

## Lesson 01 Demo 04

### Life Cycle of Servlet

**Objective:** To understand the life cycle of a Servlet and how the `init()`, `service()`, and `destroy()` methods work

**Tools Required:** Eclipse IDE

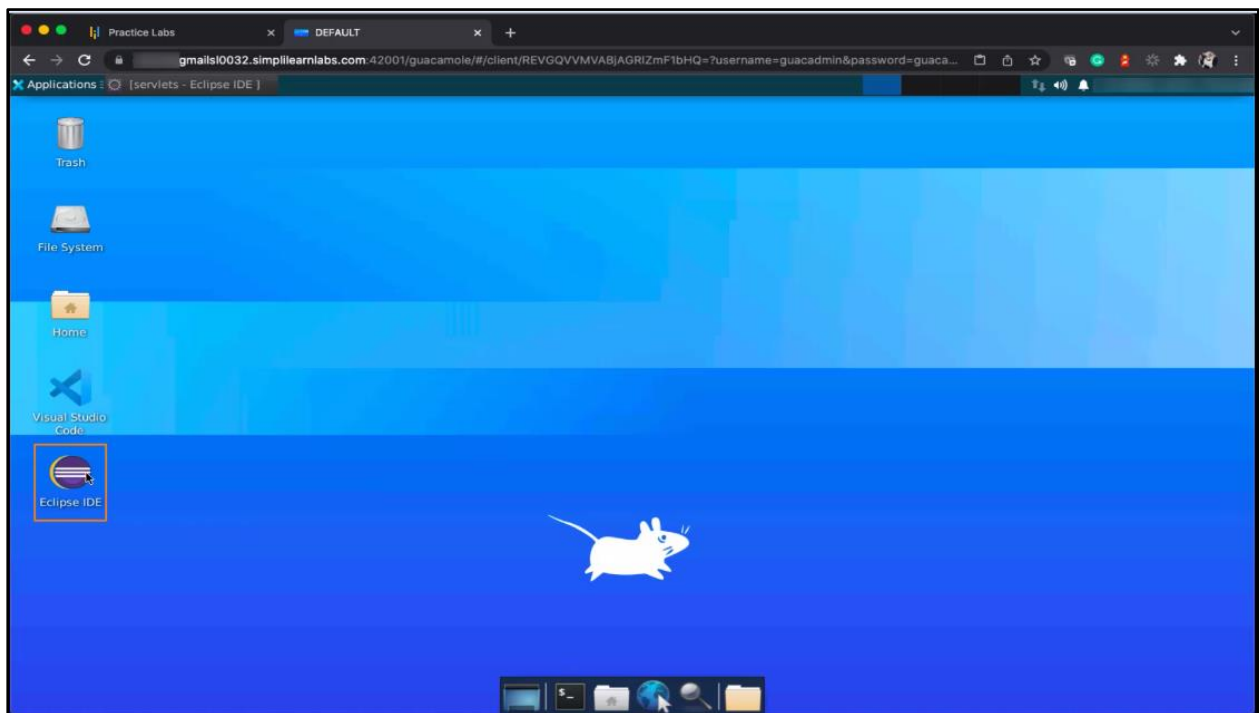
**Prerequisites:** None

#### Steps to be followed:

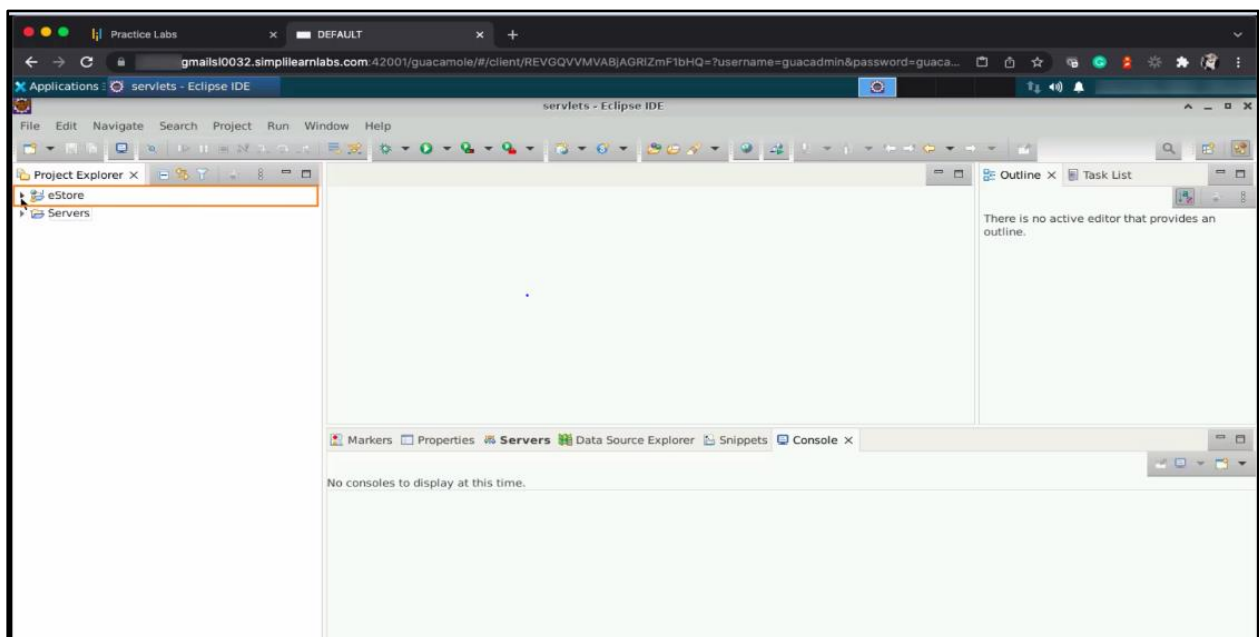
1. Create the `init` and `destroy` method
2. Make changes in internal service
3. Create a hyperlink for Hello Servlet
4. Run code to check how the life cycle works

#### Step 1: Create the `init` and `destroy` method

## 1.1 Open Eclipse IDE

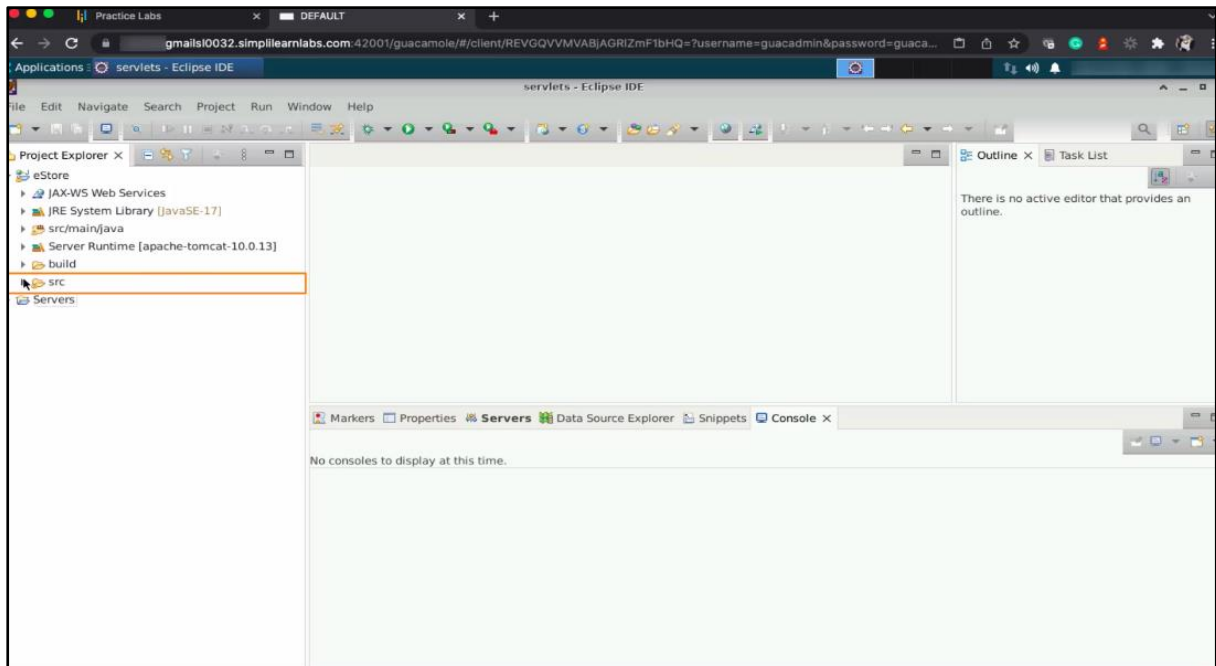


## 1.2 Open the eStore project

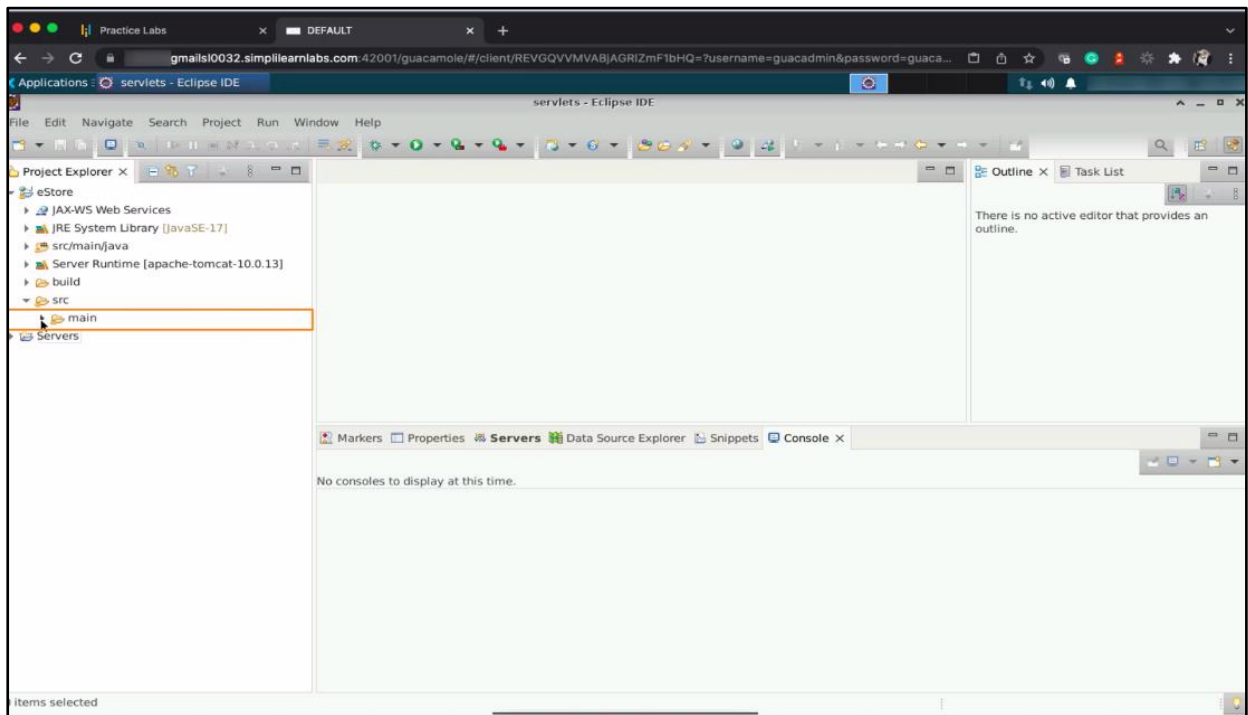


**Note:** Please refer to the previous demos on how to create the **eStore** project

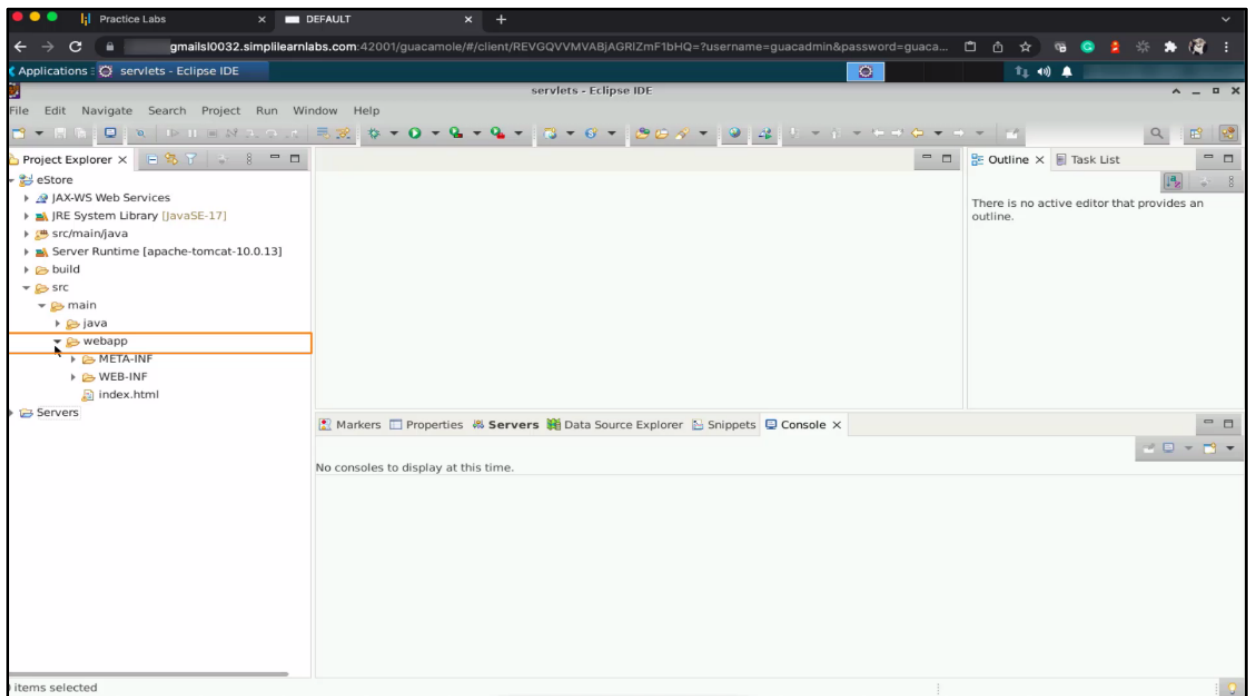
### 1.3 Go to the **src** directory



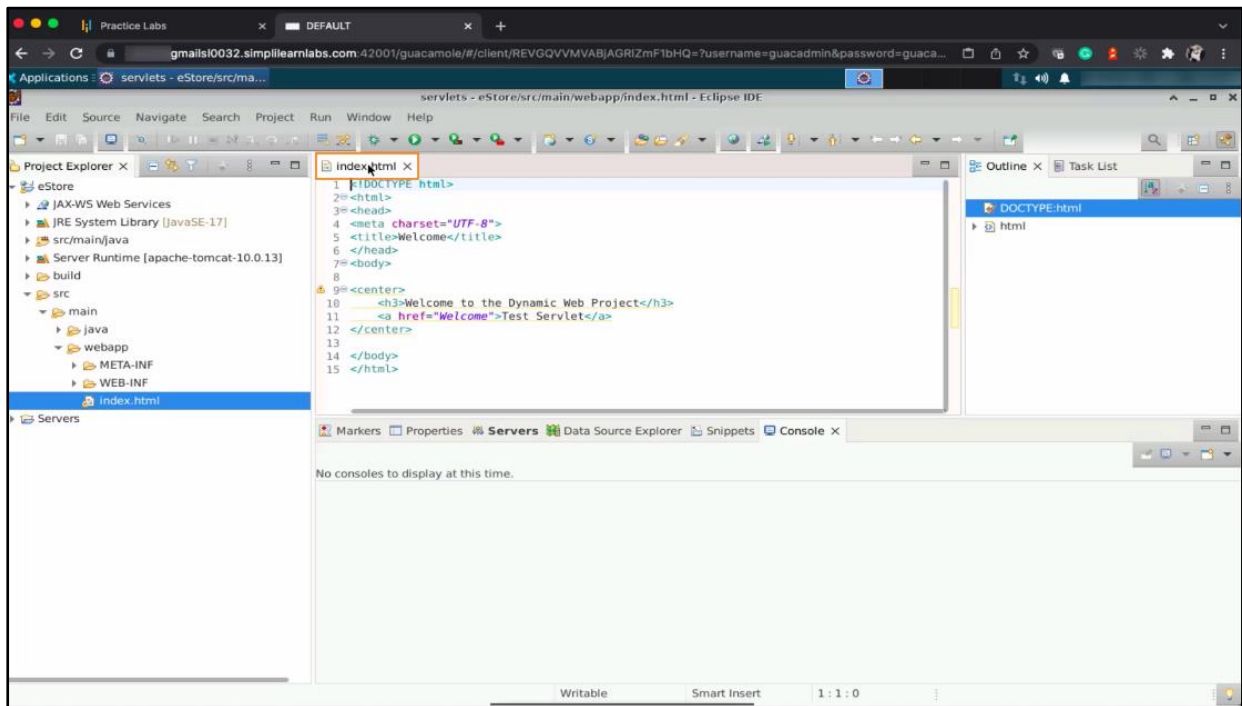
## 1.4 Select the **main** folder



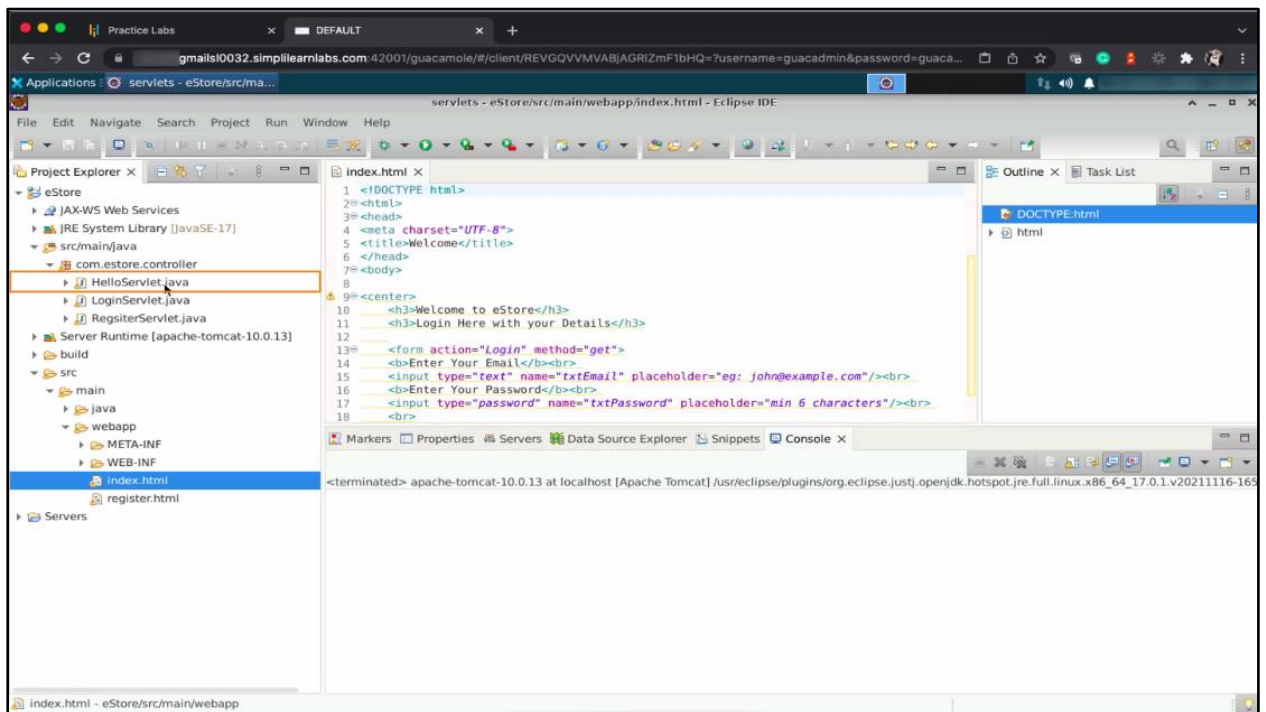
## 1.5 Open the **webapp** folder



## 1.6 Open the **index.html** file to see the HTML form that will be used later



## 1.7 Now, open the **HelloServlet.java** file to define the methods



## 1.8 Define the method `init()` and override it in the parent class `HttpServlet`

```

dex.html *HelloServlet.java X
package com.estore.controller;

import jakarta.servlet.ServletConfig;

/**
 * Servlet implementation class HelloServlet
 */
public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HelloServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

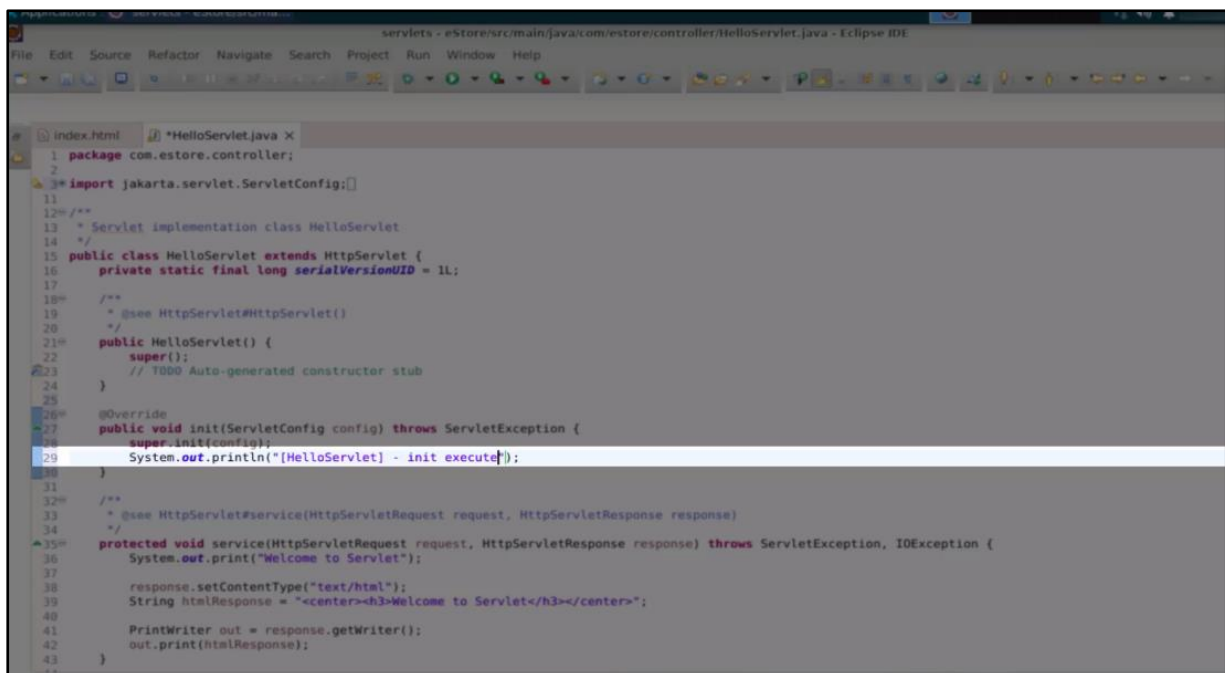
    @Override
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }

    /**
     * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
     */
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        System.out.print("Welcome to Servlet");

        response.setContentType("text/html");
        String htmlResponse = "<center><h3>Welcome to Servlet</h3></center>";
    }
}

```

## 1.9 Print the message `Hello Servlet` using the `System.out.println()` method



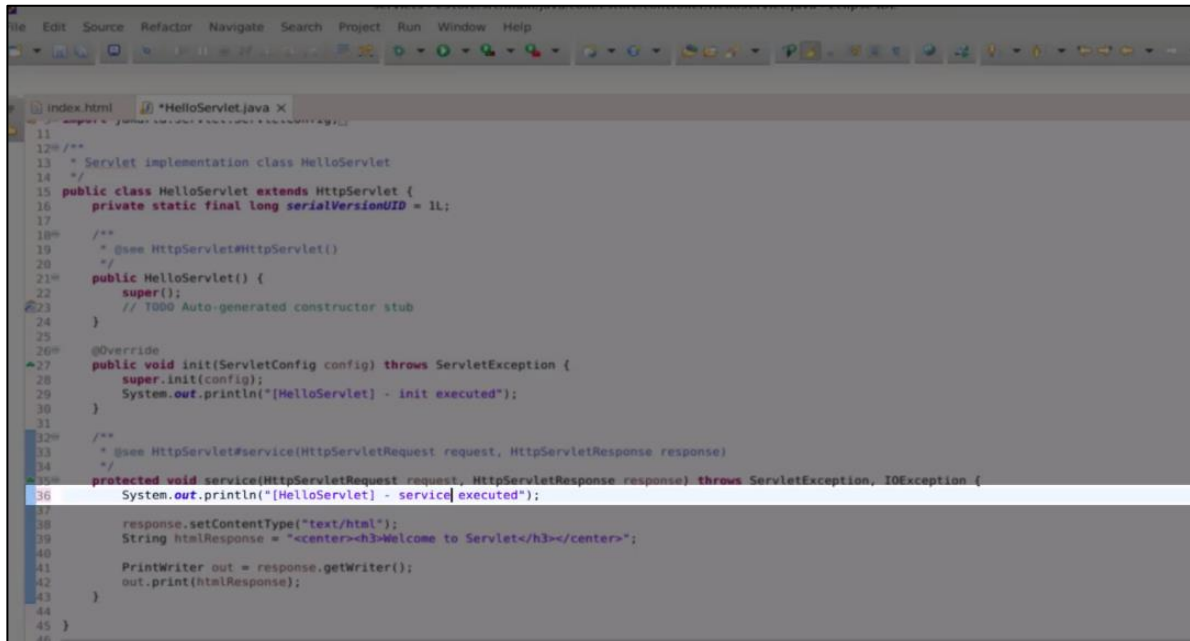
```

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

index.html *HelloServlet.java X
1 package com.estore.controller;
2
3 import jakarta.servlet.ServletConfig;
4
5
6
7
8
9
10
11
12 /**
13  * Servlet implementation class HelloServlet
14  */
15 public class HelloServlet extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     /**
19      * @see HttpServlet#HttpServlet()
20      */
21     public HelloServlet() {
22         super();
23         // TODO Auto-generated constructor stub
24     }
25
26     @Override
27     public void init(ServletConfig config) throws ServletException {
28         super.init(config);
29         System.out.println("[HelloServlet] - init execute");
30     }
31
32     /**
33      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
34      */
35     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
36         System.out.print("Welcome to Servlet");
37
38         response.setContentType("text/html");
39         String htmlResponse = "<center><h3>Welcome to Servlet</h3></center>";
40
41         PrintWriter out = response.getWriter();
42         out.print(htmlResponse);
43     }
44 }

```

## 1.10 Write another **System.out.println()** method to print the message **service executed**



```

11
12 /**
13  * Servlet implementation class HelloServlet
14  */
15 public class HelloServlet extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     /**
19      * @see HttpServlet#HttpServlet()
20      */
21     public HelloServlet() {
22         super();
23         // TODO Auto-generated constructor stub
24     }
25
26     @Override
27     public void init(ServletConfig config) throws ServletException {
28         super.init(config);
29         System.out.println("HelloServlet - init executed");
30     }
31
32     /**
33      * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
34      */
35     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
36         System.out.println("HelloServlet - service executed");
37
38         response.setContentType("text/html");
39         String htmlResponse = "<center><h3>Welcome to Servlet</h3></center>";
40
41         PrintWriter out = response.getWriter();
42         out.print(htmlResponse);
43     }
44 }
45

```

## 1.11 Create the **destroy()** method, which will be executed when the Servlet instance is going to be destructed



```

@Override
public void init(ServletConfig config) throws ServletException {
    super.init(config);
    System.out.println("HelloServlet - init executed");
}

/**
 * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
 */
protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    System.out.println("HelloServlet - service executed");

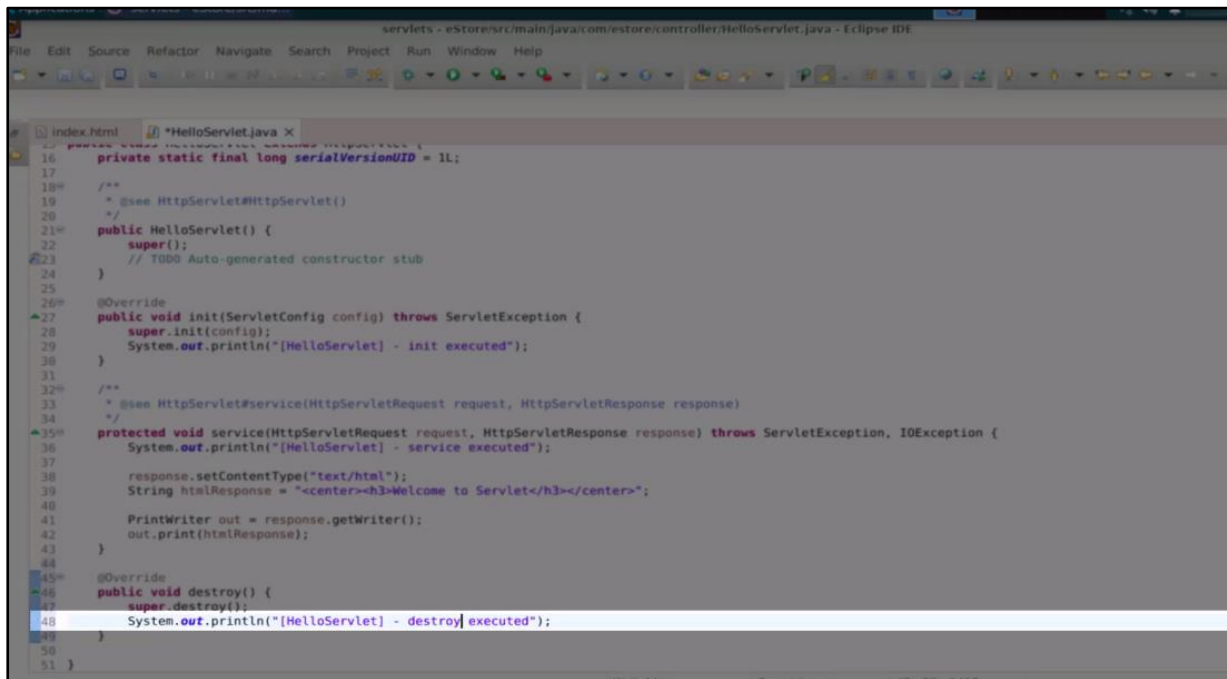
    response.setContentType("text/html");
    String htmlResponse = "<center><h3>Welcome to Servlet</h3></center>";

    PrintWriter out = response.getWriter();
    out.print(htmlResponse);
}

@Override
public void destroy() {
    super.destroy();
}

```

## 1.12 Write the message as **destroy executed**

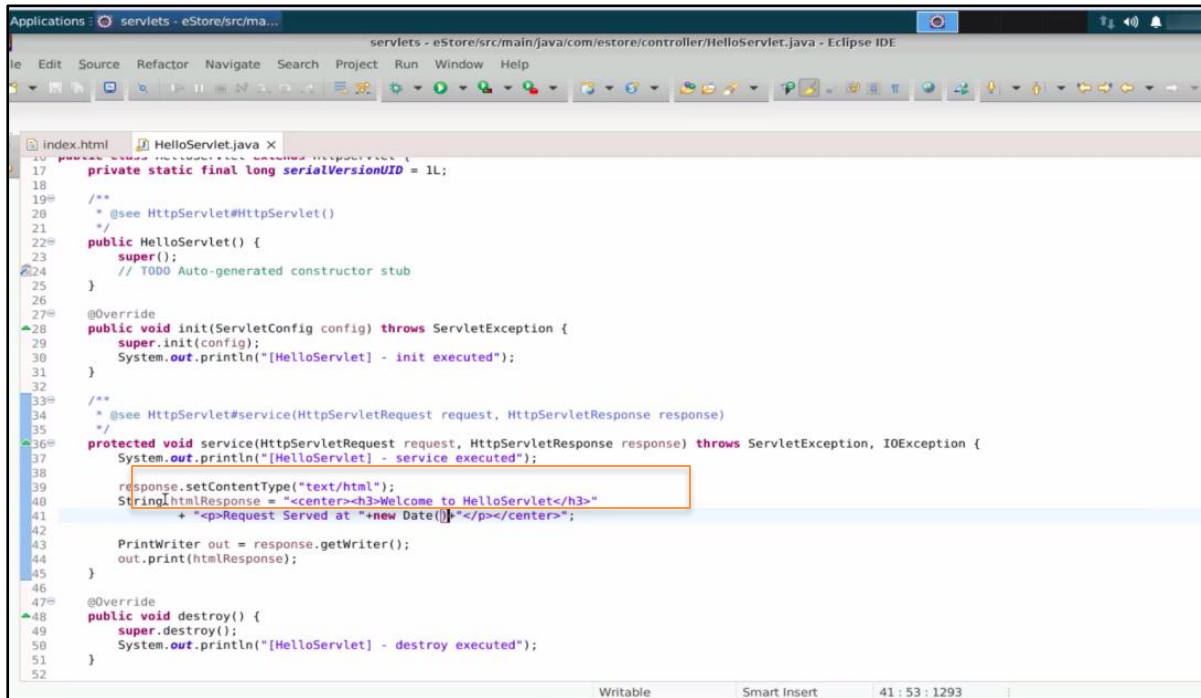


```
16 package com.estore.controller;
17
18 private static final long serialVersionUID = 1L;
19
20 /**
21  * @see HttpServlet#HttpServlet()
22  */
23 public HelloServlet() {
24     super();
25     // TODO Auto-generated constructor stub
26 }
27
28 @Override
29 public void init(ServletConfig config) throws ServletException {
30     super.init(config);
31     System.out.println("[HelloServlet] - init executed");
32 }
33
34 /**
35  * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
36  */
37 protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
38     System.out.println("[HelloServlet] - service executed");
39     response.setContentType("text/html");
40     String htmlResponse = "<center><h3>Welcome to Servlet</h3></center>";
41     PrintWriter out = response.getWriter();
42     out.print(htmlResponse);
43 }
44
45 @Override
46 public void destroy() {
47     super.destroy();
48     System.out.println("[HelloServlet] - destroy executed");
49 }
50
51 }
```



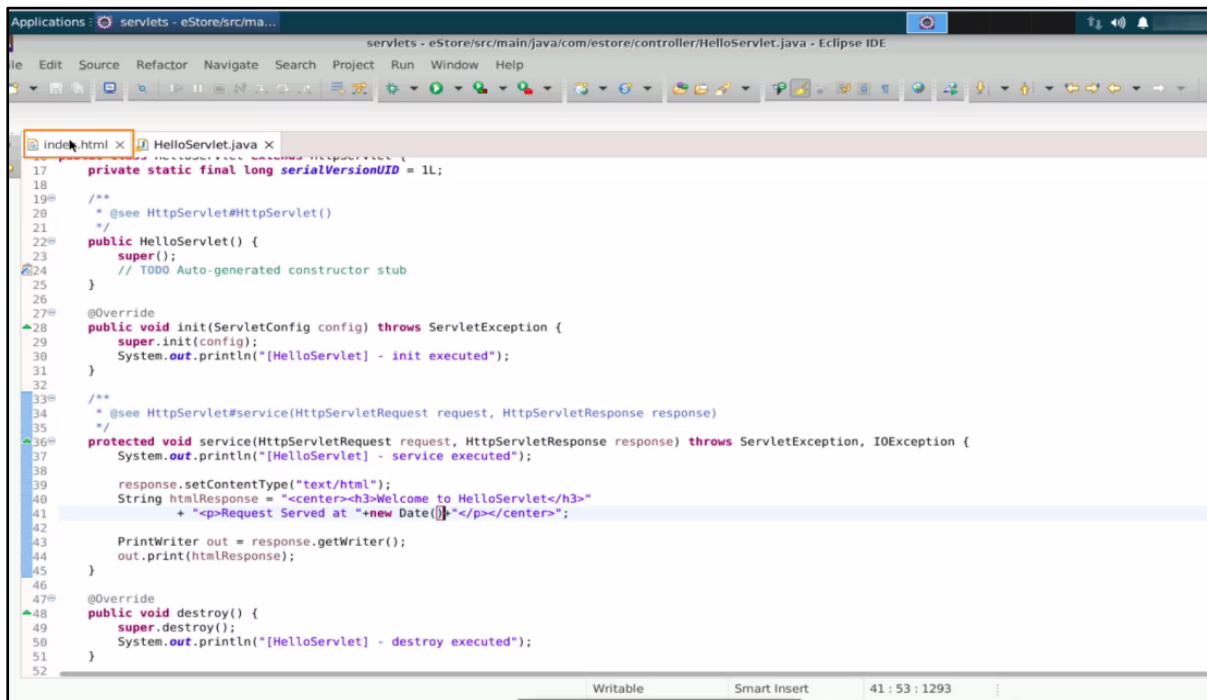
## Step 2: Make changes in internal service

2.1 Update the **HTML response** as **Welcome to HelloServlet**. Add a **paragraph** tag and print the timestamp at which the request was served (lines 39 and 40)



```
10 private void doGet(HttpServletRequest request, HttpServletResponse response) {
11     // TODO Auto-generated method stub
12 }
13
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 @Override
25 public void init(ServletConfig config) throws ServletException {
26     super.init(config);
27     System.out.println("[HelloServlet] - init executed");
28 }
29
30 /**
31  * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
32  */
33 protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
34     System.out.println("[HelloServlet] - service executed");
35
36     response.setContentType("text/html");
37     String htmlResponse = "<center>h3>Welcome to HelloServlet</h3>"
38         + "<p>Request Served at " + new Date() + "</p></center>";
39
40     PrintWriter out = response.getWriter();
41     out.print(htmlResponse);
42 }
43
44 @Override
45 public void destroy() {
46     super.destroy();
47     System.out.println("[HelloServlet] - destroy executed");
48 }
49
50
51
52
```

## 2.2 Open the `index.html` page



The screenshot shows the Eclipse IDE interface. In the Project Explorer on the left, the `index.html` file is selected. The main editor displays the `HelloServlet.java` file. The code is as follows:

```

17 private static final long serialVersionUID = 1L;
18
19 /**
20  * @see HttpServlet#HttpServlet()
21  */
22 public HelloServlet() {
23     super();
24     // TODO Auto-generated constructor stub
25 }
26
27 @Override
28 public void init(ServletConfig config) throws ServletException {
29     super.init(config);
30     System.out.println("[HelloServlet] - init executed");
31 }
32
33 /**
34  * @see HttpServlet#service(HttpServletRequest request, HttpServletResponse response)
35  */
36 protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37     System.out.println("[HelloServlet] - service executed");
38
39     response.setContentType("text/html");
40     String htmlResponse = "<center><h3>Welcome to HelloServlet</h3>"
41         + "<p>Request Served at " + new Date() + "</p></center>";
42
43     PrintWriter out = response.getWriter();
44     out.print(htmlResponse);
45 }
46
47 @Override
48 public void destroy() {
49     super.destroy();
50     System.out.println("[HelloServlet] - destroy executed");
51 }
52

```

The status bar at the bottom indicates the file is "Writable", "Smart Insert" is active, and the cursor is at line 41, column 53, position 1293.

## Step 3: Create a hyperlink for Hello Servlet

### 3.1 Create a href link that will send a request to the HelloServlet

The screenshot shows the Eclipse IDE with the `index.html` file open. The code is as follows:

```

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 eStore</h3>
11 <h3>Login Here with your Details</h3>
12
13 <form action="Login" method="get">
14 <b>Enter Your Email</b><br>
15 <input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
16 <b>Enter Your Password</b><br>
17 <input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
18 <br>
19 <input type="submit" value="LOGIN"/>
20
21 </form>
22
23 <br>
24 <a href="register.html">New User? Register Here</a>
25 <br>
26 <a href="register.html">New User? Register Here</a>
27
28 </center>
29
30 </body>
31 </html>

```

The status bar at the bottom indicates the cursor is at `html/body/center/a/href`.

### 3.2 Open the web.xml file and copy the <url-pattern> for HelloServlet

The screenshot shows the Eclipse IDE with the `web.xml` file open. The code is as follows:

```

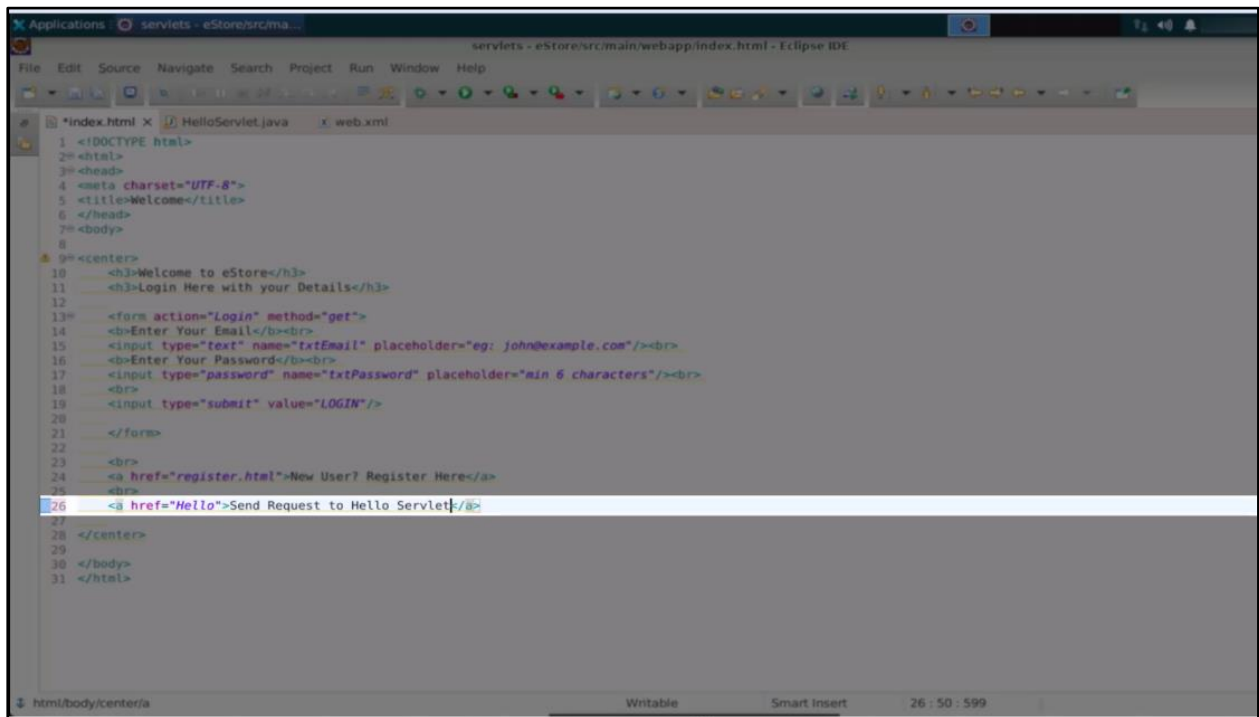
<?xml version="1.0" encoding="UTF-8"
  (module-name? | (((description*, display-name*, icon*)) | distribut
    http://www.w3.org/2001/XMLSchema-instance
    https://jakarta.ee/xml/ns/jakartaee
    http://xmlns.jcp.org/xml/ns/javaee
    https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee
    WebApp_ID
    5.0
    eStore
    (welcome-file+))
  Design Source

```

The status bar at the bottom indicates the cursor is at `<terminated> apache-tomcat-10.0.13 at localhost [Apache Tomcat] /usr/eclipse/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.linux.x86_64_17.0`.

```
index.html  HelloServlet.java  web.xml X
5      <welcome-file>index.html</welcome-file>
6      </welcome-file-list>
7      <servlet>
8          <description></description>
9          <display-name>HelloServlet</display-name>
10         <servlet-name>HelloServlet</servlet-name>
11         <servlet-class>com.estore.controller.HelloServlet</servlet-class>
12     </servlet>
13     <servlet-mapping>
14         <servlet-name>HelloServlet</servlet-name>
15         <url-pattern>/HelloServlet</url-pattern>
16         <url-pattern>/Hello</url-pattern>
17         <url-pattern>/Welcome</url-pattern>
18     </servlet-mapping>
19     <servlet>
20         <description></description>
21         <display-name>LoginServlet</display-name>
22         <servlet-name>LoginServlet</servlet-name>
23         <servlet-class>com.estore.controller.LoginServlet</servlet-class>
24     </servlet>
25     <servlet-mapping>
26         <servlet-name>LoginServlet</servlet-name>
27         <url-pattern>/LoginServlet</url-pattern>
28         <url-pattern>/Login</url-pattern>
29     </servlet-mapping>
30     <servlet>
31         <description></description>
32         <display-name>RegsiterServlet</display-name>
33         <servlet-name>RegsiterServlet</servlet-name>
34         <servlet-class>com.estore.controller.RegisiterServlet</servlet-class>
35     </servlet>
36     <servlet-mapping>
```

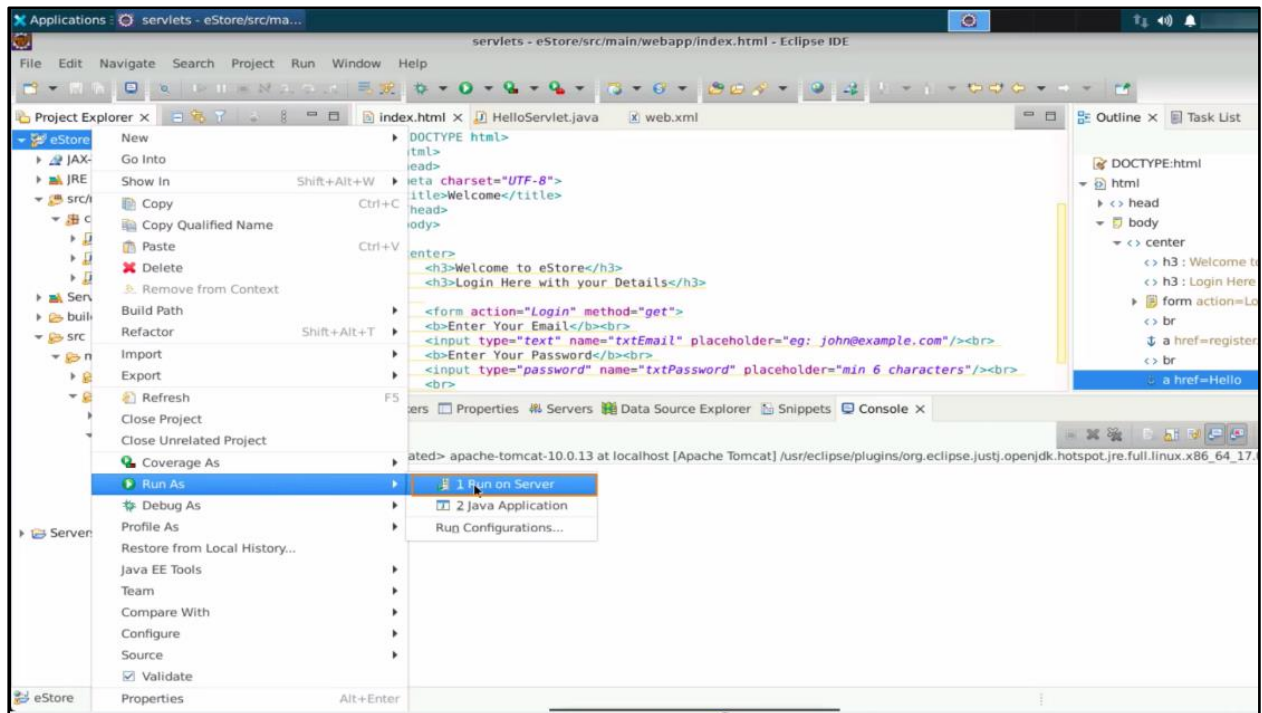
### 3.3 Go back to the `index.html` page and add the URL mapping inside the `href` tag



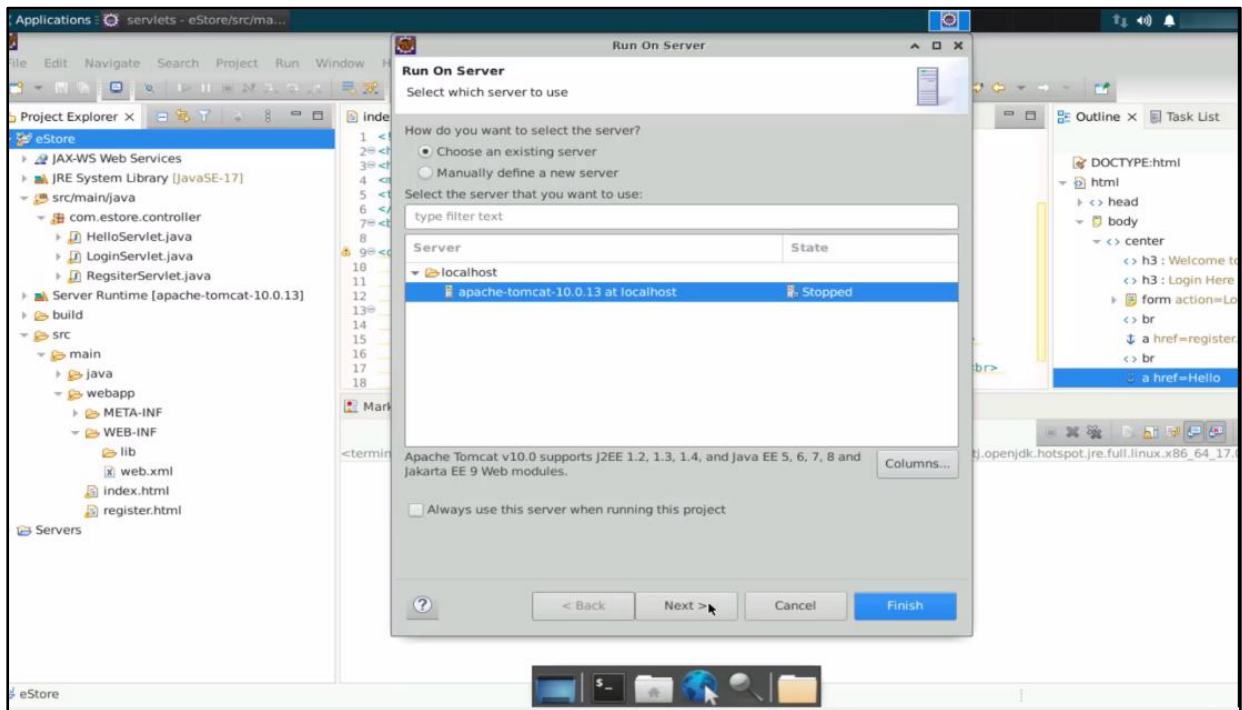
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Welcome</title>
6 </head>
7 <body>
8
9 <center>
10 <h3>Welcome to eStore</h3>
11 <h3>Login Here with your Details</h3>
12
13 <form action="Login" method="get">
14 <b>Enter Your Email</b><br>
15 <input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
16 <b>Enter Your Password</b><br>
17 <input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
18 <br>
19 <input type="submit" value="LOGIN"/>
20
21 </form>
22
23 <br>
24 <a href="register.html">New User? Register Here</a>
25 <br>
26 <a href="Hello">Send Request to Hello Servlet</a>
27
28 </center>
29
30 </body>
31 </html>
```

## Step 4: Run code to check how the lifecycle method works

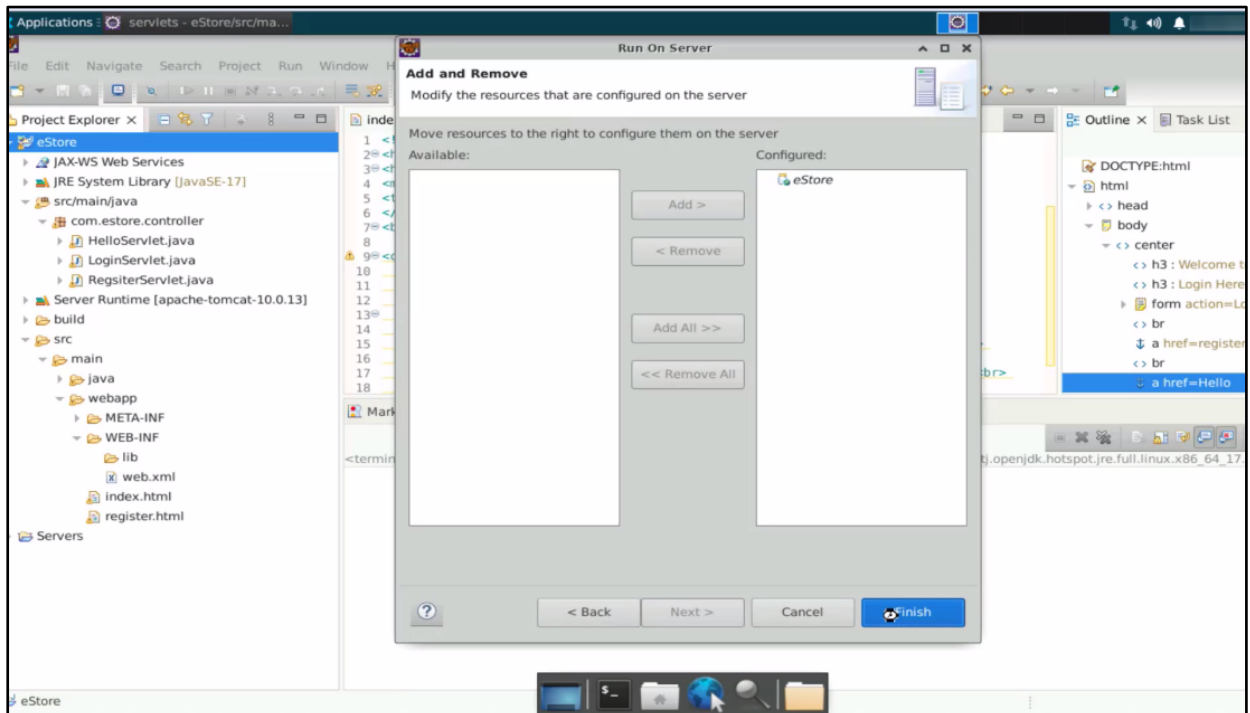
4.1 Save and run the code on the server. Right-click on the project, click on **Run As**, and select **Run on Server**



## 4.2 Choose the **apache-tomcat-10.0.13** at **localhost** server and click **Next**

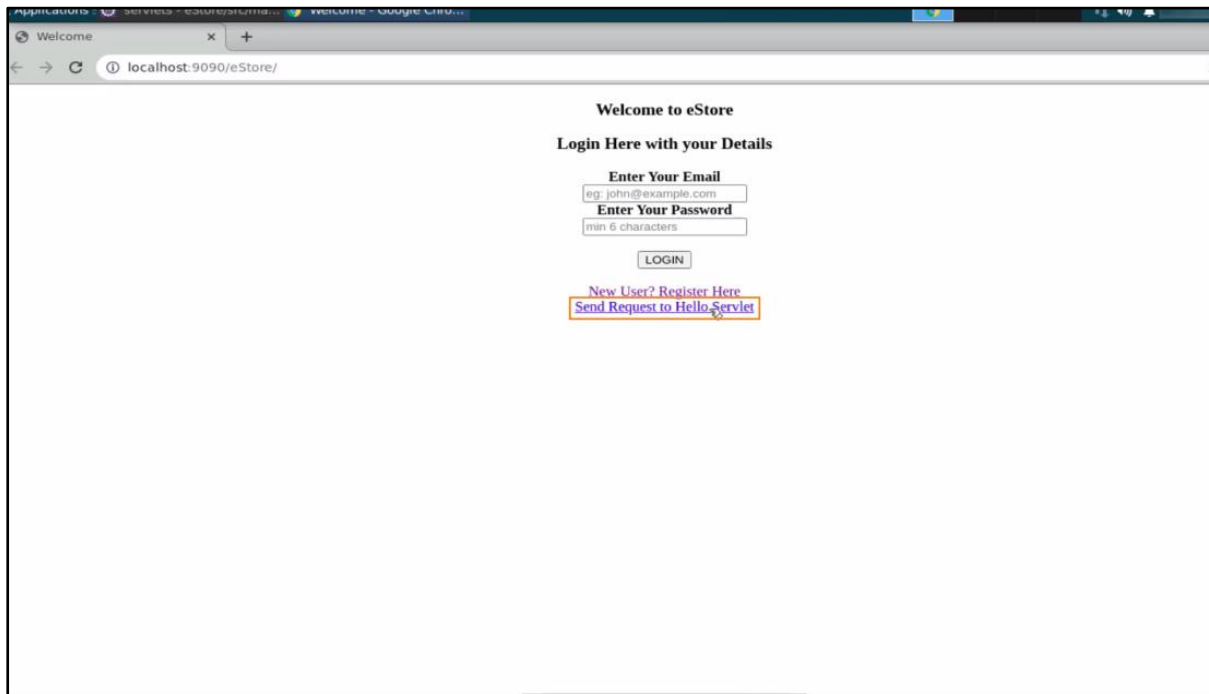


## 4.3 Click **Finish**

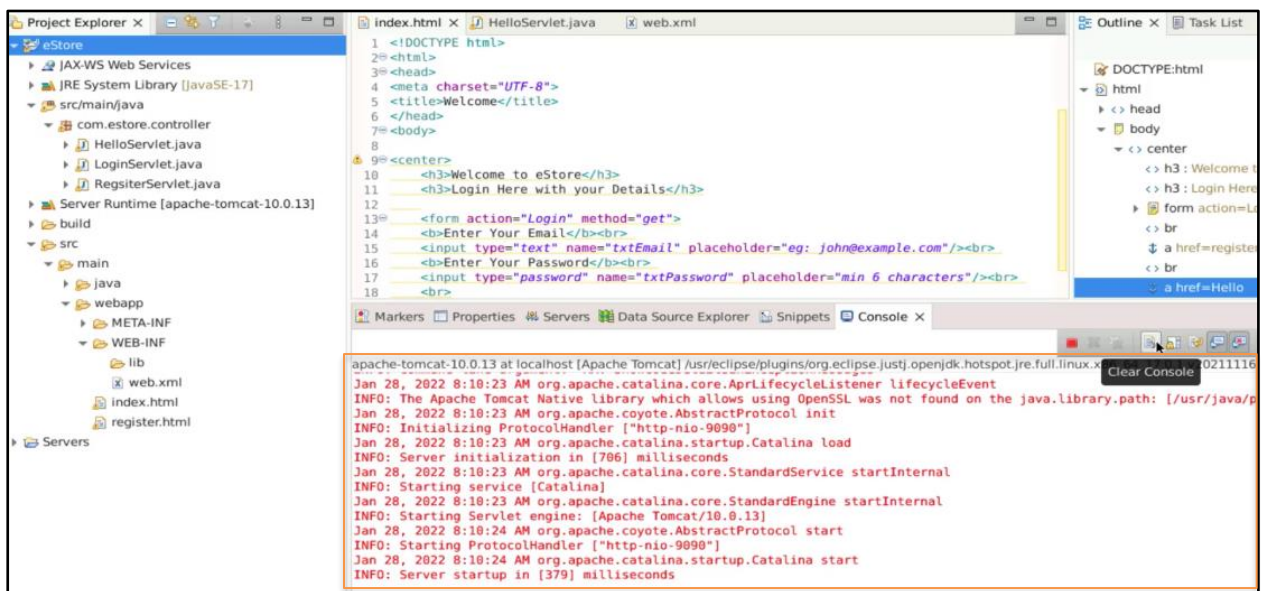




You can see the hyperlink as **Send Request to Hello Servlet**.

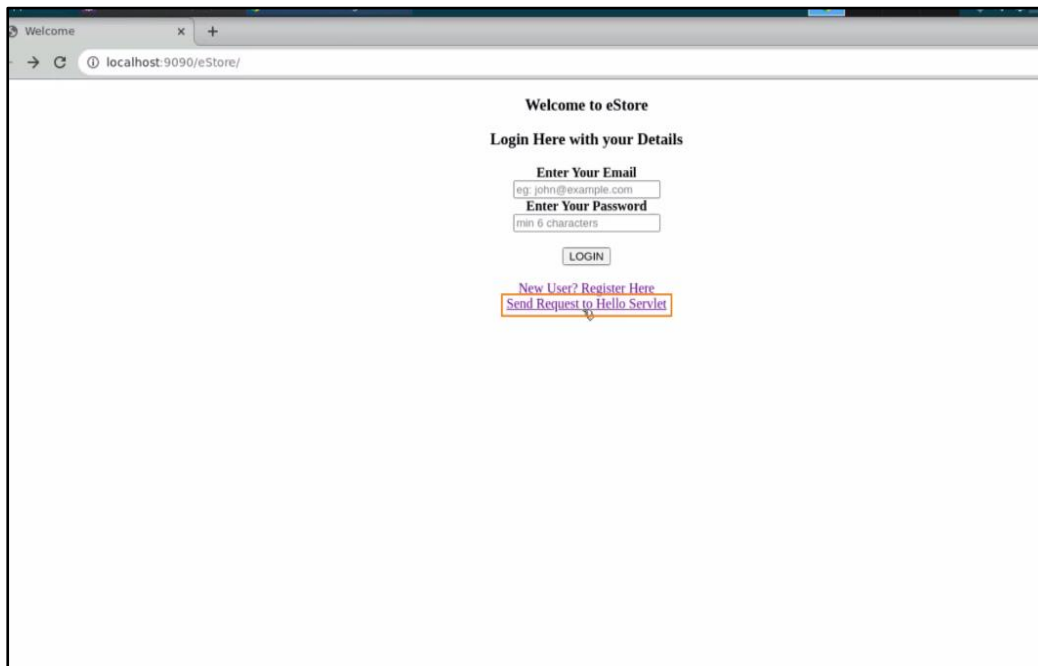


4.4 Go back to **Eclipse IDE** and check the **Console** tab. You will not be able to see any log or print statement as shown below:

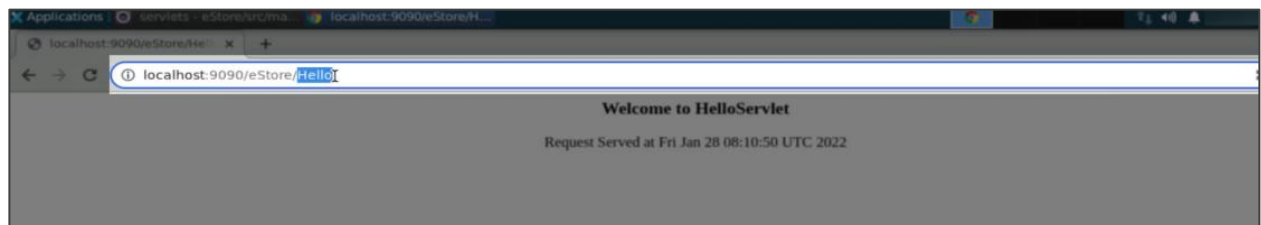




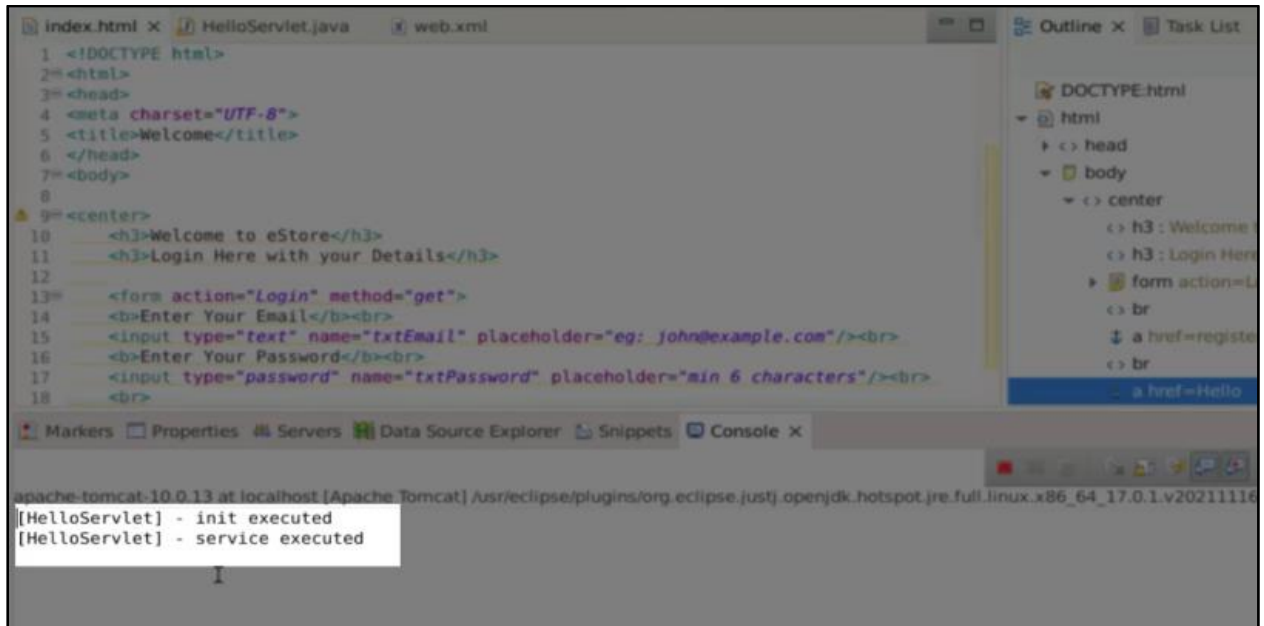
#### 4.5 Go back and click on the **Send Request to Hello Servlet** link



You can see that the **HelloRequest** has been forwarded to the **Hello Servlet**, and it prints the timestamp of when the request was served.



4.6 Go back to the Console tab. You can see that the two methods, init and service, are executed.



```

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 eStore</h3>
11 <h3>Login Here with your Details</h3>
12
13 <form action="Login" method="get">
14 <b>Enter Your Email</b><br>
15 <input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
16 <b>Enter Your Password</b><br>
17 <input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
18 </form>

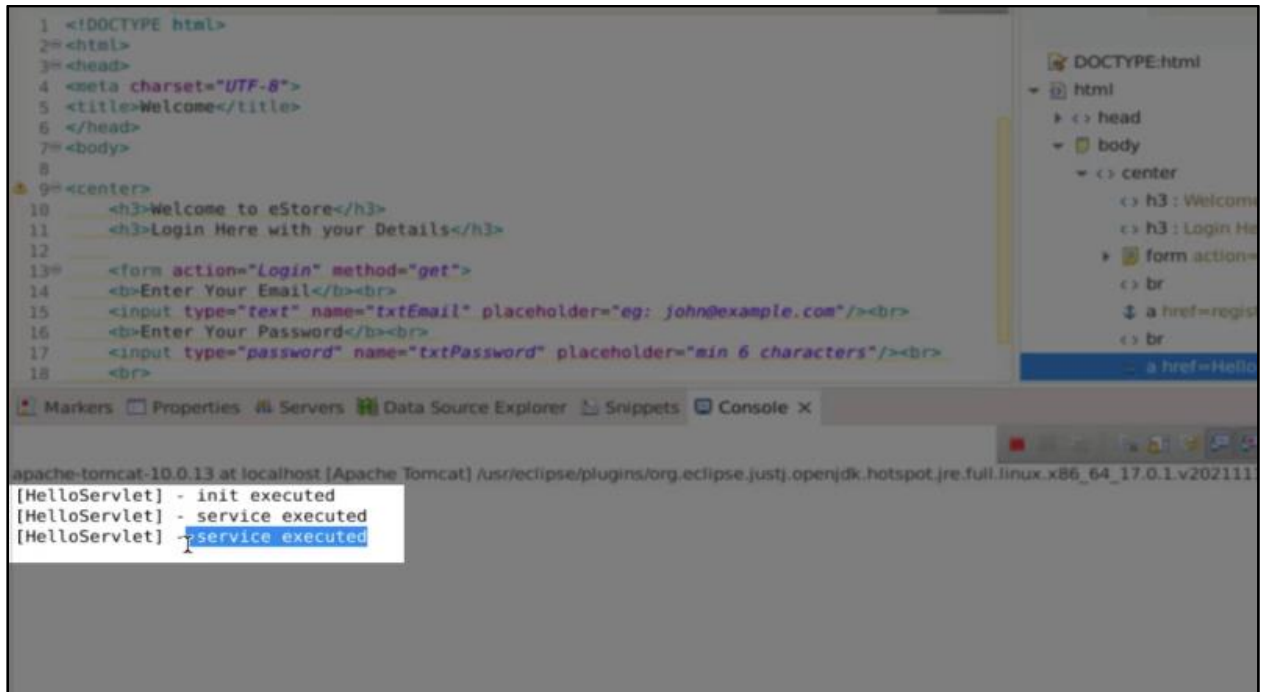
```

```

[HelloServlet] - init executed
[HelloServlet] - service executed

```

4.7 Reclick on the hyperlink in the browser. You will see one more service executed in the **Console** tab, as the init method gets executed only once.



The screenshot shows an IDE with two main panels. The top panel displays an HTML file named 'DOCTYPE.html'. The code in the file is as follows:

```

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 eStore</h3>
11 <h3>Login Here with your Details</h3>
12
13 <form action="Login" method="get">
14 <b>Enter Your Email</b><br>
15 <input type="text" name="txtEmail" placeholder="eg: john@example.com"/><br>
16 <b>Enter Your Password</b><br>
17 <input type="password" name="txtPassword" placeholder="min 6 characters"/><br>
18 <br>

```

The right panel shows a tree view of the HTML document structure, including 'html', 'head', 'body', 'center', and 'form' elements.

The bottom panel shows the 'Console' tab with the following output:

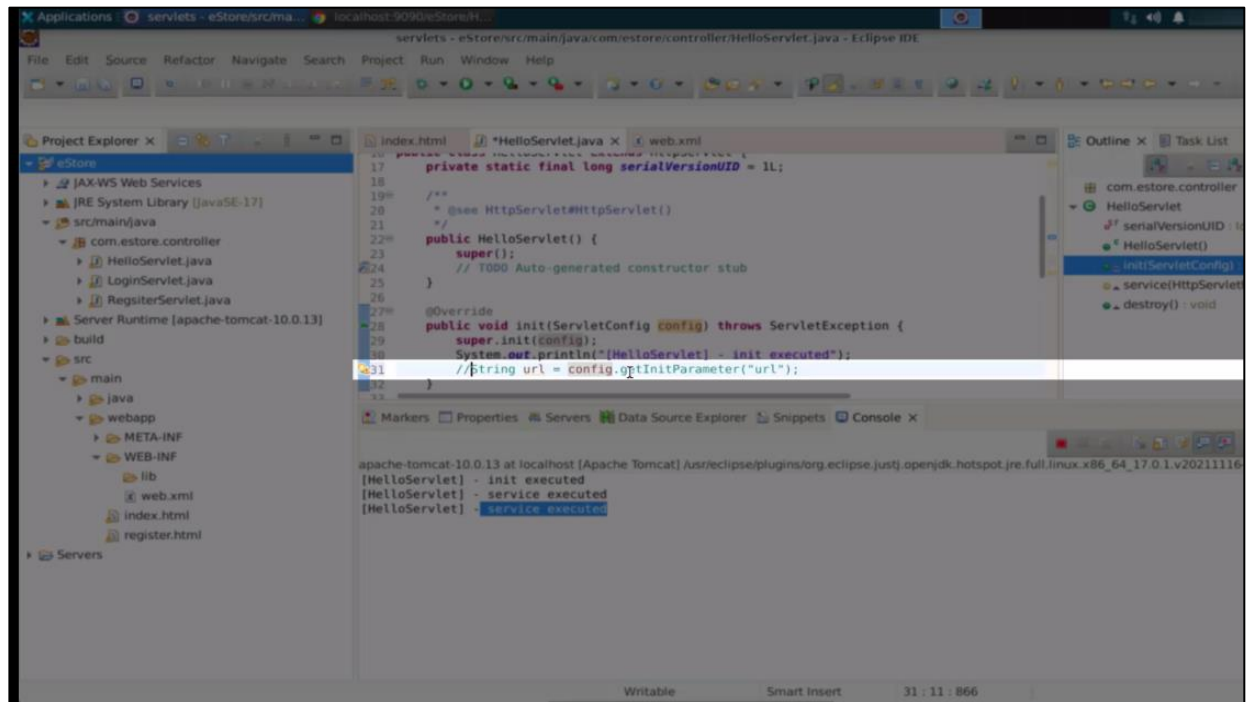
```

apache-tomcat-10.0.13 at localhost [Apache Tomcat] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.1.v2021111
[HelloServlet] - init executed
[HelloServlet] - service executed
[HelloServlet] - service executed

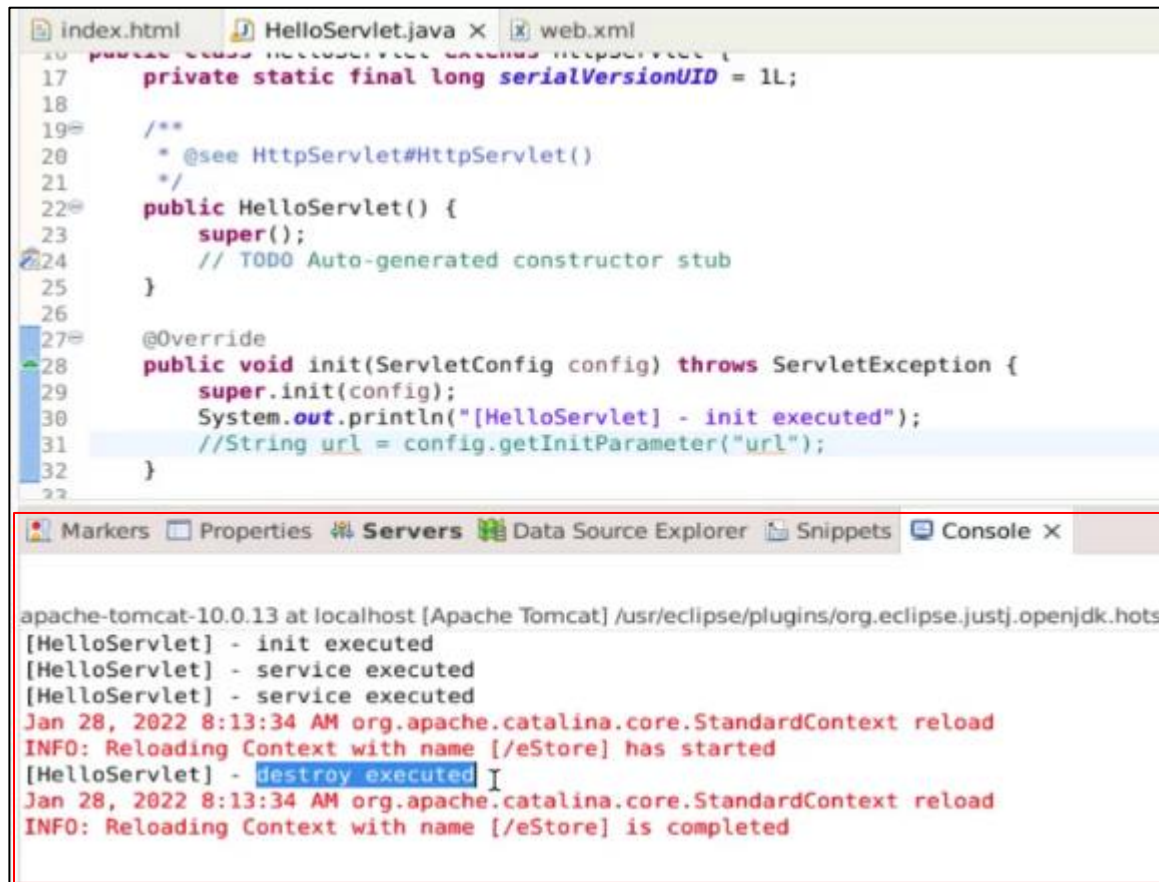
```

The third line of the console output, '[HelloServlet] - service executed', is highlighted with a blue selection box.

4.8 In **HelloServlet.java**, use the **config** method and get the **init** parameter from the server, which can be a URL



4.9 Go to the **Console** tab in the **HelloServlet.java** file. You will see that the Servlet is automatically destroyed, and it prints **destroy executed**.



The screenshot shows the Eclipse IDE with the **HelloServlet.java** file open. The code defines a **HelloServlet** class that extends **HttpServlet**. It includes a **serialVersionUID**, a constructor, and an **init** method. The **init** method prints **[HelloServlet] - init executed**. The **Console** tab at the bottom shows the following output:

```
apache-tomcat-10.0.13 at localhost [Apache Tomcat] /usr/eclipse/plugins/org.eclipse.justj.openjdk.hotspot
[HelloServlet] - init executed
[HelloServlet] - service executed
[HelloServlet] - service executed
Jan 28, 2022 8:13:34 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/eStore] has started
[HelloServlet] - destroy executed
Jan 28, 2022 8:13:34 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/eStore] is completed
```

**Observation:** There are two important callbacks associated with the Servlet lifecycle: the **init** and **destroy** methods. Between these two steps, an HTTP method should be implemented. The **init** and **destroy** methods will be executed only once.