

1. Required Software:

- a. Must have the Java Development Kit (JDK)

<https://www.oracle.com/cis/java/technologies/downloads/#java17>

- b. To build JSP/Servlet applications, you need:

- Java Application Server – Tomcat

<https://tomcat.apache.org/download-10.cgi>

- Java Integrated Developer Environment (IDE) - **Eclipse IDE for Java EE Developers**

<https://www.eclipse.org/downloads/packages/release/2024-06/r>

2. Install Tomcat on Windows

a. Step 1: Download Tomcat for Windows

- Browser to the site and Locate the *Download* section and click the **latest Tomcat version** available.

<https://tomcat.apache.org/>

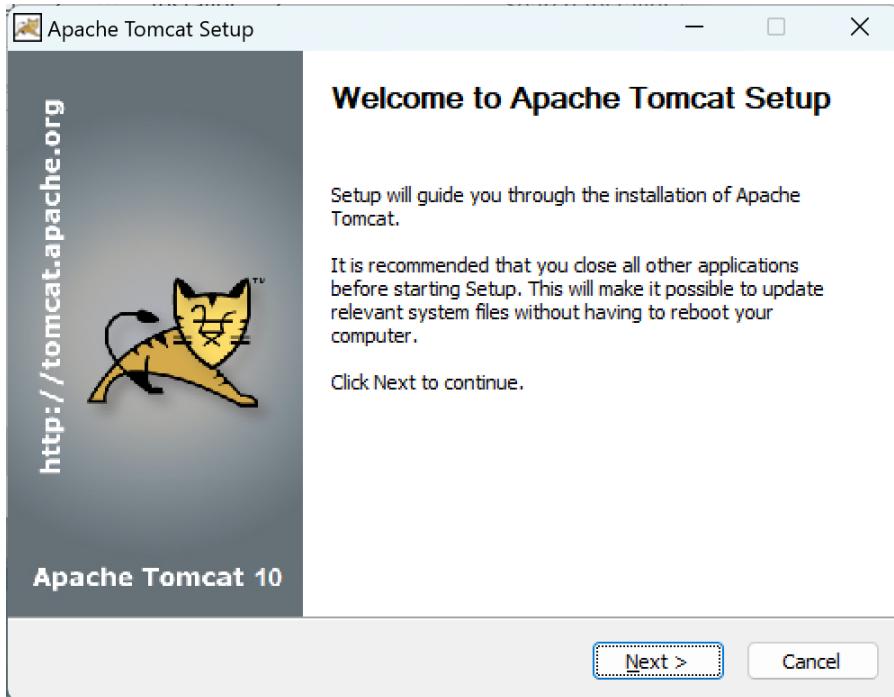
The screenshot shows the Apache Tomcat website at tomcat.apache.org. The page features the Apache feather logo and the text "Apache Tomcat®". A search bar and a "GO" button are visible. On the left, there's a sidebar with links for "Community", "Hangzhou China July 26-28 2024", "Apache Tomcat" (Home, Taglibs, Maven Plugin), "Download" (Tomcat 11 (beta), Tomcat 10, Tomcat 9, Tomcat 8, Tomcat Migration Tool for Jakarta EE, Tomcat Connectors, Tomcat Native, Taglibs, Archives), and "Documentation" (Tomcat 11.0 (beta), Tomcat 10.1, Tomcat 9.0). The main content area has a heading "Apache Tomcat" and text about the Jakarta EE platform. It highlights the release of "Tomcat 10.1.25 Released" with a link to the "Tomcat 10.1.25 Changelog". A "Download" button is located at the bottom right of this section.

- Click download link for the Windows Service Installer or the 32bit/64bit Windows zip file

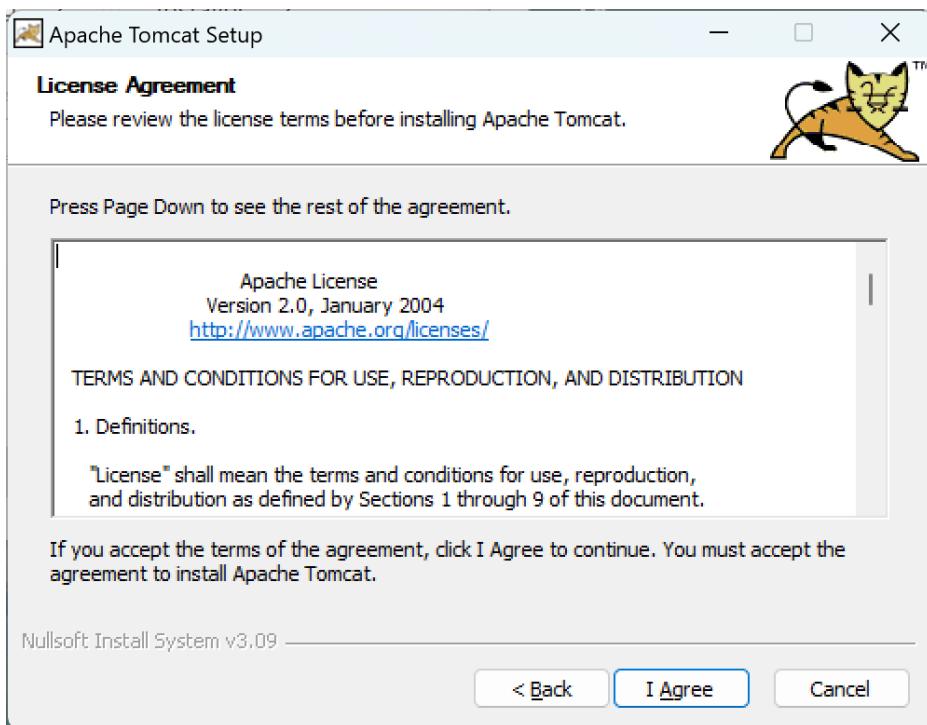
The screenshot shows the Apache Tomcat download page at tomcat.apache.org/download-10.cgi. The left sidebar lists versions from 8 to 11.0 (beta). The main content area is titled 'Mirrors' and shows the URL <https://dlcdn.apache.org/>. Below it, the '10.1.25' section provides packaging information and links to 'Binary Distributions'. A red box highlights the '32-bit/64-bit Windows Service Installer (pgp, sha512)' link.

b. Step 2: Install Tomcat

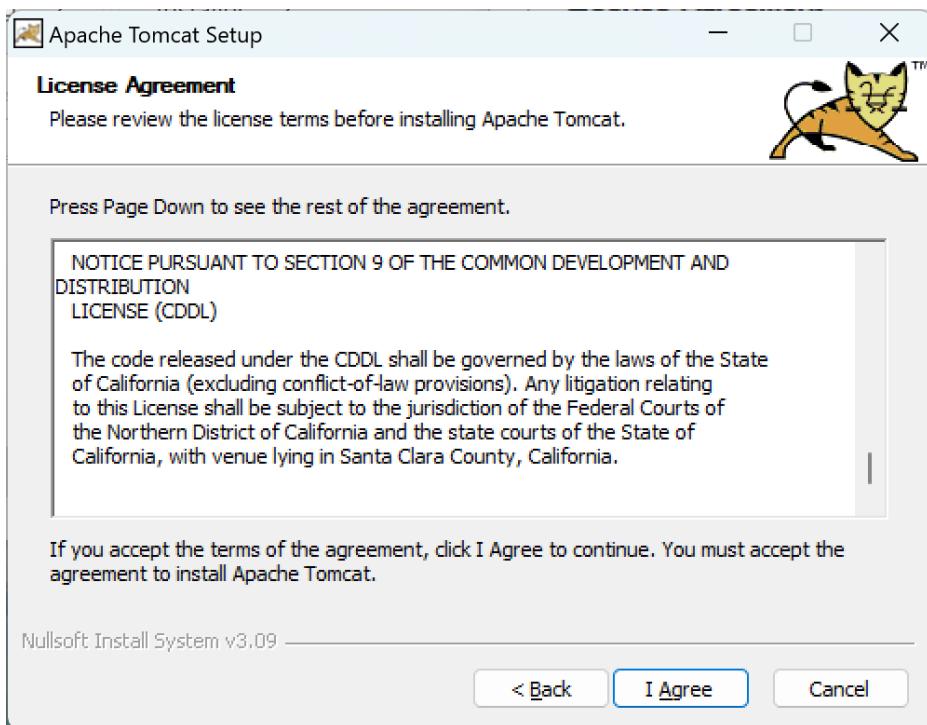
- Open the downloaded Windows Service Installer file to start the installation process.



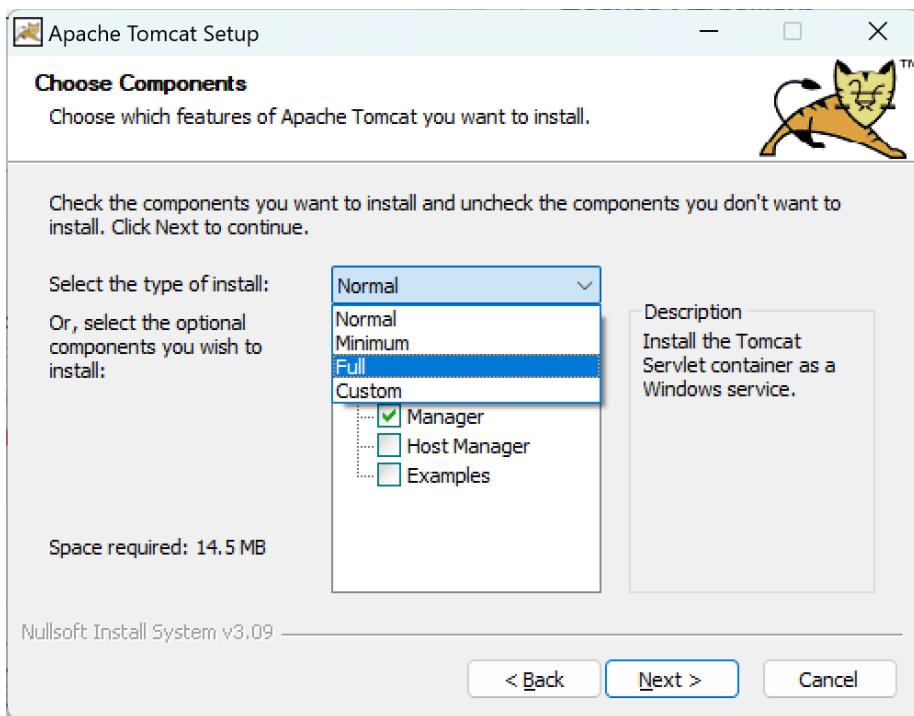
- In the Tomcat Setup welcome screen, click **Next** to proceed.



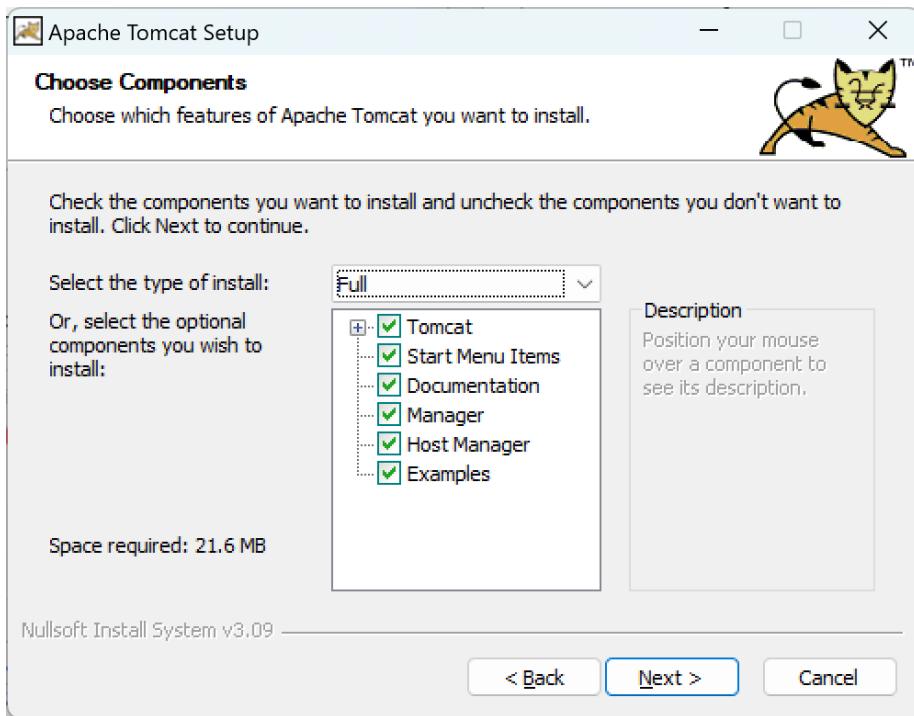
- Read the License Agreement and if you agree to the terms, click **I Agree** to proceed to the next step.



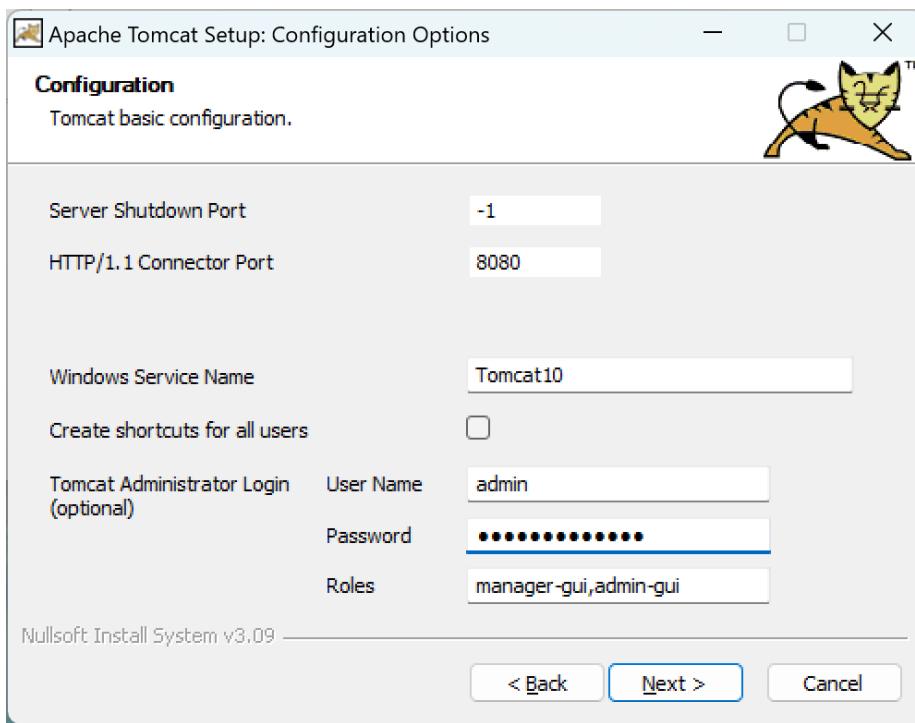
- Click **I Agree** button to continue



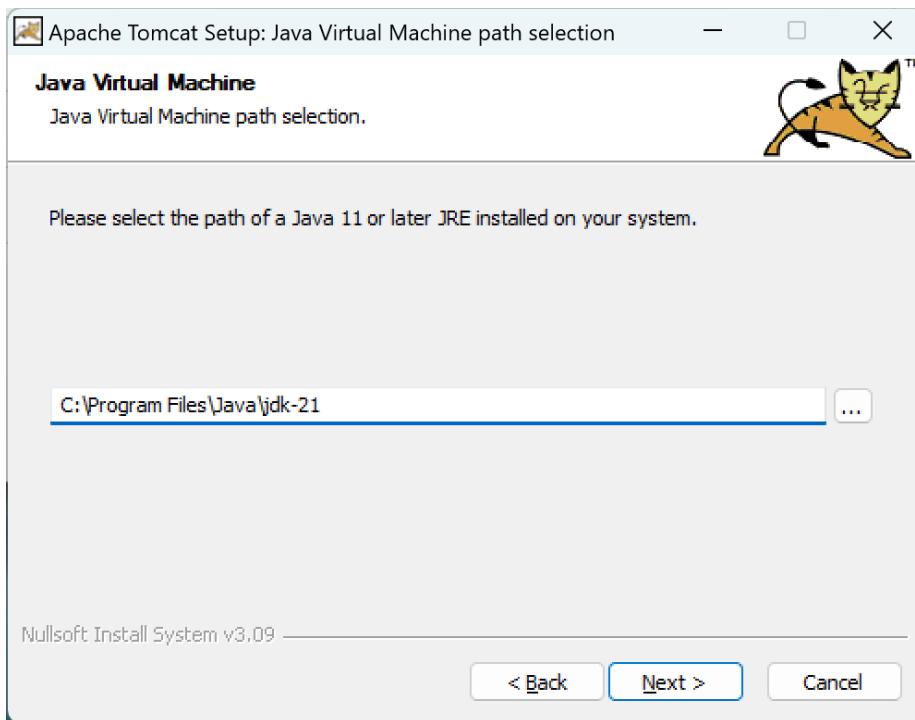
- Choose **Full** type of install



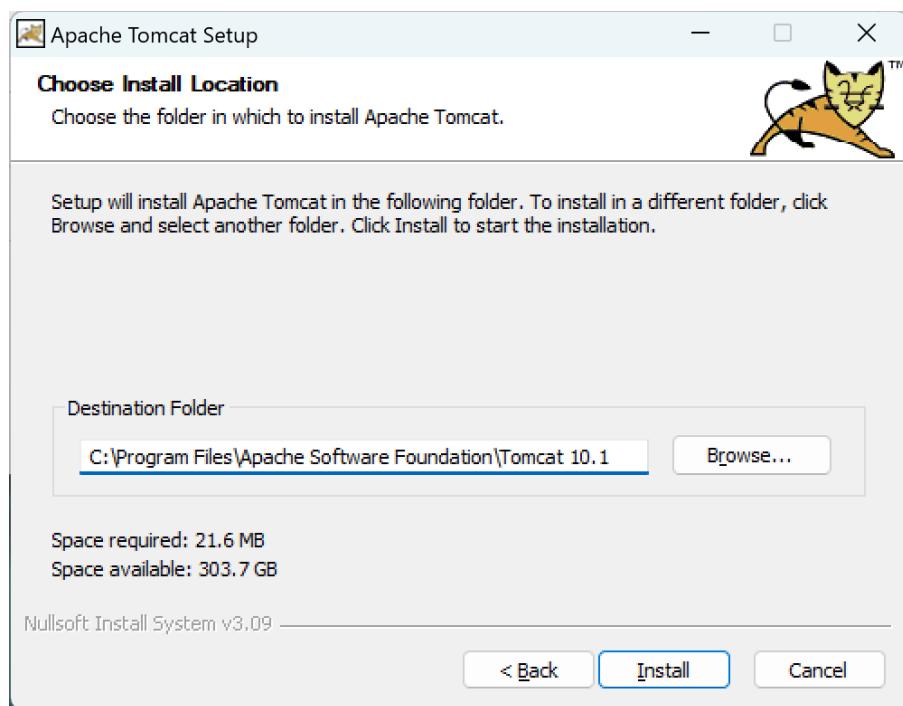
- Click **Next** button to continue. Fill user and password for Tomcat Administrator Login.



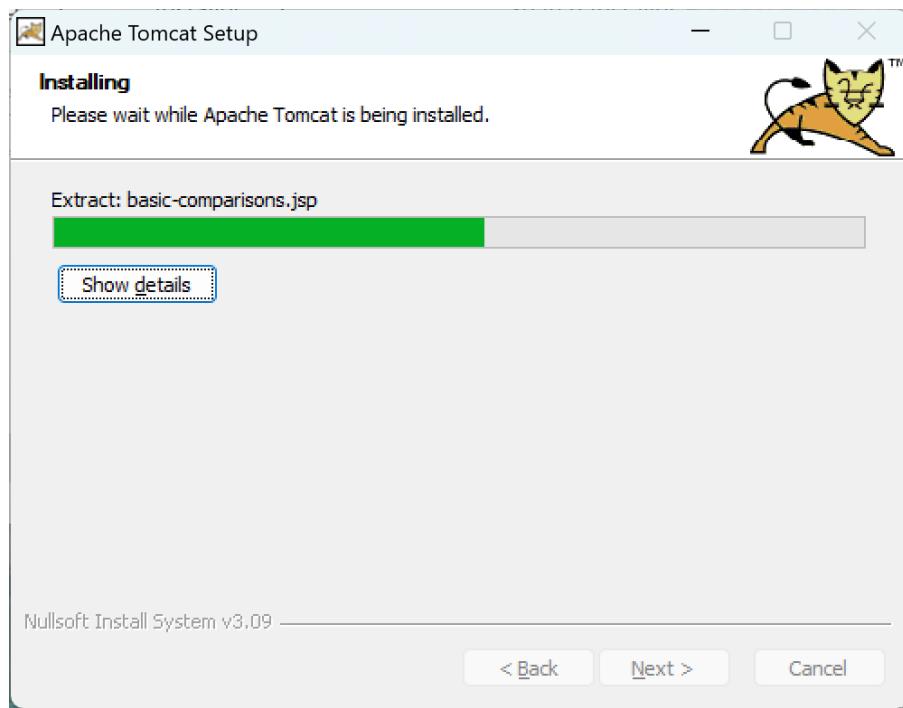
- Click Next button



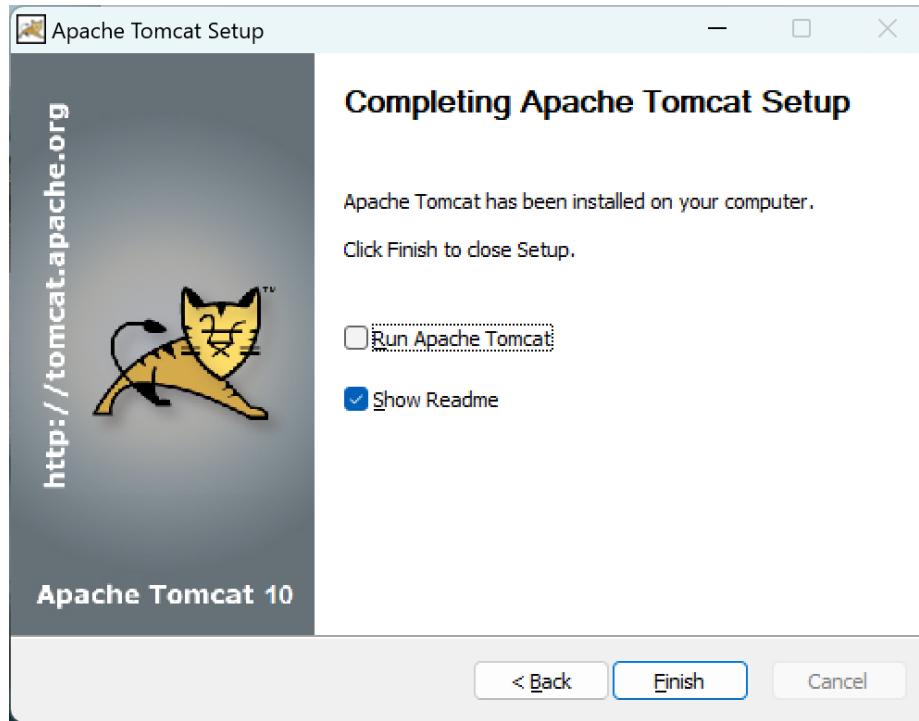
- Unless you have other installed version of Java that you want to use with Tomcat, just click Next button.



- Accept the suggested Install Location and click **Install** button



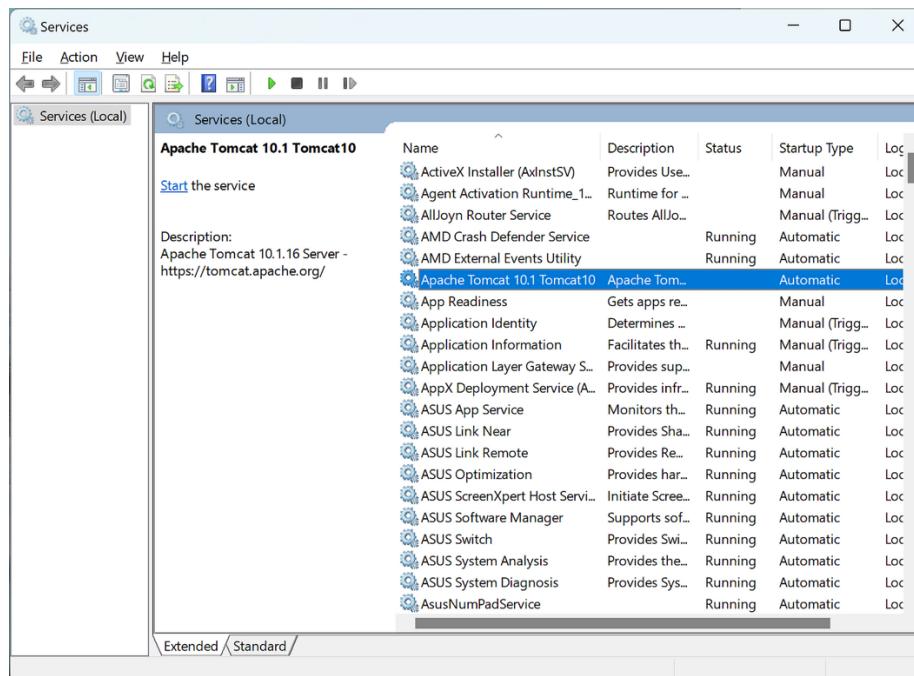
- Wait for the installation process.



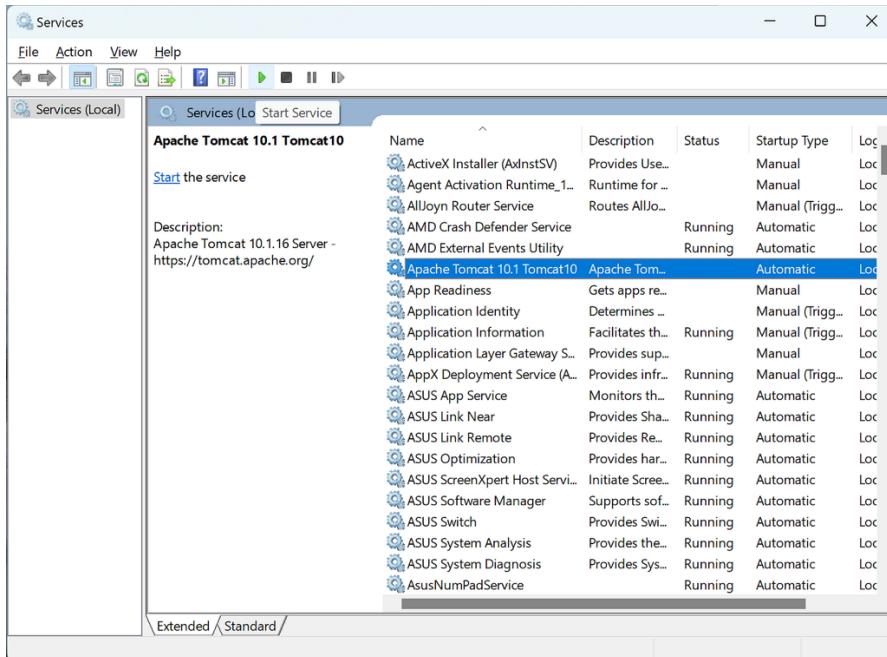
- Uncheck Run Apache Tomcat, then click **Finish** button to end installation.

c. Run Apache TomCat

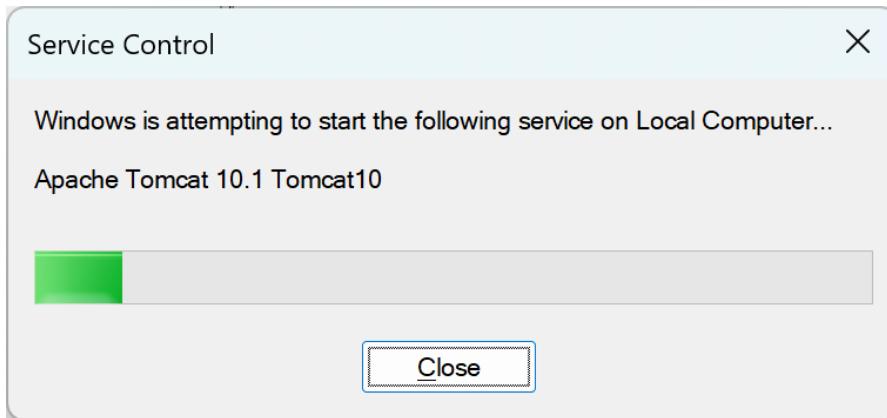
- Press Windows logo key +, type **services**, and press **Enter** key



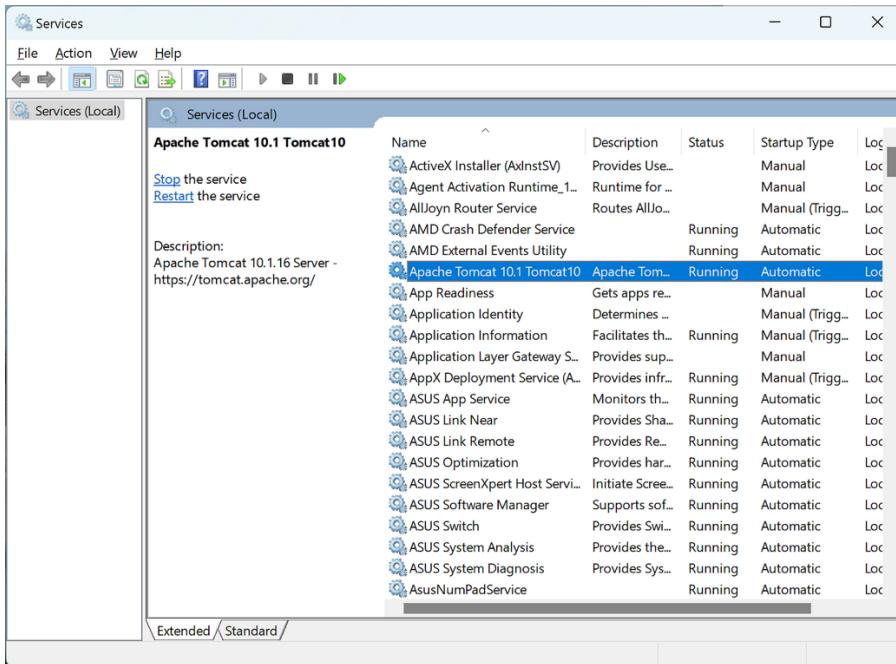
- Locate the **Apache Tomcat 10.1 Tomcat 10**



- Click green ► button to *Start Service*

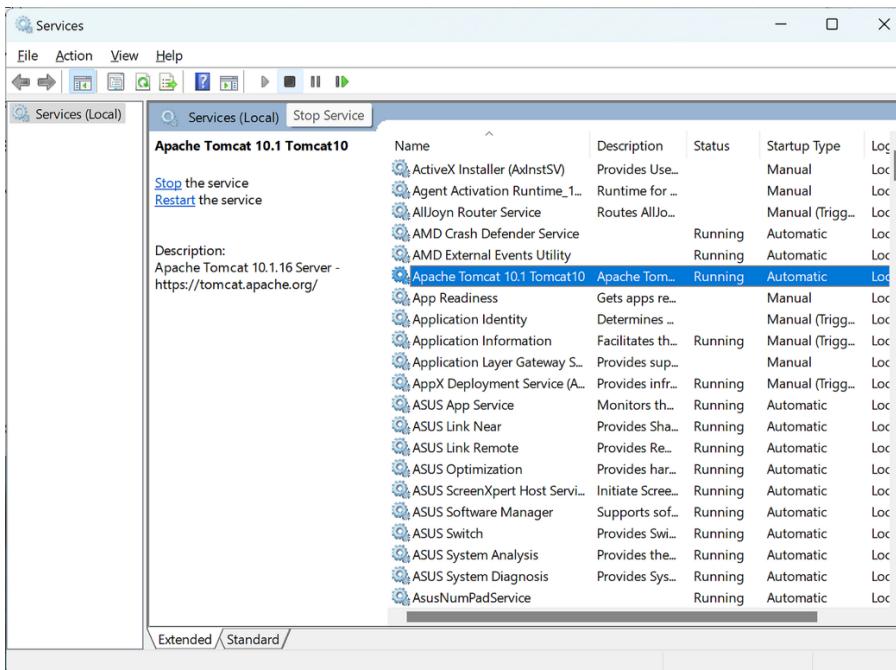


- Wait Windows to start the service

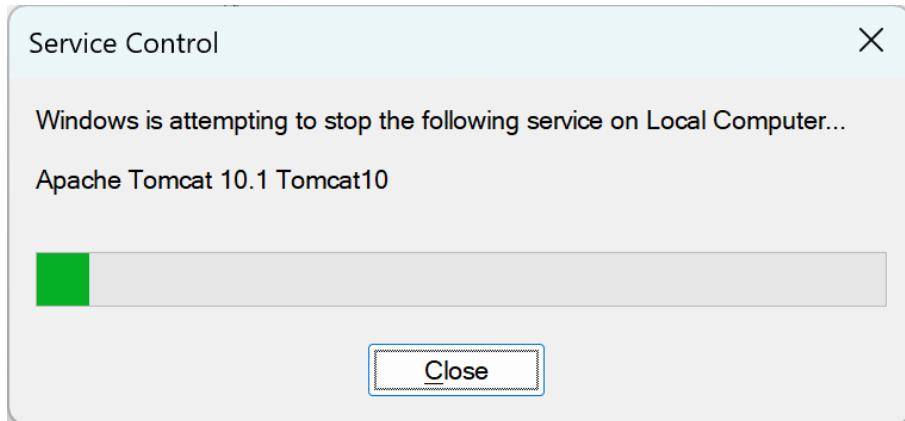


d. Stop Apache TomCat

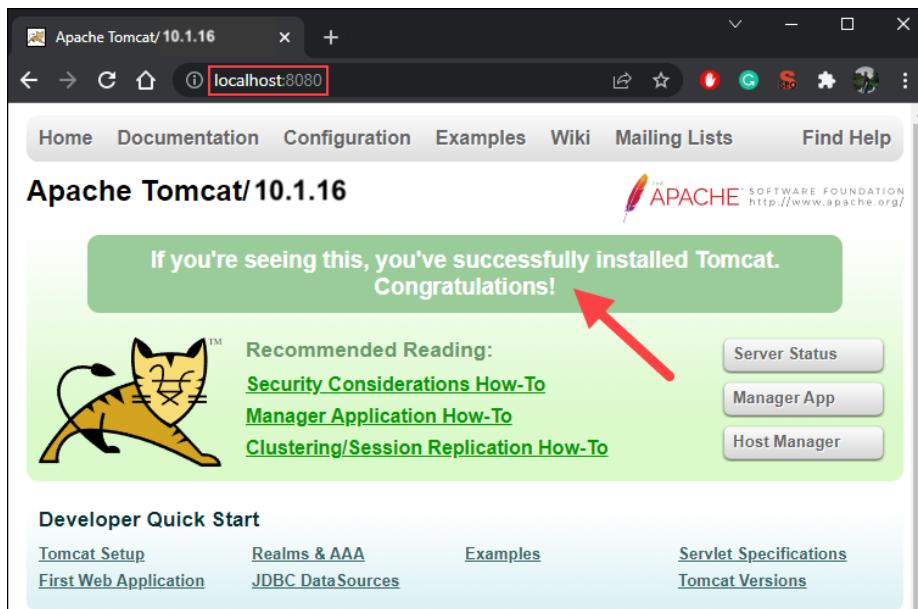
- Open Services Window and locate **Apache Tomcat 10.1 Tomcat 10**.



- Press button to stop the service.



- Access the server using a browser as an HTTP client. Browse to <http://localhost:8080> and access the Tomcat welcome page to ensure the server works.



CÁCH TẠO 1 ỨNG DỤNG WEB VÀ SỬ DỤNG APACHE TOMCAT SERVER

1. Kiến trúc thư mục của 1 ứng dụng Web (Web Application)

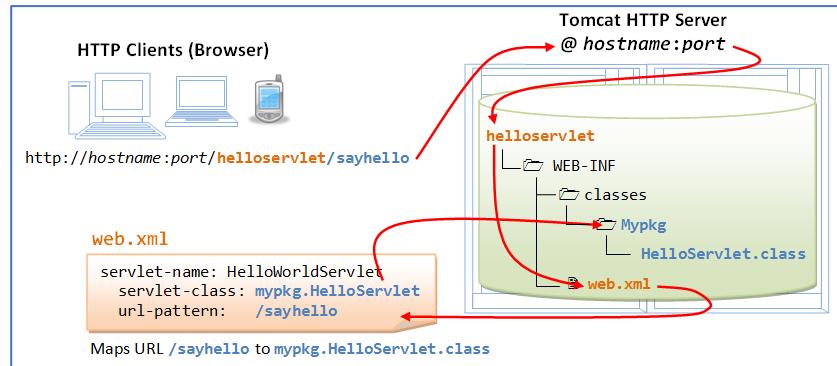
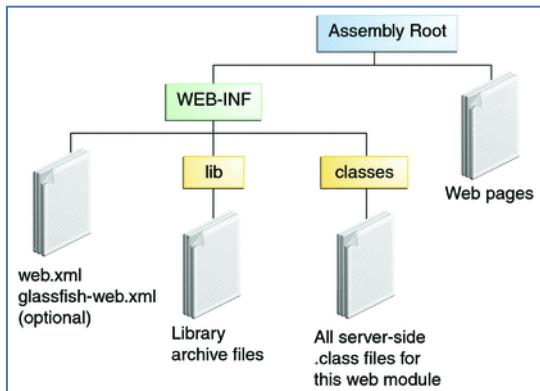
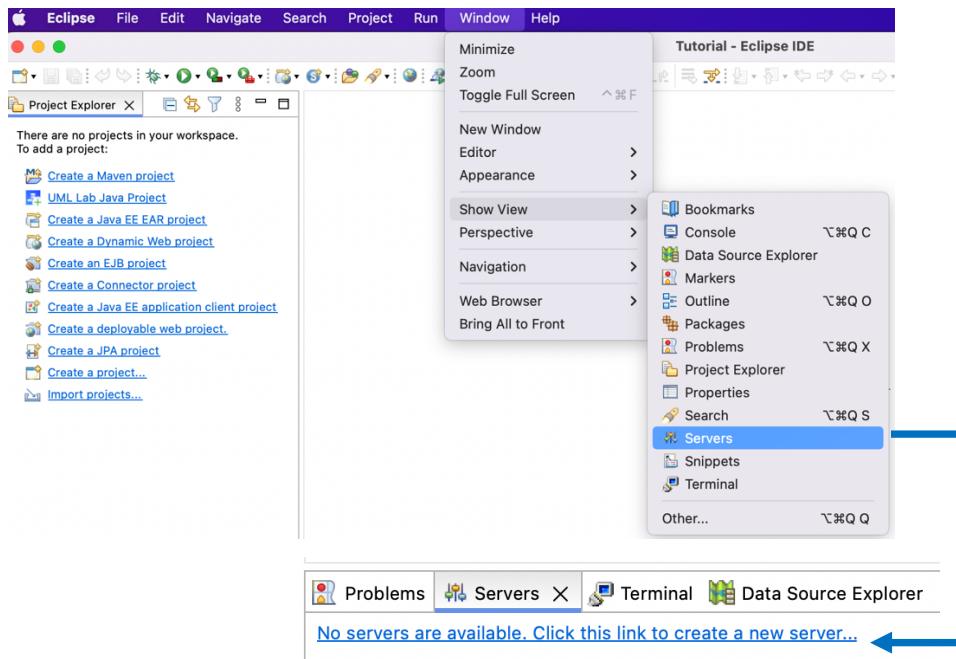


Fig1. Java web-application Structure

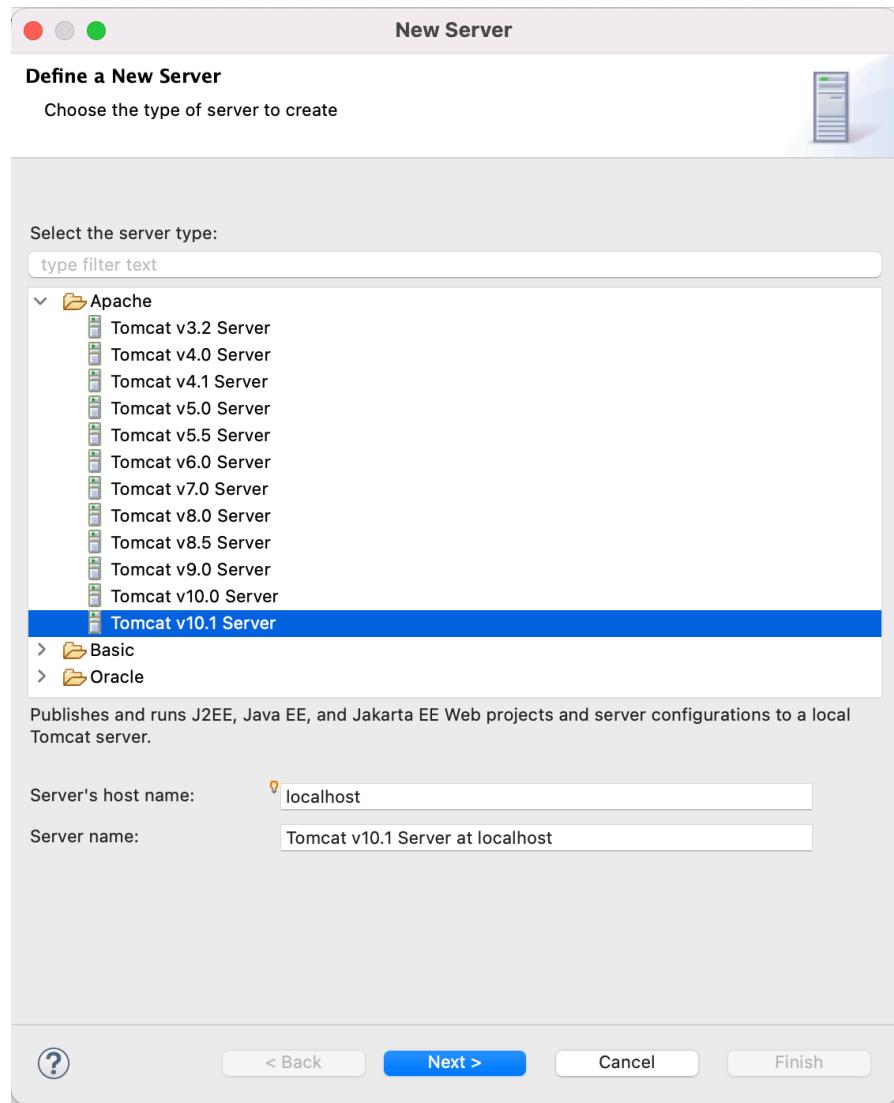
- Tạo cấu trúc thư mục
 - Đặt resources vào thư mục tương ứng
 - Copy chúng vào thư mục webapps của Tomcat Web server
- sẽ tự động deploy

2. Tạo kết nối giữa Eclipse và Apache Tomcat Server

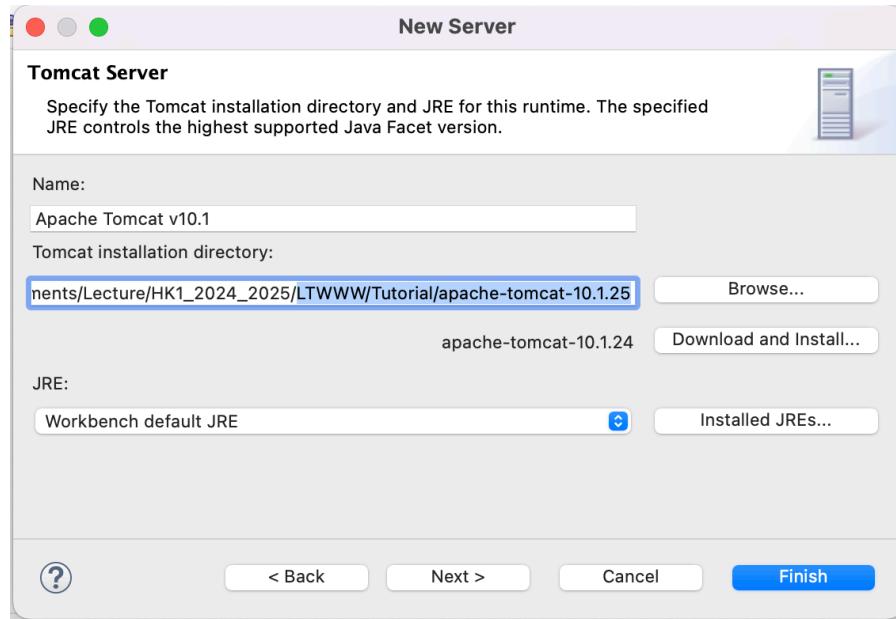
a. Step 1: Open Eclipse → Window → Show View → Servers



- b. **Step 2:** Click link “No servers are available. Click this link to create a new server ...” → Display popup

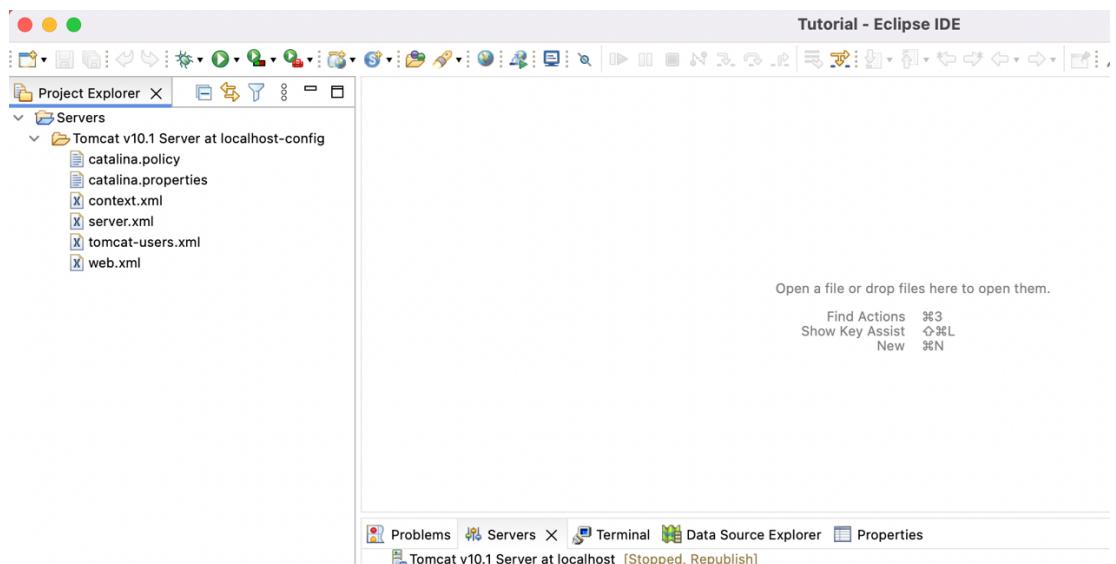


- c. **Step 3:** Select “Tomcat v10.1 Server” → Click “Next”



d. **Step 4:**

- i. Nếu Tomcat Server đã được download (hoặc đã được install) → Click “Browse...” → Chọn đường dẫn folder chứa Tomcat Server đã được download (hoặc install) → Click “Finish”
- ii. Nếu Tomcat Server chưa được download → Click “Download and Install ...” → Click “Finish”



e. **Step 5: Start Tomcat server**



3. Triển khai Servlet. Có 2 cách triển khai:

a. Cách 1: thực hiện trong file WEB-INF/web.xml

```
<servlet>
    <servlet-name>UserServlet(Tên Servlet)</servlet-name>
    <servlet-name>UserServlet(Tên class của Servlet)</servlet-name>
    <init-param>
        <param-name>name(Tên parameter)</param-name>
        <param-value>Not provided(Value của parameter)</param-value>
    </init-param>
    <init-param>
        <param-name>email</param-name>
        <param-value>Not provided</param-value>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>UserServlet(Tên Servlet)</servlet-name>
    <url-pattern>/userServlet(url - route)</url-pattern >
</servlet-mapping>
```

b. Cách 2: sử dụng Annotation

```
@WebServlet(
    name = "UserServlet",
    urlPatterns = "/userServlet",
    initParams={
        @WebInitParam(name="name", value="Not provided"),
        @WebInitParam(name="email", value="Not provided")
    }
)
public class UserServlet extends HttpServlet {
    @Override
    protected void doGet( HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
        // TODO
    }
}
```

```

    }
}

```

Name	Type	Required	Description
value or urlPatterns	<i>String[]</i>	Required	Specify one or more URL patterns of the servlet. Either of attribute can be used, but not both.
name	<i>String</i>	Optional	Name of the servlet
displayName	<i>String</i>	Optional	Display name of the servlet
description	<i>String</i>	Optional	Description of the servlet
asyncSupported	<i>boolean</i>	Optional	Specify whether the servlet supports asynchronous operation mode. Default is false.
initParams	<i>WebInitParam[]</i>	Optional	Specify one or more initialization parameters of the servlet. Each parameter is specified by @WebInitParam annotation type.
loadOnStartup	<i>int</i>	Optional	Specify load-on-startup order of the servlet. Default: -1 if the integer is negative: The container loads servlet at any time. if the integer is 0 or positive: The servlet marked with lower integers are loaded before servlets marked with higher integers.
smallIcon	<i>String</i>	Optional	Specify name of the small icon of the servlet.
largeIcon	<i>String</i>	Optional	Specify name of the large icon of the servlet.

4. Khởi tạo parameter (Init Parameters):

- a. Sử dụng trong Servlet: sử dụng init-param (hoặc sử dụng annotation initParams) và khai báo bên trong Servlet

```

< servlet >
    < servlet-name >UserServlet(Tên Servlet) </ servlet-name >
    < servlet-name >UserServlet(Tên class của Servlet) </ servlet-name >
    < init-param >
        < param-name >name(Tên parameter) </ param-name >
        < param-value >Not provided(Value của parameter) </ param-value >

```

```

</init-param>
<init-param>
    <param-name>email</param-name>
    <param-value>Not provided</param-value>
</init-param>
</servlet>

@WebServlet(
    name = "UserServlet",
    urlPatterns = "/userServlet",
    initParams={
        @WebInitParam(name="name", value="Not provided"),
        @WebInitParam(name="email", value="Not provided")
    }
)

```

- b. Sử dụng trong toàn cục (global): khi cần sử dụng các biến bất biến (immutable data) và truy xuất toàn cục thì chúng ta nên sử dụng application-wide context để khởi tạo parameters để lưu trữ dữ liệu (web.xml).

```

<?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"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd"
id="WebApp_ID" version="6.0">
    <context-param>
        <param-name>province</param-name>
        <param-value>Mendoza</param-value>
    </context-param>

    <context-param>
        <param-name>country</param-name>
        <param-value>Argentina</param-value>
    </context-param>
</web-app>

```

Cách sử dụng trong Servlet:

```
@WebServlet(
```

```

name = "UserServlet",
urlPatterns = "/userServlet",
initParams={

    @WebInitParam(name="name", value="Not provided"),
    @WebInitParam(name="email", value="Not provided")
}

public class UserServlet extends HttpServlet {

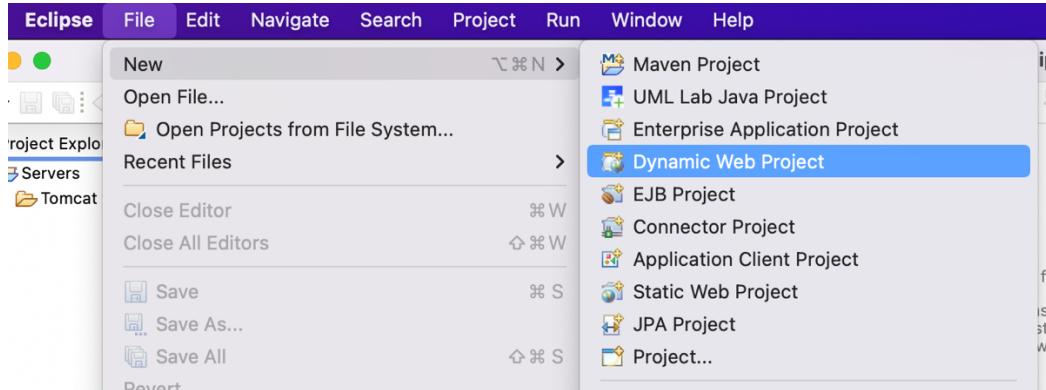
    @Override
    protected void doGet( HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
        // TODO
    }

    protected String getRequestParameter( HttpServletRequest request,
String name) {
        String param = request.getParameter(name);
        return !param.isEmpty() ? param : getInitParameter(name);
    }
}

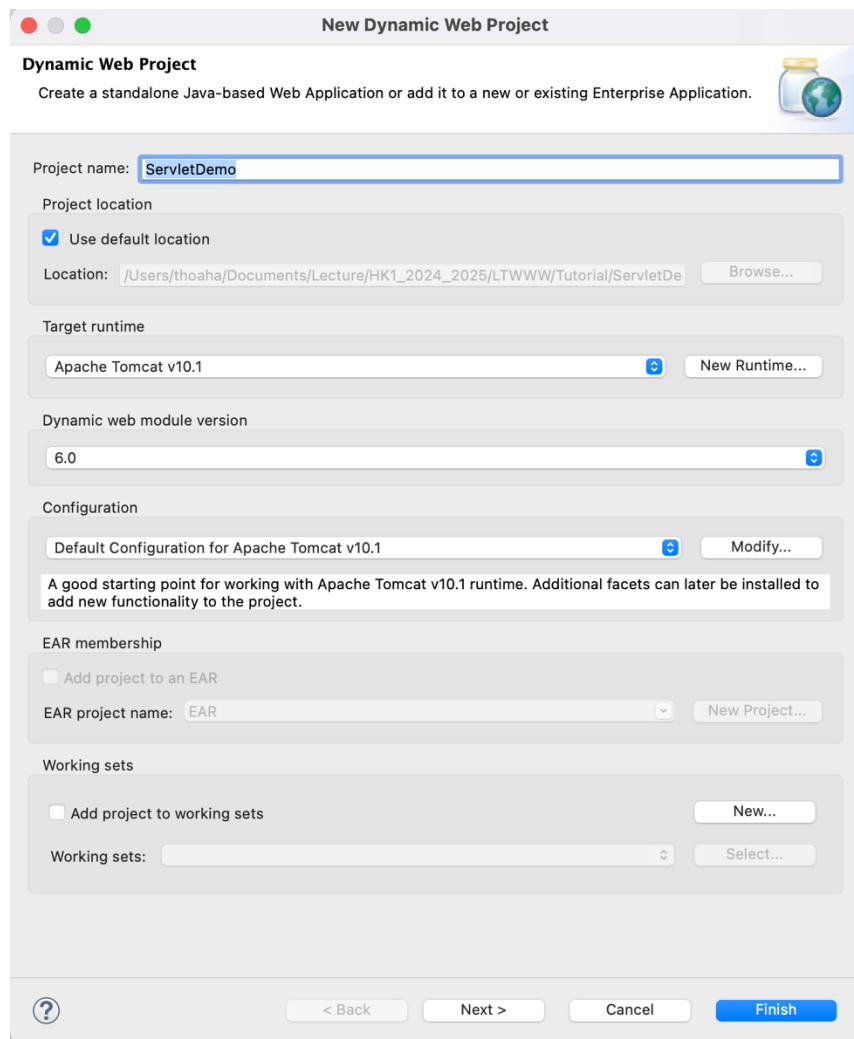
```

5. Tạo ứng dụng Servlet với Eclipse IDE

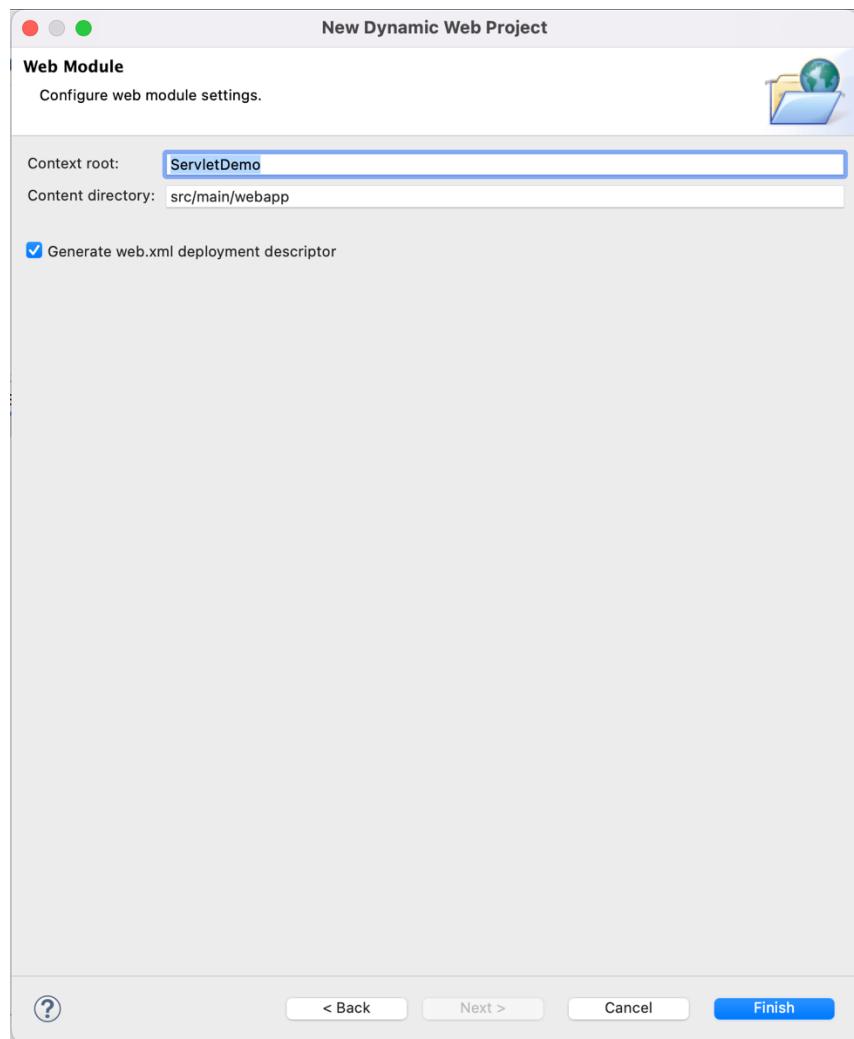
a. Step 1: File → New → Dynamic Web Project



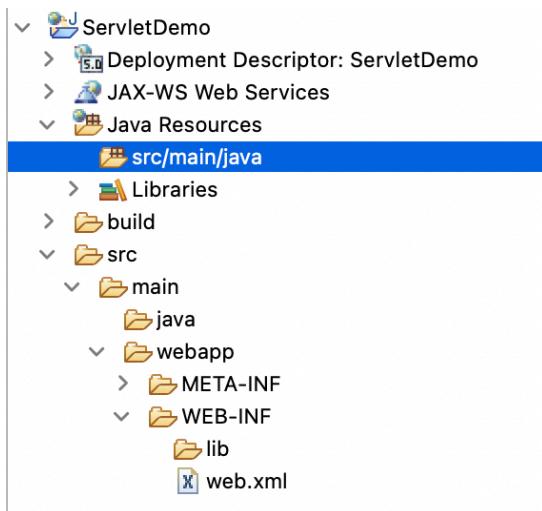
b. Step 2: Điền tên Project và click Next



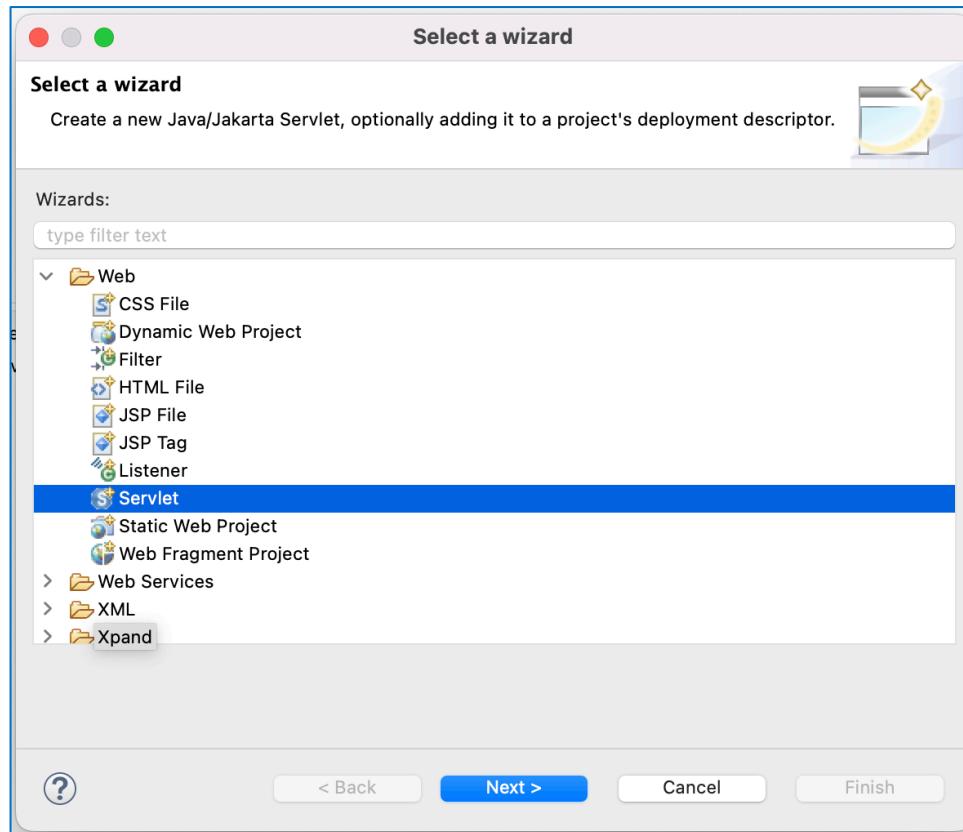
c. **Step 3:** Check Generate web.xml deployment descriptor và click Finish



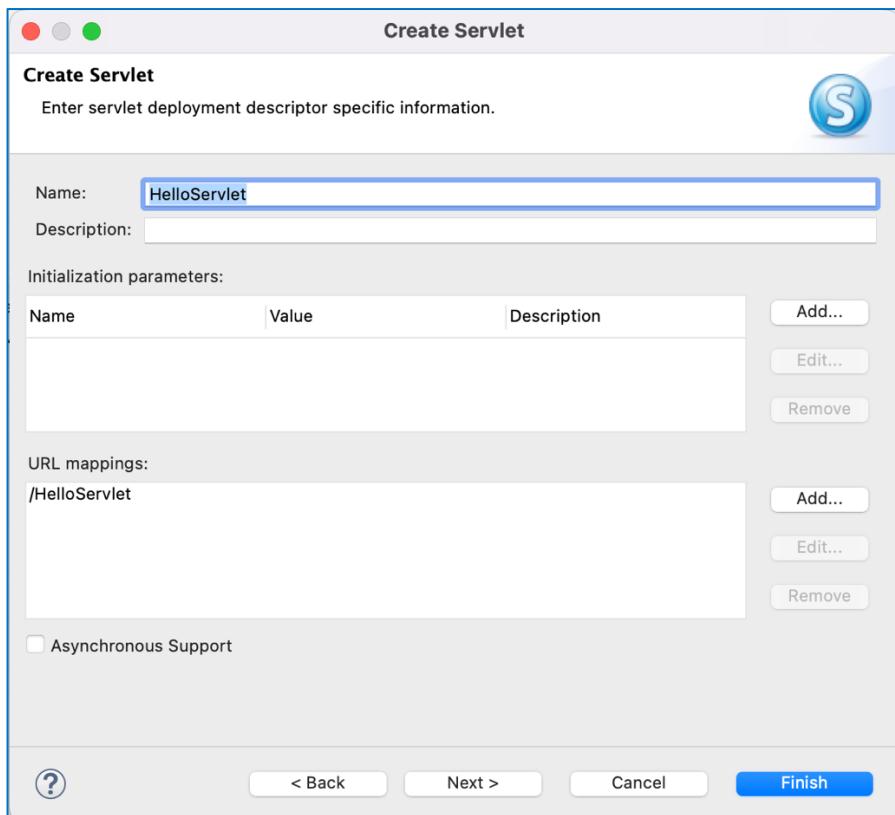
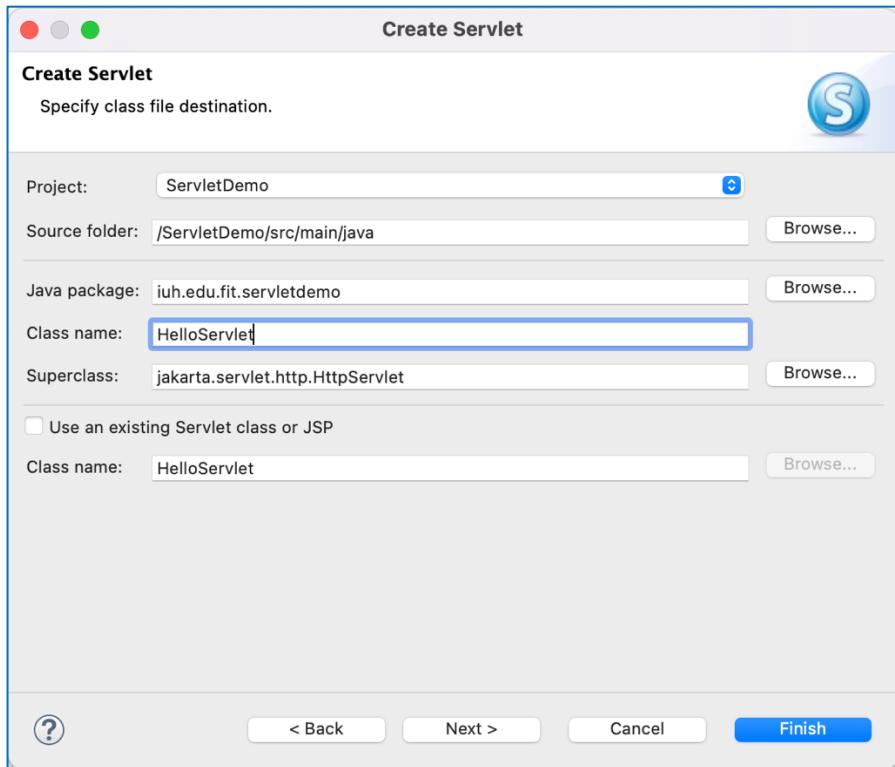
d. *Step 4:* Cấu trúc directory của Project sẽ được tạo như bên dưới



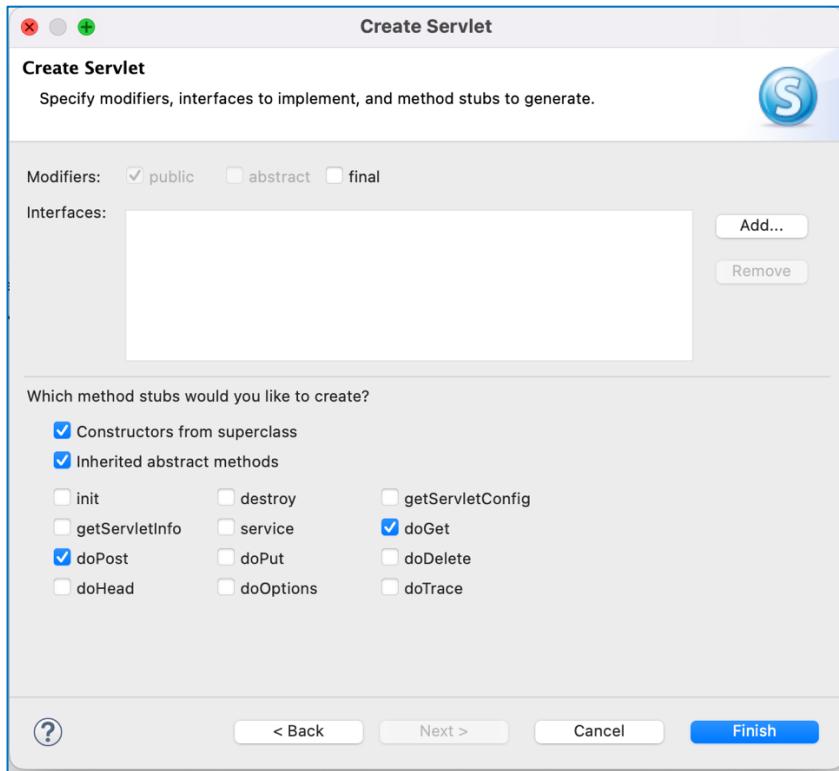
e. **Step 5:** Click vào Project Name vừa tạo: tìm Java Resources → src. Right click trên src → Chọn New → Servlet



f. **Step 6:** Đặt tên cho Servlet và click Next



g. Để default hoặc chọn method cần thiết và click Next



h. Servlet đã được tạo

The screenshot shows the Eclipse IDE's Project Explorer and Java Editor. The Project Explorer on the left displays a project structure with servers, deployment descriptors, and source code. The Java Editor on the right shows the generated code for `HelloServlet.java`.

```

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;

/*
 * Servlet implementation class HelloServlet
 */
private static final long serialVersionUID = 1L;

public HelloServlet() {
    super();
    // TODO Auto-generated constructor stub
}

/*
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.getWriter().append("Served at: ").append(request.getContextPath());
}

/*
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}

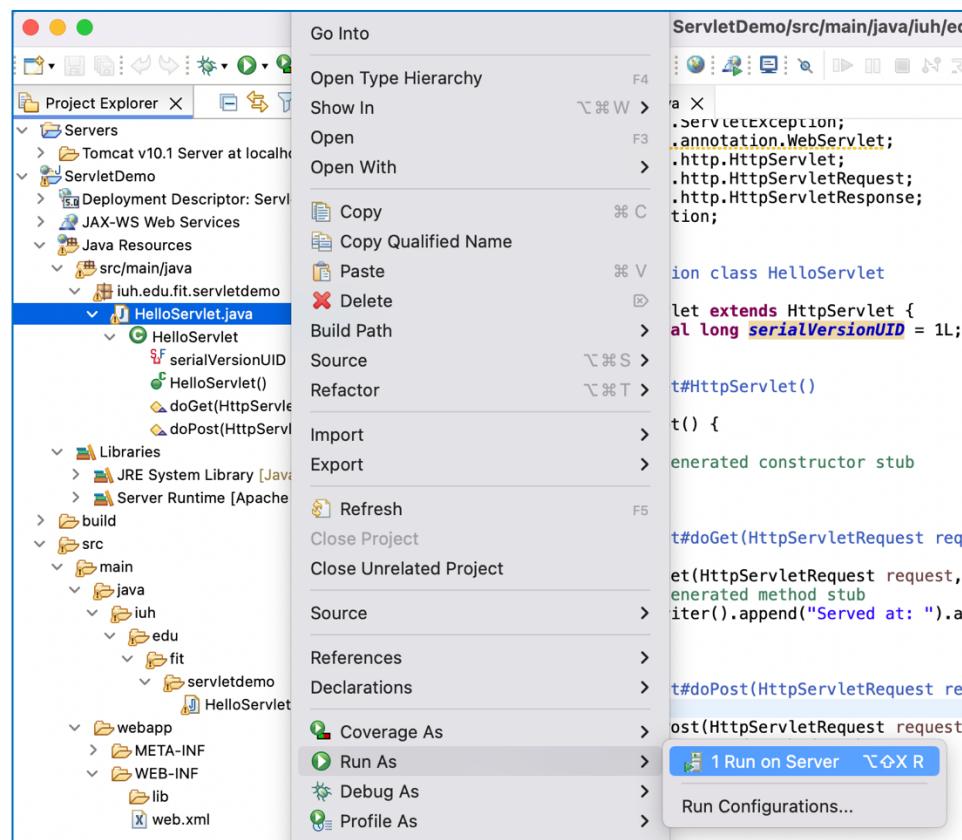
```

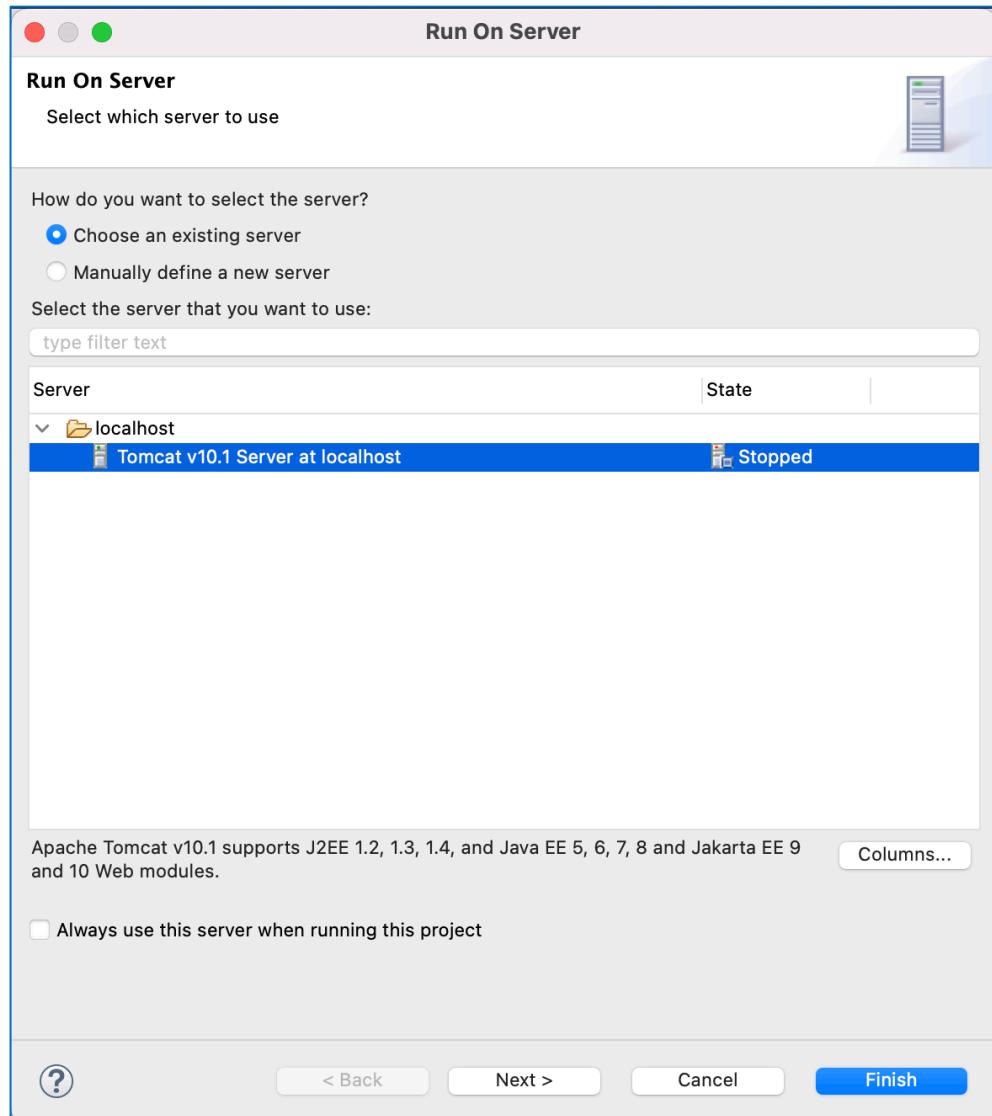
```

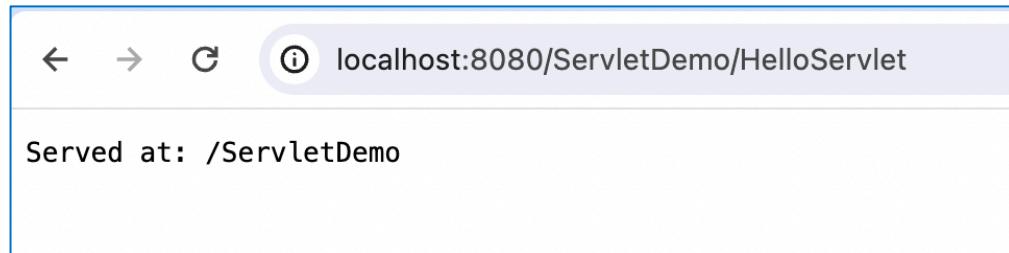
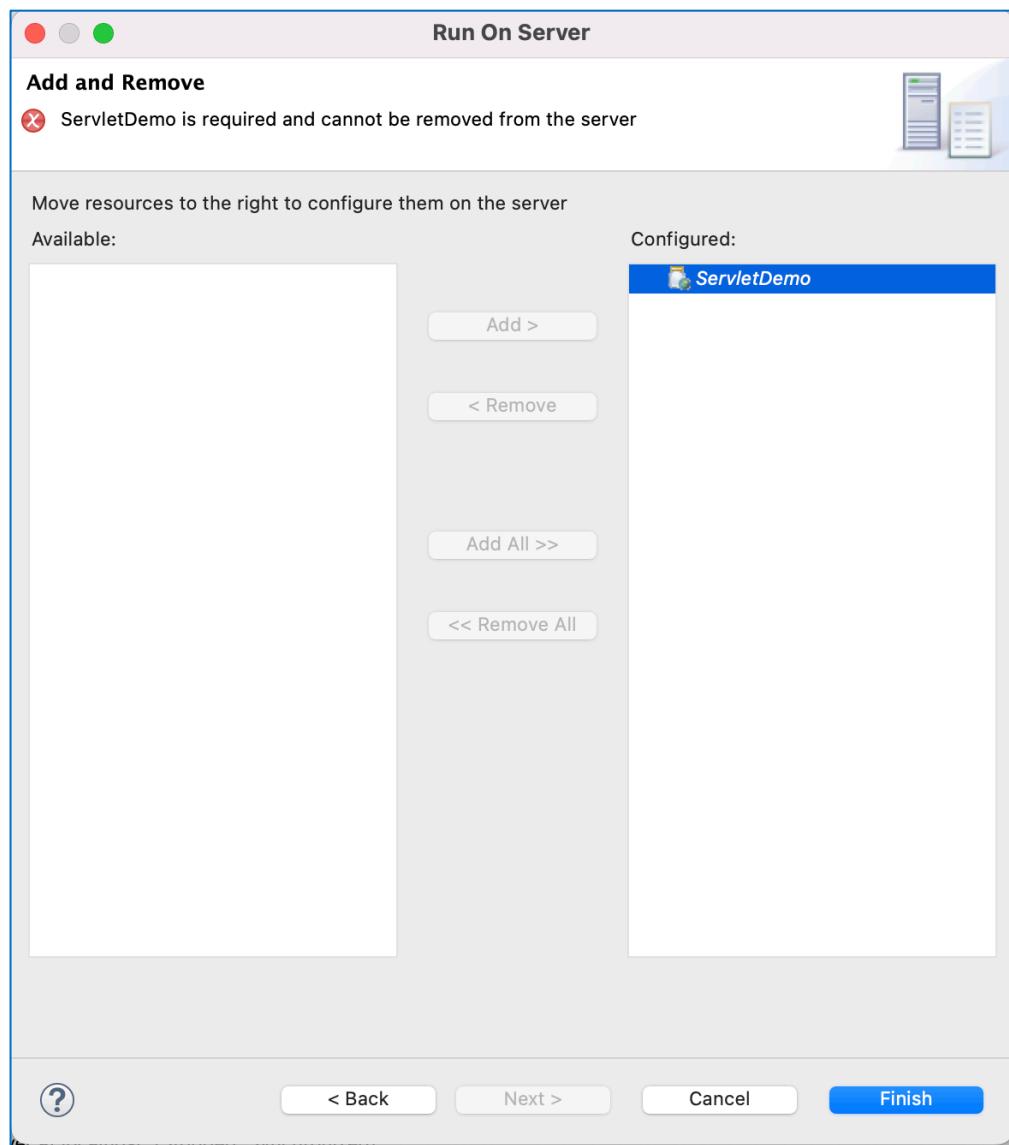
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="https://jakarta.ee/xml/ns/jakartae" xsi:schemaLocation="http://jakarta.ee/xml/ns/jakartae/web-app_6_0.xsd" display-name="ServletDemo">
    <welcome-file-list>
        <welcome-file>index.html</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>
    <servlet>
        <description></description>
        <display-name>HelloServlet</display-name>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>iuh.edu.fit.servletdemo.HelloServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/HelloServlet</url-pattern>
    </servlet-mapping>
</web-app>

```

i. Start server Tomcat và run application







- j. Apply code nhu bên dưới và thay đổi URL cho Servlet

```
/*
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
{
    // TODO Auto-generated method stub
    response.getWriter().append("Served at: ").append(request.getContextPath());

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head><title>Hello Servlet</title></head>");

    out.println("<body>");
    out.println("<h3 style='color: blue'>Hello World</h3>");
    out.println("This is my first Servlet");
    out.println("</body>");
    out.println("</html>");
}
```

```
<!-- Định nghĩa servlet có tên helloServlet, gắn với class HelloServlet -->
<servlet>
    <description></description>
    <display-name>HelloServlet</display-name>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>iuh.edu.fit.servletdemo.HelloServlet</servlet-class>
</servlet>

<!-- Định nghĩa đường dẫn truy cập vào Servlet này -->
<servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/hello</url-pattern>
</servlet-mapping>
```



k. Nguyên tắc hoạt động của Servlet

(1) Browser

localhost:8080/ServletDemo/hello

(2) web.xml

```
<!-- Định nghĩa servlet có tên helloServlet, gắn với class HelloServlet -->
<servlet>
  <description></description>
  <display-name>HelloServlet</display-name>
  <servlet-name>HelloServlet</servlet-name>
  <servlet-class>iuh.edu.fit.servletdemo.HelloServlet</servlet-class>
</servlet>

<!-- Định nghĩa đường dẫn truy cập vào Servlet này -->
<servlet-mapping>
  <servlet-name>HelloServlet</servlet-name>
  <url-pattern>/hello</url-pattern>
</servlet-mapping>
```

(3) HelloServlet.java

```
/*
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    // TODO Auto-generated method stub
    response.getWriter().append("Served at: ").append(request.getContextPath());

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("  <head><title>Hello Servlet</title></head>");

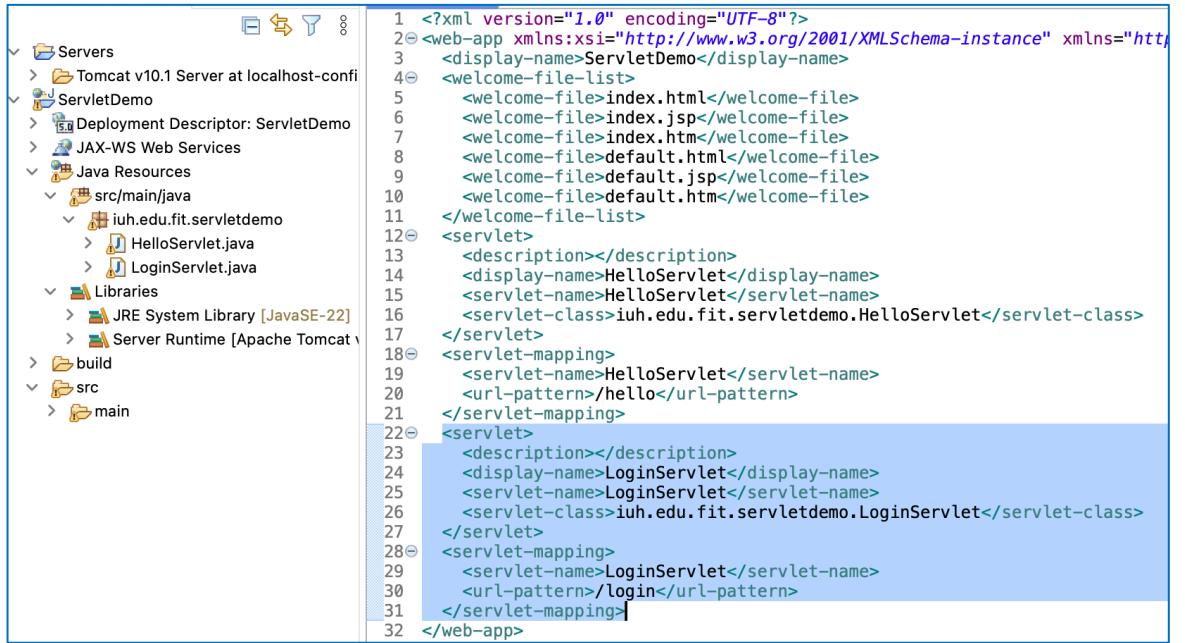
    out.println("  <body>");
    out.println("    <h3 style='color: blue'>Hello World</h3>");
    out.println("    This is my first Servlet");
    out.println("  </body>");
    out.println("</html>");
}
```

(4) Hiển thị ở Client Side & Page Source

The browser window shows the output of the servlet: "Hello World" and "This is my first Servlet". The developer tools show the page source code, which includes the HTML structure and the Java code that generated it.

```
<html>
  <head> == $0
    <title>Hello Servlet</title>
  </head>
  <body cz-shortcut-listen="true">
    <h3 style="color: blue">Hello World</h3>
    " This is my first Servlet "
  </body>
</html>
```

1. Khi thêm Servlet mới (LoginServlet) thì file web.xml sẽ add thêm Servlet mới



The screenshot shows the Eclipse IDE interface with the Project Explorer on the left and the code editor on the right. The Project Explorer displays a Java-based web application structure:

- Servers: Tomcat v10.1 Server at localhost-config
- ServletDemo: Deployment Descriptor (web.xml)
- Java Resources:
 - src/main/java:
 - iuh.edu.fit.servletdemo:
 - HelloServlet.java
 - LoginServlet.java
 - Libraries:
 - JRE System Library [JavaSE-17]
 - Server Runtime [Apache Tomcat]
 - build
 - src
 - main

The code editor shows the content of the web.xml file, which defines two servlets: HelloServlet and LoginServlet, and their mappings to URL patterns /hello and /login respectively.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" version="4.0">
    <display-name>ServletDemo</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>
    <servlet>
        <description></description>
        <display-name>HelloServlet</display-name>
        <servlet-name>HelloServlet</servlet-name>
        <servlet-class>iuh.edu.fit.servletdemo>HelloServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
    <servlet>
        <description></description>
        <display-name>LoginServlet</display-name>
        <servlet-name>LoginServlet</servlet-name>
        <servlet-class>iuh.edu.fit.servletdemo>LoginServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>LoginServlet</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>
</web-app>
```