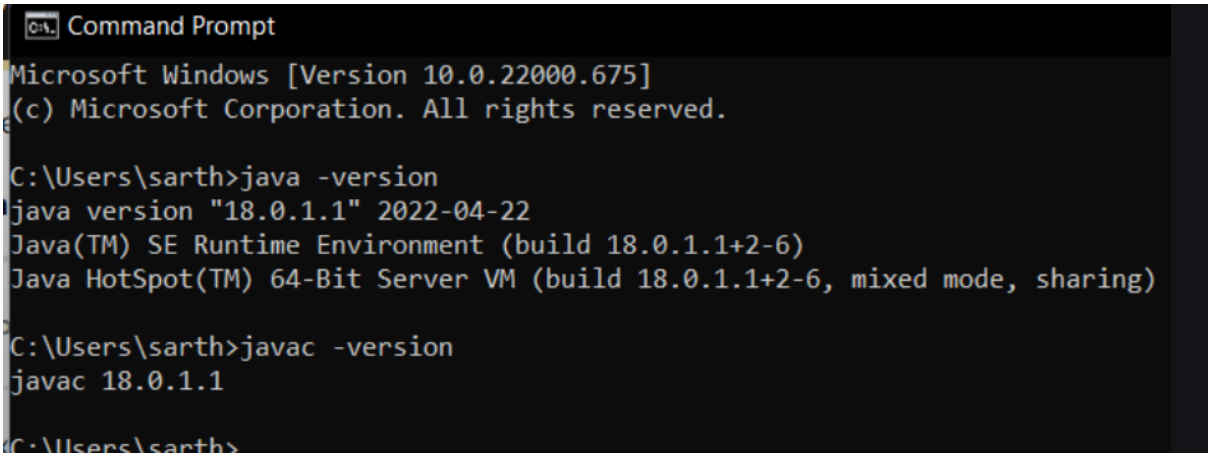
Again click on path now click on edit option



Click on new option -- click empty row – paste it java path and save it .

Ex: C:\Program Files\Java\jdk-11\bin

Check it java is running or not – open command prompt – enter java –version



Maven installation:

Maven link: https://dlcdn.apache.org/maven/maven-3/3.9.6/binaries/apache-maven-3.9.6-bin.zip

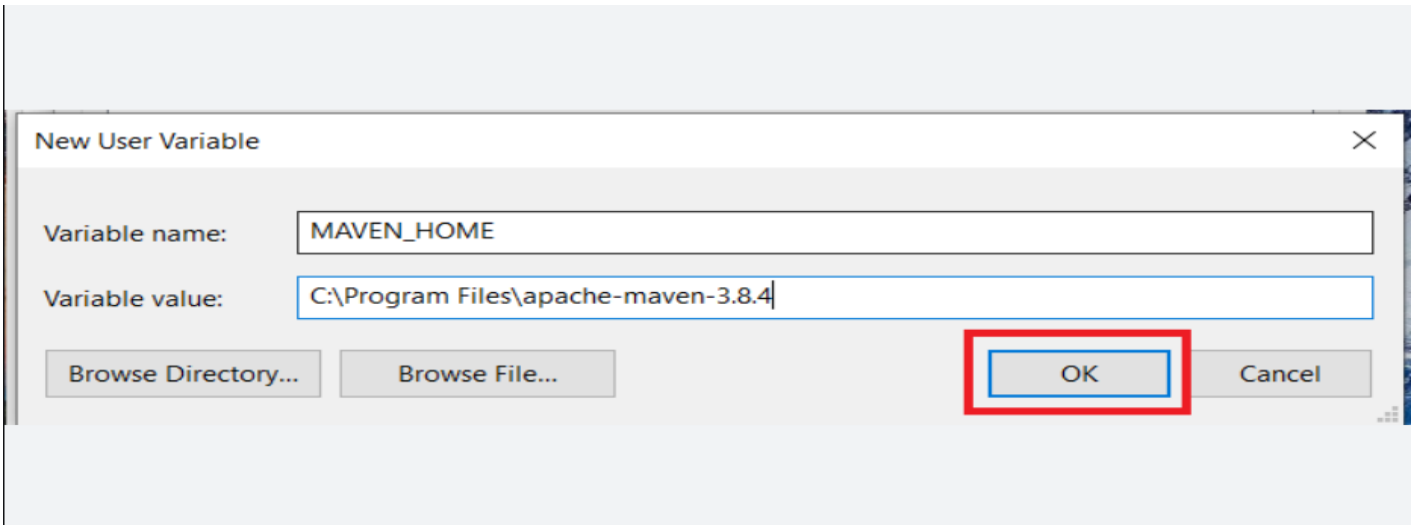
After download maven go to downloads extract that maven zip file .

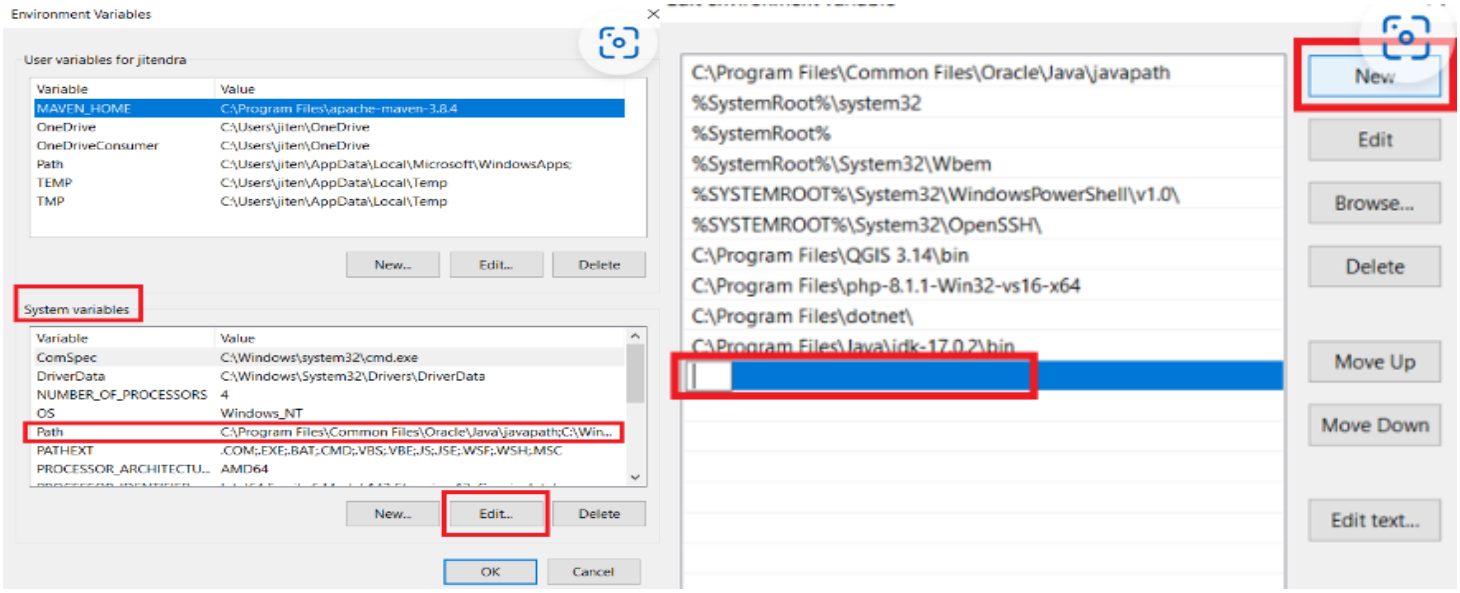
That extract file copy -- go to c drive – click on program file – paste in program file folder

Open maven fille click om bin --- copy that path

Same do it java installation processes as well here but replace name MAVEN_HOME .

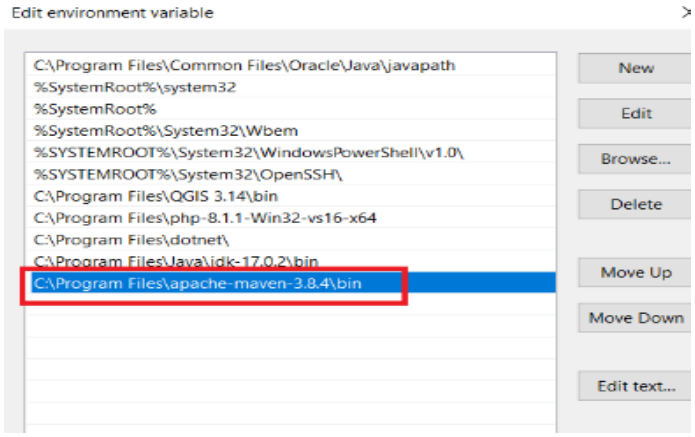
And also paste maven path also .





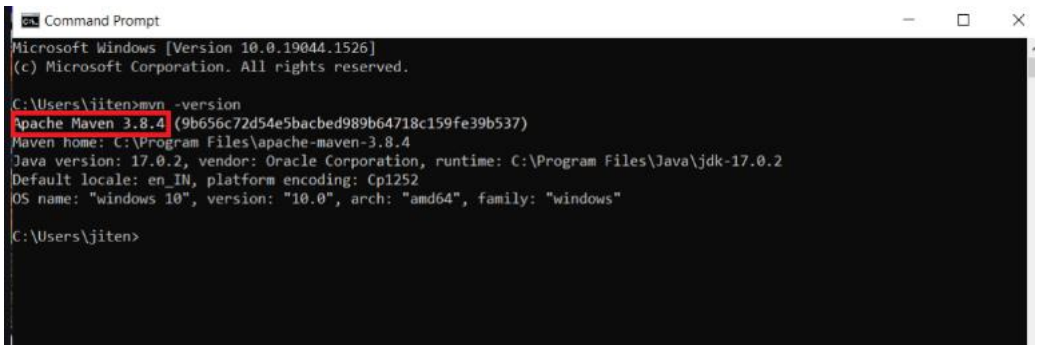
Paste maven path full and click ok

Click in all ok ok ok ….



Now check it run or not open command prompt

Enter command : mvn -version (or) mvn -v

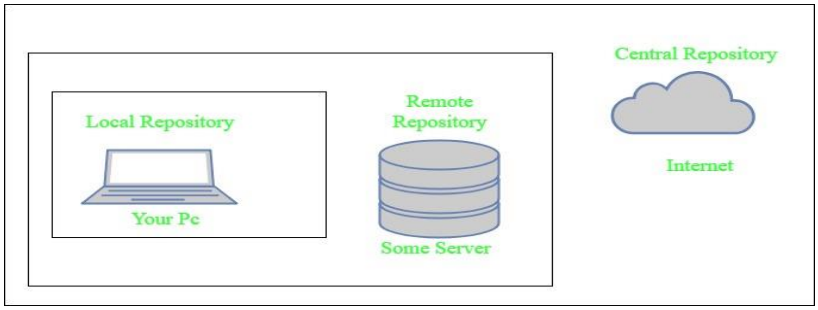


How to run the maven :

1. Create a folder in desktop --open it (or) were are project folder open gitbash
2. Right Click -- gitbash click it
3. Enter mvn archetype :generate

```
Administrator@himabindu MINGW64 ~/Desktop/MAVEN-OPERATIONS
$ mvn archetype:generate
```

Downloaded all plugins were we got that plugins we have 3 repos.



Once u enter mvn archetype:generate downloaded all plugins, dependency from central repository

```
3403: remote -> xyz.luan.generator:xyz-gae-generator (-)
3404: remote -> xyz.luan.generator:xyz-generator (-)
3405: remote -> za.co.absa.hyperdrive:component-archetype (-)
3406: remote -> za.co.absa.hyperdrive:component-archetype_2.11 (-)
3407: remote -> za.co.absa.hyperdrive:component-archetype_2.12 (-)
Choose a number or apply filter (format: [groupId:]artifactId, case s
ensitive contains): 2131: 3
Choose am.ik.archetype:graalvm-springmvc-blank-archetype version:
```

Artifactory id means your organization Id ,

Now we enter randomly dependency number I enter :3

Next choose version number :1 (I choose) .enter

```
chetype/graalvm-springmvc-blank-archetype/0.1.0/graalvm-springmvc
nk-archetype-0.1.0.jar (54 kB at 207 kB/s)
efine value for property 'groupId': abcdorg
efine value for property 'artifactId': abcdorg
efine value for property 'version' 1.0-SNAPSHOT: : 1.0.0
efine value for property 'package' abcdorg: : war
```

Give group id :abcdorg { I ill show only sample how to run& work maven }

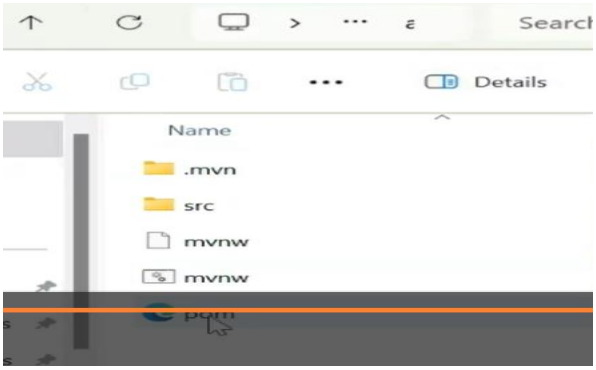
Artifactory :abcdorg

Version : 1.0.0

package : war

After enter u see build success and u see in your folder

pom is created



```
-OPERATIONS\abcdorg\src\main\resources\application.properties
[INFO] Project created from Archetype in dir: C:\Users\Admini
Desktop\MAVEN-OPERATIONS\abcdorg
[INFO] ------------------------------------------------------
---------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------
---------
[INFO] Total time:  03:00 min
[INFO] Finished at: 2024-04-05T08:41:17+05:30
[INFO] ------------------------------------------------------
```

If u validate code go to in side pom file folder in cmd , then enter mvn validate

```
Administrator@himabindu MINGW64 ~/Desktop/MAVEN-OPERATIONS
$ ls
abcdorg/

Administrator@himabindu MINGW64 ~/Desktop/MAVEN-OPERATIONS
$ cd abcdorg/

Administrator@himabindu MINGW64 ~/Desktop/MAVEN-OPERATIONS/abcdorg
$ ls
mvnw*  mvnw.cmd  pom.xml  src/

Administrator@himabindu MINGW64 ~/Desktop/MAVEN-OPERATIONS/abcdorg
$ mvn validate
```

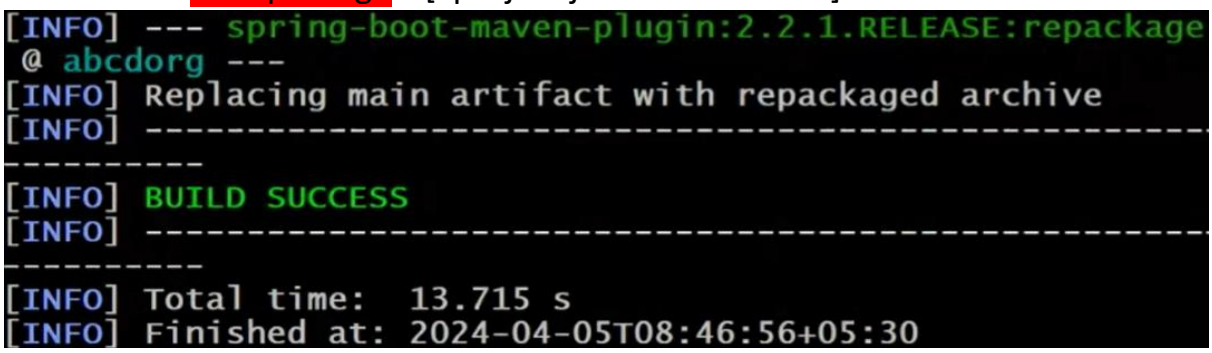
After u enter mvn compile command (u build your code here downloaded particularproject pom plugins. Build success …[u check it in desktop project folder in side target folder created].

```
EN-OPERATIONS\abcdorg\target\classes
[INFO] ------------------------------------------
---------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------
---------
[INFO] Total time:  01:14 min
[INFO] Finished at: 2024-04-05T08:45:57+05:30
[INFO] ------------------------------------------
```

After mvn test command [u see in test clases folder also created in project file]

```
running for 3.487)
INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time e
 3.046 s - in war.AppTest
INFO] Running war.GreetingControllerTest
INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time e
 0.325 s - in war.GreetingControllerTest
024-04-05 08:46:30.016  INFO 12068 --- [extShutdownHook] o.s.s
rent.ThreadPoolTaskExecutor  : Shutting down ExecutorService '
tionTaskExecutor'
INFO]
INFO] Results:
INFO]
INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
INFO]
INFO] ------------------------------------------------------
---------
INFO] BUILD SUCCESS
INFO] ------------------------------------------------------
```

After enter Mvn package [project jar file is created]



Here u can't deploy because we can't give source and testing ……it ll see Jenkins and jfrog practice time it can deploy ……..