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

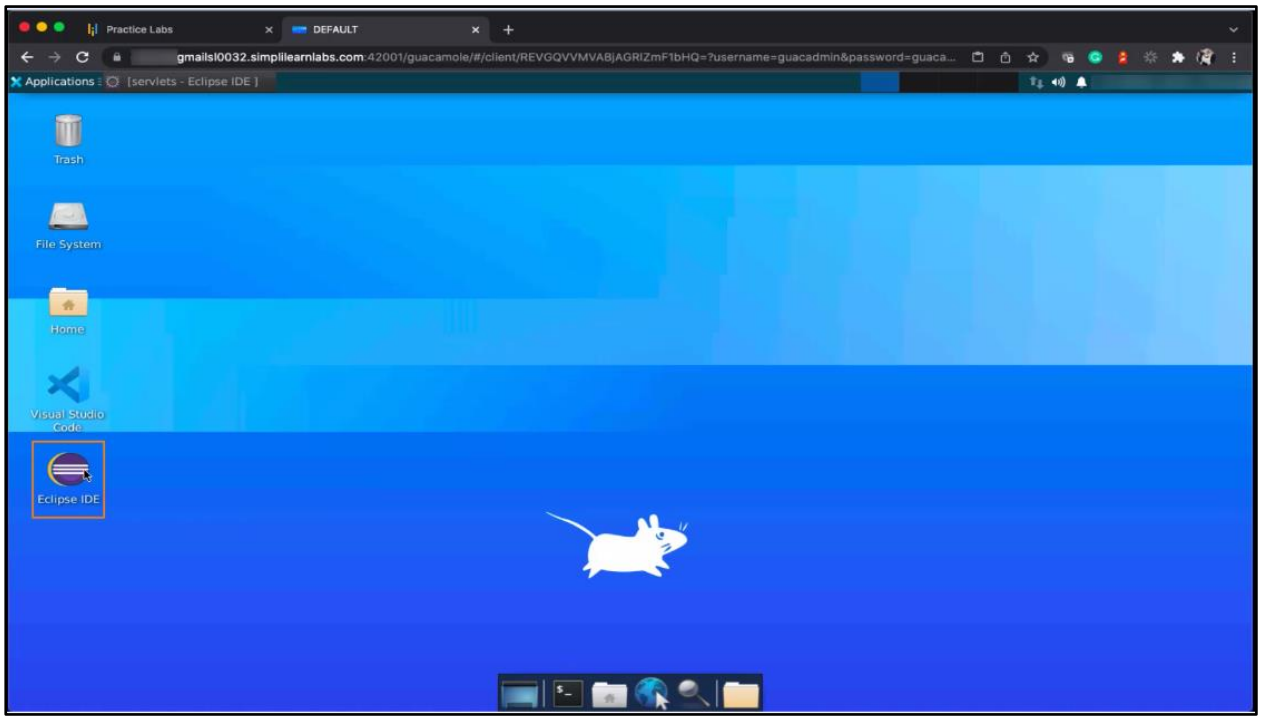
Prerequisites: None

Steps to be followed:

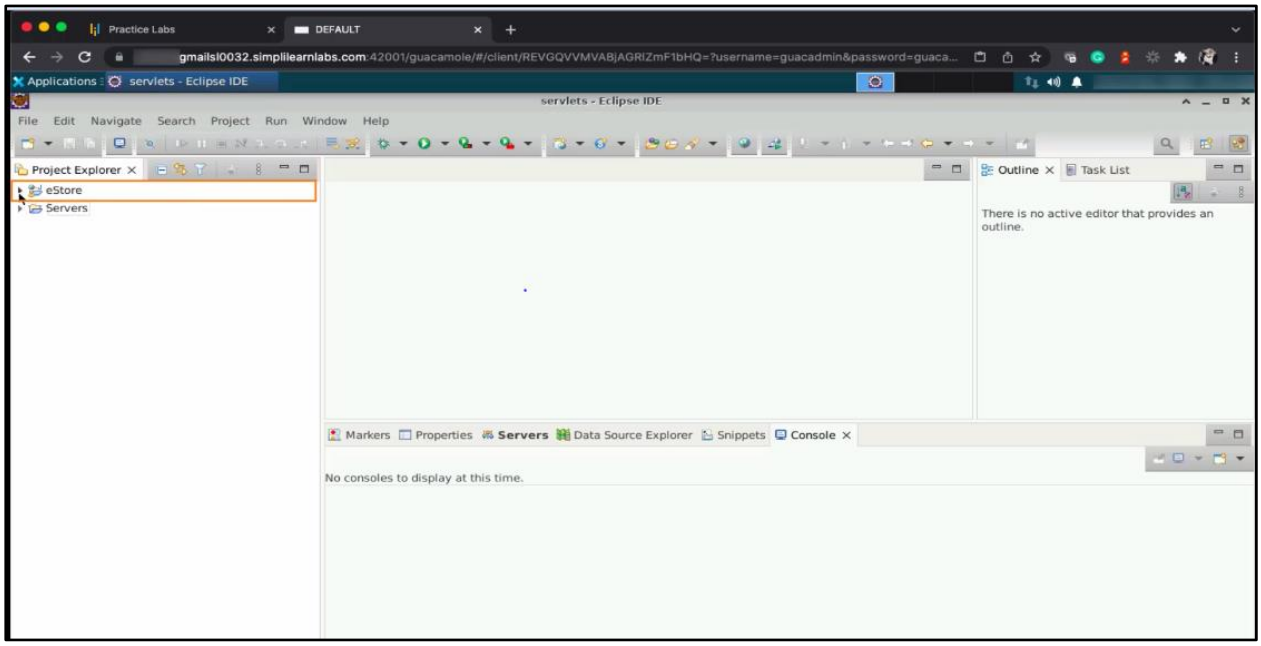
1. Create the init and destroy method
2. Make changes in internal service
3. Create a hyperlink for Hello Servlet
4. Run code to check how the life cycle works

Step 1: Create the init and destroy method

1.1 Open **Eclipse IDE**

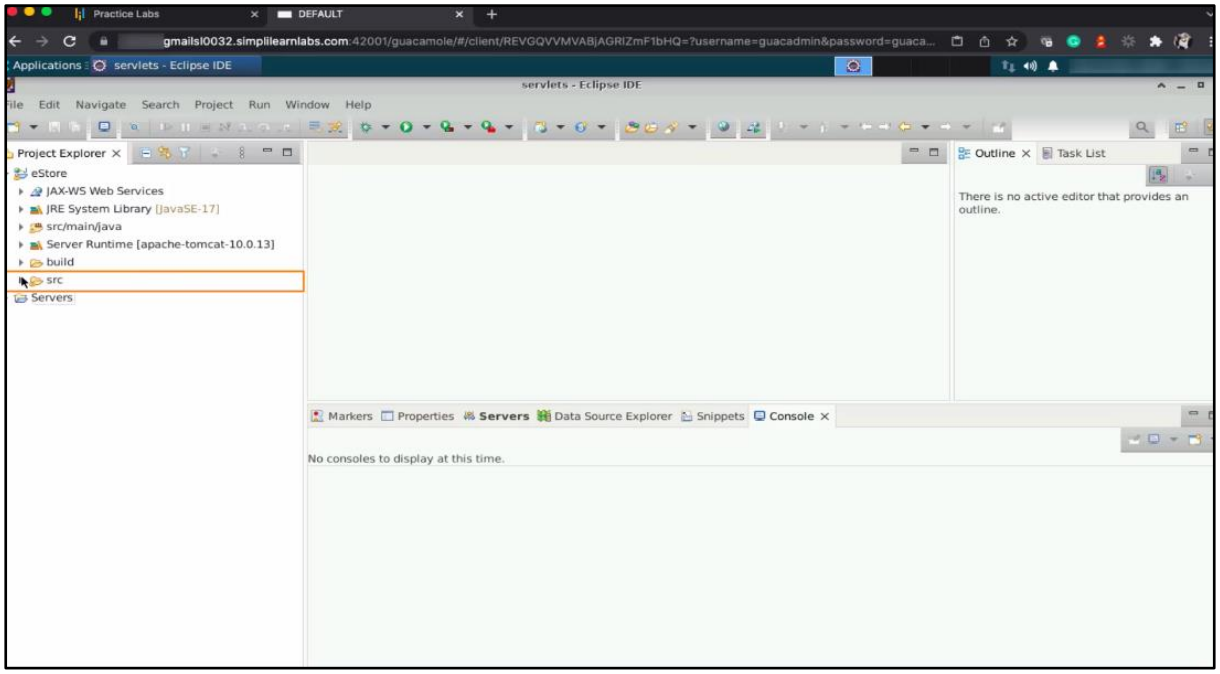


1.2 Open the **eStore** project

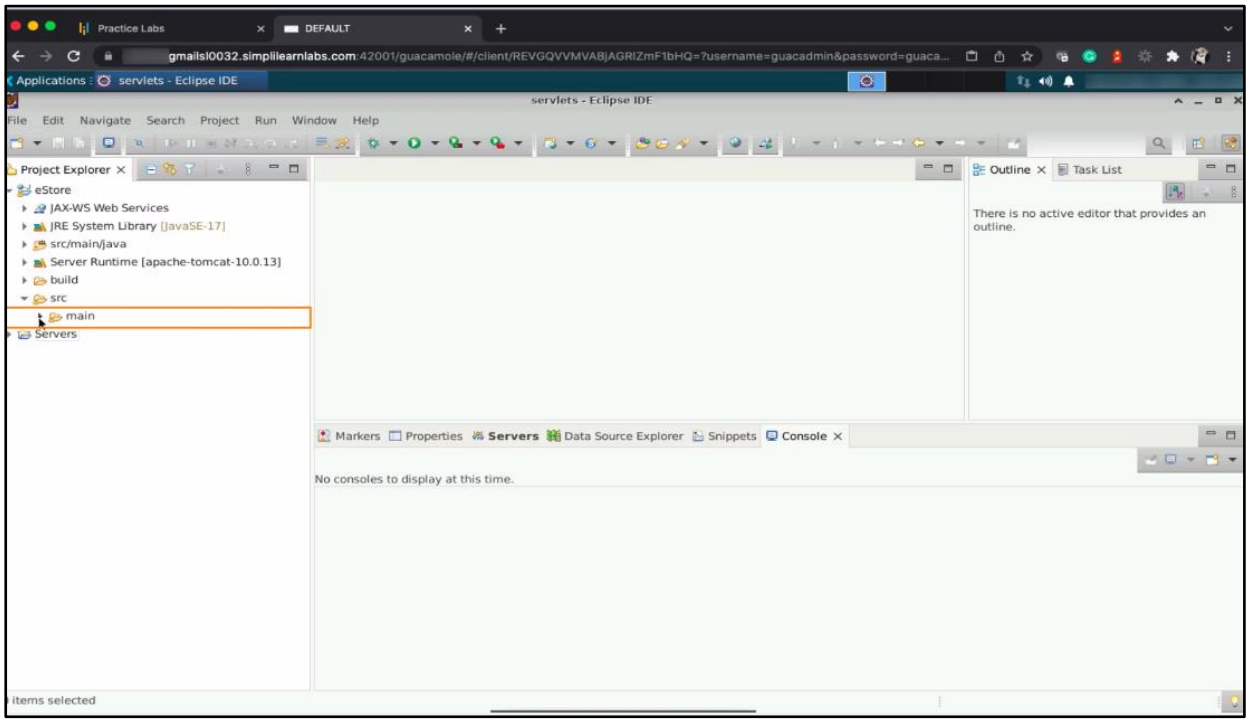


Note: Please refer to the previous demos on how to create the **eStore** project

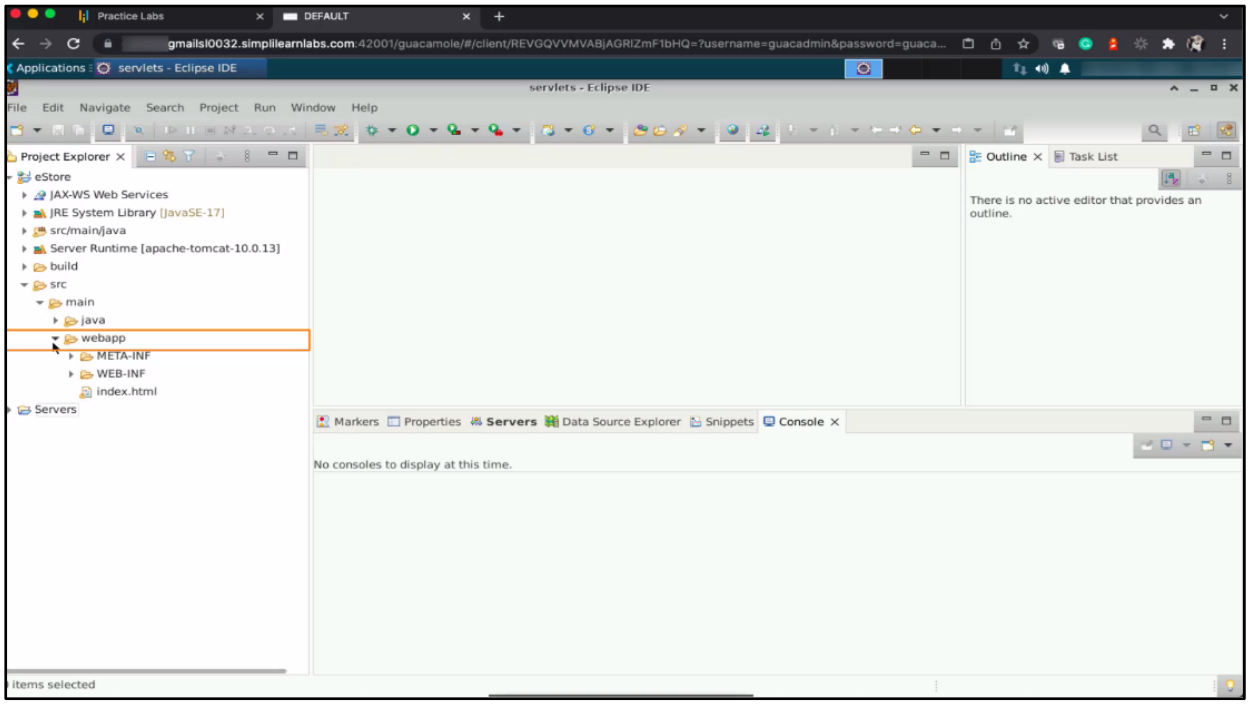
1.3 Go to the **src** directory



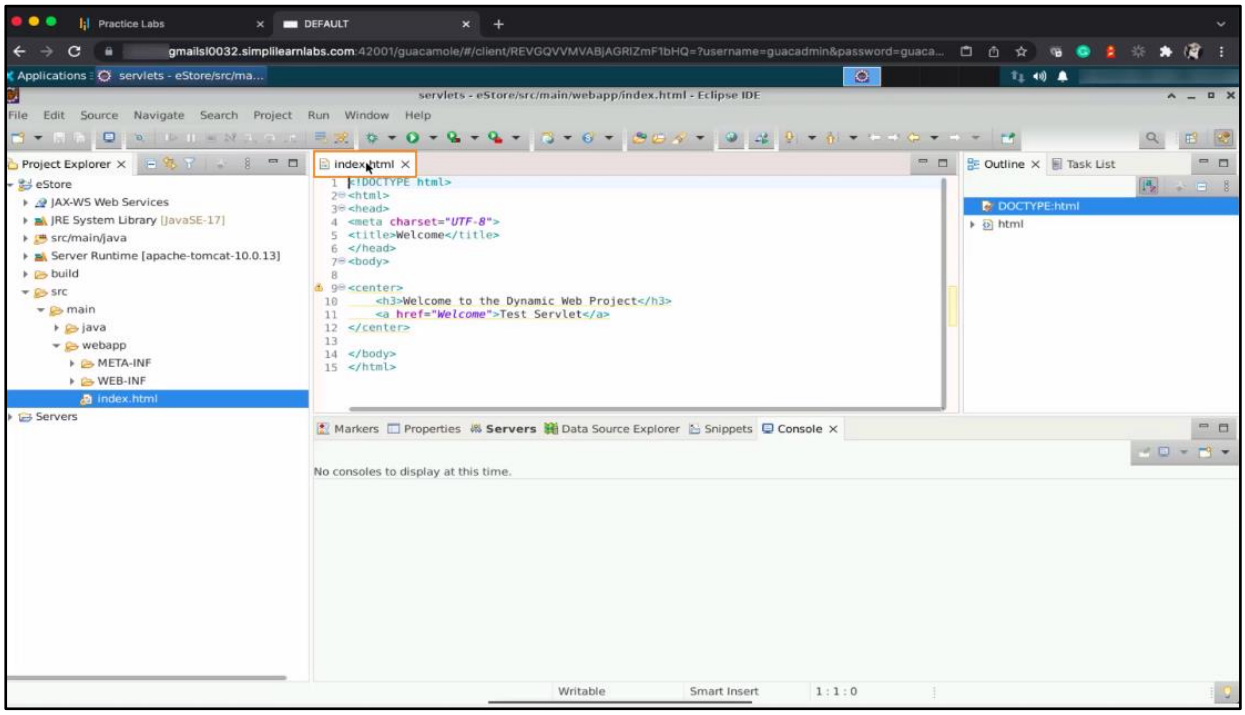
1.4 Select the **main** folder



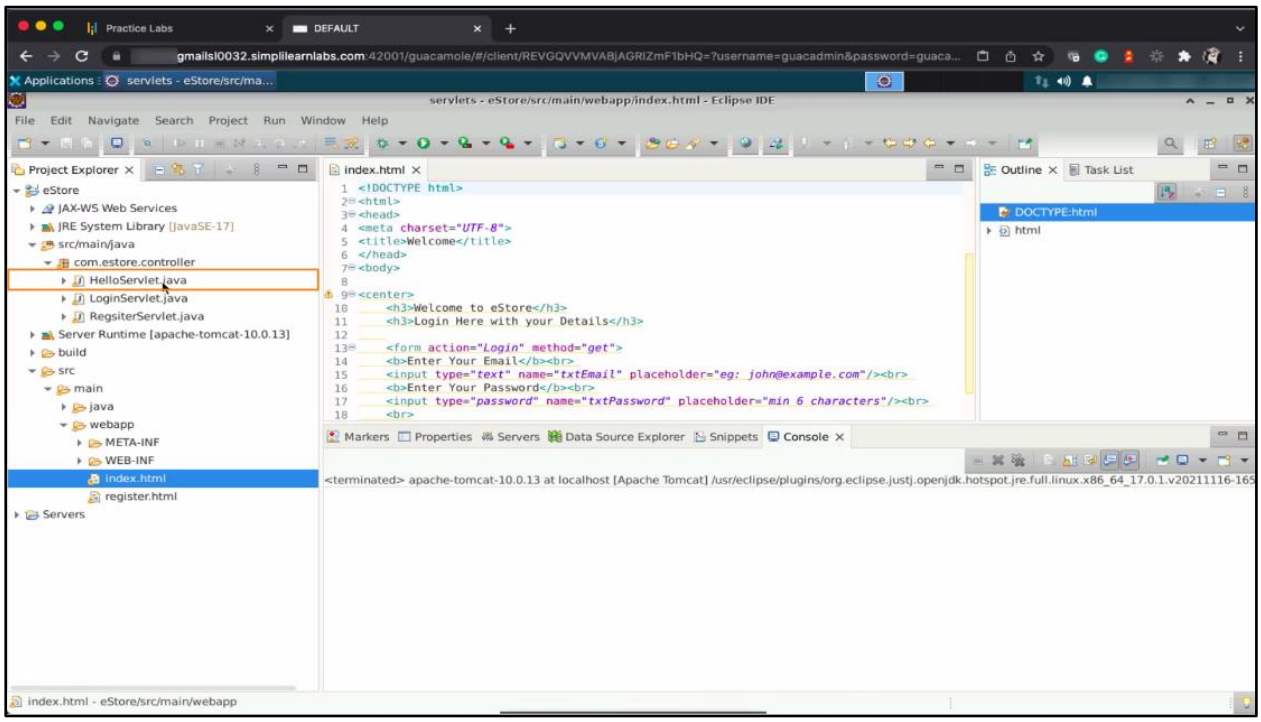
1.5 Open the **webapp** folder



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



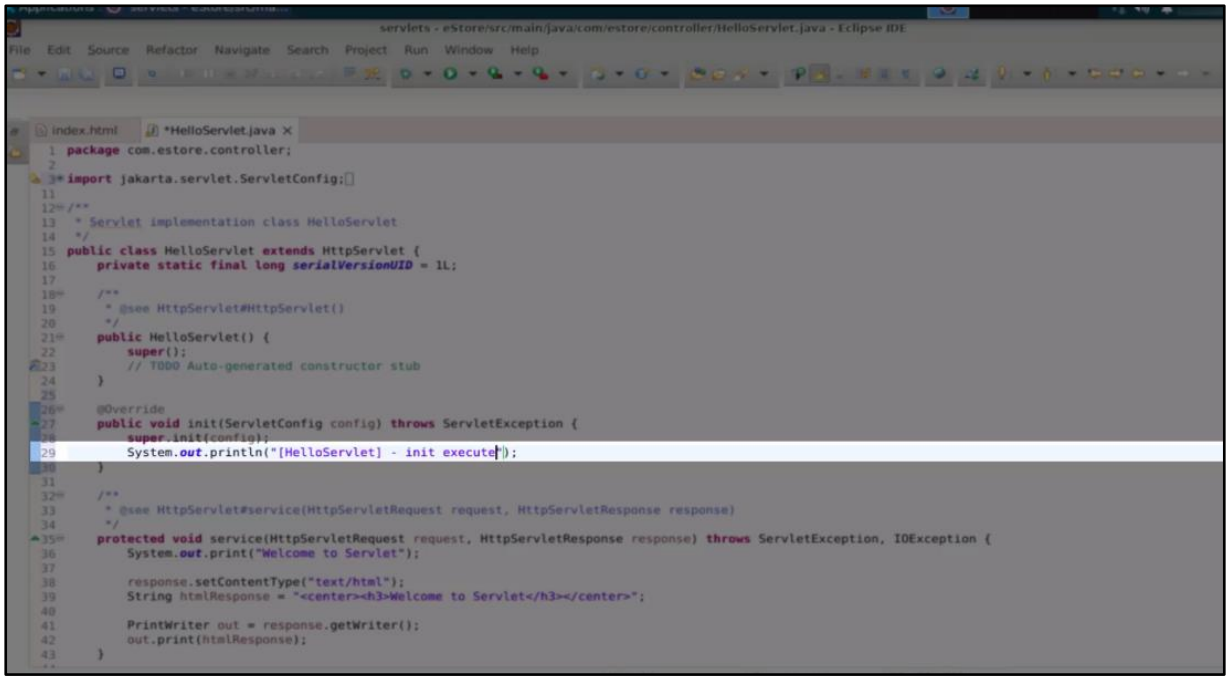
1.7 Now, open the **HelloServlet.java** file to define the methods



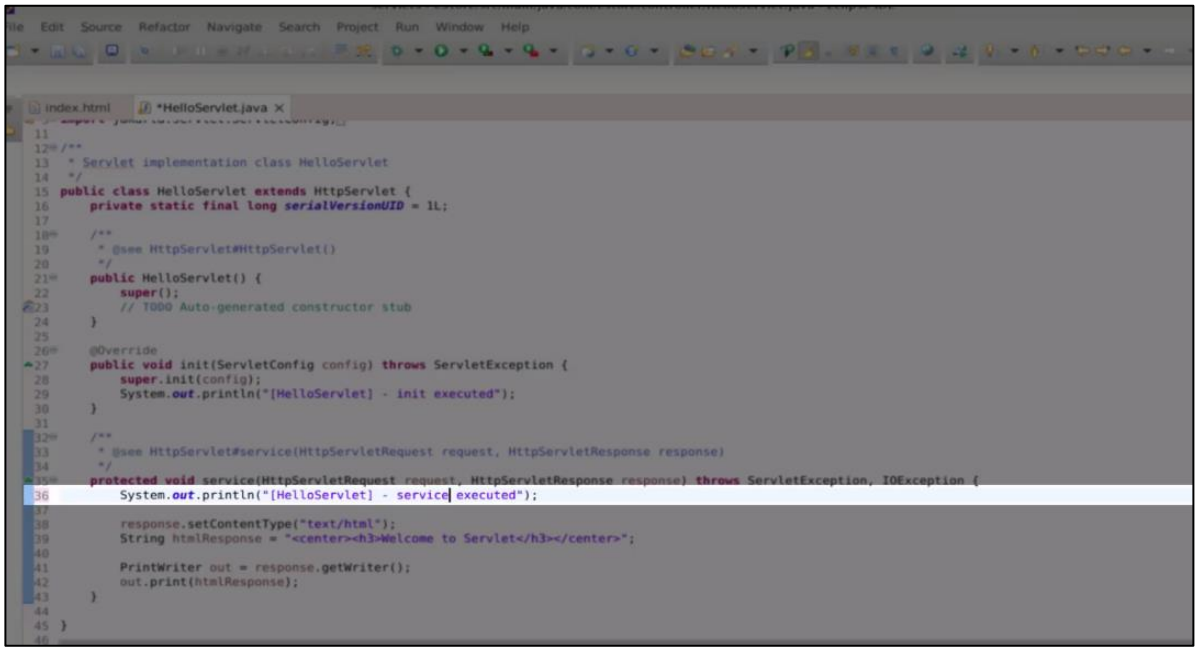
1.8 Define the method **init()** and override it in the parent class **HTTPServlet**

```
dex.html   *HelloServlet.java ×
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



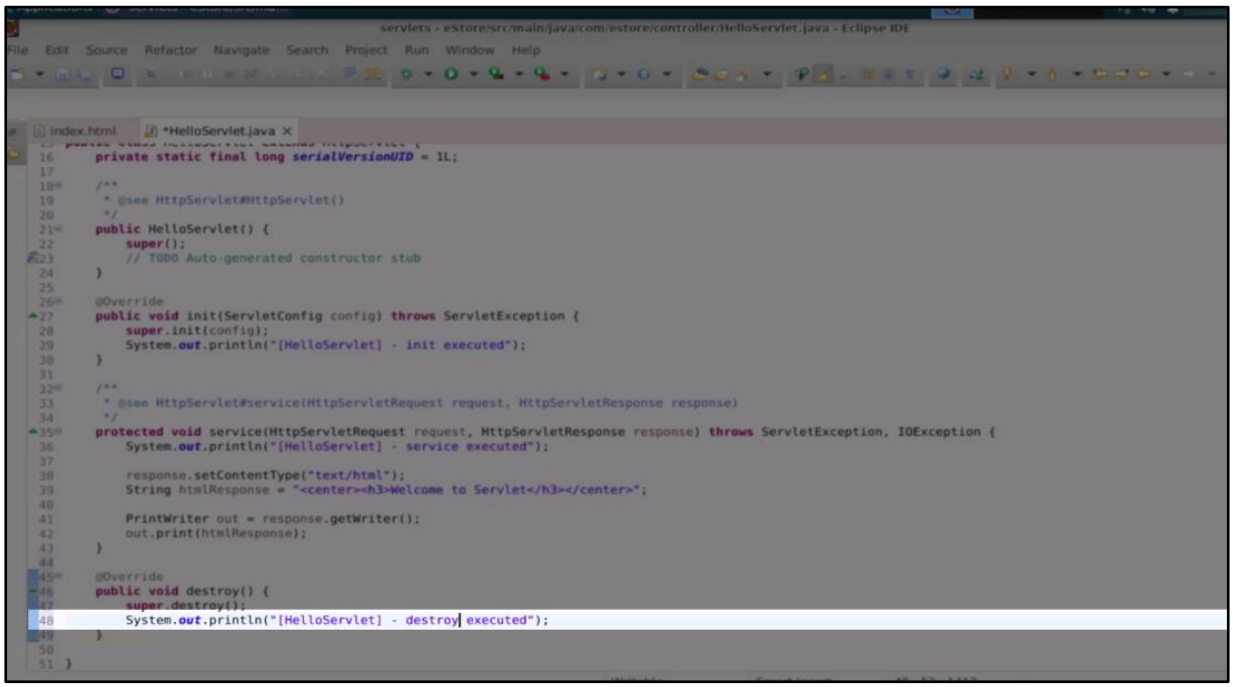
1.10 Write another **System.out.println()** method to print the message **service executed**



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

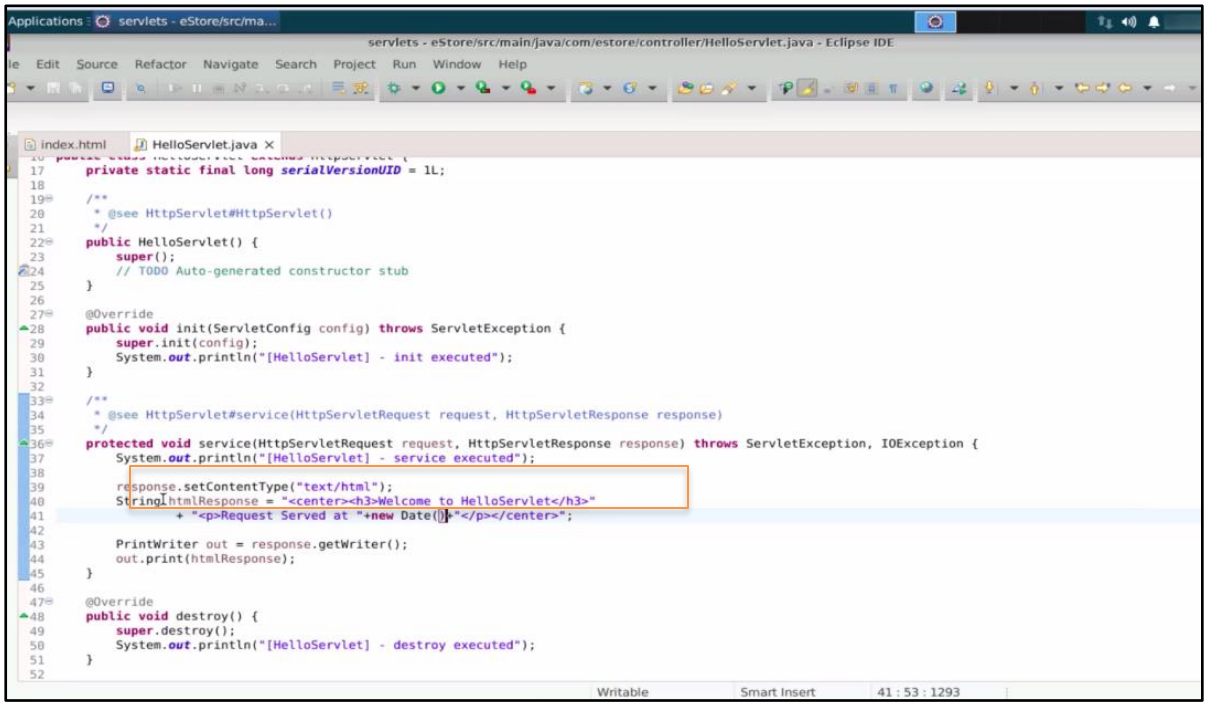
```
@Override
public void init(ServletConfig config) throws ServletException {
    super.init(config);
    System.out.println("[HelloServlet] - init executed");
}

/**
 * @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**

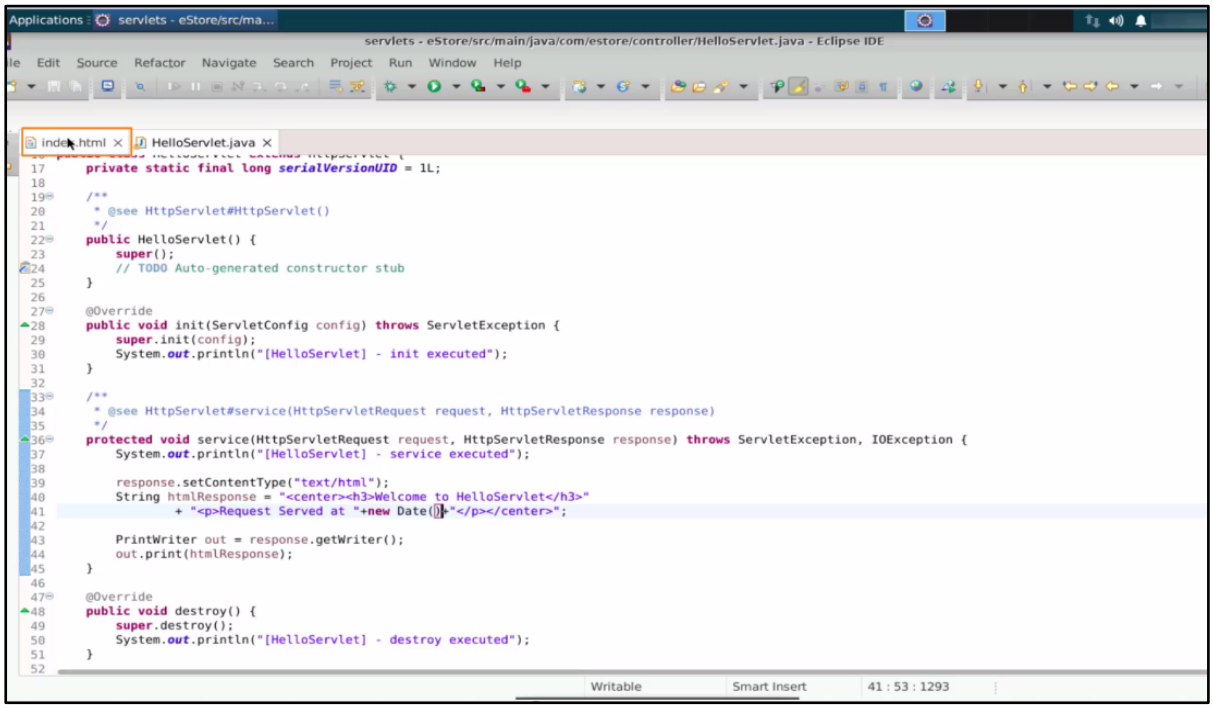


Step 2: Make changes in internal service

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)

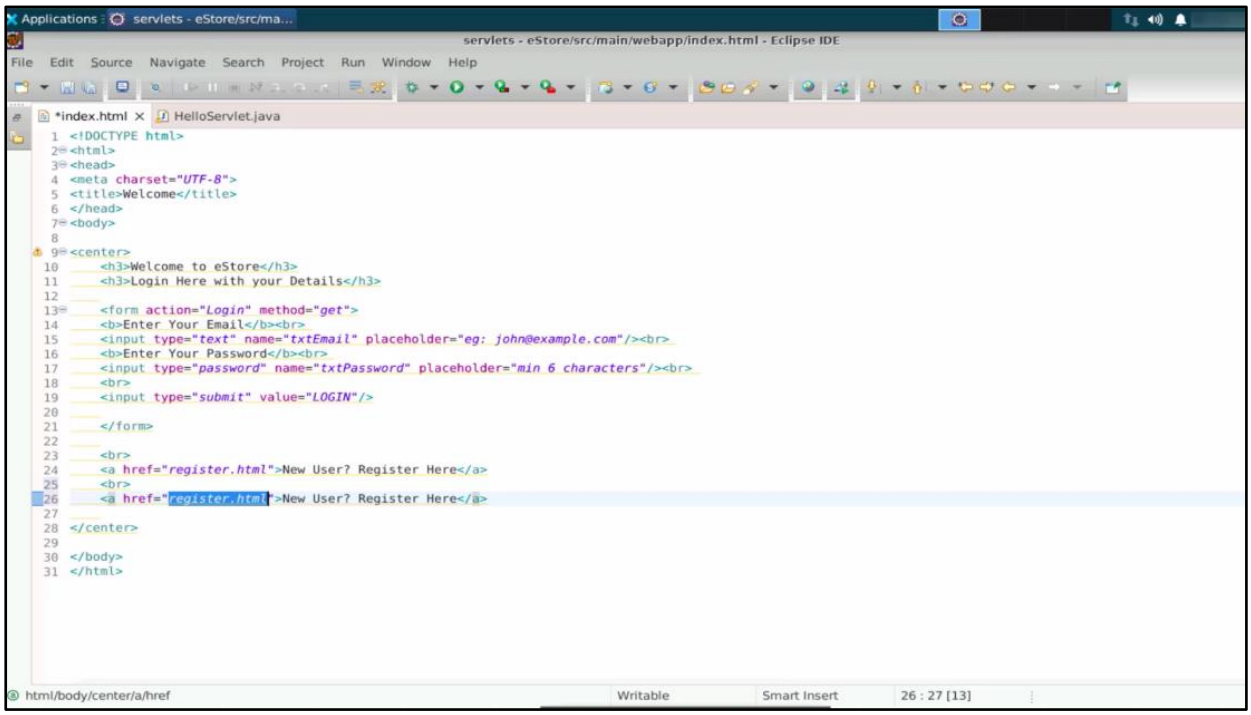


2.2 Open the **index.html** page

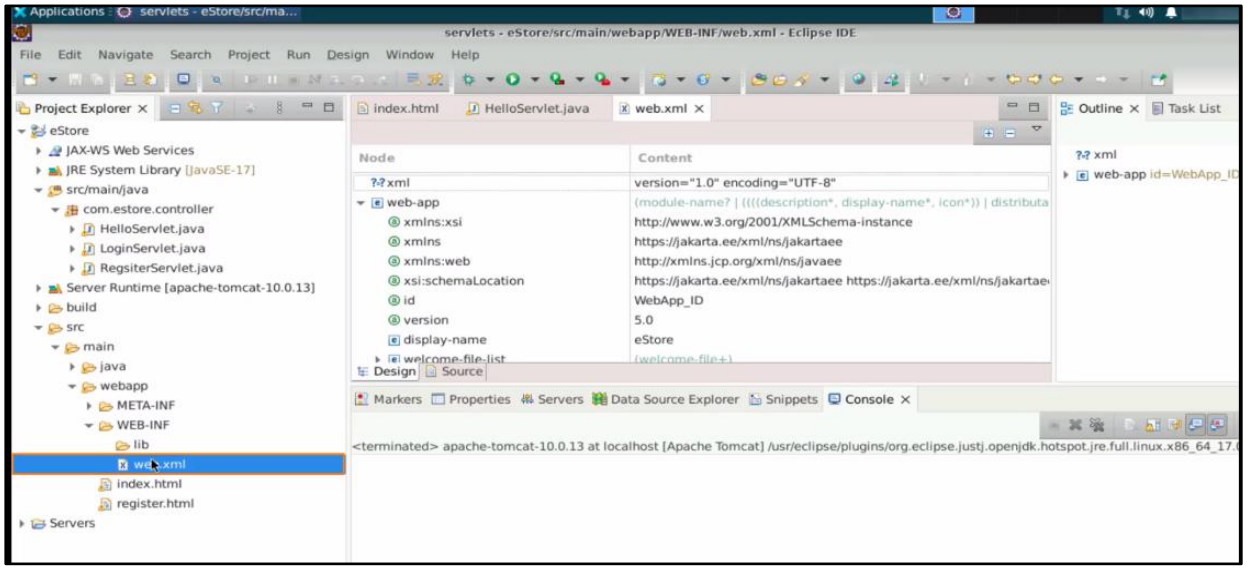


Step 3: Create a hyperlink for Hello Servlet

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



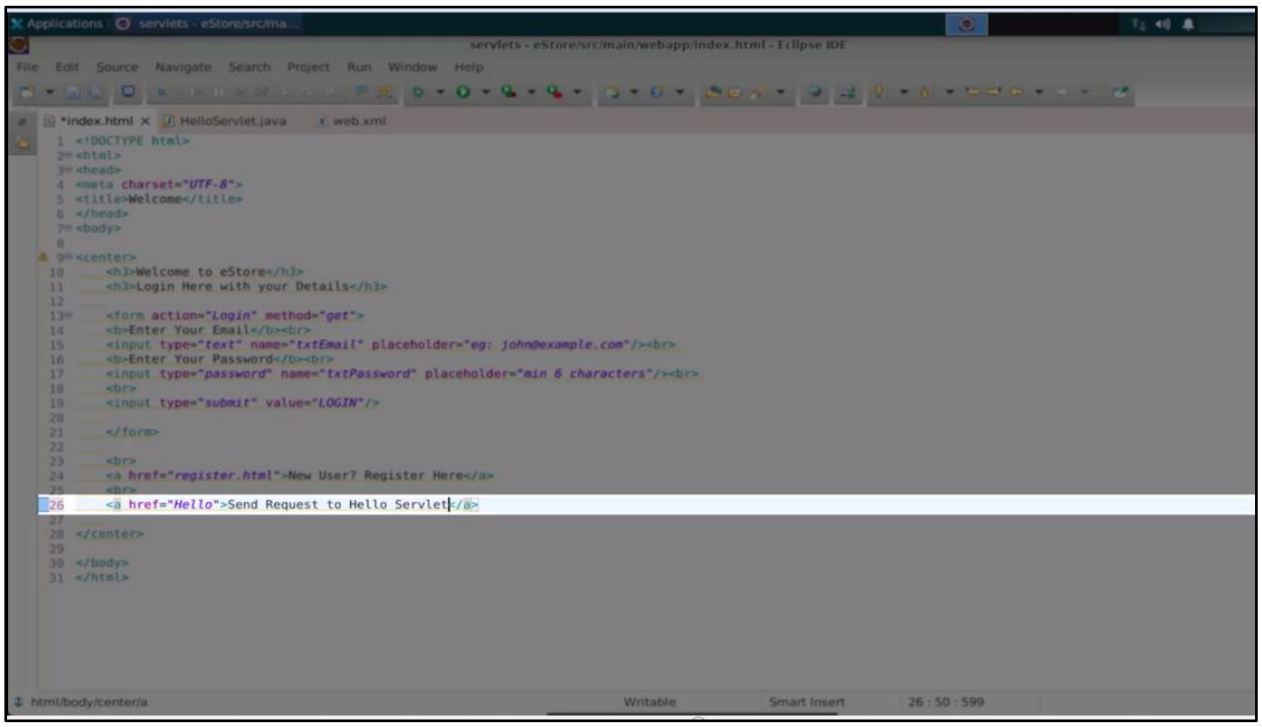
3.2 Open the **web.xml** file and copy the **<url-pattern>** for **HelloServlet**



index.html | HelloServlet.java | web.xml

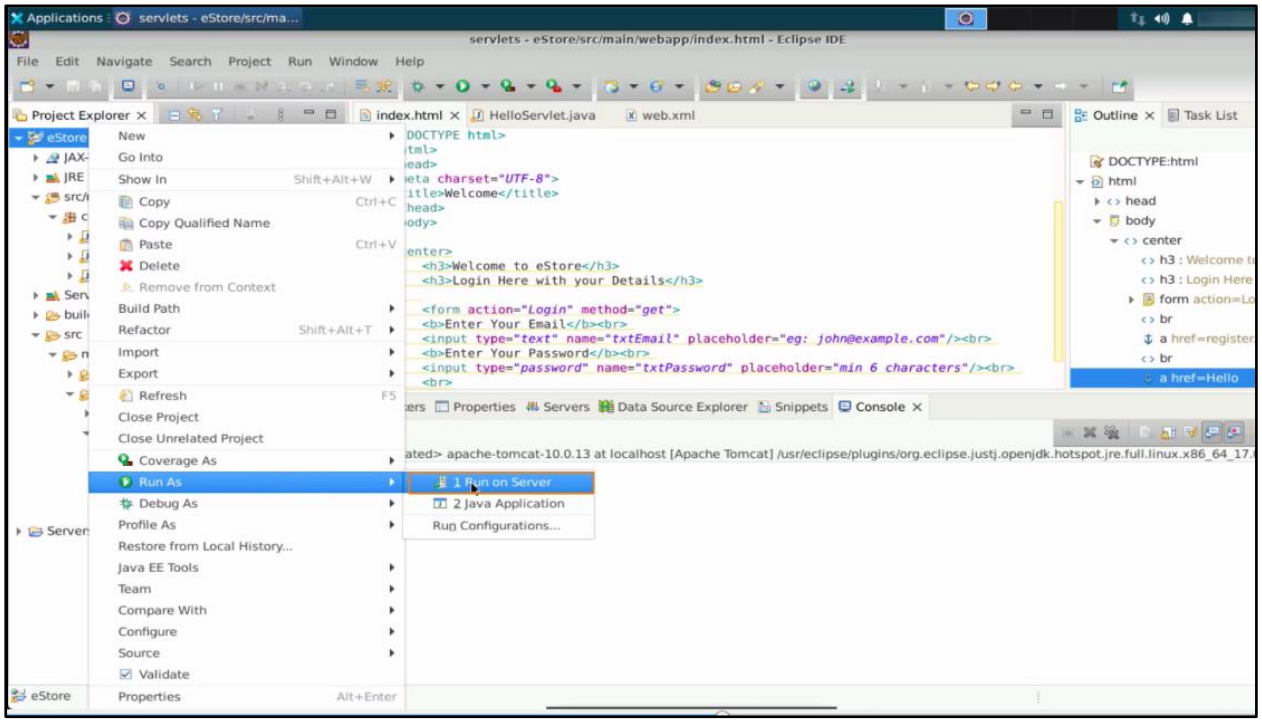
```
5     <welcome-file>index.html</welcome-file>
6   </welcome-file-list>
7   <servlet>
8     <description></description>
9     <display-name>HelloServlet</display-name>
10    <servlet-name>HelloServlet</servlet-name>
11    <servlet-class>com.estore.controller.HelloServlet</servlet-class>
12  </servlet>
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.RegsiterServlet</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

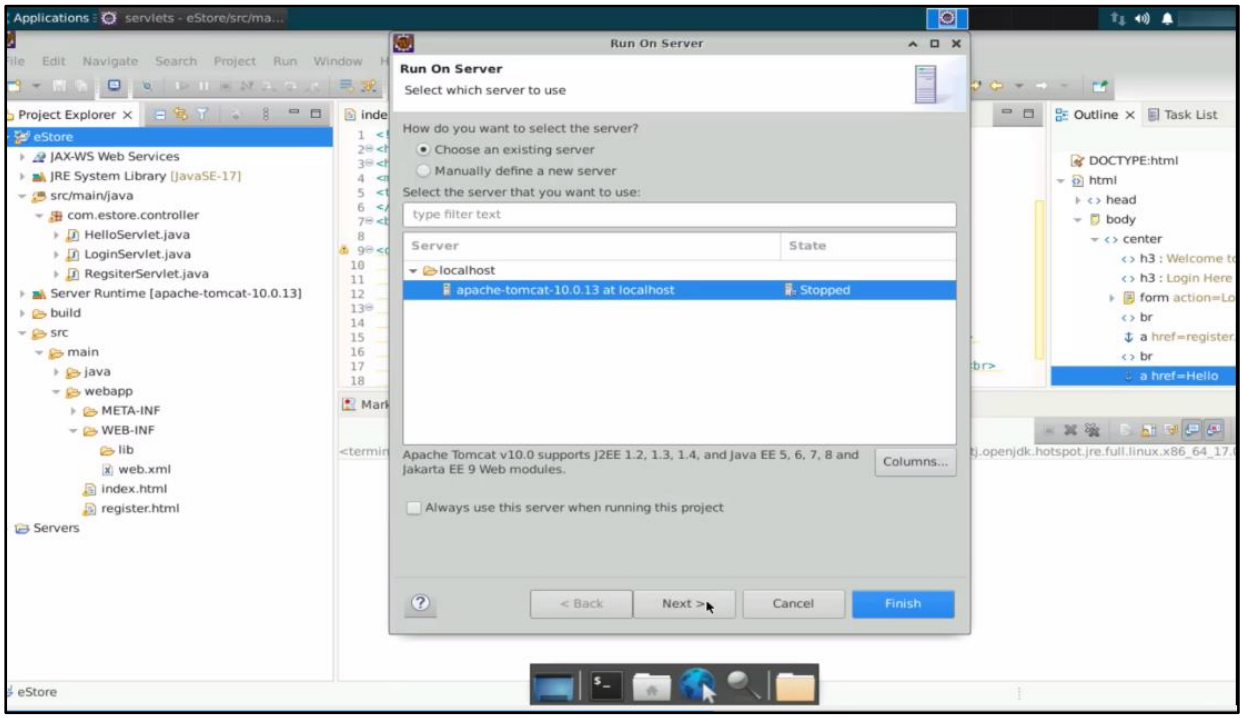


Step 4: Run code to check how the lifecycle method works

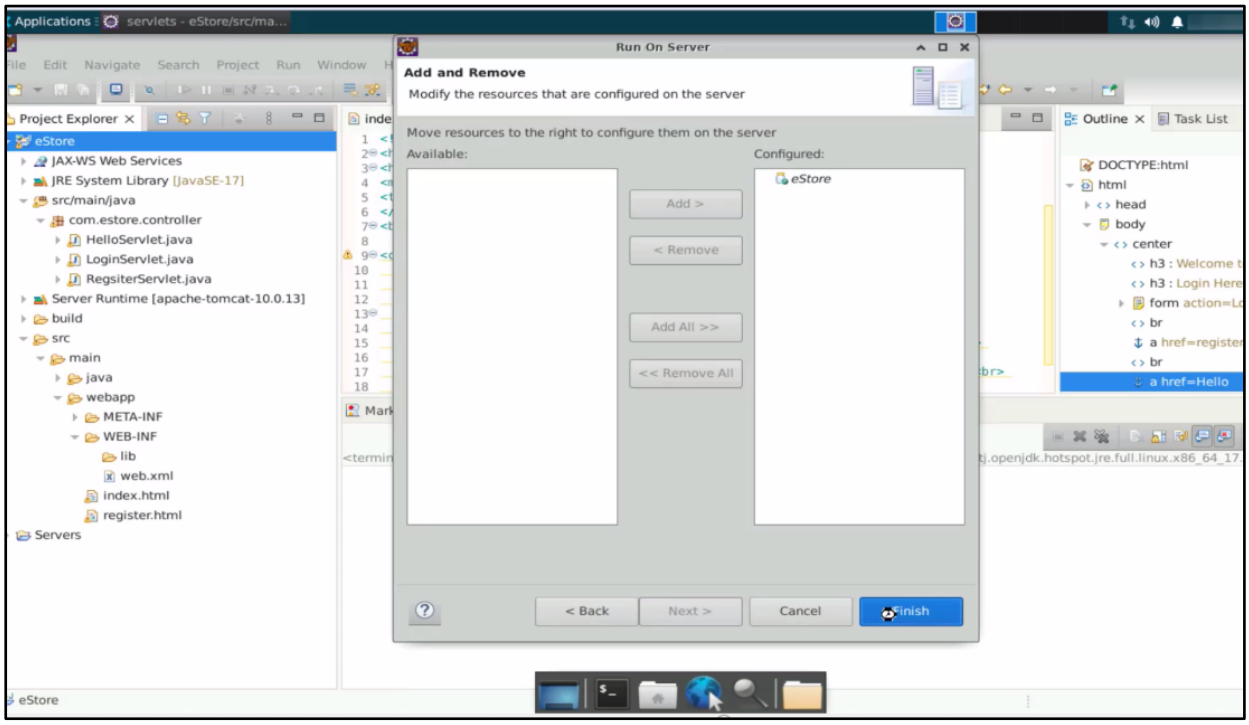
4.1 Save and run the code on the server. Right-click on the project, click on **Run As,** and select **Run on Server**



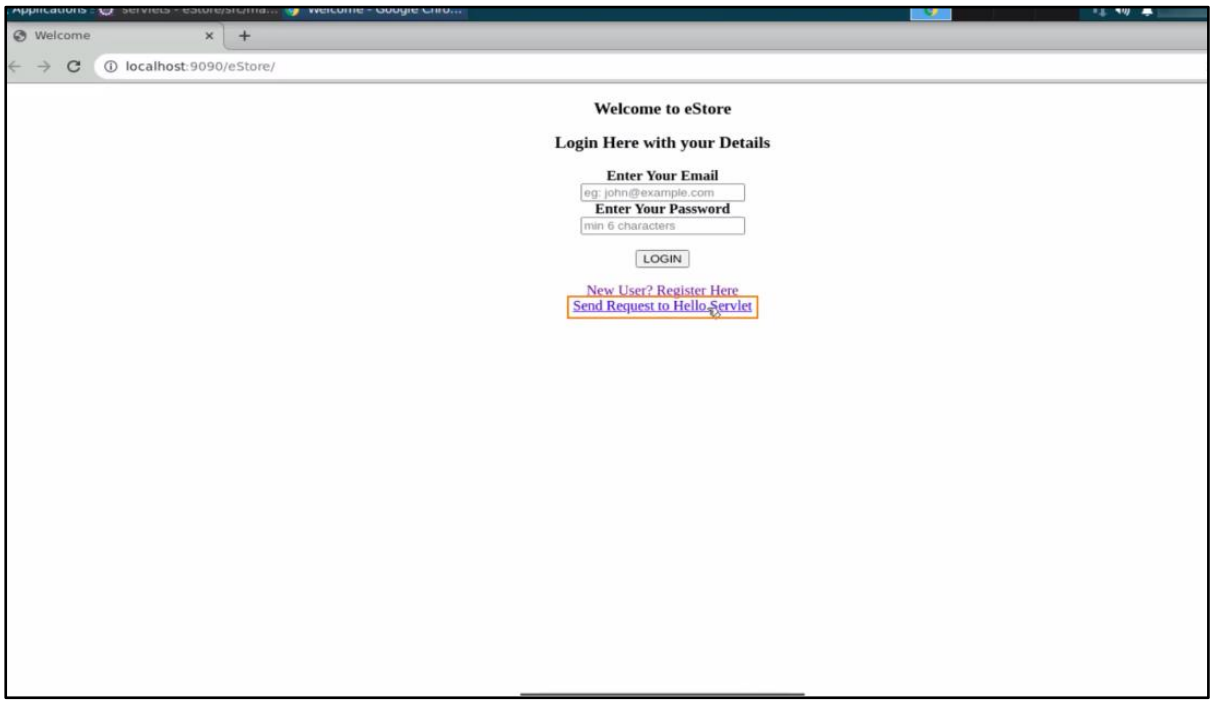
4.2 Choose the **apache-tomcat-10.0.13 at localhost** server and click **Next**



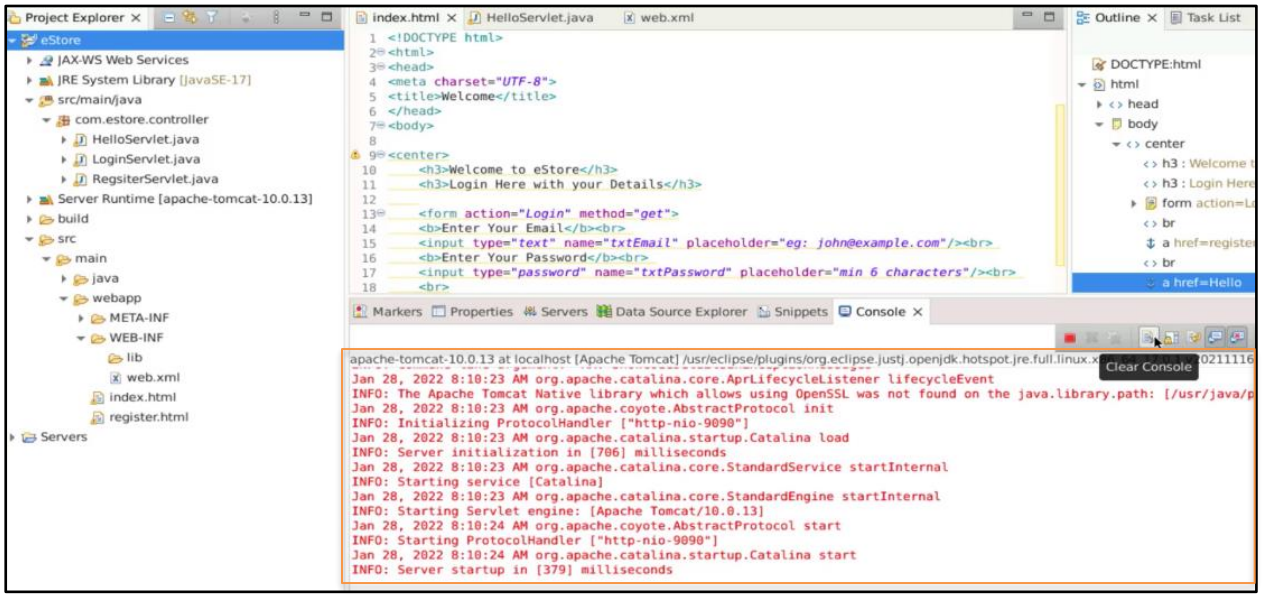
4.3 Click **Finish**



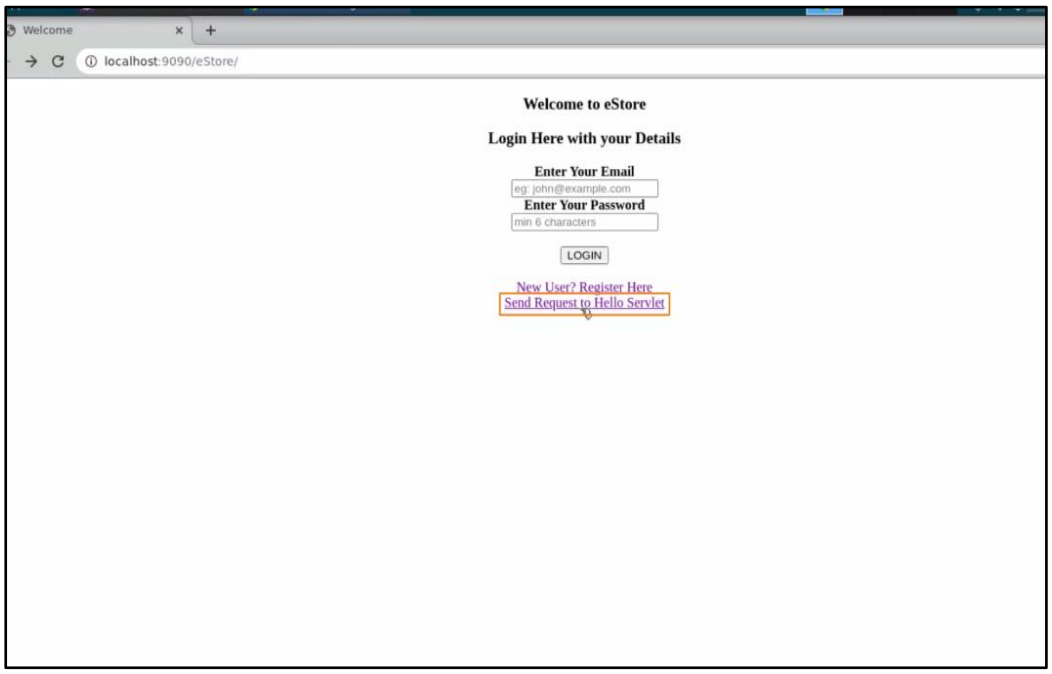
You can see the hyperlink as **Send Request to Hello Servlet**.



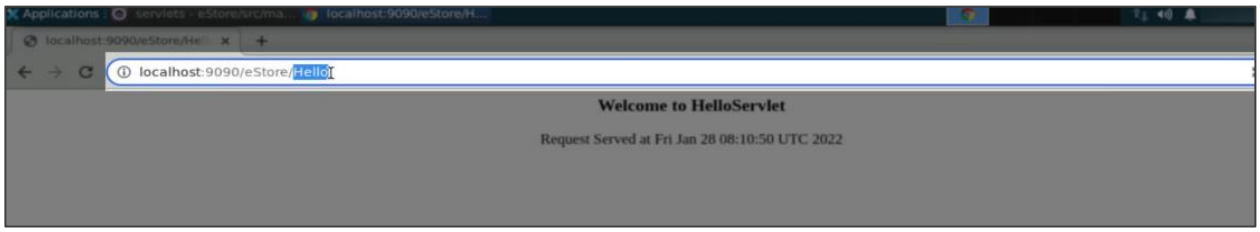
4.4 Go back to **Eclipse IDE** and check the **Console** tab. You will not be able to see any log or print statement as shown below:



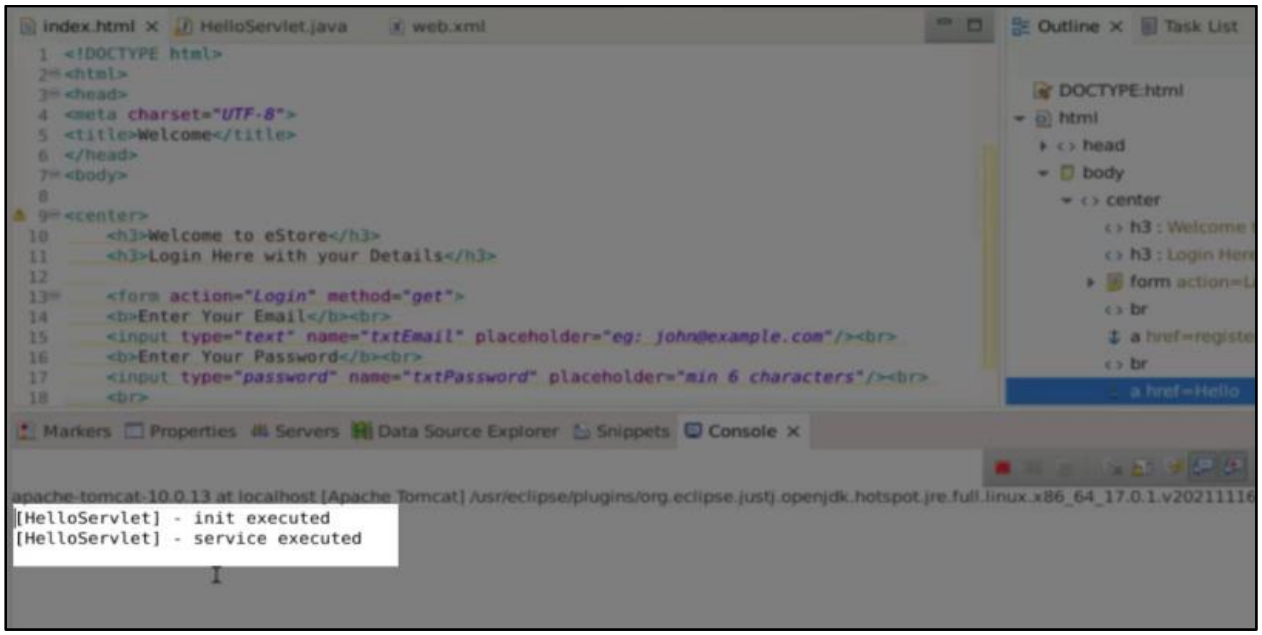
4.5 Go back and click on the **Send Request to Hello Servlet** link



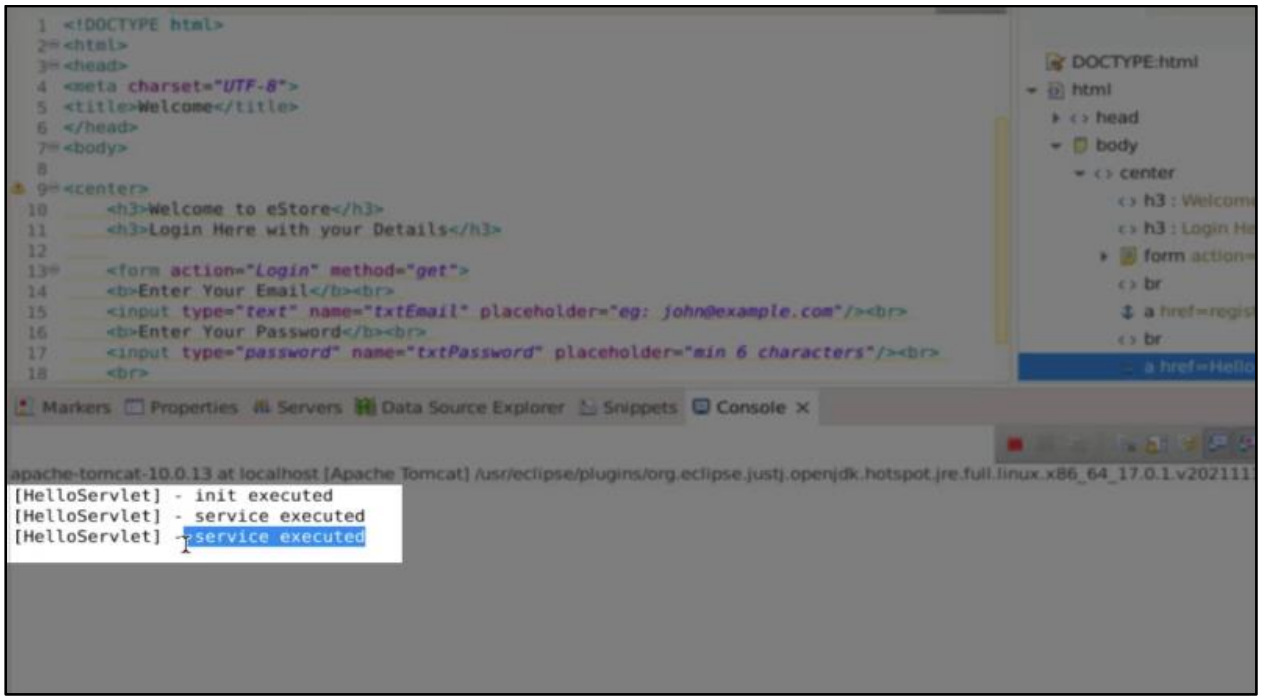
You can see that the **HelloRequest** has been forwarded to the **Hello Servlet**, and it prints the timestamp of when the request was served.



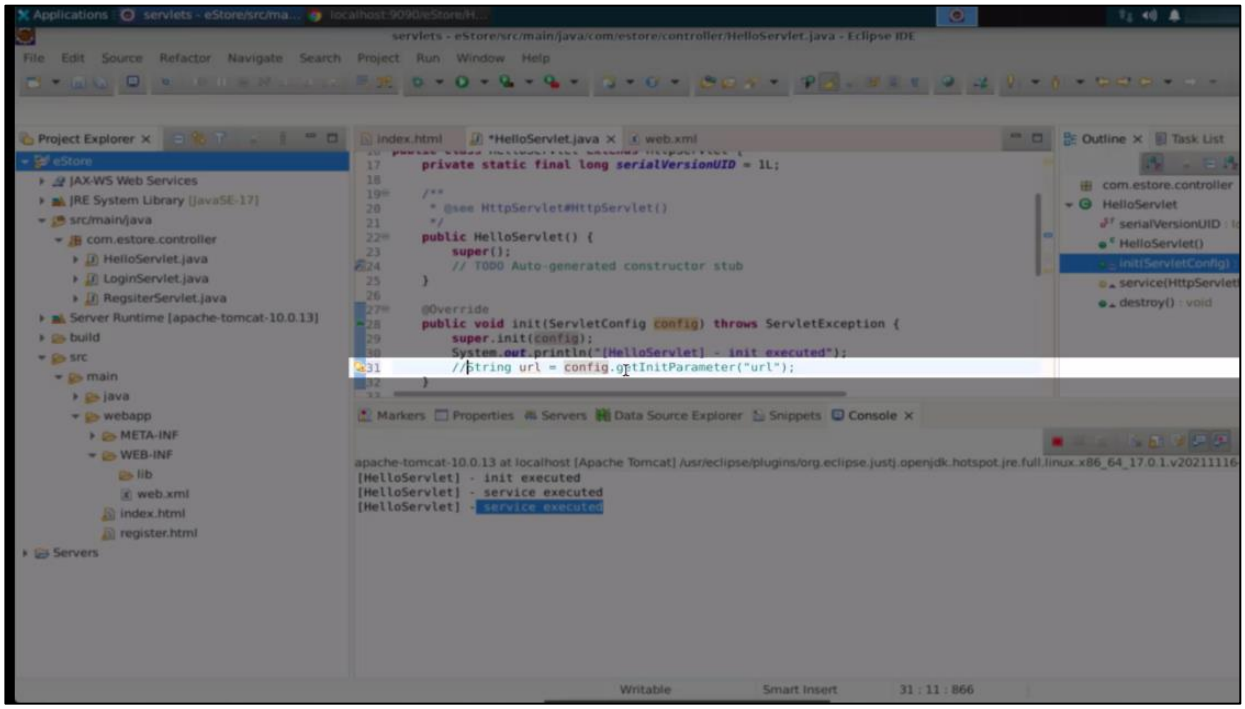
4.6 Go back to the Console tab. You can see that the two methods, init and service, are executed.



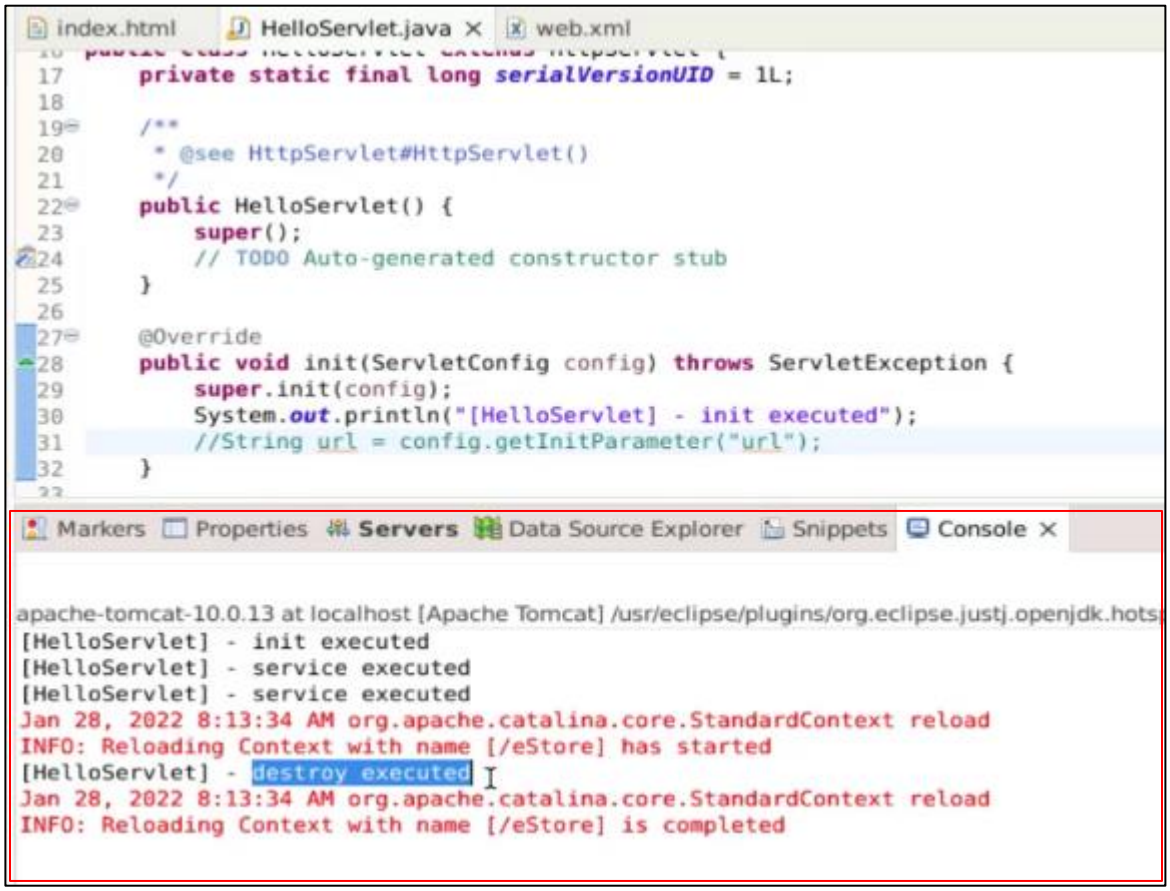
4.7 Reclick on the hyperlink in the browser. You will see one more service executed in the **Console** tab, as the init method gets executed only once.



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.**



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.