

Chapter 06

액션 태그 (action tag)

JSP 액션 태그의 개요

■ JSP 액션 태그 (action tag)

- JSP 문서를 통해 컨테이너에 작업을 요청하는 또다른 방법

- ▶ Java 코드를 대신해 사용

- 스크립트 내부에서는 사용할 수 없음

- ▶ 액션 태그의 용도

- 페이지의 이동
- 페이지를 모듈화
- 자바 빈(Been)의 사용
- 커스텀 태그의 사용

구분	액션 태그	설명
제어 이동	<jsp:forward>	다른 JSP 문서로 제어를 이동
모듈화	<jsp:include>	다른 JSP 문서를 현재의 JSP문서에 포함
자바빈 사용	<jsp:useBean>	자바 빈 객체의 생성
	<jsp:setProperty>	자바 빈 인스턴스의 속성 값을 지정
	<jsp:getProperty>	자바 빈 인스턴스 속성에서 속성 값을 추출

forward 액션 태그

forward 액션 태그

■ 제어의 이동

- 하나의 웹 문서가 다른 웹 문서를 호출하여 브라우저에서 실행되도록 하는 것
 - ▶ 하나의 문서가 다른 문서를 호출하는 구문을 실행함으로써 제어의 이동이 발생됨
 - 사용자가 실행하는 것이 아니라 문서가 다른 문서를 호출해 실행하는 것임
 - [예] 게시판에 글을 입력한 후 [저장]을 실행하면, 글이 저장되고, 게시 글의 리스트를 출력하는 문서가 실행됨
 - ▶ 웹 어플리케이션은 많은 제어의 이동으로 기능을 수행함
- 제어를 이동시키기 위한 다양한 방법이 존재함
 - ▶ 서블릿을 사용
 - RequestDispatcher 클래스를 사용
 - ▶ JSP를 사용
 - response 객체의 sendRedirect() 메서드를 사용하는 방법
 - pageContext 객체의 forward() 메서드를 사용하는 방법
 - 액션 태그(action tag)를 사용하는 방법
 - ▶ 자바스크립트를 사용
 - location 객체의 replace() 메서드나 href 속성을 사용

forward 액션 태그의 개요

■ <jsp:forward> 액션 태그

```
<jsp:forward page="이동할 문서(또는 URL)"/>
```

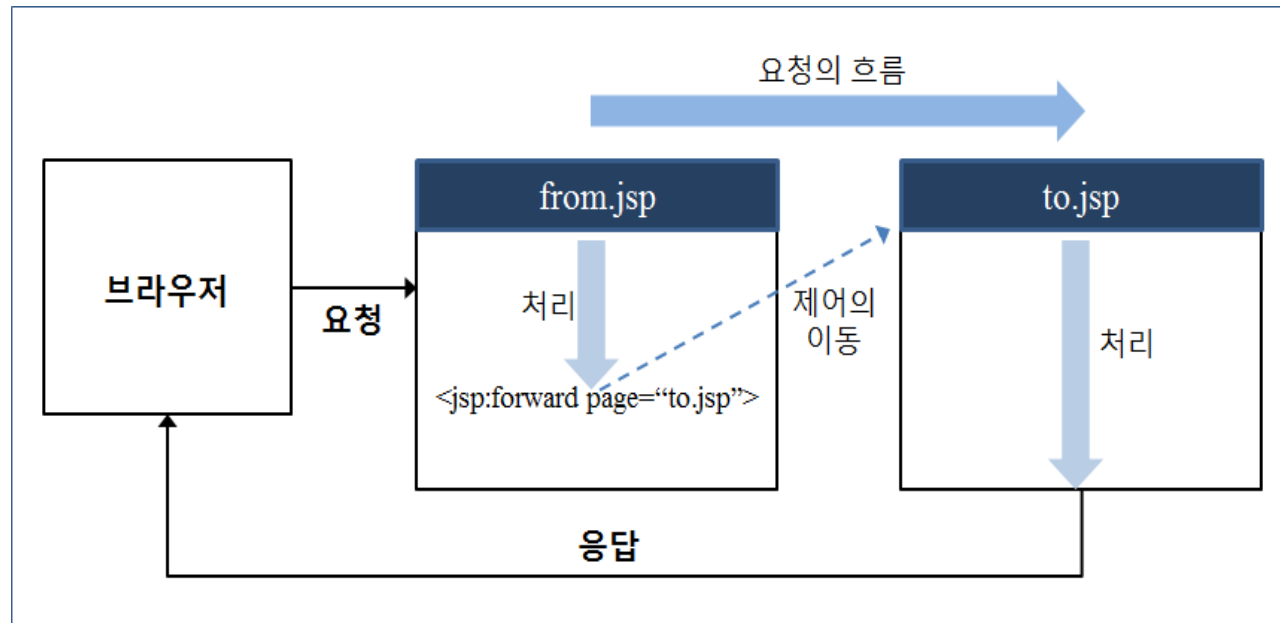
- 현재의 제어를 page 속성의 값으로 지정된 문서나 URL로 이동
 - ▶ 이동할 문서나 URL은 page 속성의 값으로 지정
 - ▶ <jsp:forward 사이에 공백이 있어서는 안됨
 - ▶ <jsp:forward> 액션 태그는 스크립트릿 내부에는 사용할 수 없음
 - ▶ [예] 현재의 JSP 문서에서 현재 폴더에 있는 target.jsp 문서로 제어를 이동시키는 경우

```
<jsp:forward page="/target.jsp"/>
```

```
<% String targetURL = "/target.jsp" %>  
<jsp:forward page = "<%= targetURL %>" />
```

forward 액션 태그의 개요

■ <jsp:forward> 액션 태그의 동작



- ▶ from.jsp 문서를 처리하는 도중 <jsp:forward> 구문을 만날 경우
 - page 속성이 나타내는 to.jsp 문서로 제어가 이동되어 to.jsp 문서가 처리됨
- ▶ 요청을 받은 문서는 from.jsp이지만 응답을 하는 문서는 to.jsp 문서임
 - 제어가 to.jsp 문서로 이동될 경우 제어는 from.jsp 문서로 다시 돌아오지 않음

forward 액션 태그의 개요

■ 제어를 이동하는 메서드의 비교

제어 이동 방법	차이점	공통점
response객체의 sendRedirect()	이동의 주체가 브라우저임 - sendRedirect() 메서드가 실행되면 브라우저가 컨테이너에 이동을 요청 브라우저의 주소표시줄에 이동된 문서(URL)이 나타남	제어를 이동시킴
pageContext객체의 forward()	이동의 주체가 컨테이너임 - forward() 메서드가 forward 액션 태그 실행되면 컨테이너가 제어를 이동시킴	
forward 액션 태그	브라우저의 주소표시줄에 이동되기 전의 문서(URL)이 여전히 출력됨 - 브라우저는 제어가 이동 된 것을 인지하지 못함	

forward 액션 태그와 버퍼

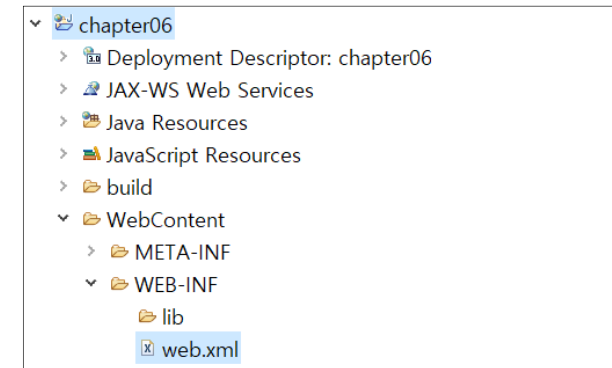
■ forward 액션 태그와 버퍼와의 관계

- forward 액션 태그는 제어를 이동하기 전에 `clear()` 메서드를 호출함
 - ▶ 이전에 버퍼에 존재했던 모든 내용이 제거됨
 - ▶ forward 액션 태그를 수행하기 전에 `flush()` 메서드를 호출할 경우 `IOException`이 발생함
 - forward 액션 태그는 수행되기 전 `clear()`를 호출하며, `clear()`가 수행된 후 `flush()`가 수행되면 `IOException`이 발생함
 - ▶ `autoflush`가 발생될 경우 `flush()` 메서드를 호출하지 않았더라도 `IOException`이 발생
- JSP 규약은 브라우저에 문자가 전송된다면 forward 액션 태그를 사용해 제어를 이동할 수 정의하고 있음
 - ▶ 이러한 이유로 forward 액션 태그는 수행되기 전 `clear()` 메서드를 호출함
- 결론적으로
 - ▶ 특정 내용을 브라우저에 출력한 후 다른 문서로 제어를 이동시키기 위해 사용하면 안됨
 - ▶ 브라우저에 어떤 내용도 출력하지 않고 순수하게 제어 만을 이동시키는 목적으로 사용해야 함

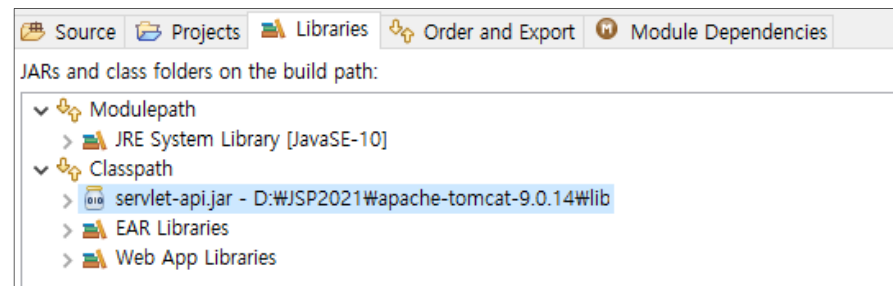
forward 액션 태그와 버퍼

■ 프로젝트 생성

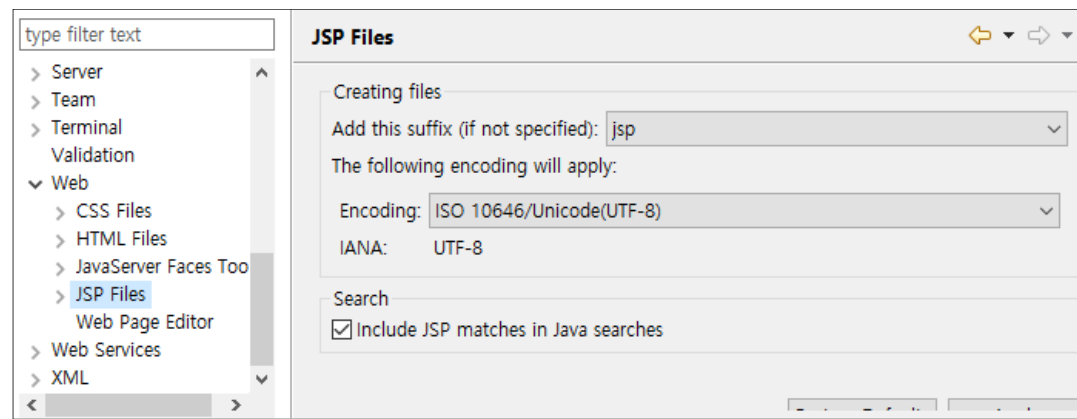
- 프로젝트 이름 : chapter06
 - ▶ web.xml 파일은 반드시 생성해야 함



- servlet-api 라이브러리 추가

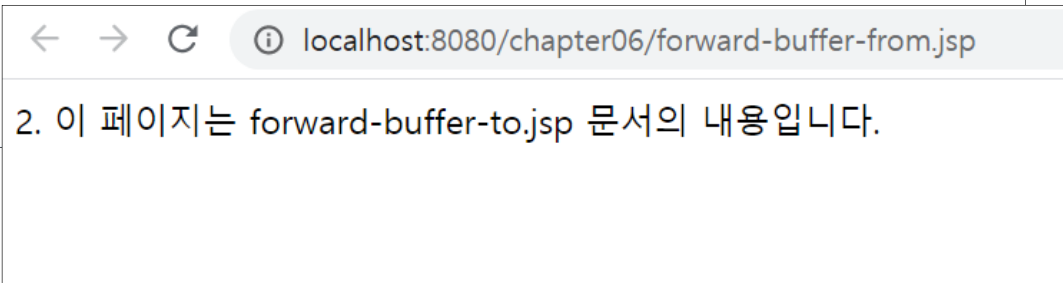


- JSP 문서 문자 집합 지정
 - ▶ UTF-8로 지정



```
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 </head>
8 <body>
9
10   1. 이 문장은 forward-buffer-from-1.jsp 문서의 내용입니다.<BR>
11
12   <jsp:forward page="forward-buffer-to.jsp"/>
13
14 </body>
15 </html>
```

```
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 </head>
8 <body>
9
10     2. 이 페이지는 06-forward-buffer-to.jsp 문서의 내용입니다.<BR>
11
12
13 </body>
14 </html>
15
```



forward 액션 태그는 이전의 버퍼 내용을 모두 제거하고 제어를 이동시킴
- forward-buffer-form-1.jsp 문서의 문장은 출력되지 않음

forward 액션 태그와 파라미터

■ 액션 태그에서 파라미터의 전달

```
<jsp:forward page="이동할 문서">  
  
    <jsp:param name="파라미터 이름1" value="파라미터 값1" />  
    <jsp:param name="파라미터 이름2" value="파라미터 값2" />  
    .....  
  
</jsp:forward>
```

- forward 액션 태그를 사용해 제어를 이동하면서 파라미터까지 함께 전달하는 것
 - ▶ 제어가 이동되는 문서로 이동하면서 파라미터를 함께 전달할 수 있음
- <jsp:param> 액션 태그
 - ▶ <jsp:forward> 액션 태그의 자식 요소로 전달할 파라미터를 지정
 - name 속성 - 파라미터의 이름을 지정
 - value 속성 - name 속성으로 지정한 파라미터의 값을 지정
 - ▶ 전달할 파라미터 수에는 제한이 없음

```
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 </head>
8 <body>
9
10 <%
11     String msg = request.getParameter("msg");
12     if(msg != null){
13         out.println("<H2>" + msg + "</h2>");
14     }
15 %>
16
17 <form action="forward-param-to-1.jsp" method="post">
18
19     이름 <input type="text" name="username">
20     <input type="submit" value="SAVE">
21
22 </form>
23
24 </body>
25 </html>
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3
4 <%! String msg="아이디를 입력하지 않았습니다."; %>
5
6 <% request.setCharacterEncoding("UTF-8"); %>
7
8 <%
9     String username = request.getParameter("username");
10    if(username.length() == 0) {
11 %>
12
13 <jsp:forward page="forward-param-from-1.jsp">
14     <jsp:param name="msg" value="<%=msg%>" />
15 </jsp:forward>
16
17 <%
18     }
19 %>
20 <h1>환영합니다. <%=username%> 님.</h1>
```

forward 액션 태그와 파라미터

forward-param-to-1.jsp

← → ↻ ⓘ localhost:8080/chapter06/forward-param-from-1.jsp ☆

이름

← → ↻ ⓘ localhost:8080/chapter06/forward-param-to-1.jsp ☆

아이디를 입력하지 않았습니다.

이름

← → ↻ ⓘ localhost:8080/chapter06/forward-param-to-1.jsp ☆

아이디를 입력하지 않았습니다.

이름

← → ↻ ⓘ localhost:8080/chapter06/forward-param-to-1.jsp ☆

환영합니다. 홍길동 님.

forward 액션 태그와 파라미터

■ 파라미터로 한글 문자열을 전달하는 경우

- 문자열을 전달한 JSP 문서
 - 인코딩하여 전달해야 함
 - URLEncoder 클래스의 encode() 메서드를 사용
- 문자열을 전달받은 JSP 문서
 - 디코딩하여 브라우저에 출력해야 함
 - URLDecoder 클래스의 decode() 메서드를 사용
- 인코딩과 디코딩이 이루어지지 않으면 한글 문자는 정상적으로 출력되지 않음

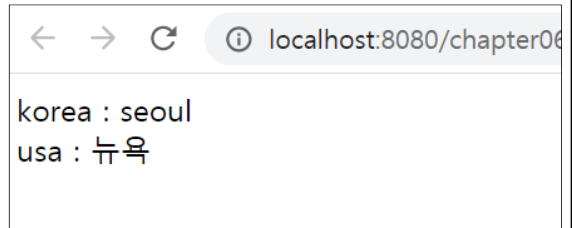
forward 액션 태그와 파라미터

forward-param-form-2.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.net.URLEncoder" %>
4
5 <jsp:forward page="forward-param-to-2.jsp">
6
7   <jsp:param name="korea" value='seoul' />
8   <jsp:param name="usa" value='<%=URLEncoder.encode("뉴욕", "UTF-8")%>' />
9
10 </jsp:forward>
```

forward-param-to-2.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.net.URLDecoder"%>
4
5 <%
6   String parm1 = request.getParameter("korea");
7   String parm2 = URLDecoder.decode(request.getParameter("usa"));
8
9   out.println("korea : "+ parm1+"<br>");
10  out.println("usa : "+ parm2);
11 %>
```



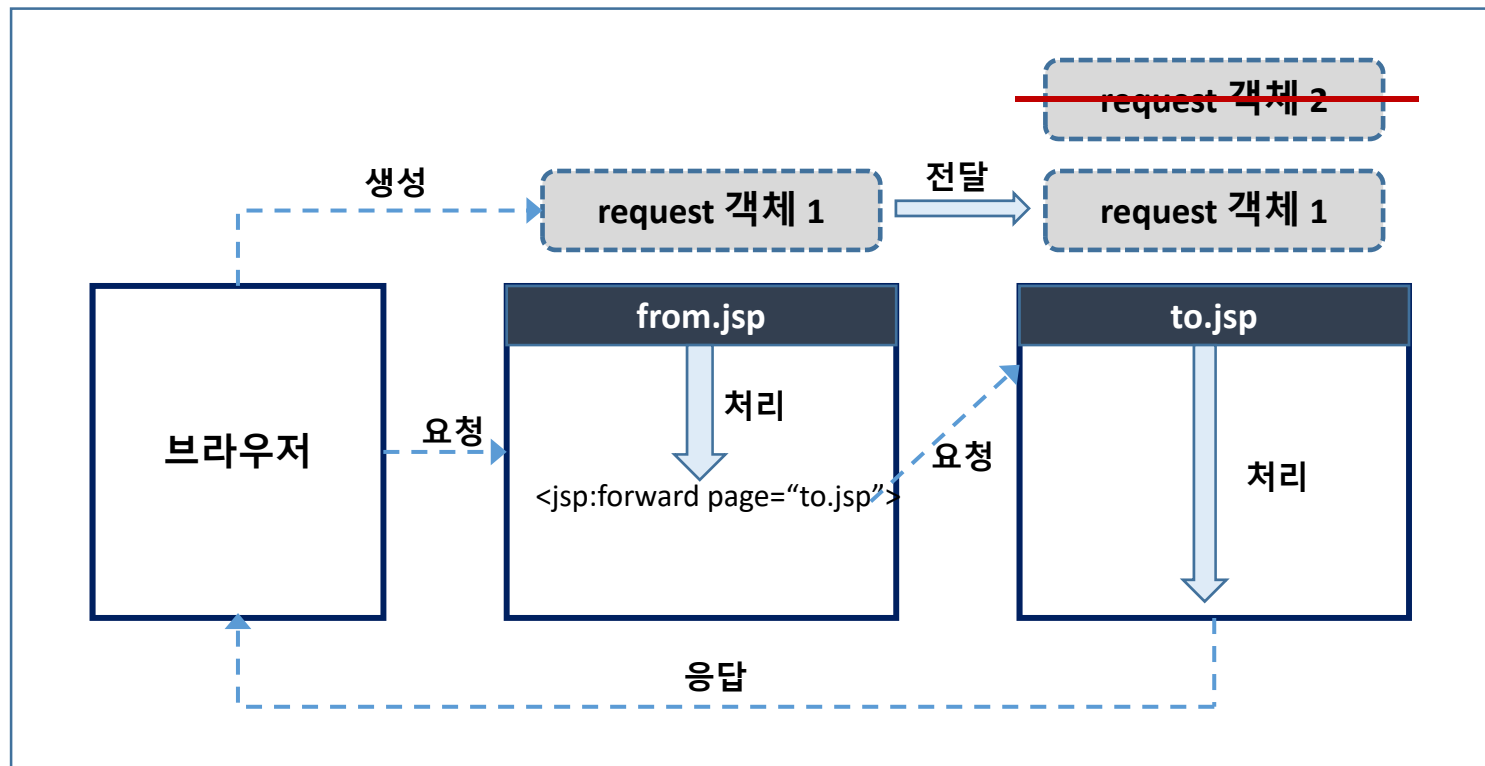
localhost:8080/chapter06

korea : seoul
usa : 뉴욕

forward 액션 태그와 request 객체

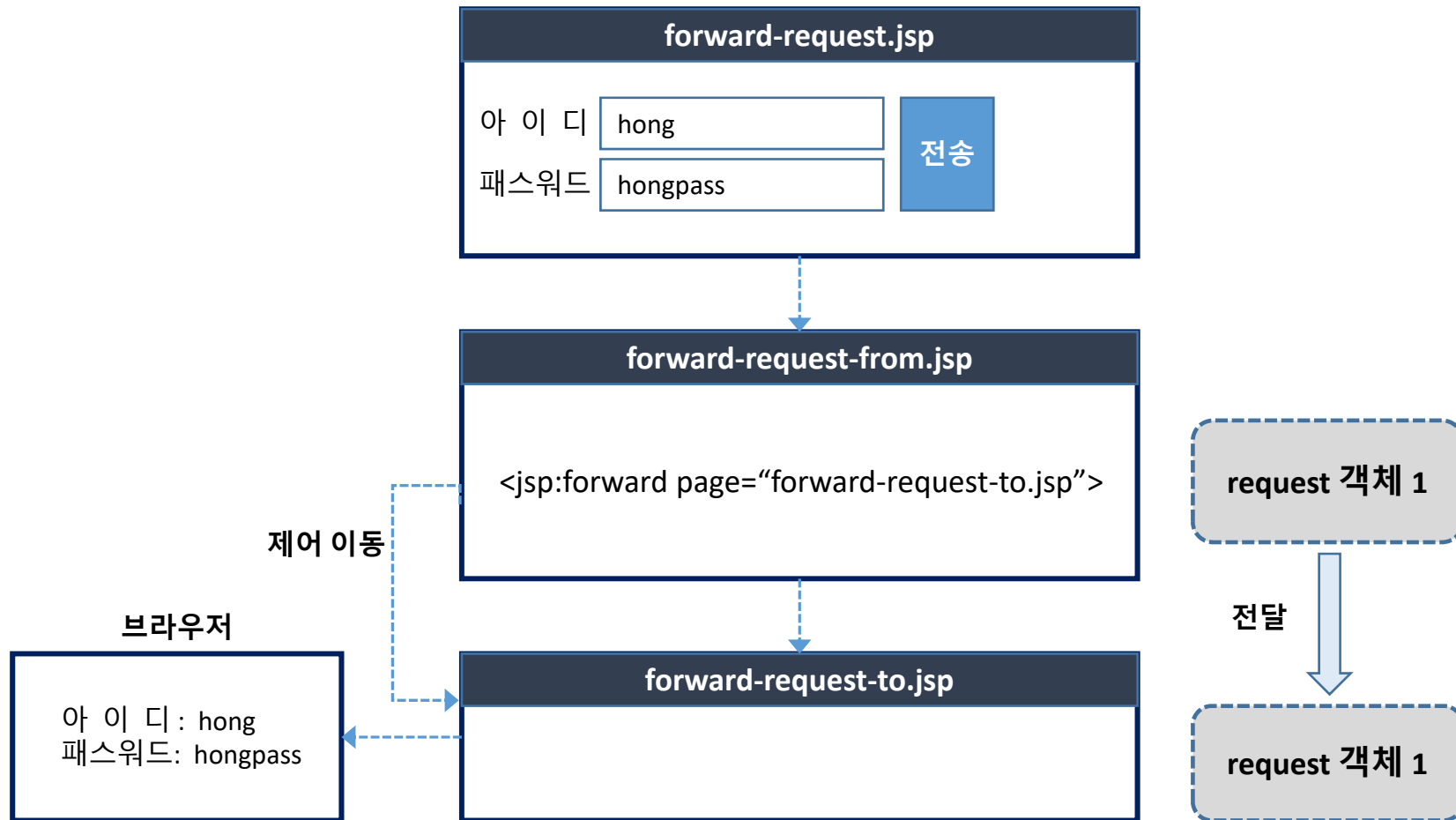
■ forward 액션 태그와 request 내장 객체의 관계

- 액션 태그에 의해 제어가 이동된 경우 기존에 생성되었던 request 내장 객체가 그대로 전달됨
 - ▶ 제어가 변경되더라도 새로운 request 객체 생성되지 않음



forward 액션 태그와 request 객체

■ [실습] forward 액션 태그의 request 객체 전달



forward 액션 태그와 request 객체

forward-request.jsp

```
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 </head>
8 <body>
9   <form name="myform" method=post action="forward-request-from.jsp">
10    <fieldset>
11      <legend>회원로그인</legend>
12      <table>
13        <tr>
14          <td>아 이 디 </td>
15          <td><input type=text name="id" size=20></td>
16          <td><input type=submit value="전송"></td>
17        </tr>
18        <tr>
19          <td>패스워드</td>
20          <td><input type=password name="pw" size=20></td>
21          <td><input type=reset value="취소"></td>
22        </tr>
23      </table>
24    </fieldset>
25  </form>
26 </body>
27 </html>
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"  
2     pageEncoding="UTF-8"%>  
3  
4 <jsp:forward page="forward-request-to.jsp"/>
```

forward 액션 태그와 request 객체

forward-request-to.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%
4   request.setCharacterEncoding("utf-8");
5
6   String id = request.getParameter("id");
7   String pw = request.getParameter("pw");
8 %>
9 <!DOCTYPE html>
10 <html>
11 <head>
12   <meta charset="UTF-8">
13 </head>
14 <body>
15   <fieldset>
16     <legend>회원로그인</legend>
17     <table>
18       <tr>
19         <td>아 이 디 </td>
20         <td> <%=id%> </td>
21       </tr>
22       <tr>
23         <td>패스워드</td>
24         <td> <%=pw%> </td>
25       </tr>
26     </table>
27   </fieldset>
28 </body>
29 </html>
```

회원로그인

아 이 디

패스워드

회원로그인

아 이 디 홍길동

패스워드 hong1234

제어를 이동하는 방법

■ 제어를 이동하는 방법

- response 객체의 sendRedirect() 메서드를 사용하는 방법

```
response.sendRedirect(이동할 문서나 URL)
```

- pageContext 객체의 forward() 메서드를 사용하는 방법

```
pageContext.forward(이동할 문서나 URL)
```

- 자바스크립트를 사용하는 방법

- ▶ location 객체의 replace() 메서드를 사용하는 방법

```
location.replace(이동할 문서나 URL)
```

- ▶ location 객체의 href 속성을 사용하는 방법

```
location.href=이동할 문서나 URL
```

제어를 이동하는 방법

- RequestDispatcher 클래스의 `getRequestDispatcher()` 메소드를 사용하는 방법

```
RequestDispatcher dispatch = request.getRequestDispatcher(이동할 문서나 URL);  
dispatch.forward(request, response);
```


제어를 이동하는 방법

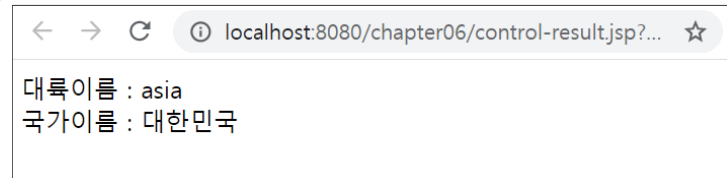
■ response 객체의 sendRedirect() 메서드를 사용하는 방법

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ page import="java.net.URLEncoder"%>
4
5 <%
6     String cnt = "asia";
7     String nation = URLEncoder.encode("대한민국", "UTF-8");
8
9     response.sendRedirect("control-result.jsp?cnt=" + cnt + "&nation=" + nation);
10 %>
```

control-sendRedirect.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%
4
5     String cnt = request.getParameter("cnt");
6     String nation = request.getParameter("nation");
7
8     out.println("대륙이름 : " + cnt + "<BR>");
9     out.println("국가이름 : " + nation);
10
```

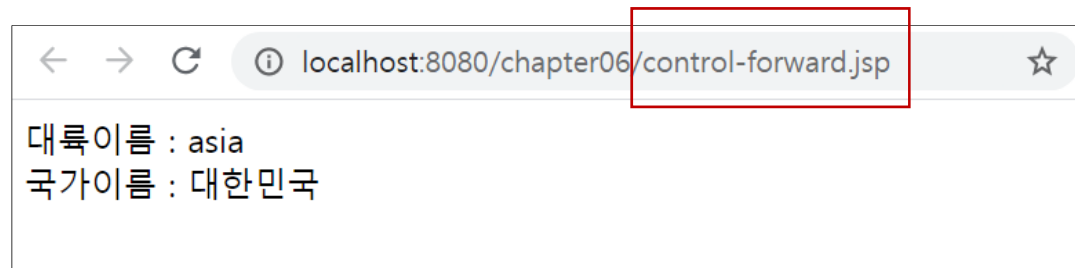
control-result.jsp



■ pageContext 객체의 forward() 메서드를 사용하는 방법

- forward 액션 태그를 사용하는 방법과 동일한 효과

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.net.URLEncoder"%>
4
5 <%
6   String cnt = "asia";
7   String nation = URLEncoder.encode("대한민국", "UTF-8");
8
9   pageContext.forward("control-result.jsp?cnt=" + cnt + "&nation=" + nation);
10 %>
```



■ 자바스크립트를 사용하는 방법

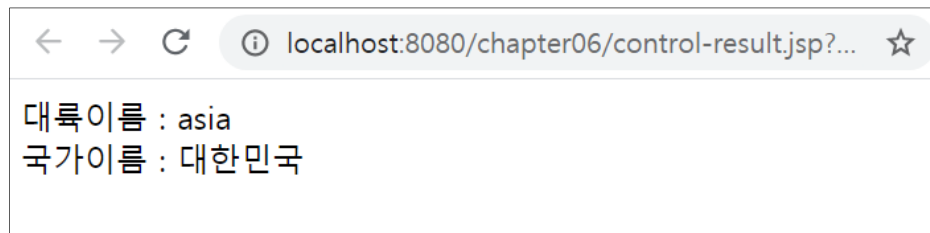
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.net.URLEncoder"%>
4
5 <%
6   String cnt = "asia";
7   String nation = URLEncoder.encode("대한민국", "UTF-8");
8 %>
9
```

```
10 <script type="text/javascript">
11   location.replace("<%= "control-result.jsp?cnt=" + cnt + "&nation=" + nation%>");
12 </script>
```



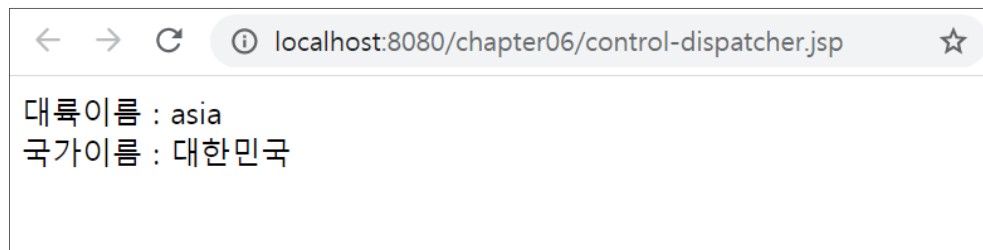
동일한 기능 수행

```
10 <script type="text/javascript">
11   location.href = "<%= "control-result.jsp?cnt=" + cnt + "&nation=" + nation%>";
12 </script>
```



■ RequestDispatcher 클래스의 getRequestDispatcher() 메서드를 사용하는 방법

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.net.URLEncoder"%>
4
5 <%
6   String cnt = "asia";
7   String nation = URLEncoder.encode("대한민국", "UTF-8");
8
9   String str = "control-result.jsp?cnt=" + cnt + "&nation=" + nation;
10
11   RequestDispatcher dispatch = request.getRequestDispatcher(str);
12   dispatch.forward(request, response);
13
14 %>
```

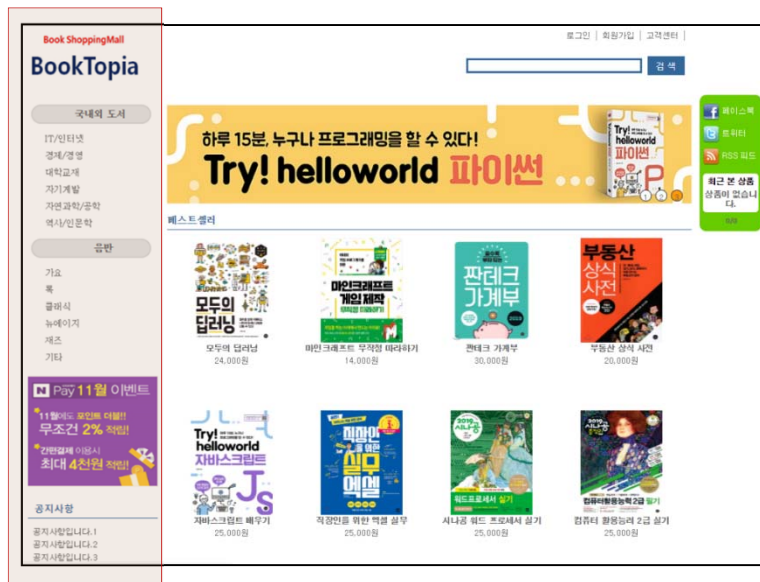


include 액션 태그

include 액션 태그의 개요

■ include 액션 태그

- 웹 페이지들에 공통 사용을 목적으로 생성되어 있는 모듈을 각 문서에 삽입하기 위한 액션 태그
 - ▶ include 지시어와 유사
 - ▶ 유지 보수 및 재사용성을 높일 수 있음



include 액션 태그의 개요

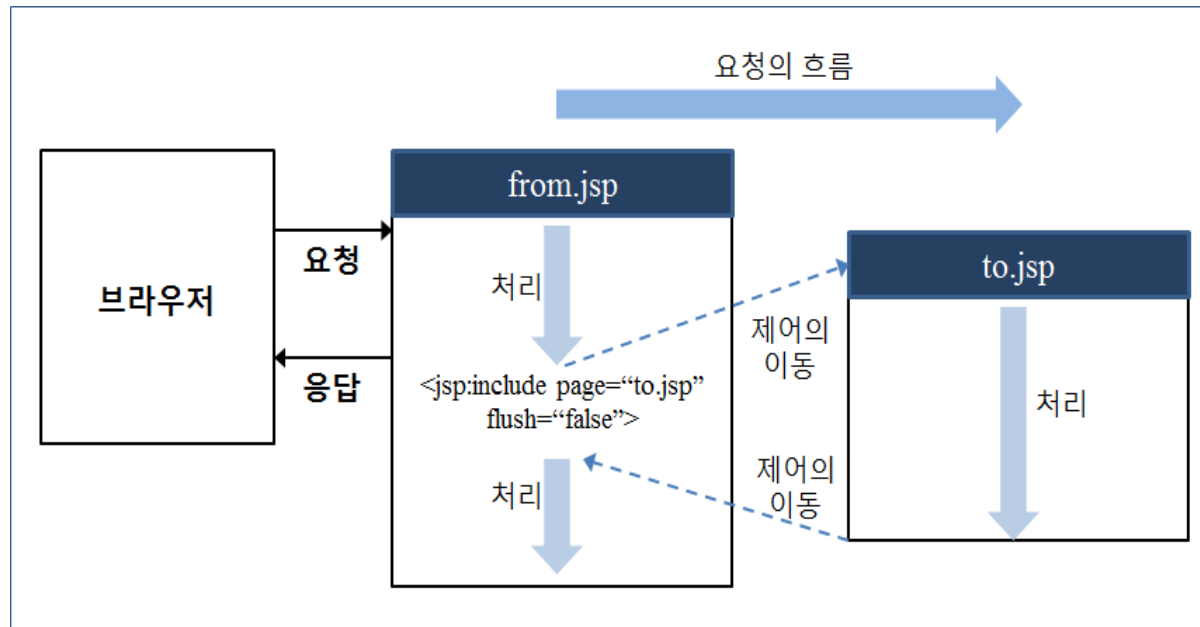
■ include 액션 태그

```
<jsp:include page="포함할 문서" flush="true | false">
```

- include 지시어와 유사하게 한 문서 내에 다른 문서를 포함시키는 기능 수행
- flush 속성의 값이 true인 경우
 - ▶ 현재 버퍼의 내용을 브라우저로 전송해 버퍼를 비운 다음 제어가 이동
 - ▶ 제어가 이동된 문서에서 브라우저에 전송될 내용이 버퍼에 저장
 - ▶ 기본 값은 true
- flush 속성이 false인 경우
 - ▶ 현재 버퍼의 내용을 브라우저로 전송하지 않고 버퍼의 내용을 유지한 상태로 제어가 이동
 - ▶ 제어가 이동된 문서에서 브라우저에 전송될 내용이 버퍼의 기존 내용에 추가됨
- 결론적으로, include 액션태그는 clear() 메서드를 호출하지 않고 현재 버퍼의 내용을 브라우저에 출력함

include 액션 태그의 개요

■ include 액션 태그의 동작



- `<jsp:include>` 액션 태그는 이동되었던 제어가 다시 돌아옴
 - ▶ `<jsp:include>` 액션 태그는 넘겨준 제어를 다시 전달받아 나머지 부분을 수행
 - `<jsp.forward>` 액션 태그는 순수하게 제어 만을 이동시킴
 - ▶ 브라우저에 응답하는 문서는 `to.jsp` 문서가 아니라 `from.jsp`임

include 액션 태그의 개요

■ forward 액션 태그와 include 액션 태그 차이

구분	forward 액션 태그	include 액션 태그
제어의 이동	순수하게 제어 만을 이동시키며 이동된 제어는 돌아오지 않음	제어를 이동시키지만 이동된 제어는 다시 돌아옴
브라우저 응답	이동이 이루어진 문서에서 응답	제어가 돌아오기 때문에 <jsp:include> 액션 태그를 수행한 문서에서 응답
기존 버퍼의 내용	버퍼의 모든 내용을 강제로 비운 다음 제어가 이동 (수행되기 전 clear() 메서드를 호출)	브라우저로 전송해 버퍼를 모든 내용을 비우거나 전송하지 않고 버퍼의 내용을 그대로 유지한 채 제어가 이동
브라우저 출력	문서의 어떤 내용도 브라우저에 출력되지 않음	문서의 내용은 브라우저에 출력됨

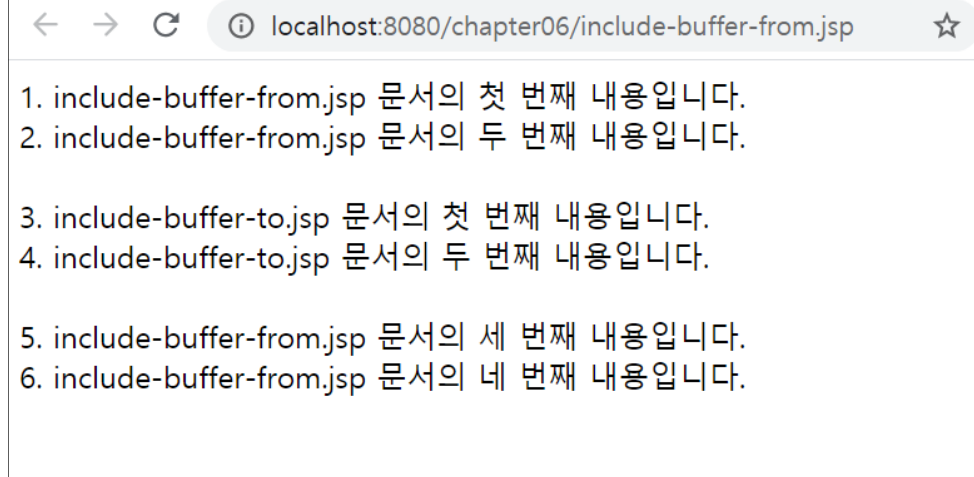
include 액션 태그의 개요

■ include 액션 태그와 include 지시어의 차이

구분	include 액션 태그	include 디렉티브
기능	레이아웃 모듈화	레이아웃 모듈화
서블릿의 생성	포함하는 파일과 포함되는 파일이 별도로 컴파일됨	포함하는 파일과 포함되는 파일을 결합해 하나의 파일로 컴파일됨
결과 처리	포함되는 문서의 실행 결과만을 포함	포함되는 문서의 소스 코드를 삽입하여 하나의 문서로 구성해 처리

```
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 </head>
8 <body>
9
10     1. include-buffer-from.jsp 문서의 첫 번째 내용입니다.<br>
11     2. include-buffer-from.jsp 문서의 두 번째 내용입니다.<br><br>
12
13     <jsp:include page="include-buffer-to.jsp" flush="false"/>
14
15     5. include-buffer-from.jsp 문서의 세 번째 내용입니다.<br>
16     6. include-buffer-from.jsp 문서의 네 번째 내용입니다.<br>
17
18 </body>
19 </html>
```

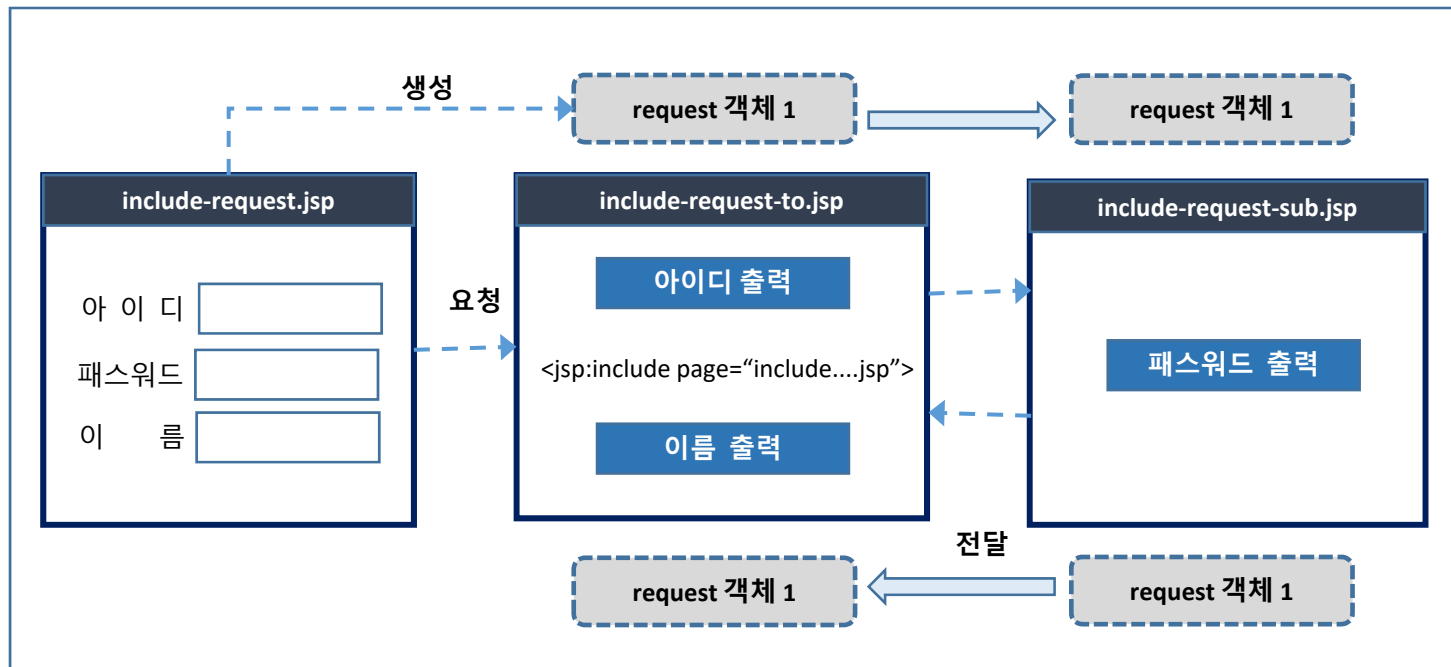
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6     <meta charset="UTF-8">
7 </head>
8 <body>
9
10     3. include-buffer-to.jsp 문서의 첫 번째 내용입니다.<br>
11     4. include-buffer-to.jsp 문서의 두 번째 내용입니다.<br><br>
12
13 </body>
14 </html>
```



include 액션 태그와 request 객체

■ include 액션 태그와 request 객체와의 관계

- forward 액션 태그처럼 생성된 request 객체는 제어가 이동되는 객체로 전달됨
 - ▶ forward 액션 태그로 제어가 이동될 때 새로운 request 객체가 생성되지 않고 기존의 request 객체가 전달됨
- forward 액션 태그는 이동되었던 제어가 다시 돌아옴
 - ▶ 제어가 이동되었던 문서로 전달된 request 객체가 다시 돌아옴



include 액션 태그와 request 객체

include-request.jsp

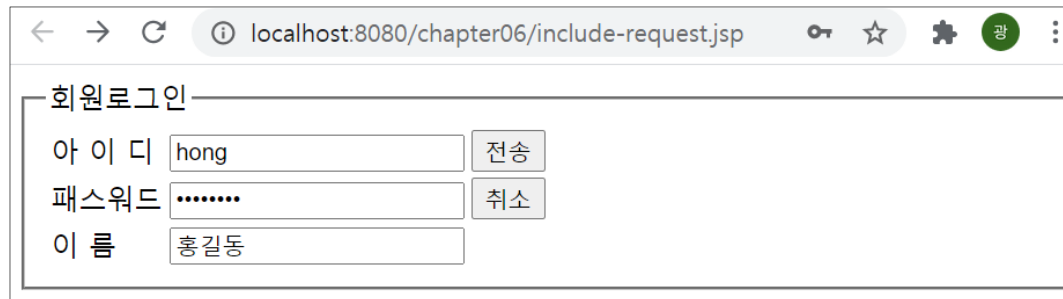
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6     <meta charset="UTF-8">
7 </head>
8 <body>
9     <form name="myform" method=post action="include-request-to.jsp">
10     <fieldset>
11     <legend>회원로그인</legend>
12     <table>
13     <tr>
14         <td>아이디</td>
15         <td><input type=text name="id" size=20></td>
16         <td><input type=submit value="전송"></td>
17     </tr>
18     <tr>
19         <td>패스워드</td>
20         <td><input type=password name="pw" size=20></td>
21         <td><input type=reset value="취소"></td>
22     </tr>
23     <tr>
24         <td>이름</td>
25         <td><input type=text name="name" size=20></td>
26         <td>&nbsp;</td>
27     </tr>
28     </table>
29 </fieldset>
30 </form>
31 </body>
32 </html>
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4 <%
5   request.setCharacterEncoding("UTF-8");
6
7   String id = request.getParameter("id");
8   out.print("아이디 : " + id + "<BR>");
9 %>
10
11 <jsp:include page="include-request-sub.jsp" flush="false"/>
12
13 <%
14   String name = request.getParameter("name");
15   out.print("이름 : " + name + "<BR>");
16 %>
17
```

include 액션 태그와 request 객체

include-request-sub.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%
4
5   request.setCharacterEncoding("UTF-8");
6
7   String pw = request.getParameter("pw");
8   out.println("패스워드 : " + pw + "<BR>");
9
10 %>
```

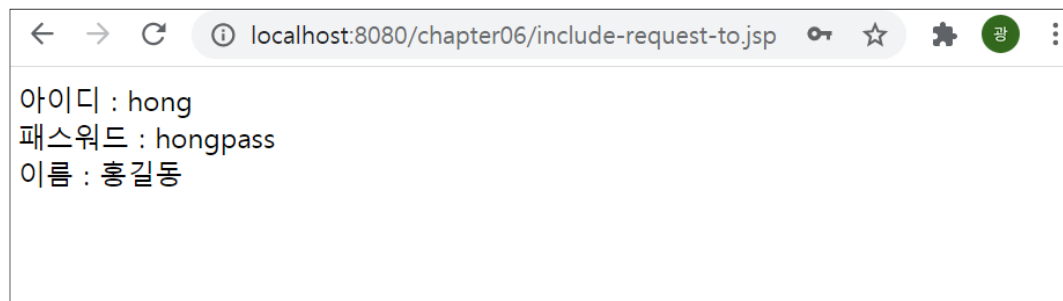


회원로그인

아이디 전송

패스워드 취소

이름



아이디 : hong
패스워드 : hongpass
이름 : 홍길동

include 액션 태그와 request 객체

■ include 액션 태그를 이용한 파라미터 전달

- 기본적으로 forward 액션 태그와 거의 동일
 - flush 속성의 사용을 제외하고 forward 액션 태그와 동일

```
<jsp:include page="이동할 문서">  
    <jsp:param name="파라미터 이름1" value="파라미터 값1" />  
    <jsp:param name="파라미터 이름2" value="파라미터 값2" />  
    .....  
</jsp:include>
```

include 액션 태그와 request 객체

include-param.jsp

```
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 </head>
8 <body>
9   <form name="myform" method=post action="include-param-to.jsp">
10  <fieldset>
11    <legend>회원로그인</legend>
12    <table>
13      <tr>
14        <td>아이디</td>
15        <td><input type=text name="id" size=20></td>
16        <td><input type=submit value="전송"></td>
17      </tr>
18      <tr>
19        <td>패스워드</td>
20        <td><input type=password name="pw" size=20></td>
21        <td><input type=reset value="취소"></td>
22      </tr>
23      <tr>
24        <td>이름</td>
25        <td><input type=text name="name" size=20></td>
26        <td>&nbsp;</td>
27      </tr>
28    </table>
29  </fieldset>
30 </form>
31 </body>
32 </html>
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.net.URLEncoder"%>
4
5 <%
6   request.setCharacterEncoding("UTF-8");
7
8   String id = request.getParameter("id");
9   out.print("아이디 : " + id + "<BR>");
10 %>
11
12 <jsp:include page="include-param-sub.jsp" flush="false">
13   <jsp:param name='nation' value='<%=URLEncoder.encode("한국","UTF-8")%>' />
14 </jsp:include>
15
16 <%
17   String pw = request.getParameter("pw");
18   String name = request.getParameter("name");
19
20   out.print("패스워드 : " + pw + "<BR>");
21   out.print("이름 : " + name + "<BR>");
22 %>
23
```

include 액션 태그와 request 객체

include-param-sub.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.net.URLDecoder"%>
4
5 <!DOCTYPE html>
6 <%
7   request.setCharacterEncoding("utf-8");
8   String nation = URLDecoder.decode(request.getParameter("nation"));
9
10  out.print("국적 : " + nation + "<BR>");
11
12 %>
```

The image shows two browser screenshots. The top screenshot displays a login form titled '회원로그인' (Member Login) with fields for '아이디' (ID) containing 'hong', '패스워드' (Password) with masked characters, and '이름' (Name) containing '홍길동'. There are '전송' (Submit) and '취소' (Cancel) buttons. The bottom screenshot shows the output of the JSP page: '아이디 : hong', '국적 : 한국' (Nationality: Korea), '패스워드 : hong1234', and '이름 : 홍길동' (Name: Hong Gil-dong).

수고하셨습니다