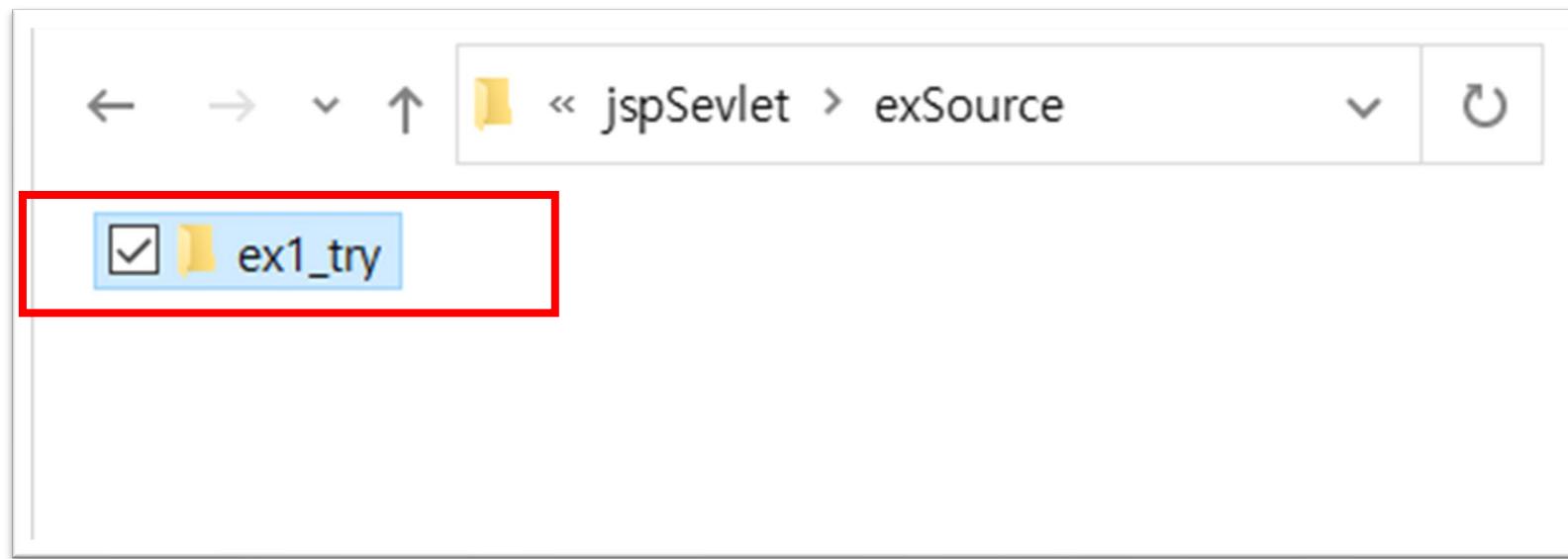


Try~ Java Web

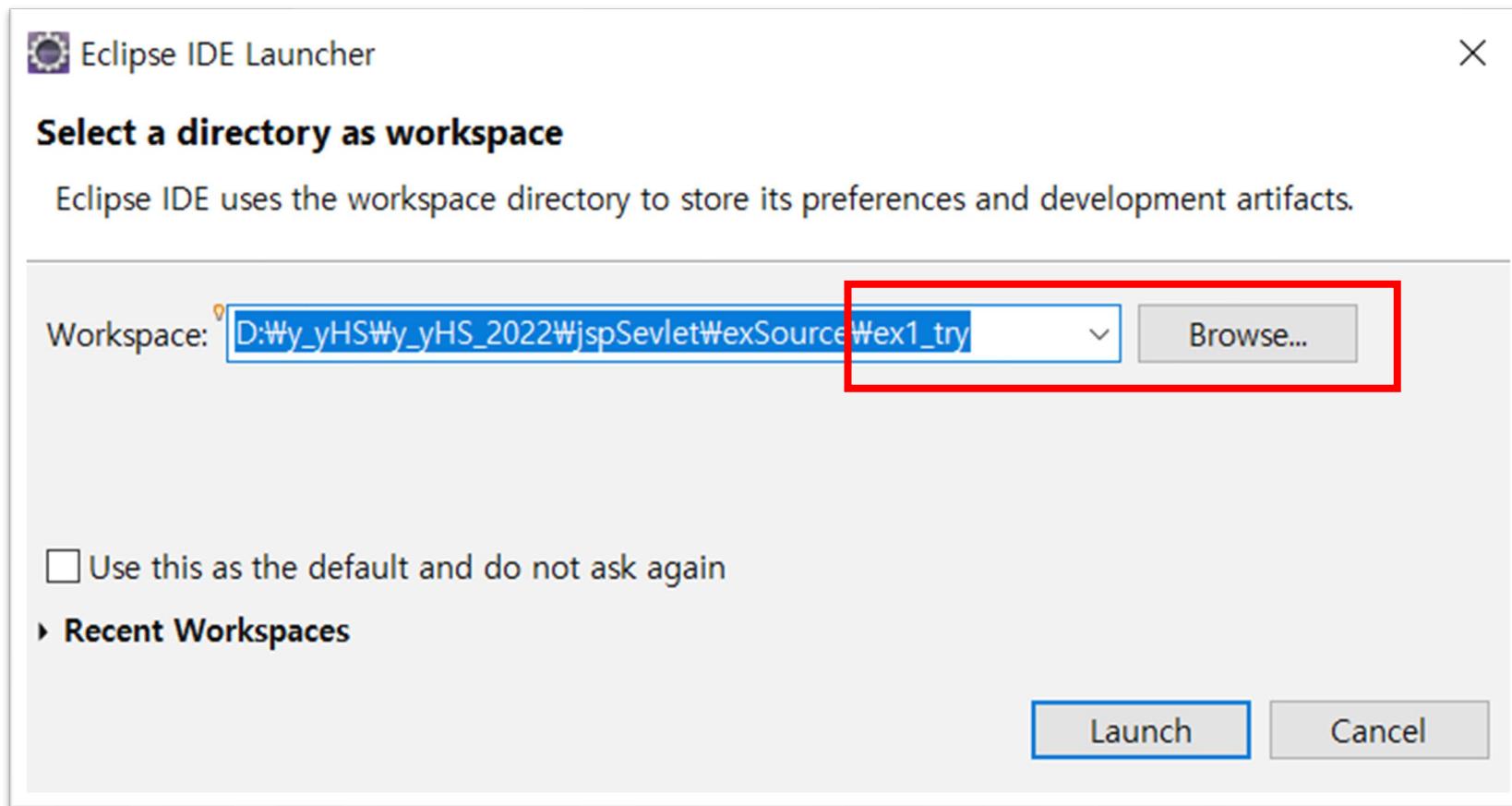
- A. 프로젝트 폴더 설정
- B. 이클립스에 WAS설정
- B_2. Tomcat 포트번호 변경
- C. Dynamic프로젝트 시작
- C_2. 이클립스 한글설정_UTF-8
- C_3. jsp페이지 작성 및 실행
- C_4. sevlet페이지 작성
- C_5. xml 매팅과 실행

교사 최선경

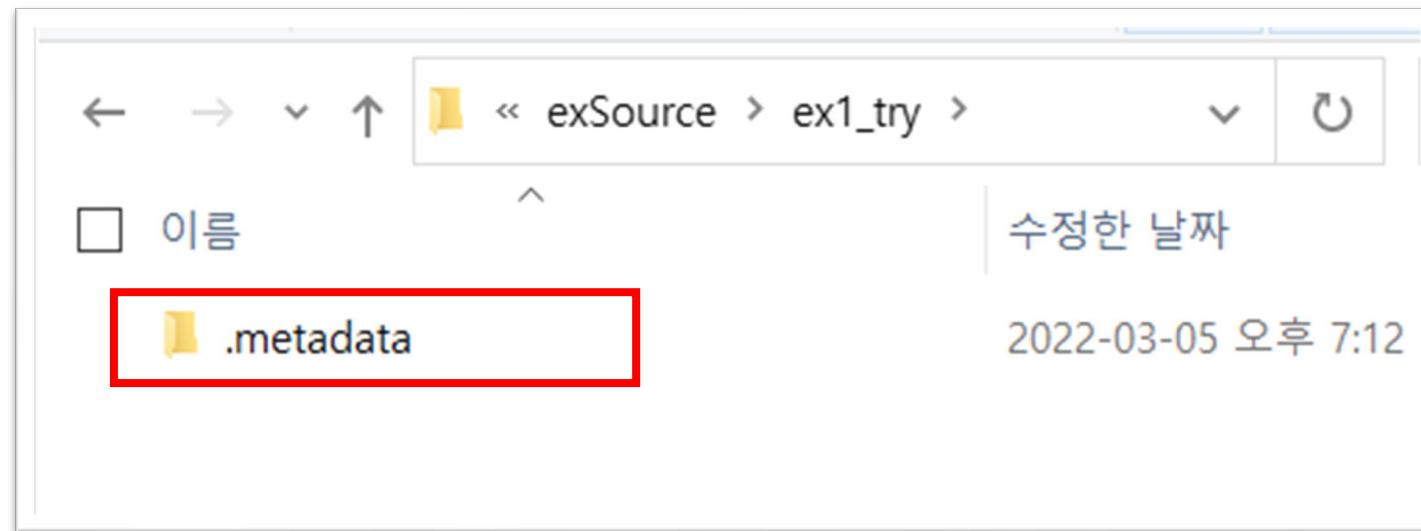
A. 프로젝트 폴더 생성



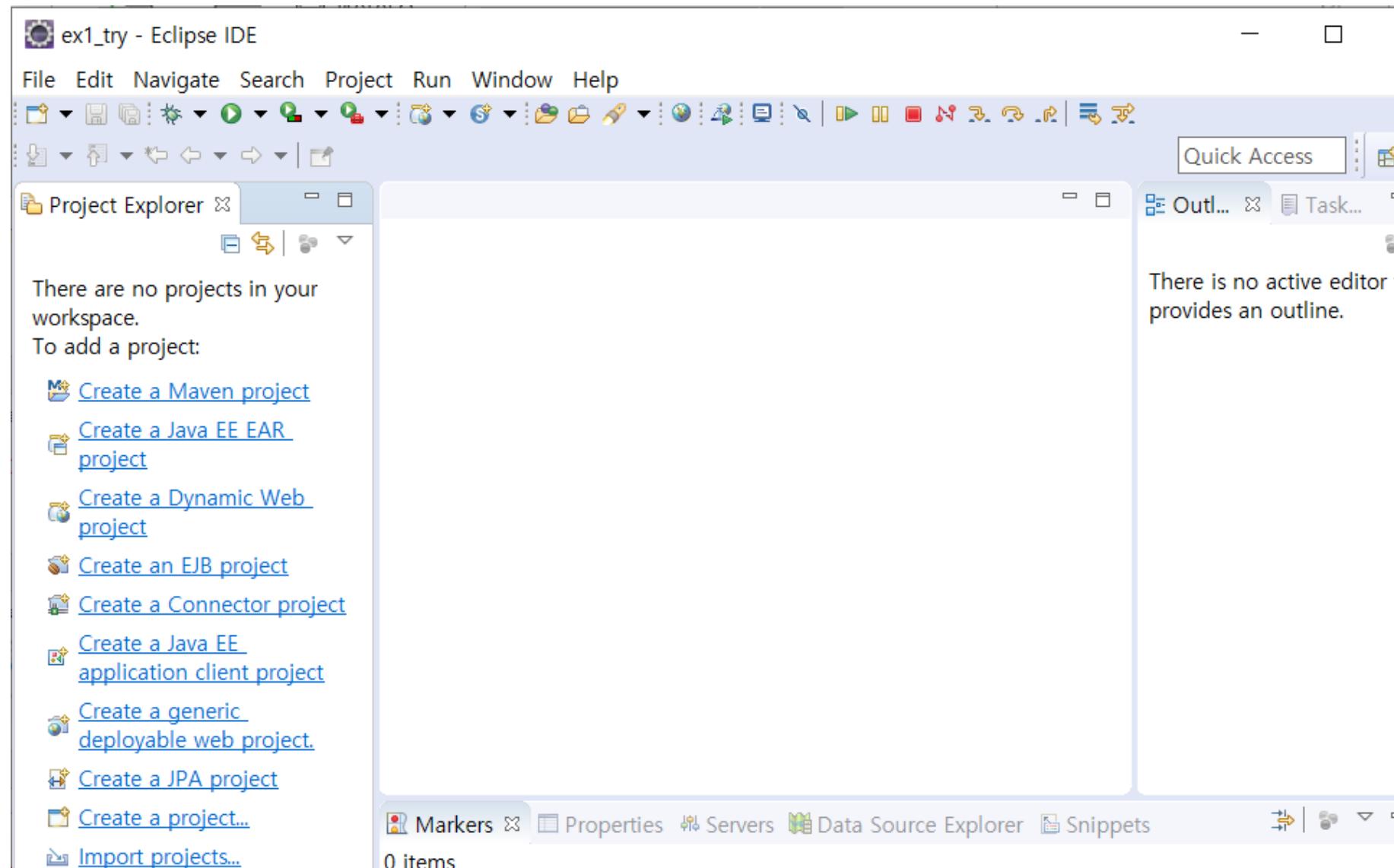
경로에 맞춰서 이클립스 실행



프로젝트폴더안의 환경파일을 확인한다.

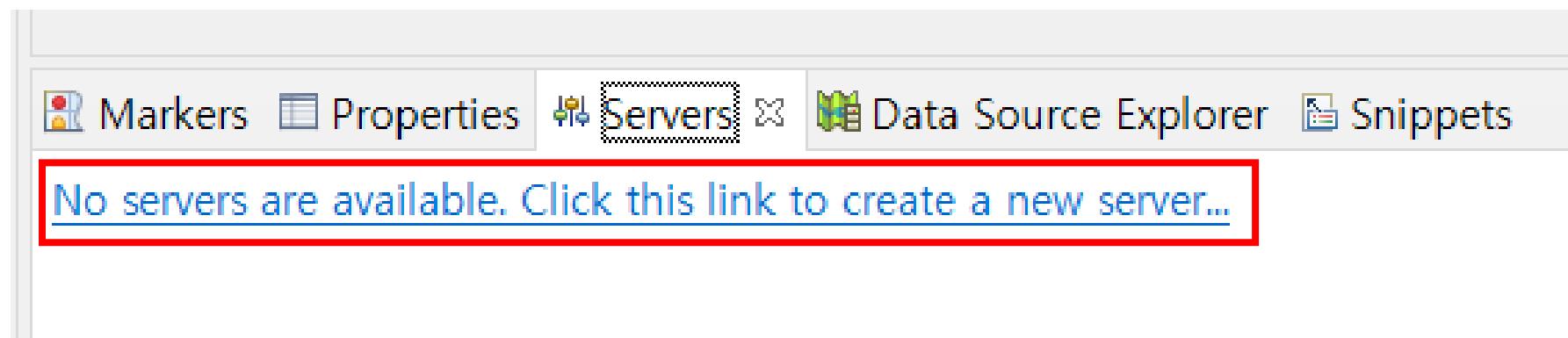


이클립스 OPEN.

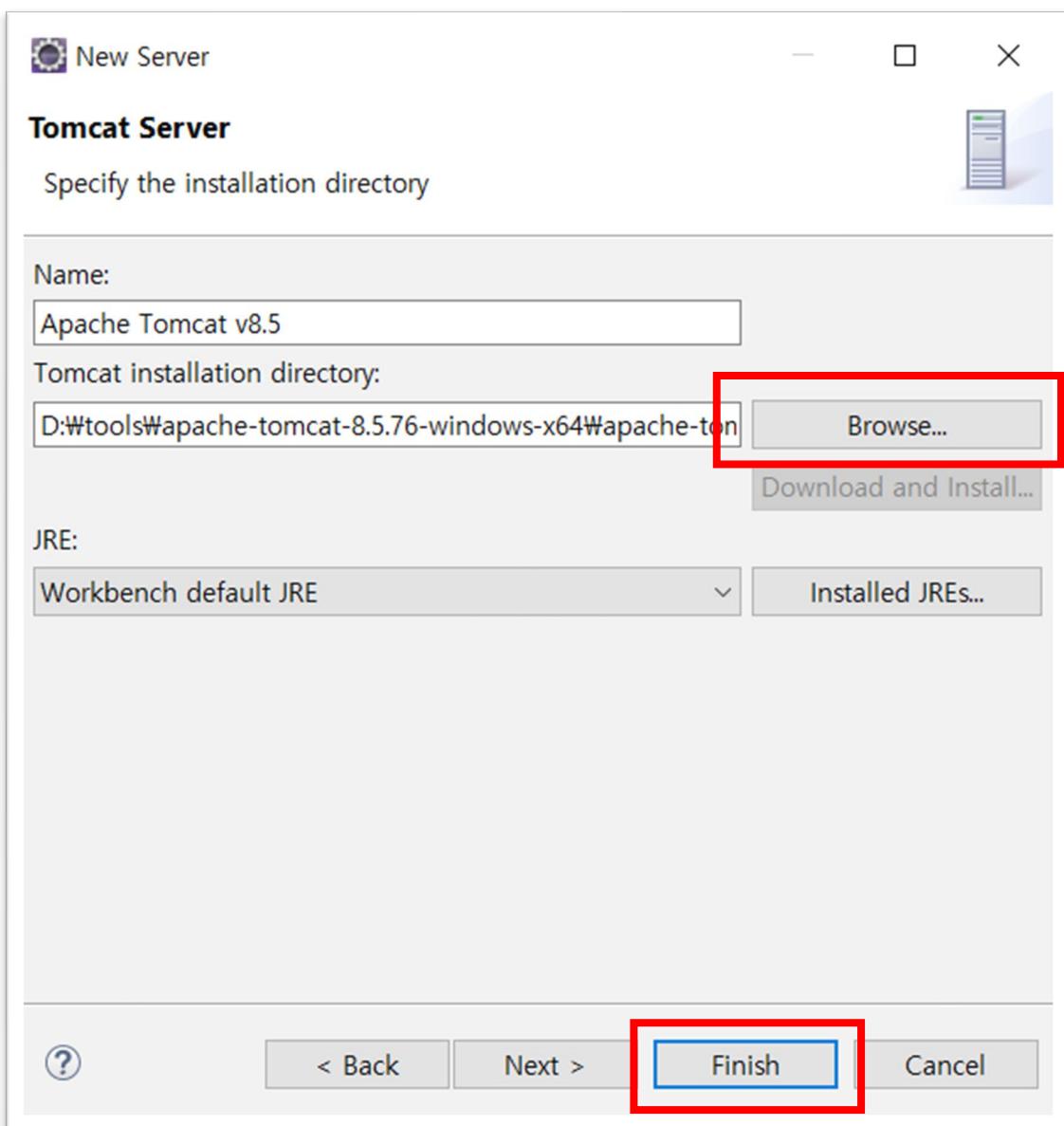
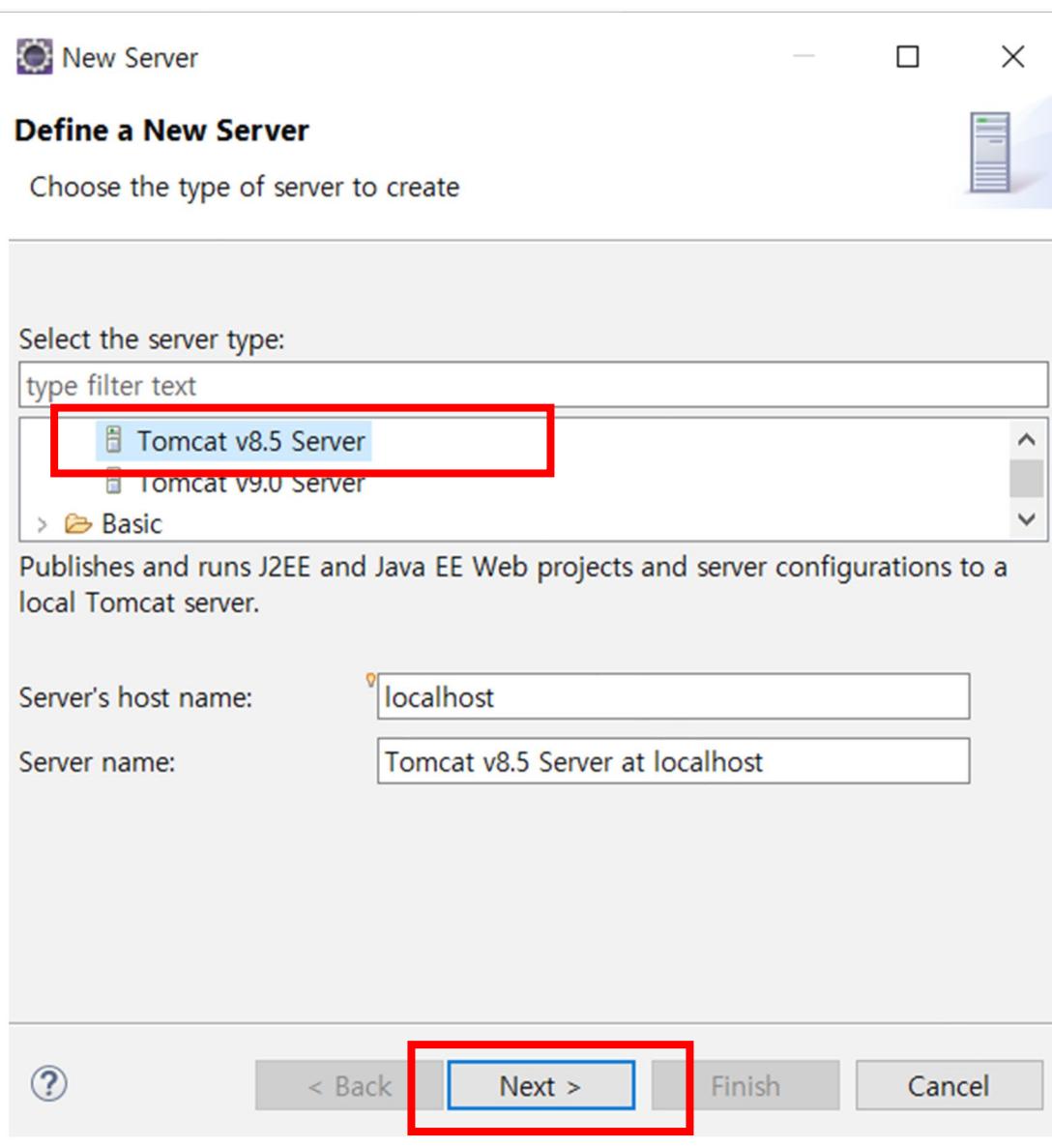


B. 자바웹을 개발하기 위한 WAS설정.

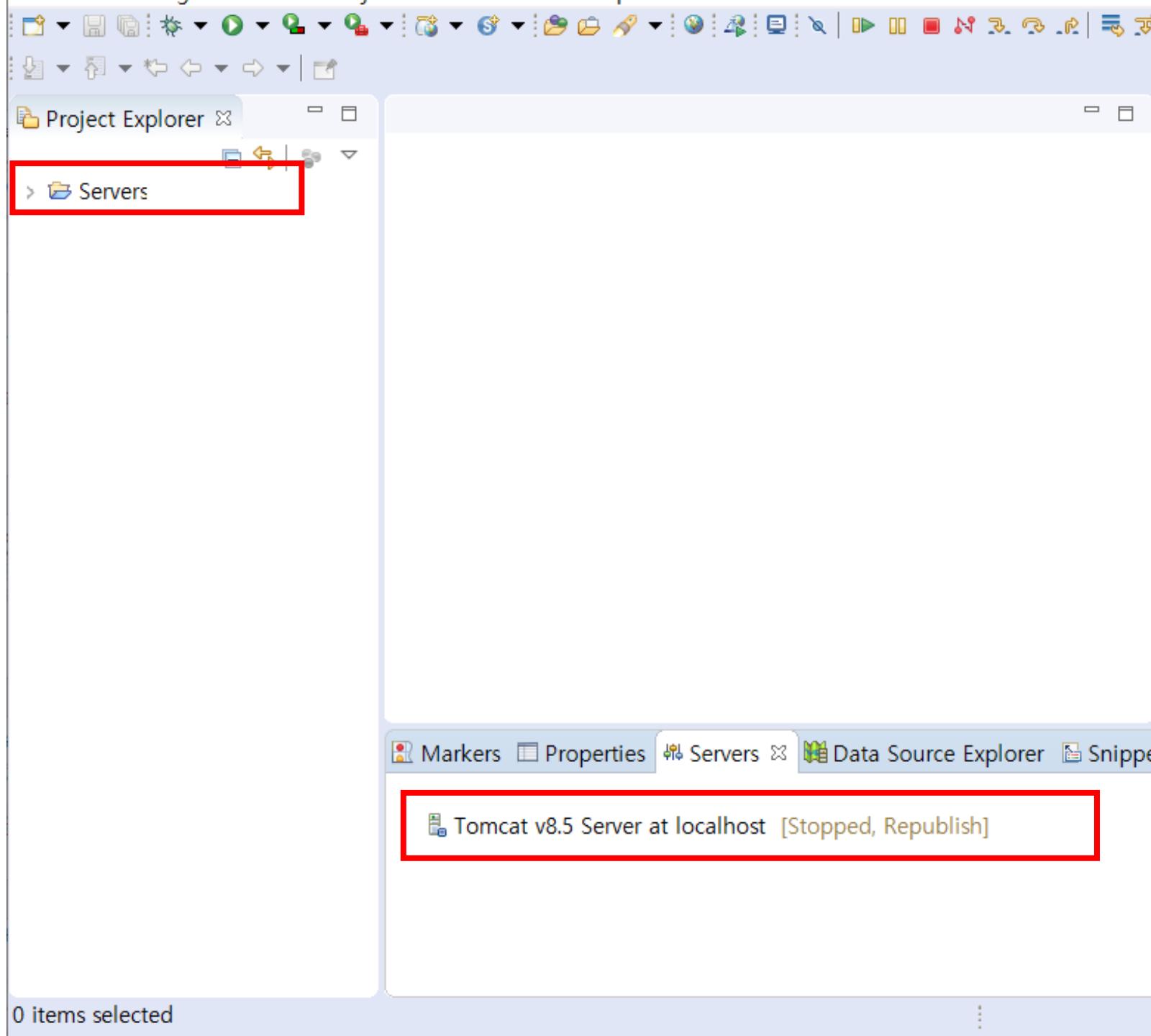
1. WebApplicationServer 설치 _ ex. Tomcat9
2. 이클립스에 Tomcat을 서버로 연결



이클립스 하단 Servers Tomcat v8.5



이클립스에 서버설정 확인



B_2. Tomcat 포트 번호변경_충돌방지

Tomcat v8.5 Server at localhost

Overview

General Information
Specify the host name and other common settings.

Server name: Tomcat v8.5 Server at localhost
Host name: localhost
Runtime Environment: Apache Tomcat v8.5
Configuration path: /Servers/Tomcat v8.5 Server at localhost
[Open launch configuration](#)

Server Locations
Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.
 Use workspace metadata (does not modify Tomcat installation)

Publishing

Timeouts

Ports
Modify the server ports.

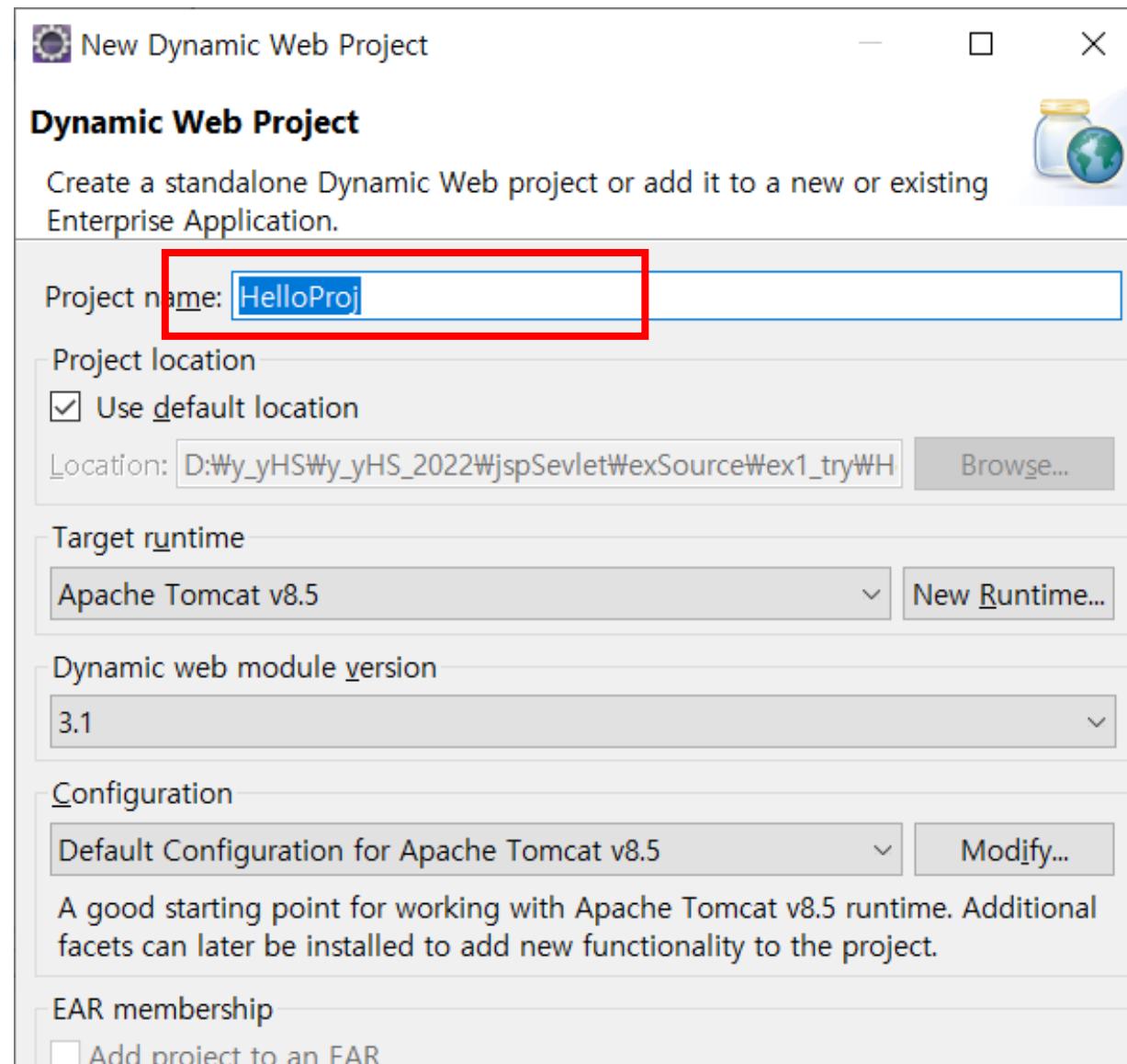
Port Name	Port Number
Tomcat admin port	8005
HTTP/1.1	8090

MIME Mappings

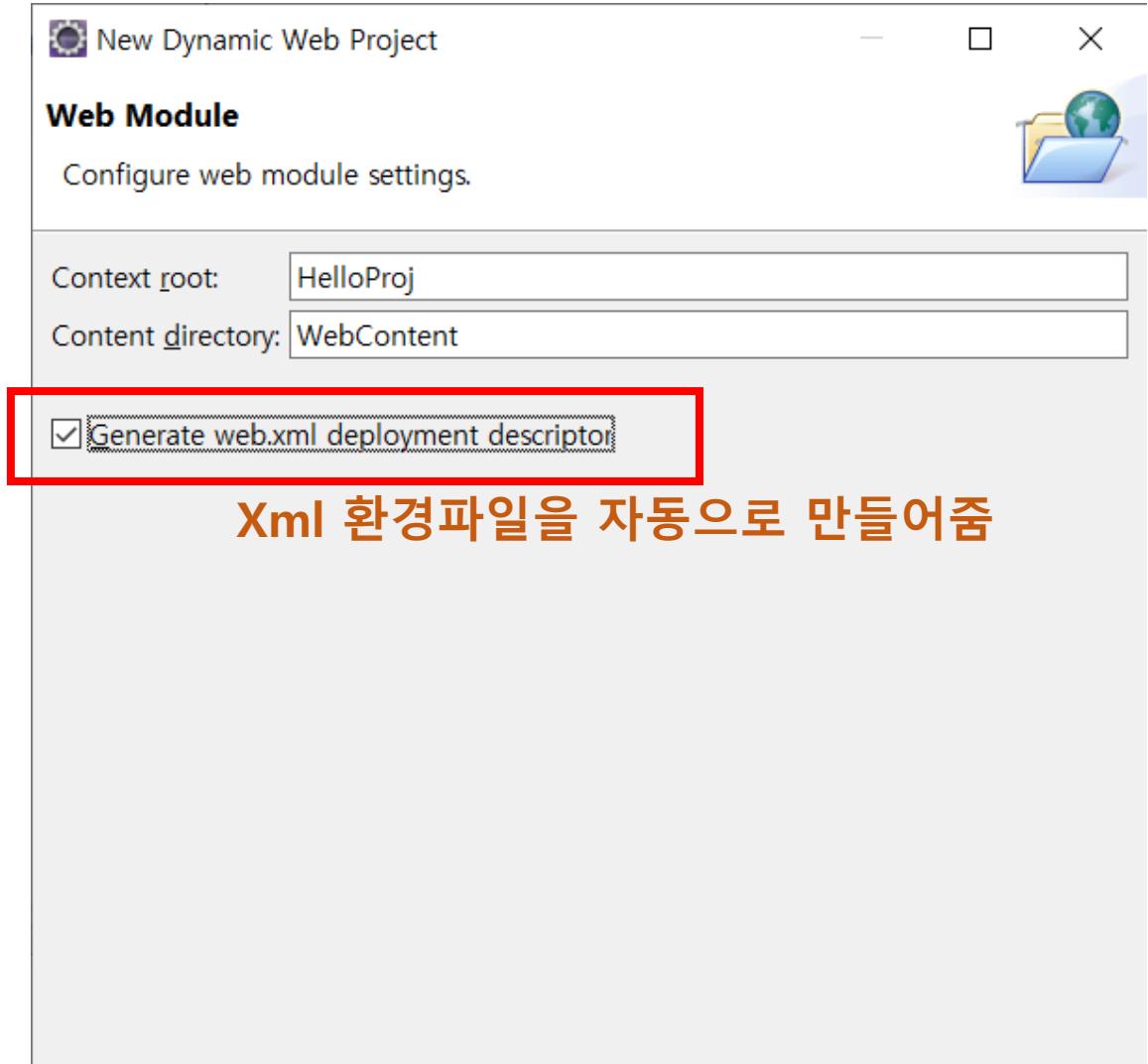
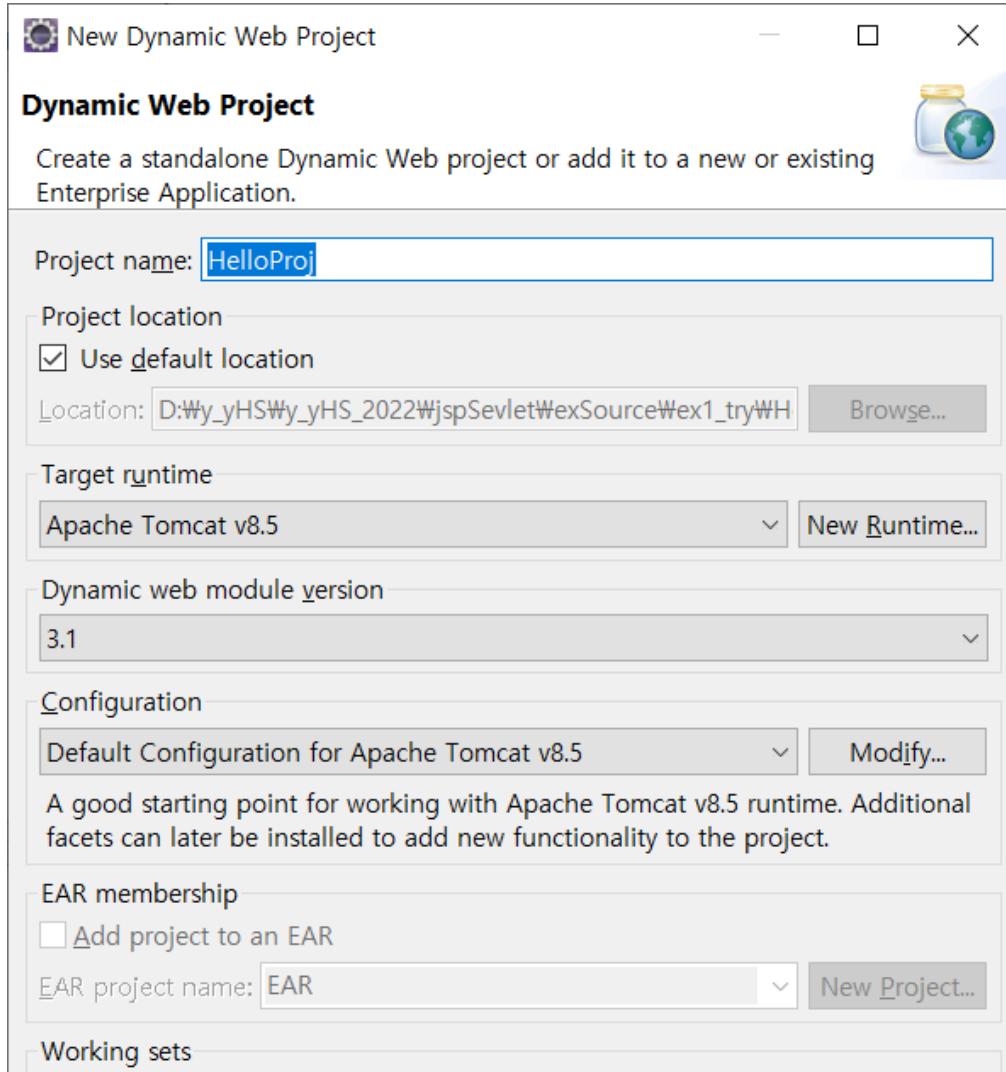
Markers Properties Servers Data Source Explorer Snippets

Tomcat v8.5 Server at localhost [Stopped, Republish]

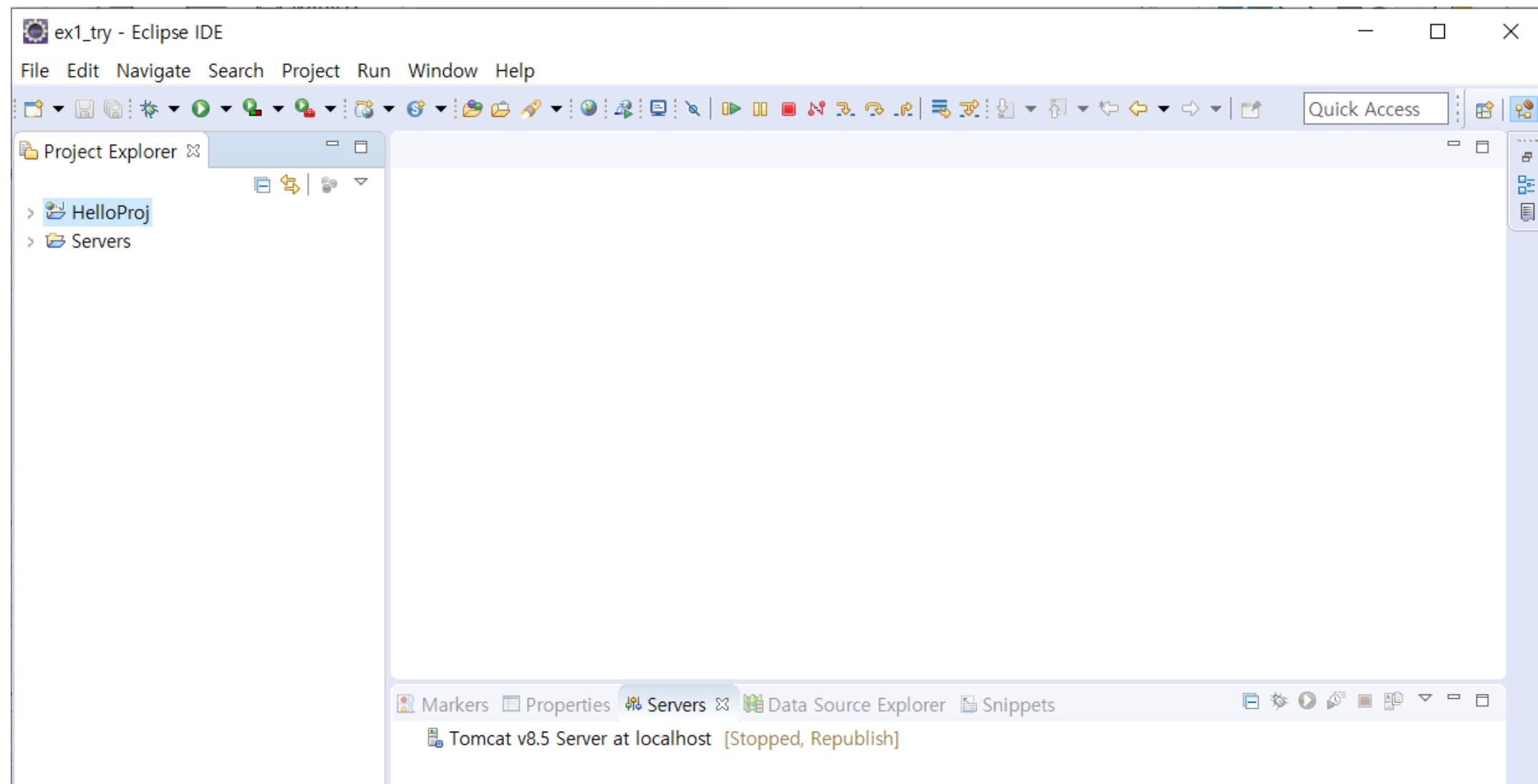
Dynamic Web Project 생성



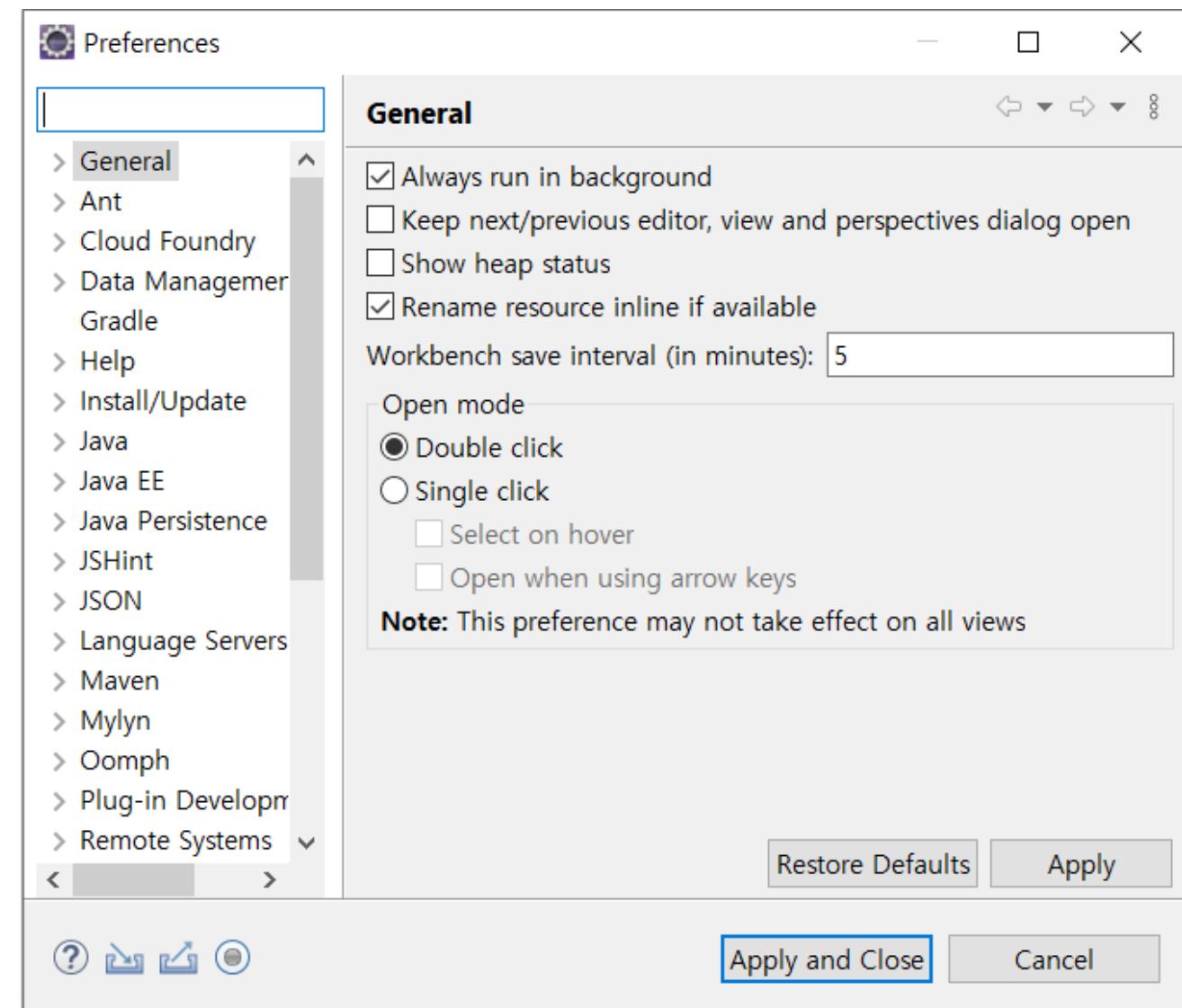
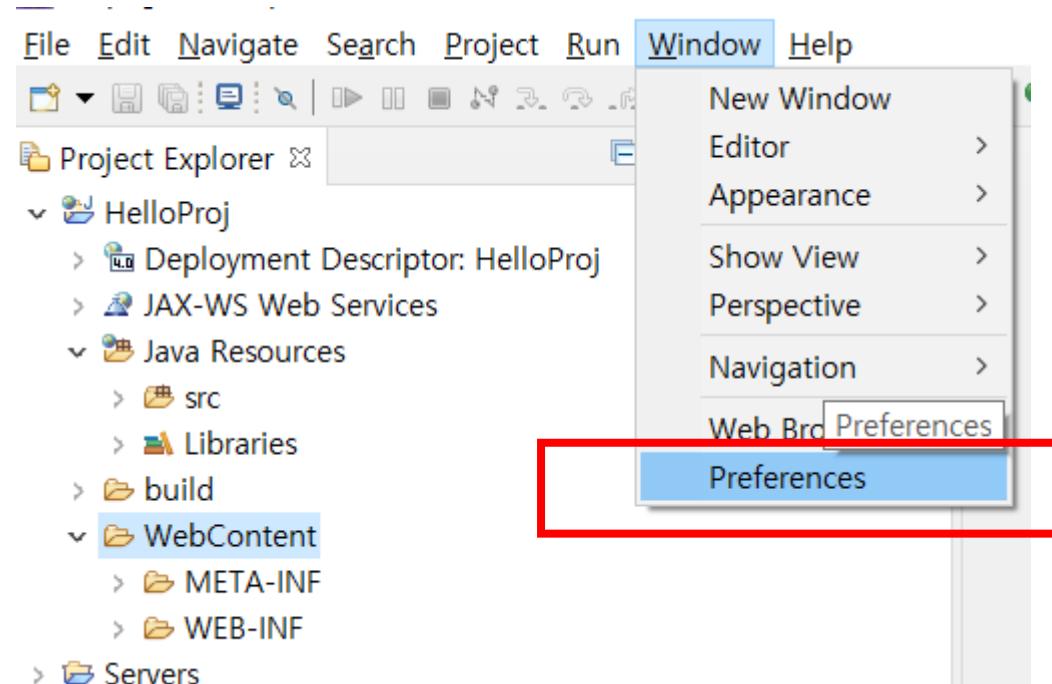
Next버튼-> Web Module에서 체크



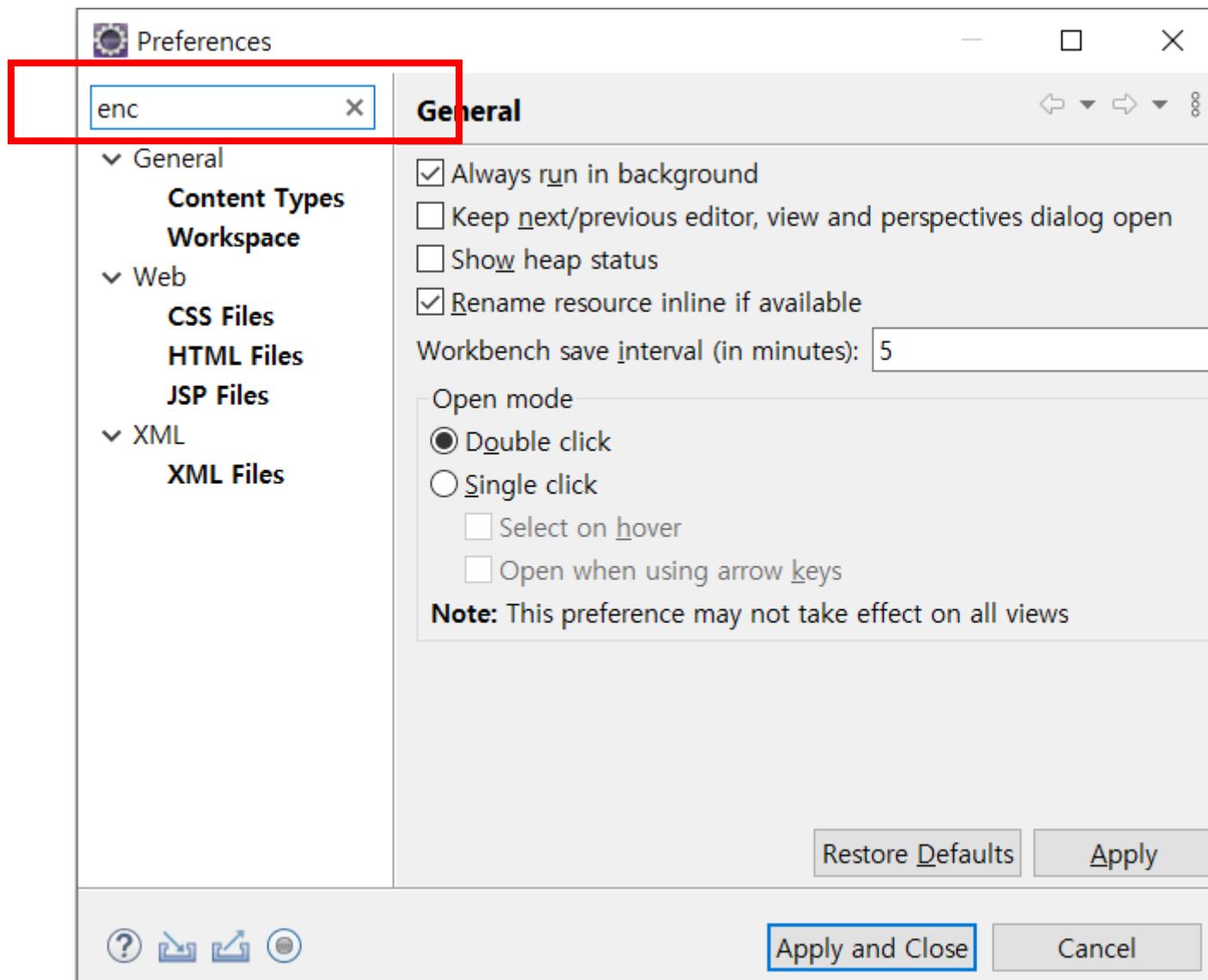
C. Dynamic 프로젝트 시작.



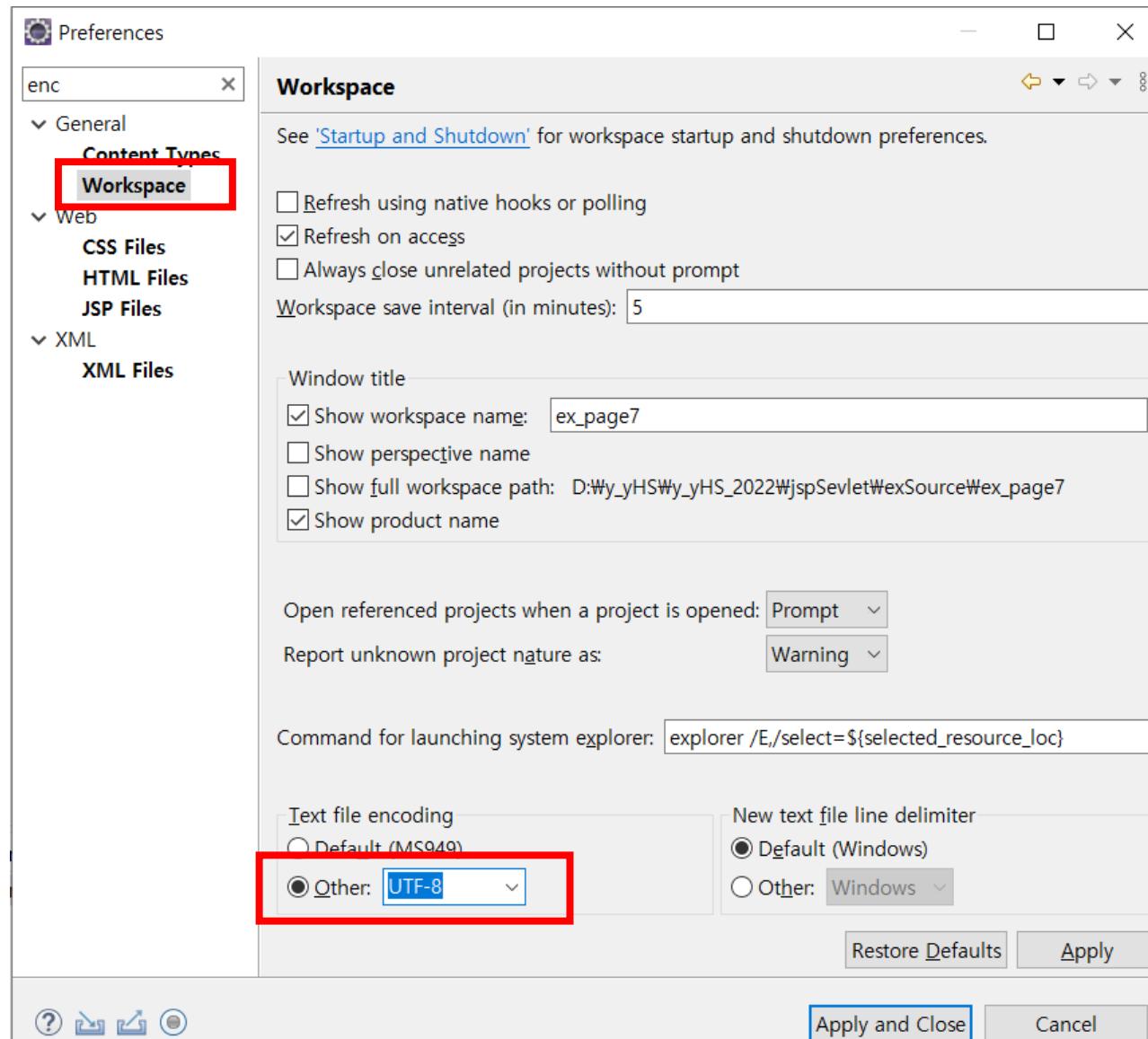
C_2. 이클립스 한글설정



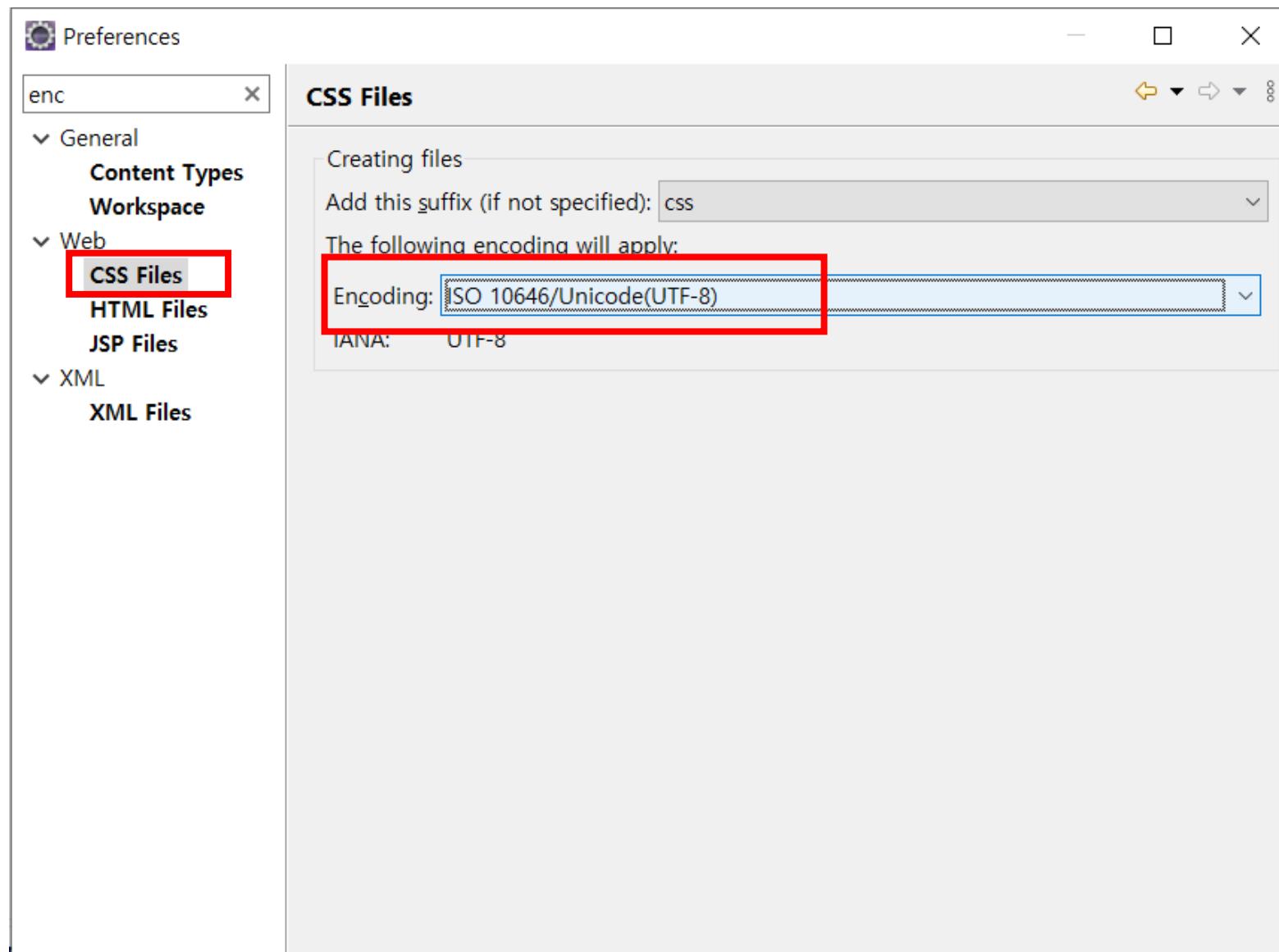
검색란에 enc 입력



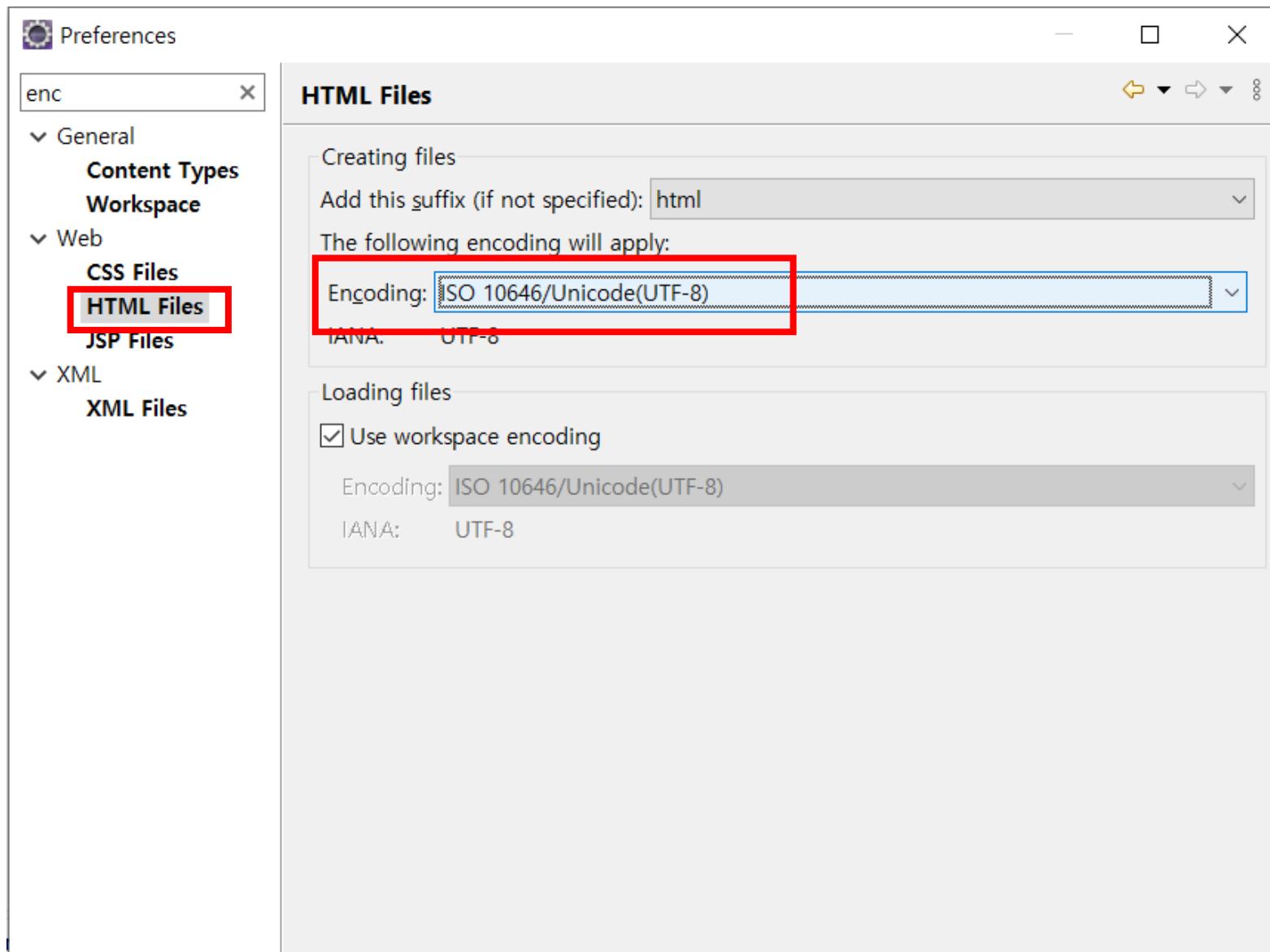
코딩영역 한글깨짐방지.



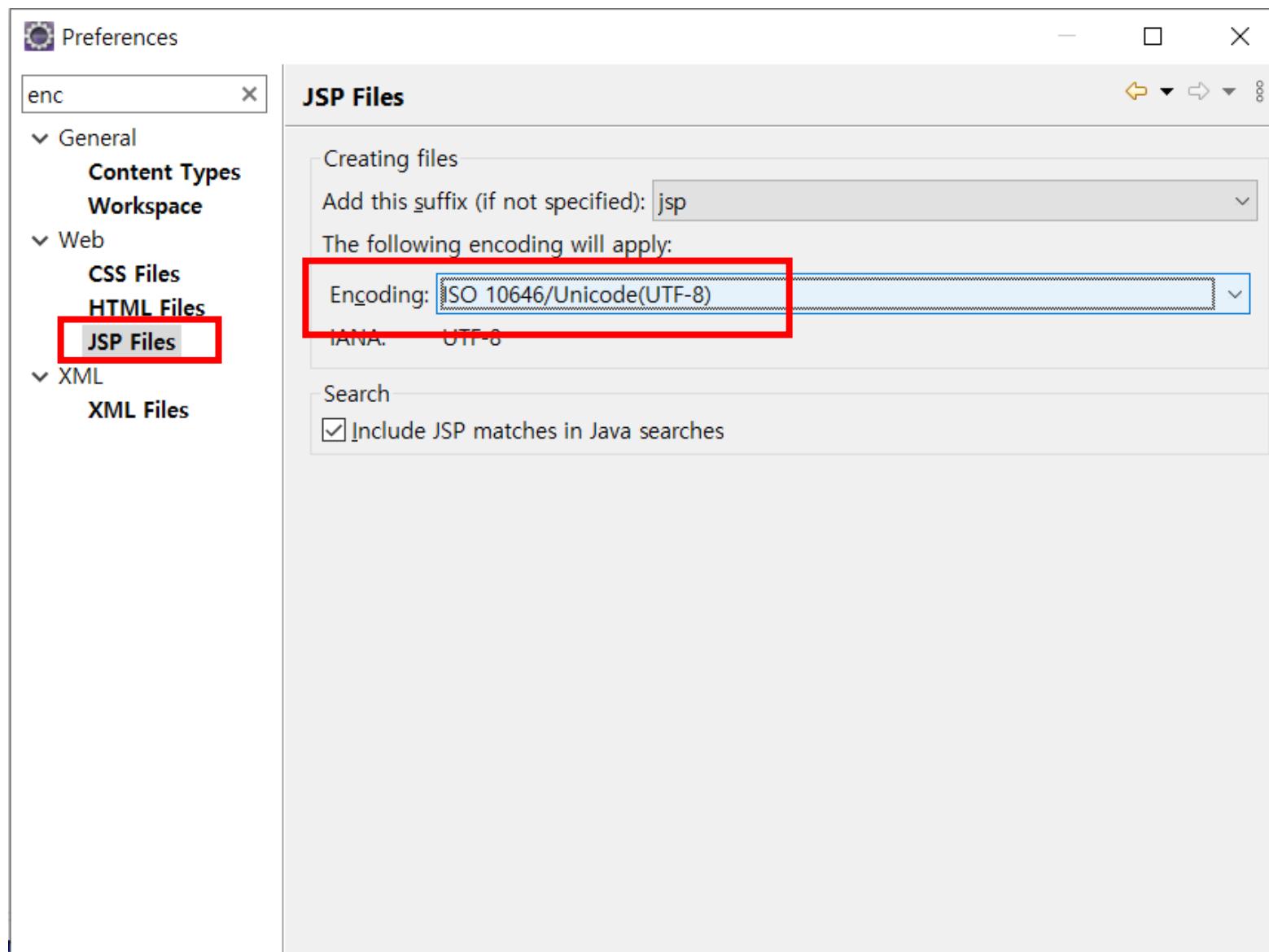
스타일시트 한글깨짐방지.



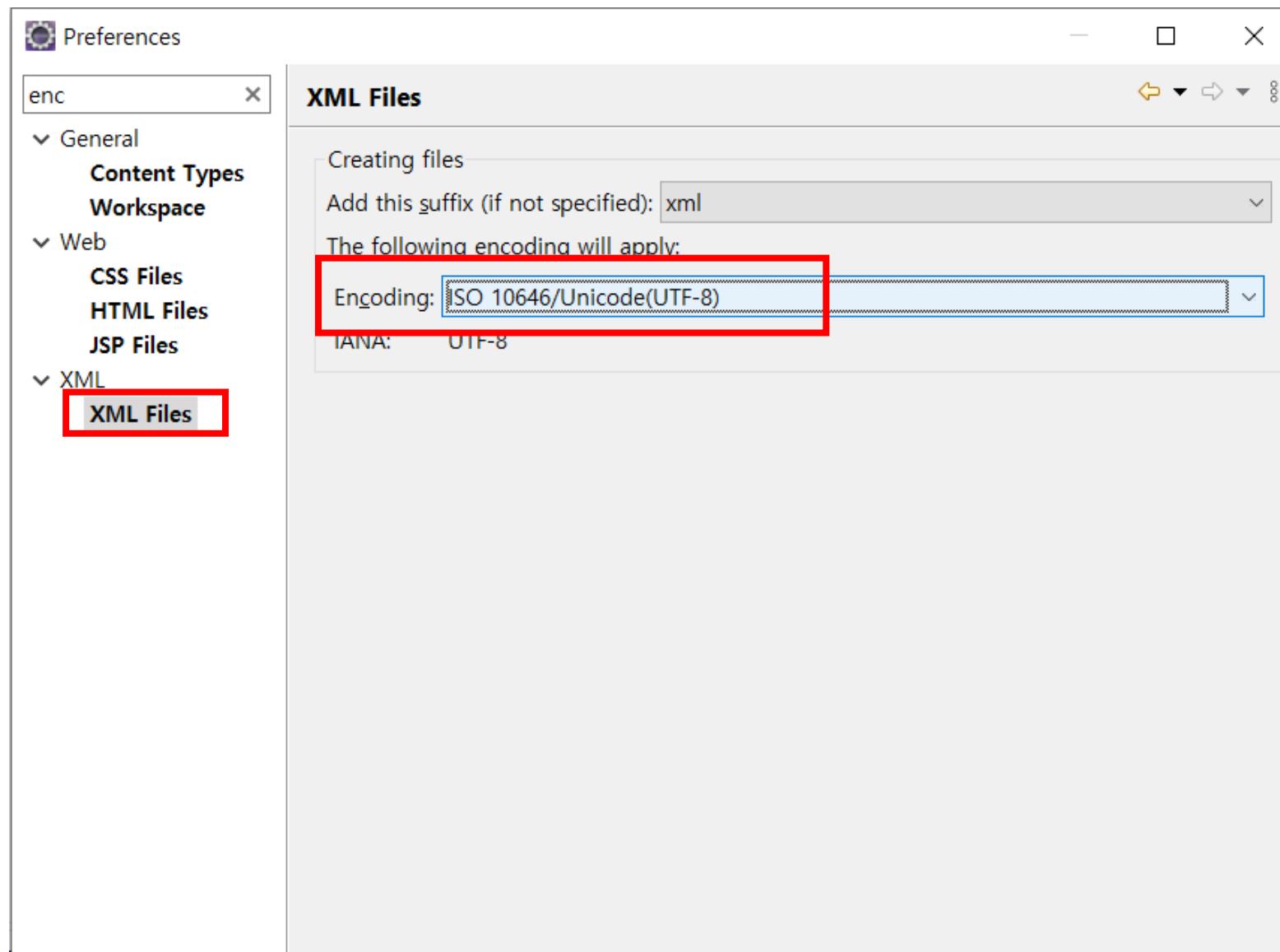
html 파일 한글깨짐방지.



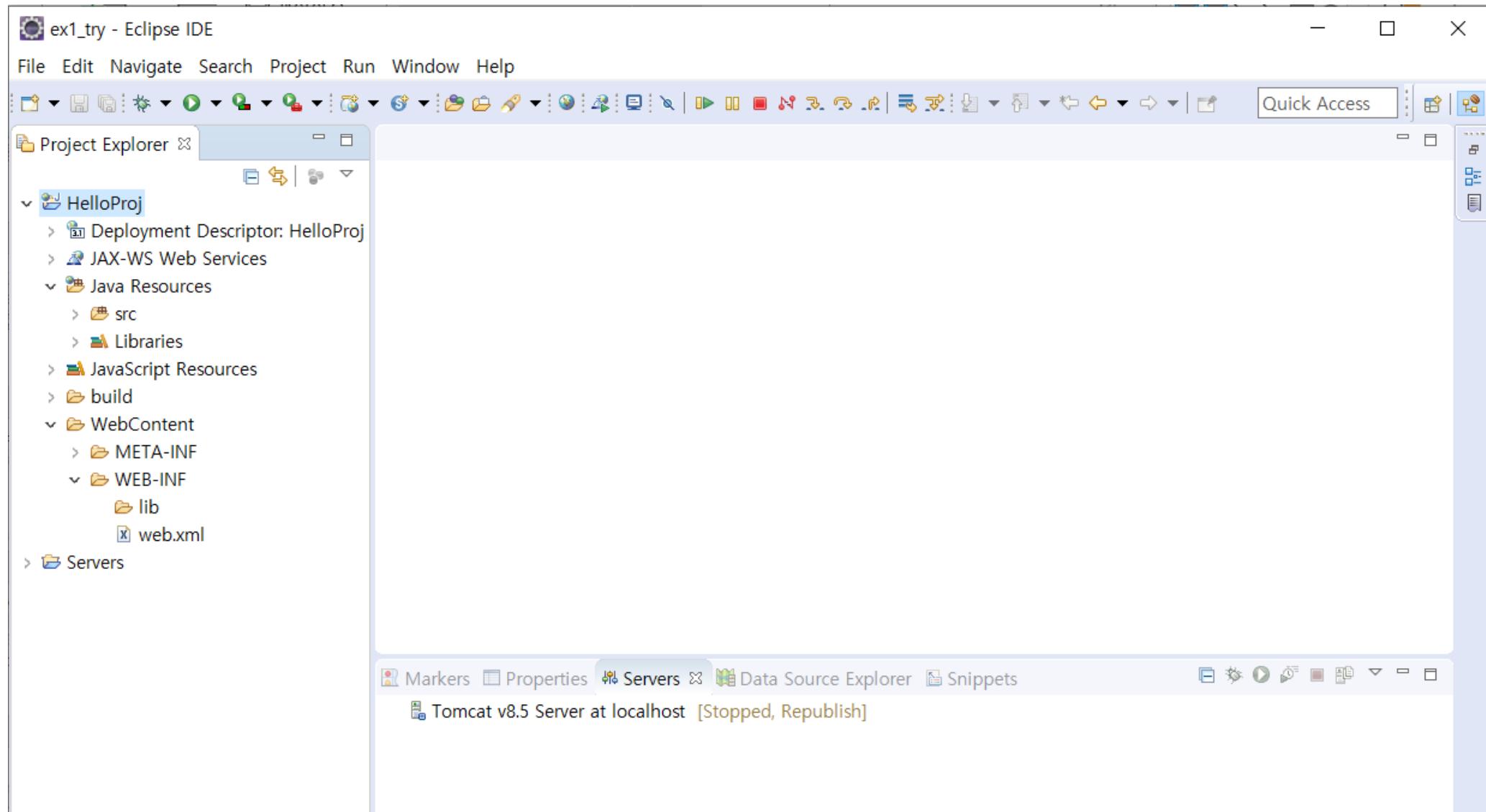
Jsp파일 한글깨짐방지



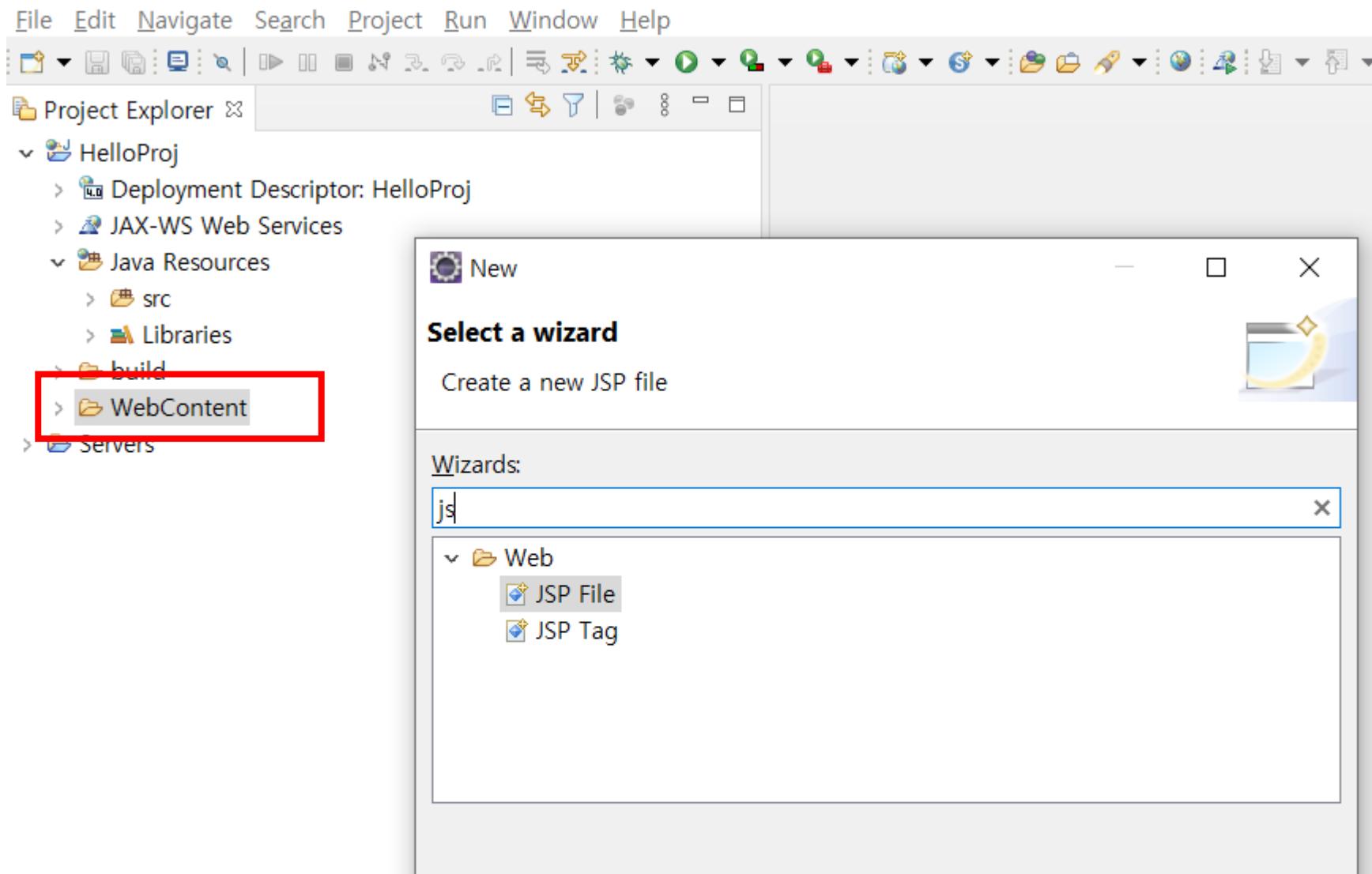
xml 파일 한글깨짐 방지



C. Dynamic 프로젝트 작성 시작



C_3. WebContent에 hello.jsp 생성



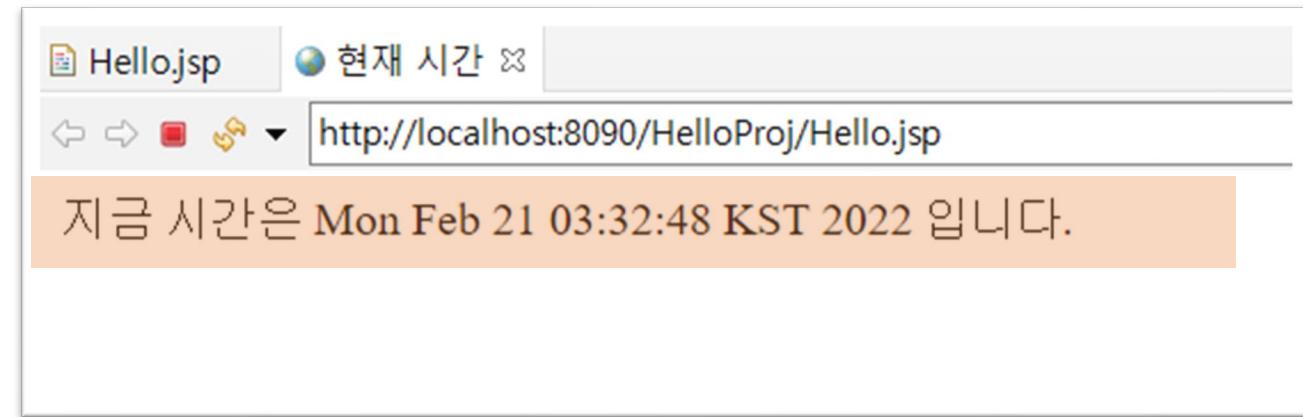
Hello.jsp

The screenshot shows a Java IDE interface with the following details:

- Menu Bar:** File, Edit, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard icons for file operations like Open, Save, Print, Find, etc.
- Project Explorer:** Shows the project structure:
 - HelloProj
 - Deployment Descriptor: HelloProj
 - JAX-WS Web Services
 - Java Resources
 - src
 - Libraries
 - build
 - WebContent
 - META-INF
 - WEB-INF
 - Hello.jsp
 - Servers
- Code Editor:** Displays the content of the Hello.jsp file.

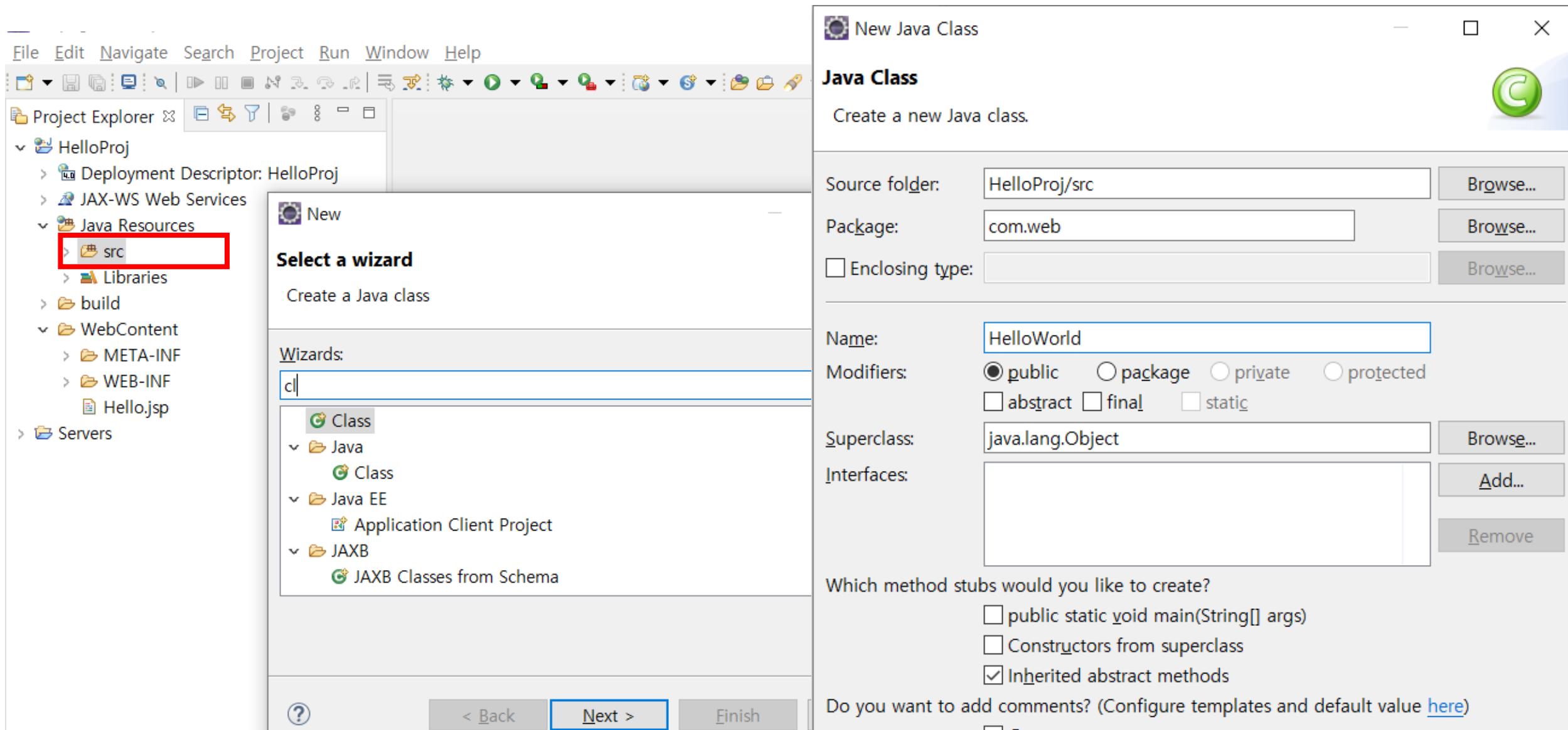
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Hello.jsp 작성 및 실행



```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>현재 시간</title>
8 </head>
9 <body>
10    지금 시간은 <%= new java.util.Date() %> 입니다.
11 </body>
12 </html>
```

C_4. servlet 페이지 작성



HelloWorld.java

The screenshot shows a Java development environment with the following interface elements:

- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard icons for file operations, search, and navigation.
- Project Explorer:** Shows the project structure:
 - HelloProj (selected)
 - Deployment Descriptor: HelloProj
 - JAX-WS Web Services
 - Java Resources
 - src
 - com.web
 - HelloWorld.java (highlighted with a red box)
 - Libraries
 - build
 - WebContent
 - META-INF
 - WEB-INF
 - Hello.jsp
 - Servers
- Code Editor:** HelloWorld.java (highlighted with a blue bar)

```
1 package com.web;
2
3 public class HelloWorld {
4
5 }
6
```

File Edit Source Refactor Navigate Search Project Run Window Help



Project Explorer X

- >HelloProj
 - > Deployment Descriptor: HelloProj
 - > JAX-WS Web Services
 - > Java Resources
 - > src
 - > com.web
 - > HelloWorld.java
 - > Libraries
 - > build
 - > WebContent
 - > META-INF
 - > WEB-INF
 - > Hello.jsp
- > Servers

HelloWorld.java X

```
1 package com.web;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 public class HelloWorld extends HttpServlet {
12
13     @Override
14     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
15
16         resp.setContentType("text/html");
17         PrintWriter out = resp.getWriter();
18         out.println("<html>");
19         out.println("<body>");
20         out.println("Hello Java Server Page ~ !!!");
21         out.println("자바 웹페이지 안녕~ !!!");
22         out.println("</body></html>");
23         out.close();
24     }
25
26 }
```

HelloWorld 서블릿페이지 작성

HelloWorld 서블릿페이지 실행

The screenshot shows a web browser window with the following details:

- Tab: HelloWorld.java (selected)
- Tab: HTTP 상태 404 – 찾을 수 없음
- Address bar: http://localhost:8090/HelloProj/servlet/com.web.HelloWorld

HTTP 상태 404 – 찾을 수 없음

타입 상태 보고

메시지 요청된 리소스 [/HelloProj/servlet/com.web.HelloWorld]은(는) 가용하지 않습니다.

설명 Origin 서버가 대상 리소스를 위한 현재의 representation을 찾지 못했거나, 그것이 존재하는지를 밝히려 하지 않습니다.

Apache Tomcat/9.0.55

C_4. 서블릿 페이지의 실행을 설정

The screenshot shows a Java IDE interface. The Project Explorer on the left lists a project named 'HelloProj' containing files like 'Deployment Descriptor: HelloProj', 'JAX-WS Web Services', 'Java Resources' (with 'src' and 'com.web' subfolders), 'Libraries', 'build', 'WebContent' (with 'META-INF' and 'WEB-INF' subfolders), 'lib', and 'web.xml'. The 'web.xml' file is highlighted with a red border. The code editor on the right displays the 'web.xml' configuration file:

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

페이지를 열어보자~

수정은 다음장에 기록...->

Web.xml 수정하기

The screenshot shows a Java development environment with the following interface elements:

- Project Explorer:** Shows the project structure of "HelloProj". It includes a "Deployment Descriptor: HelloProj", "JAX-WS Web Services", "Java Resources" folder containing a "src" folder with "com.web" package and "HelloWorld.java", "Libraries", and "build" folder.
- Editor:** Displays the content of the "web.xml" file. The code is highlighted with line numbers and syntax coloring.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-1
3     <display-name>HelloProj</display-name>
4
5     <servlet>
6         <servlet-name>HelloServlet</servlet-name>
7         <servlet-class>com.web.HelloWorld</servlet-class>
8     </servlet>
9
10    <servlet-mapping>
11        <servlet-name>HelloServlet</servlet-name>
12        <url-pattern>/Hello</url-pattern>
13    </servlet-mapping>
14
15 </web-app>
```

```
1 package com.web;
2
3 import java.io.IOException;
4
5
6 public class HelloWorld extends HttpServlet {
7
8     @Override
9     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
10
11         //resp.setContentType("text/html;charset=UTF-8");
12         resp.setContentType("text/html");
13         PrintWriter out = resp.getWriter();
14         out.println("<html>");
15         out.println("<body>");
16         out.println("Hello Java Server Page ~ !!!");
17         out.println("자바 웹페이지 안녕~ !!!");
18         out.println("</body></html>");
19         out.close();
20
21     }
22
23 }
24
25
26
27 }
28
```

서블릿 작성 및 실행 1

http://localhost:8090/HelloProj>Hello

Hello Java Server Page ~ !!! ?? ???? ??~ !!!

HelloWorld.java

```
1 package com.web;
2
3+import java.io.IOException;□
10
11 public class HelloWorld extends HttpServlet {
12
13@Override
14 protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
15
16     resp.setContentType("text/html;charset=UTF-8");
17     //resp.setContentType("text/html");
18     PrintWriter out = resp.getWriter();
19     out.println("<html>");
20     out.println("<body>");
21     out.println("Hello Java Server Page ~ !!!");
22     out.println("자바 웹페이지 안녕~ !!!");
23     out.println("</body></html>");
24     out.close();
25 }
26
27 }
28
```

서블릿 작성 및 실행 2

http://localhost:8090/HelloProj>Hello

The screenshot shows a browser window with the URL "http://localhost:8090/HelloProj>Hello". The page content is "Hello Java Server Page ~ !!! 자바 웹페이지 안녕~ !!!".

```
http://localhost:8090/HelloProj>Hello
http://localhost:8090/HelloProj>Hello
Hello Java Server Page ~ !!! 자바 웹페이지 안녕~ !!!
```