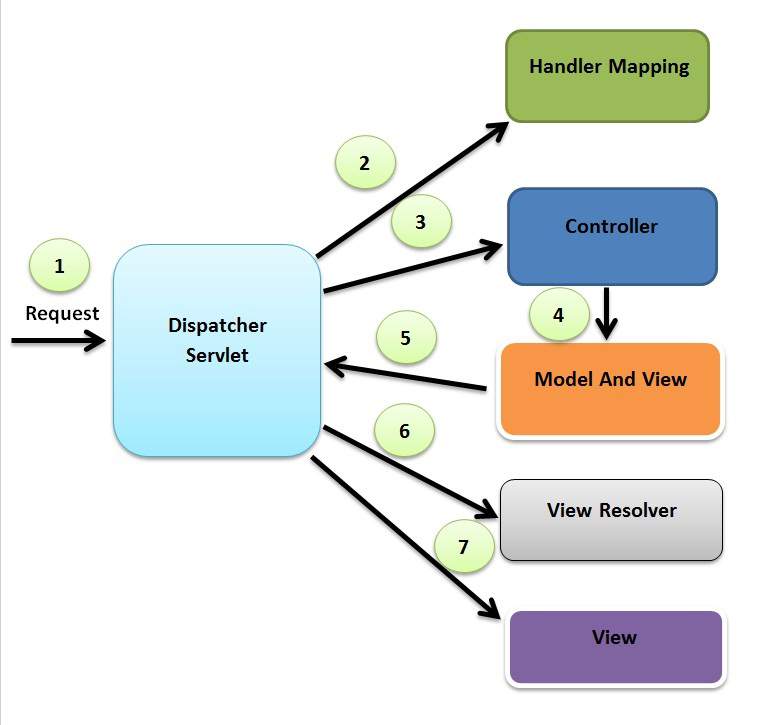
Spring MVC:

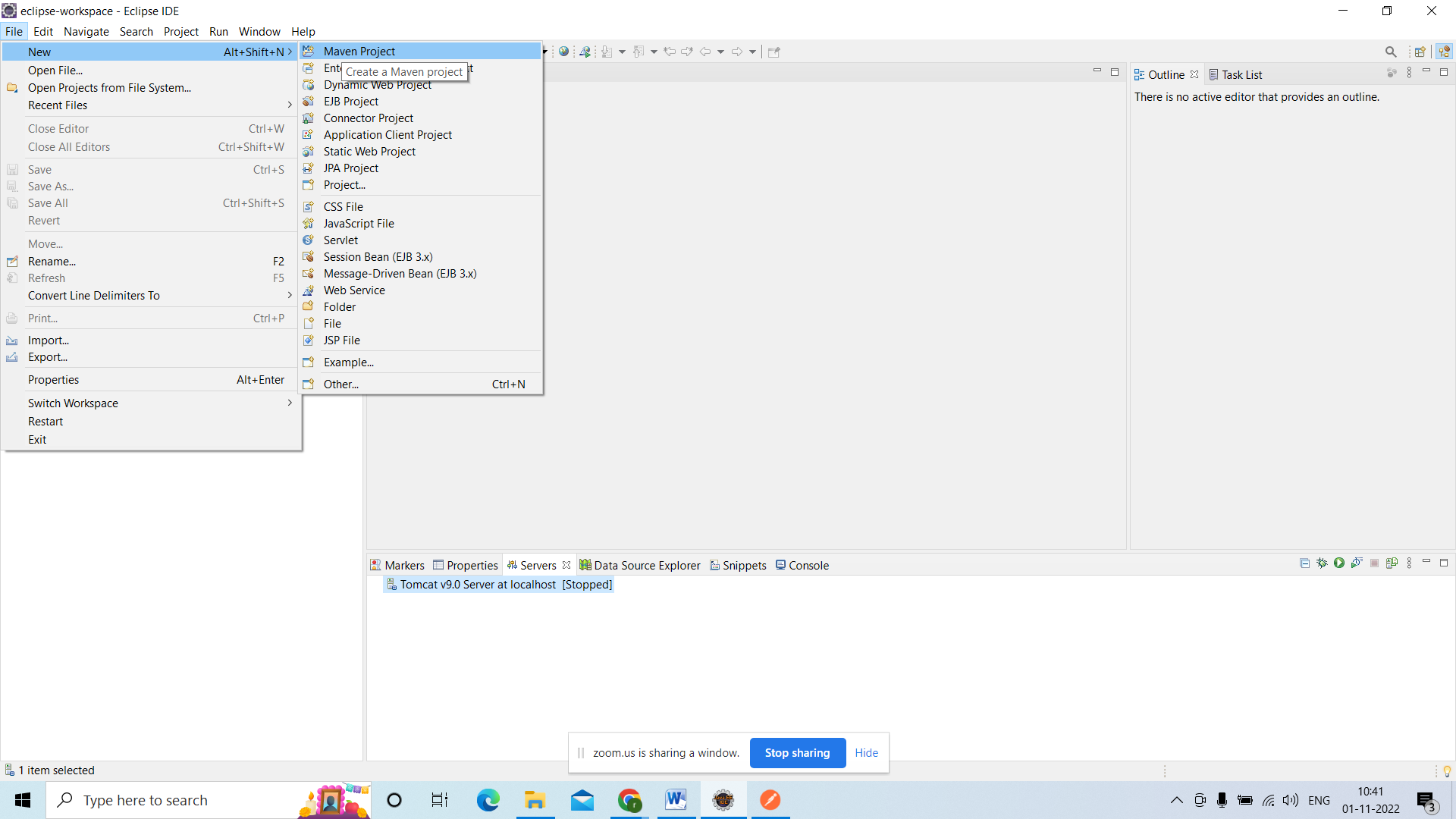
**MVC** is a design pattern which provides a solution to layer an application by separating **Business(Model), Presentation(View)**and**Control Flow(Controller)**. The **Model** contains the business logic and the **Controller** is responsible for the redirection and the interaction between **View** component and **Model**. The **View** component contains the presentation part of the application.**[](https://javainterviewpoint.com/wp-content/uploads/2015/04/Spring_MVC_Flow_Diagram1.jpg)**

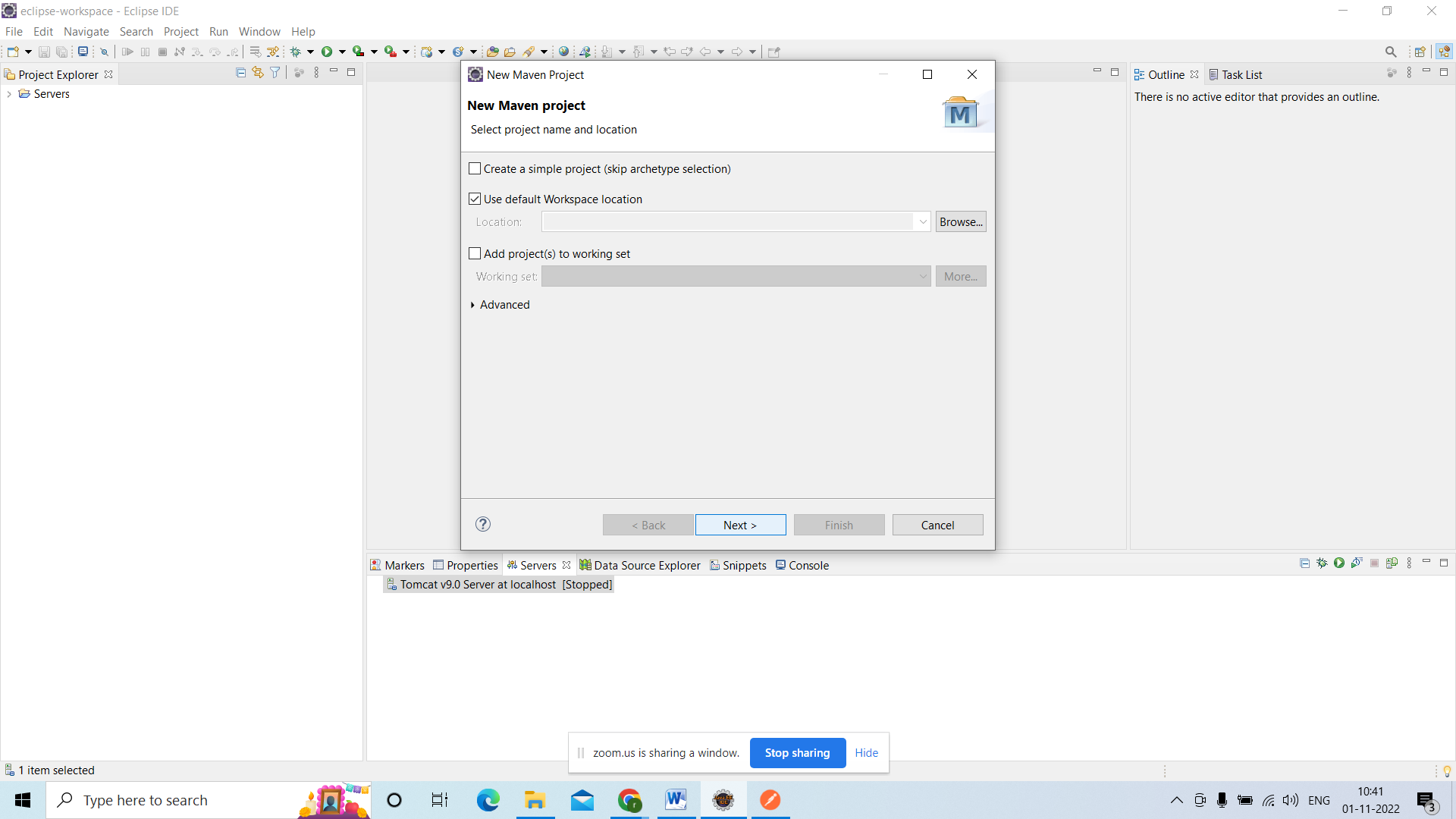
1)The request which are coming from browser or postman will be captured by Dispatcher Servlet it will act as a front controller.

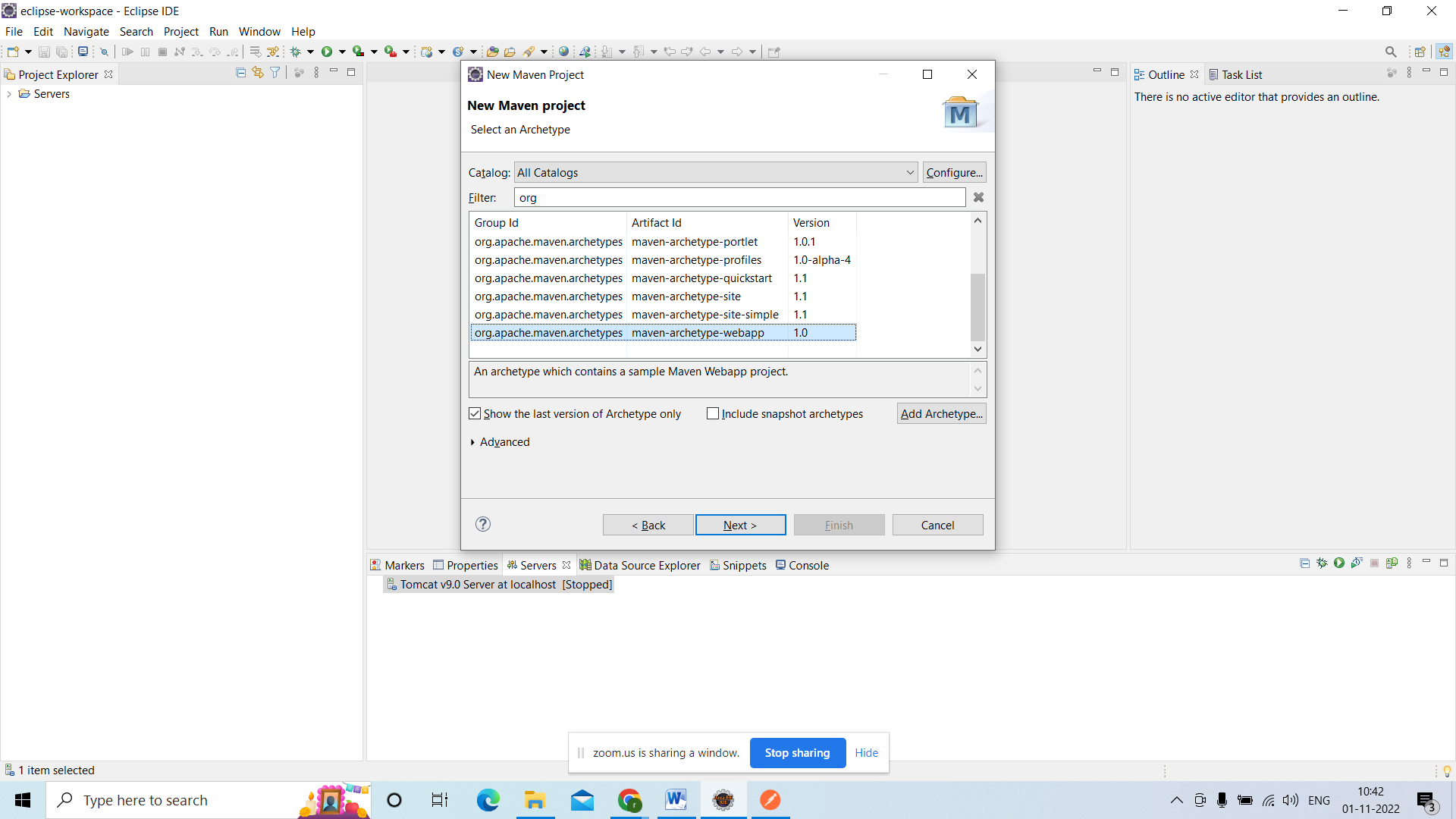
2) Dispatcher Servlet will find out the particular controller class with the help of handler mapping based on the URL

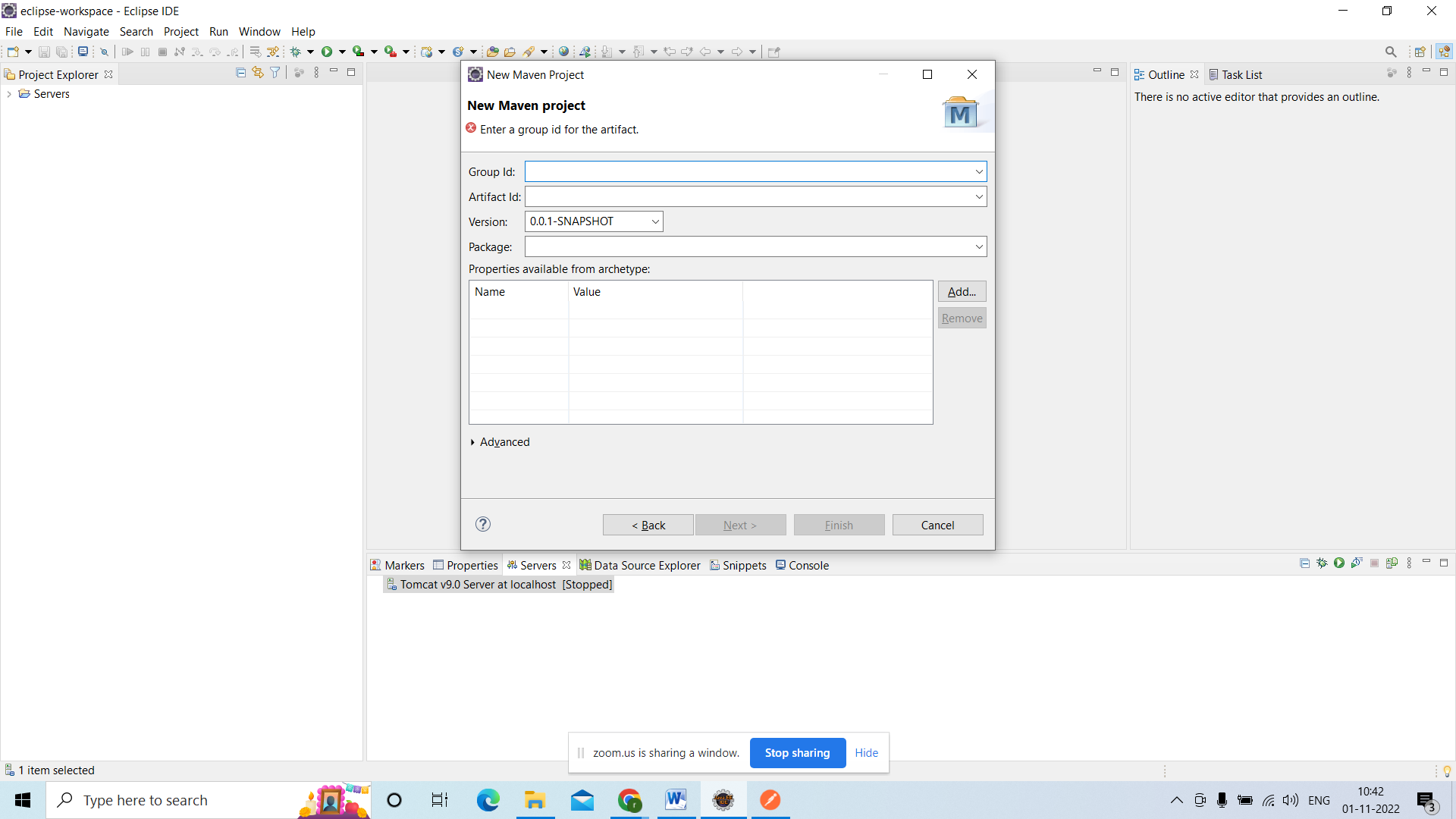
3)controller class will process the request and response send back to dispatcher servlet now dispatcher servlet display the front end pages(jsps) on browser with the help of internal resource view resolver.

Spring MVC With HelloWorld:



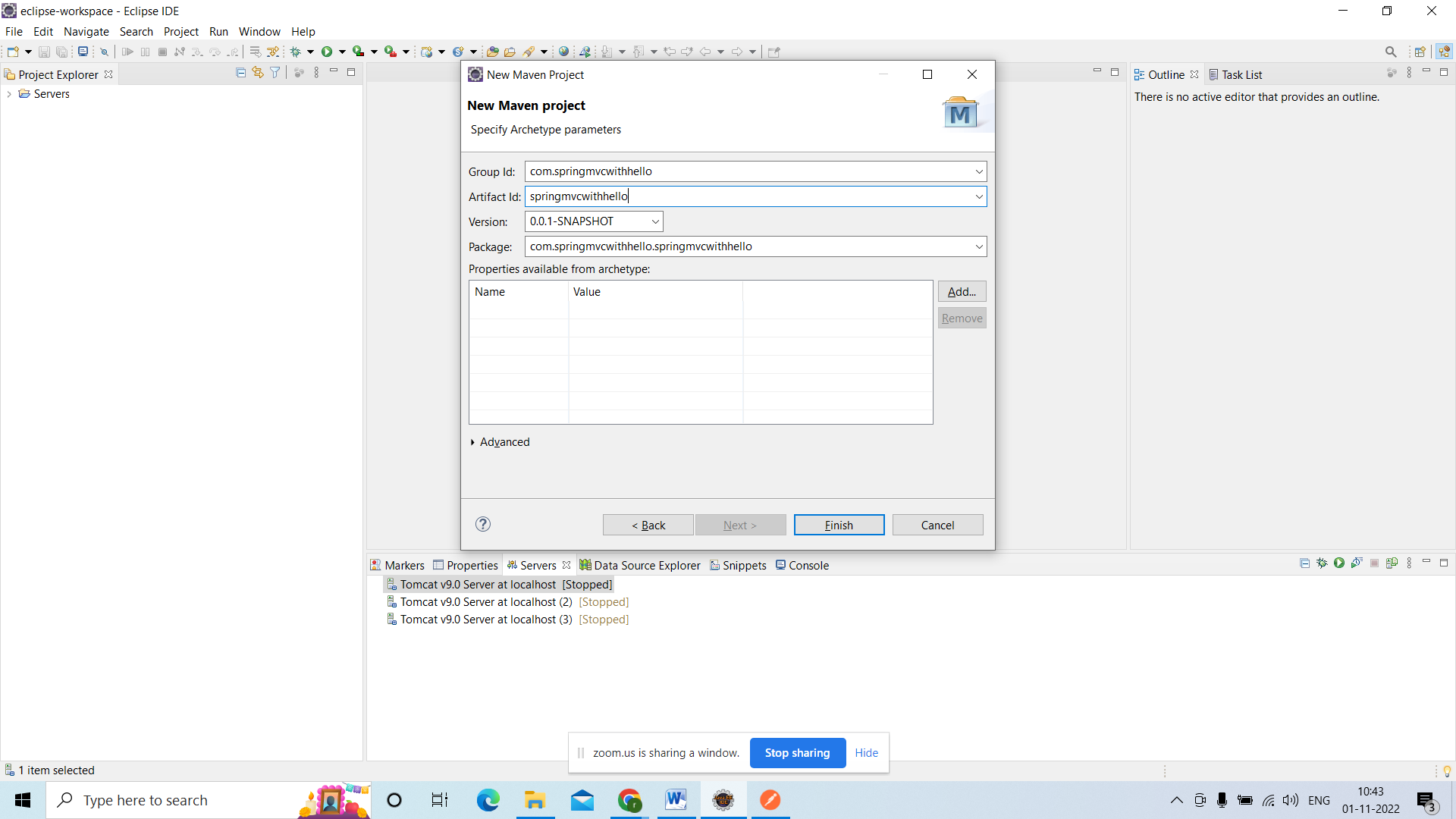






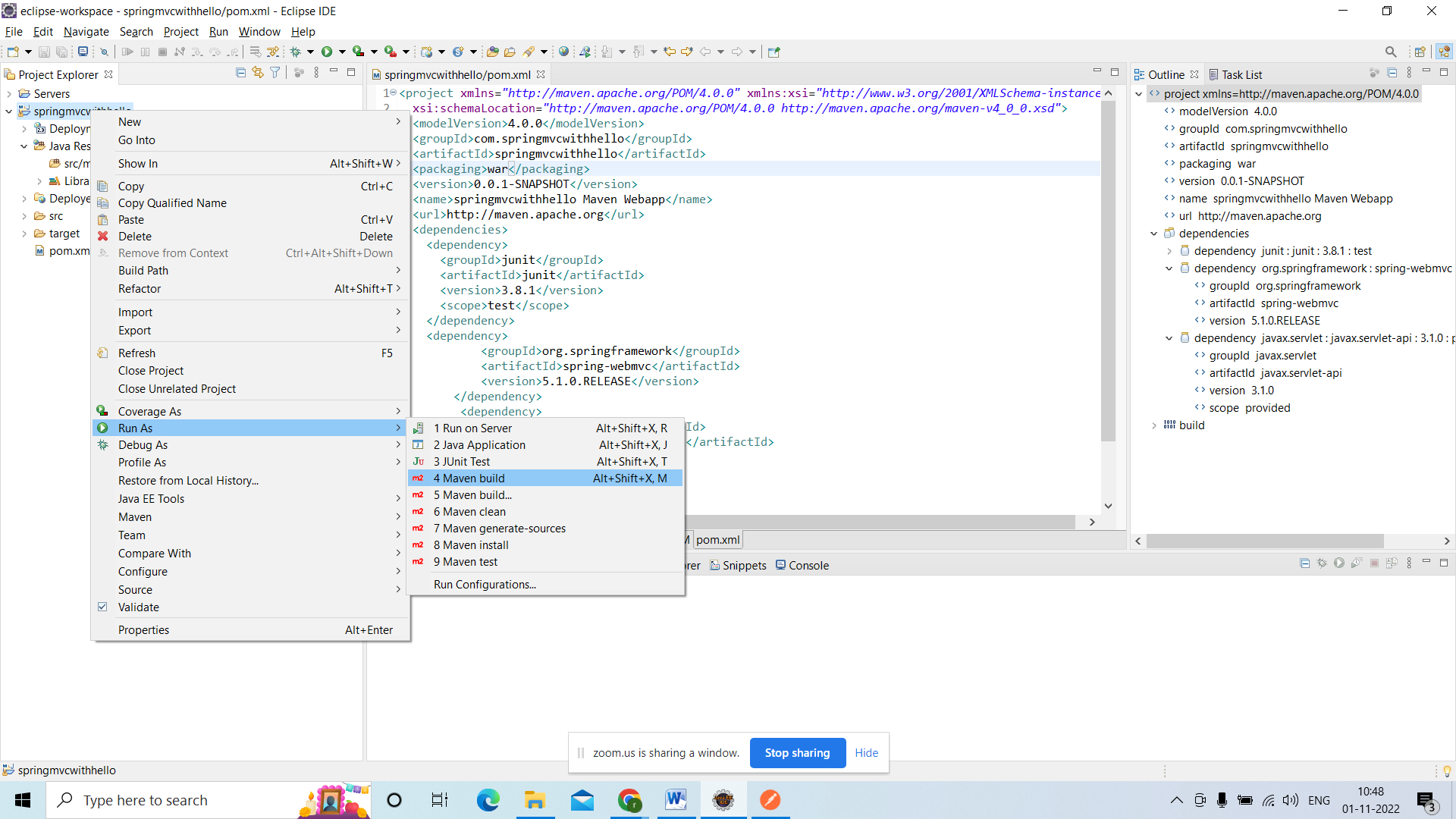
Groupid:package name

Artefact id:project name



After creating the project u need to add dependencies

U need to build the project



<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.springmvcwithhello</groupId>

<artifactId>springmvcwithhello</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>springmvcwithhello 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>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.1.0.RELEASE</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.1.0</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

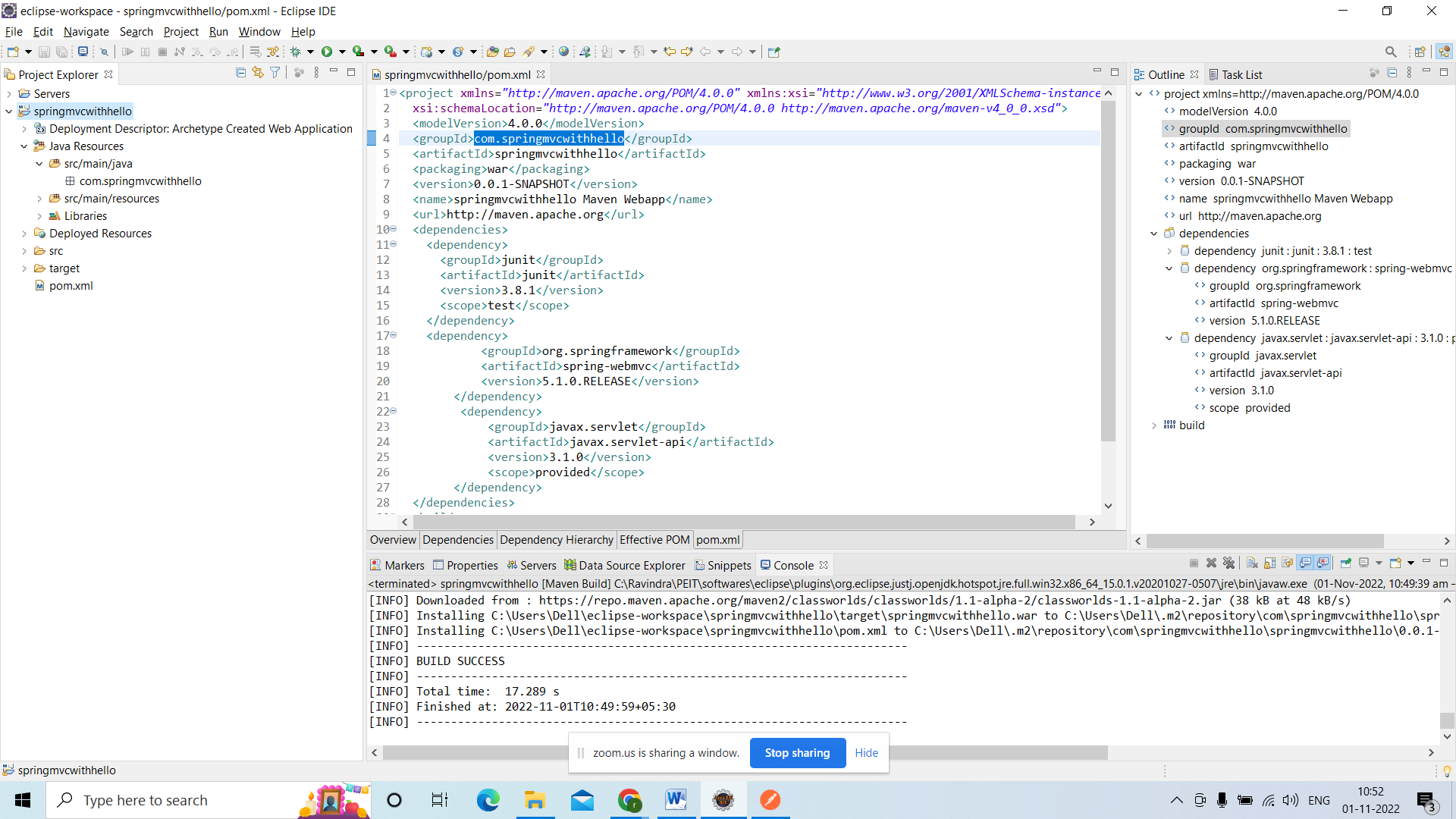
<finalName>springmvcwithhello</finalName>

</build>

</project>

In the goals u need to give clean install

Apply and run



**package** com.springmvcwithhello;

AppConfig.Java

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.web.servlet.config.annotation.EnableWebMvc;

@Configuration

@EnableWebMvc

@ComponentScan(basePackages = {

"com.springmvcwithhello"

})

**public** **class** AppConfig {

}

By using

@Configuration:

U r class will act as a configuration class.

@EnableWebMvc:

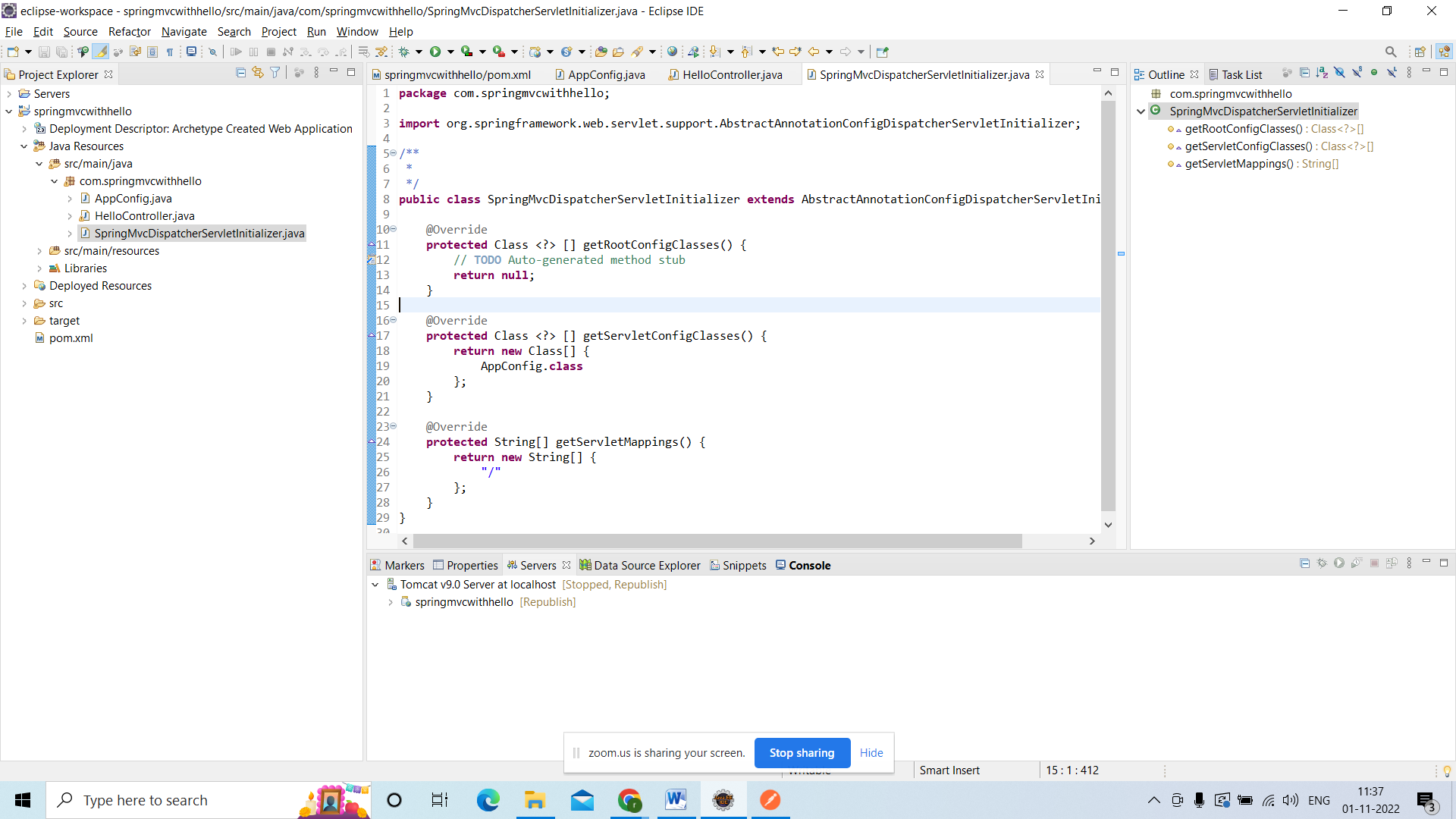
With the help of this annotation your project will support web application

@ComponentScan:

U have to declare the packages under this annotation

The classes which are there in side the packges objects going to be created automatically

Corresponding classes shoul annoated with @Component

disapatcher servlet class

HelloWorlController:

**package** com.springmvcwithhello;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.RestController;

@RestController

@RequestMapping("/hello")

**public** **class** HelloController {

@RequestMapping(value="/de", method=RequestMethod.***GET***)

**public** String demo()

{

**return** "HelloWorld";

}

@RequestMapping(value="/samp", method=RequestMethod.***GET***)

**public** String Sample()

{

**return** "Hello Sample";

}

}

@RestController:

With the help of this annotation u r class will act as a controller

@RequestMapping("/hello"):

With the help of this annotation it will map u r URL pattern to the class level or method level.

Github link:

<https://github.com/goud89105/springprojects>.