

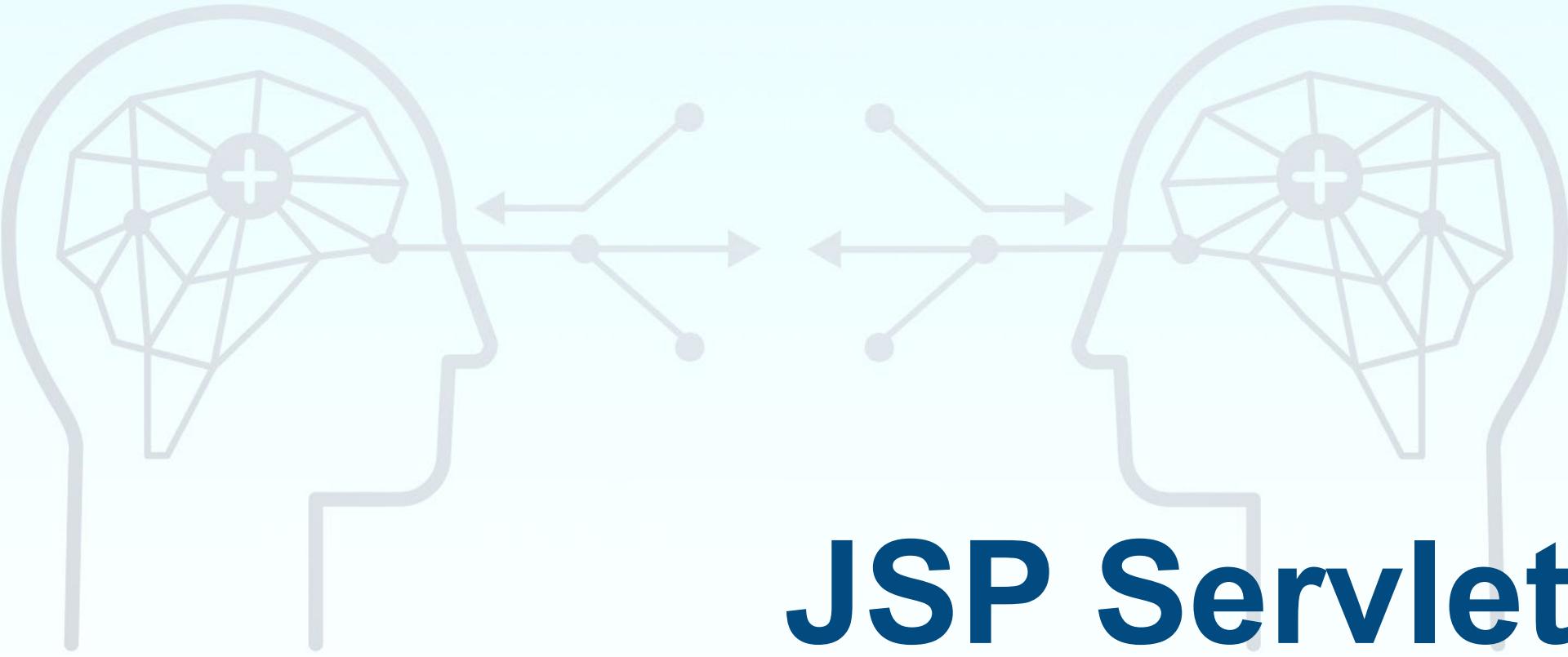


[한국폴리텍대학 성남캠퍼스 인공지능소프트웨어과]

JSP Servlet

2023. 5

김형오

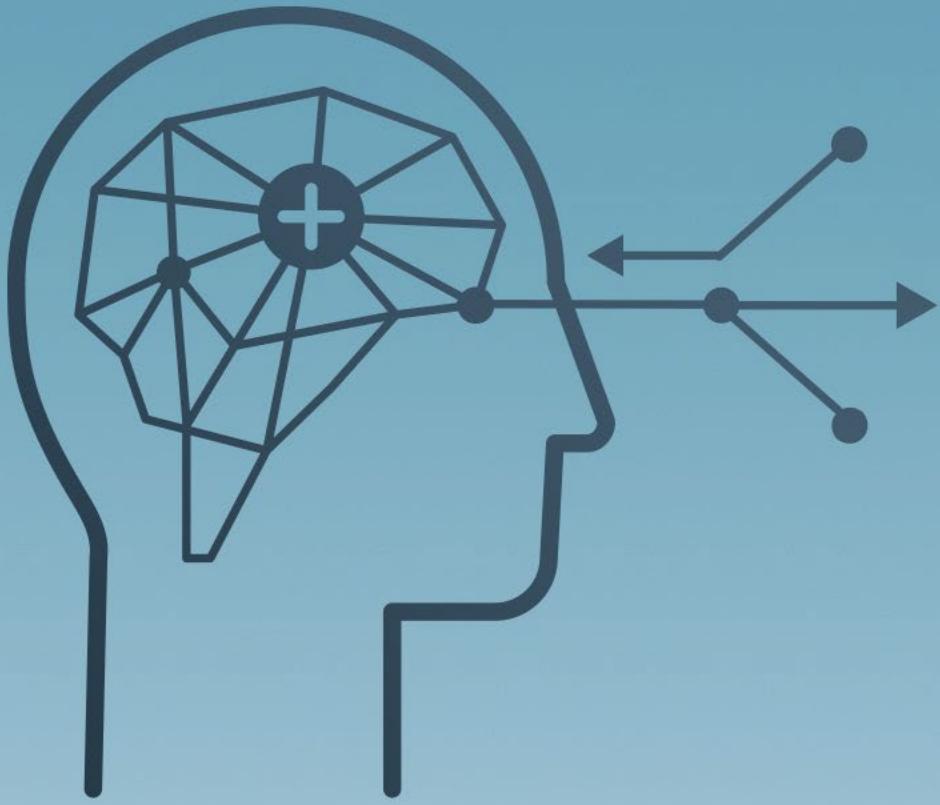


DEEP
LEARNING

MACHINE LEARNING BASED
ON ARTIFICIAL NEURAL NETWORKS

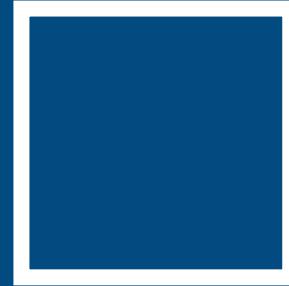
JSP Servlet

DEEP
LEARNING
MACHINE LEARNING BASED
ON ARTIFICIAL NEURAL NETWORKS



DEEP LEARNING

MACHINE LEARNING BASED
ON ARTIFICIAL NEURAL NETWORKS



목차

A table of Contents

#1, 인터넷과 웹

#2, RESTful API

#3, Java와 서블릿

#4, 공공데이터 포털

Part 1, JSP 개발 환경

한빛미디어, IT CookBook, 쉽게 배우는 JSP 웹 프로그래밍

참고문헌

이클립스

- `eclipse-jee-2021-03-R-win32-x86_64`

<https://www.eclipse.org/downloads/packages/release>

The screenshot shows the Eclipse Foundation website with a dark header. The header includes the Eclipse Foundation logo, a search bar, and navigation links for Projects, Working Groups, Members, and More. Below the header, a breadcrumb navigation shows Home / Downloads / Packages / Release / Eclipse Packaging Project (EPP) Releases. The main content area is titled "Eclipse Packaging Project (EPP) Releases" and lists a series of release dates from 2019-09 to 2023-06. To the right, there is a "Fraunhofer FOKUS" logo with the tagline "We connect everything" and a note about the Eclipse Installer 2023-03 R including a JRE for macOS, Windows and Linux. At the bottom, there is a "Get Eclipse IDE 2023-03" section with a download button for x86_64 and links for "Download Packages" and "Need Help?".

ECLIPSE FOUNDATION

Projects Working Groups Members More ▾

Home / Downloads / Packages / Release / Eclipse Packaging Project (EPP) Releases

Eclipse Installer Eclipse Packages Eclipse Developer Builds ▾

Eclipse Packaging Project (EPP) Releases

- 2023-06
- 2023-03
- 2022-12
- 2022-09
- 2022-06
- 2022-03
- 2021-12
- 2021-09
- 2021-06
- 2021-03
- 2020-12
- 2020-09
- 2020-06
- 2020-03
- 2019-12
- 2019-09

Fraunhofer
FOKUS
We connect everything

The Eclipse Installer 2023-03 R now includes a JRE for macOS, Windows and Linux.

Get Eclipse IDE 2023-03

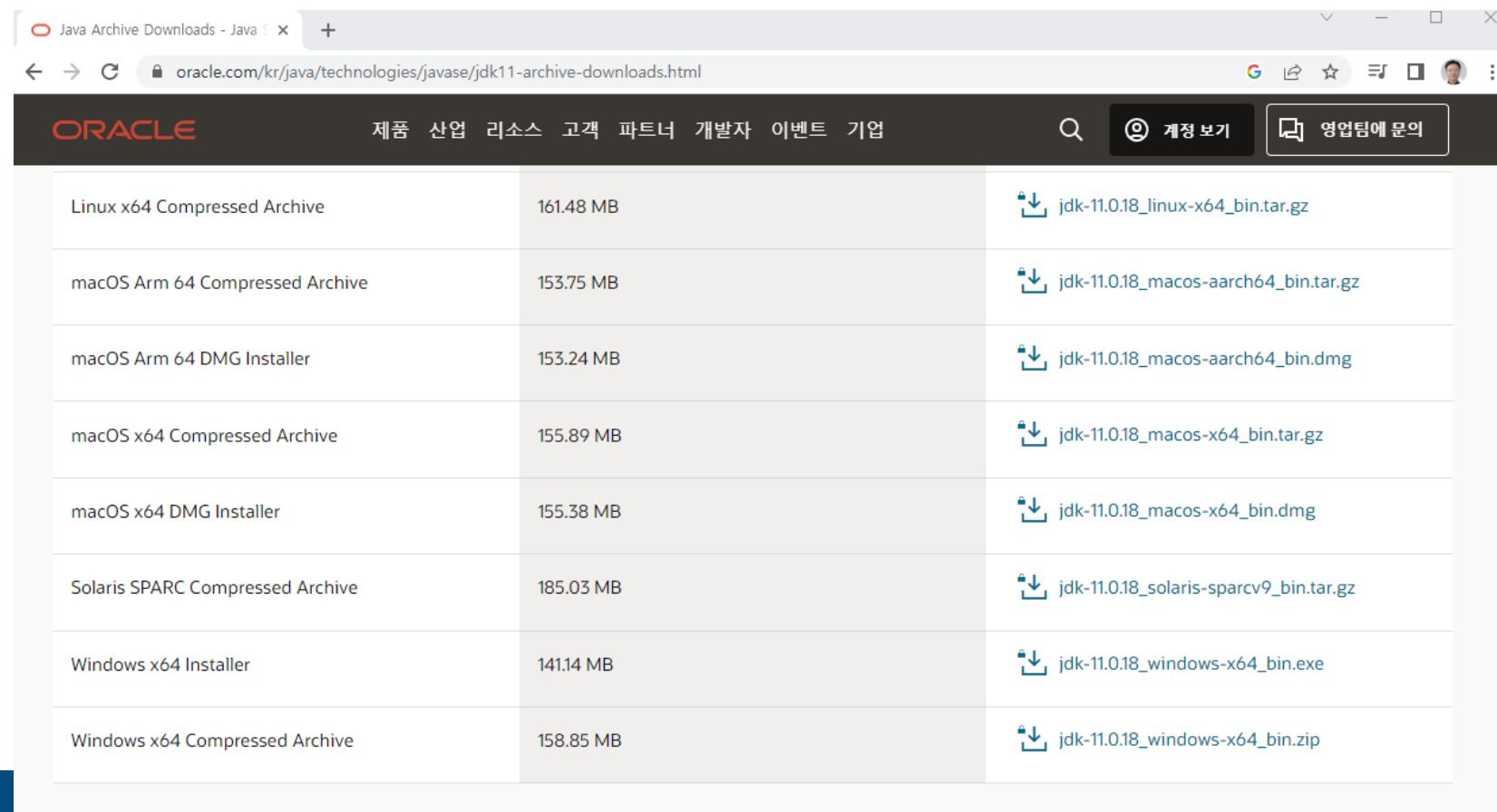
Install your favorite desktop IDE packages.

Download x86_64

[Download Packages](#) | [Need Help?](#)

jdk-11.0.18_windows-x64_bin

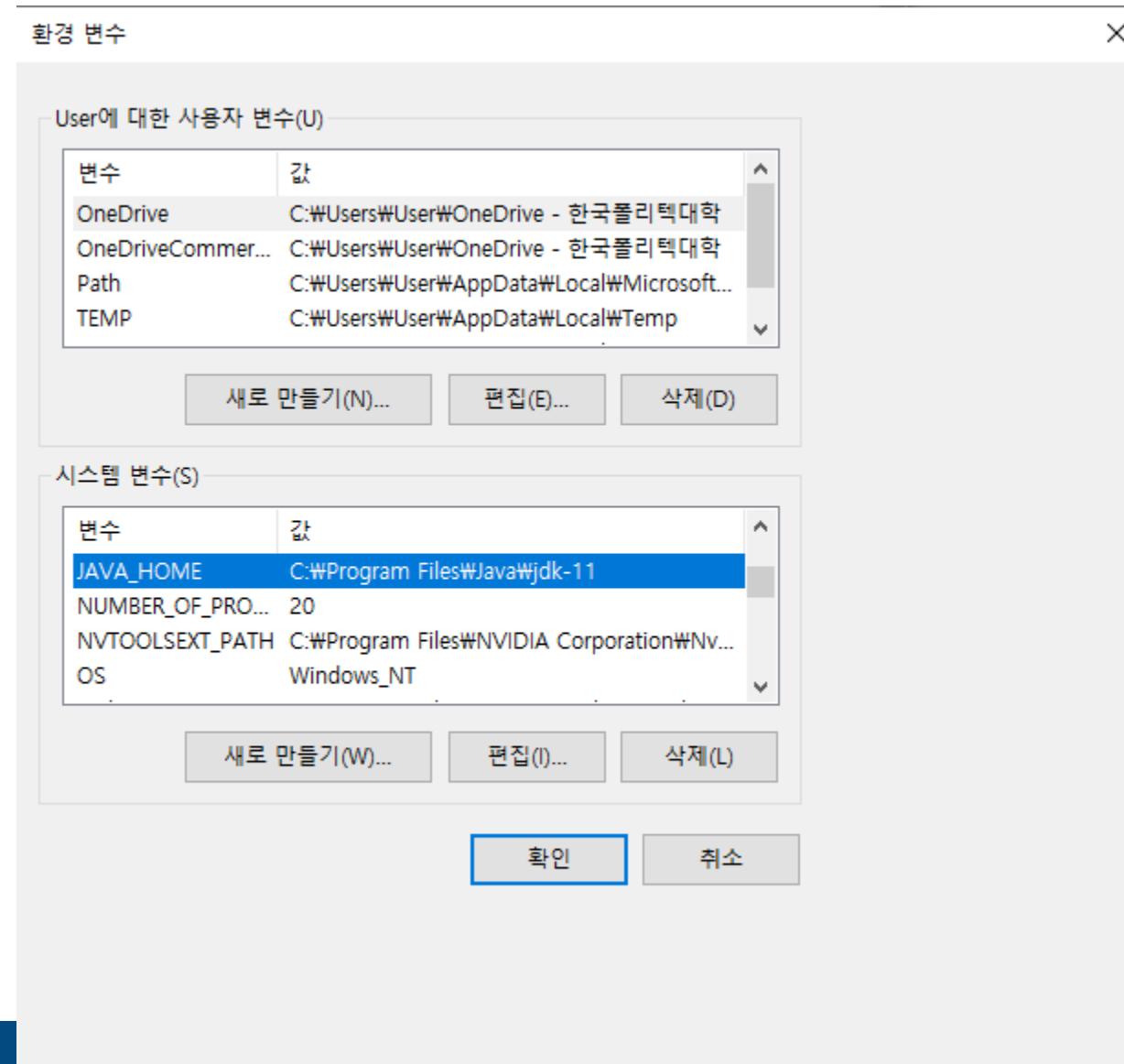
<https://www.oracle.com/kr/java/technologies/javase/jdk11-archive-downloads.html>



The screenshot shows a list of Java 11 JDK download options. The table has three columns: Platform/Type, Size, and Download Link. The platforms listed are Linux x64, macOS Arm 64, macOS Arm 64 DMG, macOS x64, macOS x64 DMG, Solaris SPARC, Windows x64, and Windows x64 Compressed Archive. The sizes range from 141.14 MB to 185.03 MB. Each row contains a download icon and a link to the corresponding archive file.

Platform/Type	Size	Download Link
Linux x64 Compressed Archive	161.48 MB	jdk-11.0.18_linux-x64_bin.tar.gz
macOS Arm 64 Compressed Archive	153.75 MB	jdk-11.0.18_macos-aarch64_bin.tar.gz
macOS Arm 64 DMG Installer	153.24 MB	jdk-11.0.18_macos-aarch64_bin.dmg
macOS x64 Compressed Archive	155.89 MB	jdk-11.0.18_macos-x64_bin.tar.gz
macOS x64 DMG Installer	155.38 MB	jdk-11.0.18_macos-x64_bin.dmg
Solaris SPARC Compressed Archive	185.03 MB	jdk-11.0.18_solaris-sparcv9_bin.tar.gz
Windows x64 Installer	141.14 MB	jdk-11.0.18_windows-x64_bin.exe
Windows x64 Compressed Archive	158.85 MB	jdk-11.0.18_windows-x64_bin.zip

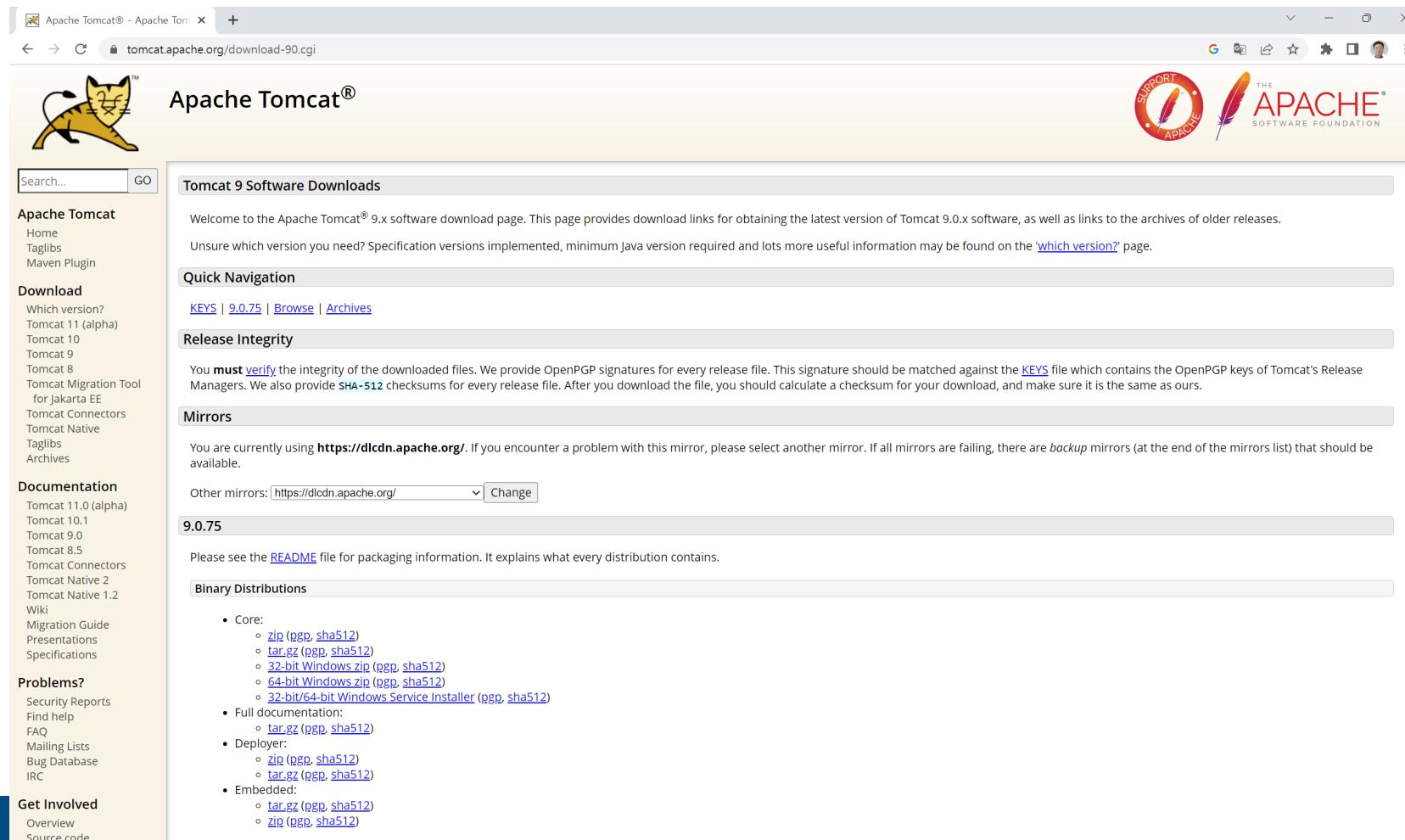
Java_HOME 설정



Apache-Tomcat

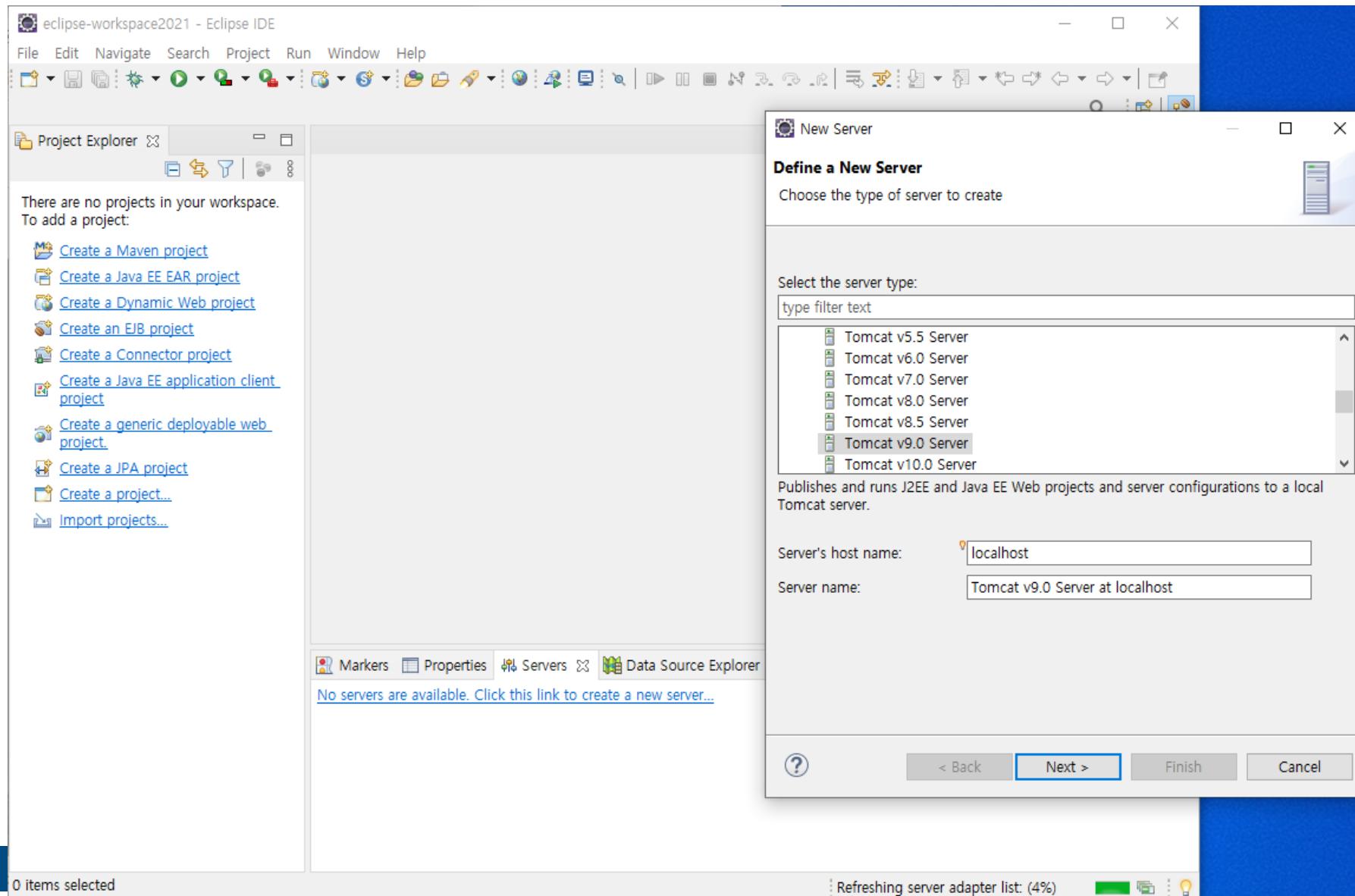
apache-tomcat-9.0.73-windows-x64

<https://tomcat.apache.org/download-90.cgi>

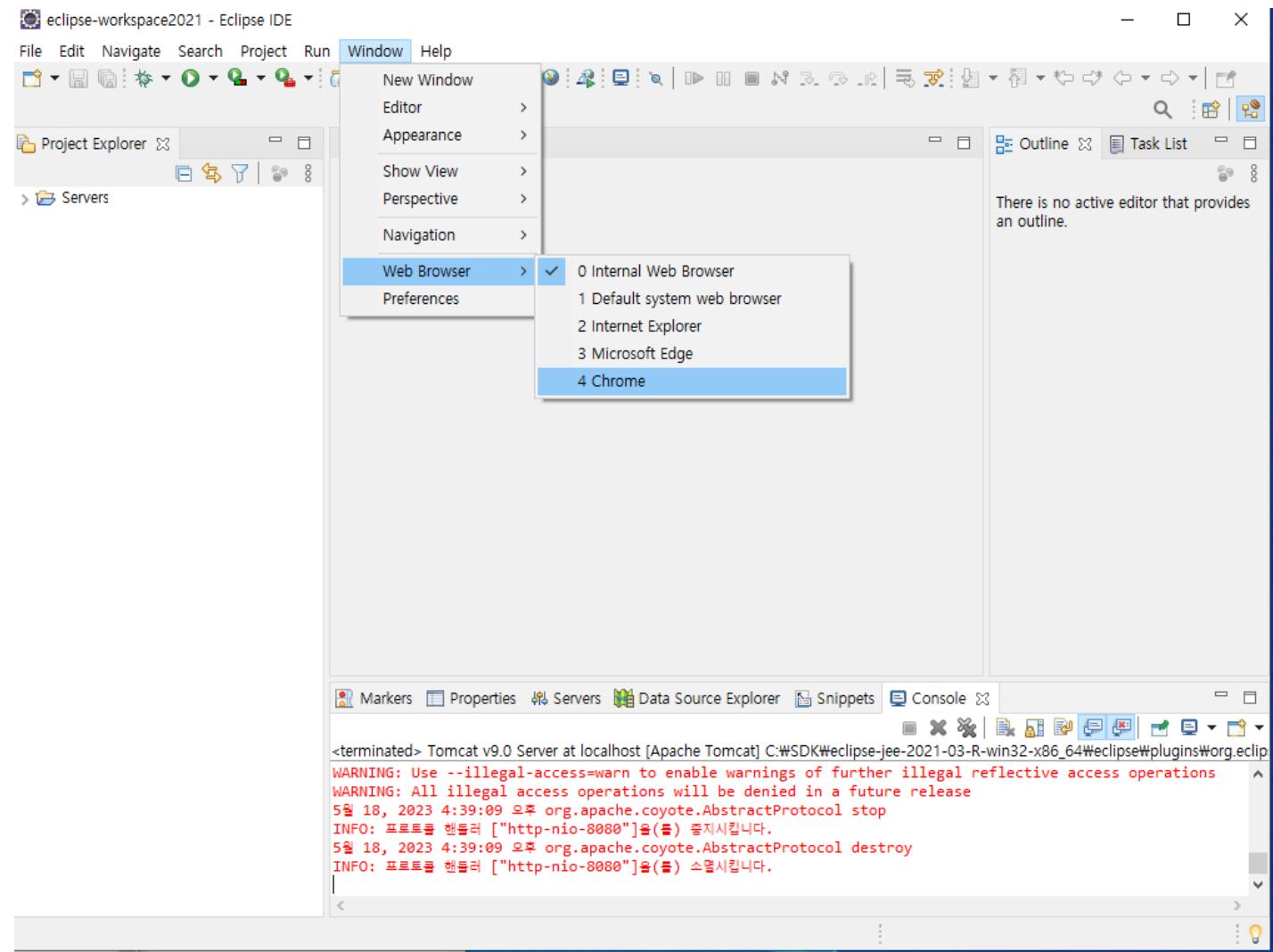


The screenshot shows a Windows desktop browser window displaying the Apache Tomcat 9 Software Downloads page. The page features the Apache logo (a yellow cat) and the Apache Software Foundation logo (two colorful feathers). The main content area is titled 'Tomcat 9 Software Downloads' and includes sections for 'Quick Navigation' (with links to 'KEYS', '9.0.75', 'Browse', and 'Archives'), 'Release Integrity' (instructions for verifying file integrity using OpenPGP signatures and SHA-512 checksums), and 'Mirrors' (a list of mirrors, with the current one being <https://dlcdn.apache.org/>). The left sidebar contains links for 'Apache Tomcat' (Home, Taglibs, Maven Plugin), 'Download' (Tomcat 11 (alpha), Tomcat 10, Tomcat 9, Tomcat 8, Tomcat 7, Tomcat Migration Tool for Jakarta EE, Tomcat Connectors, Tomcat Native, Taglibs, Archives), 'Documentation' (Tomcat 11.0 (alpha), Tomcat 10.1, Tomcat 9.0, Tomcat 8.5, Tomcat Connectors, Tomcat Native 2, Tomcat Native 1.2, Wiki, Migration Guide, Presentations, Specifications), 'Problems?' (Security Reports, Find help, FAQ, Mailing Lists, Bug Database, IRC), and 'Get Involved' (Overview, Source code).

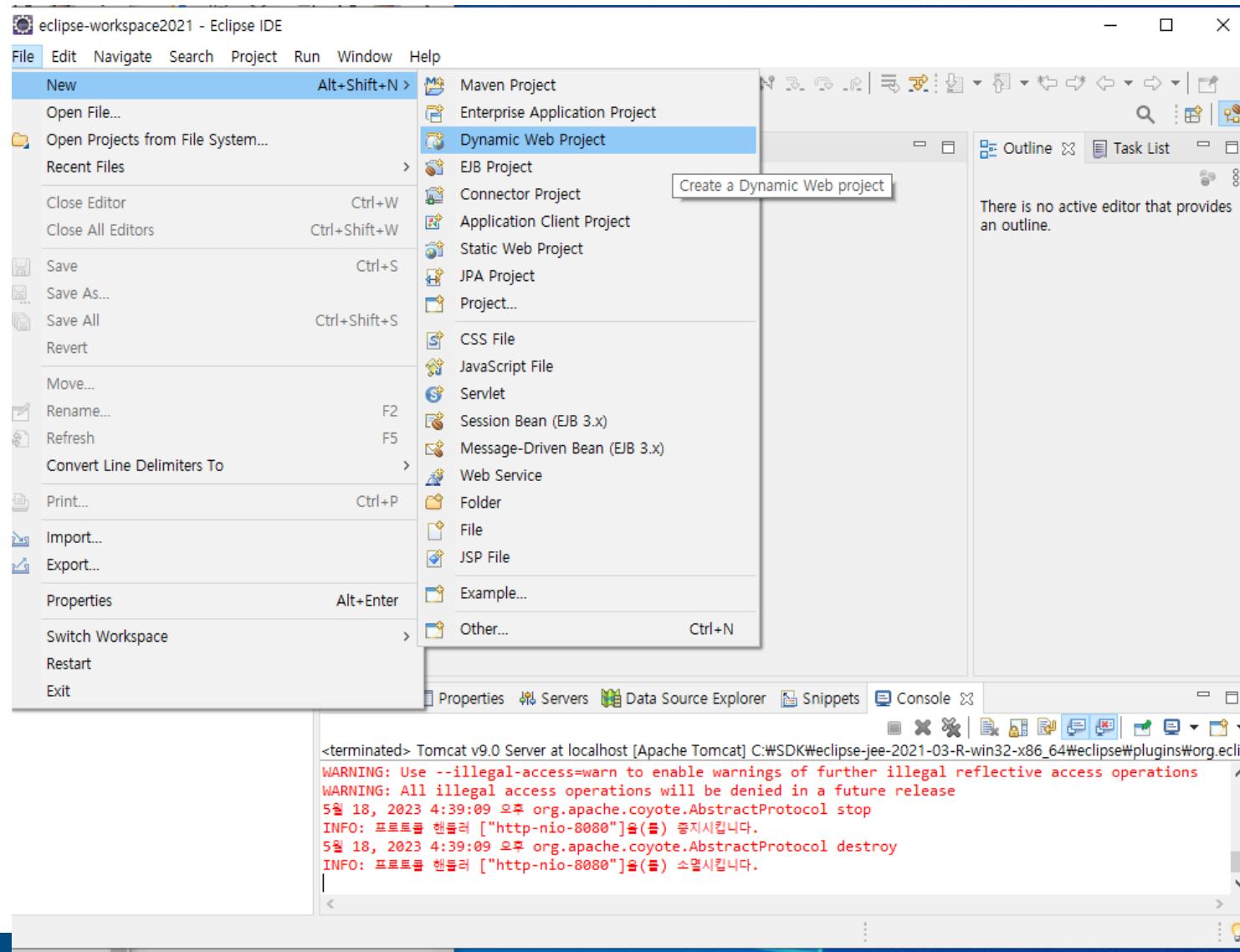
Apache-Tomcat 설정



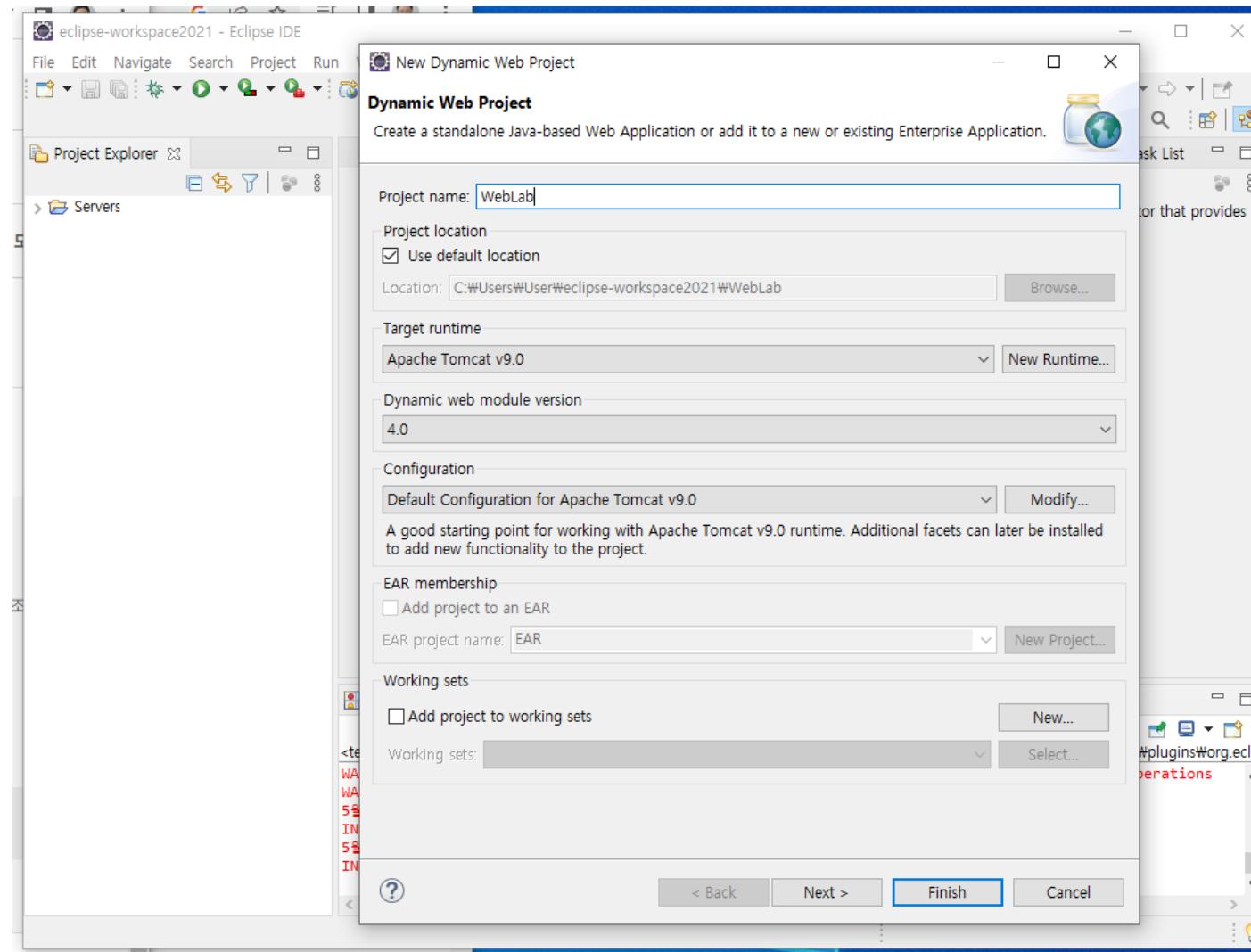
Web Browser 설정



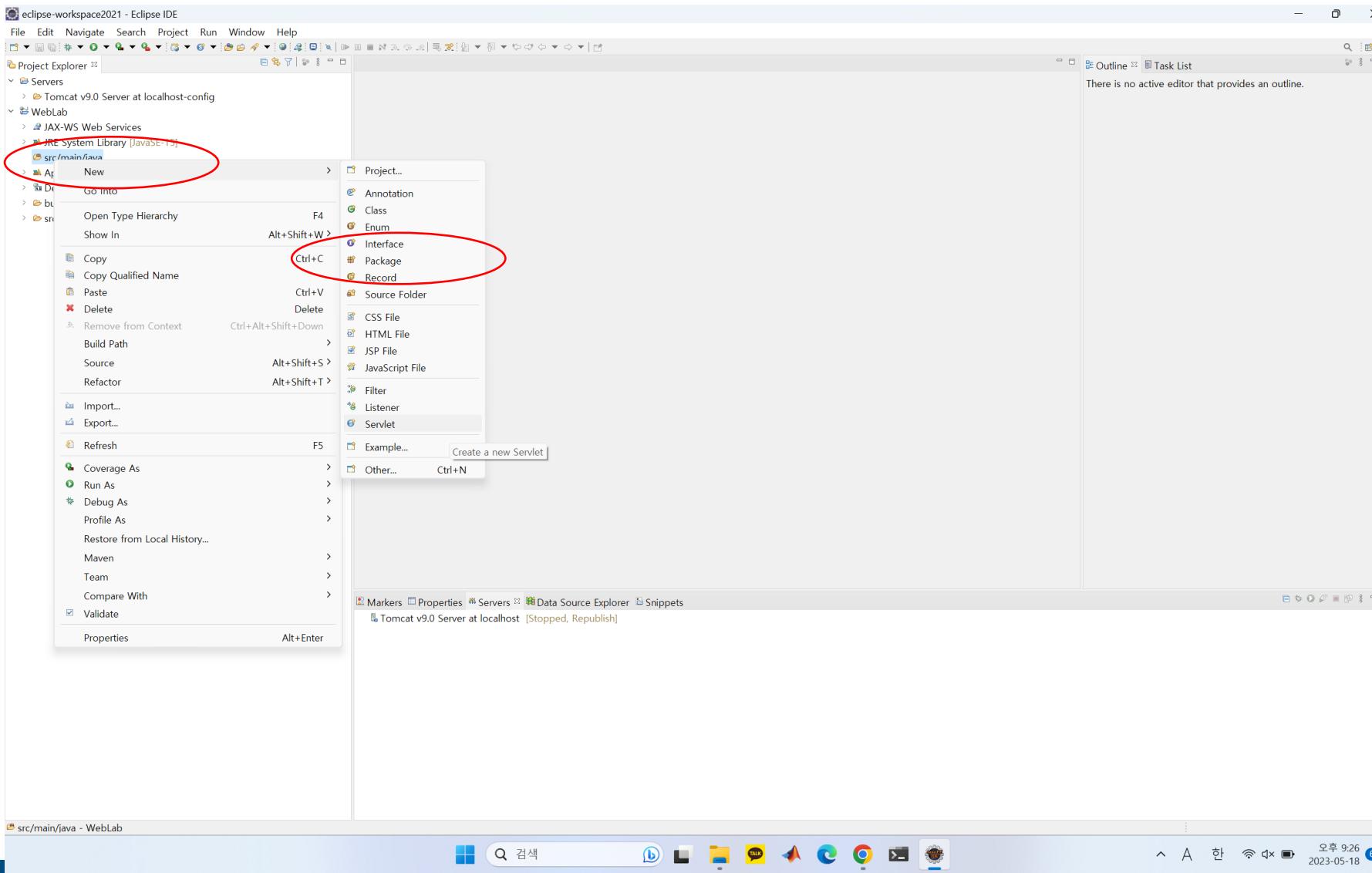
New Project



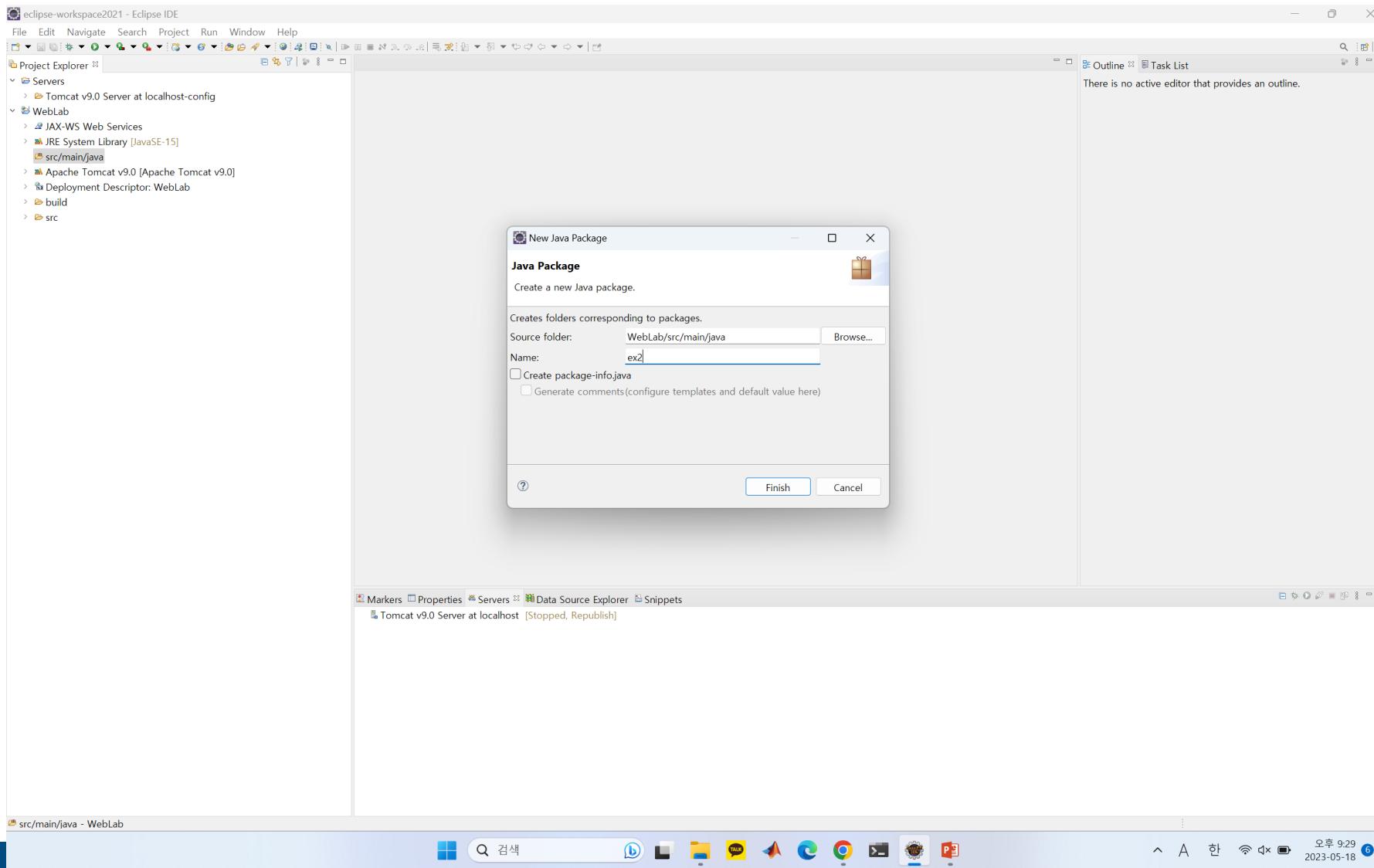
New Project



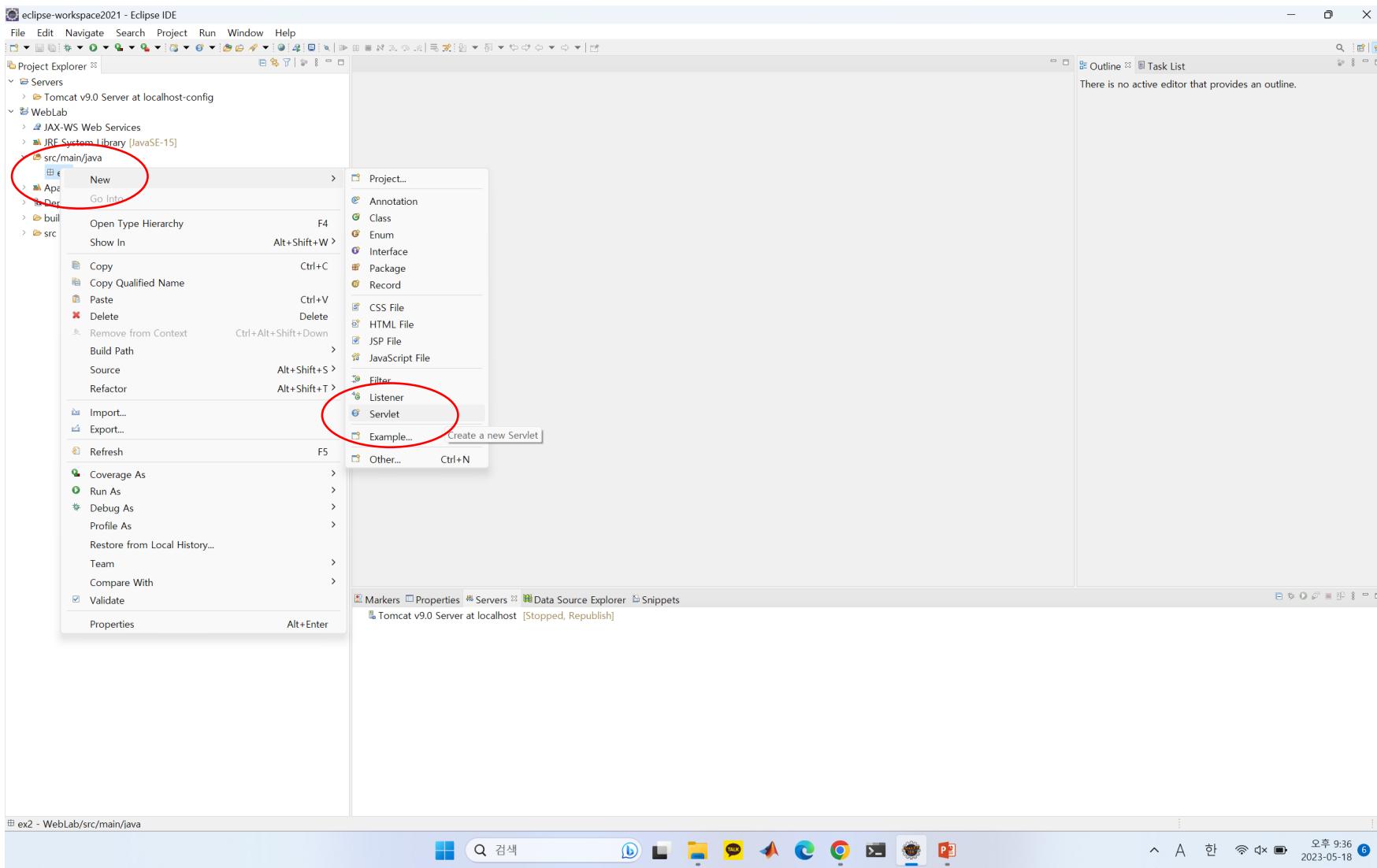
New package



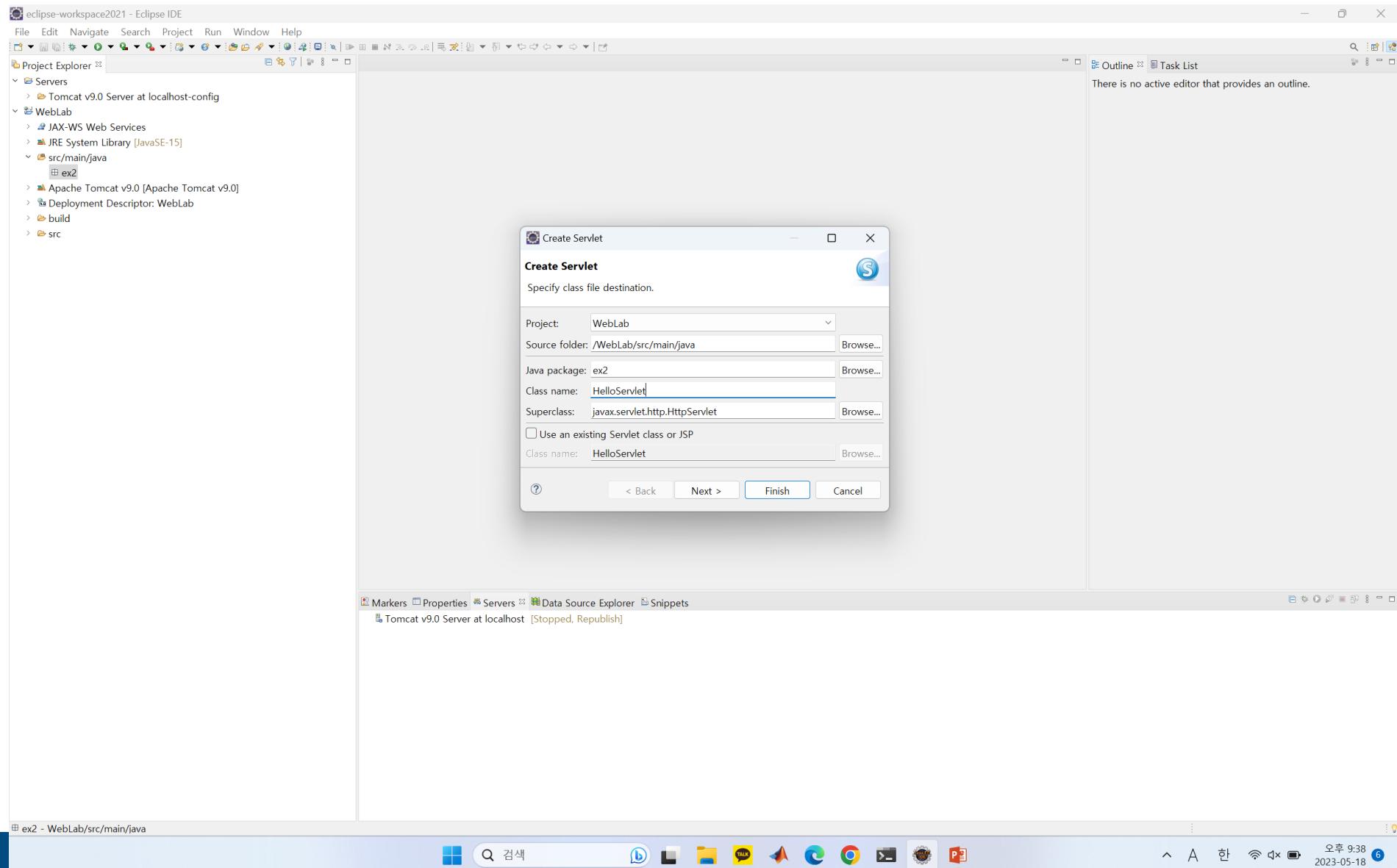
New package



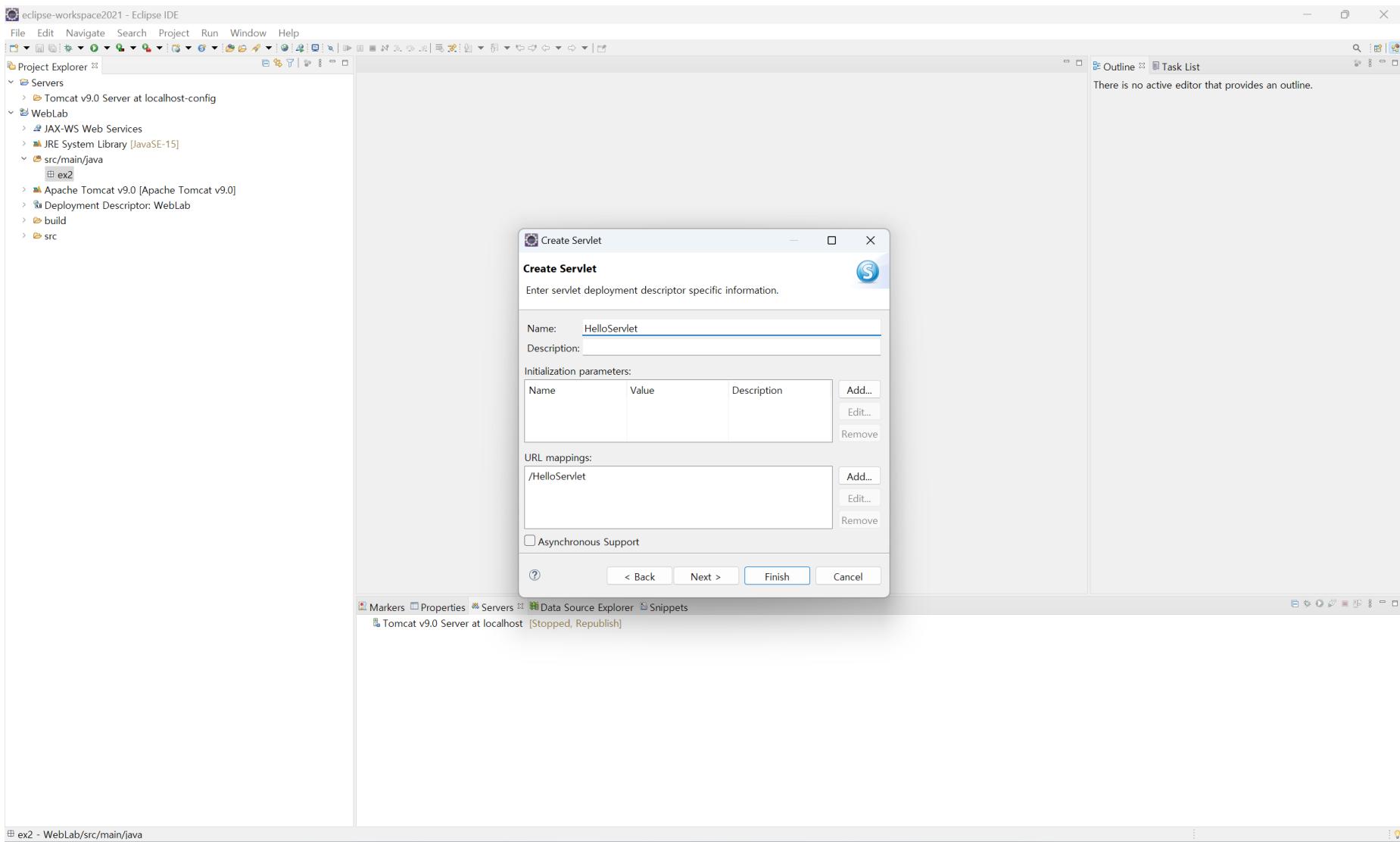
New servlet



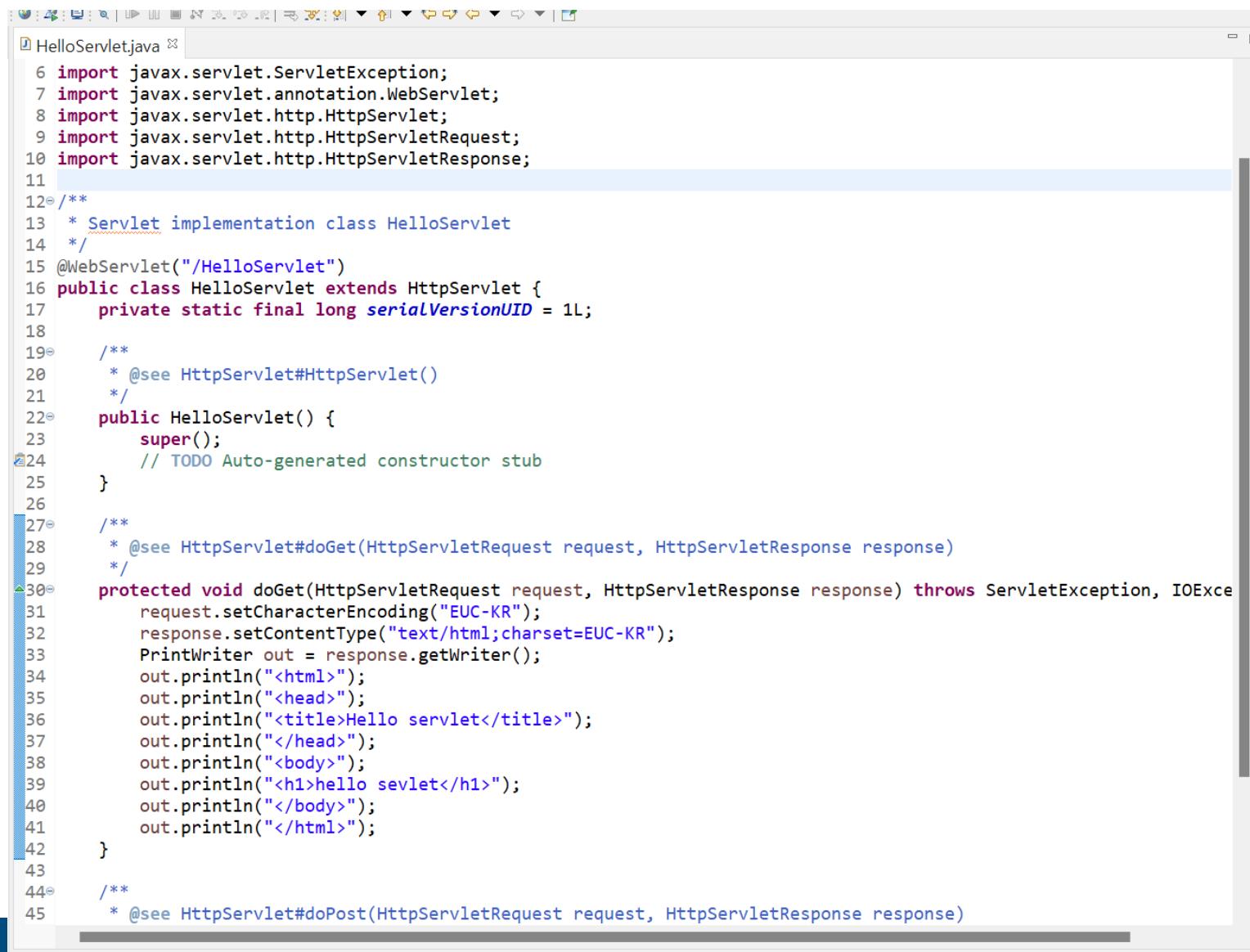
New servlet



New servlet



New servlet

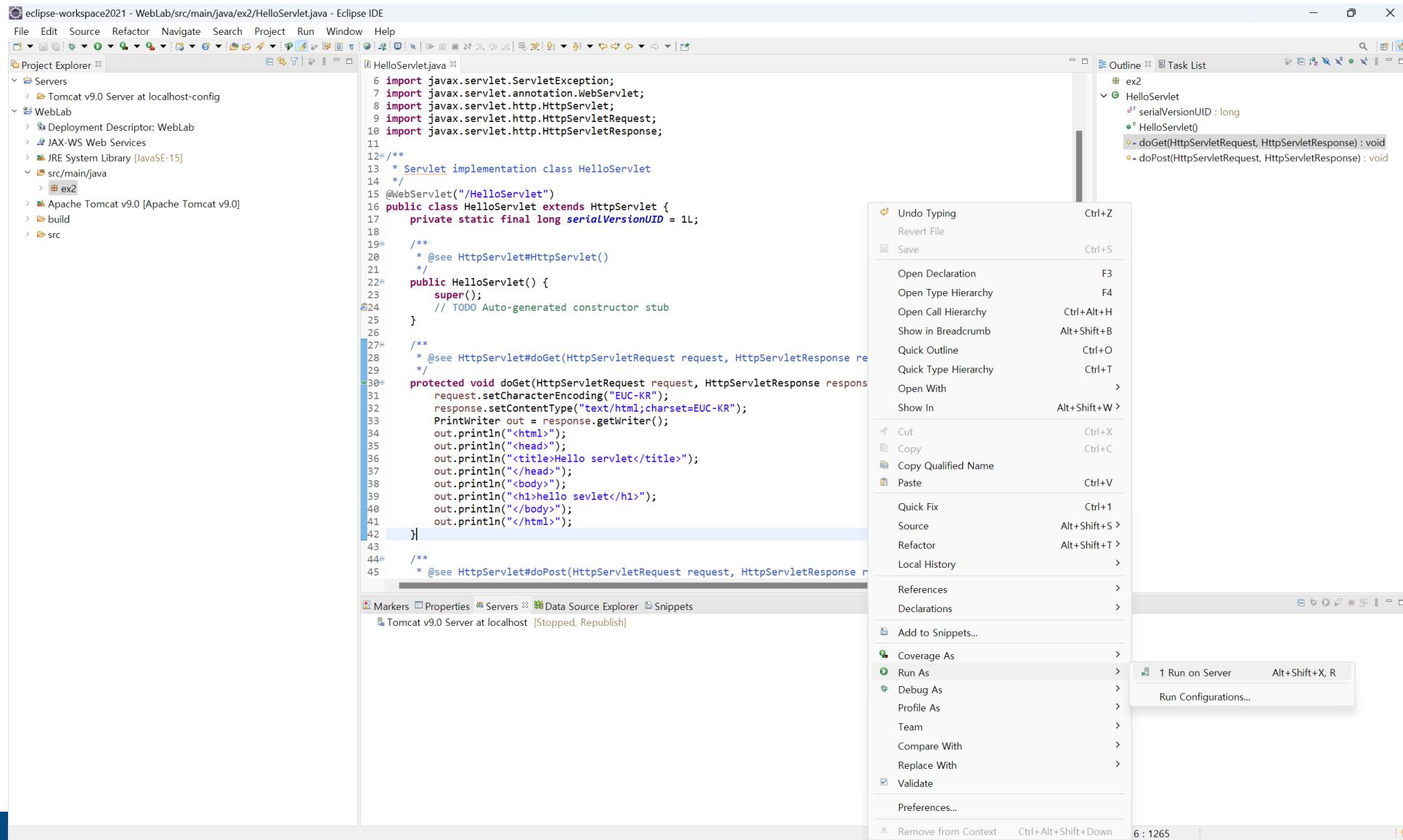


```
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 /**
13  * Servlet implementation class HelloServlet
14 */
15 @WebServlet("/HelloServlet")
16 public class HelloServlet extends HttpServlet {
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     /**
28      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
29     */
30     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
31         response.setCharacterEncoding("EUC-KR");
32         response.setContentType("text/html; charset=EUC-KR");
33         PrintWriter out = response.getWriter();
34         out.println("<html>");
35         out.println("<head>");
36         out.println("<title>Hello servlet</title>");
37         out.println("</head>");
38         out.println("<body>");
39         out.println("<h1>hello servlet</h1>");
40         out.println("</body>");
41         out.println("</html>");
42     }
43
44     /**
45      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
46     */
47 }
```

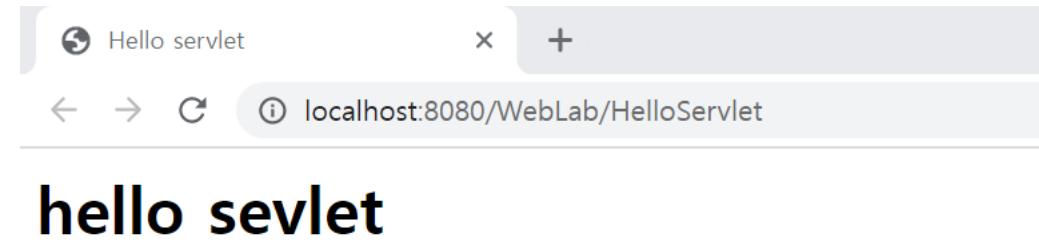
New servlet

```
request.setCharacterEncoding("EUC-KR");
response.setContentType("text/html;charset=EUC-KR");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head>");
out.println("<title>Hello servlet</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>hello sevlet</h1>");
out.println("</body>");
out.println("</html>");
```

Run



실행결과





Part 2,

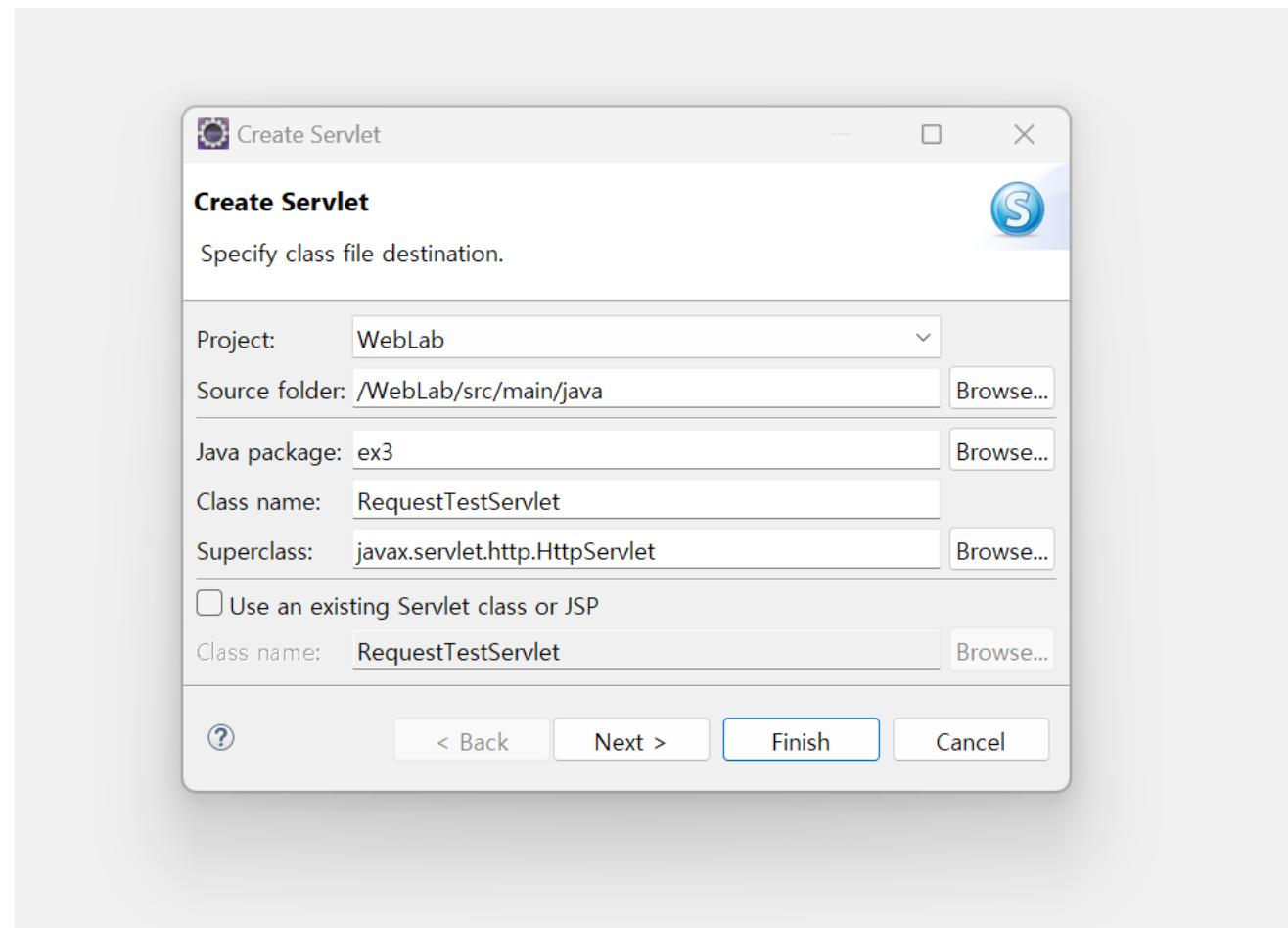
Request분석

참고문헌

Request 분석



Request 분석

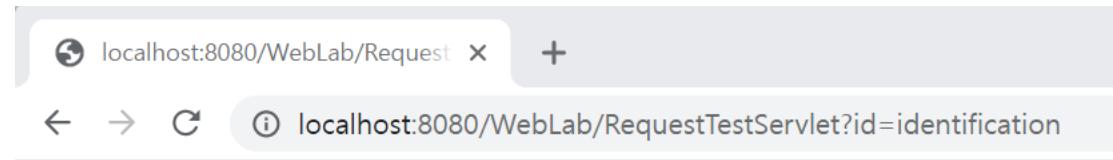


Request 분석

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String method = request.getMethod();
    String query = request.getQueryString();
    String uri = request.getRequestURI();
    String url = request.getRequestURL().toString();

    PrintWriter out = response.getWriter();
    out.println("<h1>Request Test</h1>");
    out.println("method =" + method + "<br/>");
    out.println("query =" + query + "<br/>");
    out.println("uri =" + uri + "<br/>");
    out.println("url =" + url + "<br/>");
}
```

Request 분석



Request Test

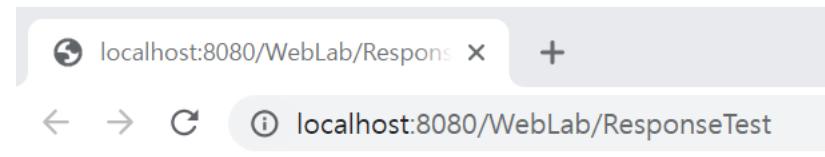
method =GET
query =id=identification
uri =/WebLab/RequestTestServlet
url =http://localhost:8080/WebLab/RequestTestServlet



Part 3, Response분석

참고문헌

Response 분석



Response Test

응답 테스트

Response 분석

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
response.setContentType("text/html;charset=EUC-KR");
response.setHeader("chache-control", "no-cache");
response.setHeader("expires", "0");

PrintWriter out = response.getWriter();
out.println("<h1>Response Test</h1>");
out.println("<h1>응답 테스트</h1>");
}
```

Response 분석 – Redirect

Servlet1st.Java

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<h1> I am One Servlet</h1>");

response.sendRedirect("Servlet2nd");
}
```

Servlet2nd.Java

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<h1>TwoServlet</h1>");
}
```

Response 분석 – Forward

Servlet1st.Java

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<h1> I am One Servlet</h1>");

RequestDispatcher rd = request.getRequestDispatcher("Servlet2nd");
rd.forward(request, response);
}
```

Servlet2nd.Java

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<h1>TwoServlet</h1>");
}
```

Response 분석 – Include

Servlet1st.Java

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<h1> I am One Servlet</h1>");

RequestDispatcher rd = request.getRequestDispatcher("Servlet2nd");
rd.include(request, response);
}
```

Servlet2nd.Java

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
PrintWriter out = response.getWriter();
out.println("<h1>TwoServlet</h1>");
}
```

Part 4,

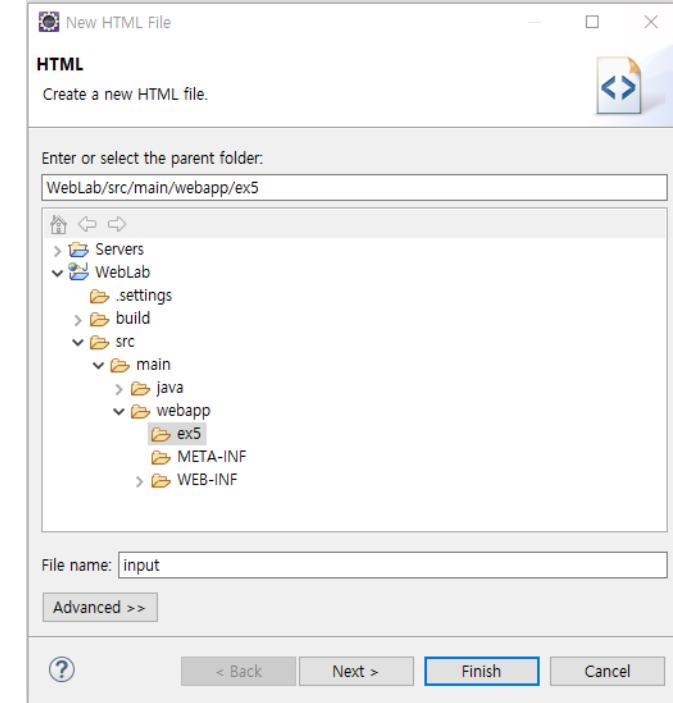
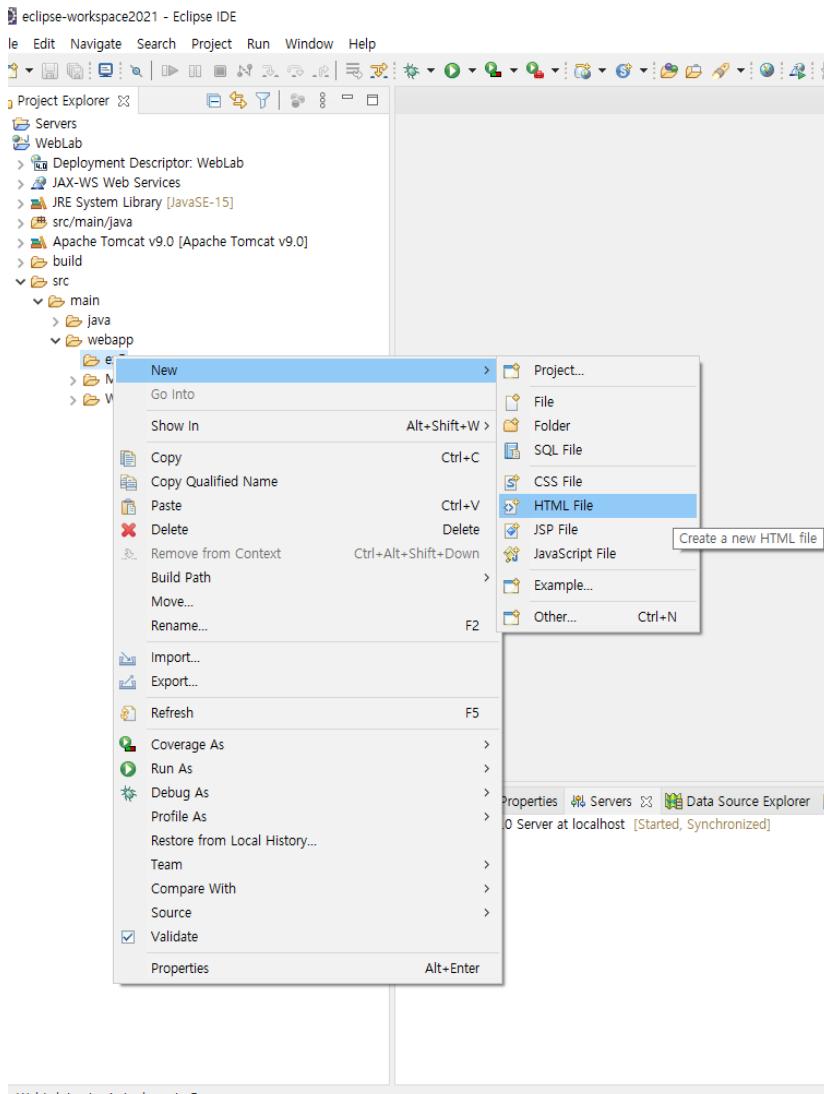
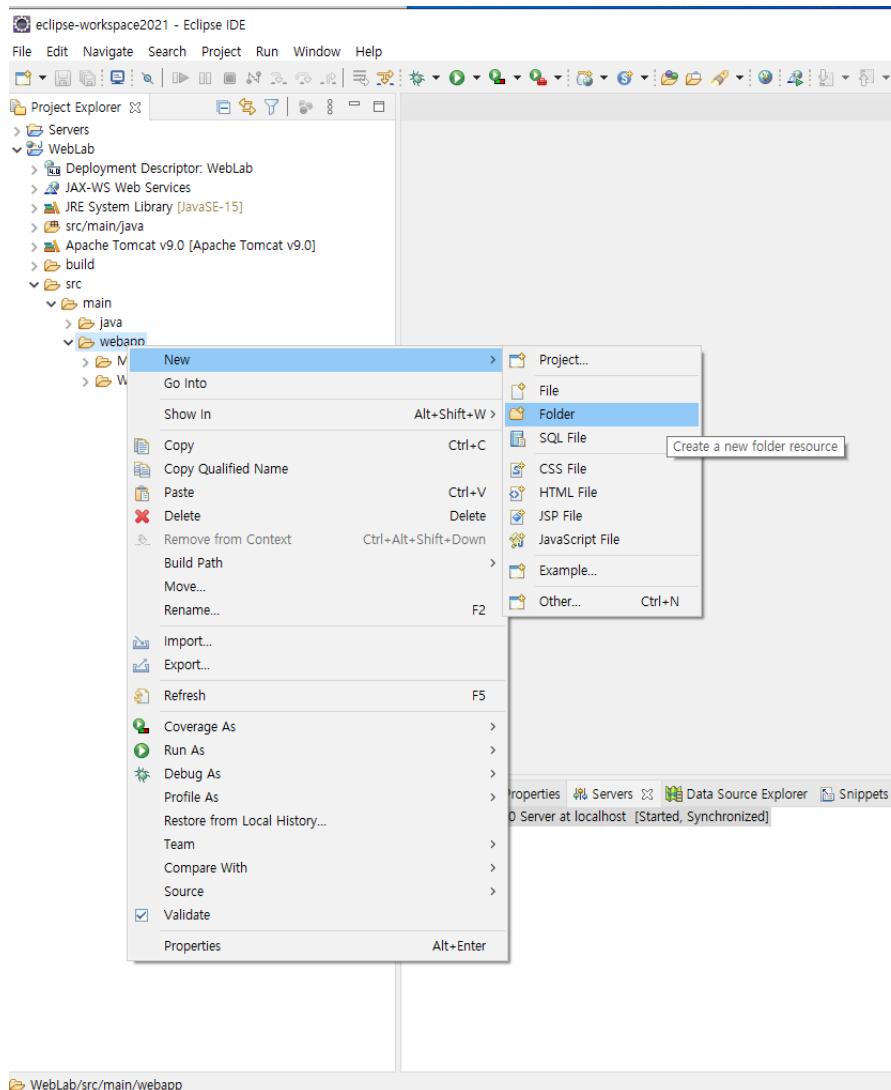
Web Query

참고문헌



Part 4

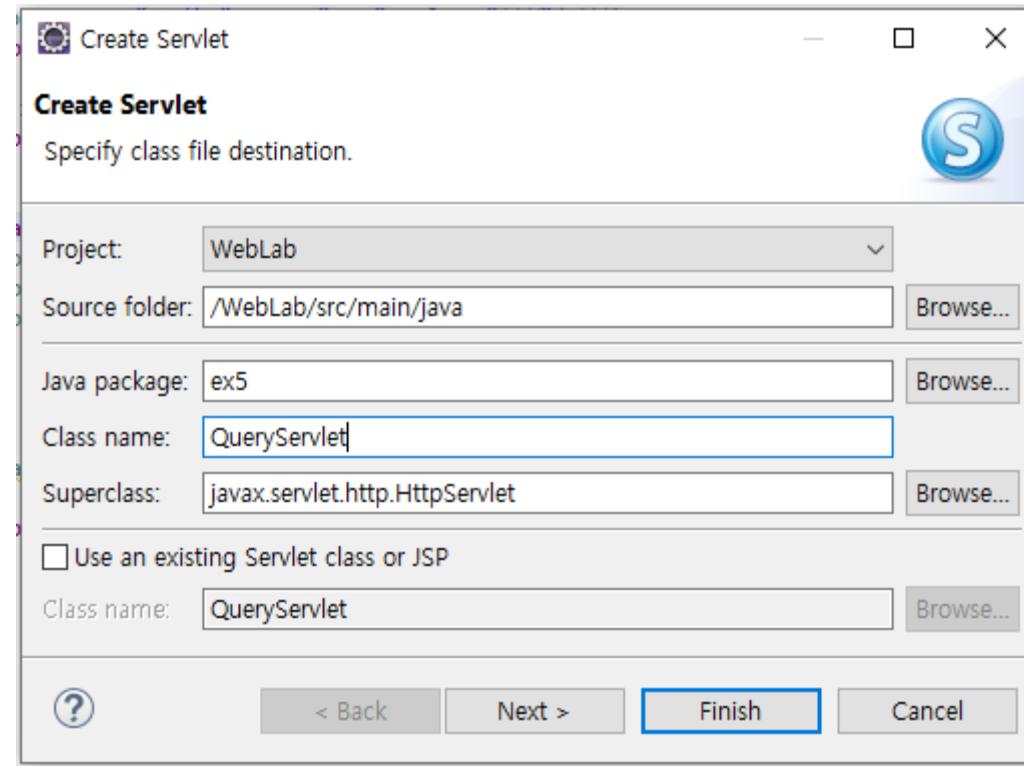
Input HTML



Input HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Insert title here</title>
</head>
<body>
<form method="get" action="/WebLab/QueryServlet">
  name : <input type="text" name="name"/><br/>
  password : <input type="password" name="password"/><br/>
  성별 : <input type="radio" name="sex" value="남성"/>남성
  <input type="radio" name="sex" value="여성"/>여성
  <br/>
  관심사항 : <input type="checkbox" name="favor" value="정치"/>정치
  <input type="checkbox" name="favor" value="경제"/>경제
  <br/>
  직업
  <select name="job">
    <option>학생
    <option>주부
    <option>회사원
  </select>
  <br/>
  자기소개
  <textarea name="comments" rows="10" cols="35"></textarea>
  </textarea>
  <br/>
  <input type="submit" value="전송"/>
</form>
</body>
</html>
```

QueryServlet

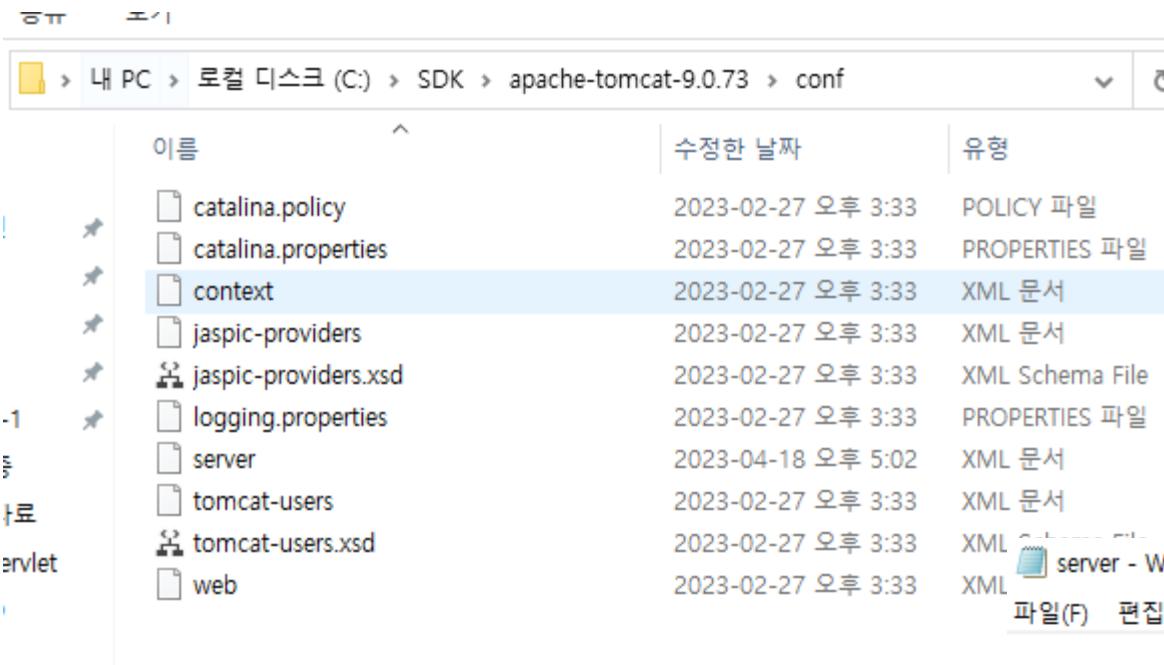


QueryServlet

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
request.setCharacterEncoding("euc-kr");
response.setContentType("text/html;charset=EUC-KR");

PrintWriter out = response.getWriter();
out.println("<h1>Query Test</h1>");
out.println("name :" + request.getParameter("name") + "<br/>");
out.println("password :" + request.getParameter("password") + "<br/>");
out.println("sex :" + request.getParameter("sex") + "<br/>");
out.println("관심사항 : ");
String favors[] = request.getParameterValues("favor");
for(String favor: favors) {
out.println(favor + ",");
}
out.println("<br/>");
out.println("job :" + request.getParameter("job") + "<br/>");
out.println("comments: " + request.getParameter("comments"));
}
```

Server.xml



이름	수정한 날짜	유형
catalina.policy	2023-02-27 오후 3:33	POLICY 파일
catalina.properties	2023-02-27 오후 3:33	PROPERTIES 파일
context	2023-02-27 오후 3:33	XML 문서
jaspic-providers	2023-02-27 오후 3:33	XML 문서
jaspic-providers.xsd	2023-02-27 오후 3:33	XML Schema File
logging.properties	2023-02-27 오후 3:33	PROPERTIES 파일
server	2023-04-18 오후 5:02	XML 문서
tomcat-users	2023-02-27 오후 3:33	XML 문서
tomcat-users.xsd	2023-02-27 오후 3:33	XML
web	2023-02-27 오후 3:33	XML

<!-- A "Connector" represents an endpoint by which requests are received and responses are returned. Documentation at :

Java HTTP Connector: </docs/config/http.html>

Java AJP Connector: </docs/config/ajp.html>

APR (HTTP/AJP) Connector: </docs/apr.html>

Define a non-SSL/TLS HTTP/1.1 Connector on port 8080

-->

```
<Connector URIEncoding = "utf-8" port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

<!-- A "Connector" using the shared thread pool-->

Query Test

name :김형오
password :1234
sex :남성
관심사항 : 정치,
job :주부
comments: 자기소개

The background of the slide features a glowing, translucent brain in shades of red and yellow, set against a dark background with floating binary digits (0s and 1s) and a network of glowing lines and dots, suggesting a digital or neural network environment.

Part 5,

Servlet 설정

참고문헌

Error_code HTML

error_code.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h1>This Page is Error Code</h1>
</body>
</html>
```

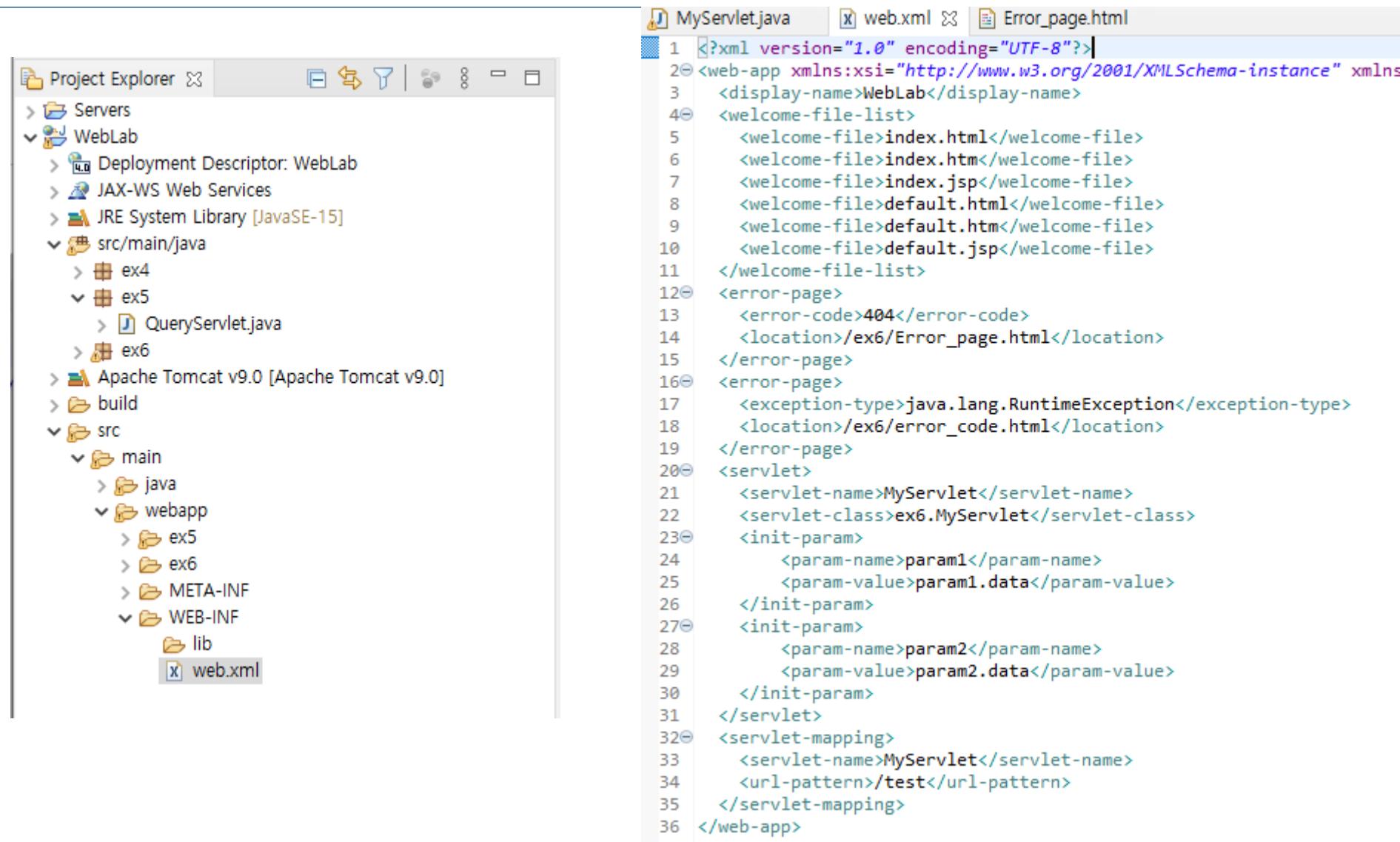
error_page.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h1>This Page is Error Page</h1>
</body>
</html>
```

MyServlet.java

```
MyServlet.java ✘
1 package ex6;
2
3+ import java.io.IOException;
9
10+ /**
11  * Servlet implementation class MyServlet
12  */
13 // @WebServlet("/MyServlet")
14 public class MyServlet extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17+ /**
18     * @see HttpServlet#HttpServlet()
19     */
20+ public MyServlet() {
21     super();
22     // TODO Auto-generated constructor stub
23 }
24
25+ /**
26     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
27     */
28+ protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29     String data=null;
30     data.indexOf(0);
31 }
32
33+ /**
34     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
35     */
36+ protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37     // TODO Auto-generated method stub
38     doGet(request, response);
39 }
40
41 }
42 }
```

Web.xml

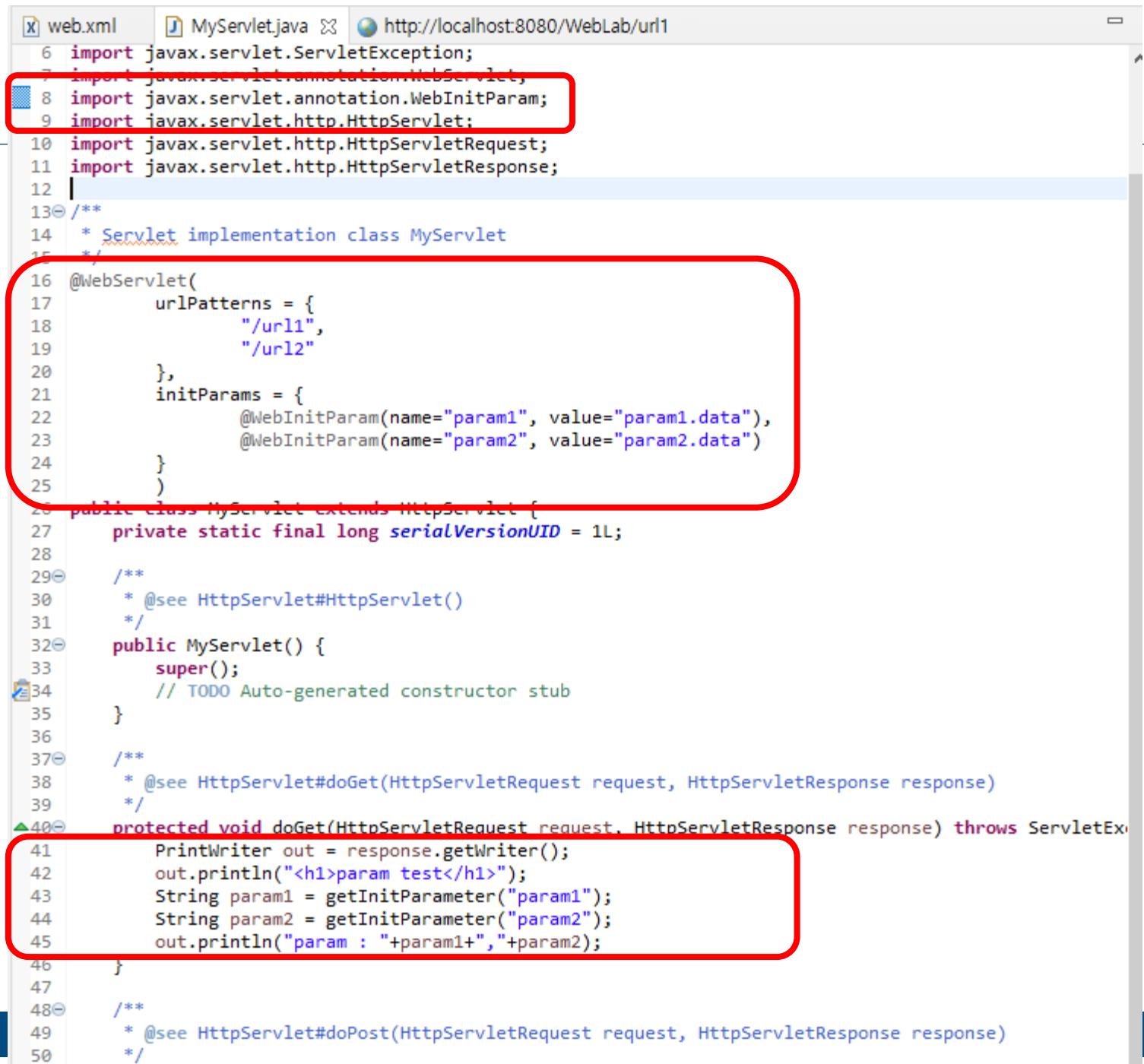


The screenshot shows a Java development environment with the following structure:

- Project Explorer:** Lists the project structure. It includes a **Servers** node, a **WebLab** node which contains a **Deployment Descriptor: WebLab**, **JAX-WS Web Services**, **JRE System Library [JavaSE-15]**, and a **src/main/java** node. **src/main/java** contains **ex4**, **ex5** (which contains **QueryServlet.java**), **ex6**, and **Apache Tomcat v9.0 [Apache Tomcat v9.0]**. It also includes a **build** node and a **src** node. The **src** node contains a **main** node, which has **java** and **webapp** sub-nodes. **webapp** contains **ex5**, **ex6**, **META-INF**, and **WEB-INF**. **WEB-INF** contains **lib** and the **web.xml** file, which is currently selected.
- Code Editor:** Displays the content of the **web.xml** file. The code is as follows:

```
<?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>WebLab</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
  <error-page>
    <error-code>404</error-code>
    <location>/ex6/Error_page.html</location>
  </error-page>
  <error-page>
    <exception-type>java.lang.RuntimeException</exception-type>
    <location>/ex6/error_code.html</location>
  </error-page>
  <servlet>
    <servlet-name>MyServlet</servlet-name>
    <servlet-class>ex6.MyServlet</servlet-class>
    <init-param>
      <param-name>param1</param-name>
      <param-value>param1.data</param-value>
    </init-param>
    <init-param>
      <param-name>param2</param-name>
      <param-value>param2.data</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>MyServlet</servlet-name>
    <url-pattern>/test</url-pattern>
  </servlet-mapping>
</web-app>
```

Annotation



```
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.annotation.WebInitParam;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 |
13 /**
14  * Servlet implementation class MyServlet
15 */
16 @WebServlet(
17     urlPatterns = {
18         "/url1",
19         "/url2"
20     },
21     initParams = {
22         @WebInitParam(name="param1", value="param1.data"),
23         @WebInitParam(name="param2", value="param2.data")
24     }
25 )
26 public class MyServlet extends HttpServlet {
27     private static final long serialVersionUID = 1L;
28
29 /**
30  * @see HttpServlet#HttpServlet()
31  */
32 public MyServlet() {
33     super();
34     // TODO Auto-generated constructor stub
35 }
36
37 /**
38  * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
39  */
40 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
41     PrintWriter out = response.getWriter();
42     out.println("<h1>param test</h1>");
43     String param1 = getInitParameter("param1");
44     String param2 = getInitParameter("param2");
45     out.println("param : "+param1+","+param2);
46 }
47
48 /**
49  * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
50 */
}
```



경청해주셔서 감사합니다.