Lesson 01 Demo 07

Using Maven in Eclipse

Objective: To explore Maven in Eclipse by installing the software and executing the project in different ways

Tools required: Eclipse IDE and Maven

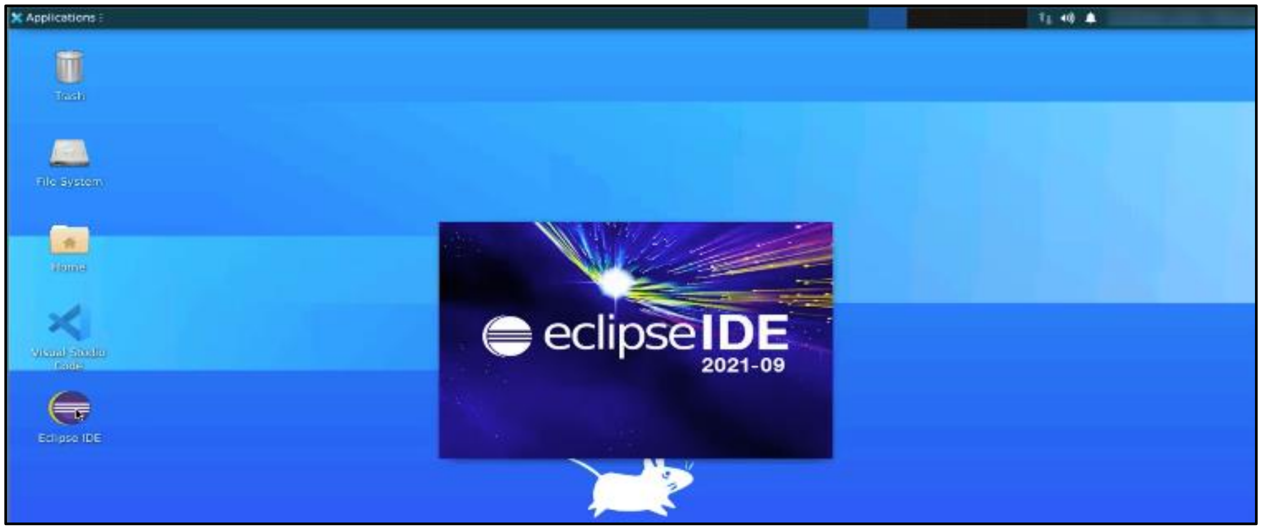
Prerequisites: None

Steps to be followed:

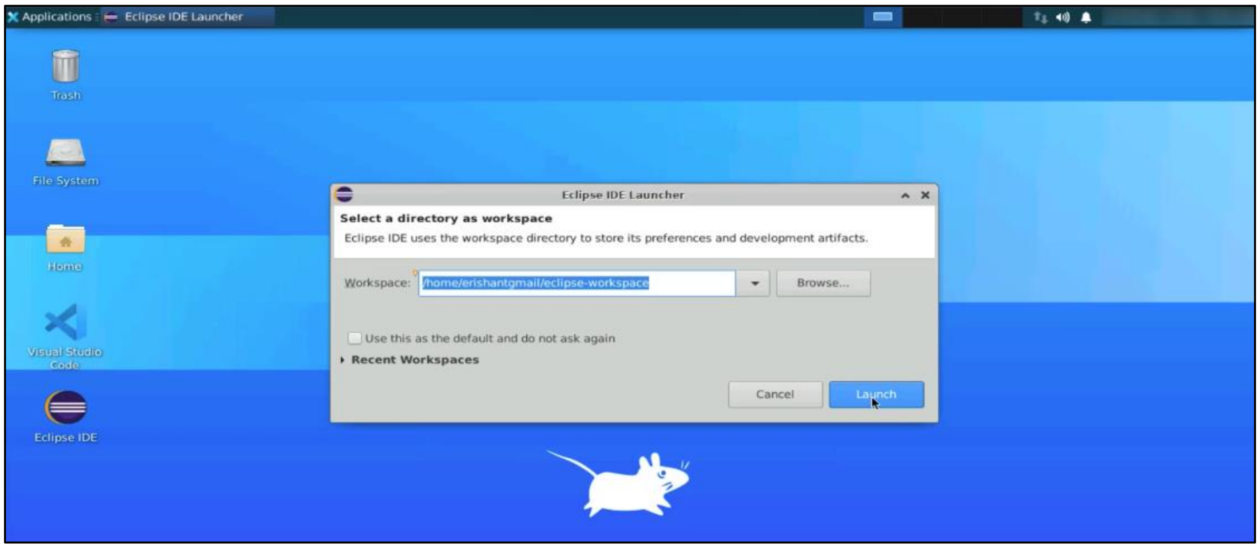
1. Install the software
2. Filter the Maven
3. Execute the Maven build
4. Run as Maven install
5. Create a Maven project
6. Add dependency in the pom.xml file

Step 1: Install the software

1.1 Launch the Eclipse IDE



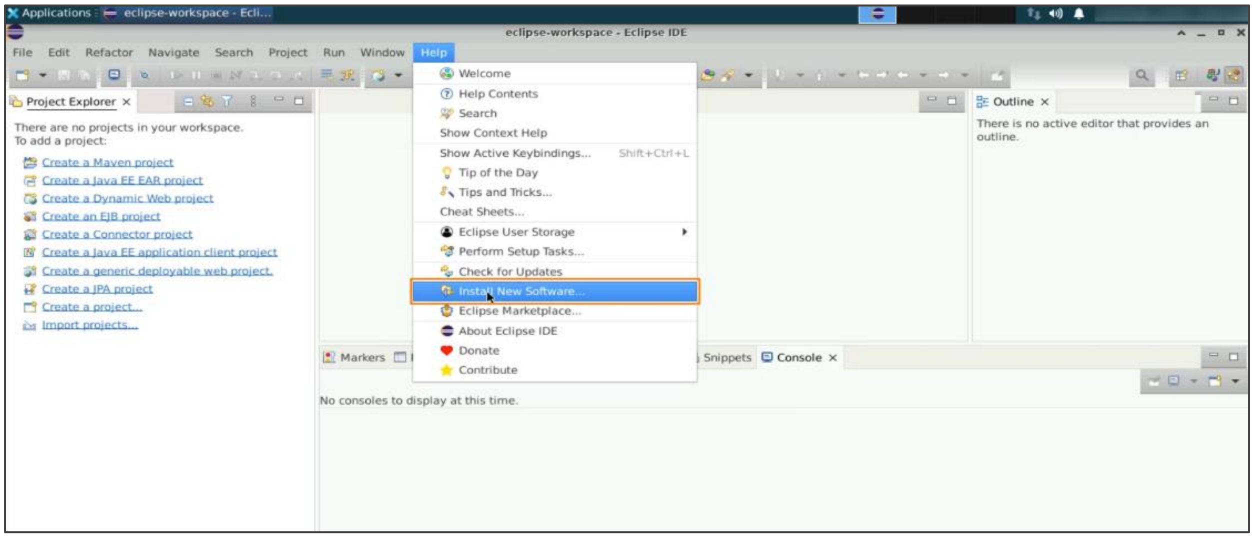
1.2 Add workspace location and click on the **Launch** button


Select a directory as workspace

Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

Workspace: /home/erishantgmail/eclipse-workspace

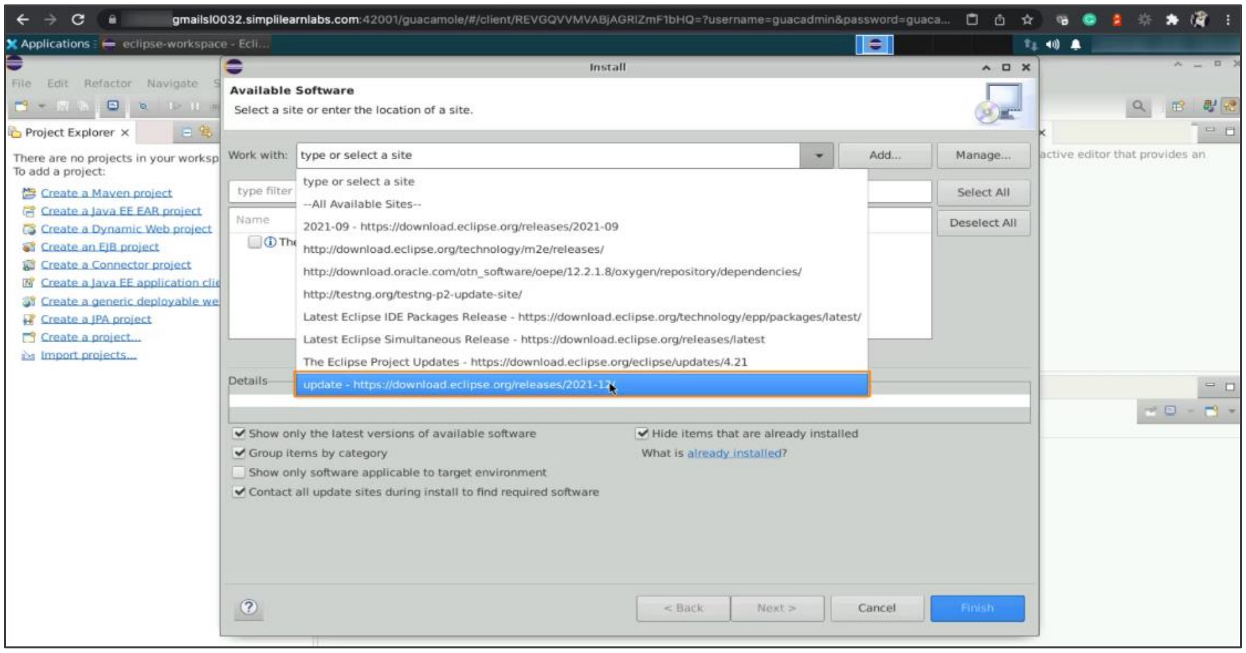
1.3 Go to **Help** and select **Install New Software**


There are no projects in your workspace.
To add a project:

There is no active editor that provides an outline.

No consoles to display at this time.

1.4 Choose the **update** option from the following options:



Available Software

Select a site or enter the location of a site.

Work with: type or select a site

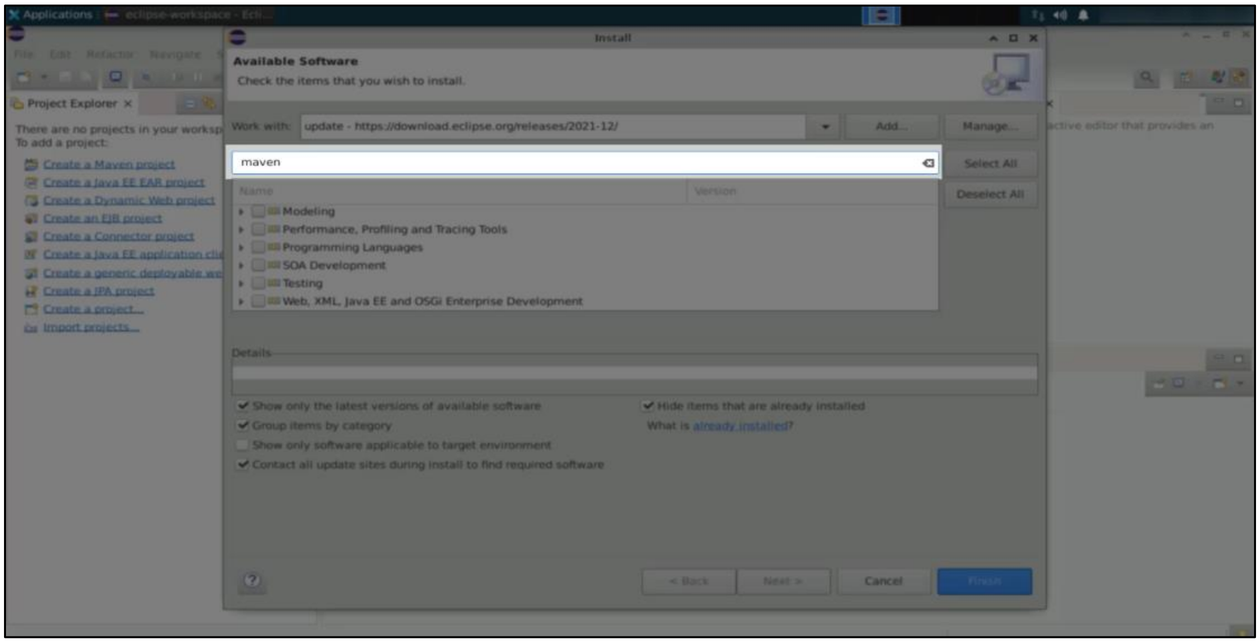
- type or select a site
- --All Available Sites--
- 2021-09 - https://download.eclipse.org/releases/2021-09
- http://download.eclipse.org/technology/m2e/releases/
- http://download.oracle.com/otn_software/oepe/12.2.1.8/oxygen/repository/dependencies/
- http://testng.org/testng-p2-update-site/
- Latest Eclipse IDE Packages Release - https://download.eclipse.org/technology/epp/packages/latest/
- Latest Eclipse Simultaneous Release - https://download.eclipse.org/releases/latest
- The Eclipse Project Updates - https://download.eclipse.org/eclipse/updates/4.21
- update - https://download.eclipse.org/releases/2021-12

- [✓] Show only the latest versions of available software
- [✓] Group items by category
- [ ] Show only software applicable to target environment
- [✓] Contact all update sites during install to find required software
- [✓] Hide items that are already installed

What is already installed?

Step 2: Filter the Maven

2.1 Search for **maven**


Install

Available Software

Check the items that you wish to install.

Work with: update - https://download.eclipse.org/releases/2021-12/

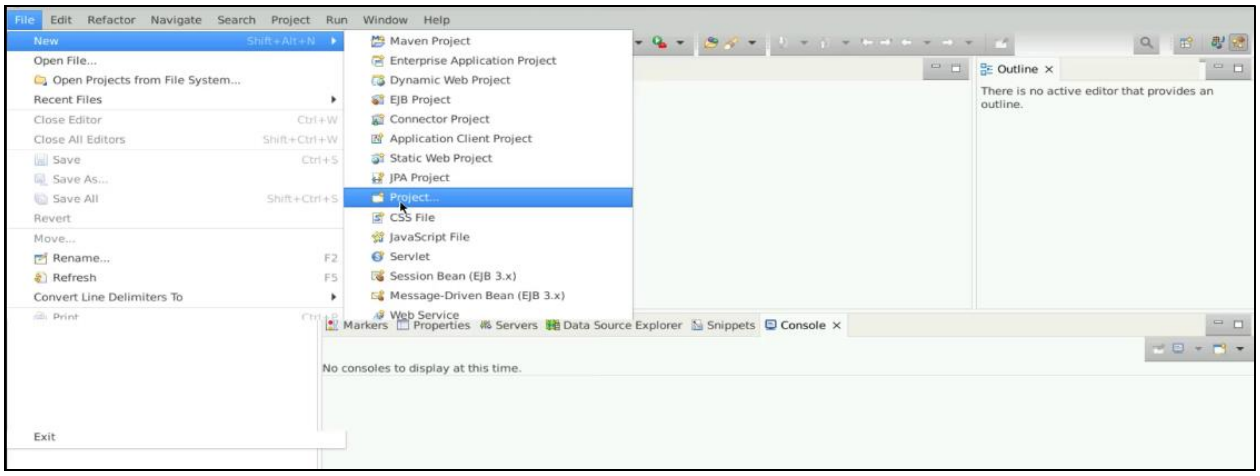
maven

- Modeling
- Performance, Profiling and Tracing Tools
- Programming Languages
- SOA Development
- Testing
- Web, XML, Java EE and OSGi Enterprise Development

- [✓] Show only the latest versions of available software
- [✓] Group items by category
- [ ] Show only software applicable to target environment
- [✓] Contact all update sites during install to find required software
- [✓] Hide items that are already installed

What is already installed?

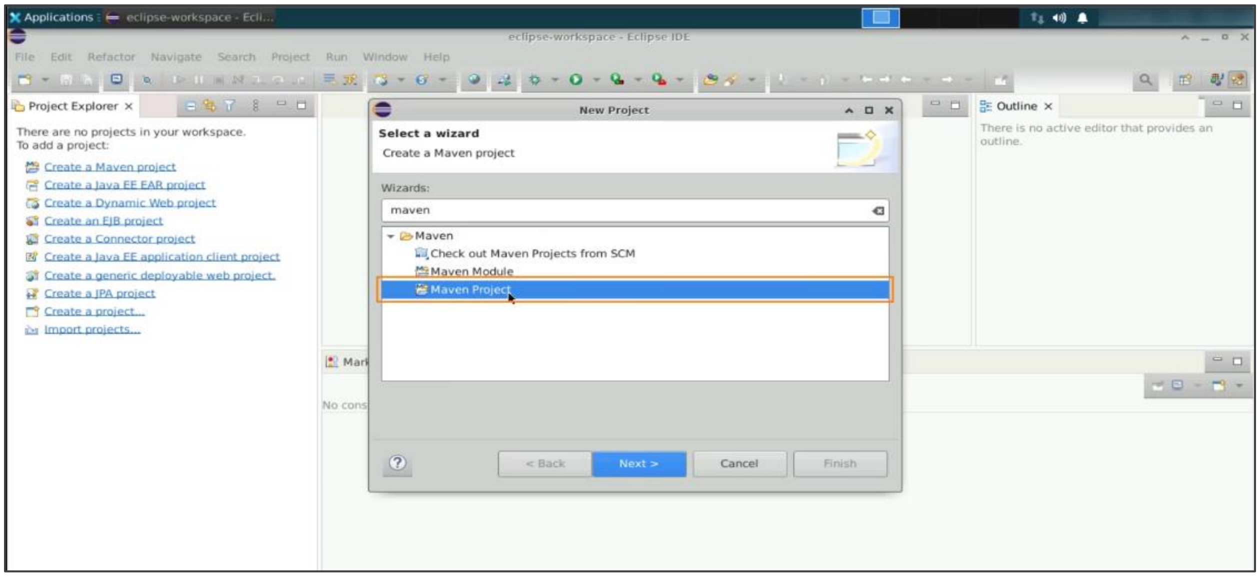
2.2 Create a Maven project**.** Go to **File**, select **New**, and then select **Project**.



There is no active editor that provides an outline.

No consoles to display at this time.

2.3 Type **maven** to filter. This will show a Maven Project.


Applications
eclipse-workspace - Ecli...
eclipse-workspace - Eclipse IDE
File Edit Refactor Navigate Search Project Run Window Help
Project Explorer
Outline
New Project
Wizards:
Maven
Check out Maven Projects from SCM
Maven Module
Maven Project
< Back
Next >
Cancel
Finish

There are no projects in your workspace.
To add a project:

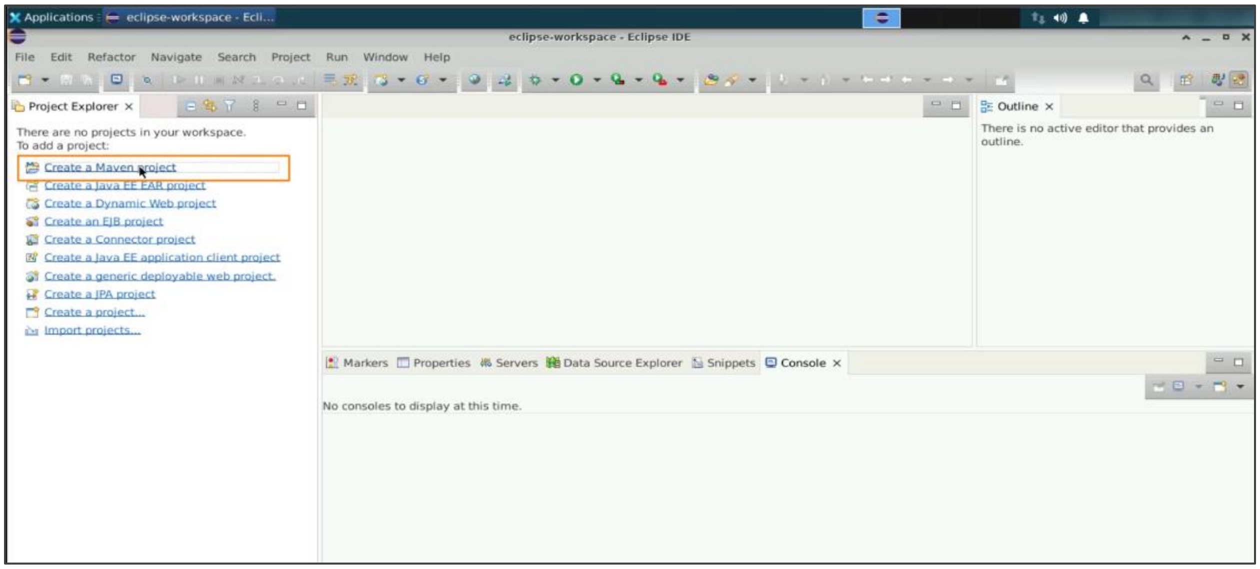
Create a Maven project
Create a Java EE EAR project
Create a Dynamic Web project
Create an EJB project
Create a Connector project
Create a Java EE application client project
Create a generic deployable web project.
Create a JPA project
Create a project...
Import projects...

Select a wizard
Create a Maven project

Wizards: maven

There is no active editor that provides an outline.

2.4 The other way is to go back and select **Create a Maven project**.


Applications
eclipse-workspace - Ecli...
eclipse-workspace - Eclipse IDE
File Edit Refactor Navigate Search Project Run Window Help
Project Explorer
Outline
Markers
Properties
Servers
Data Source Explorer
Snippets
Console

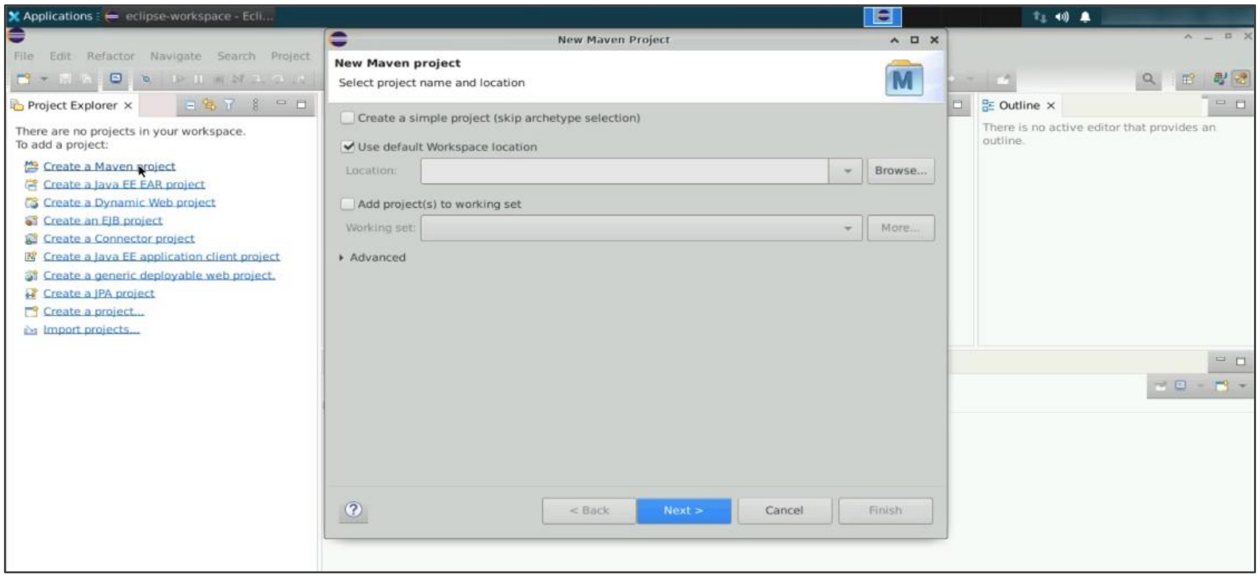
There are no projects in your workspace.
To add a project:

Create a Maven project
Create a Java EE EAR project
Create a Dynamic Web project
Create an EJB project
Create a Connector project
Create a Java EE application client project
Create a generic deployable web project.
Create a JPA project
Create a project...
Import projects...

There is no active editor that provides an outline.

No consoles to display at this time.

2.5 Choose **Use default Workspace location** and click on **Next**


New Maven Project

New Maven project

Select project name and location

- [ ] Create a simple project (skip archetype selection)
- [✓] Use default Workspace location

Location:

- [ ] Add project(s) to working set

Working set:

Advanced

Project Explorer

There are no projects in your workspace.
To add a project:

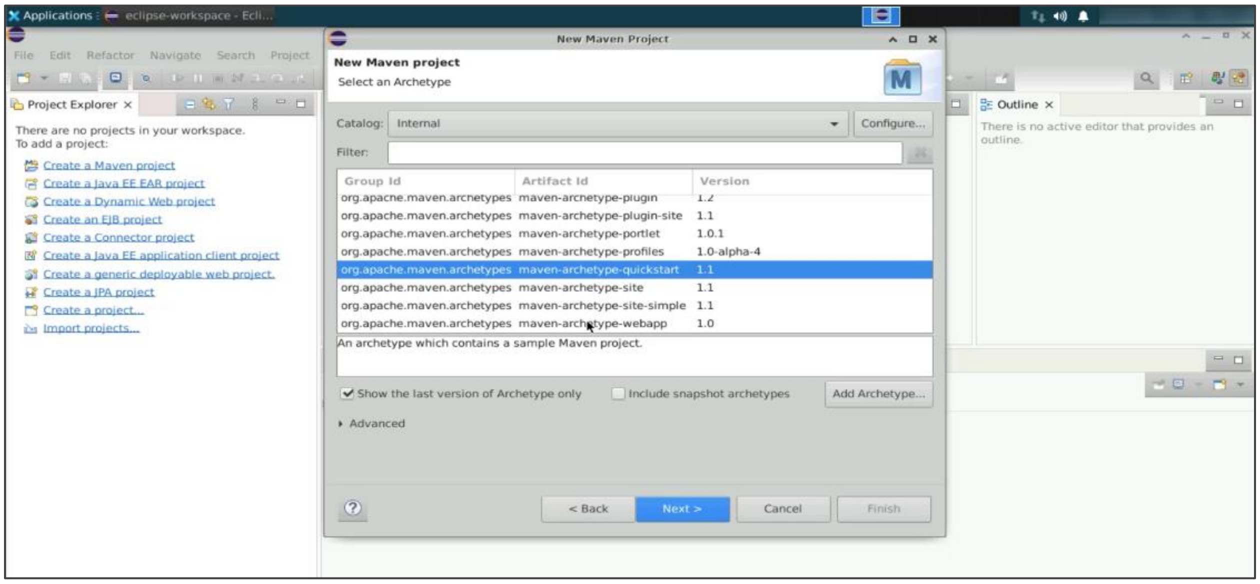
- Create a Maven project
- Create a Java EE EAR project
- Create a Dynamic Web project
- Create an EJB project
- Create a Connector project
- Create a Java EE application client project
- Create a generic deployable web project.
- Create a JPA project
- Create a project...
- Import projects...

Outline

There is no active editor that provides an outline.

2.6 Choose the following quickstart archetype and click on **Next:**

org.apache.maven.archetypes maven.archetype.quickstart


New Maven Project

New Maven project

Select an Archetype

Catalog: Internal

Filter:

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-portlet	1.0.1
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.1
org.apache.maven.archetypes	maven-archetype-site	1.1
org.apache.maven.archetypes	maven-archetype-site-simple	1.1
org.apache.maven.archetypes	maven-archetype-webapp	1.0

An archetype which contains a sample Maven project.

- [✓] Show the last version of Archetype only
- [ ] Include snapshot archetypes

Advanced

Project Explorer

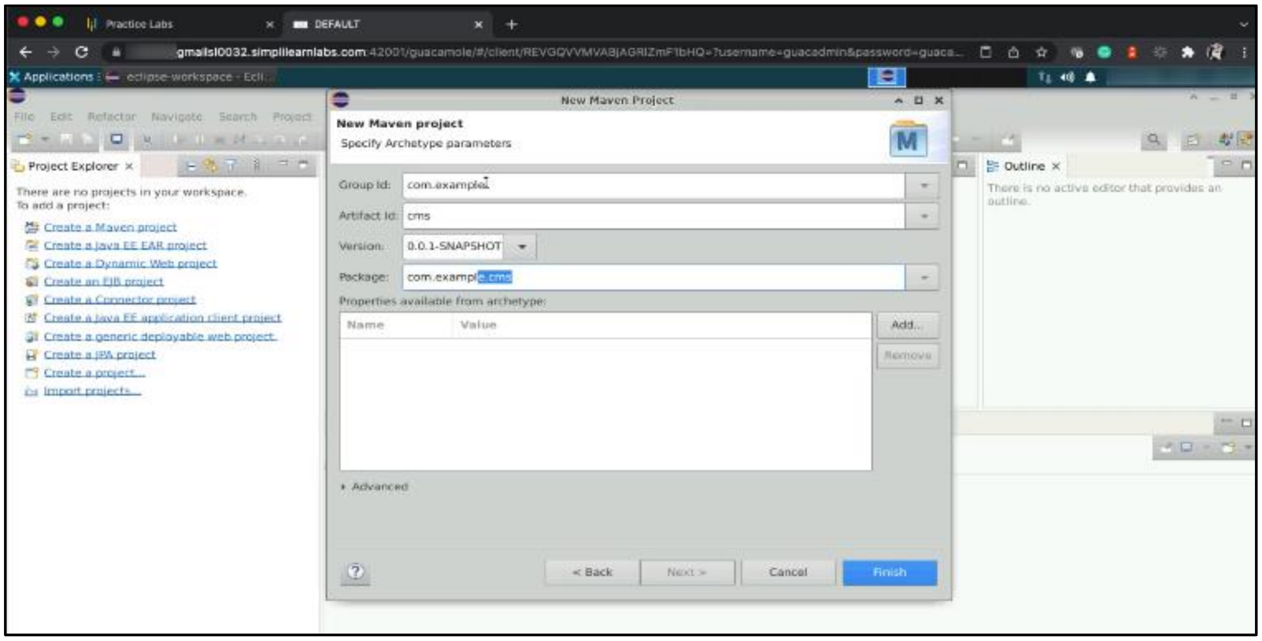
There are no projects in your workspace.
To add a project:

- Create a Maven project
- Create a Java EE EAR project
- Create a Dynamic Web project
- Create an EJB project
- Create a Connector project
- Create a Java EE application client project
- Create a generic deployable web project.
- Create a JPA project
- Create a project...
- Import projects...

Outline

There is no active editor that provides an outline.

2.7 To create the project, enter the **Group Id**, **Artifact Id**, and **Package** as **com.example, cms, and com.example.cms** respectively and click on **Finish**



New Maven Project

New Maven project

Specify Archetype parameters

Group Id:	com.example
Artifact Id:	cms
Version:	0.0.1-SNAPSHOT
Package:	com.example.cms

Properties available from archetype:

Name	Value

Add...

Remove

Advanced

< Back | Next > | Cancel | Finish

Project Explorer

There are no projects in your workspace.
To add a project:

- Create a Maven project
- Create a Java EE EAR project
- Create a Dynamic Web project
- Create an EJB project
- Create a Connector project
- Create a Java EE application client project
- Create a generic deployable web project
- Create a JPA project
- Create a project...
- Import projects...

Outline

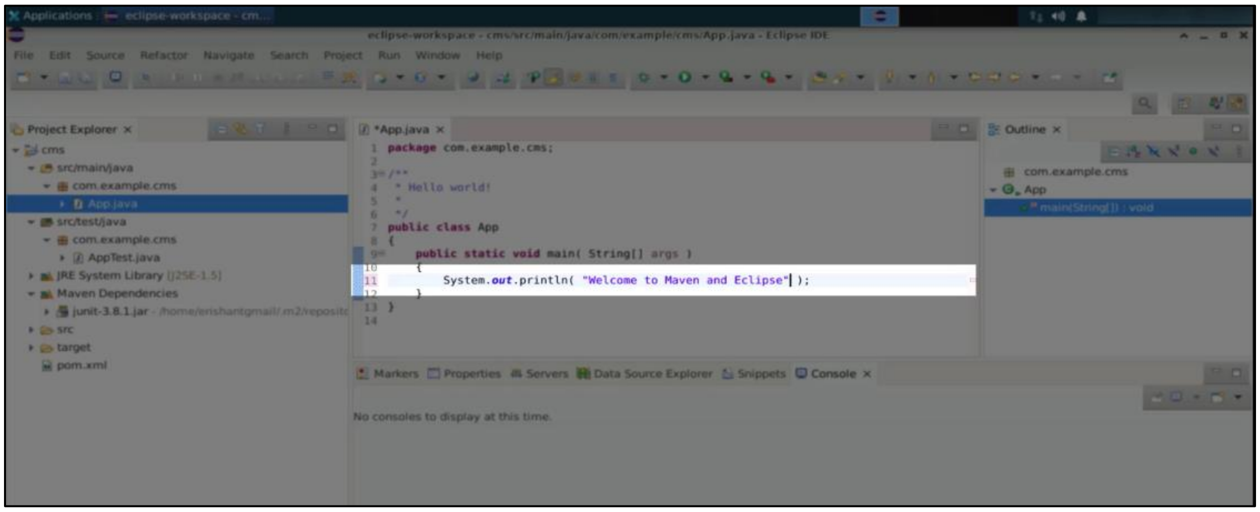
There is no active editor that provides an outline.

The **Group Id** is the company's domain name in reverse order and the **Artifact Id** is the project name. The **Package** name is automatically typed as you enter the other two values.

Step 3: Execute the Maven build

3.1 Under **src/main/ java,** open the **App.java** file and change the code snippet

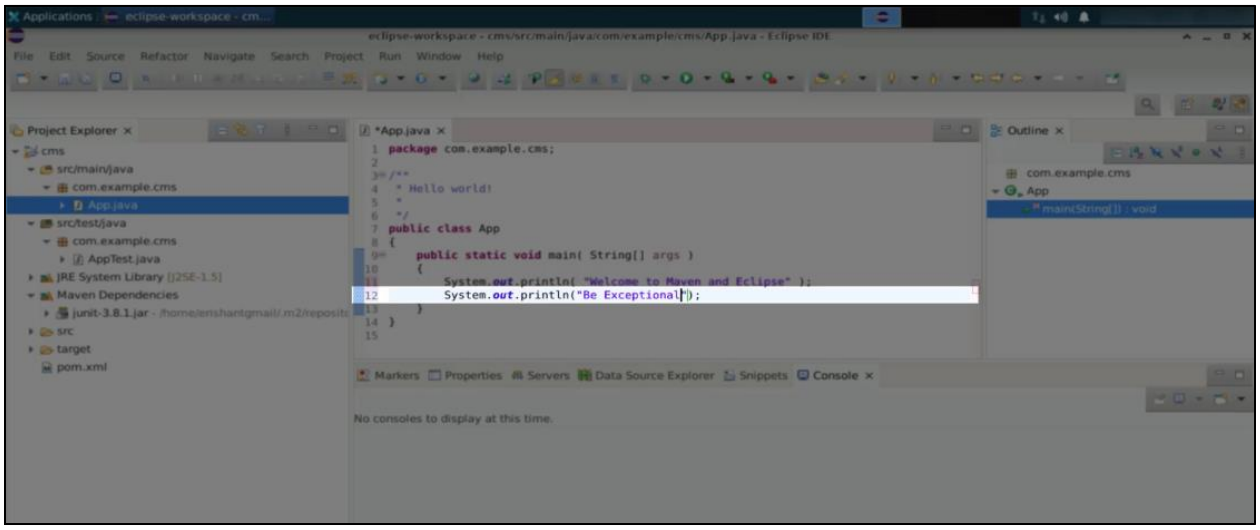
System.out.println("Welcome to Maven and Eclipse");



```
package com.example.cms;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Welcome to Maven and Eclipse" );
    }
}
```

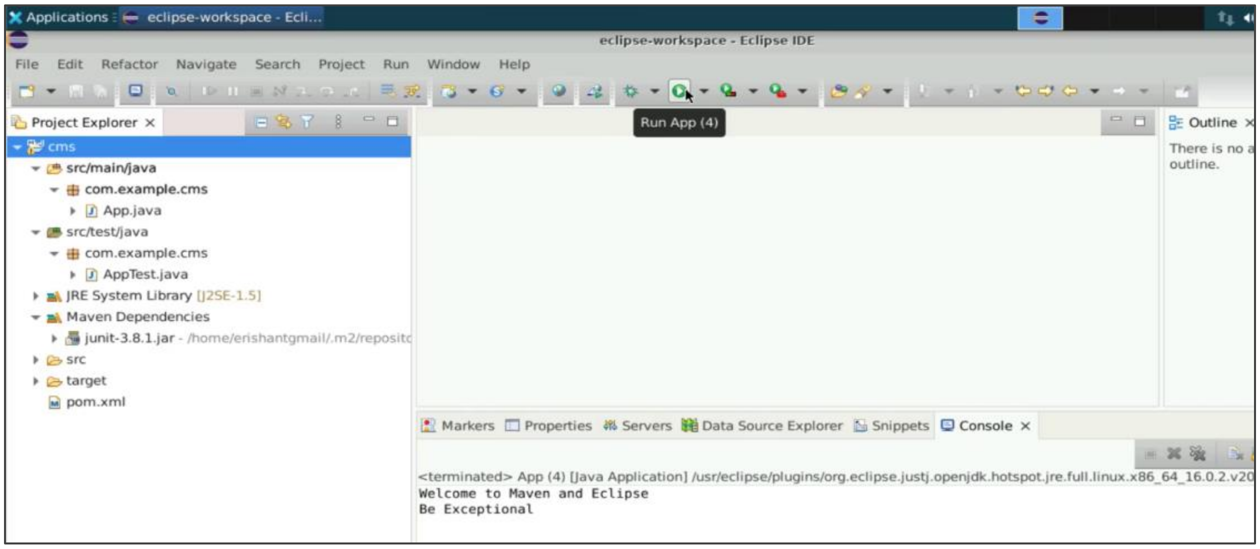
3.2 Add another **print** statement

System.out.println("Be Exceptional");



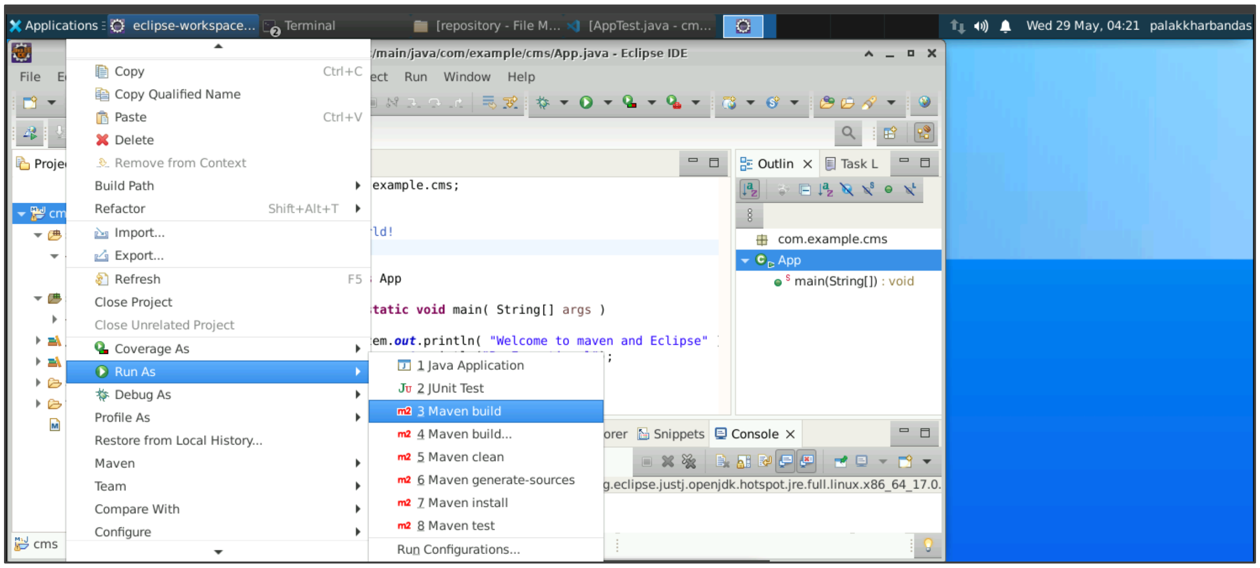
```
package com.example.cms;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Welcome to Maven and Eclipse" );
        System.out.println("Be Exceptional");
    }
}
```

3.3 Save and run the application by clicking on the green button


Applications
eclipse-workspace - Eclipse IDE
File Edit Refactor Navigate Search Project Run Window Help
Run App (4)
Project Explorer
cms
src/main/java
com.example.cms
App.java
src/test/java
com.example.cms
AppTest.java
JRE System Library [J2SE-1.5]
Maven Dependencies
junit-3.8.1.jar - /home/erishantgmail/.m2/reposito
src
target
pom.xml
Outline
There is no outline.
Markers Properties Servers Data Source Explorer Snippets Console

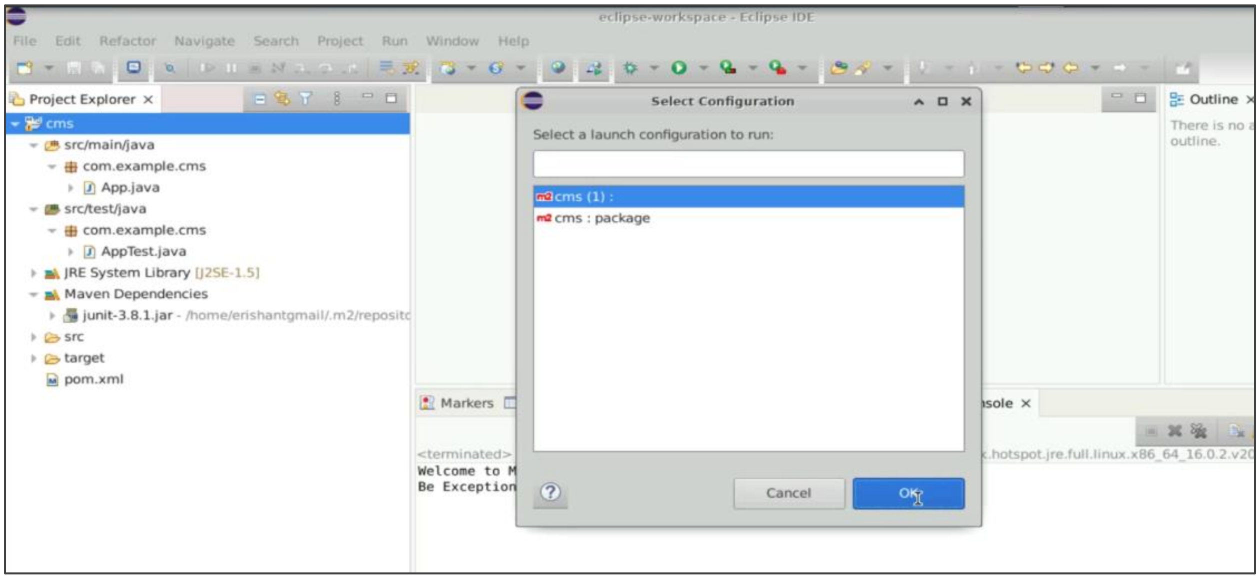
```
<terminated> App (4) [Java Application] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20
Welcome to Maven and Eclipse
Be Exceptional
```

3.4 Now, go to **Run As** to select options to run the project. Select **Maven Build.**

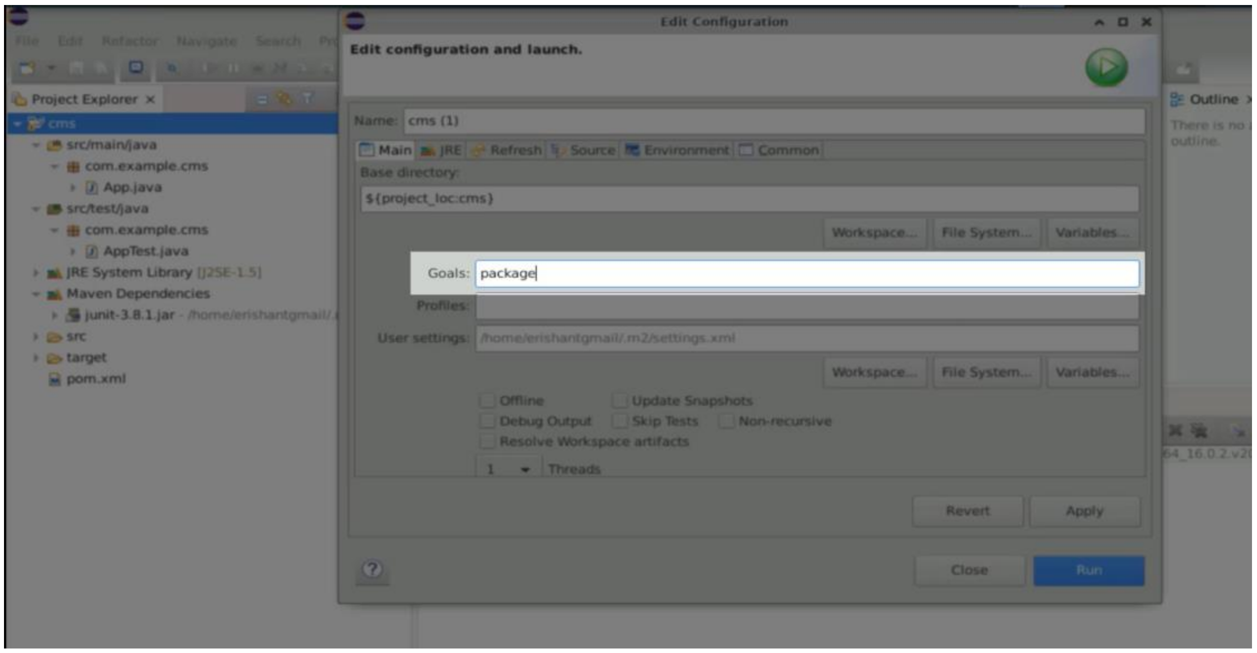


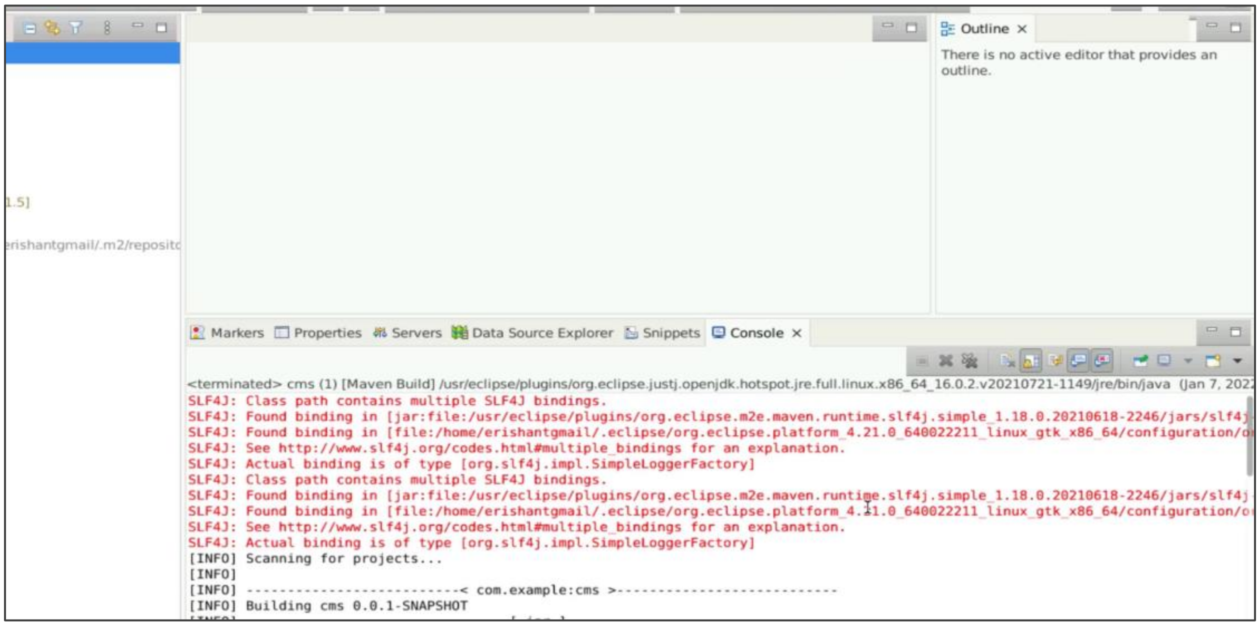
```
example.cms;
ld!
App
tatic void main( String[] args )
em.out.println( "Welcome to maven and Eclipse"
```

3.5 Click on **OK**



3.6 Specify the **Goals** as **package** and click on **Run**

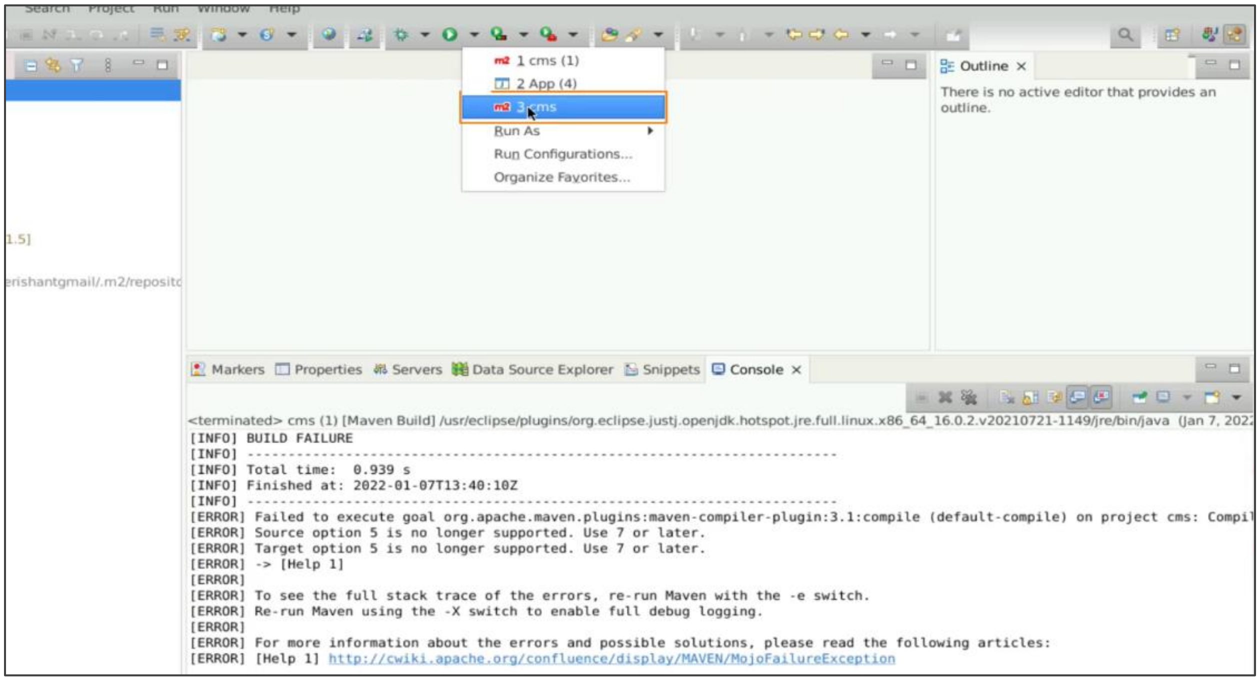

Edit Configuration
Edit configuration and launch.
Name: cms (1)
Base directory: ${project_loc:cms}
Goals: package
Close
Run



```
<terminated> cms (1) [Maven Build] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/java  (Jan 7, 202
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/eclipse/plugins/org.eclipse.m2e.maven.runtime.slf4j.simple_1.18.0.20210618-2246/jars/slf4j
SLF4J: Found binding in [file:/home/erishantgmail/.eclipse/org.eclipse.platform_4.21.0_640022211_linux_gtk_x86_64/configuration/o
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.SimpleLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/eclipse/plugins/org.eclipse.m2e.maven.runtime.slf4j.simple_1.18.0.20210618-2246/jars/slf4j
SLF4J: Found binding in [file:/home/erishantgmail/.eclipse/org.eclipse.platform_4.21.0_640022211_linux_gtk_x86_64/configuration/o
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.SimpleLoggerFactory]
[INFO] Scanning for projects...
[INFO]
[INFO] --------------------------< com.example:cms >---------------------------
[INFO] Building cms 0.0.1-SNAPSHOT
```

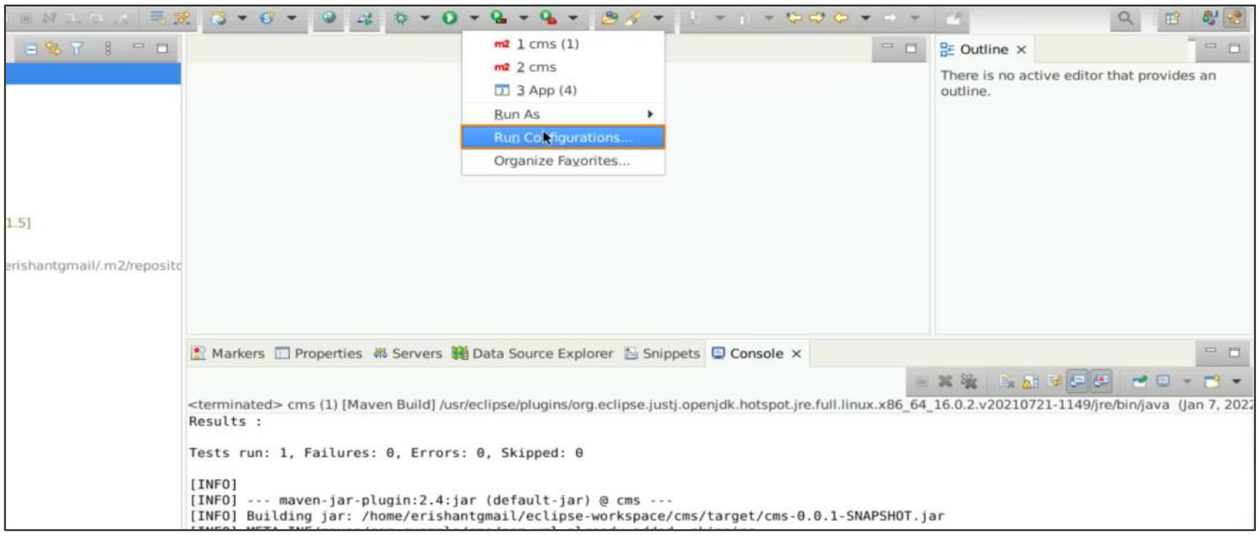
The Maven build is now executed.

3.7 Re-run the code. Click on the green button and select **cms.**



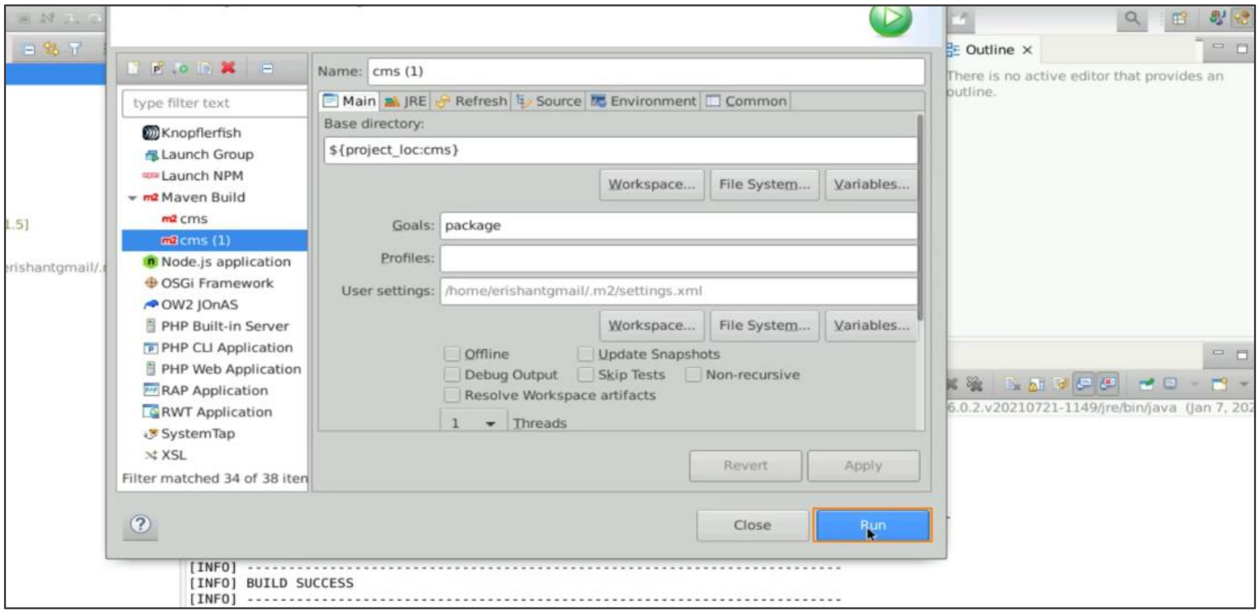
```
<terminated> cms (1) [Maven Build] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/java (Jan 7, 2022
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.939 s
[INFO] Finished at: 2022-01-07T13:40:10Z
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.1:compile (default-compile) on project cms: Compil
[ERROR] Source option 5 is no longer supported. Use 7 or later.
[ERROR] Target option 5 is no longer supported. Use 7 or later.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
```

3.8 Navigate to **Run Configurations**



```
<terminated> cms (1) [Maven Build] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/java (Jan 7, 2022
Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ cms ---
[INFO] Building jar: /home/erishantgmail/eclipse-workspace/cms/target/cms-0.0.1-SNAPSHOT.jar
```

3.9 Select **cms(1)** as run configuration and click on **Run**



Name: cms (1)

Base directory: ${project_loc:cms}

Goals: package

Profiles:

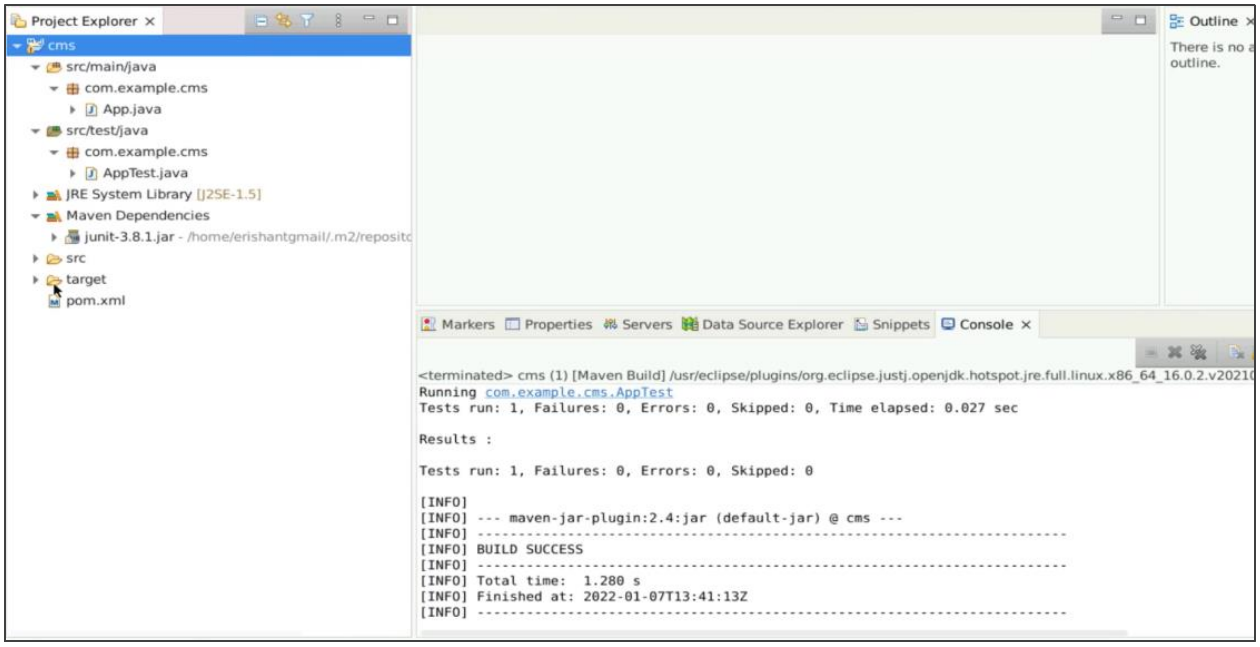
User settings: /home/erishantgmail/.m2/settings.xml

Offline, Update Snapshots, Debug Output, Skip Tests, Non-recursive, Resolve Workspace artifacts

1 Threads

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

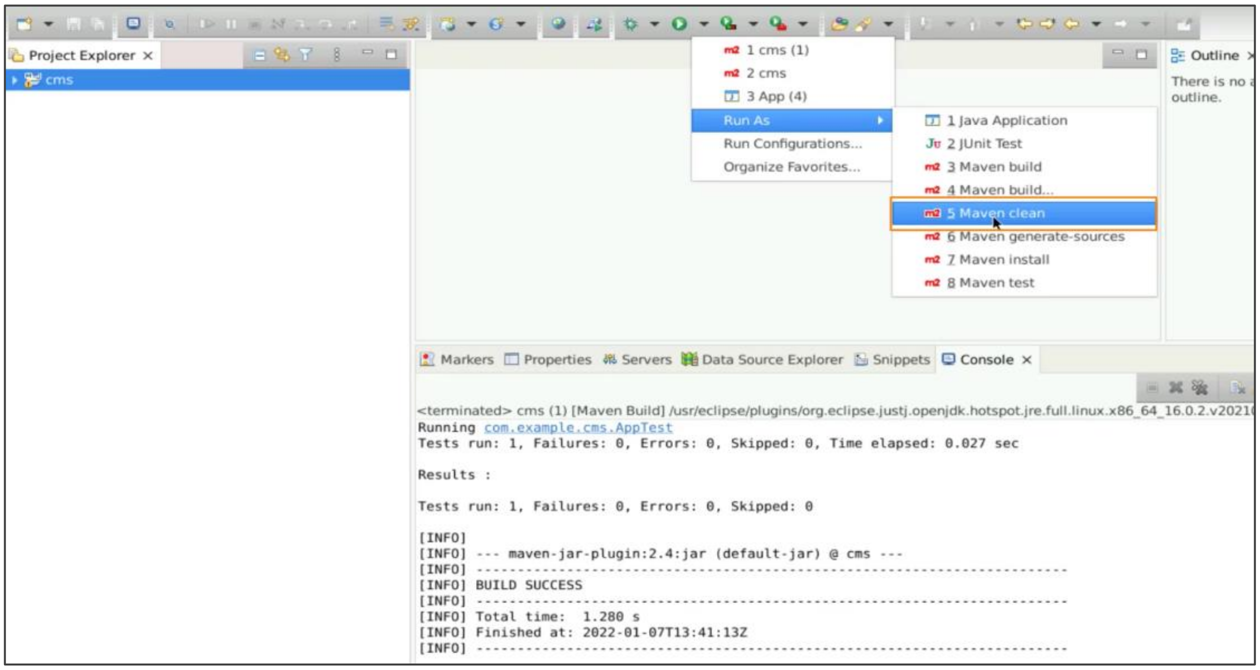
We can see BUILD SUCCESS and the **target** directory shows different structures.



```
<terminated> cms (1) [Maven Build] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210
Running com.example.cms.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.027 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ cms ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.280 s
[INFO] Finished at: 2022-01-07T13:41:13Z
[INFO] ------------------------------------------------------------------------
```

Step 4: Run as Maven install

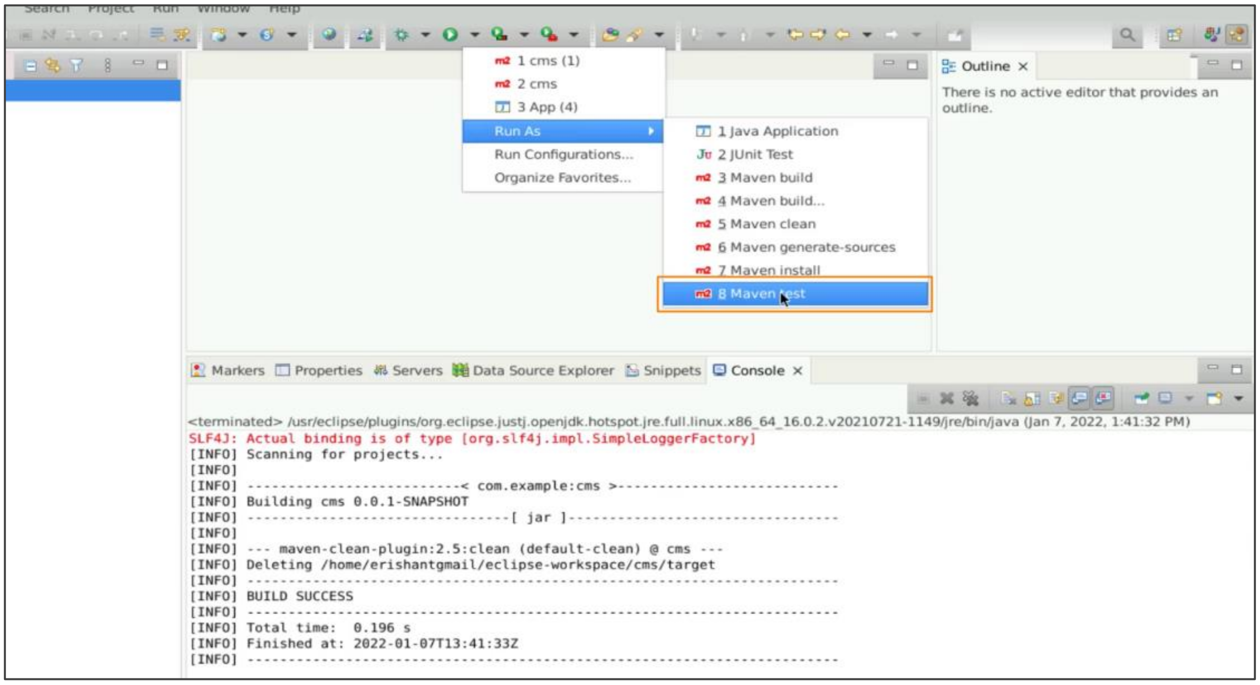
4.1 Now, run the project as **Maven clean**



```
<terminated> cms (1) [Maven Build] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v2021
Running com.example.cms.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.027 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ cms ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.280 s
[INFO] Finished at: 2022-01-07T13:41:13Z
[INFO] ------------------------------------------------------------------------
```

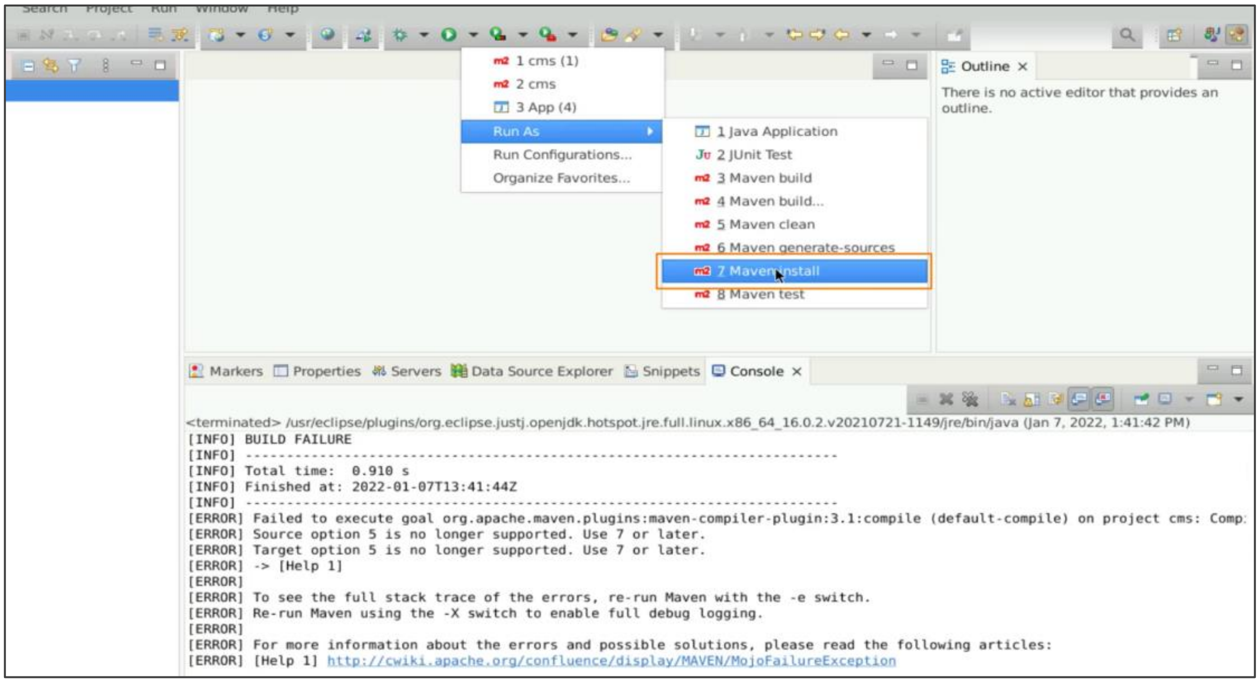
It will remove the target directories.

4.2 Select the **Maven test** to execute the test in the same way



```
<terminated> /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/java (Jan 7, 2022, 1:41:32 PM)
SLF4J: Actual binding is of type [org.slf4j.impl.SimpleLoggerFactory]
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< com.example:cms >-------------------------
[INFO] Building cms 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ cms ---
[INFO] Deleting /home/erishantgmail/eclipse-workspace/cms/target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.196 s
[INFO] Finished at: 2022-01-07T13:41:33Z
[INFO] ------------------------------------------------------------------------
```

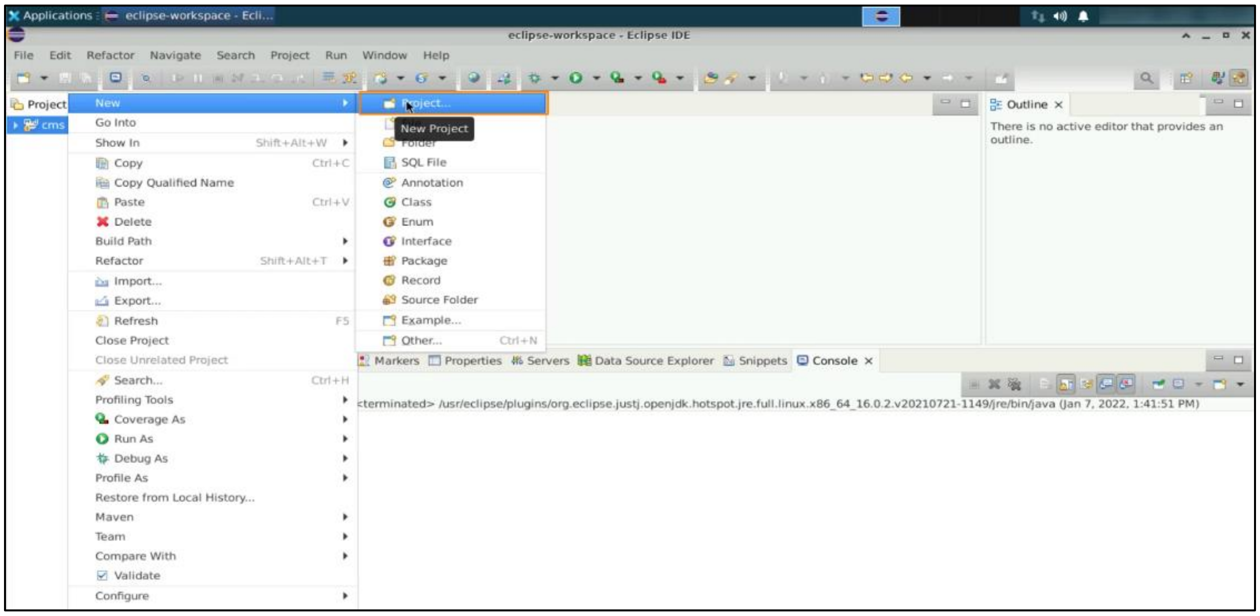
4.3 Run as **Maven install**



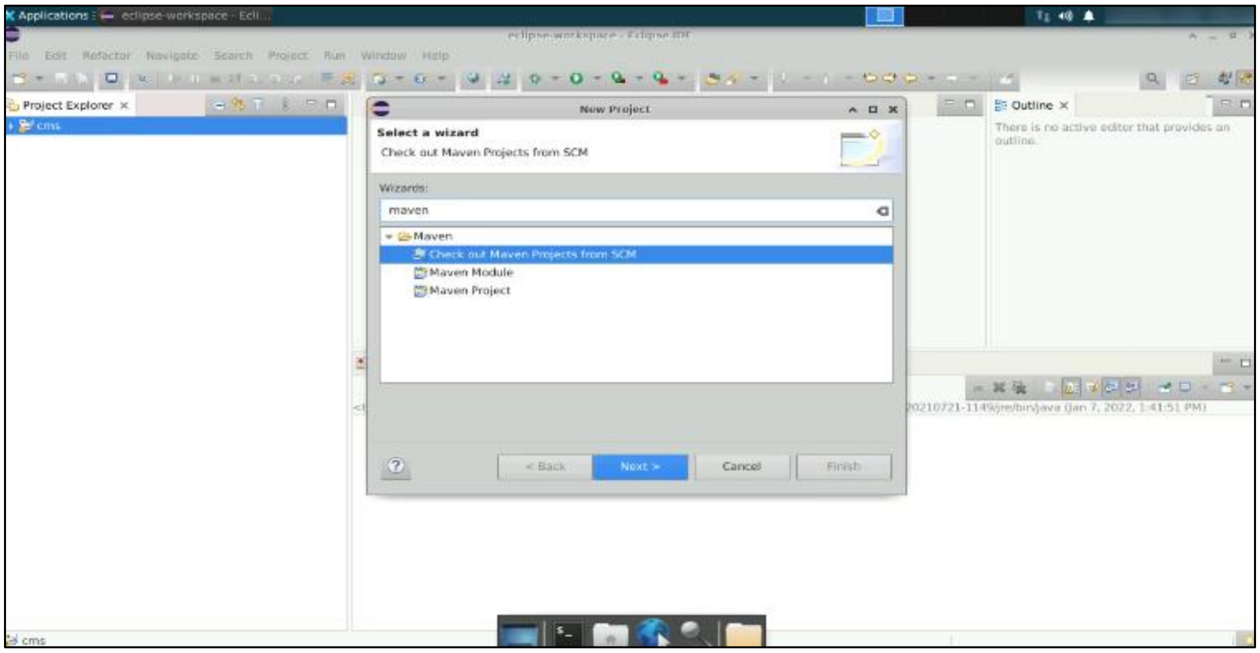
```
<terminated> /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_16.0.2.v20210721-1149/jre/bin/java (Jan 7, 2022, 1:41:42 PM)
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.910 s
[INFO] Finished at: 2022-01-07T13:41:44Z
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.1:compile (default-compile) on project cms: Comp
[ERROR] Source option 5 is no longer supported. Use 7 or later.
[ERROR] Target option 5 is no longer supported. Use 7 or later.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
```

Step 5: Create a Maven project

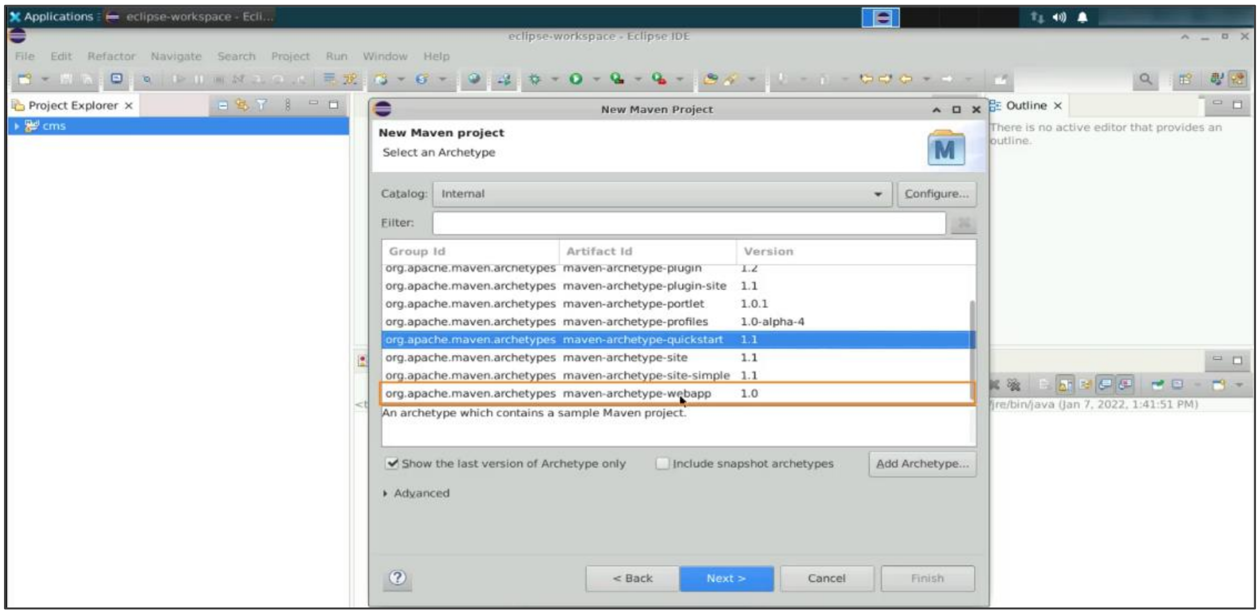
5.1 Right-click on the project **cms**, select **New**, and select **Project**



5.2 Select Maven as the wizard. Select **Maven Project** and click on **Next.**



5.3 Choose the archetype as **webapp and** click on **Next.** This means we are creating an enterprise-level project.


New Maven project

Select an Archetype

Catalog: Internal

Filter:

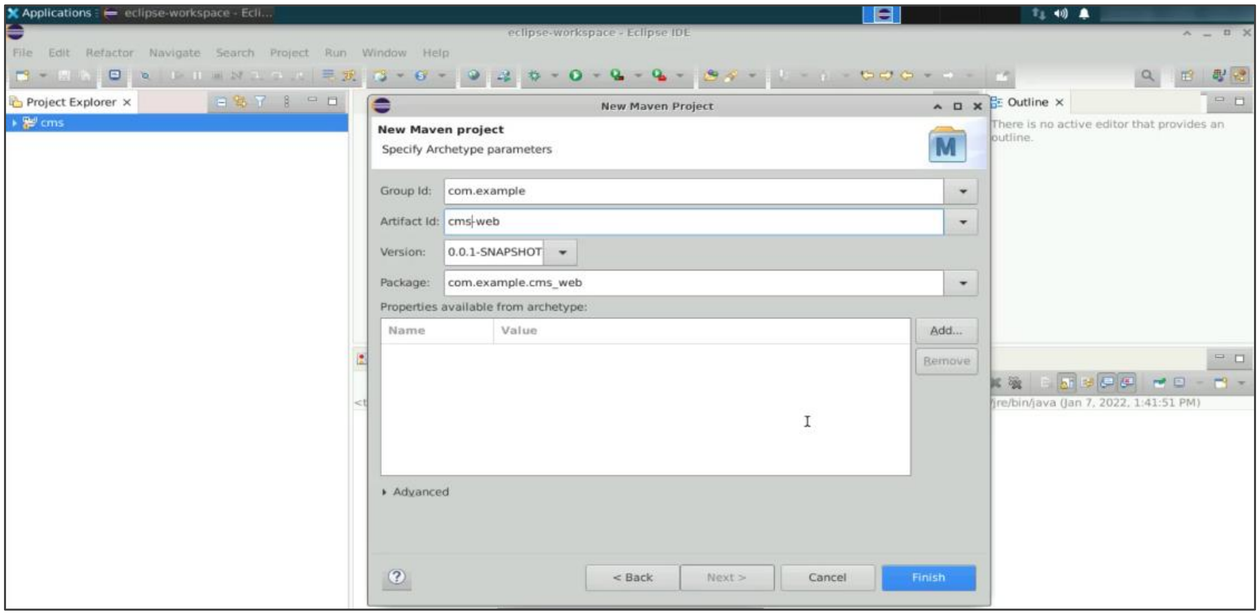
Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-portlet	1.0.1
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.1
org.apache.maven.archetypes	maven-archetype-site	1.1
org.apache.maven.archetypes	maven-archetype-site-simple	1.1
org.apache.maven.archetypes	maven-archetype-webapp	1.0

An archetype which contains a sample Maven project.

[✓] Show the last version of Archetype only
[ ] Include snapshot archetypes

Applications
eclipse-workspace - Eclipse IDE
File Edit Refactor Navigate Search Project Run Window Help
Project Explorer
cms
New Maven Project
Configure...
Add Archetype...
Advanced
< Back
Next >
Cancel
Finish
Outline
There is no active editor that provides an outline.

5.4 Add the **Artifact Id** as **cms-web Finish**


New Maven project

Specify Archetype parameters

Group Id: com.example

Artifact Id: cms-web

Version: 0.0.1-SNAPSHOT

Package: com.example.cms_web

Properties available from archetype:

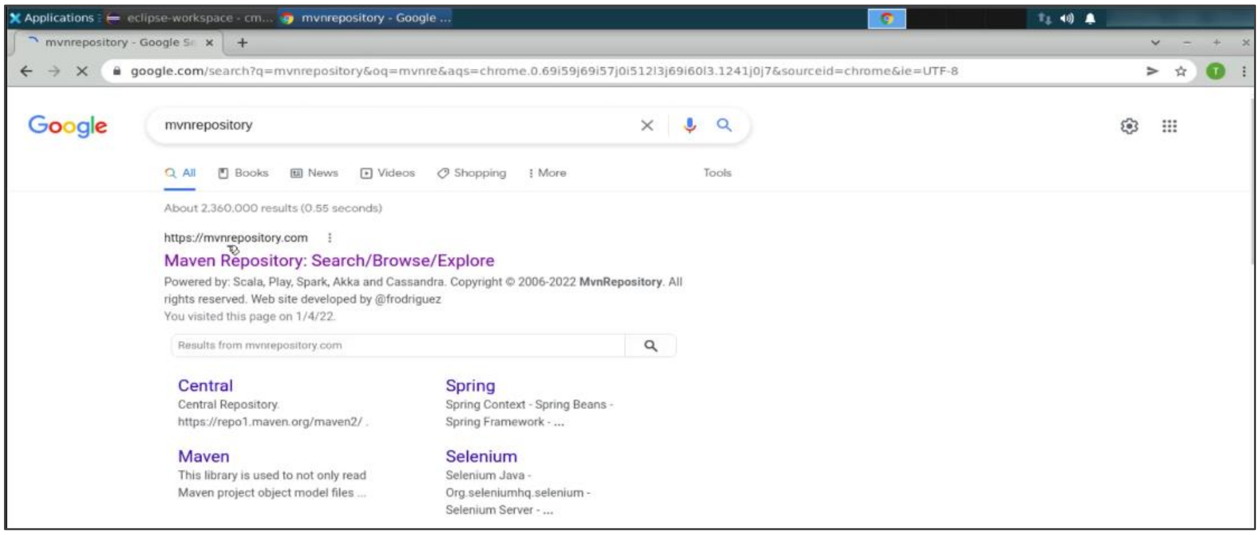
Name	Value

Applications
eclipse-workspace - Eclipse IDE
File Edit Refactor Navigate Search Project Run Window Help
Project Explorer
cms
New Maven Project
Add...
Remove
Advanced
< Back
Next >
Cancel
Finish
Outline
There is no active editor that provides an outline.

This will create the enterprise project **cms-web**.

Step 6: Add dependency in the pom.xml file

6.1 Open the browser and search for **mvnrepository**



mvnrepository

About 2,360,000 results (0.55 seconds)

https://mvnrepository.com

Maven Repository: Search/Browse/Explore

Powered by: Scala, Play, Spark, Akka and Cassandra. Copyright © 2006-2022 MvnRepository. All rights reserved. Web site developed by @frodriguez

You visited this page on 1/4/22.

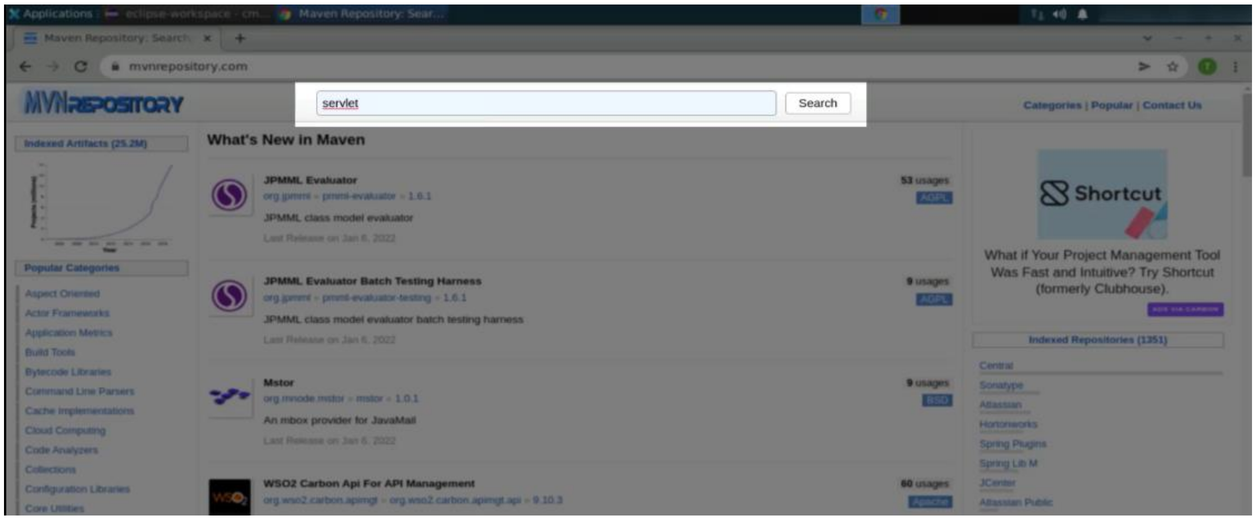
Central
Central Repository.
https://repo1.maven.org/maven2/ .

Spring
Spring Context - Spring Beans - Spring Framework - ...

Maven
This library is used to not only read Maven project object model files ...

Selenium
Selenium Java - Org.seleniumhq.selenium - Selenium Server - ...

6.2 Open the link and search for the **servlet**



servlet

What's New in Maven

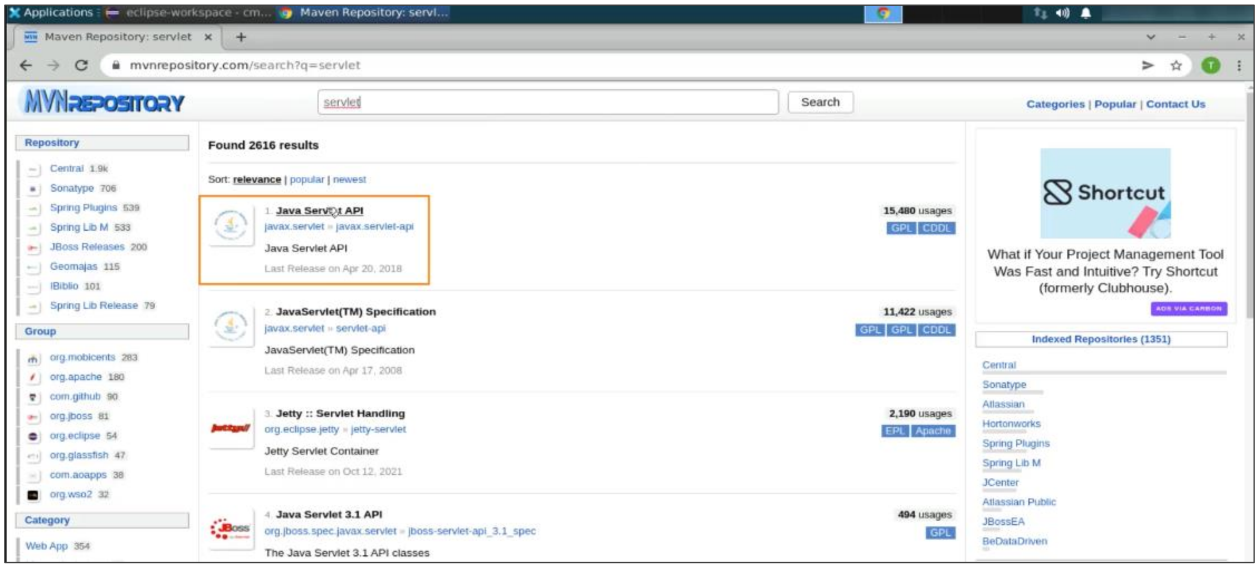
JPMML Evaluator
org.jpmml » pmml-evaluator » 1.6.1
JPMML class model evaluator

JPMML Evaluator Batch Testing Harness
org.jpmml » pmml-evaluator-testing » 1.6.1
JPMML class model evaluator batch testing harness

Mstor
org.mnode.mstor » mstor » 1.0.1
An mbox provider for JavaMail

WSO2 Carbon Api For API Management
org.wso2.carbon.apimgt » org.wso2.carbon.apimgt.api » 9.10.3

6.3 Choose the latest version from **Java Servlet API**



mvnrepository.com/search?q=servlet

Found 2616 results

Sort: relevance | popular | newest

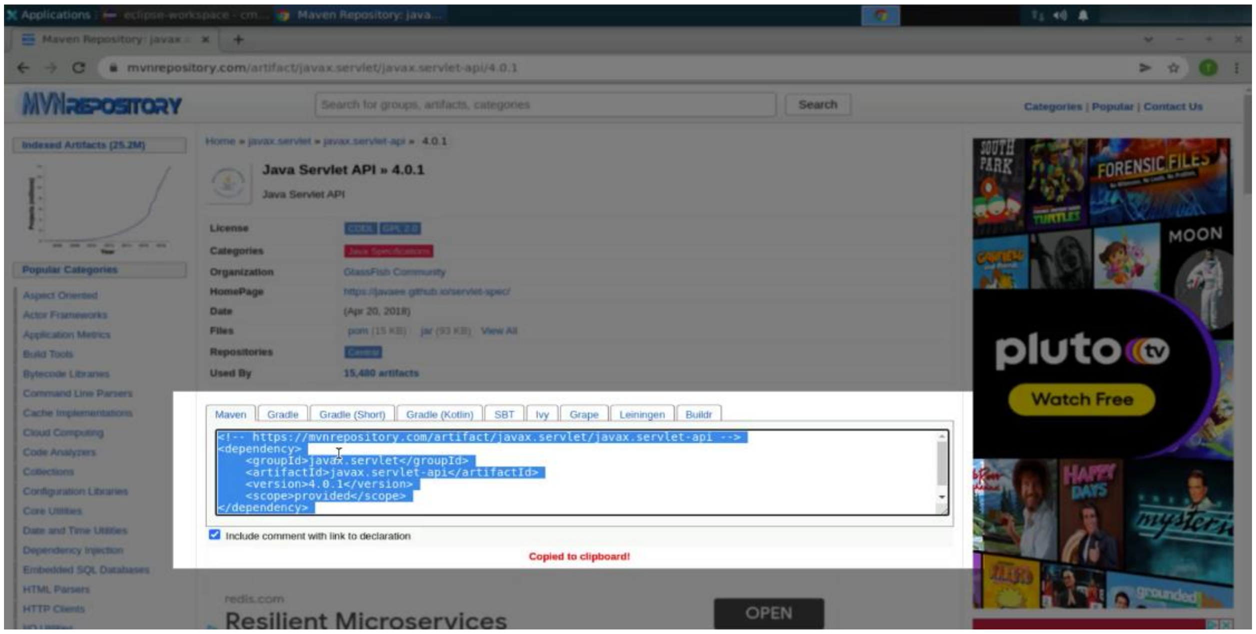
1. **Java Servlet API** — javax.servlet » javax.servlet-api — 15,480 usages — GPL CDDL
Java Servlet API
Last Release on Apr 20, 2018

2. **JavaServlet(TM) Specification** — javax.servlet » servlet-api — 11,422 usages — GPL GPL CDDL
JavaServlet(TM) Specification
Last Release on Apr 17, 2008

3. **Jetty :: Servlet Handling** — org.eclipse.jetty » jetty-servlet — 2,190 usages — EPL Apache
Jetty Servlet Container
Last Release on Oct 12, 2021

4. **Java Servlet 3.1 API** — org.jboss.spec.javax.servlet » jboss-servlet-api_3_1_spec — 494 usages — GPL
The Java Servlet 3.1 API classes

6.4 Copy the **Maven dependency**



mvnrepository.com/artifact/javax.servlet/javax.servlet-api/4.0.1

Java Servlet API » 4.0.1

Java Servlet API

License	CDDL GPL 2.0
Categories	Java Specifications
Organization	GlassFish Community
HomePage	https://javaee.github.io/servlet-spec/
Date	(Apr 20, 2018)
Files	pom (15 KB) jar (93 KB) View All
Repositories	Central
Used By	15,480 artifacts

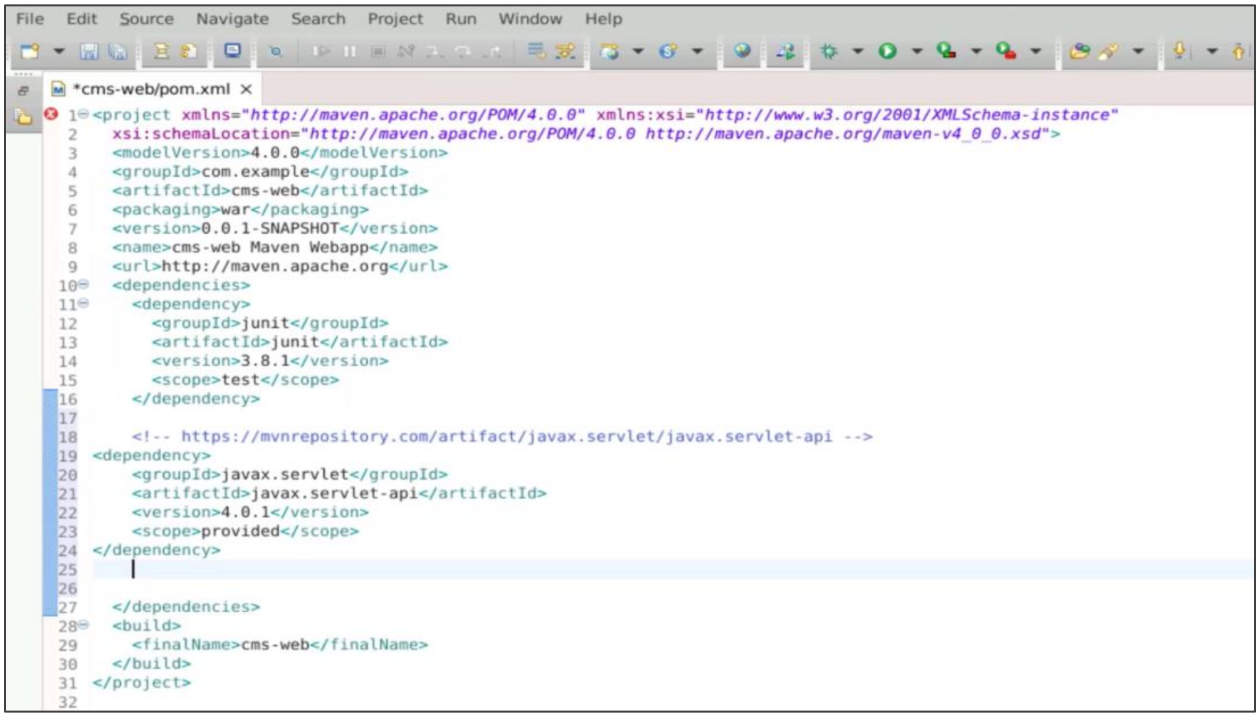
Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>
```

✓ Include comment with link to declaration

Copied to clipboard!

6.5 Go back to the Eclipse. Paste the dependency in the **cms-web/pom.xml** file and save it.



```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>cms-web</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>cms-web Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>


  </dependencies>
  <build>
    <finalName>cms-web</finalName>
  </build>
</project>
```

Once the dependency is saved, it will sync up the API and add it to the project. Similarly, different configurations can be added.

If the enterprise edition is being used, the plugin has to be installed manually.

By following these steps, you will successfully install Maven, configure it, execute builds, run installations, create projects, and manage dependencies in the pom.xml file, establishing a solid foundation for your Java projects.