**Maven:-**

Requirement of framework for testing:

* We should not manage external jars(dependencies) Manually it should be manage by framework itself
* Once we shared the project to team or PM or stakeholders , then the project should be in executable format
* Detailed analysis tool should be available after execution (report)
* Can be executed through different ways
* **Maintains should be easy and framework should be stable for multiple versions of build**

**Maven Project:-**

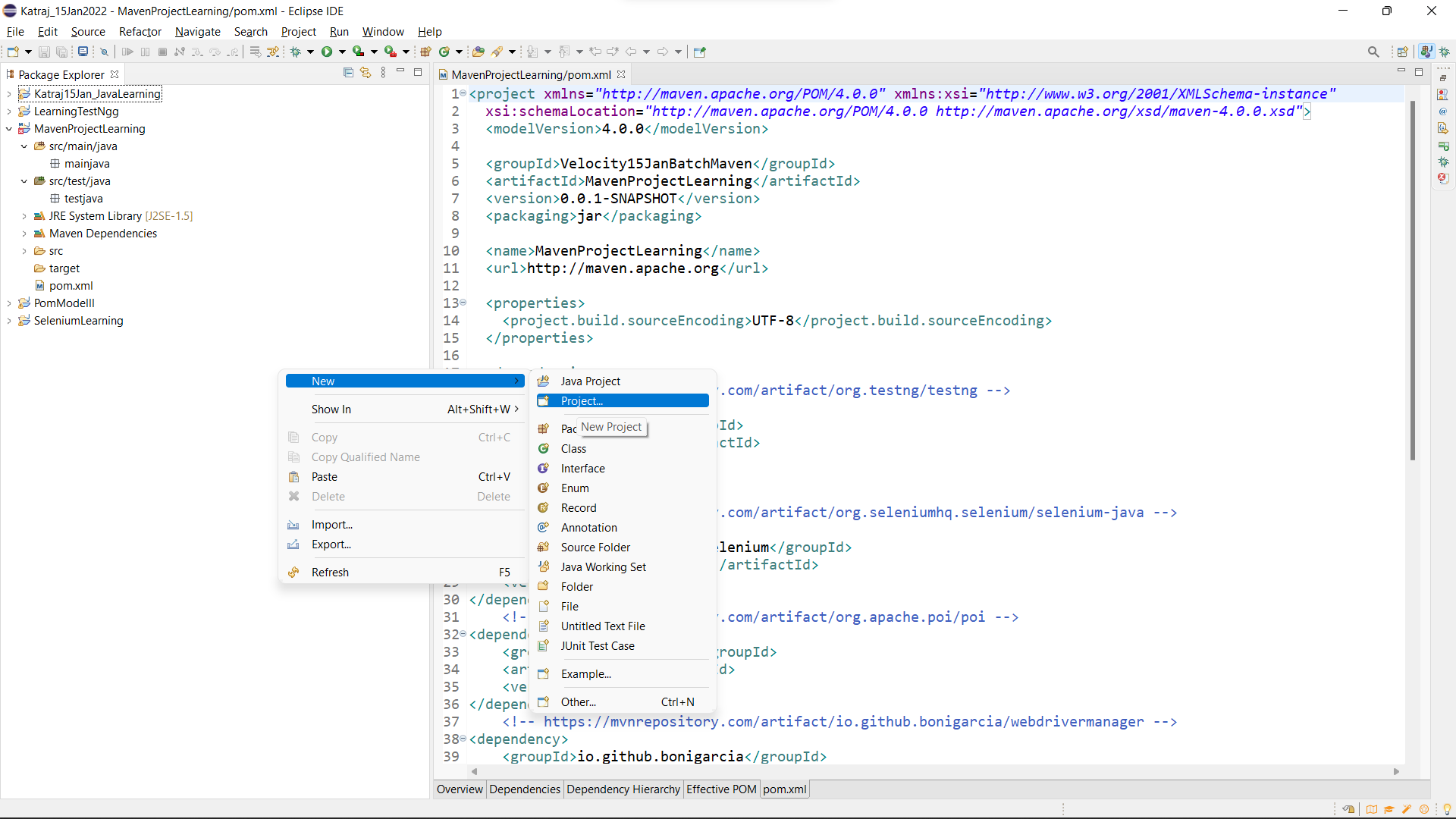
Maven- It is build automation tool which compile and build the project according to dependencies version available in the **pom.xml** file and execute the complete project

Pom.xml:-

* It contains dependencies for building the maven project
* Supports build of maven project

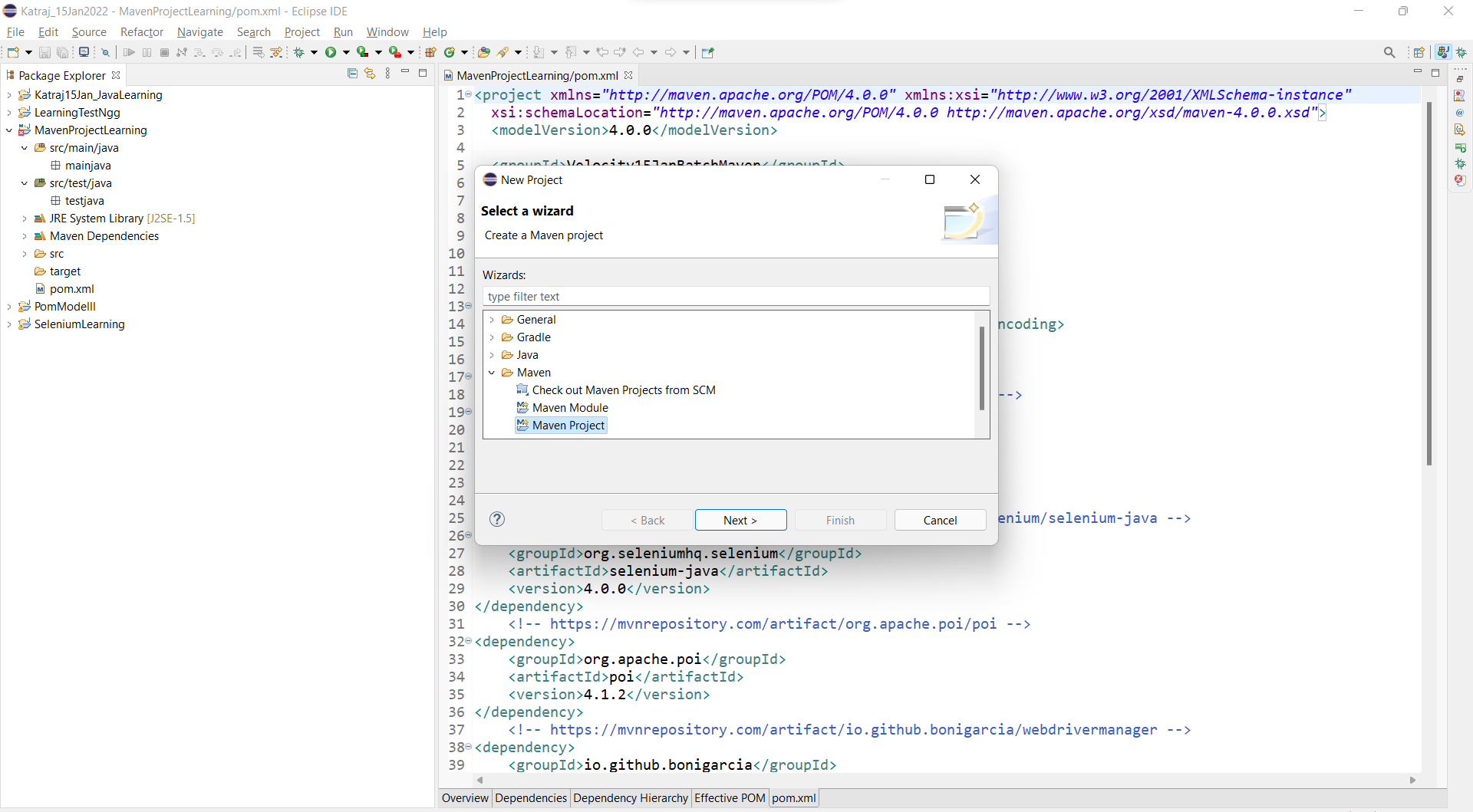
Creation of Maven:-

Step 1:-

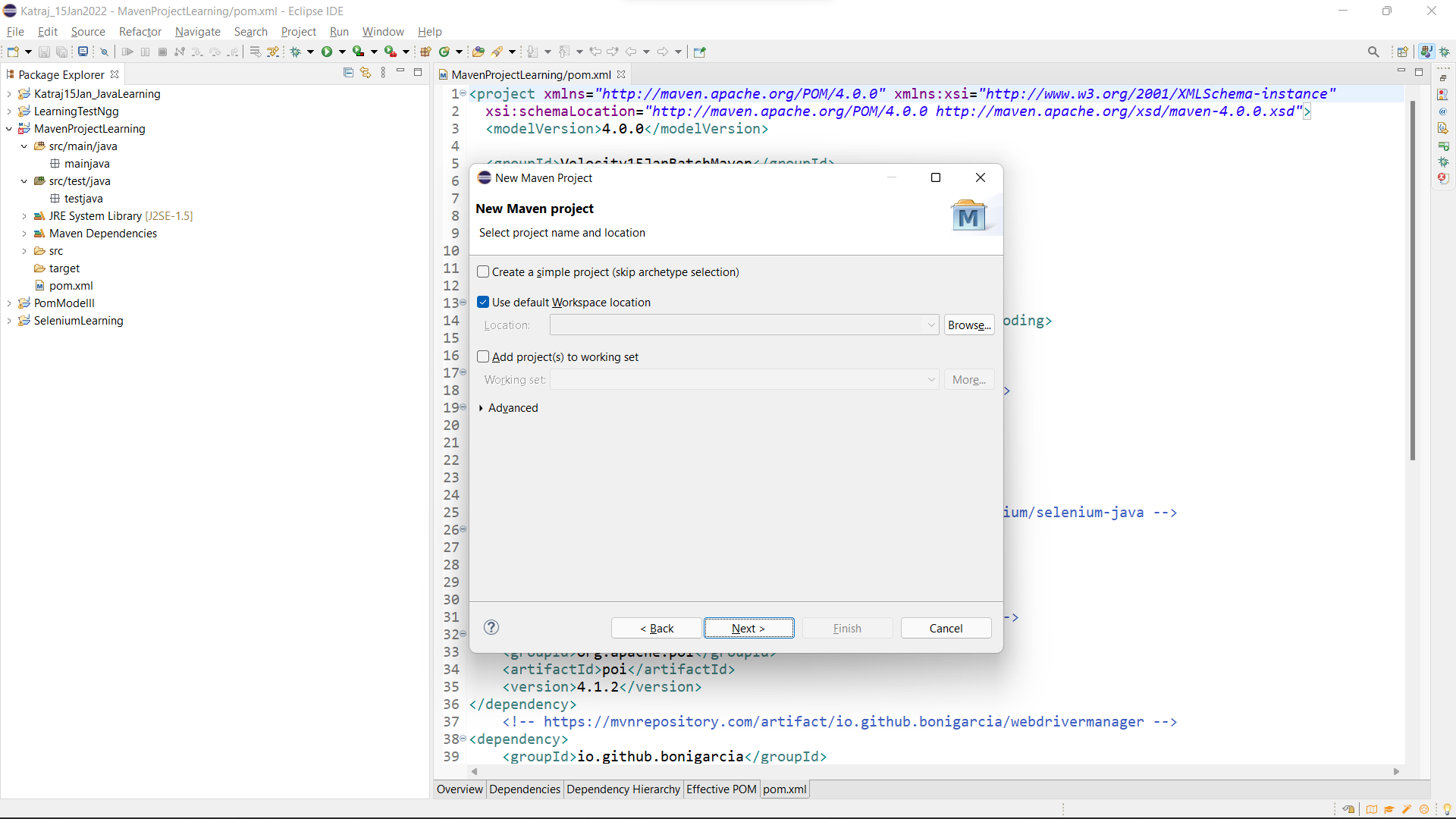


Select the project.

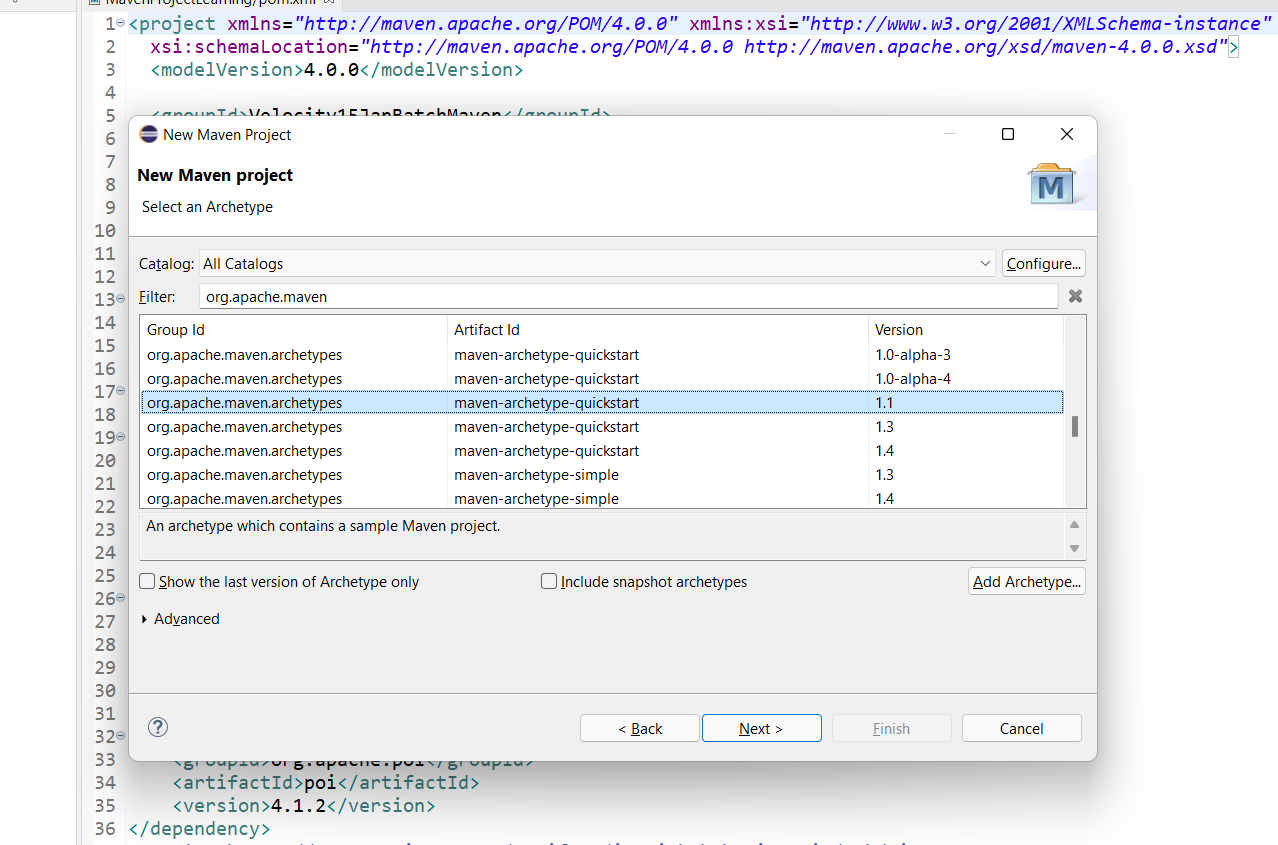
Step 2: Click on Maven and select Maven Project followed by click on the next button.



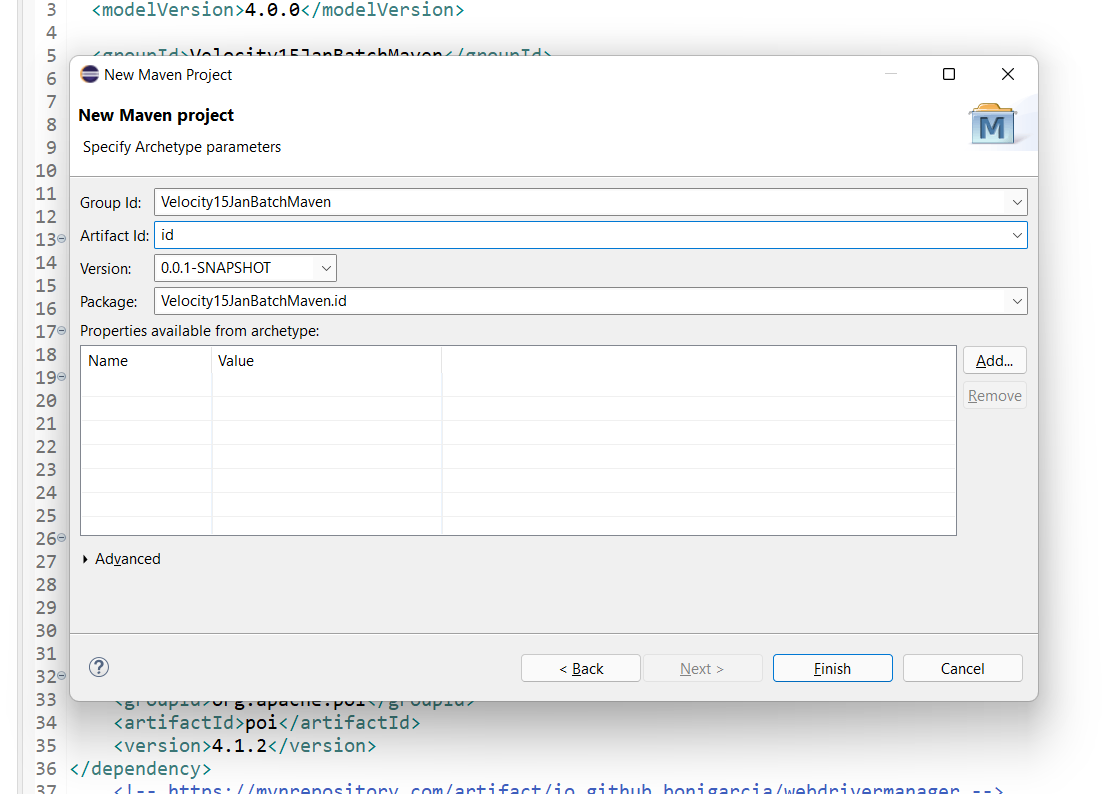
Click on Next and then again click on Next.



Step 3: Select the encircled type and click on the next button



Step 4: Provide Artifact ID and groupId then click on Finish button.



Step 5: Add all the dependencies followed by create packages inside src/main/java by the name of pages and create package testcases inside src/test/java.

Step 6: Write the code into them.

Step 7: To compile the code on java 8 we require **maven compiler** plugin and to configure our test execution we have to provide testng.xml file inside the pom.xml configuration by adding **maven surefire plugin.**

**To use maven compiler plugin copy the xml code from:**

[**https://maven.apache.org/plugins/maven-compiler-plugin/usage.html**](https://maven.apache.org/plugins/maven-compiler-plugin/usage.html)

**To use maven surefire plugin:**

[**https://maven.apache.org/surefire/maven-surefire-plugin/usage.html**](https://maven.apache.org/surefire/maven-surefire-plugin/usage.html)

Pom.xml:-



What is log4j?

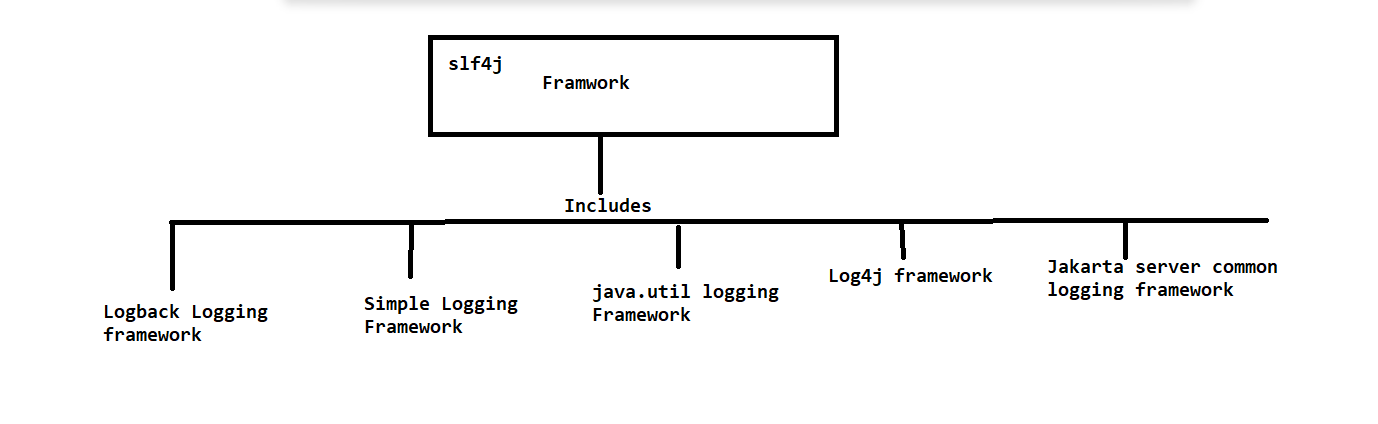
It is logging framework designed by apache, core engine base on java distributed under apache software license

It offers features like direct logging information to extent of variety destination such as database, file, console, UNIX, Syslog etc.

In recent year most of updates coming from log4j have faced massive vulnerability issue due to most of java application hampered a lot in terms of breaching of project, as well as project details and data base

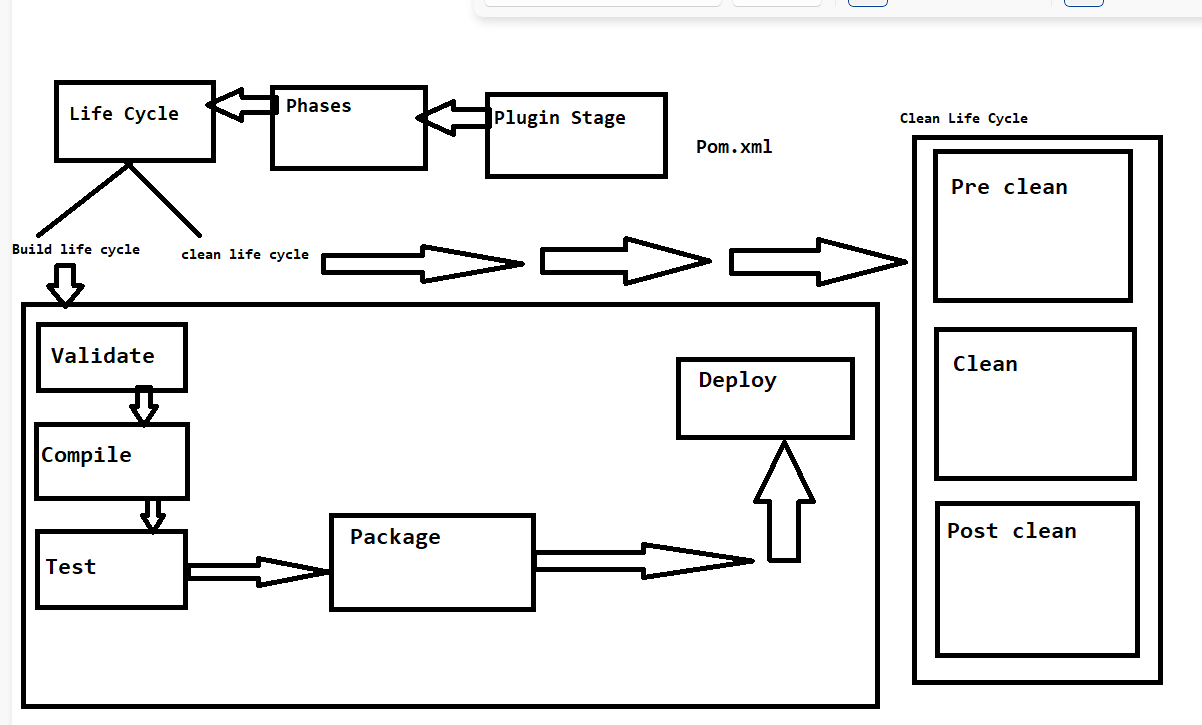
Latest version of log4j 2.17.2 is stable and updated version which is not vulnerable as of now 11May 2022, however in our project we have consider implementing slf4j plugin instead of log4j after facing several issues with versions of log4j

* Slf4j provides good stability as well as good security as compare to log4j
* Main difference between log4j and slf4j is,  
  log4j has implemented classes in framework howver slf4j has abstract classes in framework which gets implemented upon execution which help us to improve security of framework and data base



Maven Build Lifecycle:-

* Build Life cycle
* Clean Life cycle
* Site life cycle



Maven build lifecycle:

* **It is process for building and distributing a particular artifact (Project)**
* There are 3 built in lifecycle:

**Build , clean and site**

**Build** Lifecycle handles project deployment

**Clean** Lifecycle handles project cleaning (removing unwanted files)

**Site** Lifecycle handles the environment documentation creation of your site documentation and different phases of dependencies

**Build Lifecycle stage:-**

**Validation:-**

Validation performs the project correction and checks all necessary information is available or not

**Compiler:-**

Compiling the source code of the project into executable format

**Test:-**

Test the compiled source code using suitable unit testing framework

These test should not require the code be in packages or in deploy manner (Unit testing)

**Package**:-

Take compiled code and packaged it in its distributable format, such as JAR file

**Verify:-**

Run and check result (Package) of integration tests to ensure quality criteria (Integration testing)

**Install:-**

Install the package into local repository for use as dependencies other project locally

**Deploy:-**

Done in the build environment, copies the final reports and makes project ready to share

Finally local repo with jar file of project gets created which is in the distributable format which can be shared on git-hub so that team members would get access of project

**Plugin goals:-**

Plugin goals represents specific tasks which contributes to building and managing project

* Clean and package arguments are build phases which can be use as plugin goals
* Order of execution depends on the order in which the goals and build phase is provided
* **Maven project and POM with help of selenium, testing provides testing results that project desired and made of**
* **It provides good stability and easy maintains**
* **That’s the reason generally in selenium project integration of maven, TestNG, selenium and POM design is consider to be good practice**

**And therefor in our project we have implemented data driven framework with integration of maven, TestNG, selenium and POM design**

**What is Transitive and cyclic dependency?**