# Lesson 01 Demo 01

# Creating a Dynamic Web Project in Eclipse IDE with Servlet

**Objective**: To create a dynamic web project using Eclipse IDE and make the Servlet work with responses

**Tools Required:** Eclipse IDE

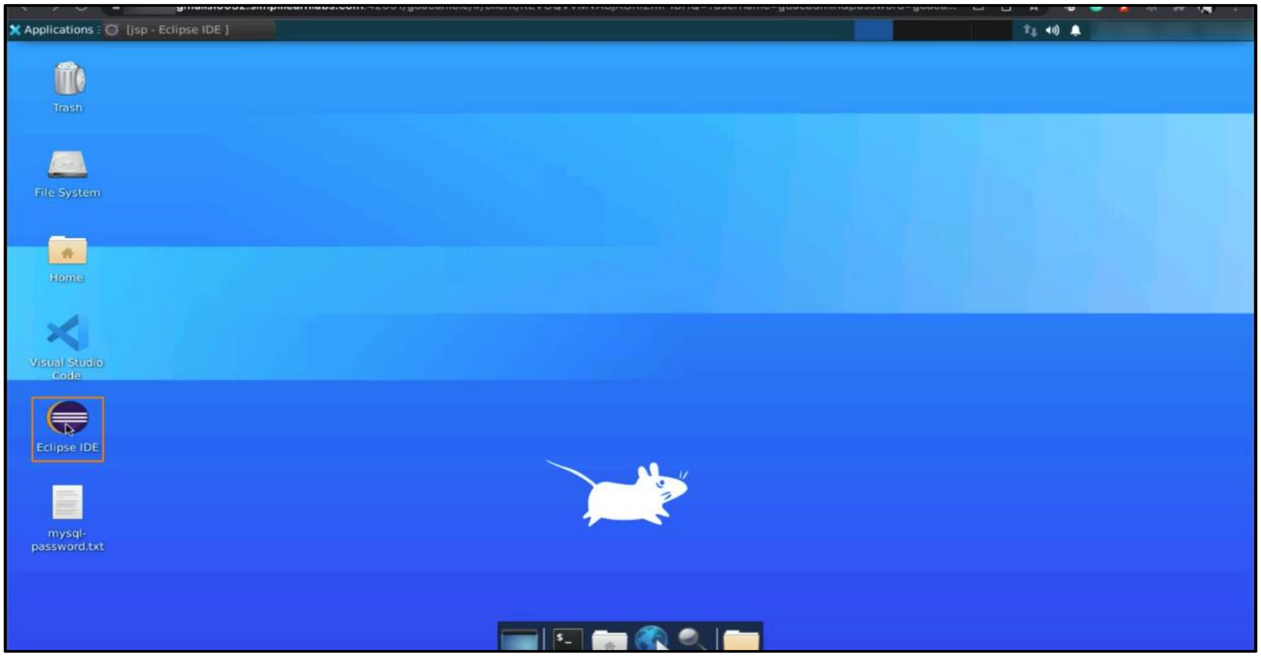
**Prerequisites:** None

**Steps to be followed:**

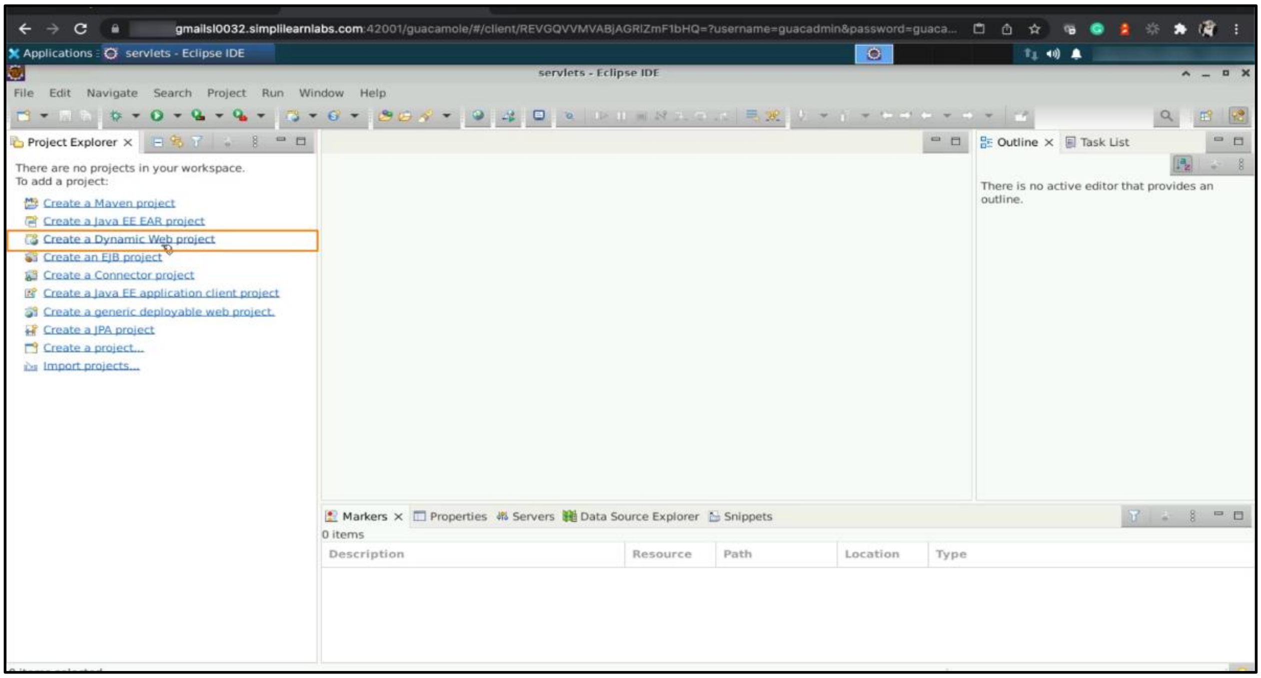
1. Create a new project
2. Create a server
3. Create a new HTML file
4. Create a Servlet
5. Run the project and handle the error
6. Make the Servlet work with a response

## Step 1: Create a new project

1.1 Open **Eclipse IDE**



1.2 Click on the **Create a Dynamic Web Project** option



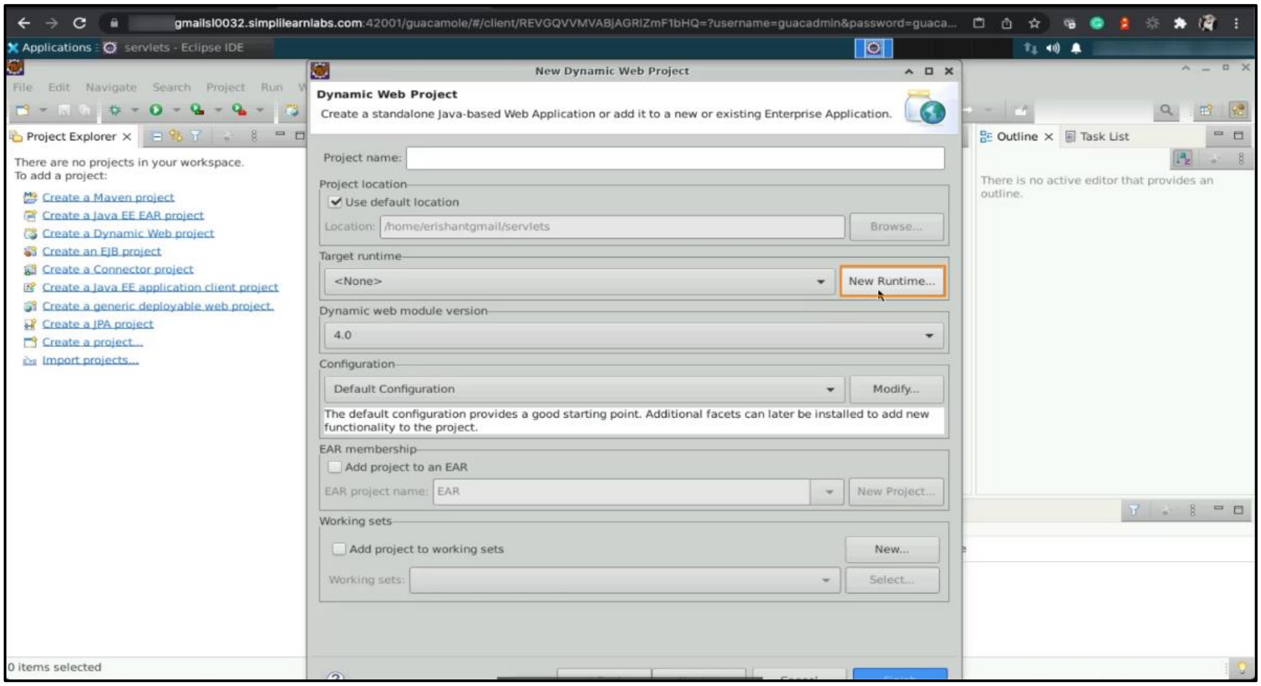
There are no projects in your workspace.
To add a project:

- 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.

0 items

1.3 Specify the **Target Runtime**. Since it is empty, click on the **New Runtime** option



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

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...

Project name:

Project location
☑ Use default location
Location: /home/erishantgmail/servlets

Target runtime
<None>

Dynamic web module version
4.0

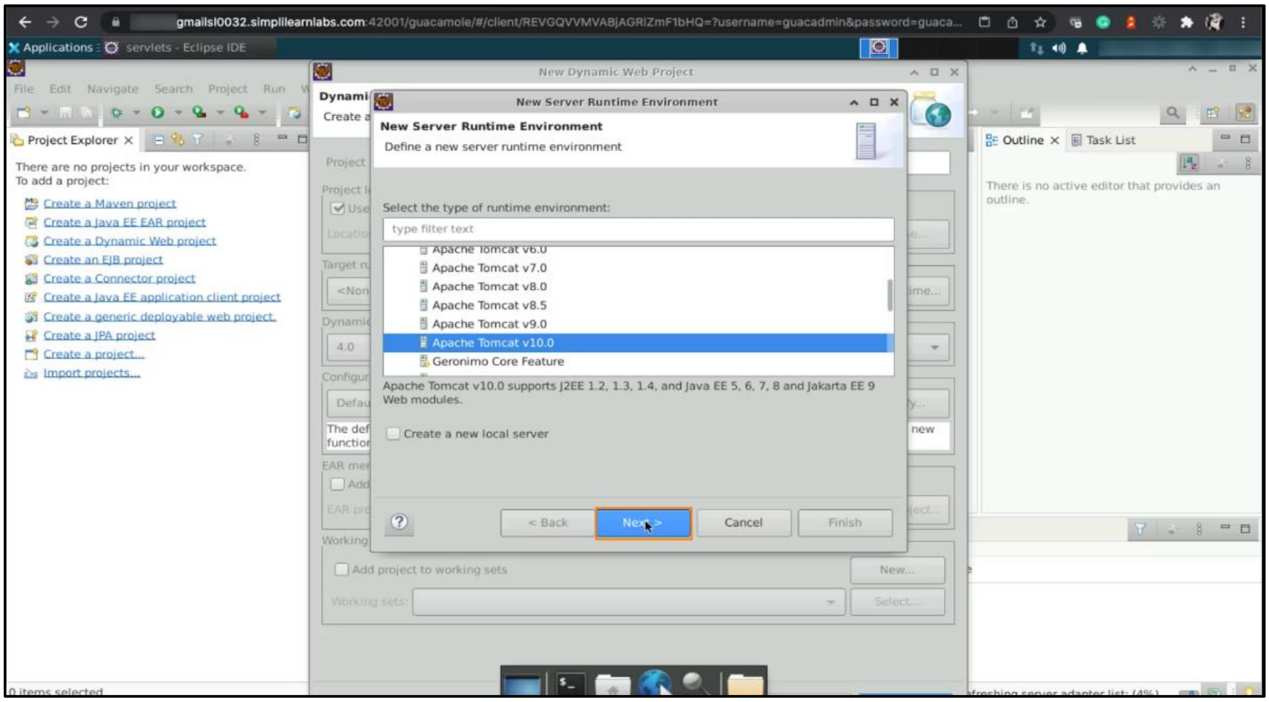
Configuration
Default Configuration
The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership
□ Add project to an EAR
EAR project name: EAR

Working sets
□ Add project to working sets
Working sets:

There is no active editor that provides an outline.

1.4 Select the **Apache Tomcat version v10.0** and click on **Next**



New Server Runtime Environment
Define a new server runtime environment

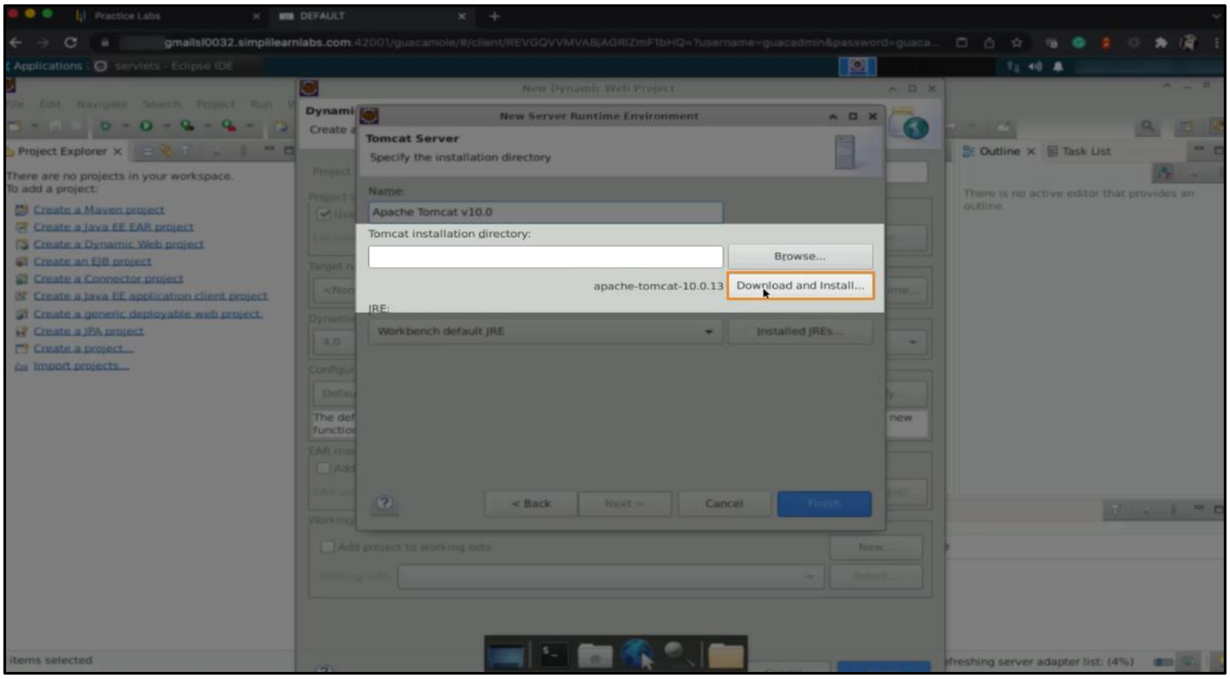
Select the type of runtime environment:
type filter text

- Apache Tomcat v6.0
- Apache Tomcat v7.0
- Apache Tomcat v8.0
- Apache Tomcat v8.5
- Apache Tomcat v9.0
- Apache Tomcat v10.0
- Geronimo Core Feature

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

1.5 Click on the **Download and Install** button



New Server Runtime Environment

Tomcat Server

Specify the installation directory

Name:

Apache Tomcat v10.0

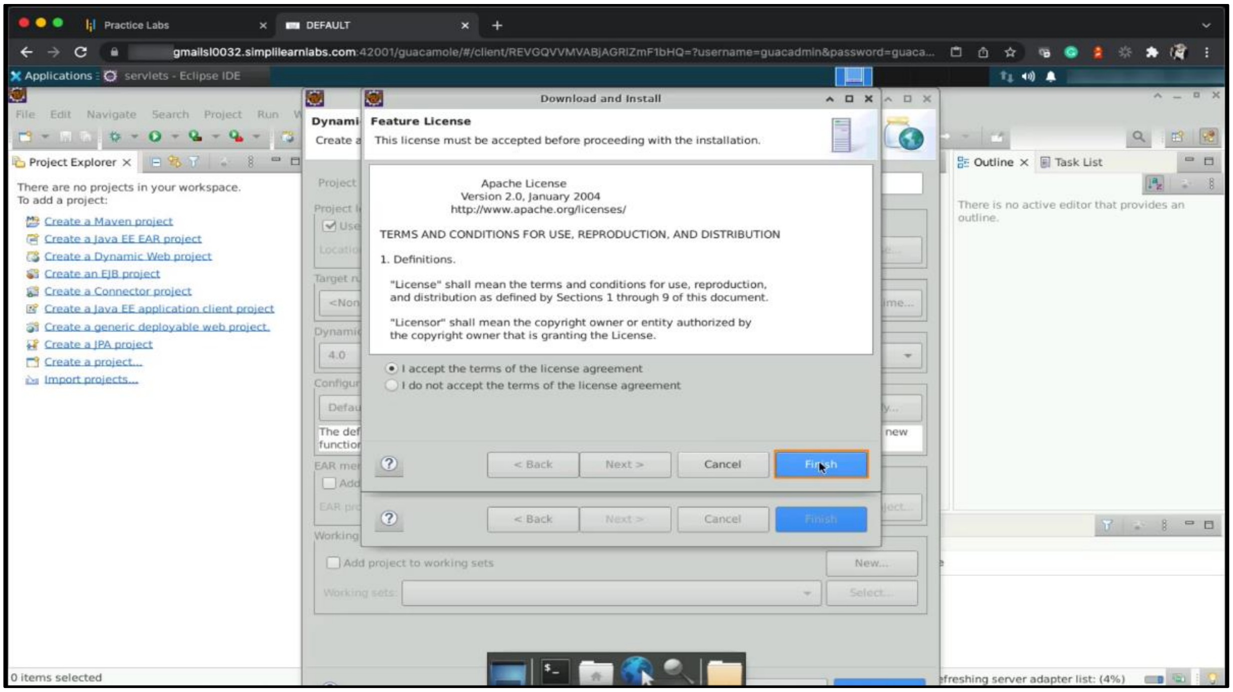
Tomcat installation directory:

apache-tomcat-10.0.13

JRE:

Workbench default JRE

1.6 Accept the terms and conditions, and then click on **Finish**



Download and Install

Feature License

This license must be accepted before proceeding with the installation.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

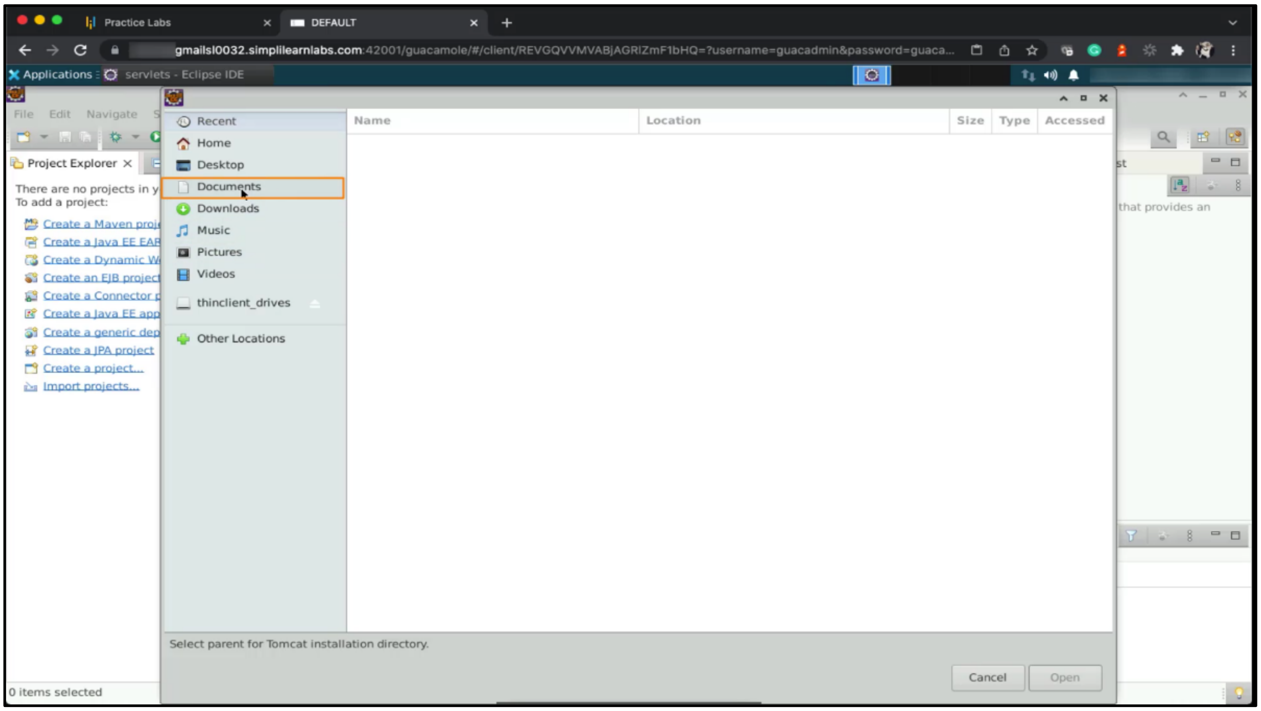
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

● I accept the terms of the license agreement

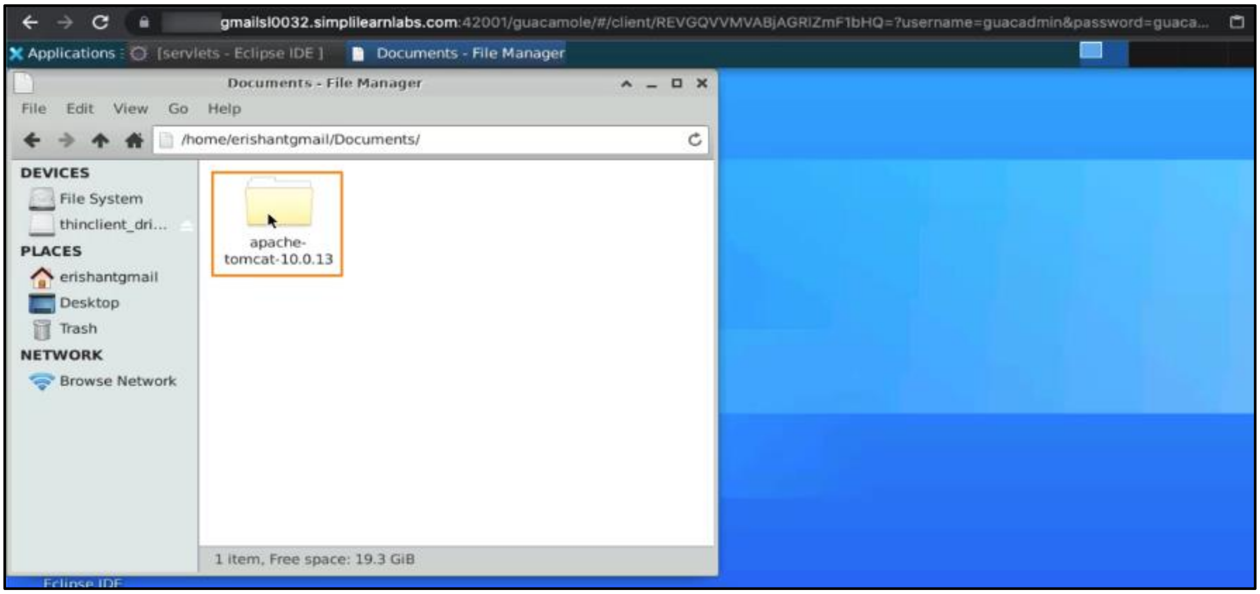
○ I do not accept the terms of the license agreement

1.7 Select **Documents**, and then click on **Open**



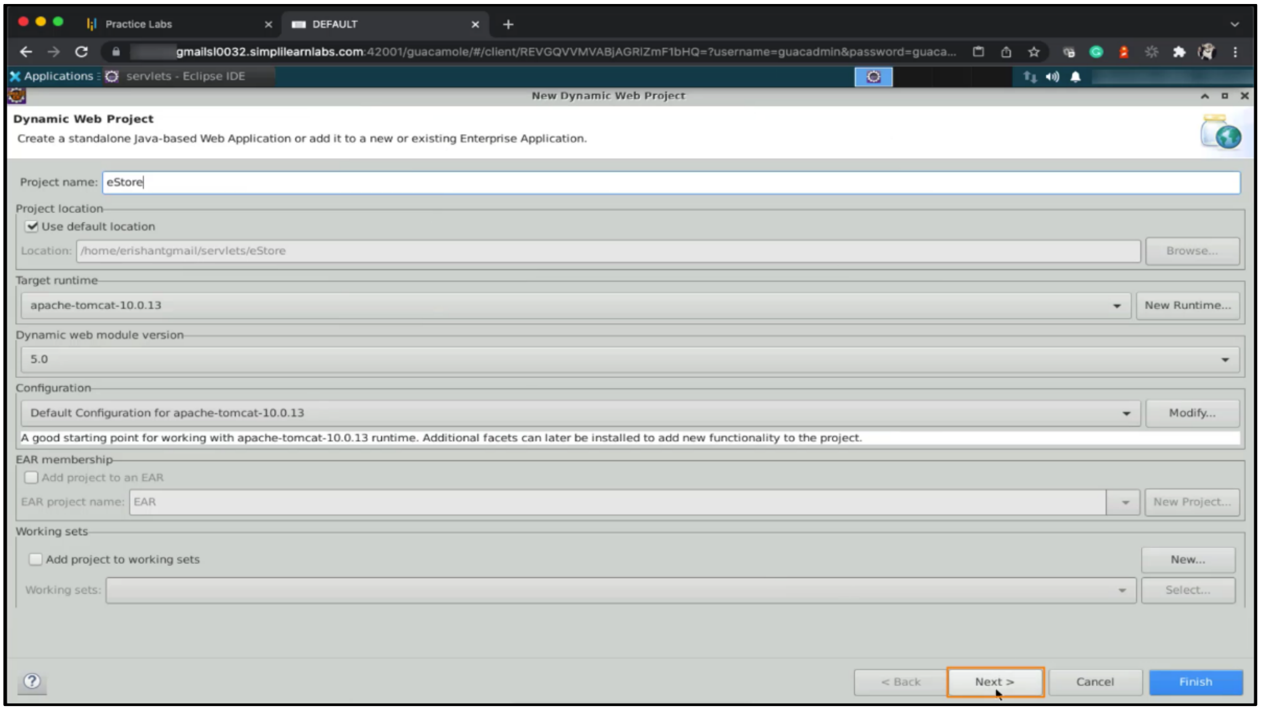
Select parent for Tomcat installation directory.

1.8 To check if Tomcat is installed, go to the **Documents** folder, and you can check if the **apache-tomcat-10.0.13** file is available



1 item, Free space: 19.3 GiB

1.9 Go back to **Eclipse IDE** and fill in the details, like the project name. Enter the name as **eStore** and click on **Next**



[redacted]gmailsl0032.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guaca...

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: eStore

Project location

☑ Use default location

Location: /home/erishantgmail/servlets/eStore

Target runtime

apache-tomcat-10.0.13

Dynamic web module version

5.0

Configuration

Default Configuration for apache-tomcat-10.0.13

A good starting point for working with apache-tomcat-10.0.13 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

□ Add project to an EAR

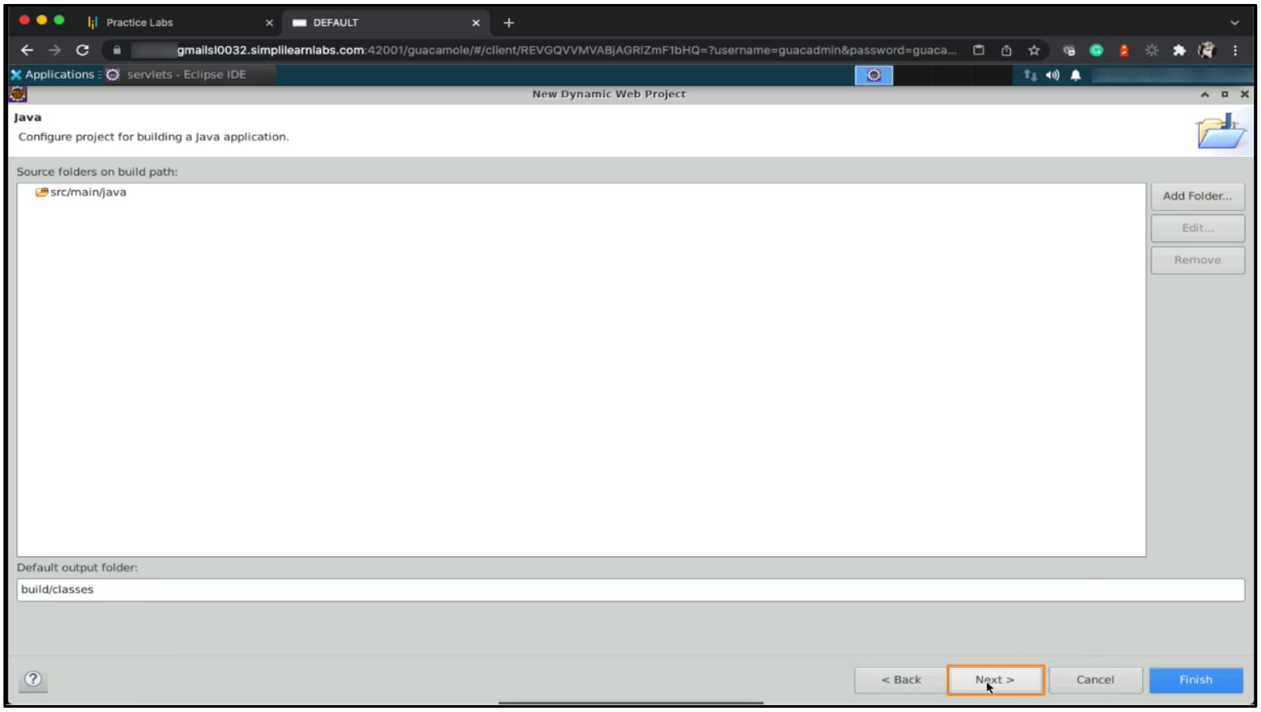
EAR project name: EAR

Working sets

□ Add project to working sets

Working sets:

You can see the source directory created. Click on **Next**



[redacted]gmailsl0032.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guaca...

New Dynamic Web Project

**Java**

Configure project for building a Java application.

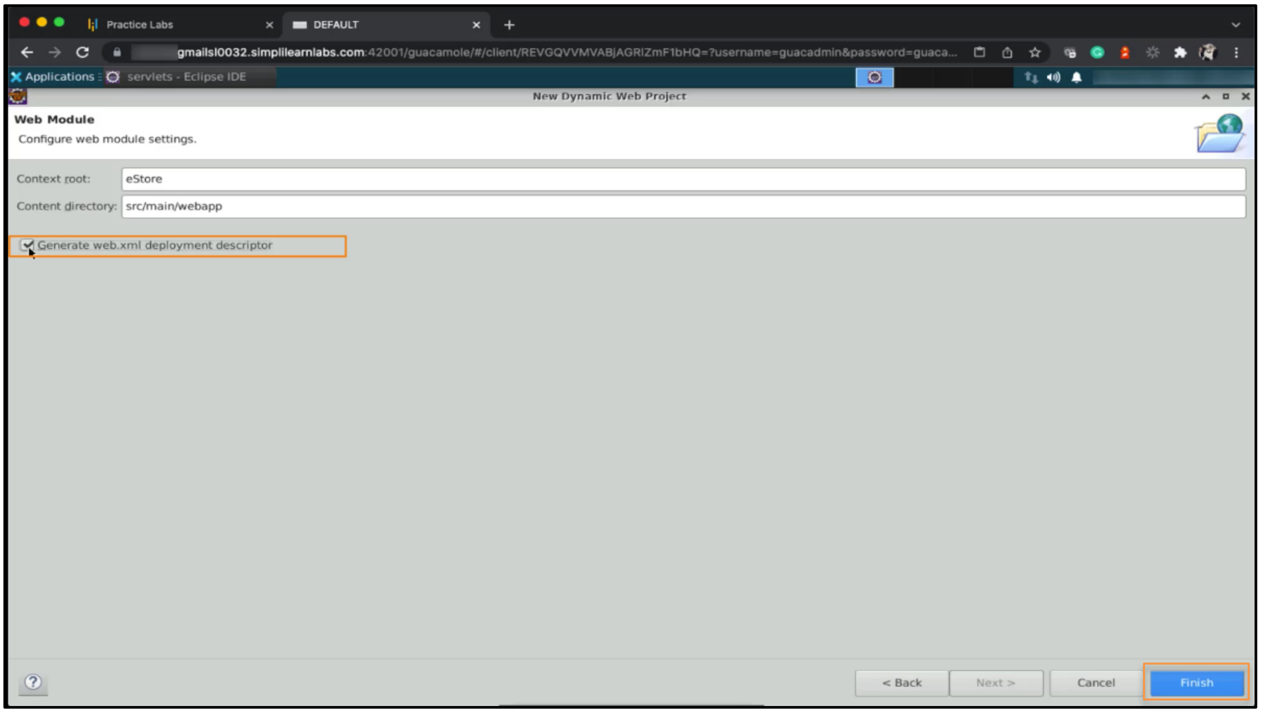
Source folders on build path:

src/main/java

Default output folder:

build/classes

1.10 Select the option **Generate web.xml deployment descriptor** and then click on **Finish**


Practice Labs
DEFAULT
[redacted] gmailsl0032.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABJAGRIZmF1bHQ=?username=guacadmin&password=guaca...
Applications
servlets - Eclipse IDE

New Dynamic Web Project

Web Module

Configure web module settings.

Context root: eStore

Content directory: src/main/webapp

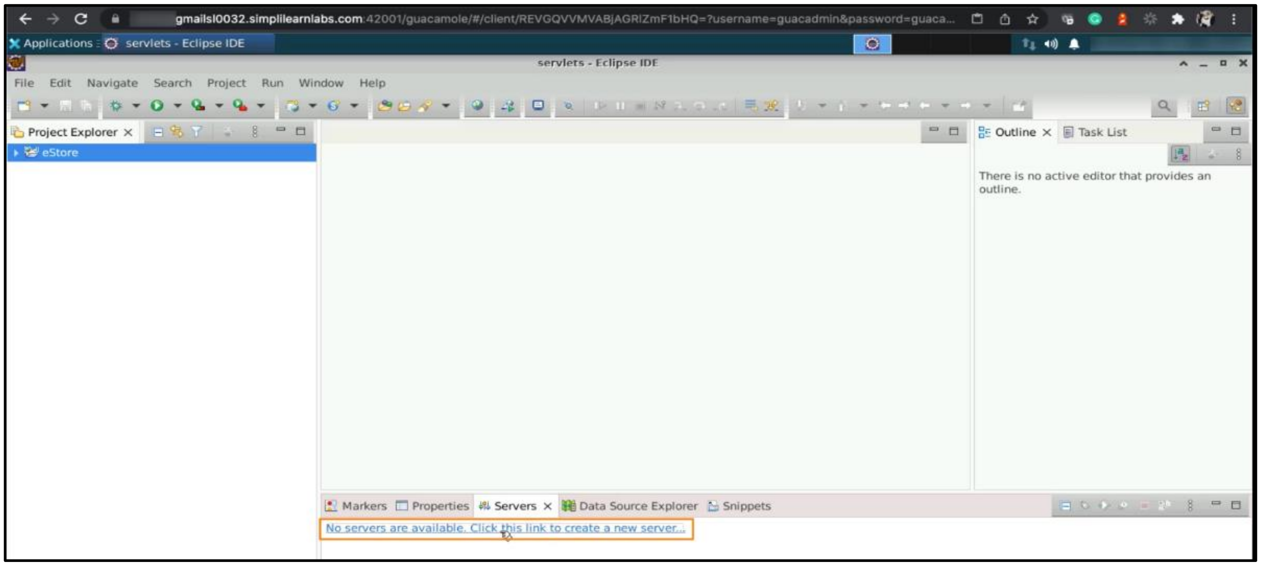
☑ Generate web.xml deployment descriptor

< Back
Next >
Cancel
Finish

The dynamic web project is now created.

## Step 2: Create a server

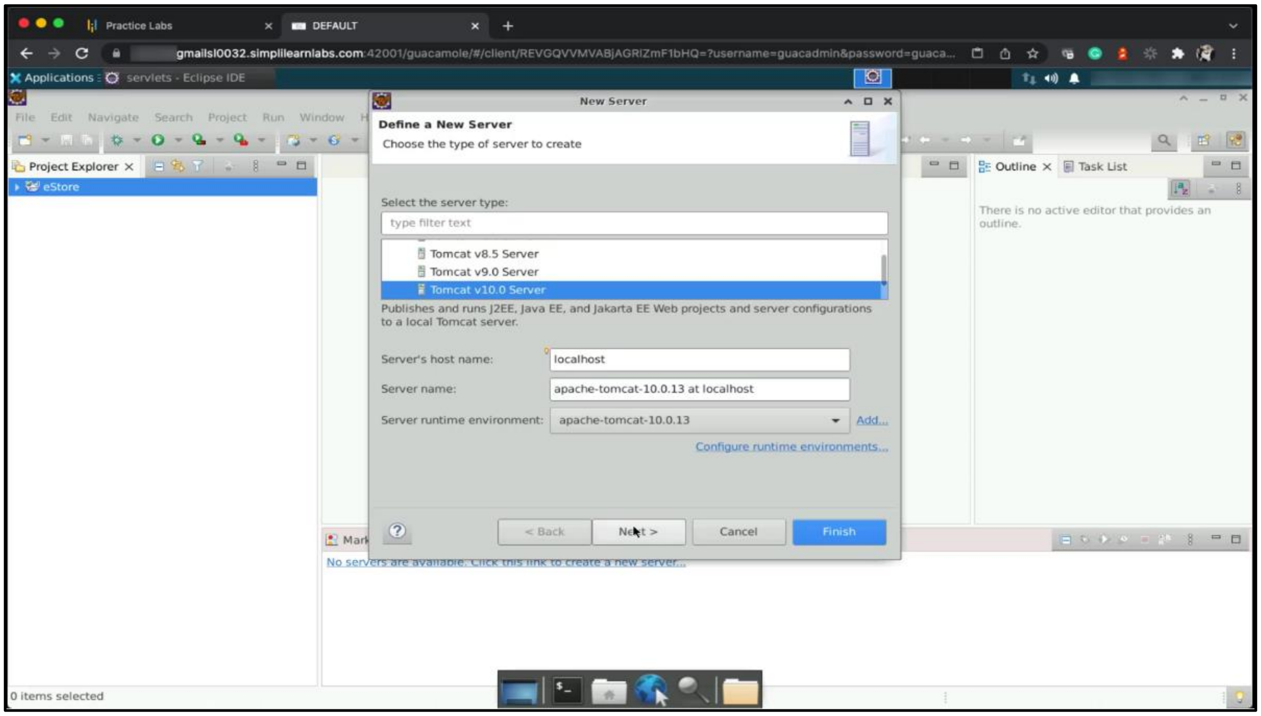
2.1 Now, go to the **Server** tab, and click on the link to **create a new server**



There is no active editor that provides an outline.

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

2.2 Select the **Apache Tomcat v.10.0 Server** and click on **Next**


Practice Labs
DEFAULT
gmailsl0032.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRlZmF1bHQ=?username=guacadmin&password=guaca...
Applications
servlets - Eclipse IDE
File Edit Navigate Search Project Run Window
Project Explorer
eStore
Outline
Task List
0 items selected
Mark
< Back
Next >
Cancel
Finish
Add...
Configure runtime environments...

New Server

**Define a New Server**

Choose the type of server to create

Select the server type:

type filter text

- Tomcat v8.5 Server
- Tomcat v9.0 Server
- Tomcat v10.0 Server

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

Server's host name: localhost

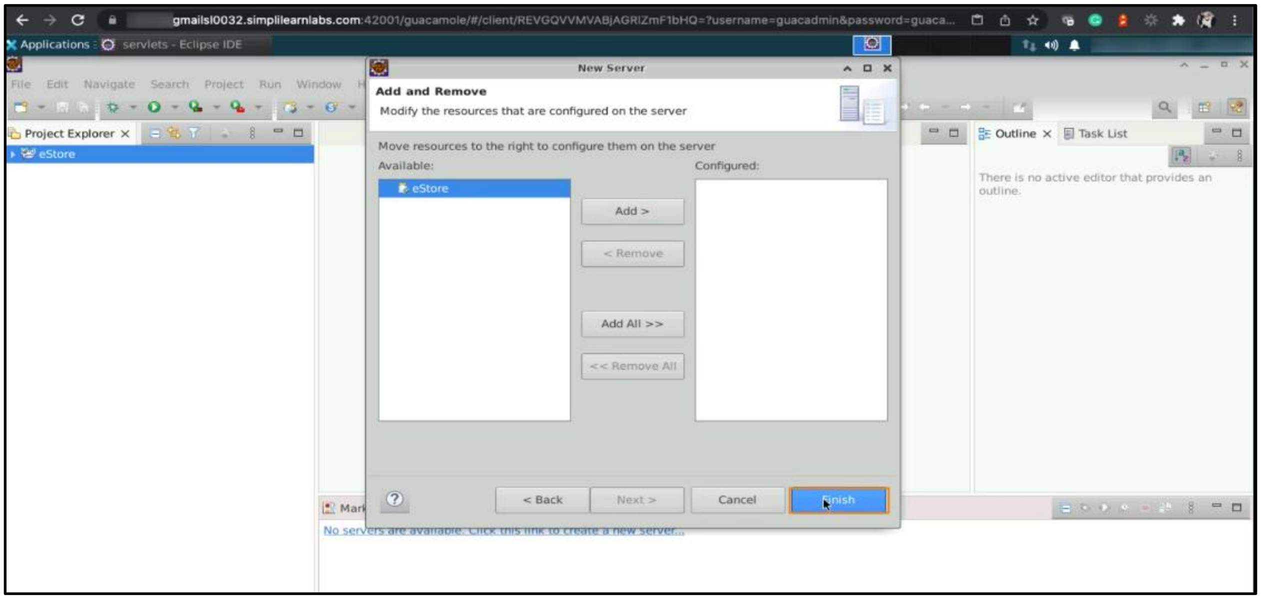
Server name: apache-tomcat-10.0.13 at localhost

Server runtime environment: apache-tomcat-10.0.13

There is no active editor that provides an outline.

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

2.3 Select the **eStore** project and click on **Finish**


gmailsl0032.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRlZmF1bHQ=?username=guacadmin&password=guaca...
Applications
servlets - Eclipse IDE
File Edit Navigate Search Project Run Window
Project Explorer
eStore
Outline
Task List
Mark
Add >
< Remove
Add All >>
<< Remove All
< Back
Next >
Cancel
Finish

New Server

**Add and Remove**

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

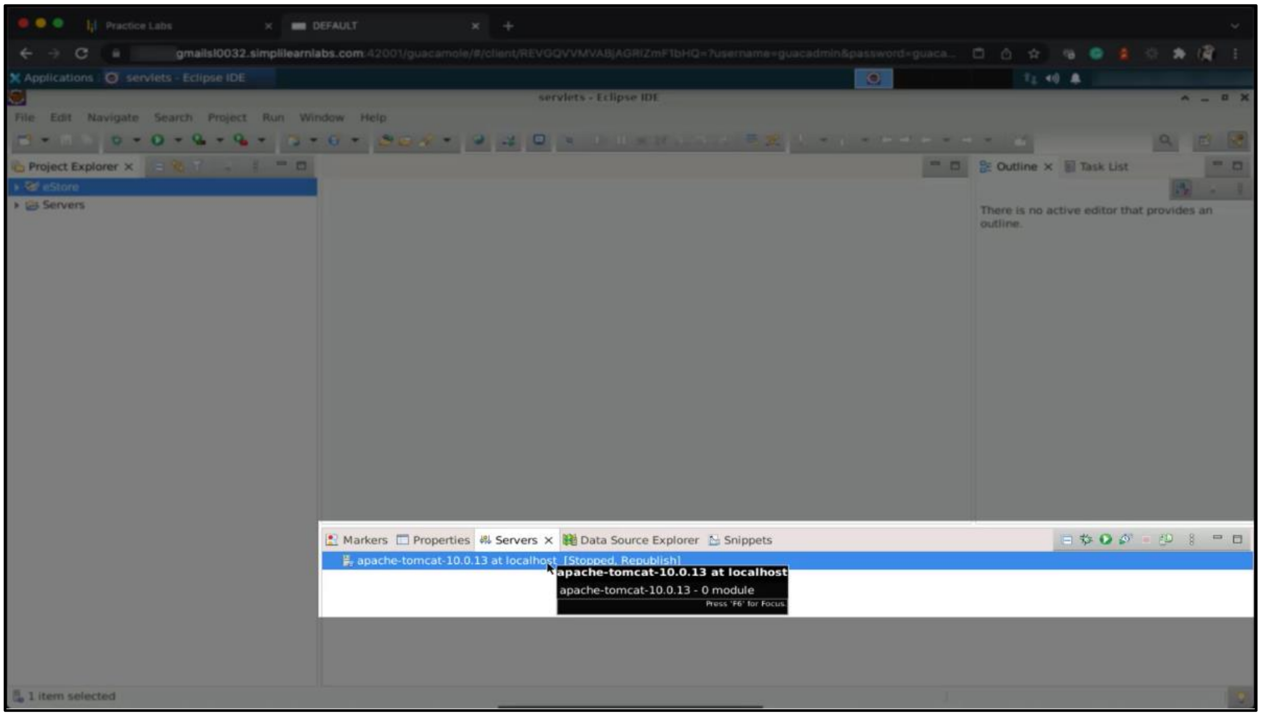
Available: eStore

Configured:

There is no active editor that provides an outline.

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

The server **apache-tomcat-10.0.13 at localhost** has been linked.

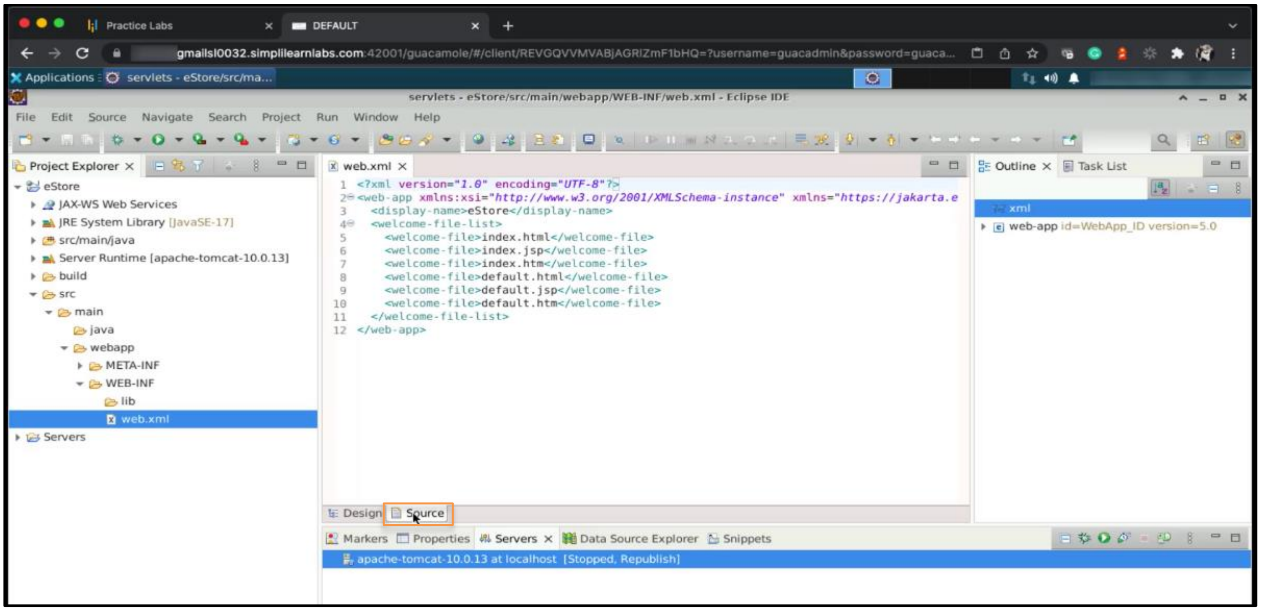


```
apache-tomcat-10.0.13 at localhost  [Stopped, Republish]
apache-tomcat-10.0.13 at localhost
apache-tomcat-10.0.13 - 0 module
Press 'F6' for Focus.
```

This is how you can configure the server for your project.

## Step 3: Create a new HTML file

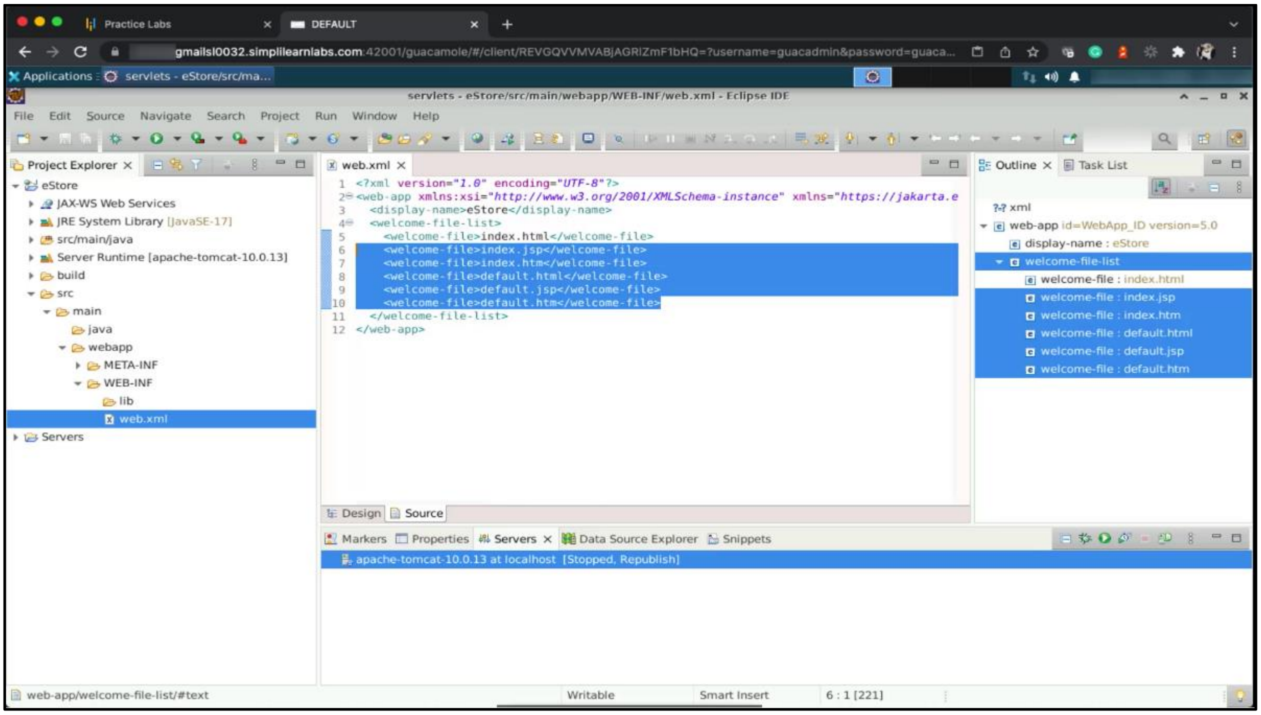
3.1 Open the **web.xml** file from **eStore -> src -> webapp -> WEB-INF -> web.xml -> Source**


servlets - eStore/src/main/webapp/WEB-INF/web.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Project Explorer
eStore
JAX-WS Web Services
JRE System Library [JavaSE-17]
src/main/java
Server Runtime [apache-tomcat-10.0.13]
build
src
main
java
webapp
META-INF
WEB-INF
lib
web.xml
Servers
web.xml
Outline
Task List
xml
web-app id=WebApp_ID version=5.0
Design
Source
Markers
Properties
Servers
Data Source Explorer
Snippets

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://jakarta.e
  <display-name>eStore</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.jsp</welcome-file>
    <welcome-file>default.htm</welcome-file>
  </welcome-file-list>
</web-app>
```

```
apache-tomcat-10.0.13 at localhost  [Stopped, Republish]
```

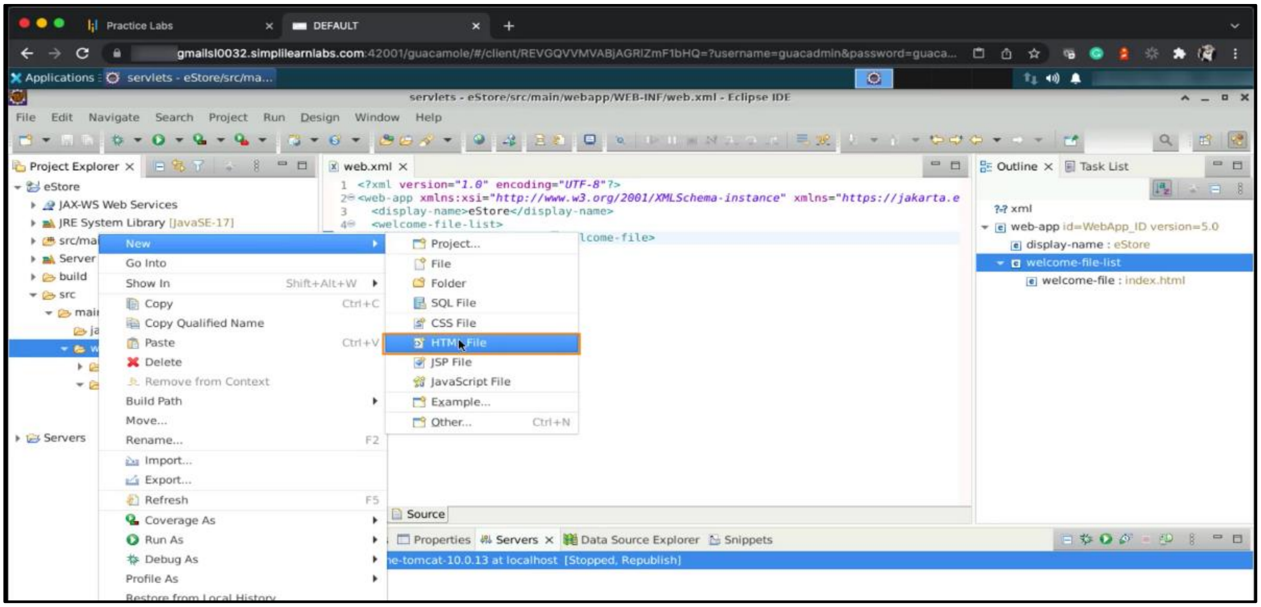
3.2 Delete all the welcome files (from lines 6 to 10)



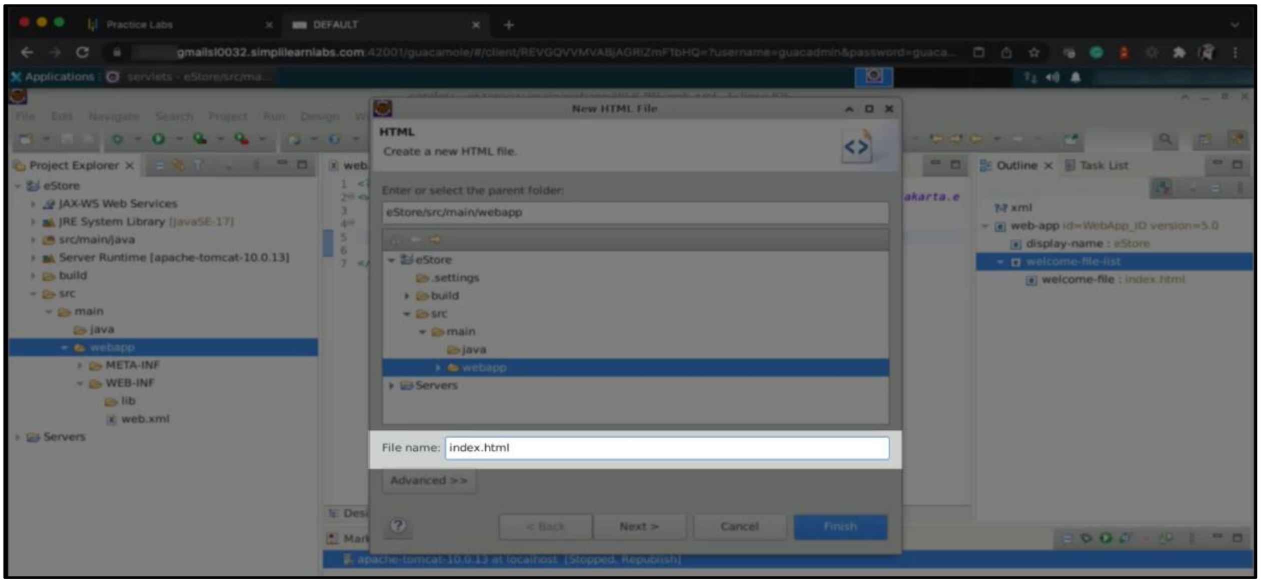
```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://jakarta.e
3    <display-name>eStore</display-name>
4    <welcome-file-list>
5      <welcome-file>index.html</welcome-file>
6      <welcome-file>index.jsp</welcome-file>
7      <welcome-file>index.htm</welcome-file>
8      <welcome-file>default.html</welcome-file>
9      <welcome-file>default.jsp</welcome-file>
10     <welcome-file>default.htm</welcome-file>
11   </welcome-file-list>
12 </web-app>
```

servlets - eStore/src/main/webapp/WEB-INF/web.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Project Explorer
eStore
JAX-WS Web Services
JRE System Library [JavaSE-17]
src/main/java
Server Runtime [apache-tomcat-10.0.13]
build
src
main
java
webapp
META-INF
WEB-INF
lib
web.xml
Servers
Outline
Task List
xml
web-app id=WebApp_ID version=5.0
display-name : eStore
welcome-file-list
welcome-file : index.html
welcome-file : index.jsp
welcome-file : index.htm
welcome-file : default.html
welcome-file : default.jsp
welcome-file : default.htm
Design
Source
Markers
Properties
Servers
Data Source Explorer
Snippets
apache-tomcat-10.0.13 at localhost [Stopped, Republish]
web-app/welcome-file-list/#text
Writable
Smart Insert
6 : 1 [221]

3.3 In the **webapp** folder, create a new HTML file. Right-click on **webapp**, select **New,** and click on **HTML File**.



3.4 Name the file **index.html**, and click on **Finish**



HTML

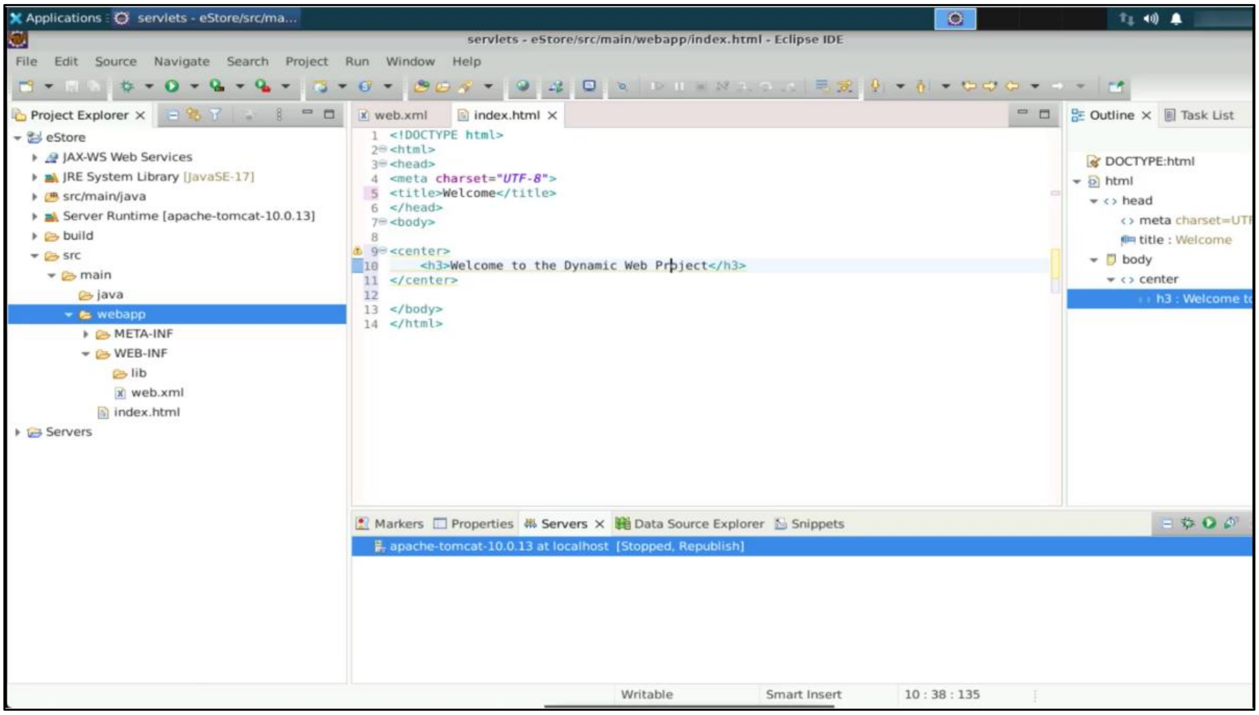
Create a new HTML file.

Enter or select the parent folder:

eStore/src/main/webapp

File name: index.html

3.5 Change the title tag message **Welcome** and create a **center** tag in the body to print **Welcome to the Dynamic Web Project.** This message will automatically come up whenever this project gets executed (from lines 5 to 10)

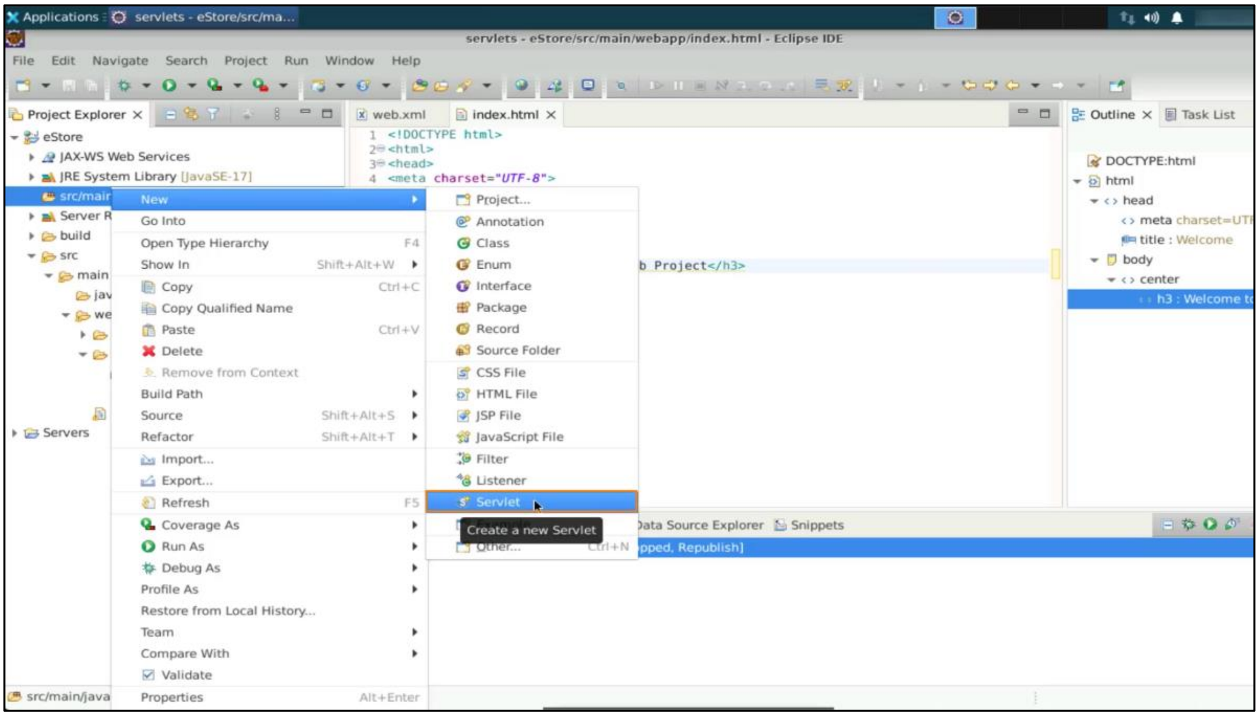

servlets - eStore/src/main/webapp/index.html - Eclipse IDE

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="UTF-8">
5  <title>Welcome</title>
6  </head>
7  <body>
8
9  <center>
10     <h3>Welcome to the Dynamic Web Project</h3>
11 </center>
12
13 </body>
14 </html>
```

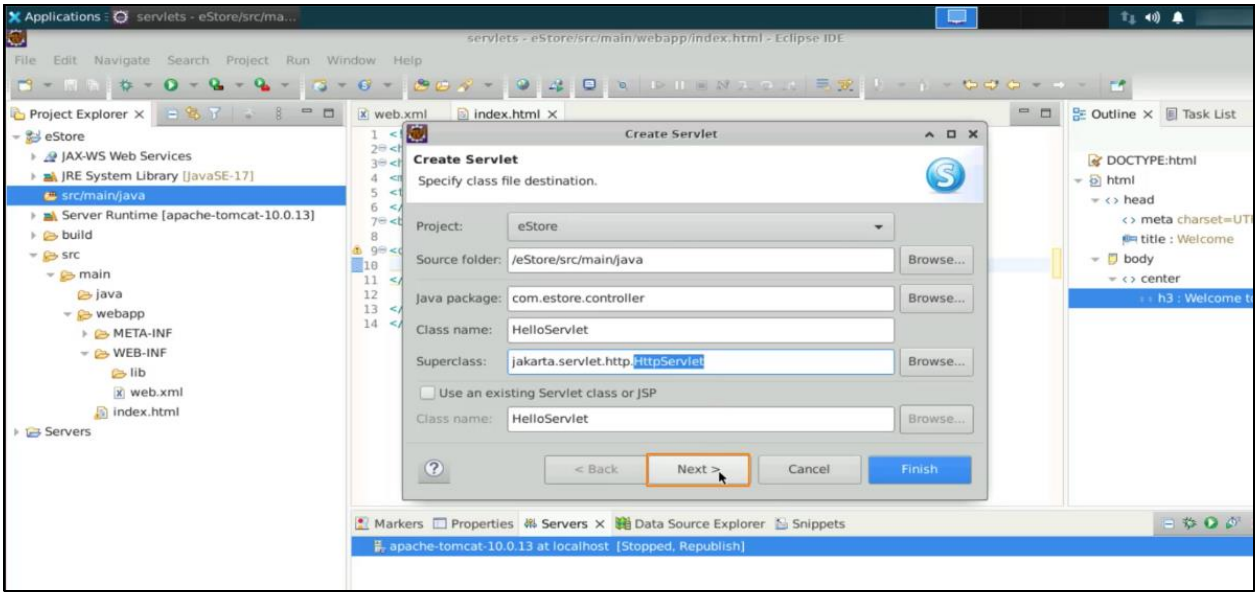
apache-tomcat-10.0.13 at localhost [Stopped, Republish]

## Step 4: Create a Servlet

4.1 Now, create a Servlet. Go to **src/main/java**, right-click on it, and select **New** > **Servlet**



4.2 Enter the package name as **com.estore.controller**, class name as **HelloServlet,** and click on **Next**


Create Servlet

Specify class file destination.

Project: eStore

Source folder: /eStore/src/main/java

Java package: com.estore.controller

Class name: HelloServlet

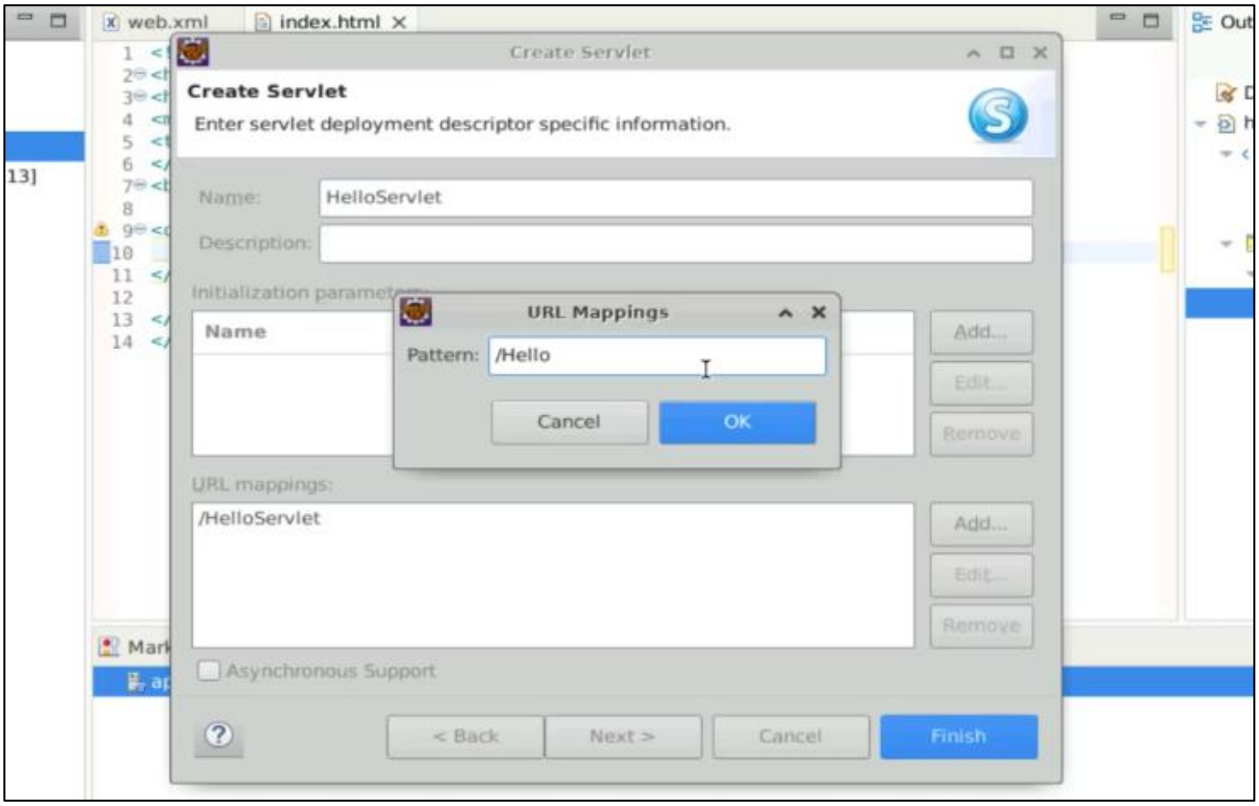
Superclass: jakarta.servlet.http.HttpServlet

□ Use an existing Servlet class or JSP

Class name: HelloServlet

Browse...
< Back
Next >
Cancel
Finish
Project Explorer
eStore
JAX-WS Web Services
JRE System Library [JavaSE-17]
src/main/java
Server Runtime [apache-tomcat-10.0.13]
build
src
main
java
webapp
META-INF
WEB-INF
lib
web.xml
index.html
Servers
Markers Properties Servers Data Source Explorer Snippets
apache-tomcat-10.0.13 at localhost [Stopped, Republish]

4.3 Click on **Add,** and add other URL mappings, like **/Hello** and **/Welcome**. Click on **Next**



Create Servlet

Enter servlet deployment descriptor specific information.

Name: HelloServlet

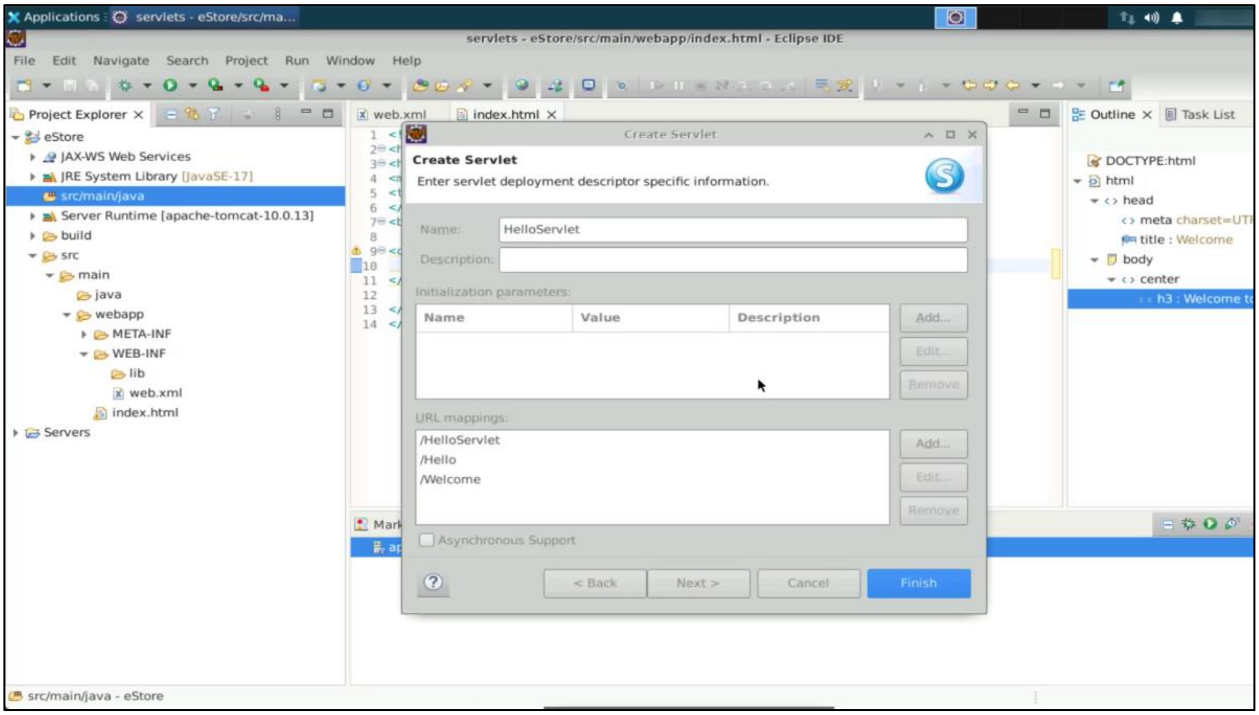
Description:

URL Mappings

Pattern: /Hello

URL mappings:

/HelloServlet



Create Servlet

Enter servlet deployment descriptor specific information.

Name: HelloServlet

Description:

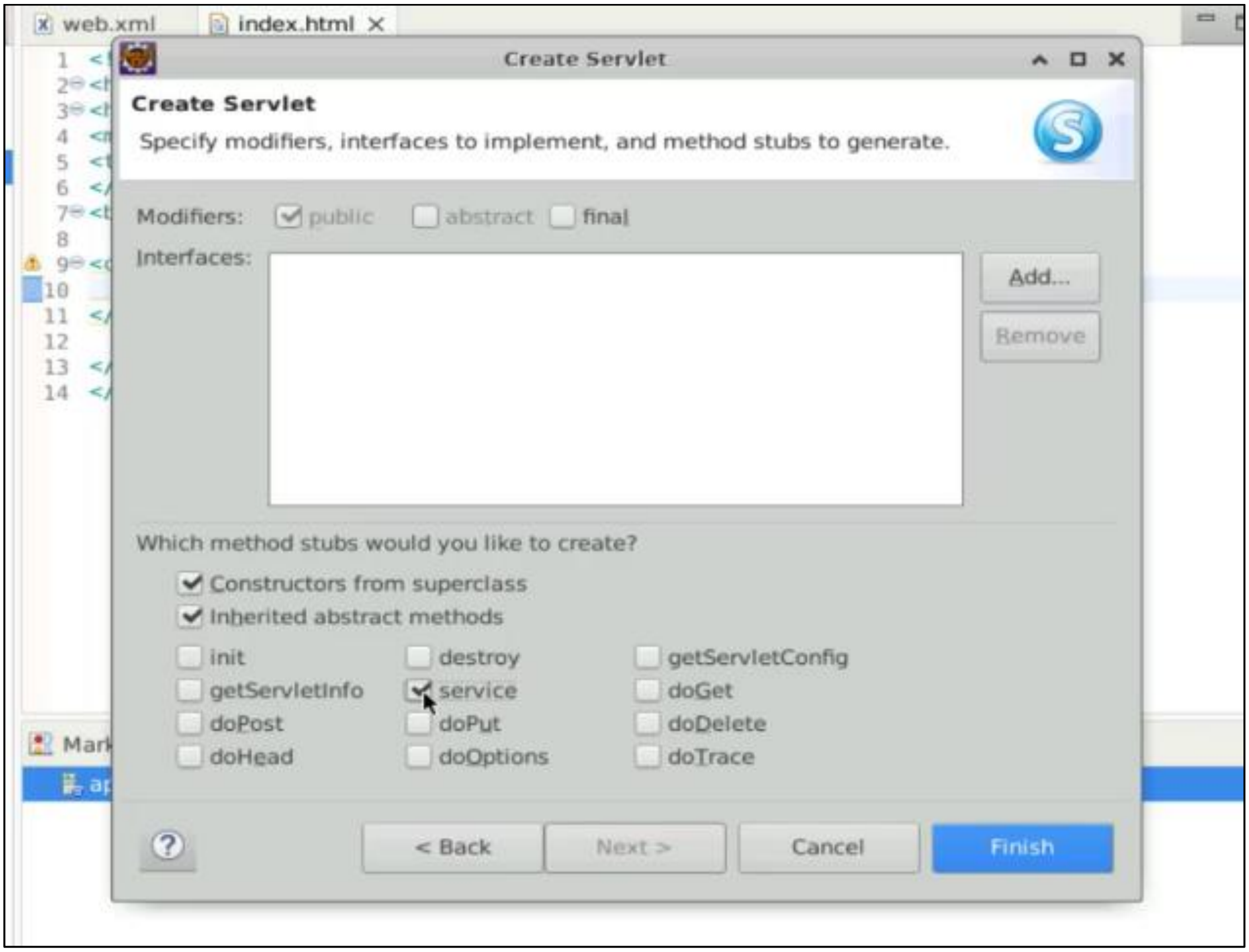
Initialization parameters:

| Name | Value | Description |
|---|---|---|

URL mappings:

/HelloServlet
/Hello
/Welcome

4.4 Select **service**, as service is the method that gets created whenever you send a request, and then click on **Finish**



Create Servlet

Specify modifiers, interfaces to implement, and method stubs to generate.

Modifiers: ☑ public □ abstract □ final

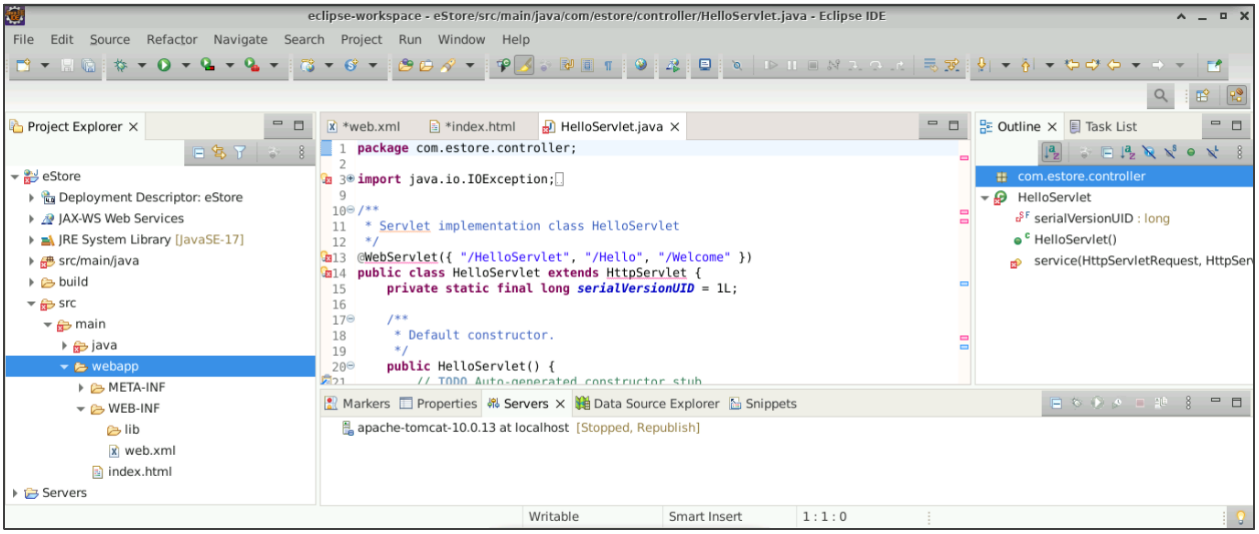
Interfaces:

Which method stubs would you like to create?

☑ Constructors from superclass
☑ Inherited abstract methods

| | | |
|---|---|---|
| □ init | □ destroy | □ getServletConfig |
| □ getServletInfo | ☑ service | □ doGet |
| □ doPost | □ doPut | □ doDelete |
| □ doHead | □ doOptions | □ doTrace |

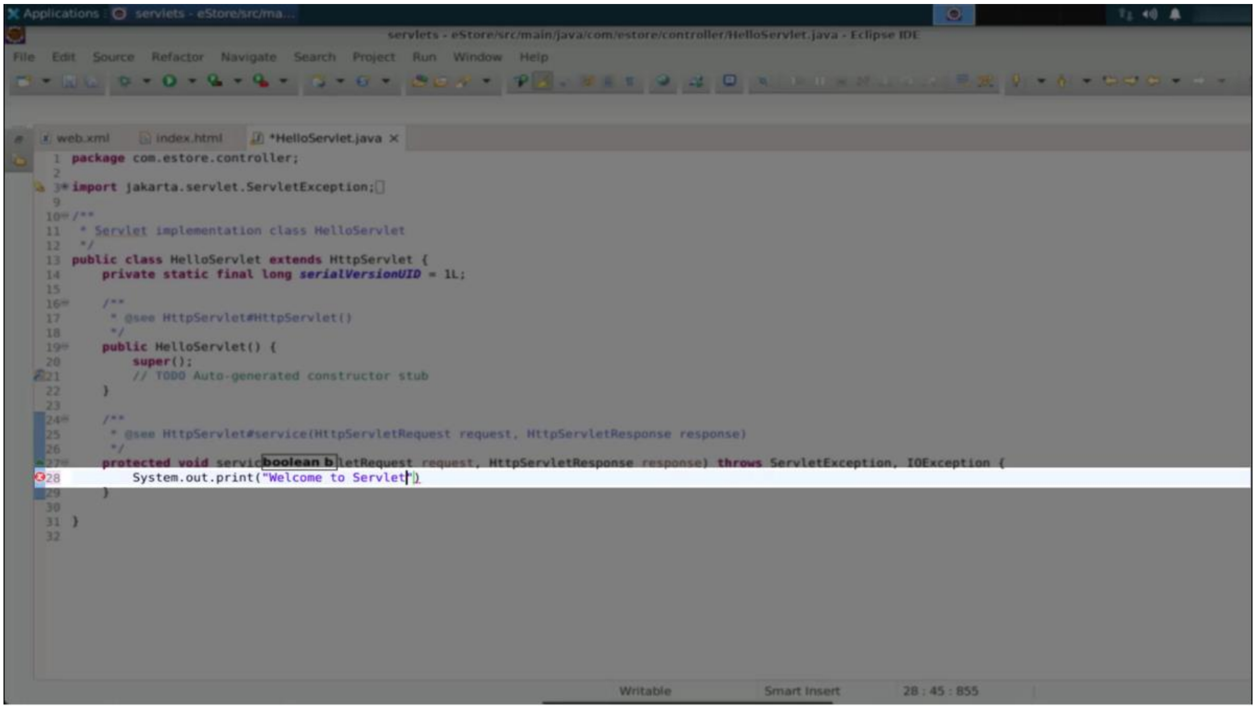
In this case, the Servlet created is a child of the HTTPServlet. A Servlet contains a request and a response.



```java
package com.estore.controller;

import java.io.IOException;

/**
 * Servlet implementation class HelloServlet
 */
@WebServlet({ "/HelloServlet", "/Hello", "/Welcome" })
public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */
    public HelloServlet() {
        // TODO Auto-generated constructor stub
```

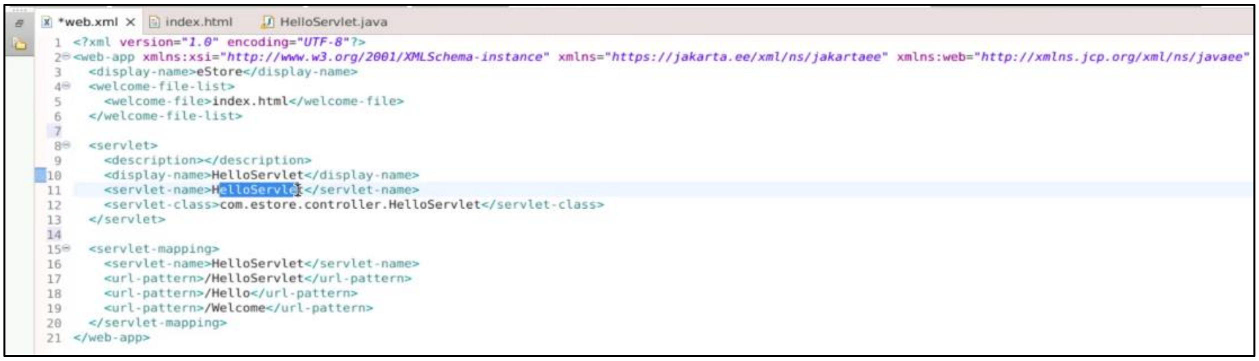
apache-tomcat-10.0.13 at localhost [Stopped, Republish]

4.5 In the service method, use the **System.out.print()** statement to display the message **Welcome to Servlet** (line 28)



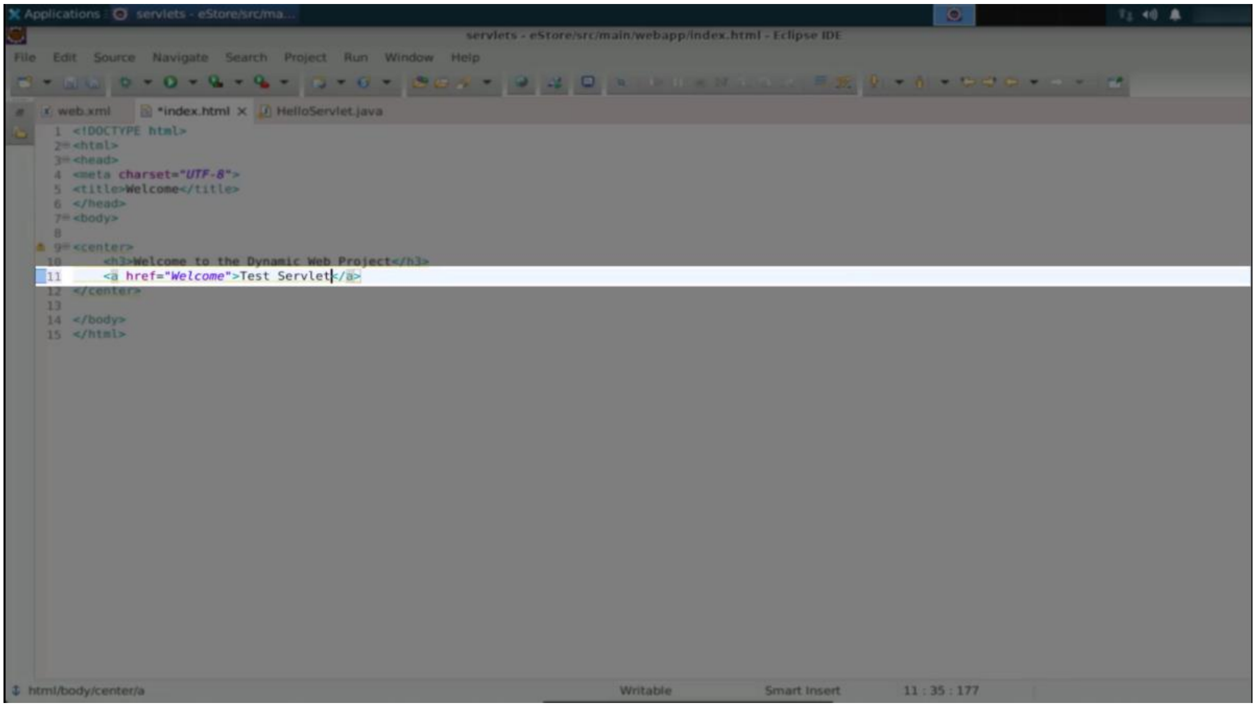
```java
1  package com.estore.controller;
2
3  import jakarta.servlet.ServletException;
9
10 /**
11  * Servlet implementation class HelloServlet
12  */
13 public class HelloServlet extends HttpServlet {
14     private static final long serialVersionUID = 1L;
15
16     /**
17      * @see HttpServlet#HttpServlet()
18      */
19     public HelloServlet() {
20         super();
21         // TODO Auto-generated constructor stub
22     }
23
24     /**
25      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
26      */
27     protected void servicboolean bletRequest request, HttpServletResponse response) throws ServletException, IOException {
28         System.out.print("Welcome to Servlet")
29     }
30
31 }
32
```

4.6 Go to the **web.xml** file, and you can see a servlet-name created as **HelloServlet**



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://jakarta.ee/xml/ns/jakartaee" xmlns:web="http://xmlns.jcp.org/xml/ns/javaee"
  <display-name>eStore</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>

  <servlet>
    <description></description>
    <display-name>HelloServlet</display-name>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>com.estore.controller.HelloServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/HelloServlet</url-pattern>
    <url-pattern>/Hello</url-pattern>
    <url-pattern>/Welcome</url-pattern>
  </servlet-mapping>
</web-app>
```

4.7 Go to the **index.html** file and create a hyperlink as **Welcome** (line 11). This is one of the URL patterns you gave while creating the Servlet.

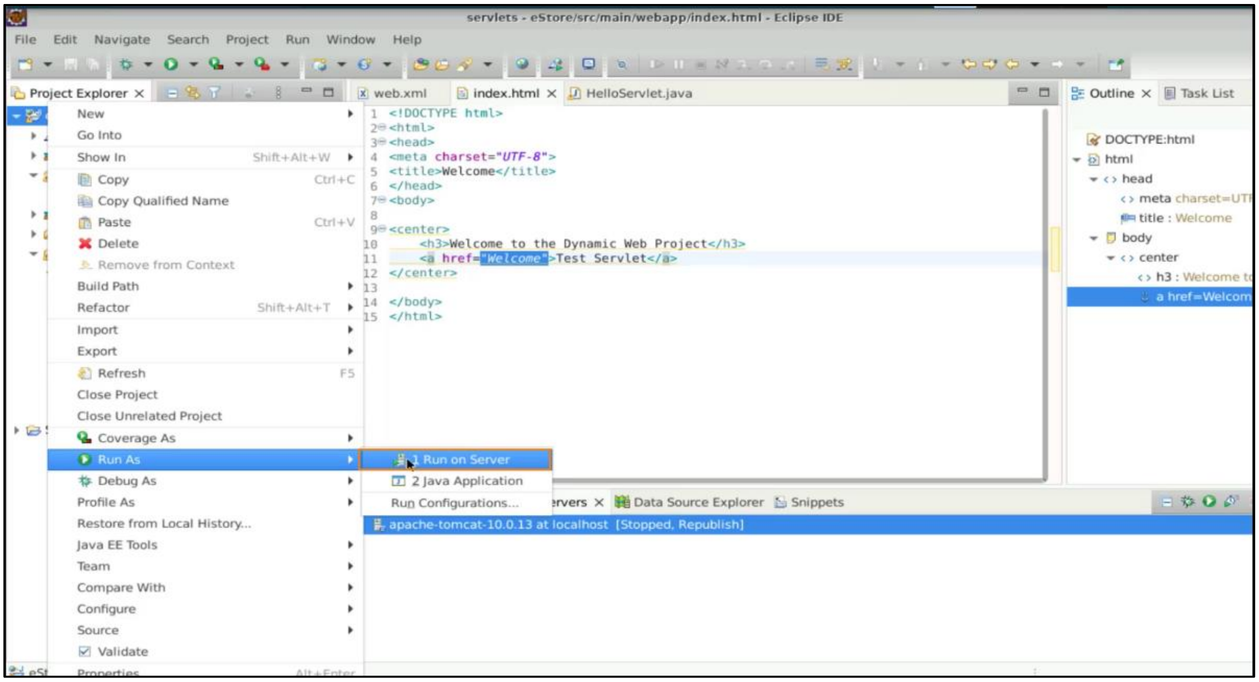


Applications
servlets - eStore/src/ma...
servlets - eStore/src/main/webapp/index.html - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
web.xml
*index.html
HelloServlet.java
html/body/center/a
Writable
Smart Insert
11 : 35 : 177

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="UTF-8">
5  <title>Welcome</title>
6  </head>
7  <body>
8
9  <center>
10     <h3>Welcome to the Dynamic Web Project</h3>
11     <a href="Welcome">Test Servlet</a>
12 </center>
13
14 </body>
15 </html>
```

## Step 5: Run the project and handle the error

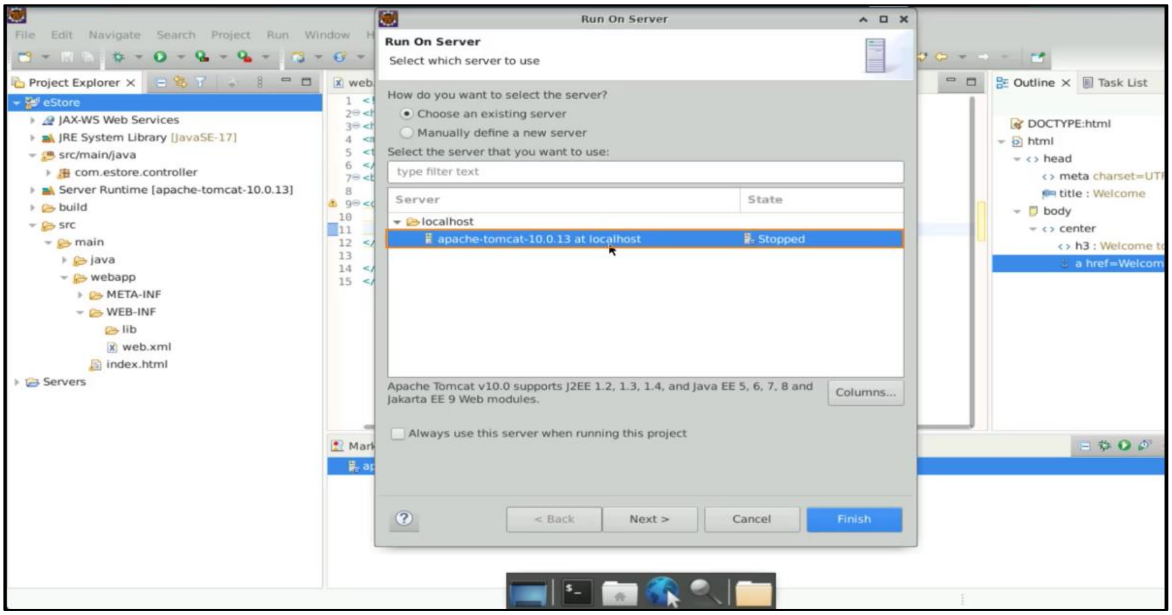
5.1 Go to the **index.html** file and run the project. Right-click on **eStore**, select **Run As,** and then click on **Run on Server**



```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Welcome</title>
</head>
<body>

<center>
    <h3>Welcome to the Dynamic Web Project</h3>
    <a href="Welcome">Test Servlet</a>
</center>

</body>
</html>
```

apache-tomcat-10.0.13 at localhost [Stopped, Republish]

5.2 Select the server **apache-tomcat-10.013 at localhost**, and then click on **Finish**



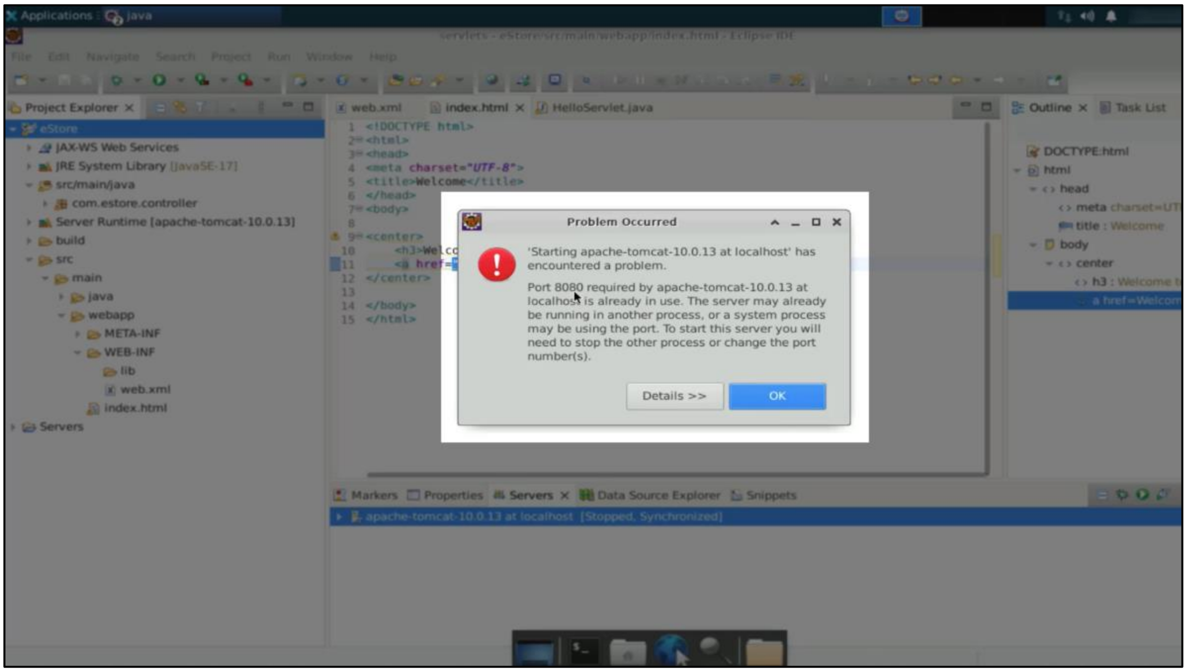
● Choose an existing server
○ Manually define a new server

| Server | State |
|---|---|
| localhost | |
| apache-tomcat-10.0.13 at localhost | Stopped |

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.

□ Always use this server when running this project

The output has the error stating that port **8080** is already in use.

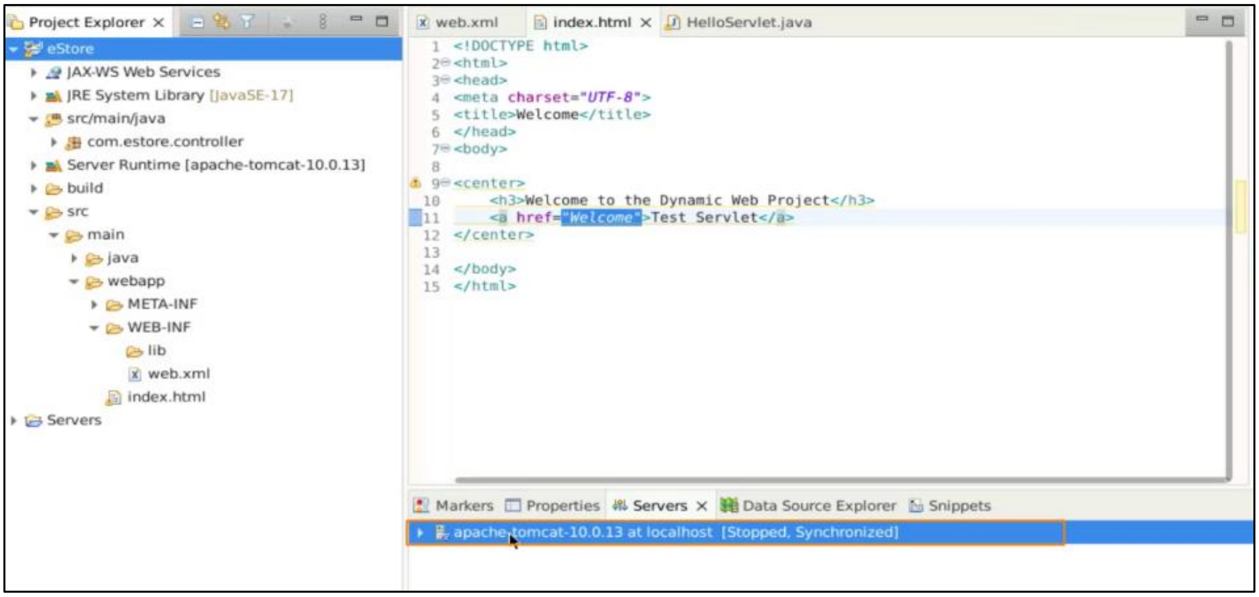


Problem Occurred

'Starting apache-tomcat-10.0.13 at localhost' has encountered a problem.

Port 8080 required by apache-tomcat-10.0.13 at localhost is already in use. The server may already be running in another process, or a system process may be using the port. To start this server you will need to stop the other process or change the port number(s).

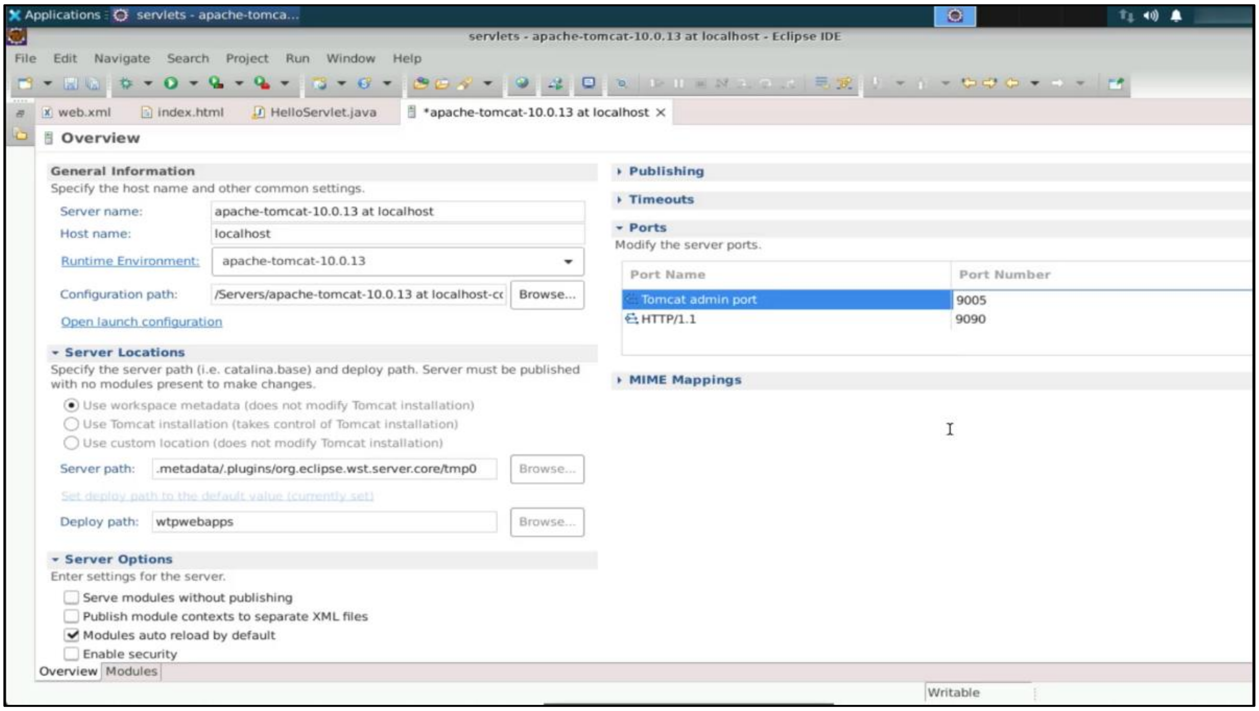
5.3 Open the tomcat file configuration file



```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Welcome</title>
</head>
<body>

<center>
    <h3>Welcome to the Dynamic Web Project</h3>
    <a href="Welcome">Test Servlet</a>
</center>

</body>
</html>
```

apache-tomcat-10.0.13 at localhost [Stopped, Synchronized]

5.4 Change the port number from **8080** to **9090** and the admin port to **9005**



Applications
servlets - apache-tomca...
servlets - apache-tomcat-10.0.13 at localhost - Eclipse IDE
File Edit Navigate Search Project Run Window Help
web.xml
index.html
HelloServlet.java
*apache-tomcat-10.0.13 at localhost
Overview
General Information
Specify the host name and other common settings.
Publishing
Timeouts
Ports
Modify the server ports.
MIME Mappings
Server Locations
Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.
Open launch configuration
Set deploy path to the default value (currently set)
Browse...
Server Options
Enter settings for the server.
Overview
Modules
Writable

Server name: apache-tomcat-10.0.13 at localhost

Host name: localhost

Runtime Environment: apache-tomcat-10.0.13

Configuration path: /Servers/apache-tomcat-10.0.13 at localhost-c

| Port Name | Port Number |
|---|---|
| Tomcat admin port | 9005 |
| HTTP/1.1 | 9090 |

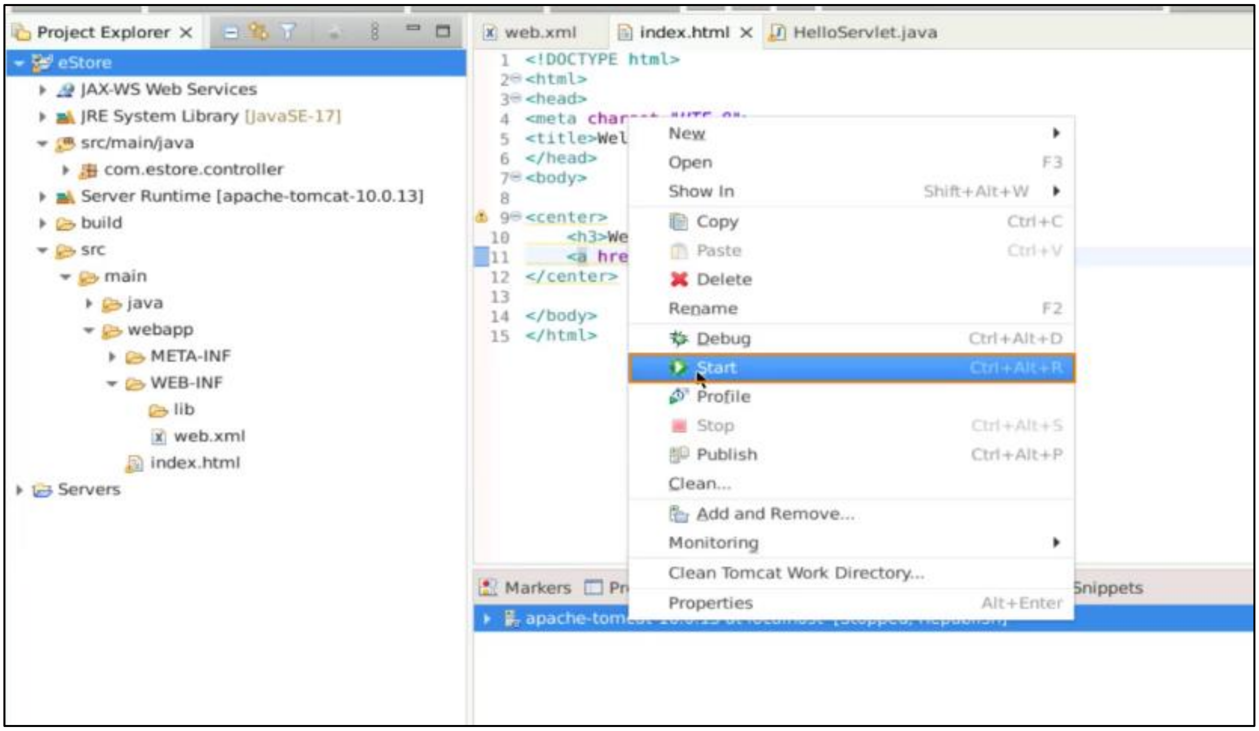
● Use workspace metadata (does not modify Tomcat installation)
○ Use Tomcat installation (takes control of Tomcat installation)
○ Use custom location (does not modify Tomcat installation)

Server path: .metadata/.plugins/org.eclipse.wst.server.core/tmp0

Deploy path: wtpwebapps

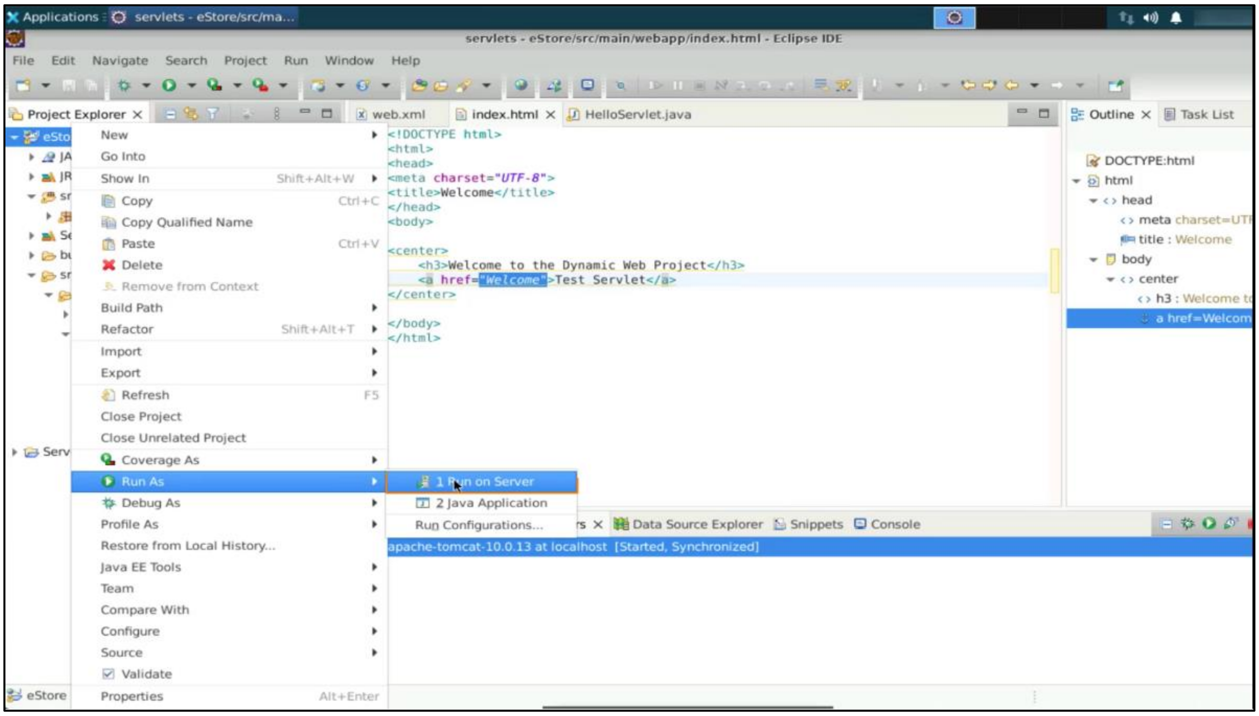
□ Serve modules without publishing
□ Publish module contexts to separate XML files
☑ Modules auto reload by default
□ Enable security

5.5 Save the file, right-click on the screen and select **Start**



```
<!DOCTYPE html>
<html>
<head>
<meta char
<title>Wel
</head>
<body>

<center>
    <h3>We
    <a hre
</center>

</body>
</html>
```

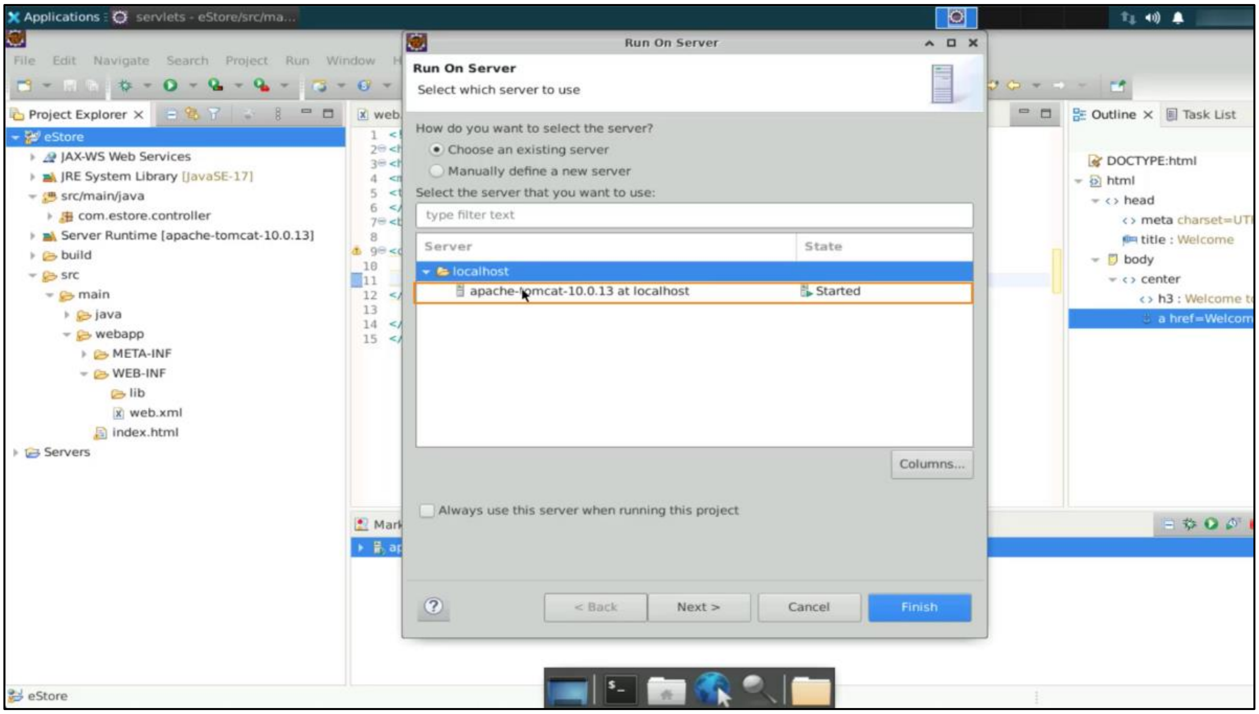
5.6 Now, repeat step 5.1



```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Welcome</title>
</head>
<body>

<center>
    <h3>Welcome to the Dynamic Web Project</h3>
    <a href="Welcome">Test Servlet</a>
</center>

</body>
</html>
```

apache-tomcat-10.0.13 at localhost [Started, Synchronized]

5.7 Repeat step 5.2



Run On Server

Select which server to use

How do you want to select the server?

● Choose an existing server

○ Manually define a new server

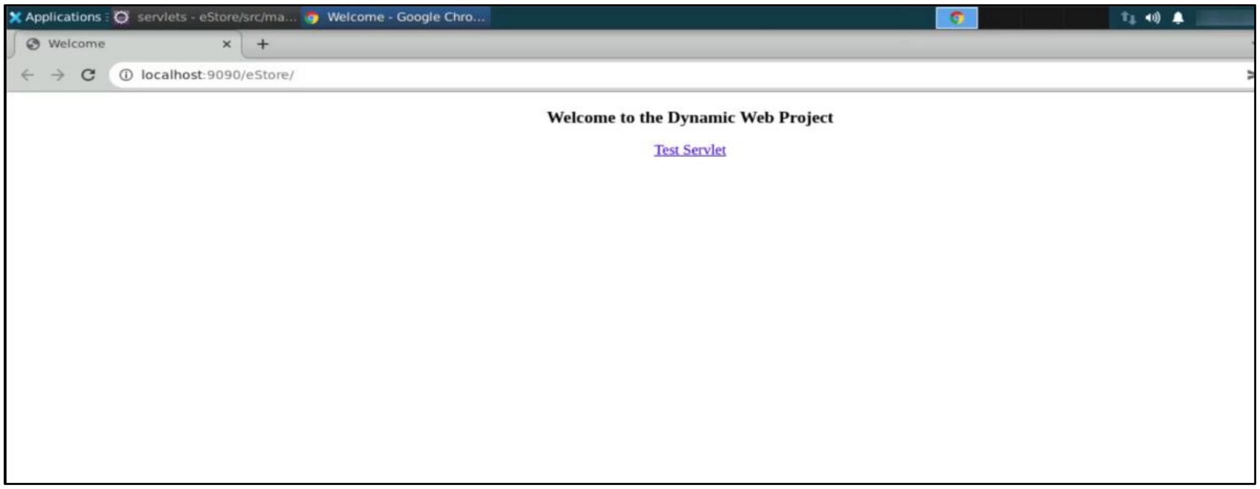
Select the server that you want to use:

type filter text

| Server | State |
|---|---|
| localhost | |
| apache-tomcat-10.0.13 at localhost | Started |

□ Always use this server when running this project

The output displayed will be, **Welcome to the Dynamic Web Project.**

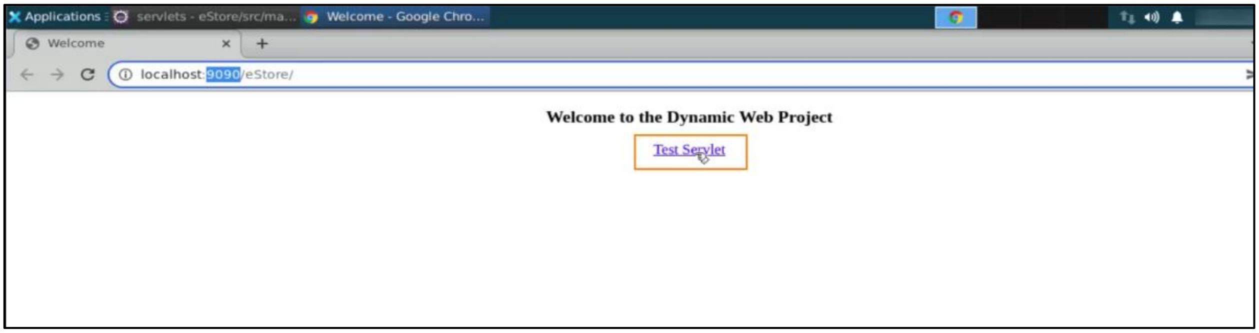


Welcome to the Dynamic Web Project

Test Servlet

You can see that the program is executed on port number **9090** since **8080** was blocked.

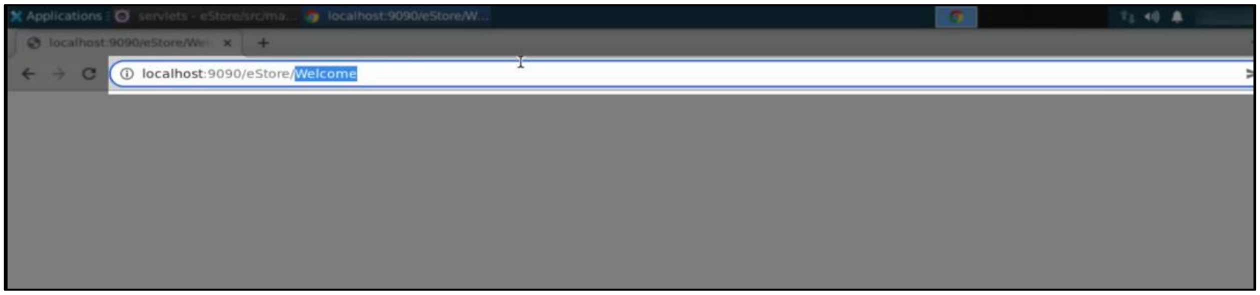
5.8 Now, click on the link **Test Servlet**



Welcome to the Dynamic Web Project

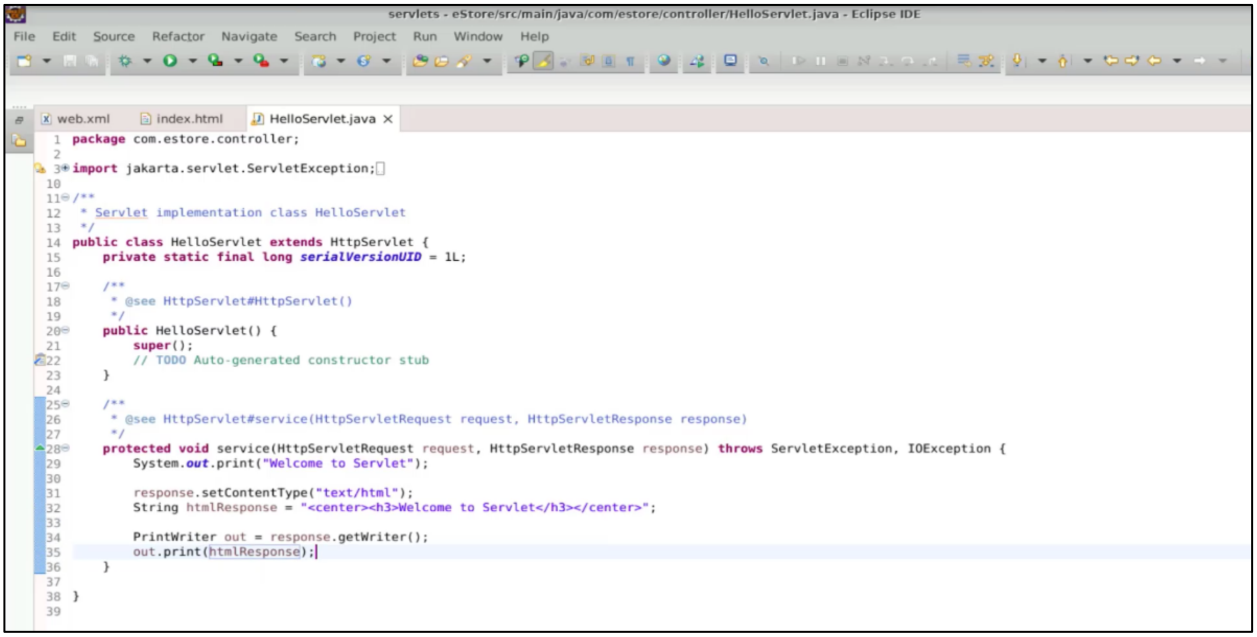
Test Servlet

You can now observe that the URL pattern is changed to **Welcome**.


Applications
servlets - eStore/src/ma...
localhost:9090/eStore/W...
localhost:9090/eStore/We
localhost:9090/eStore/Welcome

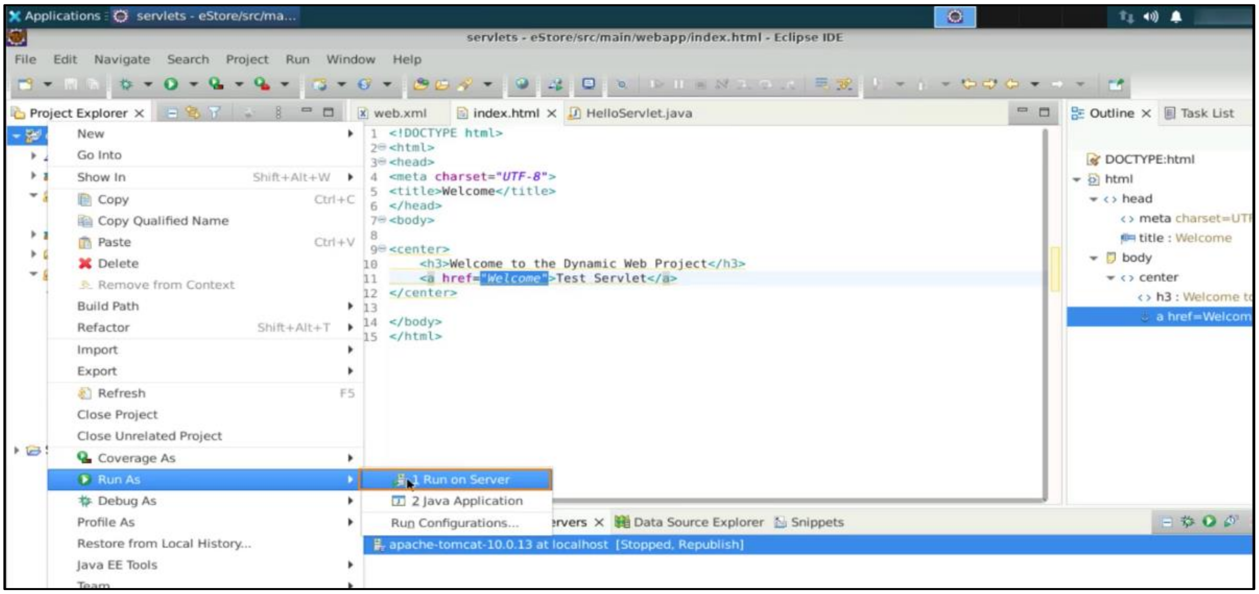
## Step 6: Make the Servlet work with a response

6.1 Go to the **Servlet.java** file and create the responses from the Servlet. Set the content type for a response using the **setContentType()** method and add the type as **text** or **HTML**. Add a **PrintWriter** and a reference variable for the response (lines 31 to 35)


servlets - eStore/src/main/java/com/estore/controller/HelloServlet.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
web.xml
index.html
HelloServlet.java

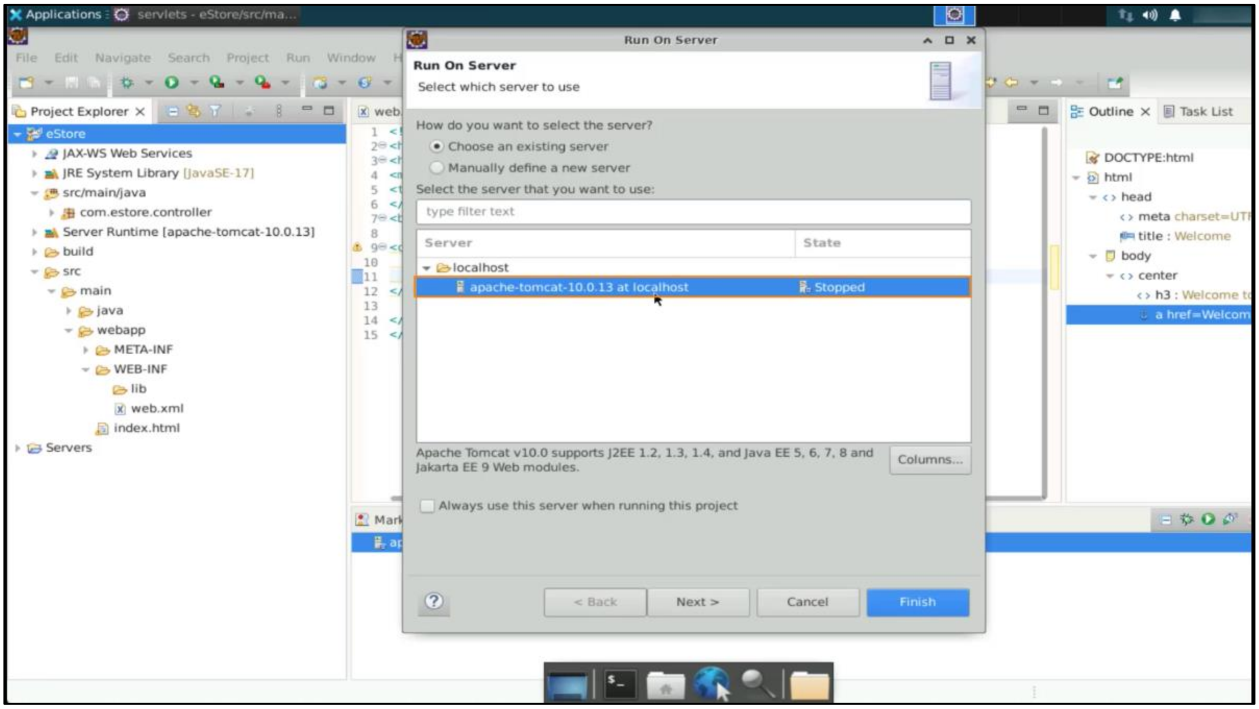
```
1  package com.estore.controller;
2
3⊕ import jakarta.servlet.ServletException;
10
11⊖ /**
12  * Servlet implementation class HelloServlet
13  */
14 public class HelloServlet extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17⊖    /**
18      * @see HttpServlet#HttpServlet()
19      */
20⊖    public HelloServlet() {
21         super();
22         // TODO Auto-generated constructor stub
23     }
24
25⊖    /**
26      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
27      */
28⊖    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29         System.out.print("Welcome to Servlet");
30
31         response.setContentType("text/html");
32         String htmlResponse = "<center><h3>Welcome to Servlet</h3></center>";
33
34         PrintWriter out = response.getWriter();
35         out.print(htmlResponse);
36     }
37
38 }
39
```

6.2 Repeat steps 5.1 and 5.2



```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Welcome</title>
</head>
<body>

<center>
    <h3>Welcome to the Dynamic Web Project</h3>
    <a href="Welcome">Test Servlet</a>
</center>

</body>
</html>
```

apache-tomcat-10.0.13 at localhost [Stopped, Republish]



Run On Server

Select which server to use

How do you want to select the server?

● Choose an existing server

○ Manually define a new server

Select the server that you want to use:

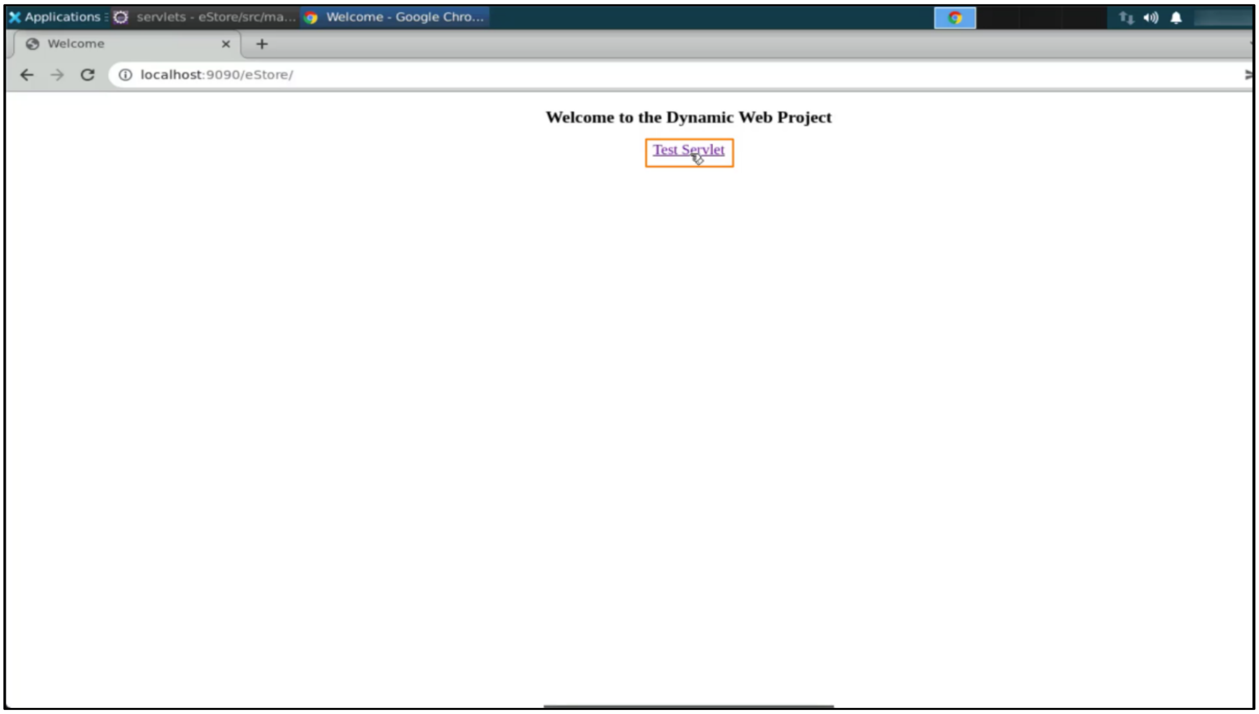
type filter text

| Server | State |
|---|---|
| localhost | |
| apache-tomcat-10.0.13 at localhost | Stopped |

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.

□ Always use this server when running this project

6.3 Go to the **browser**, open **eStore localhost**, and then click on the link **Test Servlet**

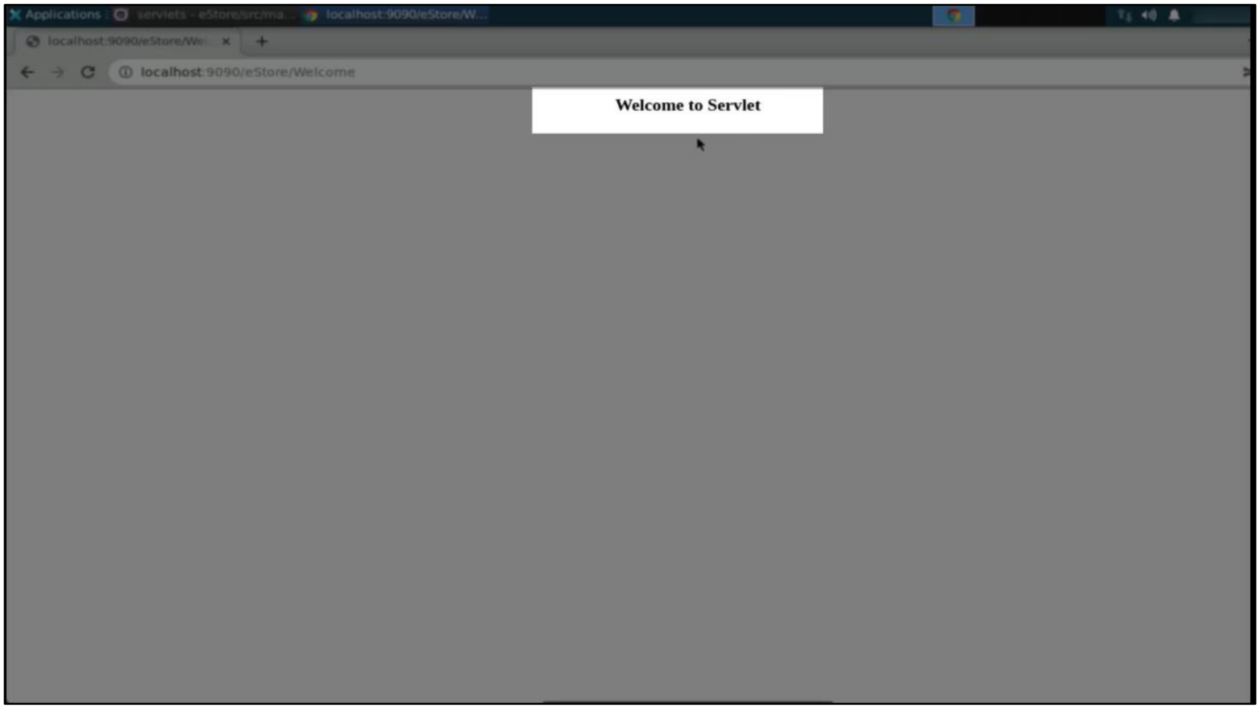


localhost:9090/eStore/

**Welcome to the Dynamic Web Project**

Test Servlet

The output can be seen as a response as **Welcome to Servlet.**


Applications
servlets - eStore/src/ma...
localhost:9090/eStore/W...
localhost:9090/eStore/We
localhost:9090/eStore/Welcome

**Welcome to Servlet**

Following these steps, you create a dynamic web project using Eclipse IDE and make the Servlet work with responses.