Setting JAVA\_HOME path in MAC OS X

vim ~/.bash\_profile

export JAVA\_HOME=$(/usr/libexec/java\_home)

Hit Esc and then :wq  and then ENTER

source ~/.bash\_profile

echo $JAVA\_HOME

8. Install java in Windows

1. Google ->JDK download -> click on [Java Downloads | Oracle](https://www.bing.com/ck/a?!&&p=bbb311e5f1bcb8b9JmltdHM9MTY5NTY4NjQwMCZpZ3VpZD0xY2JkNWE3MS04MGQ1LTZhM2YtM2NmZS00OTQyODE3ODZiYzUmaW5zaWQ9NTIxNg&ptn=3&hsh=3&fclid=1cbd5a71-80d5-6a3f-3cfe-494281786bc5&psq=jdk+download&u=a1aHR0cHM6Ly93d3cub3JhY2xlLmNvbS9qYXZhL3RlY2hub2xvZ2llcy9kb3dubG9hZHMv&ntb=1) ->see which jdk is the latest long term support -> windows -> (x64 MSI Installer) – download it ->Open file ->next ->next -> done
2. Below left side search ->Edit the system environment variables -> click on environment variables ->in system variable create new variable -> new -> Varaible name : JAVA\_HOME Variable Value : C:\Program Files\Java\jdk-21 ( Path )
3. In the system variable click on path varaible and edit and add this path ( C:\Program Files\Java\jdk-21\bin )

After successfully installed java And next you need to set this Java home path in your system variables.

So first, let's go ahead and see where Java is installed in our system.

Once you added the java home path then And also you need to give the path to this bin directory because this is the path where there are a

lot of executable files are there, so you need to set that as well.

So you might get a doubt why I am setting system variables.

Okay, you have a Java software, but when you start working in with your programming in Eclipse or

any other editor by default, it do not know where Java is sitting in your system.

But if you set your software home path in the environmental variables like this, any other software

which is in your system, if they want Java, they can directly refer to this environment variables

and find out where Java is.

So once you set, go to terminal and make sure Java is detected.

So Java hyphen version.

9. Install one editor ( Eclipse ) and create maven project

### Google -> Eclipse download -> click on eclipse org main link -> click on Download packages -> download windows ( [Eclipse IDE for Java Developers](https://www.eclipse.org/downloads/packages/release/2023-09/r/eclipse-ide-java-developers) ) ( Note : The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration )

### Open the file and extra it -> click on eclipse -> launch the eclipse

So in the eclipse folder, you will see the application launcher.

Just select that So you will see Eclipse Startup screen followed by workbench.

So here you need to select workspace.

So that means where are you creating your folders?

Where are you writing your scripts?

You need to give the path in your local system.

By default, Eclipse will create you one path asking you to start there so you can just keep it default

and say launch

click on create java project -> give project name -> And it says that it will come with one folder inside this project. ->click on finish

one java project is created with empty src ( source folder ) so here where you will go and write all your project files.

So in this project we actually have to write our selenium code,

So by default, this project do not have any knowledge about selenium.

We need to somehow give the knowledge of selenium jar selenium libraries to this project.

How do you do that?

So for that we will take the help of Maven.

high level Maven is one central repository for Java based projects.

So that means whoever developed one Java based library.

Now let's say Selenium guys have developed one library for Java

So they will host that in the Maven repositories.

So if somebody wants anything related to Java, they can go to their Maven official website and get

the selenium related library.

Testing is also one of the testing framework library built on Java.

So those people also have created one library and pushed it in Maven repository.

So basically this is the Maven repository.com.

This is the place where you will find libraries of all Java based projects.

So if you want to get download any repository, just blindly visit repository.com for Java and search

with selenium.

And that's it.

Here you will see a lot of different packages available for selenium.

What we need Selenium, Java.

Okay, so this is the relevant library which we have to use in our project.

So when you select this, you will come up with different versions here, select the whatever latest

version you see on the screen at the time of your execution.

Select that particular version.

So in this particular page, you have two different things.

One is jar files.

So the complete Selenium library is given in terms of jar files.

When you select view all here, you will see all the jars.

So basically you have to download these jars and place in your project.

That is one way of doing it.

And the next simple way.

So they are giving this four lines of code in Maven terminology we call it as a dependency.

Okay, so this is the selenium dependency.

If you place this four lines of code in your project, that's it.

So your project automatically scans this four lines of code and it detects that this is something coming

from Maven repository.com and it will automatically connect to Maven Repository and it will download

all these jars for you and it will push in the project.

All this work will be done automatically by Maven itself.

Group ID.

What is your org name?

Artifact id?

What is your project name?

With these two details, it will identify and the version you are giving right based upon the number

you provide that particular version it will get.

So right now this is simple Java project guys.

Now if you want to give Maven Knowledge this to this project by default Java Project will not have the Maven kind of philosophy.

So what you can do simply right click on your project and if you come here configure, you will see

something like this convert to Maven project.

So that means this Java project will be converted to Maven kind of project.

whenever you are trying to create Maven project, it will immediately ask what is the group

ID and artifact ID of your own project?

Similarly, earlier, when selenium people also trying to create Maven project there Also it asked the

details for selenium people.

Those people gave these details group ID, artifact ID, and once project is created they pushed it

to Maven Repository.

In the project name right click ->select configure -> Select convert maven project -> give group id, artifact id, version, ( these thing you will get it in the maven repository )select any of the packaging like jar ->click on finish

So this will convert your Java based project to again, maven based.

So the only difference what you see here that you will see one new file called Pom.xml.

This is the file where you want to give the details of your dependency to pull the selenium jars.

so once you convert, you will see this file and this is the heart of your entire maven project.

Just one line file is added for you.

Pom.xml by converting Java to Maven.

Pom.xml is added and that we need to get selenium dependencies.

Now you can see that dependency is singular,

So like this, your project may need multiple jars.

You are now getting selenium.

And then later I will show you how to generate reports for that.

Also, one library is there for logging, one library is there for assertions, one library is there.

There are a lot of dependencies.

You have to put it right.

So overall you have to call parent dependencies.

If you give plural like this.

So that means the set of dependencies going inside this parent dependencies here where you can paste

your selenium dependency.

You see this is singular.

These are child dependencies wrapped under main tag of dependencies.

Save your project and those jars now automatically will get into your project.

In that case, you see, there is one folder created

when you expand, here you go. You see all the selenium.

Four jars are now sitting in your project.

How did all this happen?

With simple one line of magic that you paste a dependency and saved your project.

So sometimes you might not see the jars coming down at that time.

What you can do.

So there are different options here.

Select project and select this option build automatically.

So that means whenever you update your pom.xml with some new code, then build this project.

Whenever this project is built, then it will connect to the repository and get the jars for you.

Make sure you check this checkbox so it will automatically do for you.

You have all the selenium jars with you and now this project have knowledge of selenium.

Now whatever code you write in Selenium, this project recognizes it.