

〈6주차 실습〉

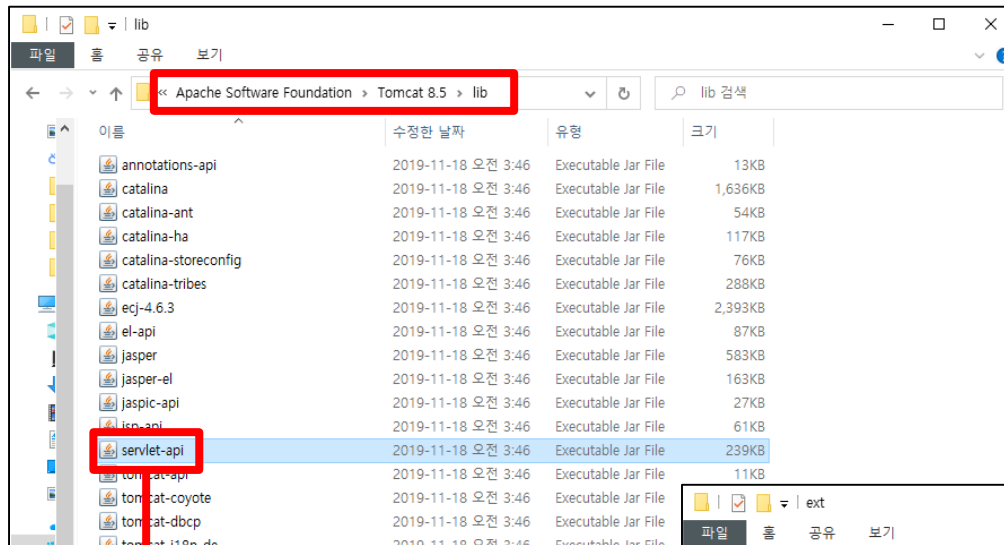
웹 프로그래밍

Web Programming

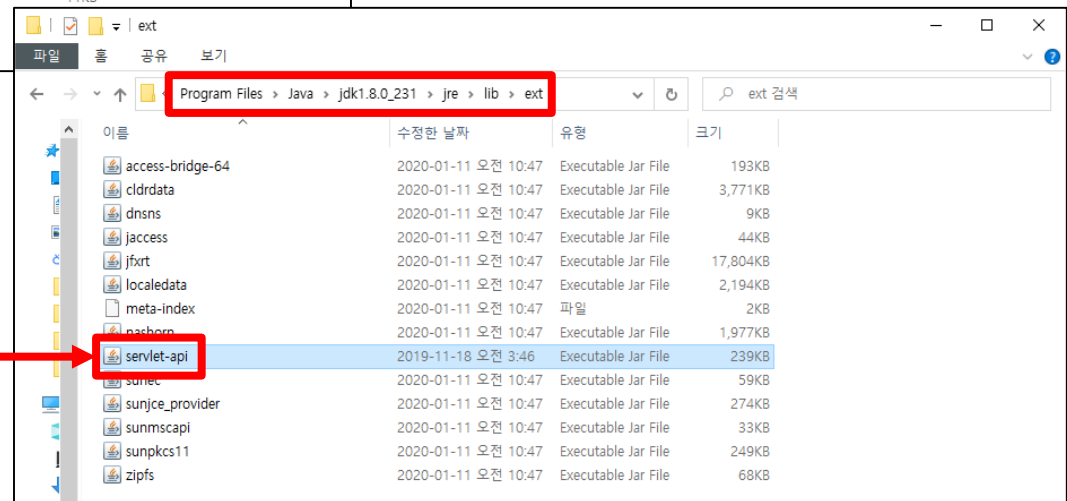
Tomcat 라이브러리 설정

1/20

- Tomcat 설치 후 Library에 Server를 지정해도 오류가 없어지지 않는 경우
- Tomcat 폴더/lib안의 servlet-api.jar을 jdk 설치 폴더/jre/lib/ext로 복사



복사

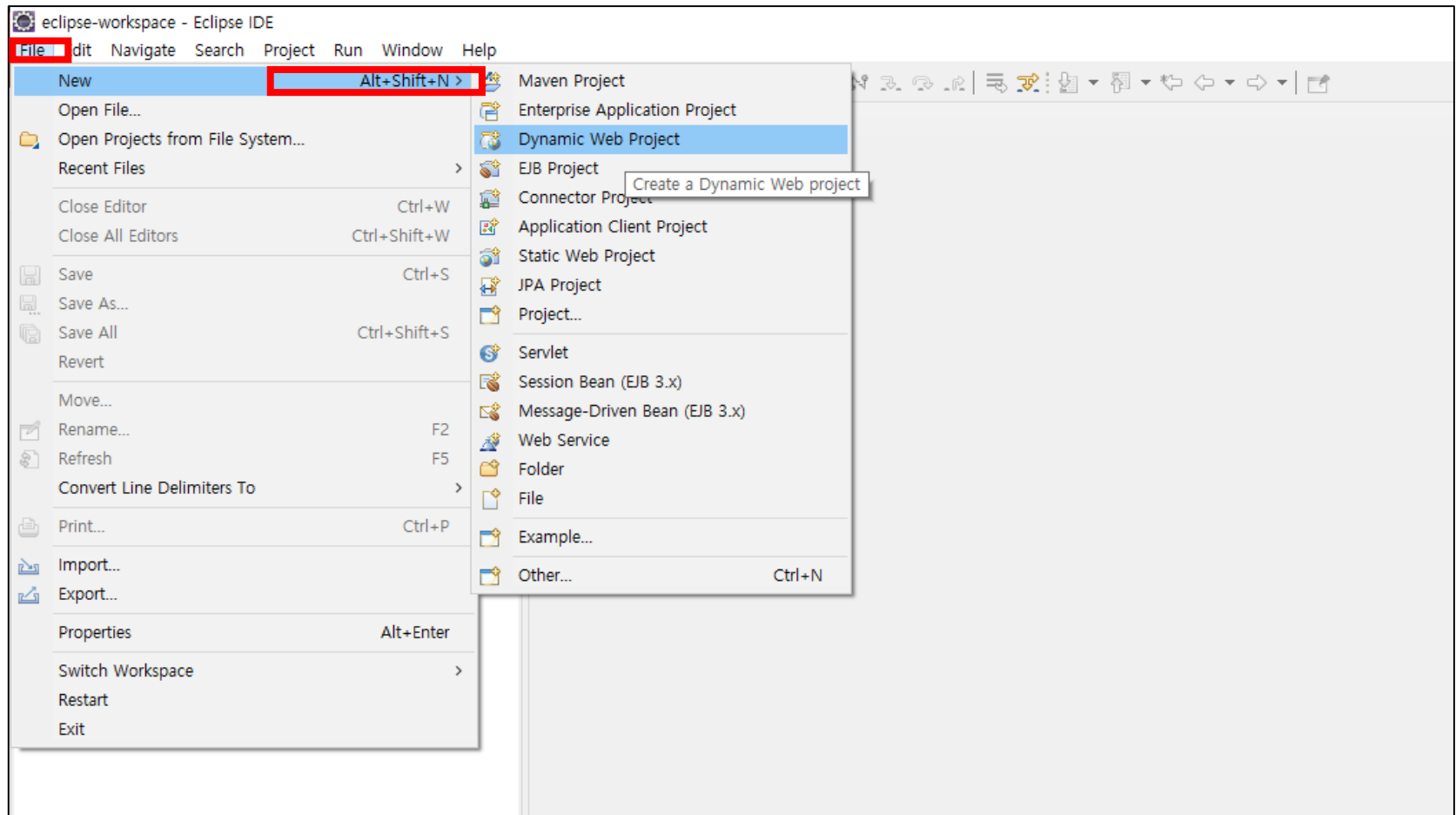


간단한 서블릿 예제 생성하기

- 1) Dynamic Web Project 생성
- 2) Eclipse에 Tomcat Server 설정
- 3) HTML 문서 만들기(Form 양식)
- 4) Servlet(Java Class) 만들기
 - Form으로 부터 전송된 값 받기
 - 응답을 위한 HTML 문서 구성
- 5) 서버 실행 및 테스트

1) Dynamic Web Project 생성

- 이클립스 실행 후 “File” → “New ” → “Dynamic Web Project” 클릭



Eclipse를 이용한 Servlet 실행 (2/12)

4/20

1) Dynamic Web Project 생성

- Project name 입력 후 → “Finish” 클릭
- 파란색 상자 부분처럼 설정 되어있지 않다면, 5주차 실습 자료 참고

New Dynamic Web Project

Dynamic Web Project
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name:

Project location
☒ Use default location
Location:

Target runtime

Dynamic web module version

Configuration

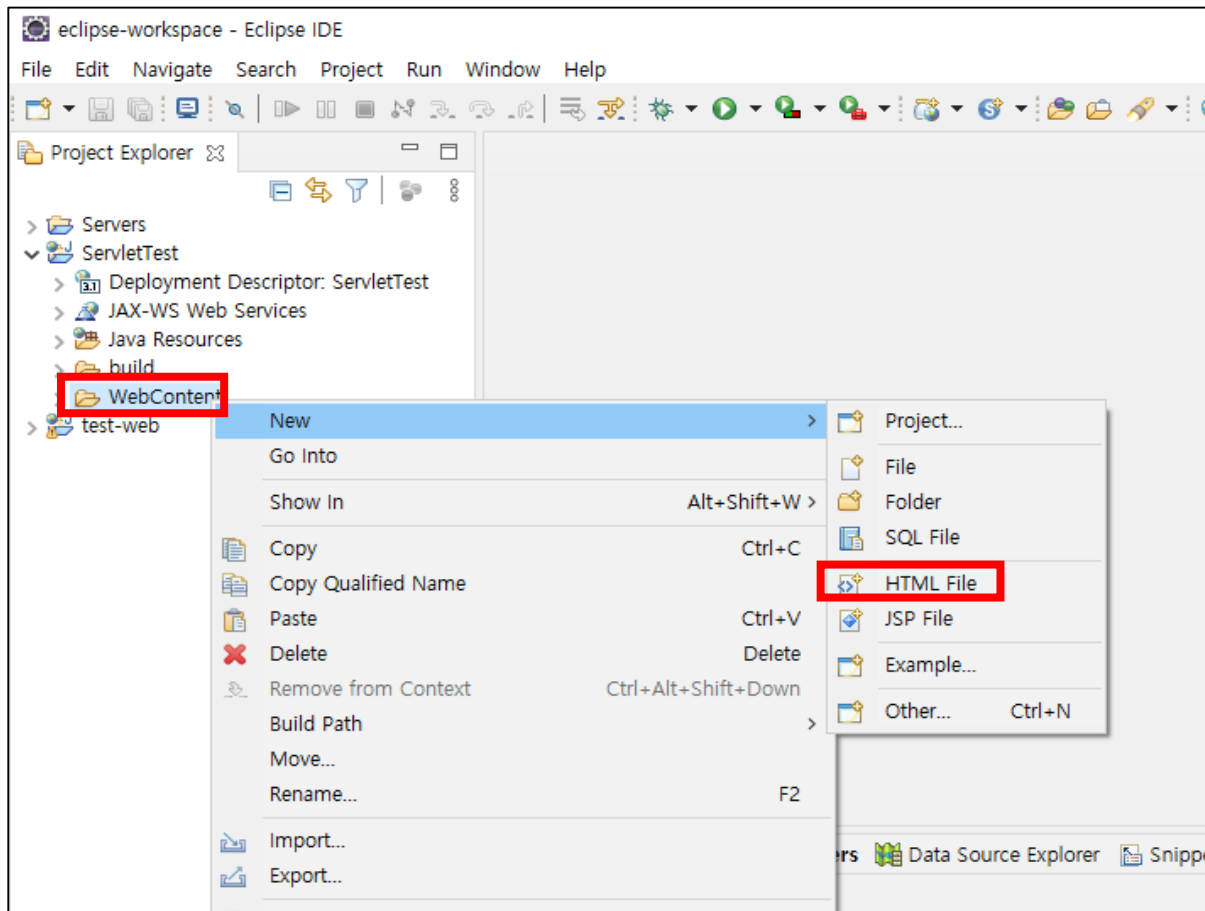
A good starting point for working with Apache Tomcat v8.5 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership
☐ Add project to an EAR
EAR project name:

Working sets
☐ Add project to working sets
Working sets:

3) HTML 문서 만들기(Form 양식)

- ServletTest 프로젝트에서 “WebContent” 우 클릭 (또는 “webapp” 우 클릭)
→ “New” → “HTML File”을 클릭하여 “login.html” 생성



3) HTML 문서 만들기(Form 양식)

<login.html>

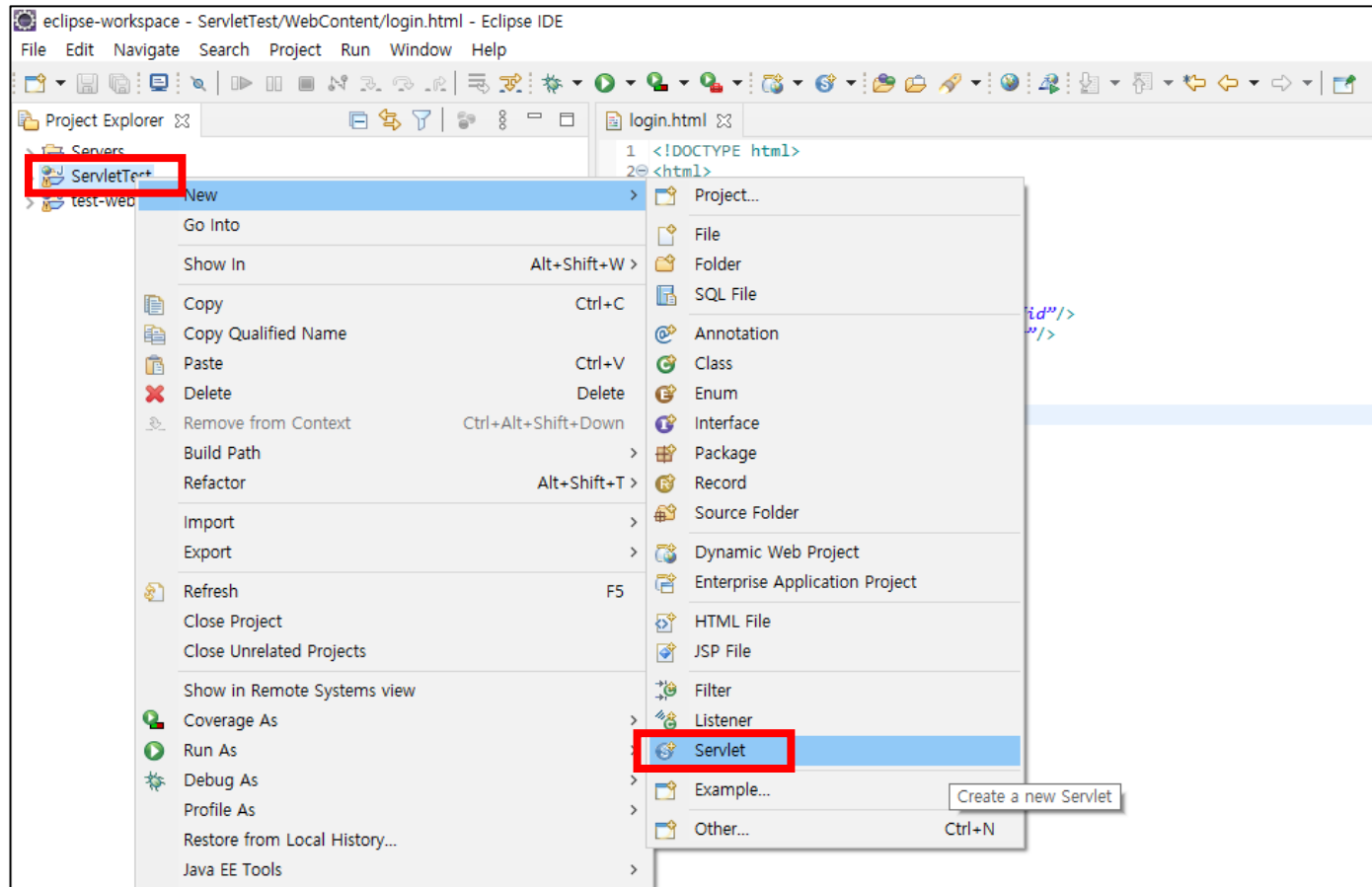
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 <form action="login" method="get">
9 <h1>색상 선택</h1>
10 <select name="color">
11 <option value="red">붉은색</option>
12 <option value="black" selected="selected">검은색</option>
13 <option value="blue">파란색</option>
14 <optgroup label="기타 색상">
15 <option value="purple">보라색</option>
16 <option value="pink">분홍색</option>
17 </optgroup>
18 </select>
19 <input type="submit" value="선택 넘기기">
20 </form>
21 </body>
22 </html>
```

- 요청은 get방식으로 전송
- 서버에서 login이 전송받음

- 서버에서 color라는 이름으로 전송

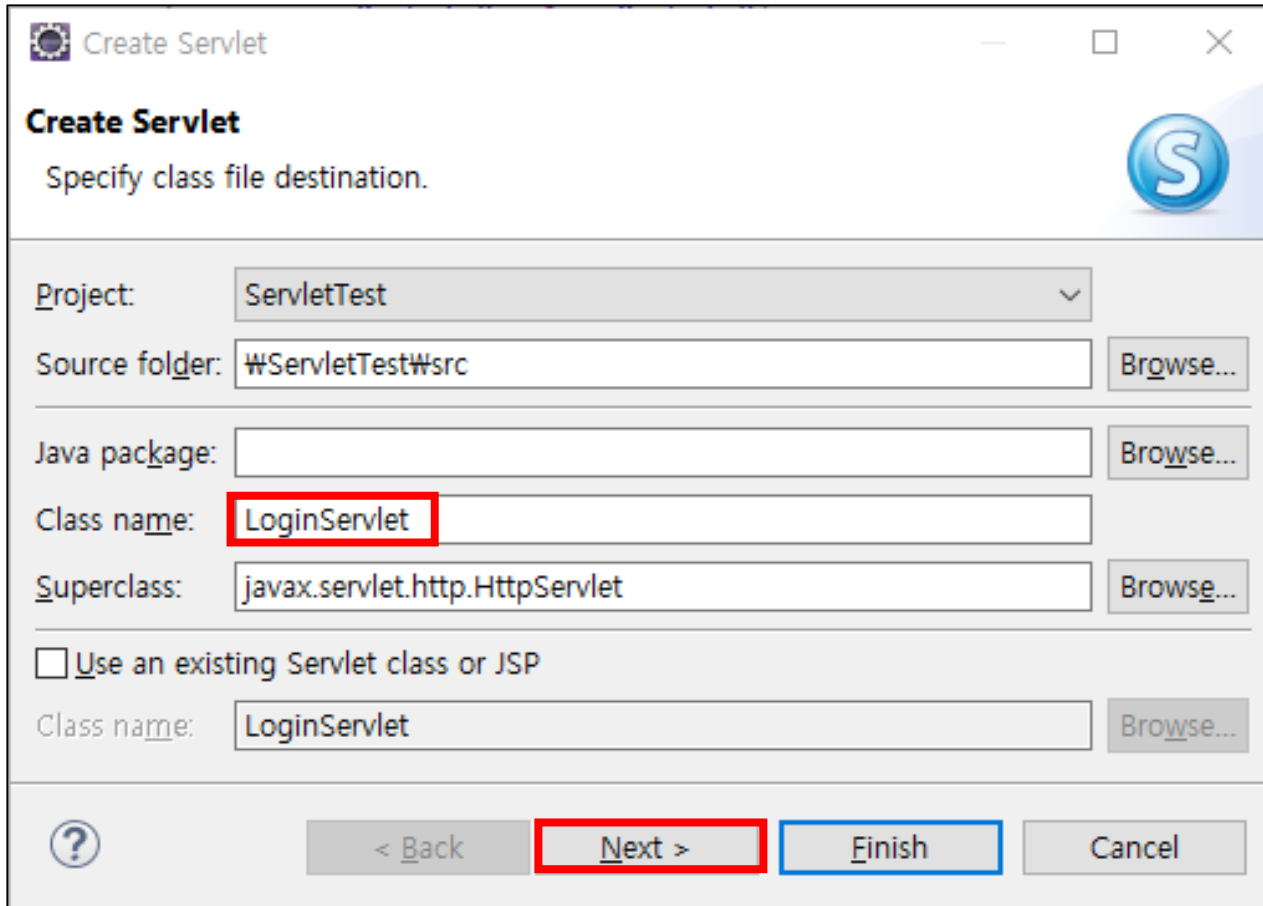
4) Servlet(Java Class) 만들기

- ServletTest 프로젝트 우 클릭 → “New” → “Servlet” 클릭



4) Servlet(Java Class) 만들기

- Class name에 “LoginServlet” 입력 → “Next” 클릭



The image shows the 'Create Servlet' dialog box in Eclipse. The title bar says 'Create Servlet'. Below the title bar, it says 'Create Servlet' and 'Specify class file destination.' There is a blue 'S' icon on the right. The dialog has several fields: 'Project:' with a dropdown menu showing 'ServletTest'; 'Source folder:' with a text field containing '#ServletTest#src' and a 'Browse...' button; 'Java package:' with an empty text field and a 'Browse...' button; 'Class name:' with a text field containing 'LoginServlet' (highlighted with a red box); 'Superclass:' with a text field containing 'javax.servlet.http.HttpServlet' and a 'Browse...' button. There is a checkbox labeled 'Use an existing Servlet class or JSP' which is unchecked. Below this, there is a 'Class name:' field with 'LoginServlet' and a 'Browse...' button. At the bottom, there are four buttons: a help button (question mark), '< Back', 'Next >' (highlighted with a red box), and 'Finish' (highlighted with a blue box), followed by a 'Cancel' button.

Create Servlet

Specify class file destination.

Project: ServletTest

Source folder: #ServletTest#src Browse...

Java package: Browse...

Class name: LoginServlet

Superclass: javax.servlet.http.HttpServlet Browse...

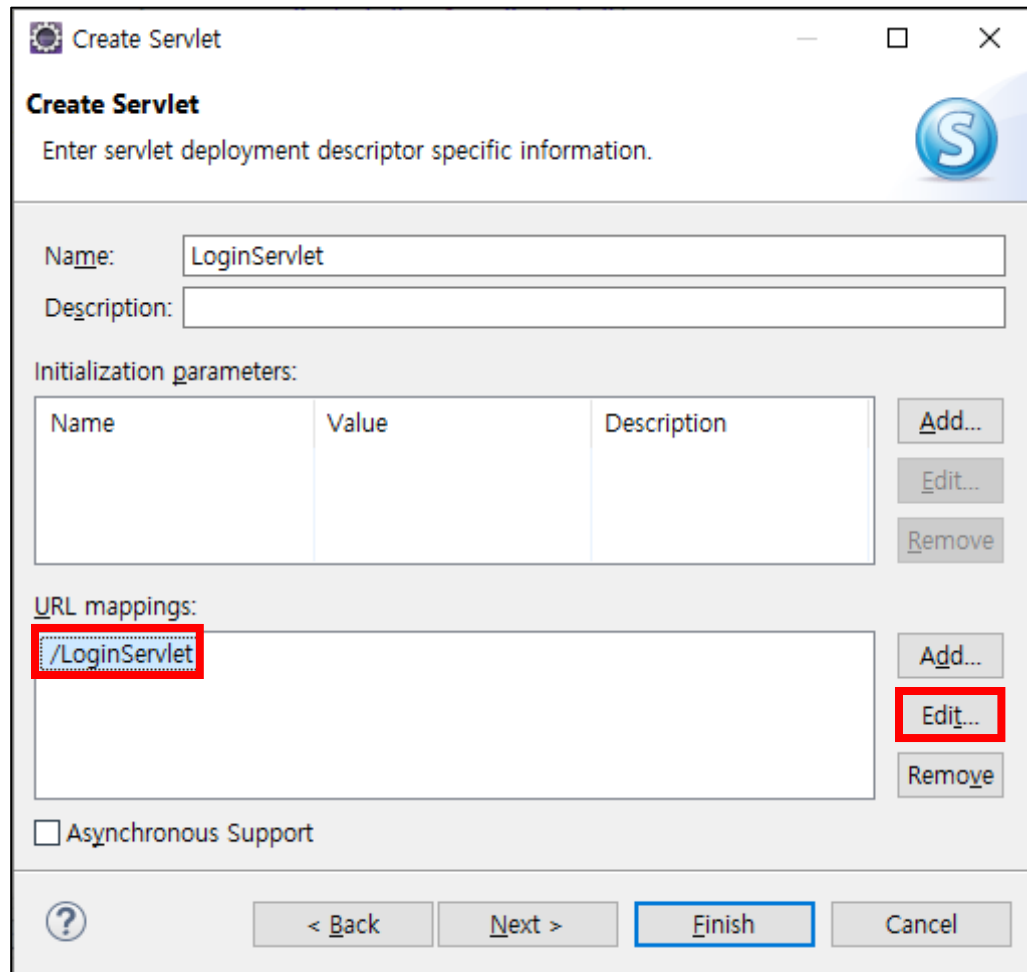
☐ Use an existing Servlet class or JSP

Class name: LoginServlet Browse...

? < Back Next > Finish Cancel

4) Servlet(Java Class) 만들기

- URL mappings 의 “/LoginServlet” 선택 → “Edit” 클릭



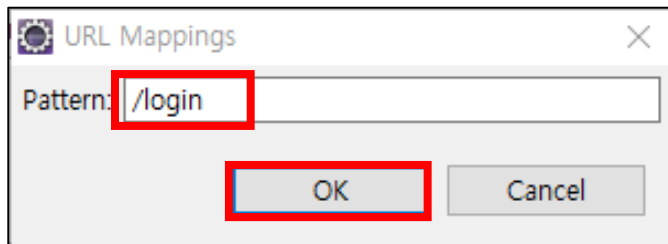
The image shows the 'Create Servlet' dialog box in Eclipse. The dialog has a title bar with the Eclipse logo and the text 'Create Servlet'. Below the title bar, there is a section titled 'Create Servlet' with a sub-instruction 'Enter servlet deployment descriptor specific information.' and a blue 'S' icon. The main area contains several fields and buttons:

- Name:** A text field containing 'LoginServlet'.
- Description:** An empty text field.
- Initialization parameters:** A table with three columns: 'Name', 'Value', and 'Description'. To the right of the table are three buttons: 'Add...', 'Edit...', and 'Remove'.
- URL mappings:** A table with one row containing '/LoginServlet'. To the right of the table are three buttons: 'Add...', 'Edit...', and 'Remove'. The 'Edit...' button is highlighted with a red rectangle.
- Asynchronous Support:** A checkbox that is currently unchecked.

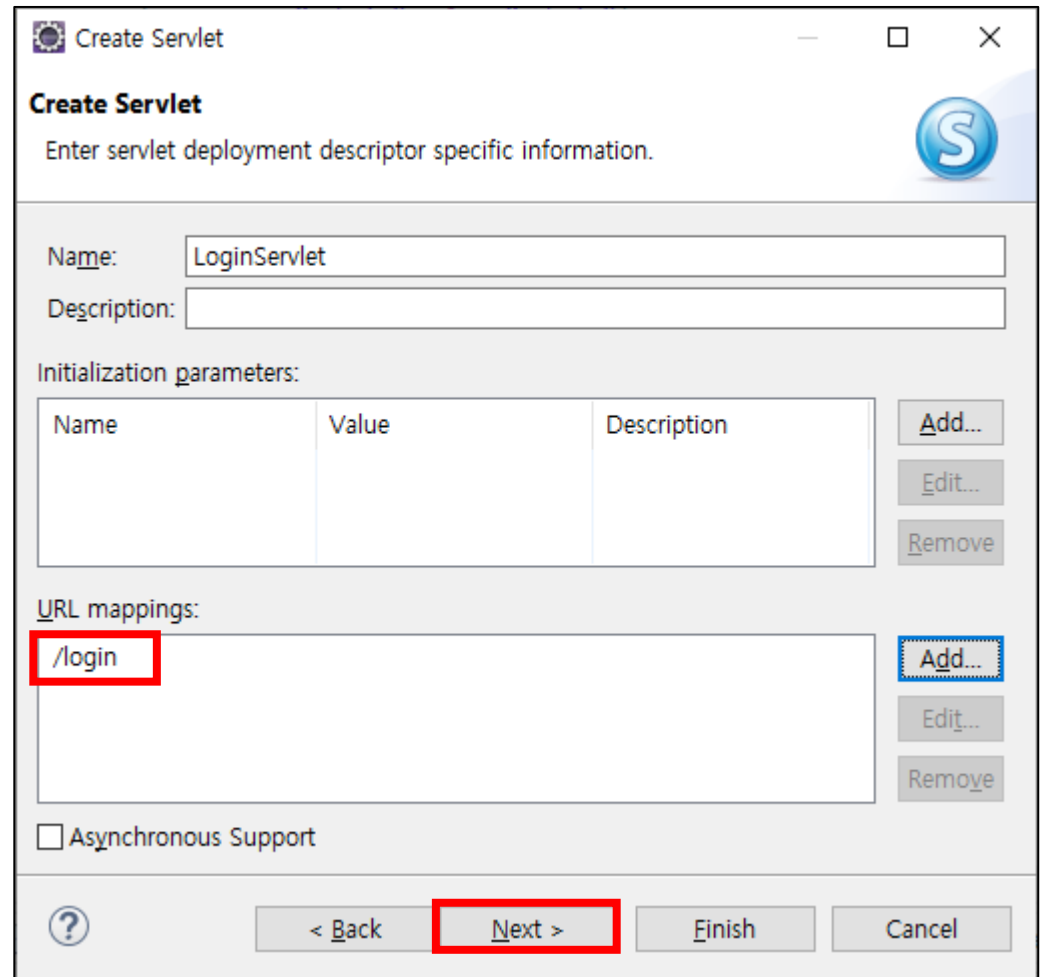
At the bottom of the dialog, there is a row of buttons: a help button (question mark icon), '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

4) Servlet(Java Class) 만들기

- Pattern에 “/login” 입력 → “OK” 클릭
- 변경된 URL mappings 확인 후 “Next” 클릭



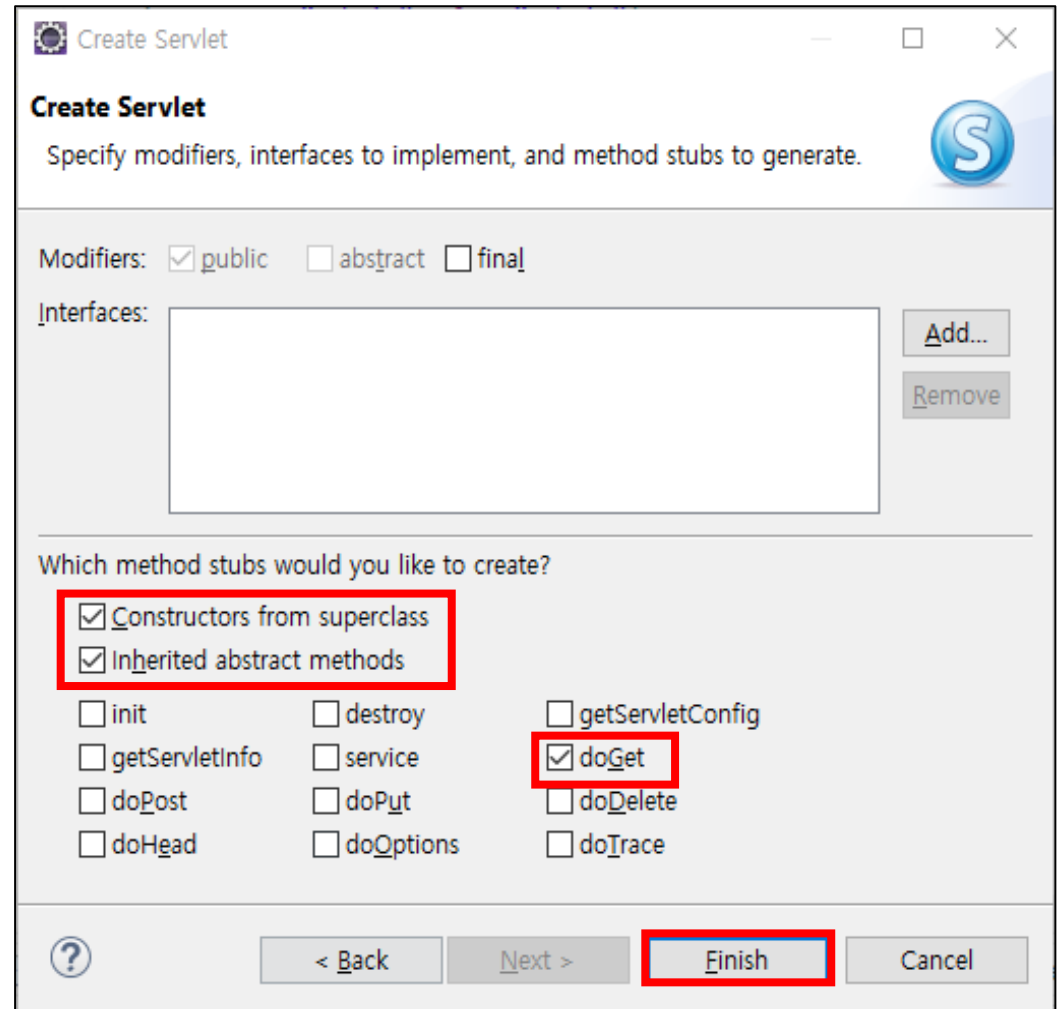
- 서블릿 LoginServlet은 login 주소로 연결됨
- Form으로 부터 login url로 전송함



4) Servlet(Java Class) 만들기

- Form에서 GET 방식으로 전송되므로 아래와 같이 doGet을 선택
→ "Finish" 클릭

- get 방식으로 전송된 요청을 처리하기 위한 코드 작성을 위해 doGet 메소드를 오버라이딩함



The image shows the 'Create Servlet' dialog box in Eclipse. The 'Modifiers' section has 'public' checked. The 'Interfaces' section is empty. The 'Which method stubs would you like to create?' section has 'Constructors from superclass' and 'Inherited abstract methods' checked. The 'doGet' method stub is also checked. The 'Finish' button is highlighted with a red box.

Create Servlet

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

Modifiers: ☒ public ☐ abstract ☐ final

Interfaces: Add... Remove

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 Cancel

4) Servlet(Java Class) 만들기

- LoginServlet.java 코드
- 붉은 상자의 코드 작성

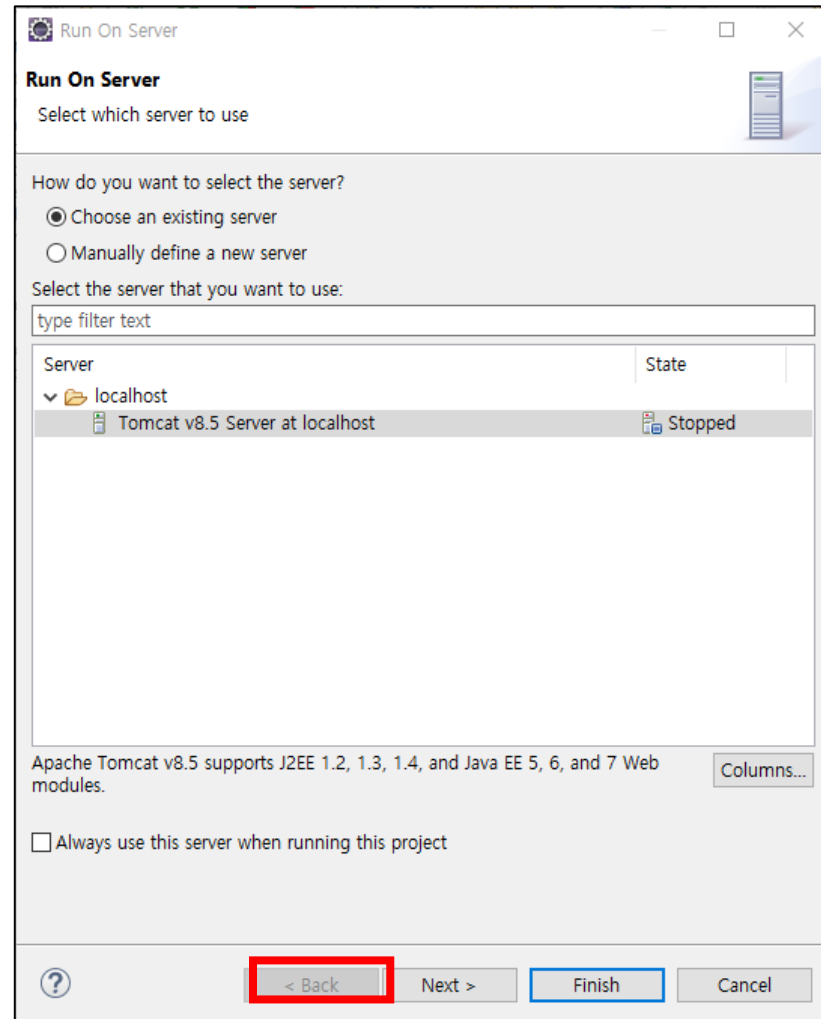
• Form으로 부터 받은 color
의 값을 가져옴

• 출력(클라이언트에 전송)을
하기 위한 코드
(HTML 문서)

```
1 import java.io.IOException;
2 import javax.servlet.ServletException;
3 import javax.servlet.annotation.WebServlet;
4 import javax.servlet.http.HttpServlet;
5 import javax.servlet.http.HttpServletRequest;
6 import javax.servlet.http.HttpServletResponse;
7 import java.io.PrintWriter;
8
9 /**
10  * Servlet implementation class LoginServlet
11  */
12 @WebServlet("/login")
13 public class LoginServlet extends HttpServlet {
14     private static final long serialVersionUID = 1L;
15
16     /**
17      * @see HttpServlet#HttpServlet()
18      */
19     public LoginServlet() {
20         super();
21         // TODO Auto-generated constructor stub
22     }
23
24     /**
25      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
26      */
27     protected void doGet(HttpServletRequest request, HttpServletResponse response)
28         throws ServletException, IOException {
29
30         // TODO Auto-generated method stub
31
32         response.setContentType("text/html;charset=UTF-8");
33
34         String color = request.getParameter("color");
35
36         PrintWriter out = response.getWriter();
37
38         out.println("<HTML>");
39         out.println("<HEAD></HEAD>");
40         out.println("<BODY>");
41         out.println("단일 선택 항목 색상 = " + color + "<br/><br/>");
42         out.println("</BODY>");
43         out.println("</HTML>");
44     }
45 }
```

5) 서버 실행 및 서블릿 테스트

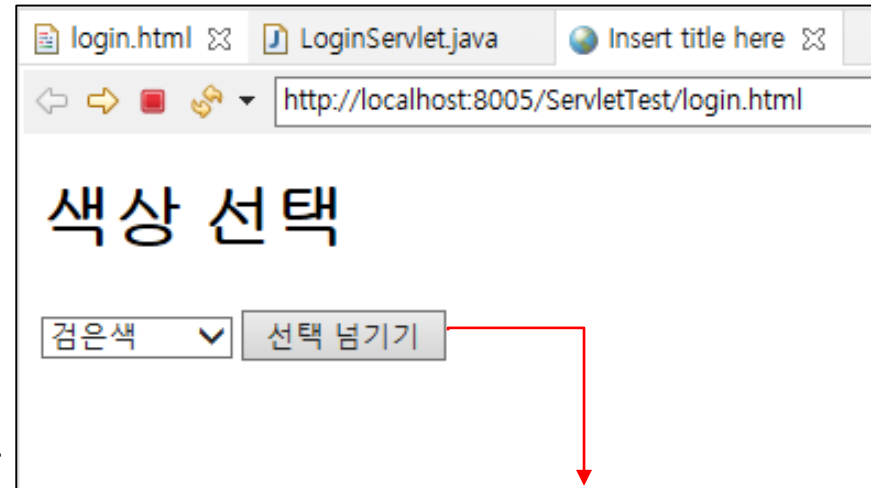
- “login.html” 실행을 위해 해당 파일 우 클릭
→ “Run AS”
→ “Run on Server” 클릭
“Finish” 클릭



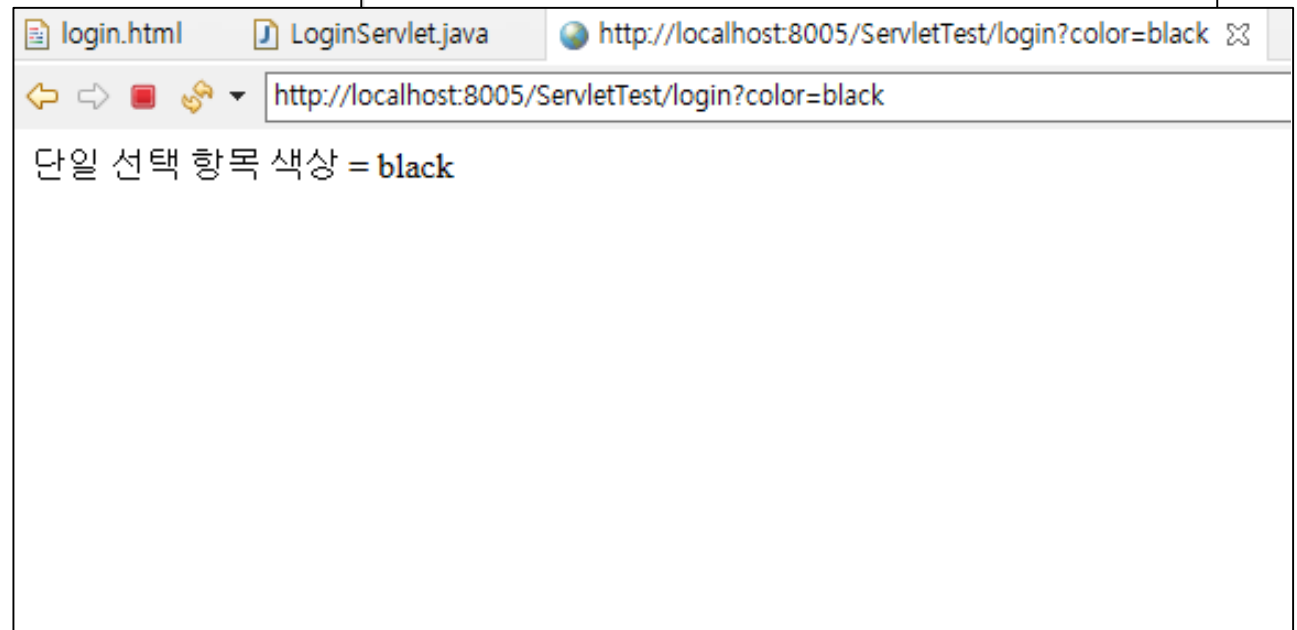
5) 서버 실행 및 서블릿 테스트

- 정상 실행 화면

<HTML 파일 실행>

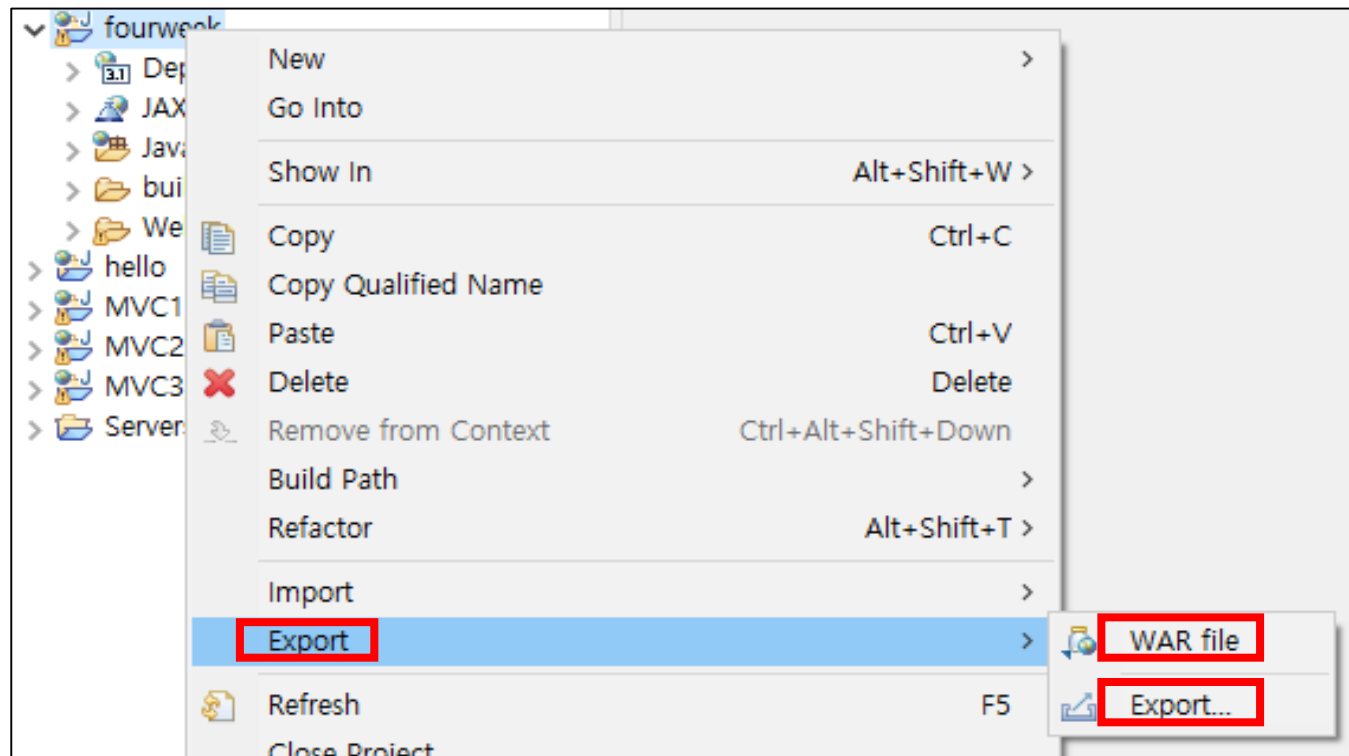


<JAVA 파일 호출>

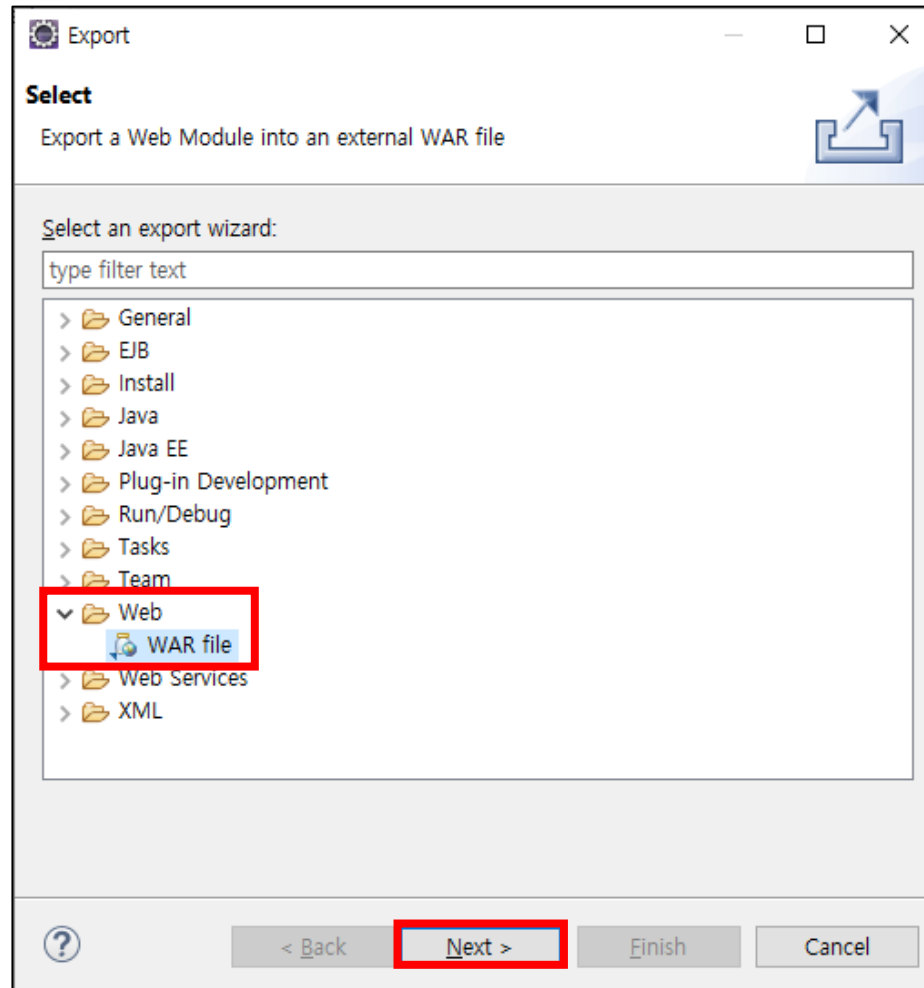


1) 프로젝트 파일 제출 시, war 파일로 만들어 제출

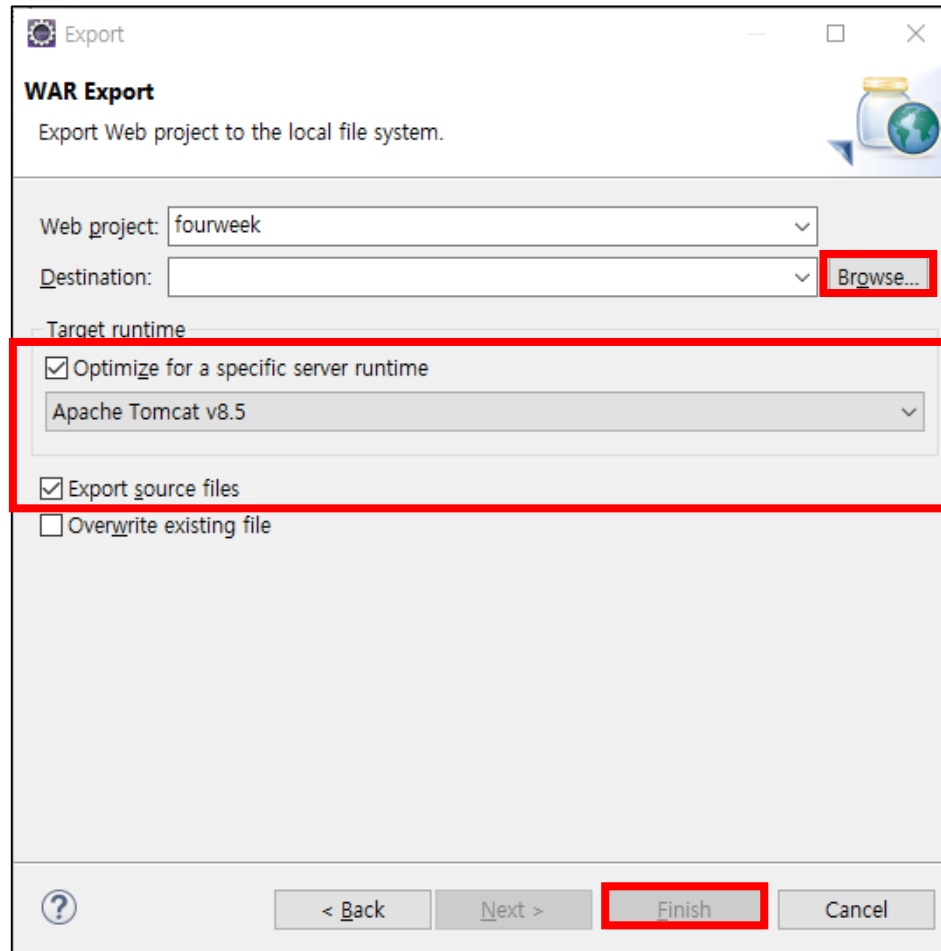
- war파일로 만들 프로젝트 파일 우 클릭
 - "Export"클릭 → "WAR file"클릭(4번으로 이동) 또는
 - "Export"클릭 → "Export"클릭(3번으로 이동)



3) "Web"선택 → "WAR file" 선택 → "Next" 클릭



- 4) "Browe"를 클릭하여 저장 장소 선택
- 5) "Optimize for a specific server runtime"에서 "Apache Tomcat" 버전 선택
- 6) "Export source files" 체크박스 선택 → "Finish" 클릭



Q1. 아래와 같은 결과가 나오도록 HTML 코드와 java 코드를 작성하고, Servlet으로 작동하도록 하시오.

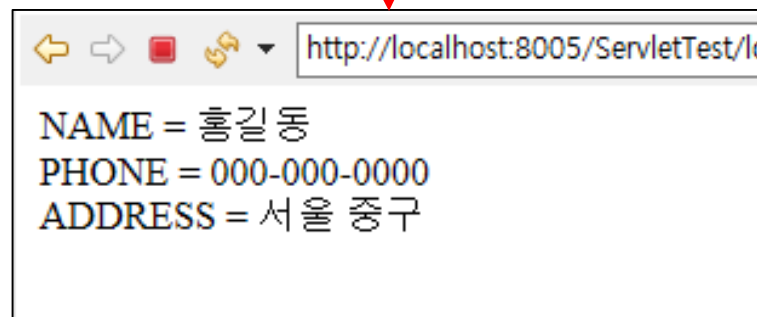
<HTML 파일 실행>



- HTML에서는 양식을 만듦
- Sevlet에서는 입력값을 받아 출력HTML생성

정보 입력 후 "제출"

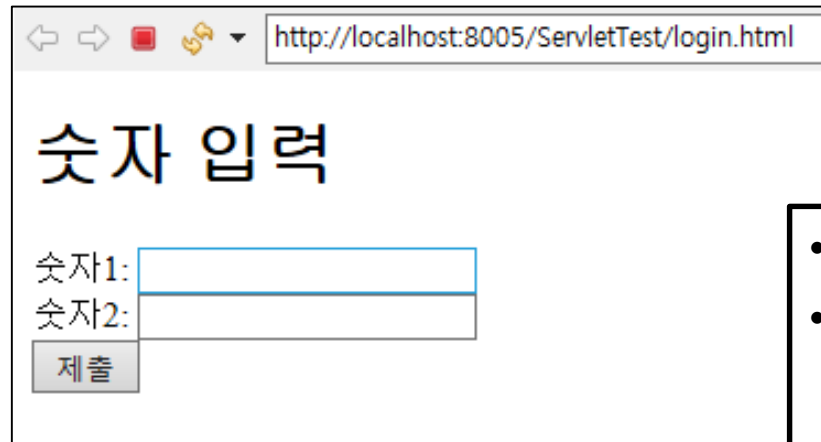
<Java 파일 호출>



```
NAME = 홍길동
PHONE = 000-000-0000
ADDRESS = 서울 중구
```

Q2. 아래와 같은 결과가 나오도록 HTML 코드와 java 코드를 작성하고, Servlet으로 작동하도록 하시오.

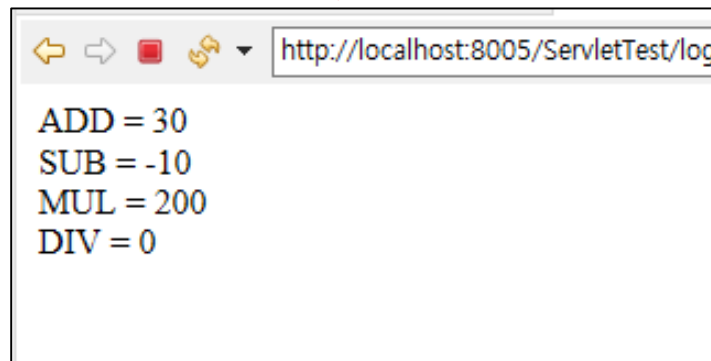
<HTML 파일 실행>



- HTML에서는 양식을 만듦
- Sevlet에서는 입력값을 받아 숫자로 변경 후 계산한 결과를 HTML 생성

값 입력 후 "제출"

<Java 파일 호출>



```
ADD = 30
SUB = -10
MUL = 200
DIV = 0
```

- 제출 방식 : E-Class를 통하여 제출
- 제출 내용 : 프로젝트 파일을 압축한 war 파일, Image File
 - 각 문제 별로 프로젝트 파일을 생성할 것
 - 각 프로젝트 명은 Q학번+ _문제 번호로 할 것
 - (ex. 1번 문제: Q2123456_1)
- 제출 형식 : ZIP File
 - (만들어지는 war 파일과 결과 화면 캡처 이미지를 압축)
- 제목 형식 : 학번_이름_주차
 - ex) 학번_홍길동_6주차.zip
- 제출 기한 : 10월 12일(목) 23시 59분
 - 연장 제출 불허