Coding Bootcamp

Servlets

TECHNOLOGY

Understanding Servlets

Learning Objectives

By the end of this lesson, you will be able to:

- Analyze the basic concepts of Servlets and its packages for developing efficient, secure, and scalable web applications
- Identify the steps to install Apache Tomcat, set up a reliable development environment, and deploy web applications
- Illustrate the concept of HTTP requests, methods, and parameters to enable proper handling of client-server interactions
- Demonstrate how to write the first Servlet program to build the skills necessary for debugging, configuration, and deployment

TECHNOLOGY

What Is a Servlet?

simplilearn

What Is a Servlet?

A Servlet is a Java programming language class that:

- ✓ Increases server capabilities that host programs accessed by the request-response programming model
- ✓ Extends the programs hosted by the web servers
- ✓ Describes HTTP-unique Servlet lessons

Why Use Servlets?

Here are the key purposes of servlets:

Platform independence	• Servlets run on any platform with a compatible web server.
Security	• They offer robust security mechanisms, including authentication and encryption.
Integration	• They can interact with various web technologies and databases.

Why Use Servlets?

Here are the key purposes of servlets:

Scalability	• They are designed to handle many concurrent requests and are suitable for high-traffic applications.
Robust API	• They provide a comprehensive API for handling HTTP requests and responses.
Performance	• They efficiently process client requests within the server environment.

Where to Use Servlets ?

A servlet is used in various scenarios like:

Web applications

- It acts as a controller in MVC frameworks.
- It manages server-side logic for dynamic content delivery.

Form data handling

- It handles data from user-submitted forms, like logins and registrations.
- It processes input for server-side validation and response.

Session tracking

- It maintains user-specific data across multiple pages via session management.
- It ensures continuity in user experience and data integrity.

Where to Use Servlets ?

A servlet is used in various scenarios like:

Data connectivity

- It performs CRUD operations with databases.
- It supports dynamic interactions in applications, such as e-commerce sites.

Technology integration

- It integrates with JSP, Spring, and Hibernate for enhanced functionality.
- It enables the creation of scalable and robust web applications.

Asynchronous processing

- It facilitates non-blocking data processing, improving performance in applications that handle long-running tasks.

Servlets: Packages

Java consists of the below packages that provide interfaces and instructions for writing Servlets.

javax.servlet

javax.servlet.http

Servlets must enforce the Servlet interface, which helps define life cycle strategies.

TECHNOLOGY

Apache Tomcat Installation: Setting Up the Environment

simplilearn

Configuring the Apache Tomcat

There are two ways to configure Apache Tomcat with the Eclipse IDE.

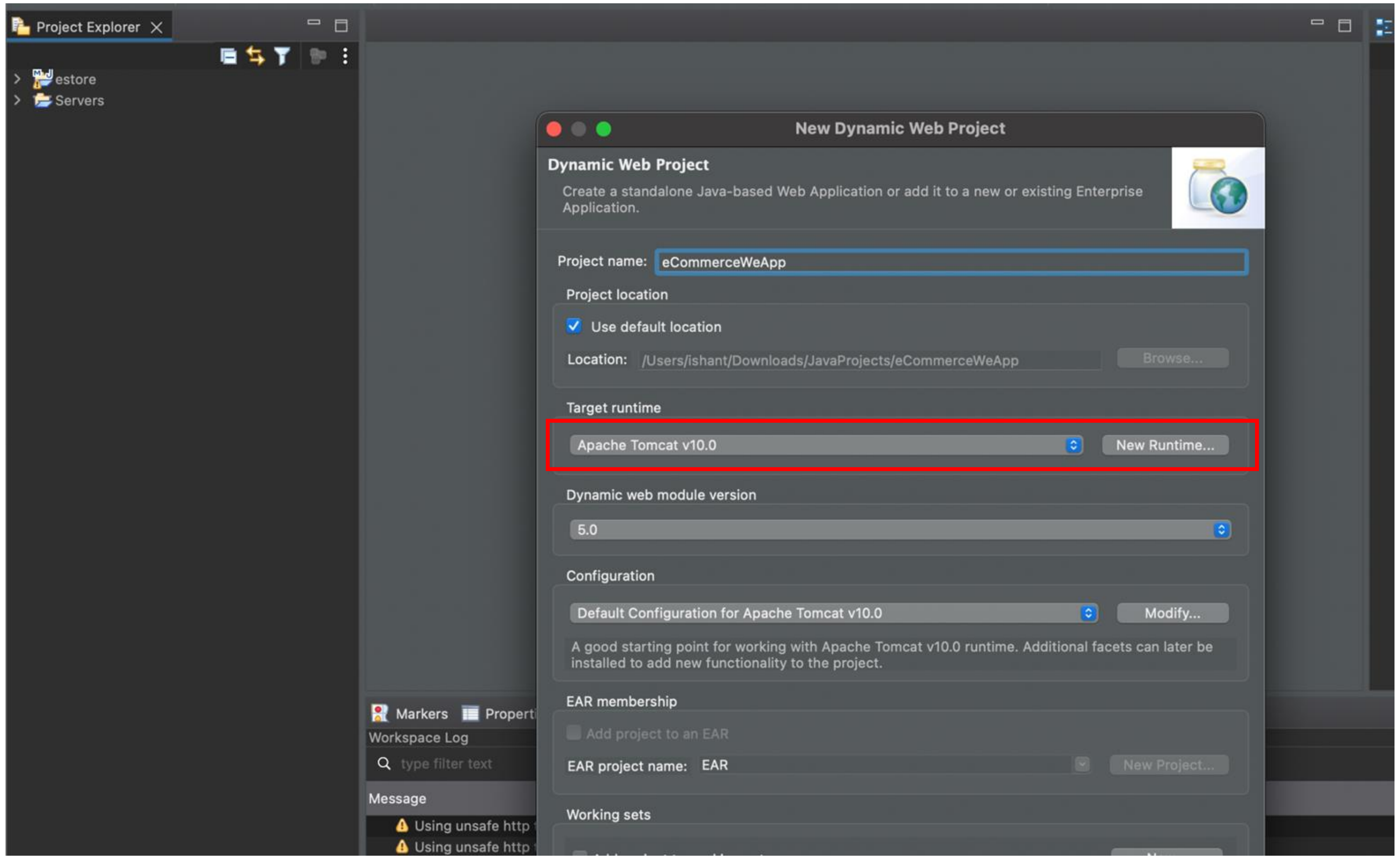
- ✓ Install the server runtime while creating the project.

Apache Tomcat

- ✓ Add the server from the server tab.

Configuring the Apache Tomcat: Steps

Install the server runtime while creating the project.



New Dynamic Web Project

Dynamic Web Project

Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.

Project name: eCommerceWeApp

Project location

Use default location: ✓

Location: /Users/ishant/Downloads/JavaProjects/eCommerceWeApp

Target runtime: Apache Tomcat v10.0

Dynamic web module version: 5.0

Configuration: Default Configuration for Apache Tomcat v10.0

A good starting point for working with Apache Tomcat v10.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

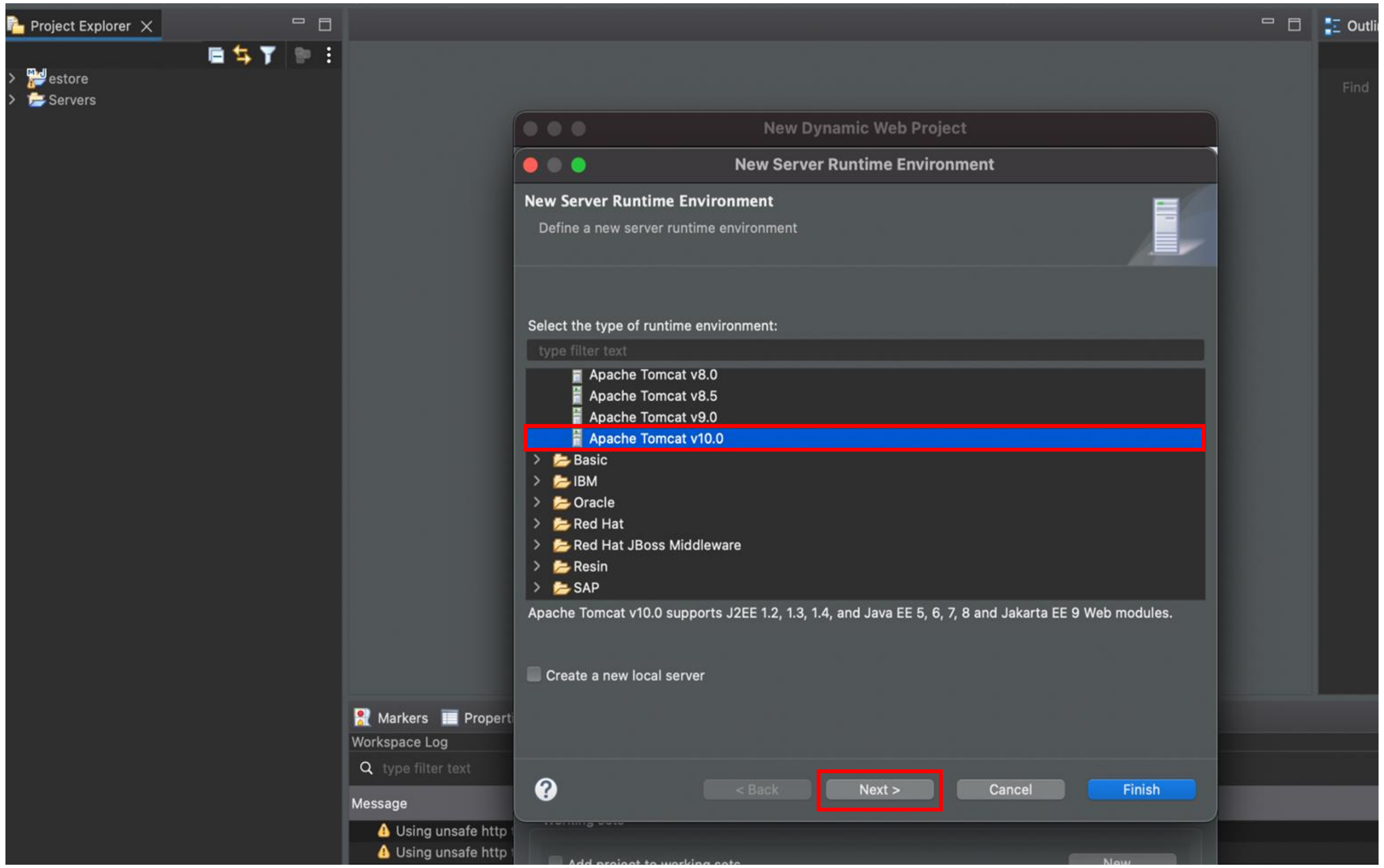
Add project to an EAR

EAR project name: EAR

Working sets

Configuring the Apache Tomcat: Steps

Select Apache with Tomcat's latest version. Now, click Next.



New Server Runtime Environment

Define a new server runtime environment

Select the type of runtime environment:

type filter text

- Apache Tomcat v8.0
- Apache Tomcat v8.5
- Apache Tomcat v9.0
- Apache Tomcat v10.0
- Basic
- IBM
- Oracle
- Red Hat
- Red Hat JBoss Middleware
- Resin
- SAP

Apache Tomcat v10.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, 7, 8 and Jakarta EE 9 Web modules.

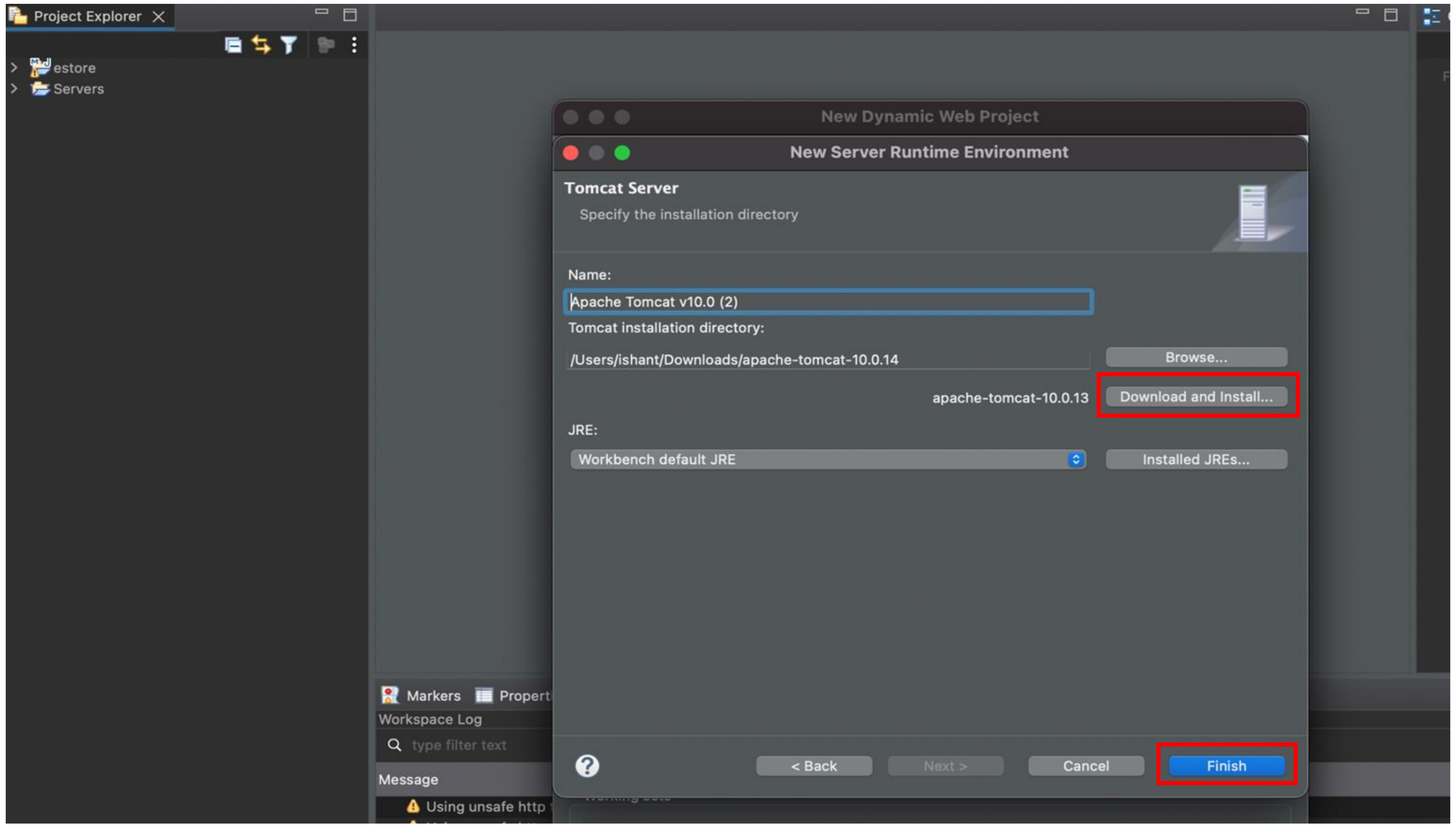
☐ Create a new local server

Using unsafe http

Using unsafe http

Configuring the Apache Tomcat: Steps

Click Download and Install, and then click Finish.



Name: Apache Tomcat v10.0 (2)

Tomcat installation directory: /Users/ishant/Downloads/apache-tomcat-10.0.14

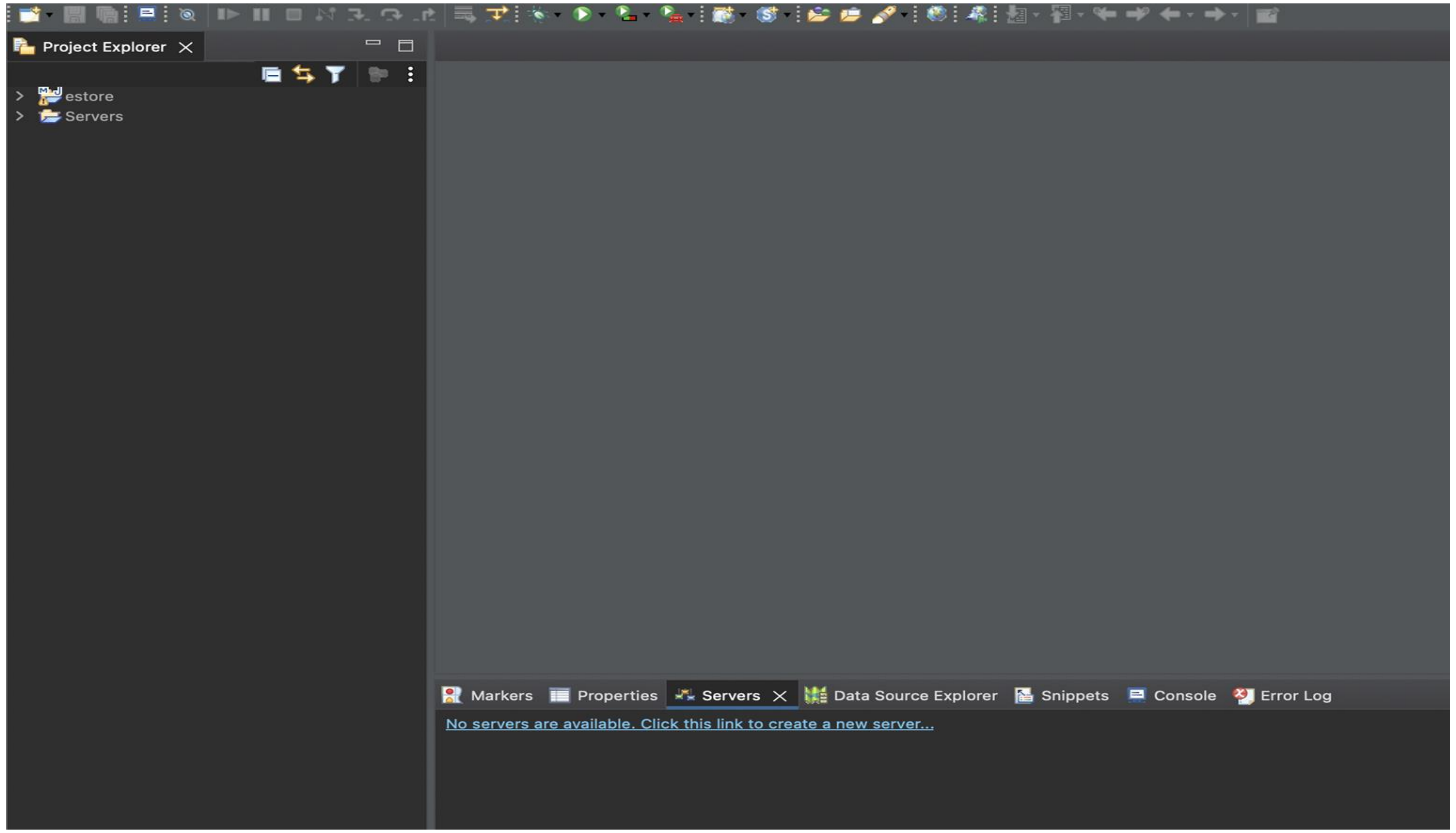
apache-tomcat-10.0.13

JRE: Workbench default JRE

Using unsafe http

Configuring the Apache Tomcat: Steps

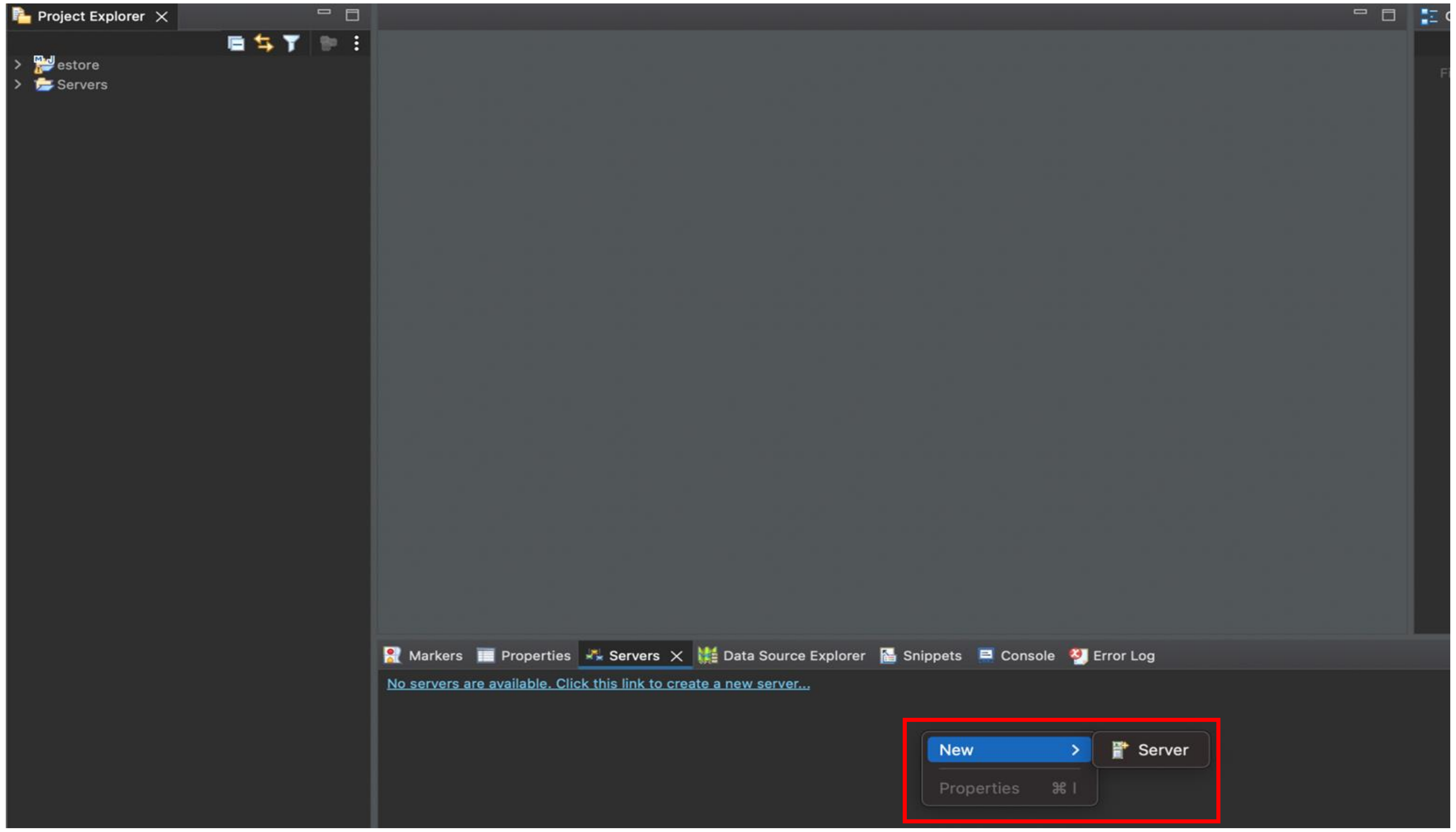
From the Server tab, add the server.



No servers are available. Click this link to create a new server...

Configuring the Apache Tomcat: Steps

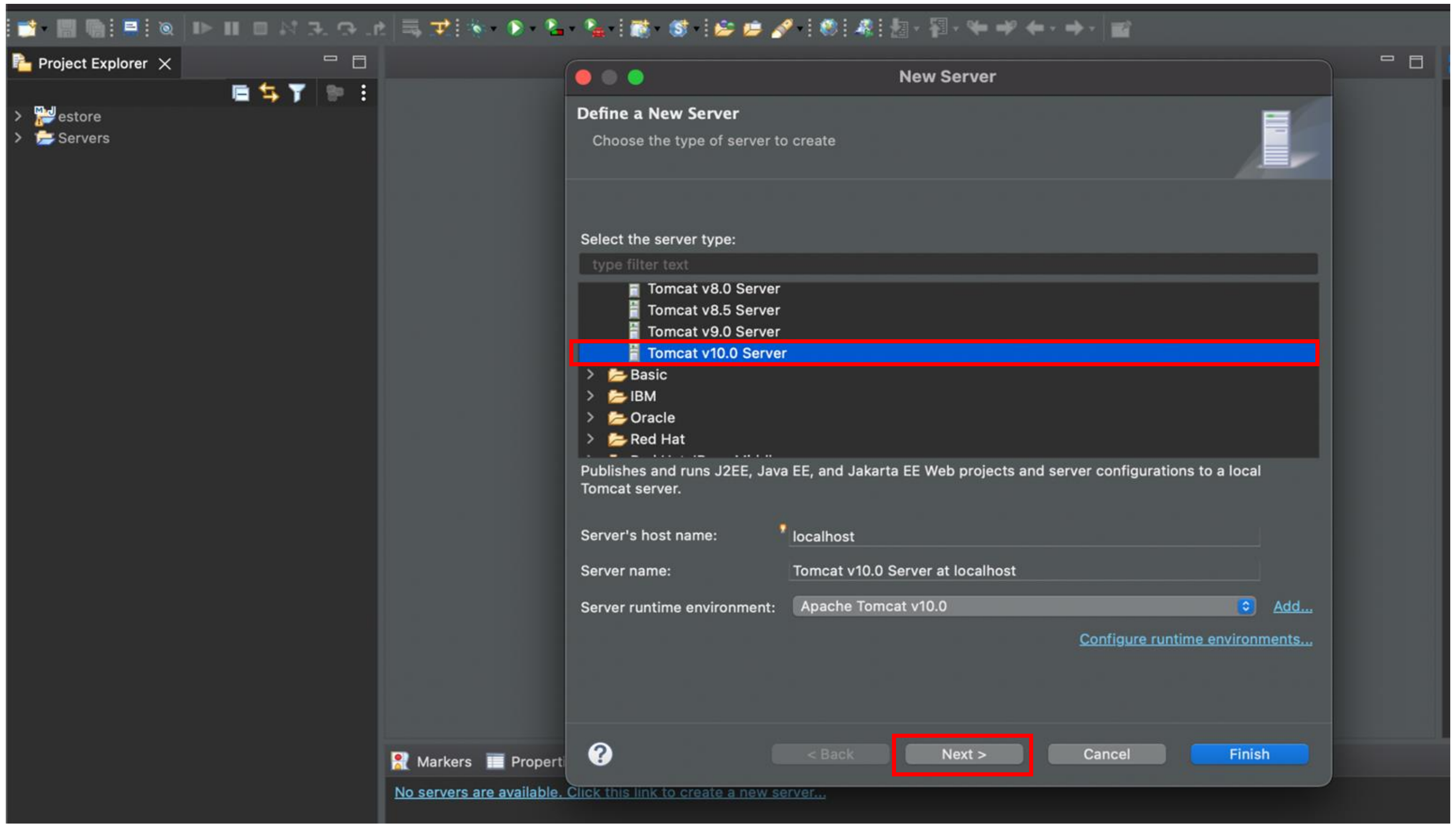
Click on the link to add a server, or right-click to add a new server.



No servers are available. Click this link to create a new server...

Configuring the Apache Tomcat: Steps

Choose the latest version of Apache Tomcat. Now, click Next.



Define a New Server

Choose the type of server to create

Select the server type:

type filter text

- Tomcat v8.0 Server
- Tomcat v8.5 Server
- Tomcat v9.0 Server
- Tomcat v10.0 Server
- Basic
- IBM
- Oracle
- Red Hat

Publishes and runs J2EE, Java EE, and Jakarta EE Web projects and server configurations to a local Tomcat server.

Server's host name: localhost

Server name: Tomcat v10.0 Server at localhost

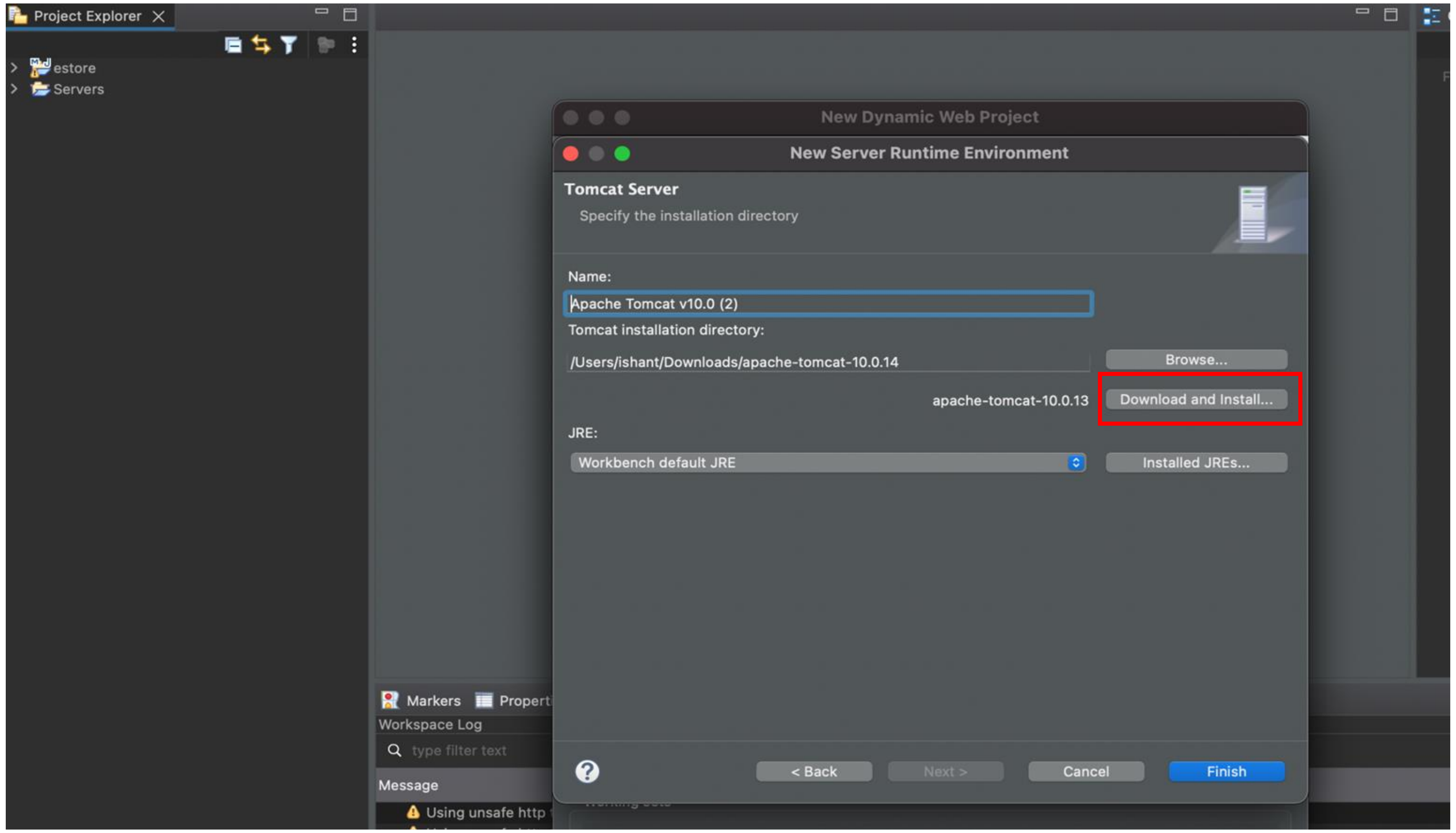
Server runtime environment: Apache Tomcat v10.0 Add...

Configure runtime environments...

No servers are available. Click this link to create a new server...

Configuring the Apache Tomcat: Steps

Click on the **Download and Install** option, and then install a new server.



Tomcat Server

Specify the installation directory

Name:

Apache Tomcat v10.0 (2)

Tomcat installation directory:

/Users/ishant/Downloads/apache-tomcat-10.0.14

apache-tomcat-10.0.13

JRE:

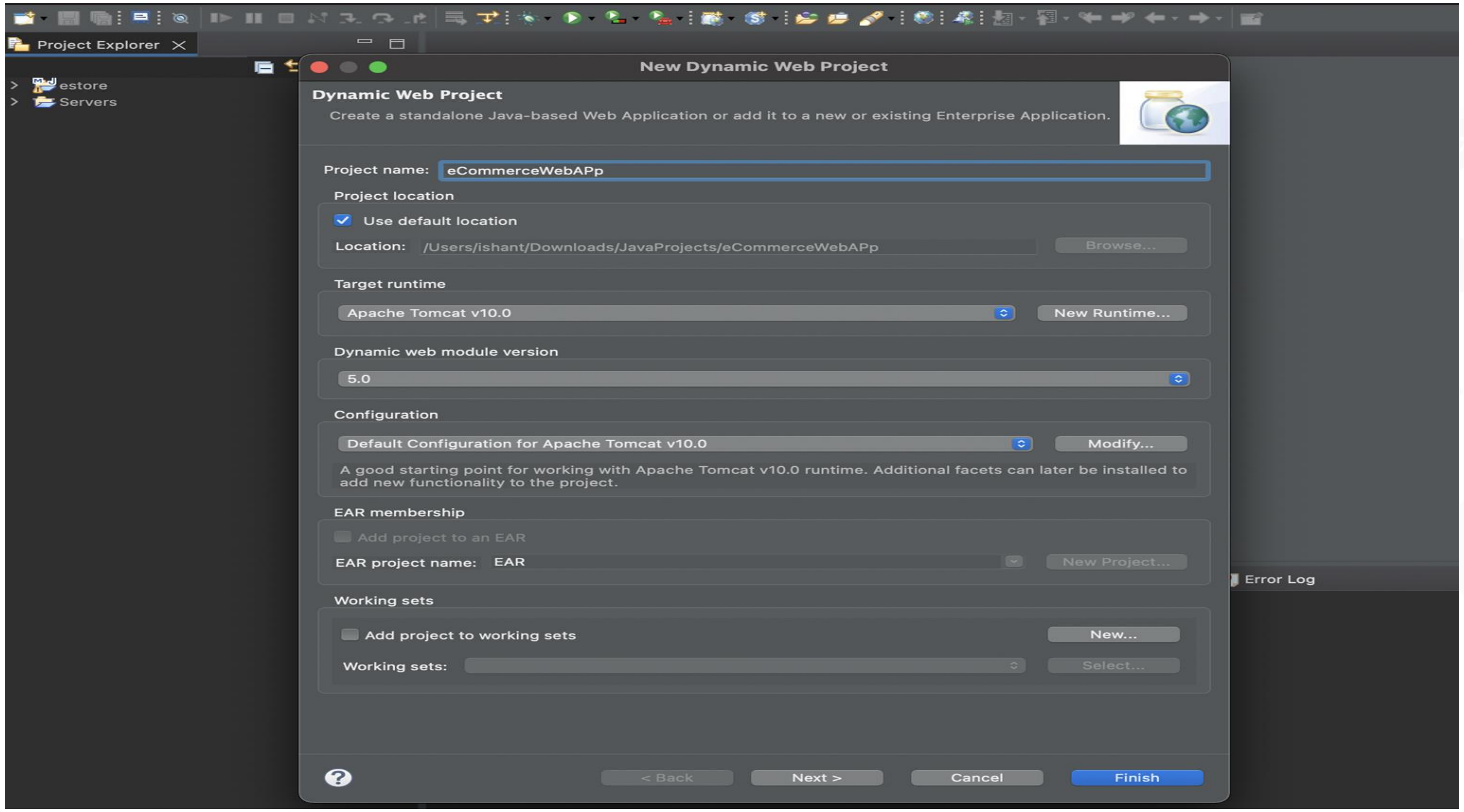
Workbench default JRE

TECHNOLOGY

Writing First Servlet Program

Writing First Servlet Program: Steps

Create a Dynamic Web Project or a Maven Web App Project.


New Dynamic Web Project

Dynamic Web Project

Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.

Project name: eCommerceWebAPp

Project location

✓ Use default location

Location: /Users/ishant/Downloads/JavaProjects/eCommerceWebAPp

Target runtime: Apache Tomcat v10.0

Dynamic web module version: 5.0

Configuration: Default Configuration for Apache Tomcat v10.0

A good starting point for working with Apache Tomcat v10.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

Add project to an EAR

EAR project name: EAR

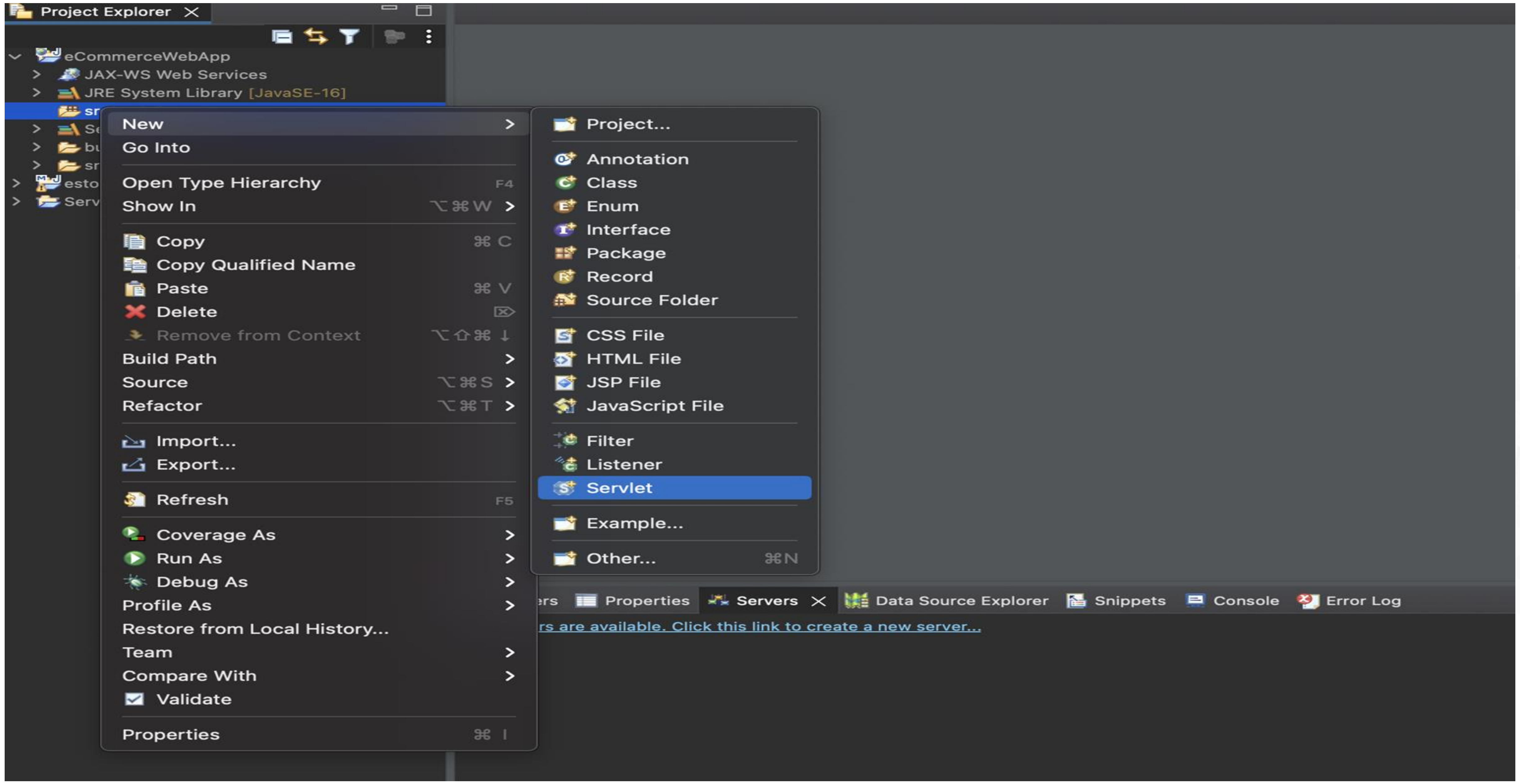
Working sets

Add project to working sets

Working sets:

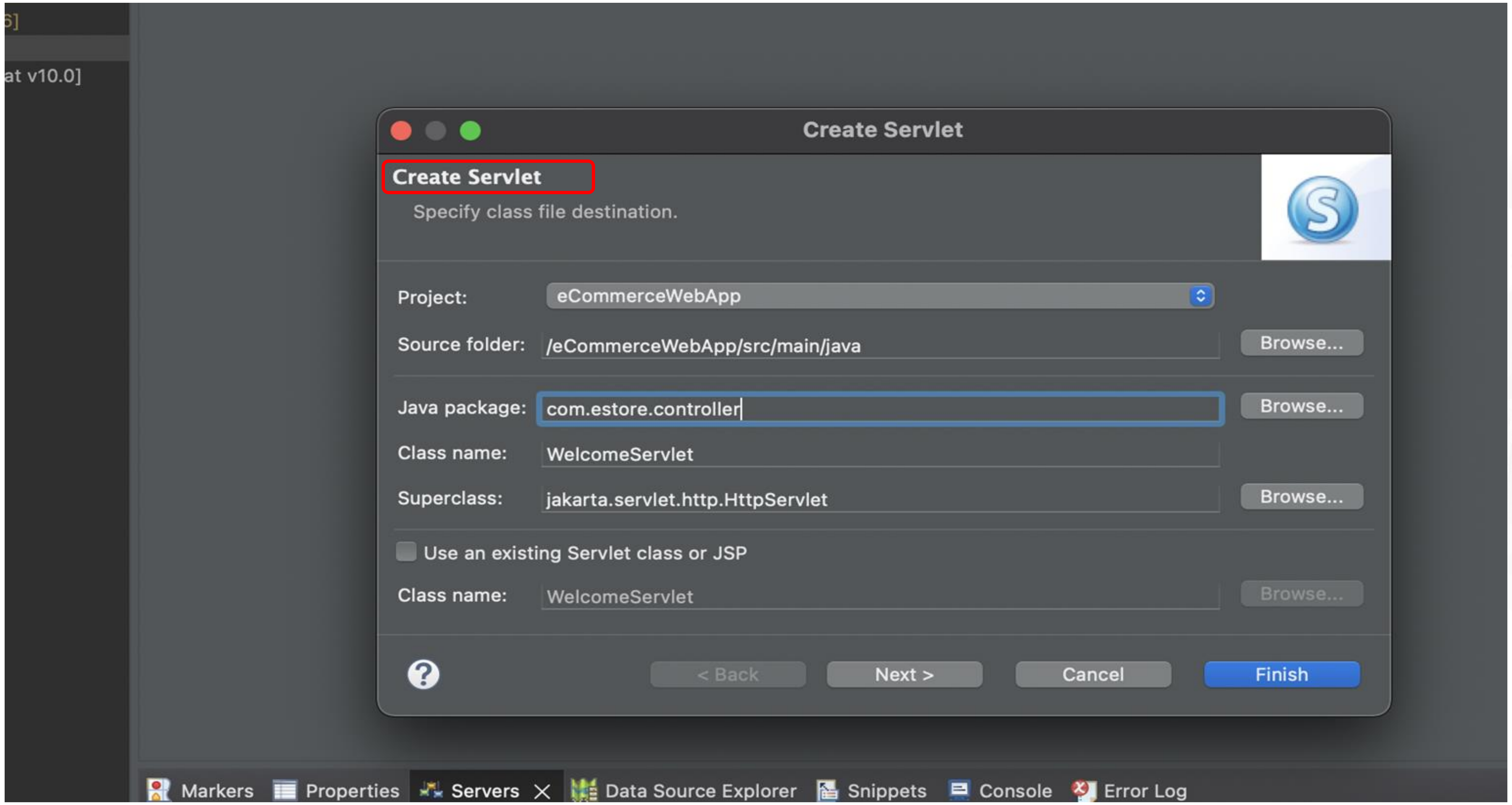
Writing First Servlet Program: Steps

Create a Servlet in the project source directory.



Writing First Servlet Program: Steps

Create the Servlet.



Create Servlet

Specify class file destination.

Project: eCommerceWebApp

Source folder: /eCommerceWebApp/src/main/java

Java package: com.estore.controller

Class name: WelcomeServlet

Superclass: jakarta.servlet.http.HttpServlet

Use an existing Servlet class or JSP

Class name: WelcomeServlet

Servlet Source Code

Here's a code example to create WelcomeServlet:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class WelcomeServlet extends HttpServlet {
 public void doGet(HttpServletRequest request,
 HttpServletResponse response)
 throws ServletException, IOException {
 PrintWriter out = response.getWriter();.
 out.println("Hello");
 out.println("My");
 out.println("Name");
 out.println("Is");
 out.println("Ryan");
 out.println("Thanks for reaching us !");
 out.println("Have a nice day :) ");
 }
}
```

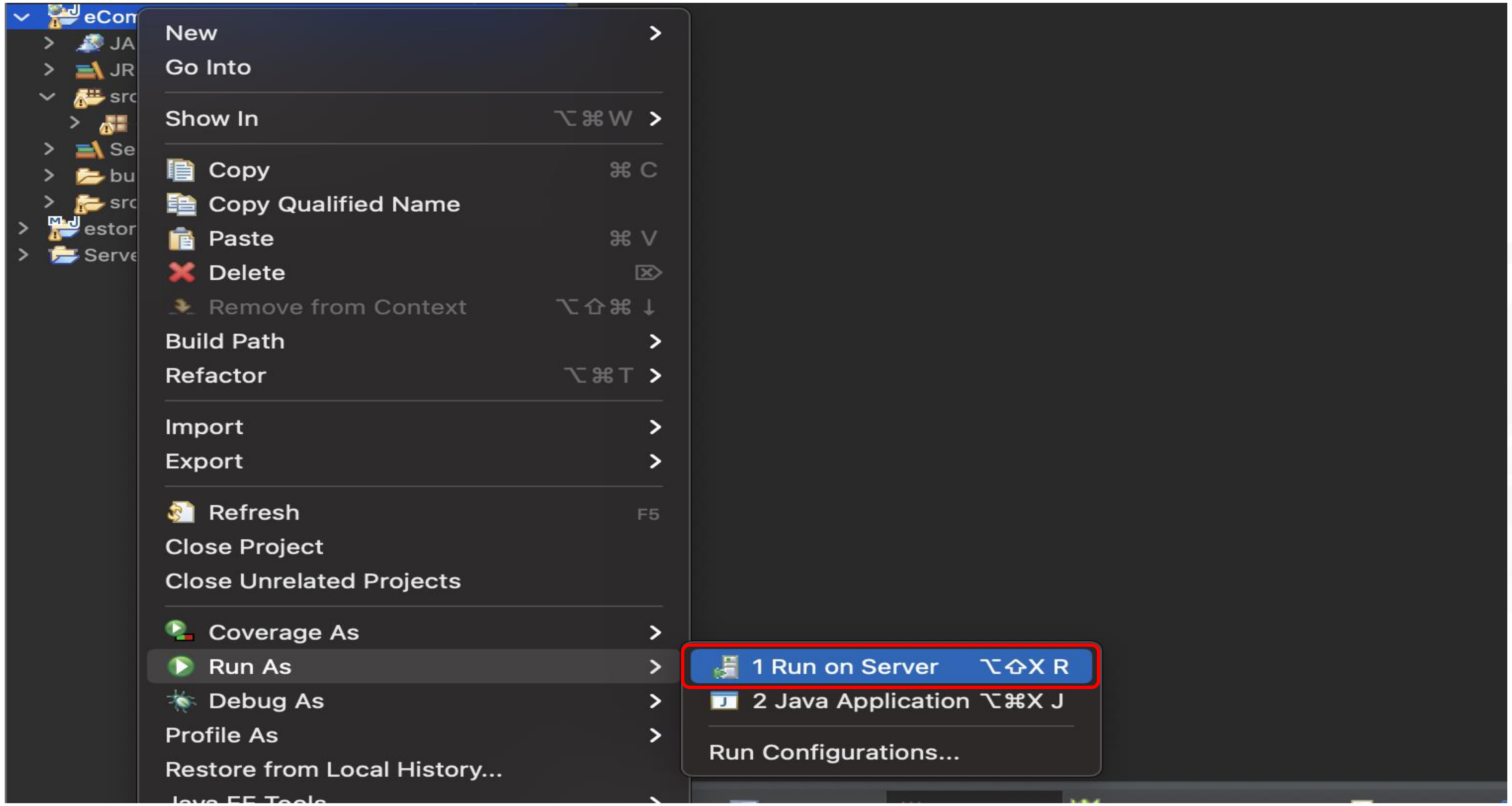
It shows the easy Servlet called WelcomeServlet.

Compiling the Code

To compile the Servlet source code, include the path of the servlet.jar file in the CLASSPATH environment.

Compiling the Code

Execute the project from the Eclipse IDE.



Compiling the Code

Compile servlet code from the command prompt by compiling through Javac.

Exchange the directory for a work listing and use the code:

```
javac -d ..\WEB-INF\classes\WelcomeServlet.java
```

In the case of Linux/UNIX, / is used to separate the listing from the subdirectory.

```
javac -d ../WEB-INF/classes/WelcomeServlet.java
```

Creating a Dynamic Web Project in Eclipse IDE with Servlet

Problem Statement:

You are given a project to create a dynamic web project using the Eclipse IDE and make the Servlet work with responses.

Outcome:

By completing this project, learners will gain a comprehensive understanding of dynamic web development using Eclipse IDE and the practical implementation of Servlets to handle responses. They will emerge equipped with the skills to confidently design, develop, and deploy dynamic web applications, enhancing their proficiency in modern web development technologies.

Note: Refer to the demo document for detailed steps: 01_Creating_a_Dynamic_Web_Project_in_Eclipse_IDE_with_Servlet

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create a new project
2. Create a server
3. Create a new HTML file
4. Create a Servlet
5. Run the project and handle the error
6. Make the Servlet work with a response

Creating an HTML Page with a Form to Send Request to a Servlet

Problem Statement:

You are given a project to create an HTML page that contains a login and registration form for sending a request to a Servlet.

Outcome:

By completing this project, learners will gain a comprehensive understanding of HTML forms, client-server communication via Servlets, and the fundamentals of web development. They will develop practical skills in creating interactive web pages with login and registration functionality, enabling them to apply this knowledge to real-world scenarios effectively.

Note: Refer to the demo document for detailed steps: 02_Creating_an_HTML_Page_with_a_Form_to_Send_Request_to_a_Servlet

Assisted Practice: Guidelines

Steps to be followed are:

1. Create a login form in HTML
2. Create a new Servlet for the login form
3. Create a method in the login Servlet for an HTTP Servlet request
4. Create a form to register users
5. Create a new Servlet for the registration form
6. Create a method in the registered Servlet for an HTTP Servlet request

HTTP Requests

HTTP Request: Overview

The HttpServlet class request method uses two parameters:

javax.servlet.http.HttpRequest

javax.servlet.http.HttpResponse

HTTP Requests: Uses

The purpose of the HttpRequest item is to represent the HTTP request that the browser sends to the internet application.

For this reason, HttpRequest needs to consume the data sent by the browser. This is done with the help of the methods present in HttpRequest.

The HttpRequest object contains methods.

HTTP Requests: Methods

The first method is the request parameters. This parameter can be sent from the browser along with the request.

HTTP Request: Methods

The example of the HttpServlet.doGet() method is:

```
protected void doGet(
    HttpServletRequest request,
    HttpServletResponse response)
      throws ServletException, IOException {

}
```

HTTP Request: Parameters

The request parameters are usually sent as part of the URL inside the query string or as part of the frame of an HTTP request.

Example:

```
http://example.com/page1.html?message=Thanks&name=John
```

The query string is part of the **URL: ?message=Thanks &name=John** element that displays two parameters with parameter values.

HTTP Request: Parameters

The entry for these parameters in the HttpRequest object is:

```
protected void doGet(
HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {

String message = request.getParameter("message");
String name = request.getParameter("name");

}
```

If there is no parameter provided with the given name, then null is returned as the response.

Service doGet() and doPost() in Servlet

Problem Statement:

You are given a project to explore the doGet() and doPost() services in Servlet.

Outcome:

By actively engaging with the project to explore the `doGet()` and `doPost()` services in Servlet, learners will acquire a comprehensive understanding of how these methods operate within Servlets. They will gain practical experience in handling HTTP GET and POST requests effectively, enhancing their proficiency in Java web development.

Note: Refer to the demo document for detailed steps:
03_Service_doGet_and_doPost_in_Servlet

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Perform the HTTP POST method
2. Perform the HTTP GET method
3. Create a doGet() method
4. Create a doPost() method

Life Cycle of Servlet

Problem Statement:

You are given a project to understand the life cycle of a Servlet and how the init(), service(), and destroy() methods work.

Outcome:

By completing this project, you will gain a comprehensive understanding of the Servlet life cycle, including the initiation, service handling, and termination phases. Through hands-on exploration of the init(), service(), and destroy() methods, you'll emerge with practical insights into how Servlets function in web applications, equipping you with valuable skills for developing robust and efficient web solutions.

Note: Refer to the demo document for detailed steps: 04_Life_Cycle_of_Servlet

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create the init and destroy method
2. Make changes in internal service
3. Create a hyperlink for Hello Servlet
4. Run code to check how the life cycle works

TECHNOLOGY

Setting Up Database Connection

Setting Up Database Connection

Here’s the syntax to set up a database connection in Servlets:

```
try (Connection connect =
DriverManager.getConnection(
     "jdbc:<class name>",
     "username",
     "password", “email”)) {
     /* we use the connection here */
}  //
```

Setting Up Database Connection

Using the doGet, doPost, or service method, one can extract the data from the Client Http Request using the HttpServletRequest Object and then create the Model Object.

Product Object can store the data as shown below:

```
public class ProductController extends HttpServlet {

      protected void service(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException
{
            Product IO Exception = new Product();
            product.code =
Integer.parseInt(request.getParameter("txtCode"));
            product.name = request.getParameter("txtName");
            product.price =
Integer.parseInt(request.getParameter("txtPrice"));
System.out.println("[Product Servlet] Product Details: "+product);
```

Servlet Configuration

Problem Statement:

You are given a project to demonstrate the configuration of Servlets.

Outcome:

By completing this project, learners will gain a comprehensive understanding of Servlet configuration, including mapping Servlets to URLs, handling HTTP requests and responses, and managing Servlet lifecycle. This hands-on experience will empower you to efficiently deploy Servlets in real-world web applications, enhancing your proficiency in Java web development.

Note: Refer to the demo document for detailed steps: 05_Servlet_Configuration

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create a new servlet
2. Create a hyperlink and run an application to check the output
3. Create a getServerConfig method
4. Create and extract parameters
5. Create multiple parameters and methods

Key Takeaways

- Servlet increases server capabilities that host programs accessed by the request-response programming model.
- Java consists of packages that provide interfaces and instructions for writing Servlets.
- Servlets must enforce the Servlet interface, which helps define life cycle strategies.
- The purpose of the HttpRequest item is to represent the HTTP request that the browser sends to the internet application.

TECHNOLOGY

Thank You

simplilearn