

Servlet & JSP

Part-3

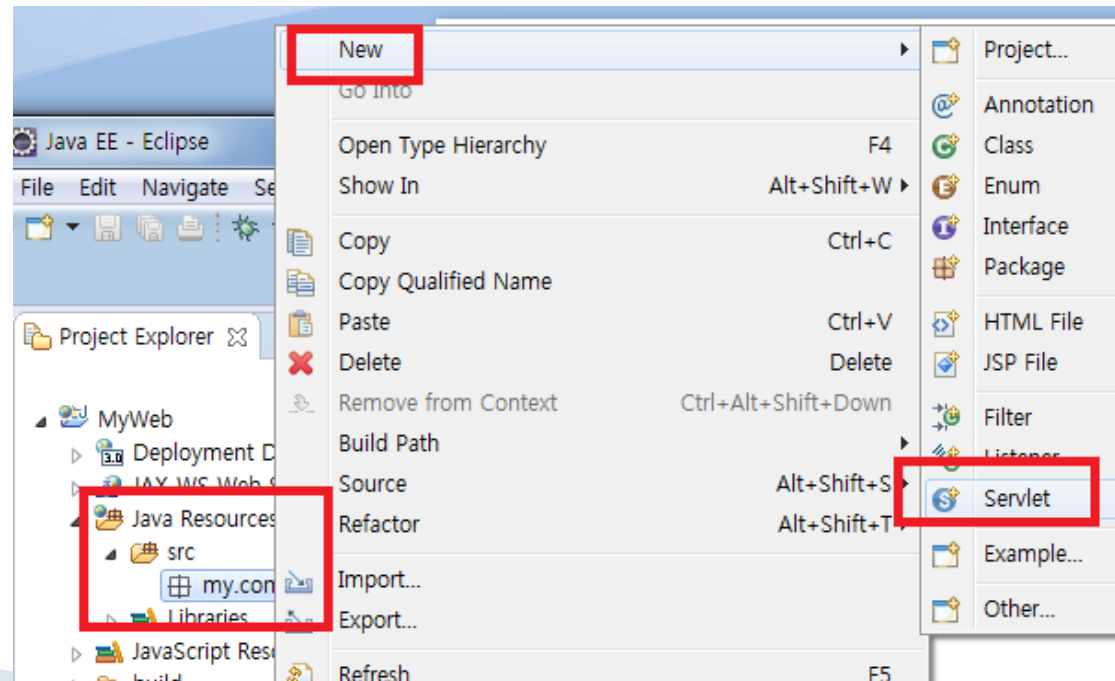
백 성 애

1. Servlet 실행을 위한 환경설정

- ▶ 1) D:\Java\apache-tomcat-7.0.54\lib\ 아래에 있는 **servlet-api.jar** 파일과 **jsp-api.jar** 파일,
▶ **el-api.jar** 파일을
▶ D:\Java\jdk1.7.0 _07\jre\lib\ext\ 아래에 붙여 넣기 하자.
- ▶ 2) 오라클 드라이버 **odjbc6.jar** 파일을 복사하여
▶ D:\Java\apache-tomcat-7.0.54\lib\ 아래 붙여 넣기 한다.

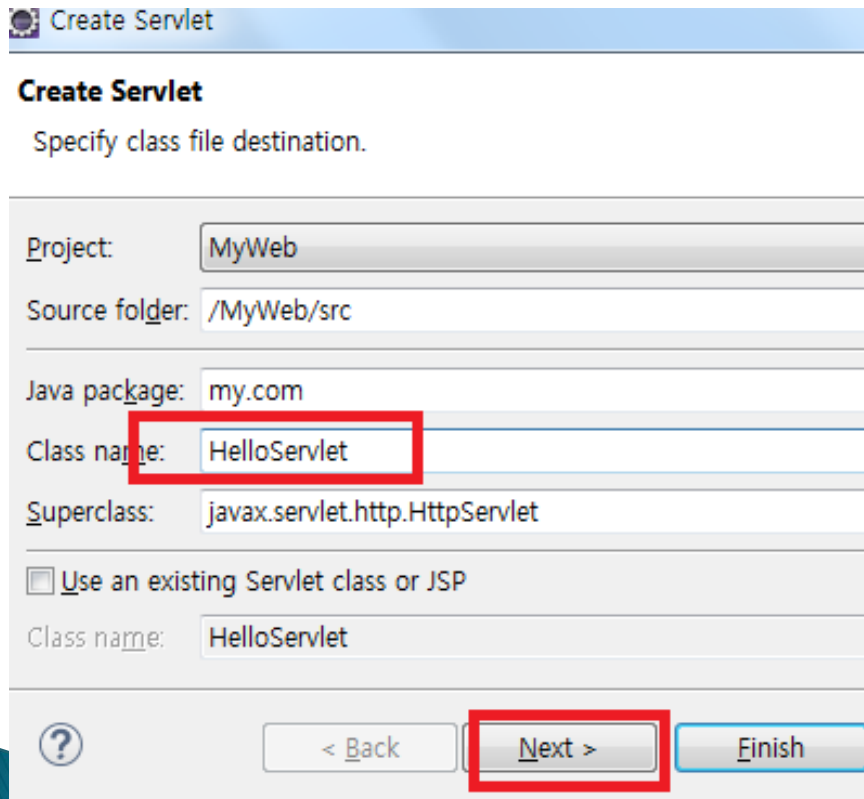
2. Servlet 만들기

- ▶ Dynamic Web Project로 생성한 MyWeb 의 JavaResources / src에 my.com 패키지를 만들자.
- ▶ 해당 패키지에 File-New-Servlet



클래스명을 HelloServlet으로 주고
Next

-> URL mappings 에 디폴트로
/HelloServlet으로 나오는 것을 주의 깊
게 보자.



Create Servlet

Specify class file destination.

Project: MyWeb

Source folder: /MyWeb/src

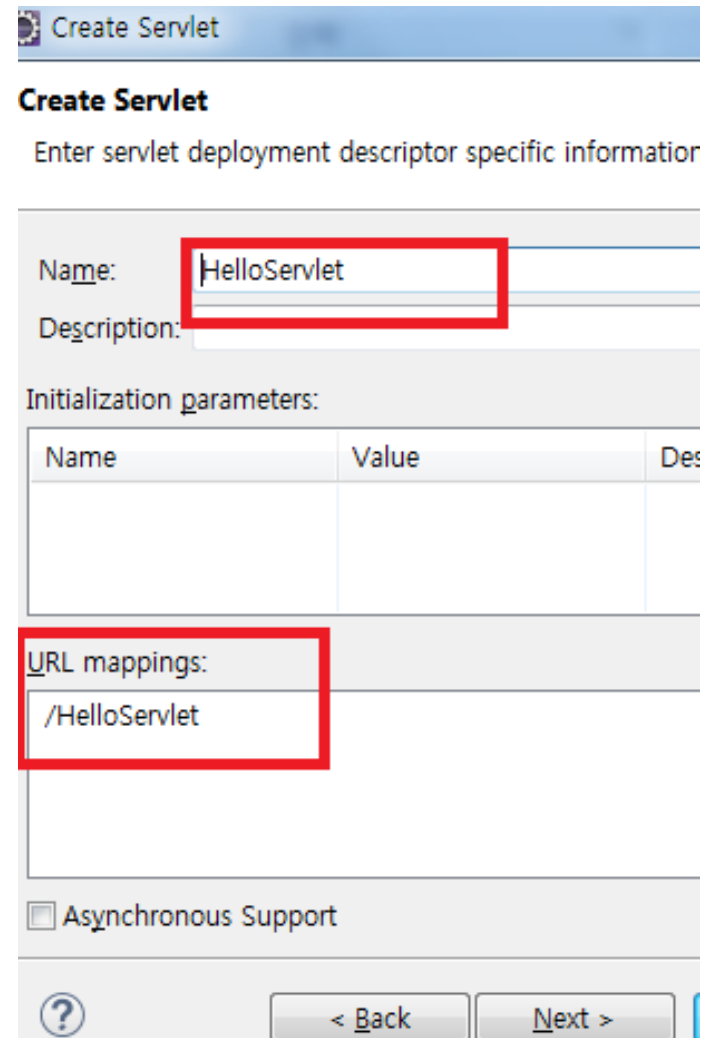
Java package: my.com

Class name: HelloServlet

Superclass: javax.servlet.http.HttpServlet

☐ Use an existing Servlet class or JSP

Class name: HelloServlet



Create Servlet

Enter servlet deployment descriptor specific information

Name: HelloServlet

Description:

Initialization parameters:

Name	Value	Description

URL mappings:

/HelloServlet

☐ Asynchronous Support

doGetMethod 오버라이드 체크

Create Servlet

Create Servlet

Specify modifiers, interfaces to implement, and method stubs to generate.

Modifiers: ☒ public ☐ abstract ☐ final

Interfaces:

Which method stubs would you like to create?

☐ Constructors from superclass

☒ Inherited abstract methods

☐ init ☐ destroy ☐ getServletConfig

☐ getServletInfo ☐ service ☒ doGet

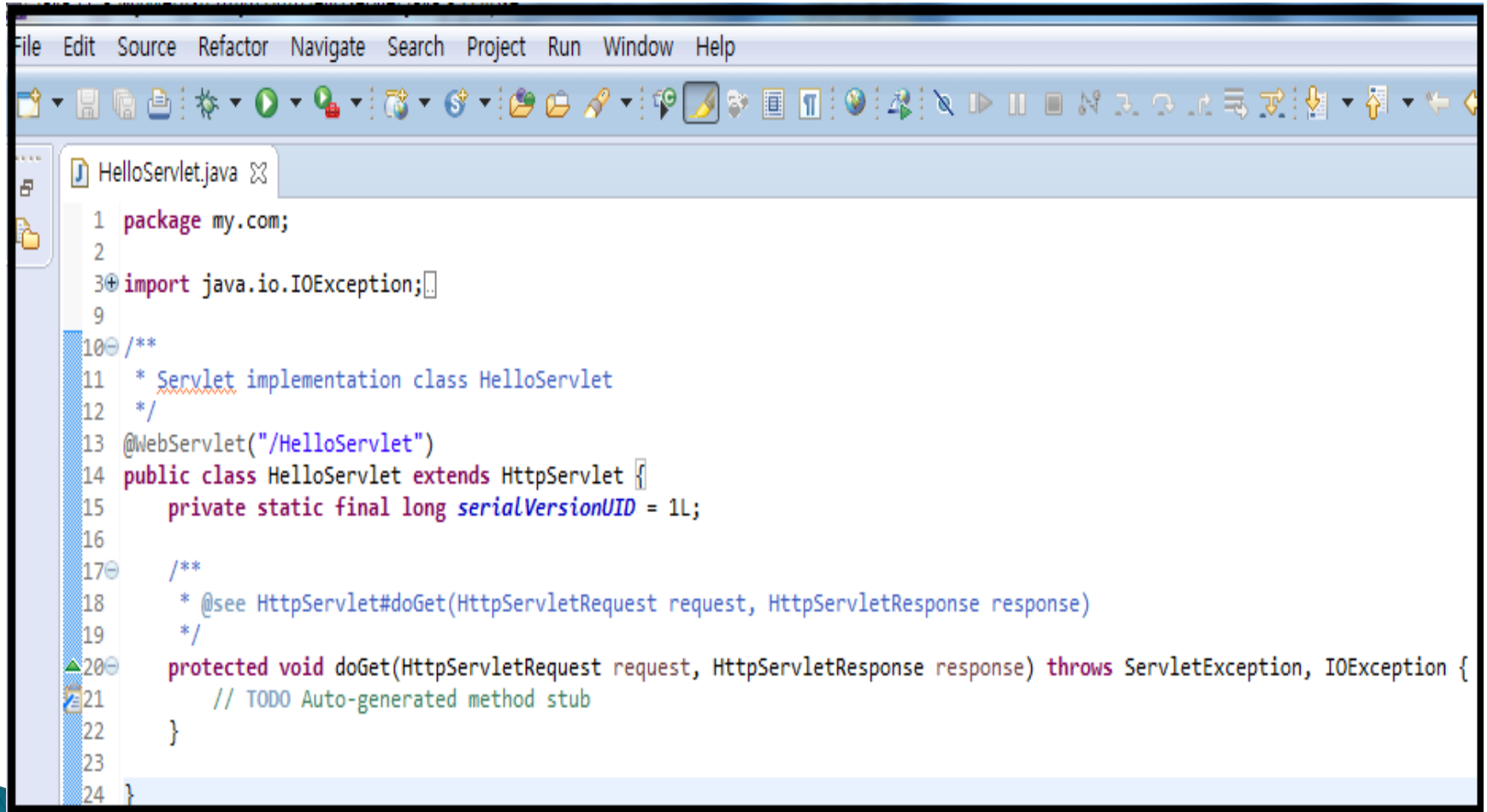
☐ doPost ☐ doPut ☐ doDelete

☐ doHead ☐ doOptions ☐ doTrace

? < Back Next > Finish

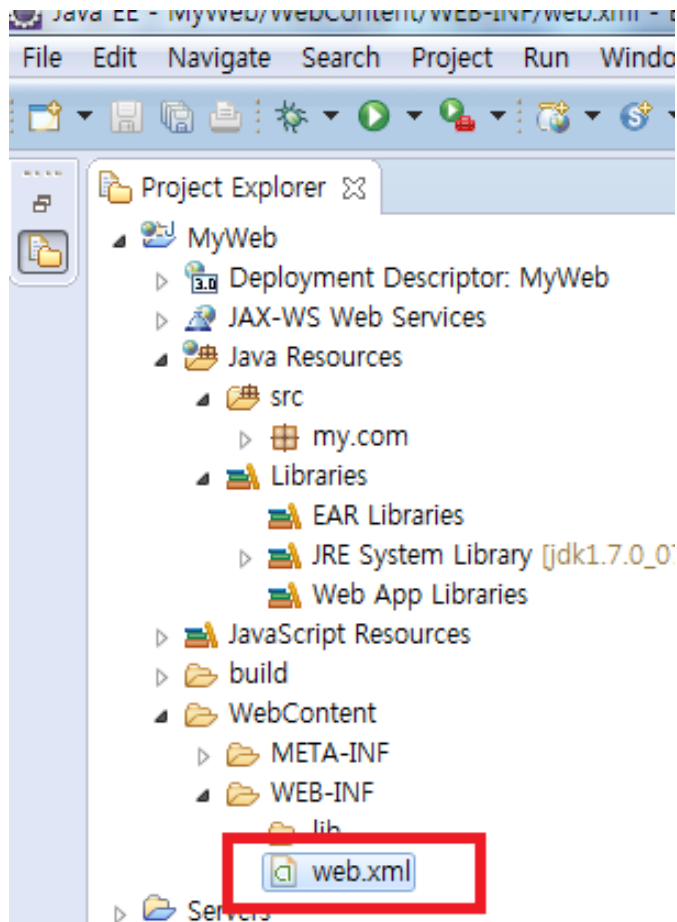
HelloServlet에 구성
할 메소드를 doGet
으로 선택하고
Finish

하단과 같이 HelloServlet클래스 가 생성되는 것을 볼 수 있다.

A screenshot of an IDE window showing the source code of a Java file named 'HelloServlet.java'. The window has a menu bar with 'File', 'Edit', 'Source', 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu bar is a toolbar with various icons for file operations, running, and debugging. The code editor shows the following code:

```
1 package my.com;
2
3 import java.io.IOException;
4
5
6
7
8
9
10 /**
11  * Servlet implementation class HelloServlet
12  */
13 @WebServlet("/HelloServlet")
14 public class HelloServlet extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17     /**
18      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
19      */
20     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
21         // TODO Auto-generated method stub
22     }
23
24 }
```

-web.xml 편집



MyWeb/WebContent/
WEB-INF/**web.xml**

파일을 열어 서블릿 설정에 관한 부분을 추가한다.

web.xml에 서블릿 등록

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
id="WebApp_ID" version="3.0">
  <display-name>MyWeb</display-name>
  <servlet>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>my.com.HelloServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/HelloServlet</url-pattern>
  </servlet-mapping>

</web-app>
```

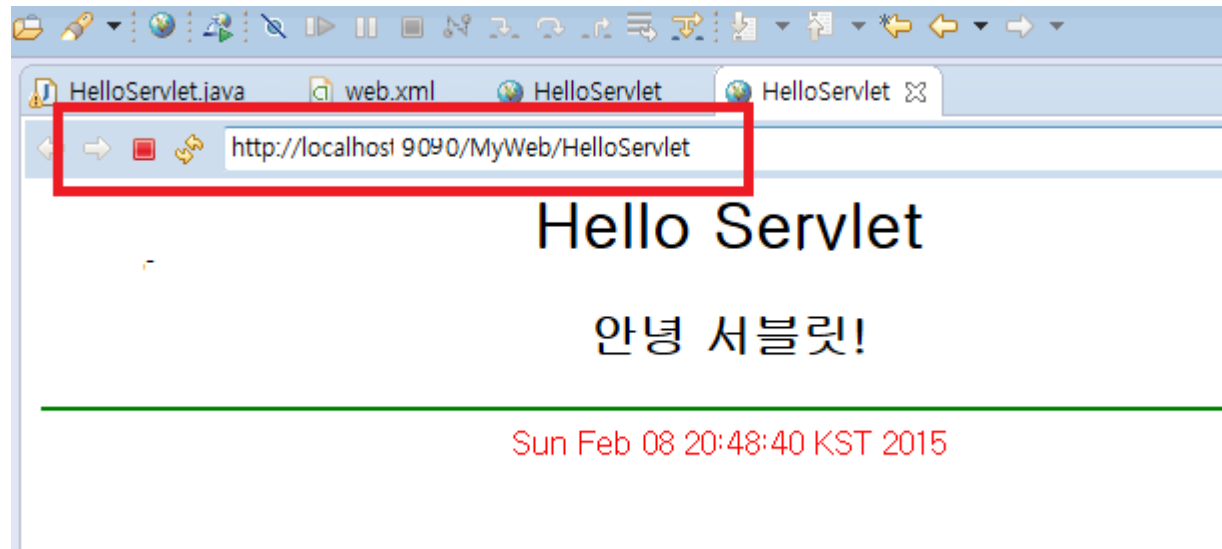

HelloServlet 소스 수정

```
package my.com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 * Servlet implementation class HelloServlet
 */
// @WebServlet("/HelloServlet")
/*web.xml에 서블릿 설정을 했다면 @WebServlet 어노테이션을 주석
처리 해줘야 서블릿이 실행된다. @WebServlet 어노테이션은
web.xml 설정을 대신 하고자 할 때 사용. 따라서 중복 설정하면 안
됨.*/
public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

```
response.setContentType("text/html; charset=EUC-KR");  
//응답보낼 문서의 형식을 지정(마임타입->html의 경우  
// text/html  
PrintWriter out=response.getWriter();  
//웹브라우저에 동적으로 html을 쓰기 위해 스트림 연결  
out.println("<html><head><title>");  
out.println("HelloServlet</title></head>");  
out.println("<body><center>");  
out.println("<h1>Hello Servlet</h1>");  
out.println("<h2>안녕 서블릿!</h2>");  
out.println("<hr color='green'>");  
java.util.Date today=new java.util.Date();  
out.println("<font color='red'>");  
out.println(today.toString());  
out.println("</font>");  
out.println("</center></body>");  
out.println("</html>");  
out.close();  
}  
}
```

- 서블릿 실행

- ▶ HelloServlet.java파일을 선택하고 오른 마우스
- ▶ -> Run As-> Run on Server



web.xml 설정하지 않고 Annotation 으로 하는 방법

- ▶ web.xml에 설정한 HelloServlet 관련 태그
- ▶ `<!--`
- ▶ `<servlet> ...</servlet>`
- ▶ `<servlet-mapping> ...</servlet-mapping>`
- ▶ `-->`
- ▶ 를모두 주석 처리한다.
- ▶ HelloServlet클래스 앞에 선언된
- ▶ `@WebServlet("/HelloServlet")`
- ▶ 에 주석처리한 부분을 주석 해제한다.
- ▶ `@WebServlet` 어노테이션은
- ▶ 서블릿의 복잡한 xml설정을 줄이고자 새로 도입된 방식으로 `"/HelloServlet"` 부분이 `url-pattern`이 된다.