

3

Servlet의 핵심 사항들



이 장에서 다룰 내용

1

클라이언트에서 서블릿으로 요청하는 방식

2

서블릿을 이용한 클라이언트에서 전송되는
요청처리

3

서블릿에서 한글 처리하기

4

하나의 파라미터 이름으로 여러 개의
파라미터 값 처리하기

5

서블릿에서 세션 살펴보기



클라이언트에서 서블릿으로 요청하는 방식

□ GET 방식

- 사용 방식 : `[2]`
- GET 방식으로 요청이 전송되는 경우
 - 브라우저 주소 표시줄에 주소를 직접 입력해서 요청을 전송하는 경우
 - Html의 a 태그를 사용해서 링크를 걸어서 전송하는 경우
`목록보기`
 - Html 폼 태그에서 method 속성을 GET로 지정하는 경우
`<form action="" name="" method="GET">`



[그림 3-1] GET 방식의 요청 처리



클라이언트에서 서블릿으로 요청하는 방식

□ POST 방식

- 사용 방식 : `<form name=" " action=" " method="POST">`
- 회원 가입 요청, 게시판 글쓰기 요청, 자료실 업로드 등을 처리할 때 사용하는 방식



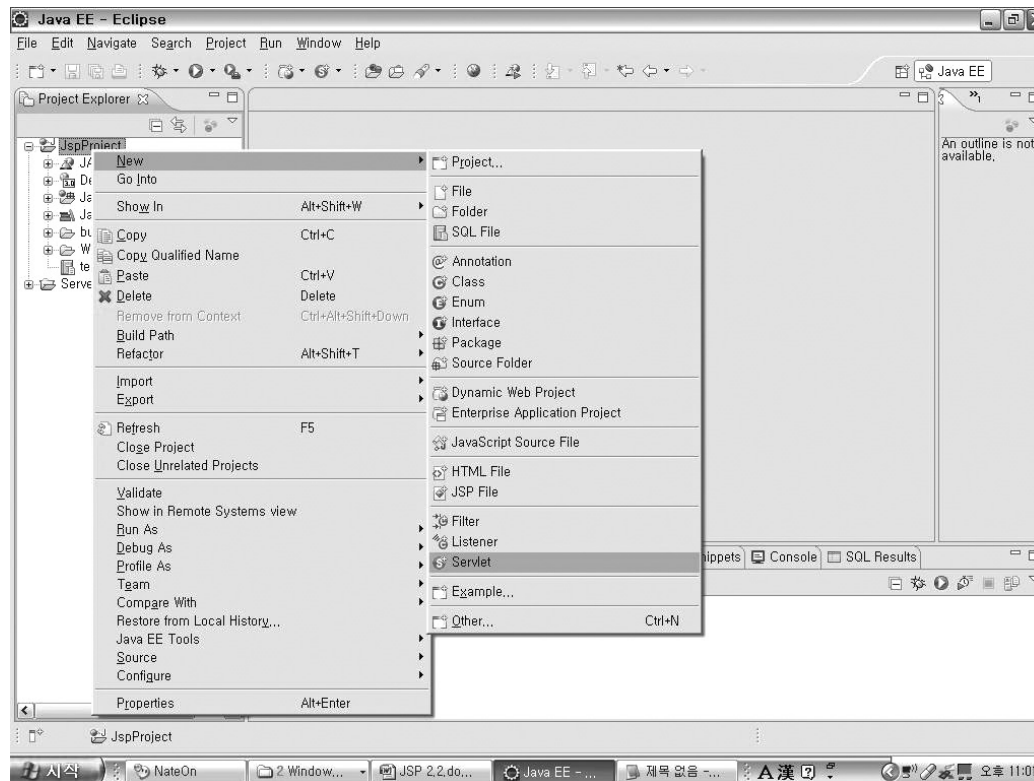
[그림 3-2] POST 방식의 요청 처리



서블릿을 이용한 클라이언트에서 전송되는 요청 처리

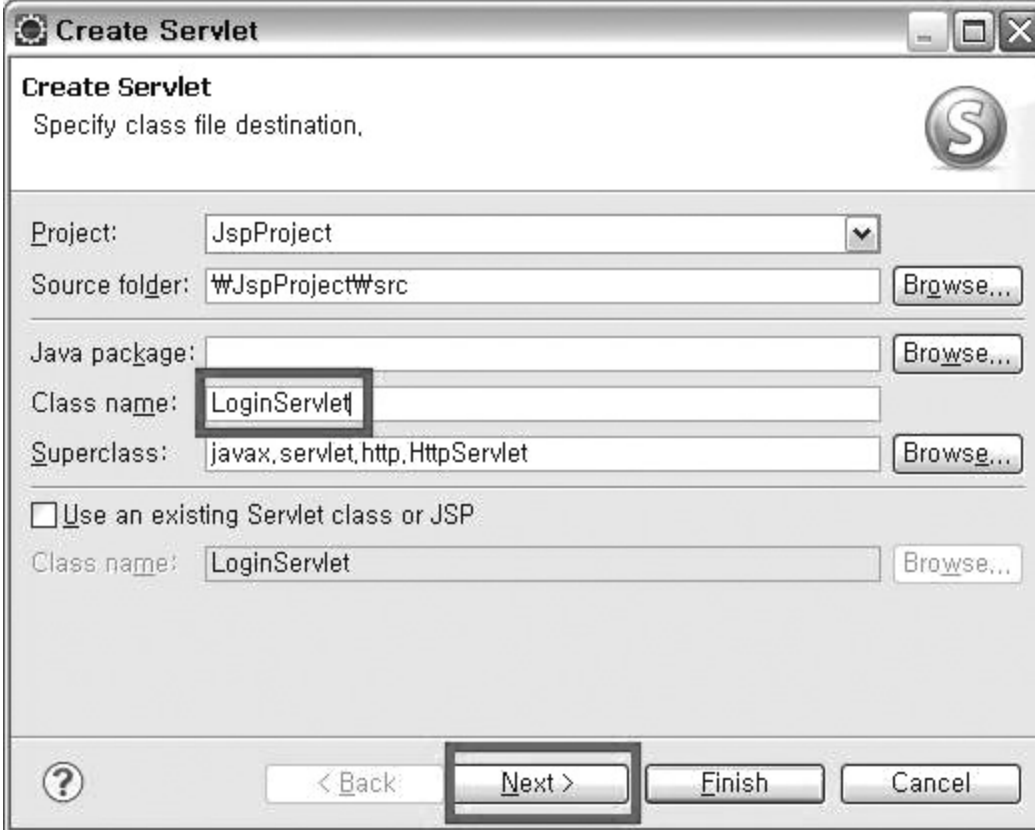
□요청이 GET 방식으로 전송되어 올 경우

- 클라이언트 페이지 생성하기
 - login.html (교재 72p 참조)
- 서블릿 생성하기
 - 01. [New]-[Servlet]을 클릭



서블릿을 이용한 클라이언트에서 전송되는 요청 처리

- 02. Class name에 LoginServlet을 입력하고 [Next] 버튼을 클릭



The image shows a 'Create Servlet' dialog box from an IDE. The 'Class name' field is highlighted with a black rectangle and contains the text 'LoginServlet'. The 'Next >' button at the bottom is also highlighted with a black rectangle. Other fields include 'Project' (JspProject), 'Source folder' (WJspProject\src), 'Java package' (empty), 'Superclass' (javax.servlet.http.HttpServlet), and a checkbox for 'Use an existing Servlet class or JSP' which is unchecked. The 'Finish' and 'Cancel' buttons are also visible.

Create Servlet
Specify class file destination.

Project: JspProject

Source folder: WJspProject\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**



서블릿을 이용한 클라이언트에서 전송되는 요청 처리

- 03. 다음 화면에서 URL mappings 항목을 선택하고 [Edit] 버튼을 클릭



The image shows a 'Create Servlet' dialog box with the following sections:

- Create Servlet**: Enter servlet deployment descriptor specific information.
- Name:** LoginServlet
- Description:** (empty)
- Initialization parameters:**

Name	Value	Description
------	-------	-------------
- URL mappings:**

/LoginServlet

Buttons on the right side of the dialog include 'Add...', 'Edit...', and 'Remove' for both the initialization parameters and URL mappings sections. The 'Edit...' button for the URL mappings section is highlighted with a black box.

At the bottom of the dialog are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.



서블릿을 이용한 클라이언트에서 전송되는 요청 처리

- 04. 클라이언트 폼 태그의 요청 경로가 login(<form action= “login” >)으로 설정되어 있으므로 URL mappings 값을 /login으로 수정한다. 이 부분은 web.xml 설정 파일에서 <url-pattern> 항목의 내용으로 추가되는 부분이다.

Create Servlet

Create Servlet
Enter servlet deployment descriptor specific information.

Name: LoginServlet

Description:

Initialization parameters:

Name	Value	Description
------	-------	-------------

URL mappings:

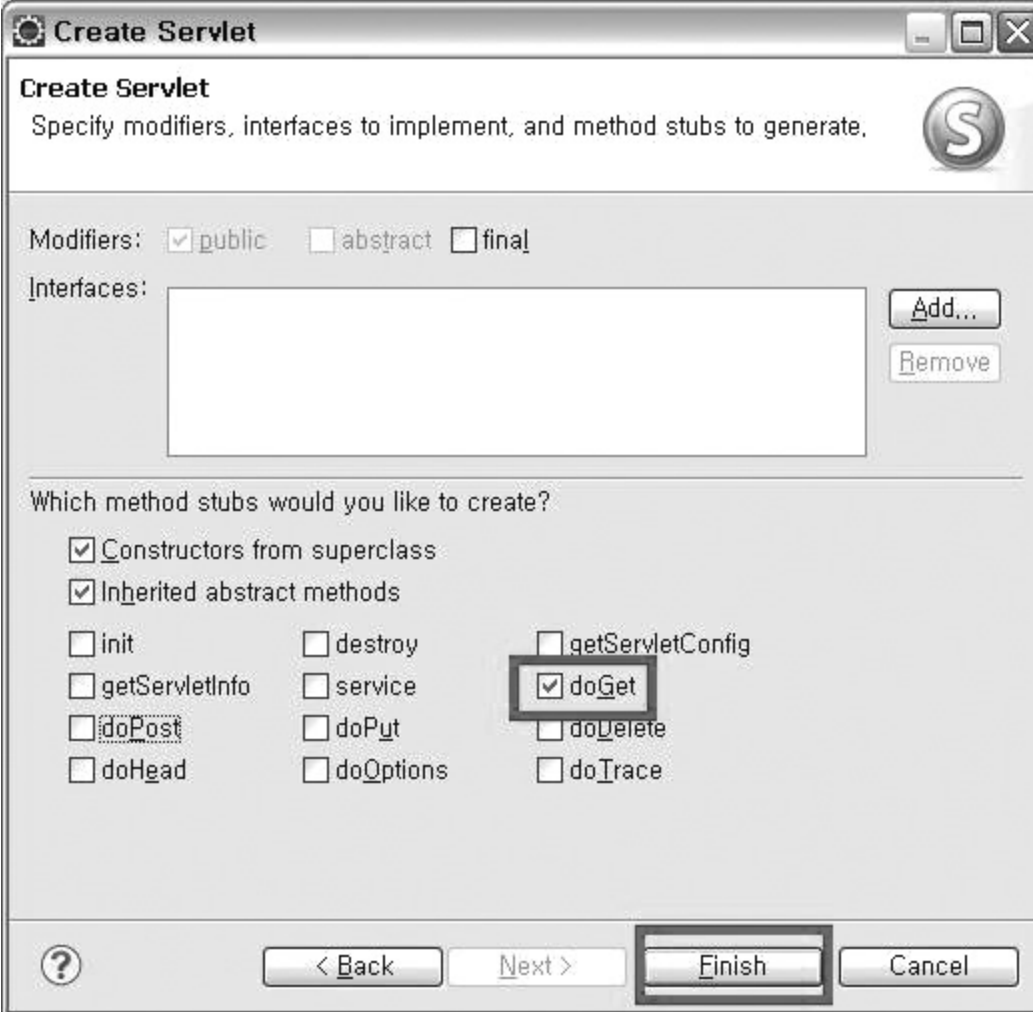
/login

< Back **Next >** Finish Cancel



서블릿을 이용한 클라이언트에서 전송되는 요청 처리

- 05. 본 예제의 요청 방식이 GET 방식이므로 Which method stubs would you like to create? 부분의 체크 박스에서 doGet 메소드만 체크한 후 [Finish] 버튼을 클릭한다.



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?

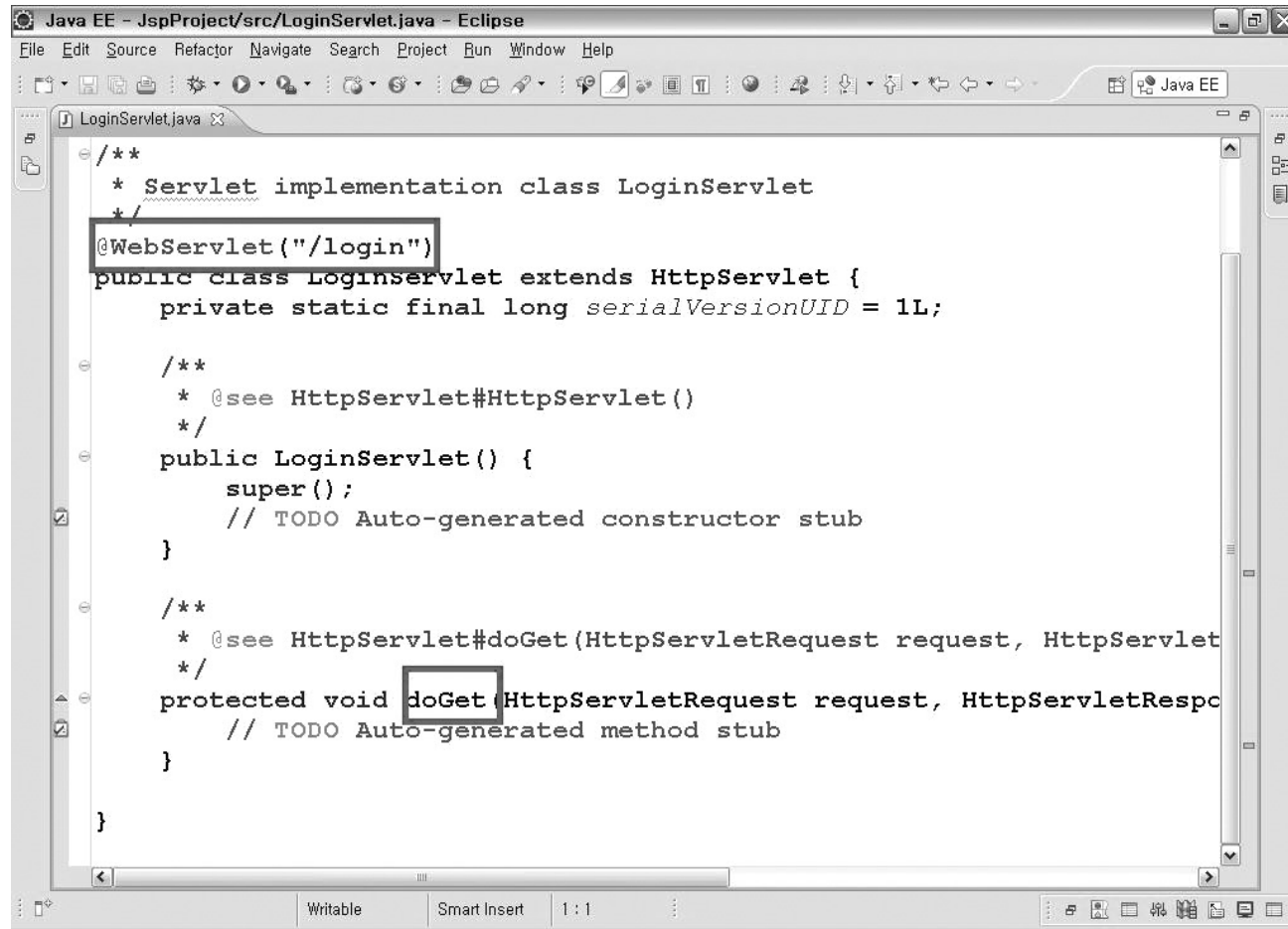
<input checked="" type="checkbox"/> Constructors from superclass		
<input checked="" type="checkbox"/> Inherited abstract methods		
<input type="checkbox"/> init	<input type="checkbox"/> destroy	<input type="checkbox"/> getServletConfig
<input type="checkbox"/> getServletInfo	<input type="checkbox"/> service	<input checked="" type="checkbox"/> doGet
<input type="checkbox"/> doPost	<input type="checkbox"/> doPut	<input type="checkbox"/> doDelete
<input type="checkbox"/> doHead	<input type="checkbox"/> doOptions	<input type="checkbox"/> doTrace

? < Back Next > **Finish** Cancel



서블릿을 이용한 클라이언트에서 전송되는 요청 처리

- 06. 다음과 같은 템플릿 페이지 생성



```
Java EE - JspProject/src/LoginServlet.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

LoginServlet.java
/**
 * Servlet implementation class LoginServlet
 */
@WebServlet("/login")
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

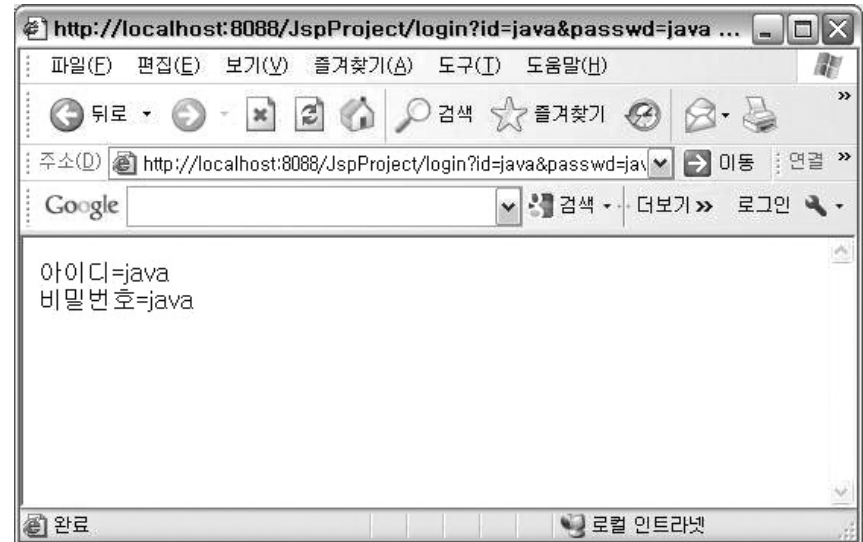
    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) {
        // TODO Auto-generated method stub
    }
}
```

- 07. LoginServlet.java 소스 코드 작성(교재 76p 참조)



서블릿을 이용한 클라이언트에서 전송되는 요청 처리

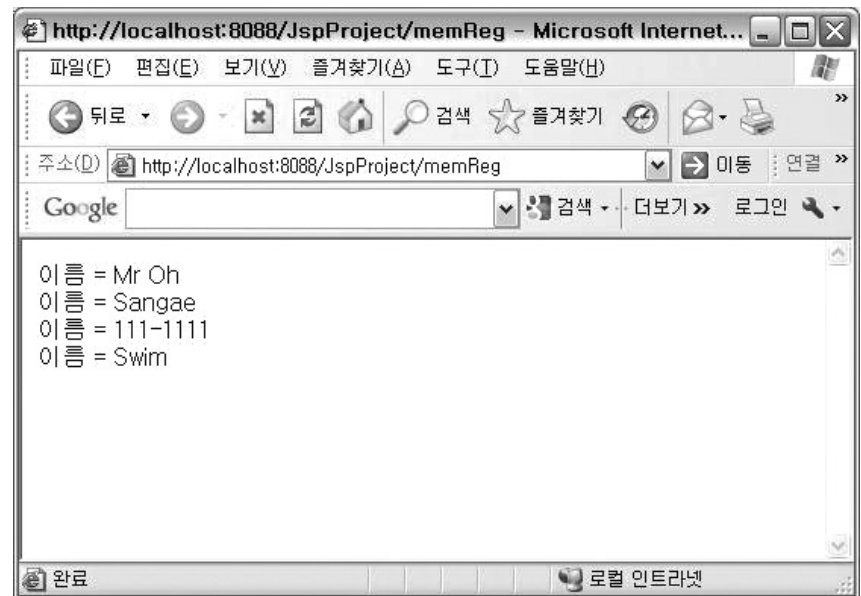
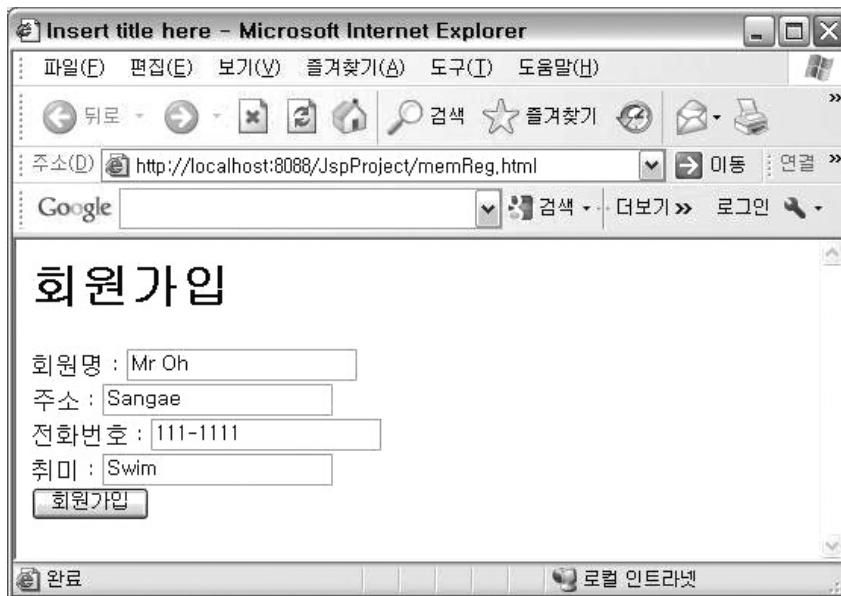
■ 결과 확인하기



서블릿을 이용한 클라이언트에서 전송되는 요청 처리

□ POST 방식으로 요청이 전송되어 올 경우

- 클라이언트 페이지 코딩
 - memReg.html (교재 78p 참조)
- 서블릿 페이지 코딩
 - MemRegServlet.java (교재 78p 참조)



서블릿에서 한글 처리하기

□ GET 방식으로 요청이 전송되어 올 경우

- 한글 처리용 클라이언트 페이지 작성
 - <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
- 개별 문자열들의 캐릭터 셋을 변경하여 처리하는 방법

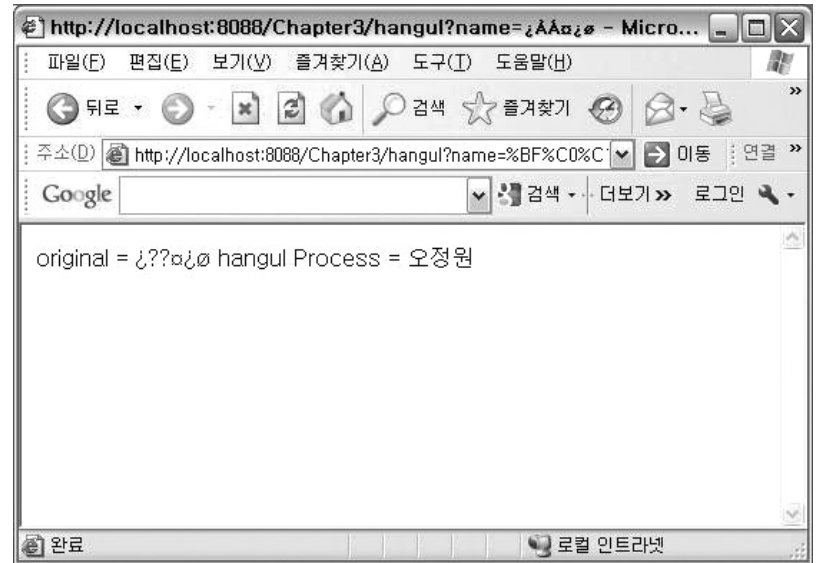
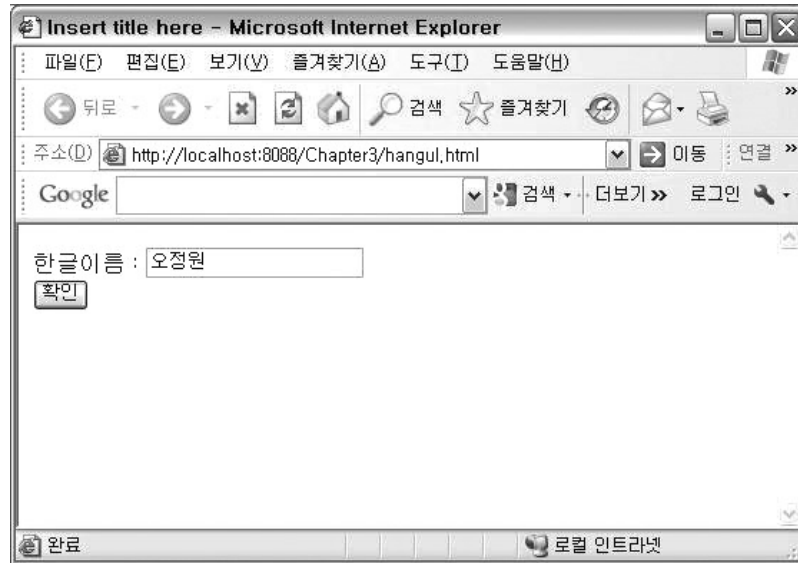
HangulUtil.java		Chapter3Wex3WHangulUtil.java
1	public class HangulUtil {	
2	public static String toKor(String eng){	
3	try{	
4	return new String(eng.getBytes("8859_1"),"euc-kr");	
5	}	
6	catch(Exception e){	
7	return null;	
8	}	
9	}	
10	}	

- **String korName = HangulUtil.toKor(name);**



서블릿에서 한글 처리하기

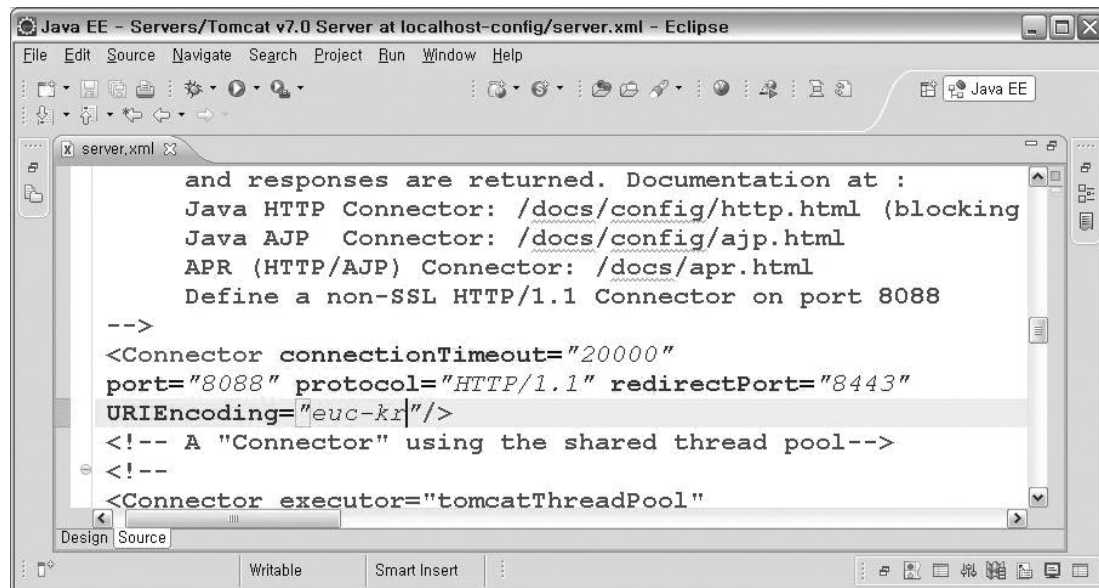
- 실행 결과



- **server.xml을 수정해서 GET 방식으로 전송되어 오는 파라미터들의 캐릭터 셋을 한 번에 수정하는 방법**
 - Connector 태그에 **URIEncoding= “euc-kr”** 을 추가한 후 서버 재시작



서블릿에서 한글 처리하기



■ 서블릿 코드 수정

```
1 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws  
2 ServletException, IOException {  
3     // TODO Auto-generated method stub  
4     String name = request.getParameter("name");  
5     String korName = name;  
6  
7     response.setContentType("text/html;charset=euc-kr");  
8     PrintWriter out = response.getWriter();  
9     out.println("original = " + name);  
10    out.println("hangul Process = " + korName);  
11 }
```

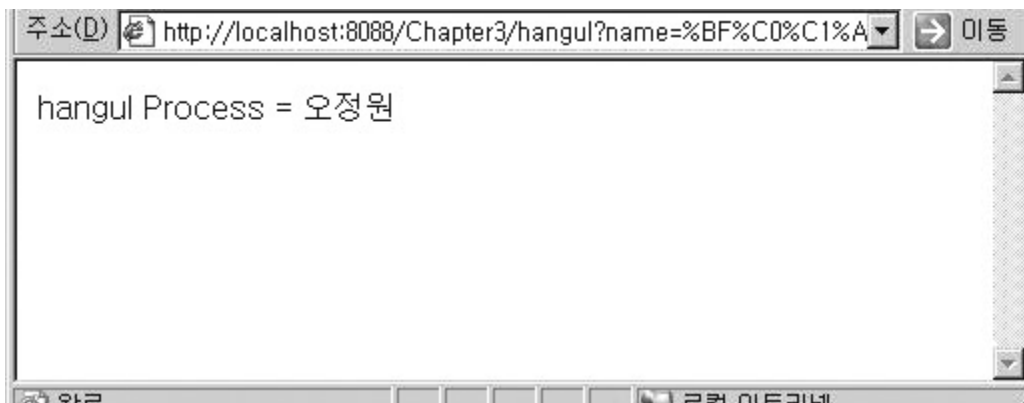


original = 오정원 hangul Process = 오정원

서블릿에서 한글 처리하기

- **server.xml connector 태그의 useBodyEncodingForURI= “true” 속성을 이용하여 한글 처리하는 방법**
 - **Connector 태그에 useBodyEncodingForURI= “true” 속성을 추가한 후 서버 재시작**

```
1  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
2  ServletException, IOException {
3      // TODO Auto-generated method stub
4      request.setCharacterEncoding("euc-kr");
5      String name = request.getParameter("name");
6      response.setContentType("text/html;charset=euc-kr");
7      PrintWriter out = response.getWriter();
8      out.println("hangul Process = " + name);
9  }
```



서블릿에서 한글 처리하기

□ POST 방식으로 요청이 전송되어 올 경우

- Html 페이지의 코딩에서 다음 폼 태그의 method 속성을 POST로 변경
- 서블릿 페이지에서는 doPost에서 request.setCharacterEncoding("euc-kr")을 처리

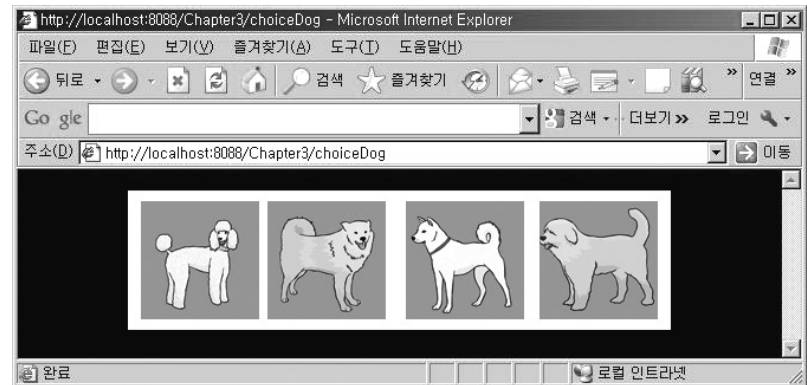
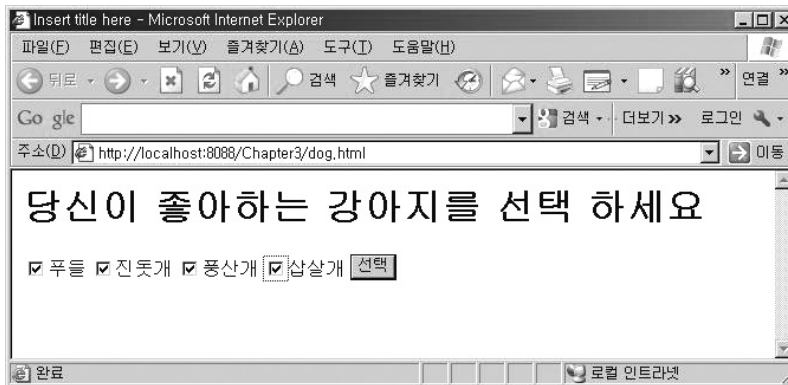
```
1  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
2  ServletException, IOException {
3      // TODO Auto-generated method stub
4      request.setCharacterEncoding("euc-kr");
5      String name = request.getParameter("name");
6      response.setContentType("text/html;charset=euc-kr");
7      PrintWriter out = response.getWriter();
8      out.println("hangul Process = " + name);
9  }
```



하나의 파라미터 이름으로 여러 개의 파라미터 값 처리하기

□ HttpServletRequest 인터페이스 이용

- 하나의 파라미터 값이 전송되는 경우
 - String `getParameter(String paramName)` 메소드를 사용
- 하나의 파라미터 이름으로 여러 개의 값이 전송되어 올 경우
 - String[] `getParameterValues(String paramName)` 메소드를 사용
- 예제
 - 클라이언트 페이지 - dog.html (교재 86p 참조)
 - 서블릿 페이지 - ChoiceDogServlet.java (교재 86p 참조)

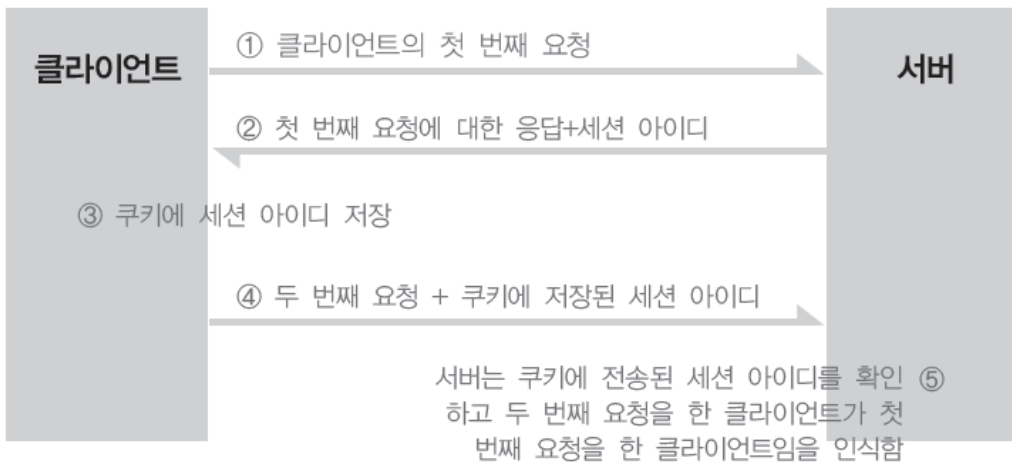


서블릿에서 세션 살펴보기

□ 세션의 개념



[그림 3-3] HTTP 프로토콜의 상태를 유지하지 않는 속성



[그림 3-4] 세션 기능이 적용된 HTTP 프로토콜 요청 처리



서블릿에서 세션 살펴보기

□ 간단한 세션 예제

■ HttpSession 인터페이스 사용

SetNameServlet.java



Chapter3\Wex5\SetNameServlet.java

```
1 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
2 ServletException, IOException {
3     // TODO Auto-generated method stub
4     HttpSession session = request.getSession();
5     session.setAttribute("name","오정원");
6     response.setContentType("text/html;charset=euc-kr");
7     PrintWriter out = response.getWriter();
8     out.println("<h1>이름저장</h1>");
9 }
```

GetNameServlet.java의 doGet



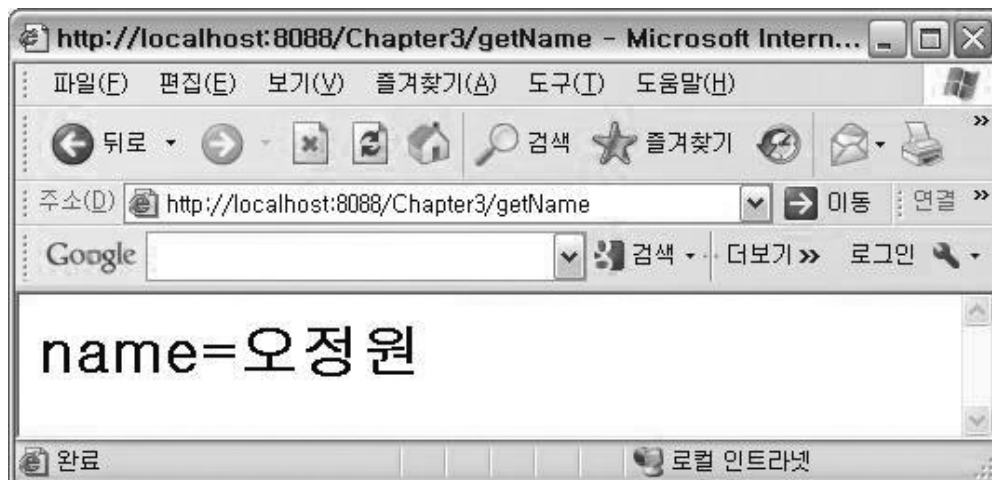
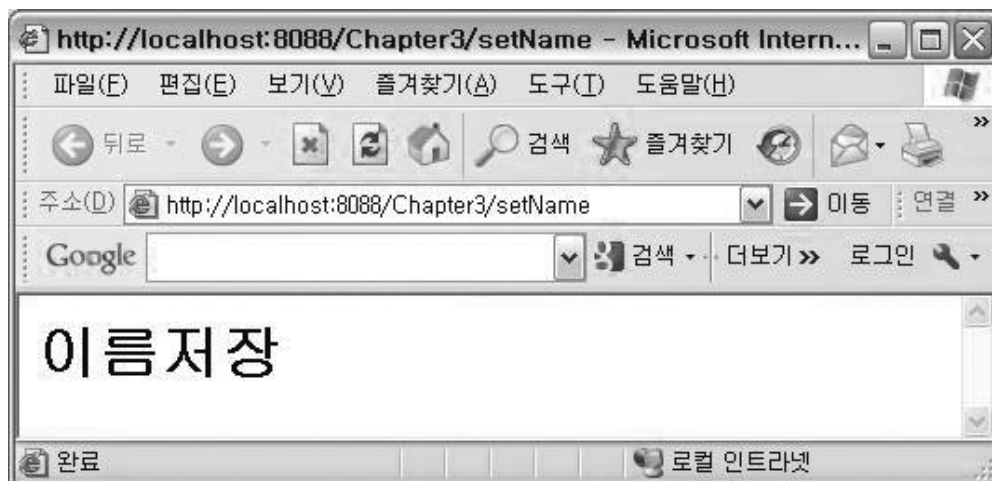
Chapter3\Wex5\GetNameServlet.java

```
1 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
2 ServletException, IOException {
3     // TODO Auto-generated method stub
4     HttpSession session = request.getSession();
5     String name=(String)session.getAttribute("name");
6     response.setContentType("text/html;charset=euc-kr");
7     PrintWriter out = response.getWriter();
8     out.println("<h1>name="+name+"</h1>");
9 }
```



서블릿에서 세션 살펴보기

■ 결과 확인



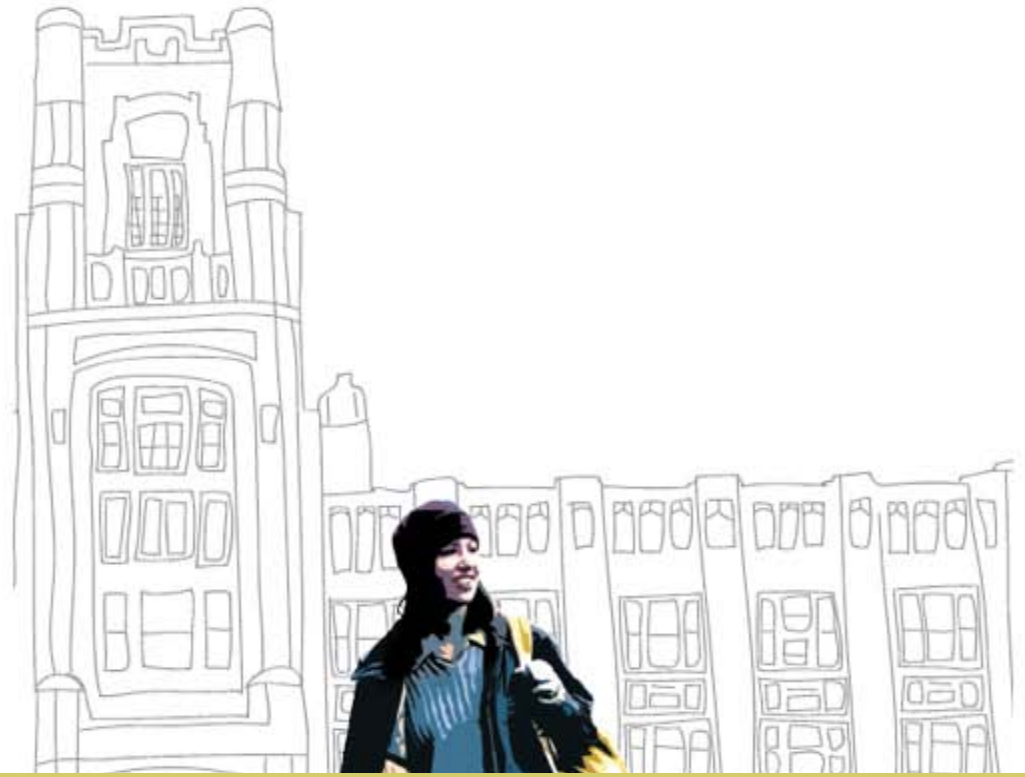
서블릿에서 세션 살펴보기

■ 동작 과정



[그림 3-5] 세션 객체를 생성하여 자신의 세션 객체에 속성 저장하고 사용하기





Thank You