

13장 자바 코드를 없애는 액션 태그

13.1 인클루드 액션 태그 사용하기

13.2 포워드 액션 태그 사용하기

13.3 useBean, setProperty, getProperty 액션 태그
사용하기

13.1 인클루드 액션 태그 사용하기

액션 태그 등장 배경

- 화면이 복잡해짐에 따라 디자이너는 HTML 태그에 자바 코드를 같이 써야 하는 불편이 생김
- 따라서 스크립트릿의 자바 코드를 대체하는 액션 태그가 등장하게 됨

JSP의 여러 가지 액션 태그

이름	형식	설명
인클루드 액션 태그	<jsp:include>	이미 있는 JSP를 현재 JSP에 포함하는 태그
포워드 액션 태그	<jsp:forward>	서블릿에서 <code>RequestDispatcher</code> 클래스의 포워딩 기능을 대신하는 태그
유즈빈 액션 태그	<jsp:useBean>	객체를 생성하기 위한 <code>new</code> 연산자를 대신하는 태그
셋프로퍼티 액션 태그	<jsp:setProperty>	<code>setter</code> 를 대신하는 태그
겟프로퍼티 액션 태그	<jsp:getProperty>	<code>getter</code> 를 대신하는 태그

13.1 인클루드 액션 태그 사용하기

인클루드 액션 태그(Include Action Tag)

- 인클루드 디렉티브 태그처럼 화면을 분할해서 관리하는데 사용되는 태그
- 화면의 유지 보수 및 재사용성을 높일 수 있음

쇼핑몰 메인 화면

쇼핑몰 상품 상세 화면

정가	30,000원
판매가	27,000원 (10% 할인)
포인트 적립	2000P (10% 적립)
포인트 추가 적립	만원이상 구매시 1,000P, 5만원이상 구매시 2,000P 추가 적립. 권역점 배송 비용시 300P 추가 적립
발행일	2018-10-02
페이지 수	996쪽
ISBN	2323454566778
배송료	무도
배송안내	[당일배송] 당일배송 서비스 시작! [휴일배송] 휴일에도 배송받는 Bookshop
도착예정일	지금 주문 시 내일 도착 예정
수량	1

13.1 인클루드 액션 태그 사용하기

인클루드 액션 태그 형식

```
<jsp:include page="jsp페이지" flush="true 또는 false">  
...  
</jsp:include>
```

-page : 포함할 JSP 페이지명

-flush : 지정한 JSP를 실행하기 전 출력 버퍼 비움 여부 지정

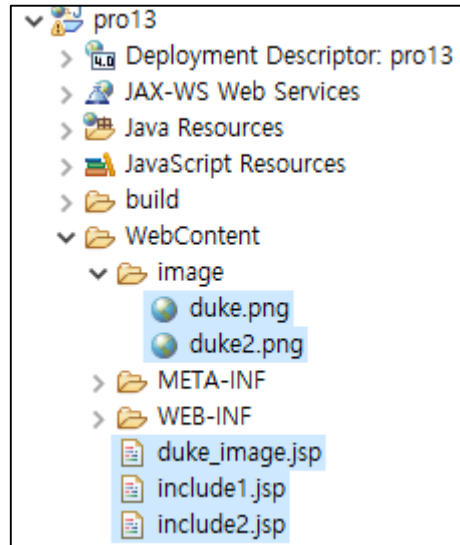
인클루드 액션 태그와 인클루드 디렉티브 태그 비교

항목	인클루드 액션 태그	인클루드 디렉티브 태그
기능	JSP 레이아웃 모듈화	JSP 레이아웃 모듈화
처리시간	요청 시간에 처리	JSP를 자바 코드로 변환 시 처리
데이터 처리 방법	param 액션 태그를 이용해 동적 처리 가능	정적 처리만 가능
포함된 JSP 자바 파일 변환 여부	포함되는 JSP가 각각 자바 파일로 생성	포함되는 JSP가 포함하는 JSP에 합쳐진 후 한 개의 자바 파일로 생성

13.1 인클루드 액션 태그 사용하기

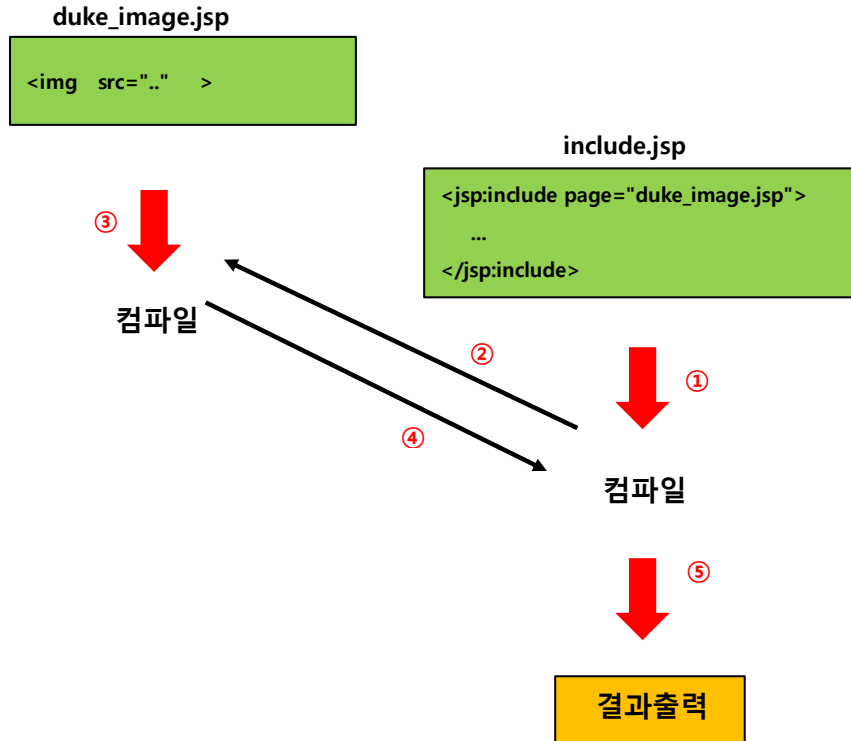
- 13.1.1 JSP 페이지에 이미지 포함 실습

1. 새 프로젝트 pro13을 만들고 다음과 같이 실습에 필요한 이미지 파일(duke.png, duke2.png)과 duke_image.jsp, include1.jsp, include2.jsp 파일을 추가합니다.



13.1 인클루드 액션 태그 사용하기

인클루드 액션 태그 처리 과정



- ① 브라우저 요청 시 JSP 파일을 컴파일합니다.
- ② 컴파일 시 `<jsp:include>`가 지시하는 JSP를 요청합니다.
- ③ 요청된 JSP를 컴파일합니다.
- ④ 컴파일된 JSP가 응답을 보냅니다.
- ⑤ JSP는 브라우저에서 요청한 응답 결과를 출력합니다.

13.1 인클루드 액션 태그 사용하기

2. 자식 JSP에 해당하는 duke_image.jsp를 다음과 같이 작성합니다.

코드 13-1 pro13/WebContent/duke_image.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");
    String name = request.getParameter("name" ) ;
    String imgName = request.getParameter("imgName" ) ;
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
    <title>듀크이미지</title>
</head>
<body>
    <br><br>
    <h1>이름은 <%=name %>입니다. </h1> <br><br>
    
</body>
</html>
```

param 액션 태그로 전달된 매개변수를
getParameter() 메서드를 이용해 가져
옵니다.

13.1 인클루드 액션 태그 사용하기

3. 부모 JSP인 include1.jsp를 다음과 같이 작성합니다.

코드 13-2 pro13/WebContent/include1.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

```
<%
```

```
request.setCharacterEncoding("utf-8");
```

```
%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>include1.jsp</title>
```

```
</head>
```

```
<body>
```

```
    안녕하세요. 쇼핑몰 중심 JSP 시작입니다!!!
```

```
    <br>
```

```
    <jsp:include page="duke_image.jsp" flush="true" >
```

```
        <jsp:param name="name" value="듀크" />
```

```
        <jsp:param name="imgName" value="duke.png" />
```

```
    </jsp:include>
```

```
    <br>
```

```
    안녕하세요. 쇼핑몰 중심 JSP 끝 부분입니다.!!!
```

```
</body>
```

포워딩합니다.

duke_image.jsp를 동적으로 전달합니다.

param 액션 태그를 이용해 duke_image.jsp로
이름과 파일 이름을 전달합니다.

13.1 인클루드 액션 태그 사용하기

4. 이번에는 다른 부모 JSP인 include2.jsp를 다음과 같이 작성합니다.

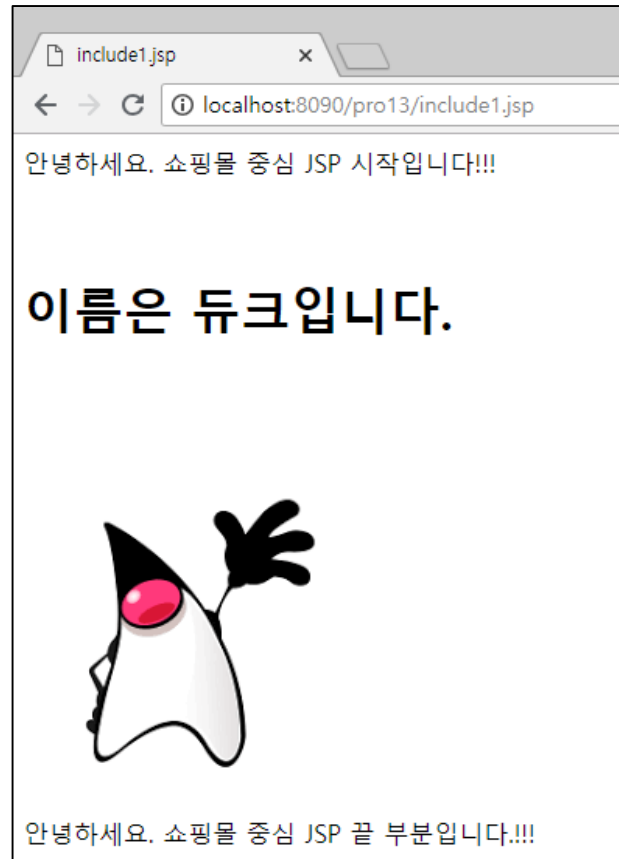
코드 13-3 pro13/WebContent/include2.jsp

```
...  
<body>  
    안녕하세요. 쇼핑몰 중심 JSP 시작입니다!!!  
    <br>  
    <jsp:include page="duke_image.jsp" flush="true" >  
        <jsp:param name="name" value="듀크2"/>  
        <jsp:param name="imgName" value="duke2.png"/>  
    </jsp:include>  
    <br>  
    안녕하세요. 쇼핑몰 중심 JSP 끝 부분입니다.!!!  
</body>  
</html>
```

param 액션 태그를 이용해 duke_image.jsp로
이름과 파일 이름을 전달합니다.

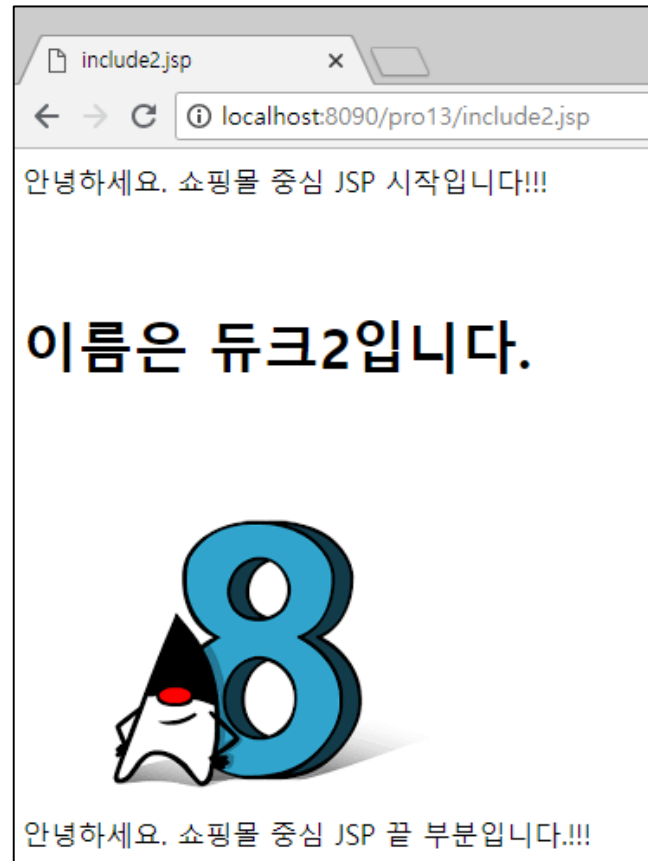
13.1 인클루드 액션 태그 사용하기

5. 브라우저에서 JSP 파일을 요청하면 각각 다른 이미지와 이름이 출력됩니다.
먼저 <http://localhost:8090/pro13/include1.jsp>로 요청합니다.



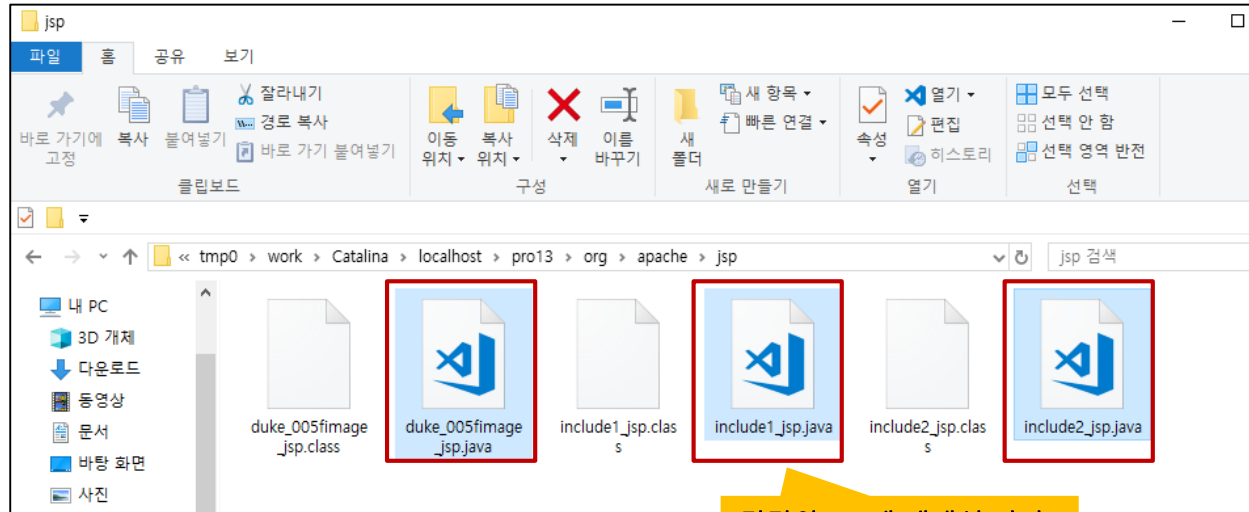
13.1 인클루드 액션 태그 사용하기

6. <http://localhost:8090/pro13/include2.jsp>로 요청하면 앞서와는 다른 이미지와 이름이 출력되는 것을 볼 수 있습니다.



13.1 인클루드 액션 태그 사용하기

7. 다음 경로의 윈도우 탐색기로 가면 JSP 파일이 자바 파일로 변환된 것을 볼 수 있습니다.



각각의 JSP에 대해서 자바 파일이 생성됩니다.

13.1 인클루드 액션 태그 사용하기

8. 인클루드 액션 태그로 요청한 include1.jsp의 자바 파일을 열어볼까요?
 <jsp:param> 태그로 전달한 매개변수들이 자식 JSP로 전달됩니다.

```

120 request.setCharacterEncoding("utf-8");
121
122 out.write(" \r\n");
123 out.write("\r\n");
124 out.write("<!DOCTYPE html>\r\n");
125 out.write("<html>\r\n");
126 out.write("<head>\r\n");
127 out.write("<meta charset=\"UTF-8\">\r\n");
128 out.write("<title>include1.jsp</title>\r\n");
129 out.write("</head>\r\n");
130 out.write("<body>\r\n");
131 out.write(" 안녕하세요. 쇼핑몰 중심 JSP 시작입니다!!!\r\n");
132 out.write("<br>\r\n");
133 org.apache.jasper.runtime.JspRuntimeLibrary.include(request, response, "duke_image.jsp" + "?" +
134 org.apache.jasper.runtime.JspRuntimeLibrary.URLEncode("name", request.getCharacterEncoding()) + "=" +
135 org.apache.jasper.runtime.JspRuntimeLibrary.URLEncode("듀크", request.getCharacterEncoding()) + "&" +
136 org.apache.jasper.runtime.JspRuntimeLibrary.URLEncode("imgName", request.getCharacterEncoding()) + "=" +
137 org.apache.jasper.runtime.JspRuntimeLibrary.URLEncode("duke.png", request.getCharacterEncoding()), out, true);
138 out.write("\r\n");
139 out.write("<br>\r\n");
140 out.write(" 안녕하세요. 쇼핑몰 중심 JSP 끝 부분입니다!!!\r\n");
141 out.write("</body>\r\n");
142 out.write("</html>\r\n");
143 } catch (java.lang.Throwable t) {
144 if (!(t instanceof javax.servlet.jsp.SkipPageException)){
145 out = _jspx_out;
146 if (out != null && out.getBufferSize() != 0)
  
```

인클루드 액션 태그의 <param> 태그의 값이 request 객체에 바인딩되어서 포함되는 JSP로 전달됩니다.

13.2 포워드 액션 태그 사용하기

포워드 액션 태그

- RequestDispatcher 클래스를 대신해서 포워딩하는 방법을 제공
- 포워딩 시 값을 전달할 수 있음

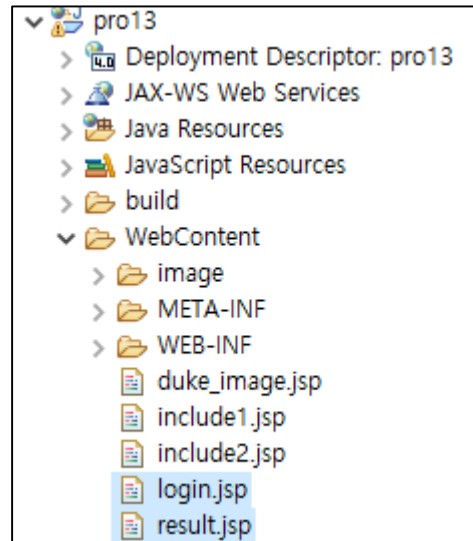
포워드 액션 태그 형식

```
<jsp:forward page="포워딩할 JSP 페이지" >  
...  
</jsp:forward>
```

13.2 포워드 액션 태그 사용하기

포워드 액션 태그 실습

1. 다음과 같이 실습 파일 login.jsp, result.jsp를 생성합니다.



13.2 포워드 액션 태그 사용하기

2. 로그인창에서 ID와 비밀번호를 입력한 후 action의 result.jsp로 전달하도록 login.jsp를 작성합니다.

코드 13-4 pro13/WebContent/login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");
%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>로그인창</title>
</head>
<body>
    <h1>아이디를 입력하지 않았습니다. 아이디를 입력해 주세요. </h1>
    <form action="result.jsp" method="post">
        아이디: <input type="text" name="userID"><br>
        비밀번호: <input type="password" name="userPw"><br>
        <input type="submit" value="로그인">
        <input type="reset" value="다시입력">
    </form>
</body>
</html>
```


13.2 포워드 액션 태그 사용하기

3. ID를 입력하지 않은 경우 자바의 `RequestDispatcher`를 사용하지 않고 포워드 액션 태그를 사용해 다시 로그인창으로 이동하도록 `result.jsp`를 작성합니다.

코드 13-5 pro13/WebContent/result.jsp

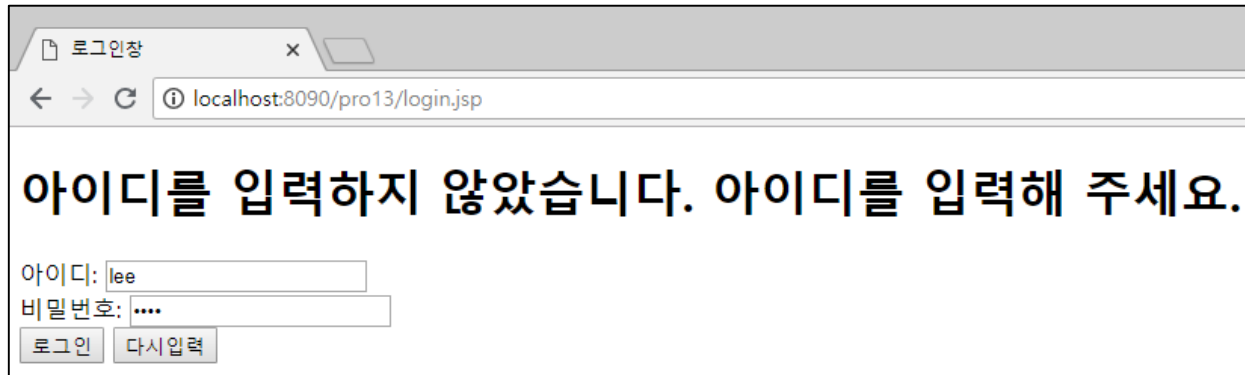
```
...
<body>
<%
    String userID = request.getParameter("userID" );
    if(userID.length()==0) {
        /*
        RequestDispatcher dispatch = request.getRequestDispatcher("login.jsp");
        dispatch.forward(request, response);
        */
    }
%>
<jsp:forward page="login.jsp" />
<%
    }
%>
    <h1> 환영합니다 <%= userID %>님!! </h1>
</body>
</html>
```

RequestDispatcher를 사용해 포워딩
하지 않아도 됩니다.

ID를 입력하지 않았으면 다시 <jsp:forward>
태그를 사용해 로그인창으로 포워딩합니다.

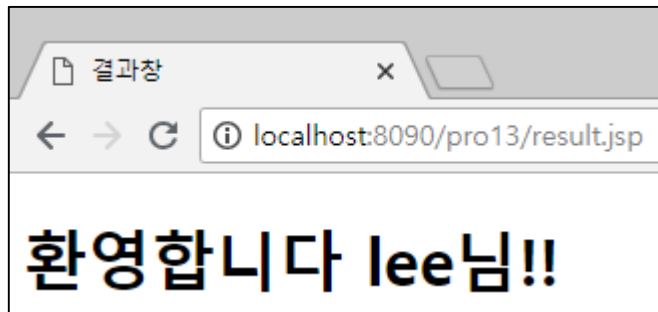
13.2 포워드 액션 태그 사용하기

4. <http://localhost:8090/pro13/login.jsp>로 요청하여 ID와 비밀번호를 입력하고 로그인합니다.



The screenshot shows a web browser window with the title '로그인창' (Login Window) and the address bar displaying 'localhost:8090/pro13/login.jsp'. The main content area displays a large error message in bold black text: '아이디를 입력하지 않았습니다. 아이디를 입력해 주세요.' (You did not enter an ID. Please enter an ID). Below the message, there are two input fields: '아이디: lee' (ID: lee) and '비밀번호:' (Password:). At the bottom, there are two buttons: '로그인' (Login) and '다시입력' (Re-enter).

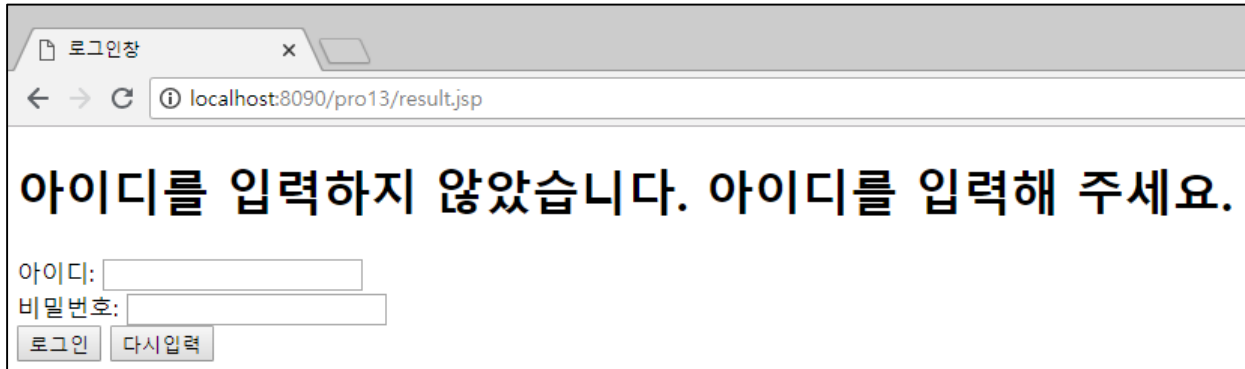
5. ID를 입력하면 정상적인 메시지를 출력합니다.



The screenshot shows a web browser window with the title '결과창' (Result Window) and the address bar displaying 'localhost:8090/pro13/result.jsp'. The main content area displays a large success message in bold black text: '환영합니다 lee님!!' (Welcome lee!!).

13.2 포워드 액션 태그 사용하기

6. 하지만 ID를 입력하지 않고 로그인을 시도하면 로그인창으로 포워딩하여 다음과 같은 메시지를 출력합니다.

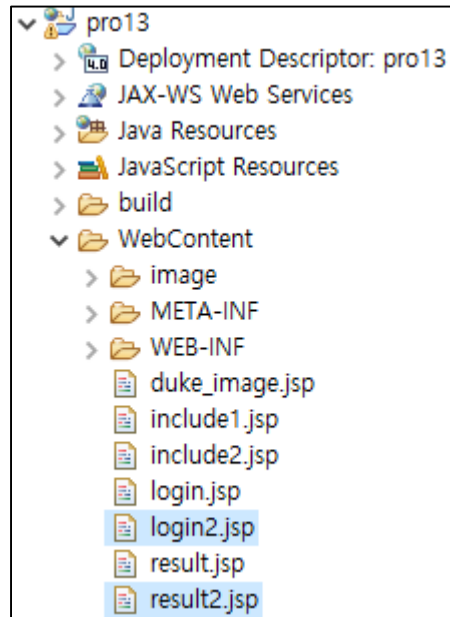


The screenshot shows a web browser window with the title '로그인창' (Login Window) and the address bar displaying 'localhost:8090/pro13/result.jsp'. The main content area displays a message in Korean: '아이디를 입력하지 않았습니다. 아이디를 입력해 주세요.' (You did not enter an ID. Please enter an ID). Below the message are two input fields: '아이디:' (ID) and '비밀번호:' (Password). At the bottom, there are two buttons: '로그인' (Login) and '다시입력' (Re-enter).

13.2 포워드 액션 태그 사용하기

- 최초 로그인창 접속 시 오류 메시지 출력하지 않기

7. login2.jsp와 result2.jsp 파일을 새로 만듭니다.



13.2 포워드 액션 태그 사용하기

8. login2.jsp를 다음과 같이 작성합니다.

코드 13-6 pro13/WebContent/login2.jsp

```
...  
<body>  
<%  
    String msg = request.getParameter("msg");  
    if(msg != null) {  
>%  
        <h1> <%=msg %> </h1>  
<%  
    }  
>%  
...  

```

브라우저에서 접속 시에는 msg 값을 가져와서 표시하고, 최초 접속 시에는 null이므로 아무것도 표시하지 않습니다.

13.2 포워드 액션 태그 사용하기

9. ID를 입력하지 않았을 경우 다시 로그인창으로 포워딩하면서 이번에는 `<jsp:param>` 태그를 이용해 msg 값을 전달합니다.

코드 13-7 pro13/WebContent/result2.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");
%>
```

login.jsp로 전달할 오류 메시지를 선언합니다.

```
<%!
    String msg = "아이디를 입력하지 않았습니다. 아이디를 입력해 주세요.";
%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>결과창</title>
</head>
<body>
<%
    String userID = request.getParameter("userID" );
    if(userID.length()==0) {
%>
```

로그인 시 입력한 이름을 가져옵니다.

이름을 입력하지 않았을 경우
`<jsp:param>` 액션 태그를 이용해 오
류 메시지를 login.jsp로 전달합니다.

```
<jsp:forward page="login2.jsp" >
```

13.2 포워드 액션 태그 사용하기

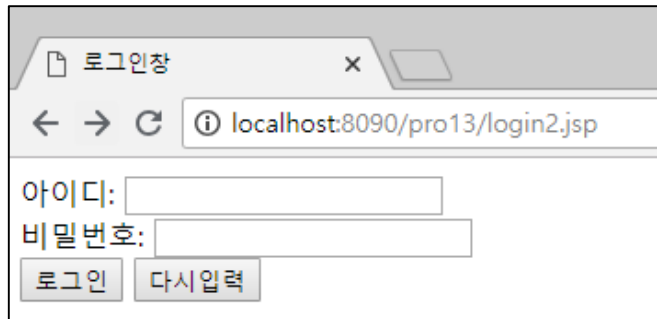
```
<jsp:param name="msg" value="<%= msg %>" />
</jsp:forward>

<%
}
%>

<h1>환영합니다. <%=userID %>님!!! </h1>
</body>
</html>
```

13.2 포워드 액션 태그 사용하기

10. `http://localhost:8090/pro13/login2.jsp`로 요청합니다. 최초 로그인창 접속 시 앞에서와는 달리 어떤 메시지도 나타나지 않습니다.



로그인창

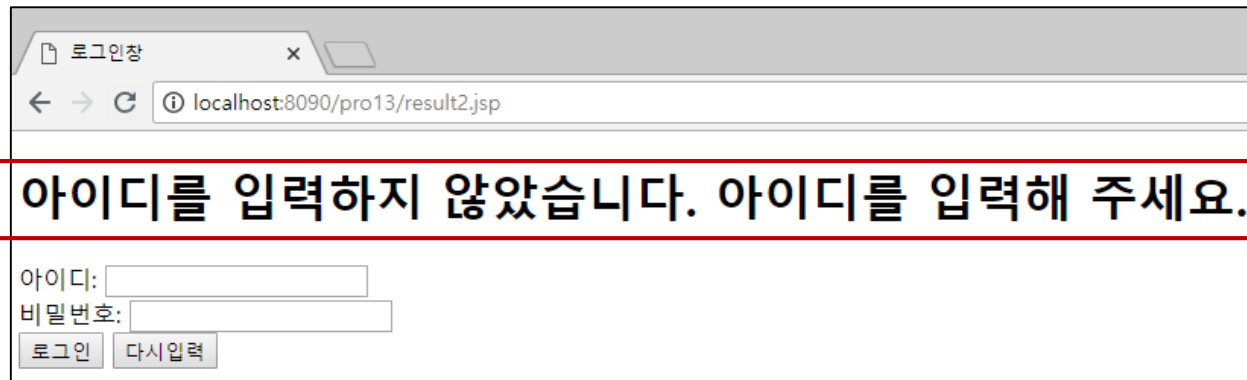
← → ↻ ⓘ localhost:8090/pro13/login2.jsp

아이디:

비밀번호:

로그인 다시입력

11. ID를 입력하지 않고 로그인하면 로그인창으로 다시 포워딩되면서 오류 메시지가 나타납니다.



로그인창

← → ↻ ⓘ localhost:8090/pro13/result2.jsp

아이디를 입력하지 않았습니다. 아이디를 입력해 주세요.

아이디:

비밀번호:

로그인 다시입력

13.3 useBean, setProperty, getProperty

• 13.3.1 자바 빈을 이용한 회원 정보 조회 실습

자바 빈(Bean)

- Java EE 프로그래밍 시 여러 객체를 거치면서 만들어지는 데이터를 저장하거나 전달하는 데 사용되는 클래스
- 자바의 DTO(Data Transfer Object, 데이터 전송 객체) 클래스, VO(Value Class, 값 객체) 클래스와 같은 개념

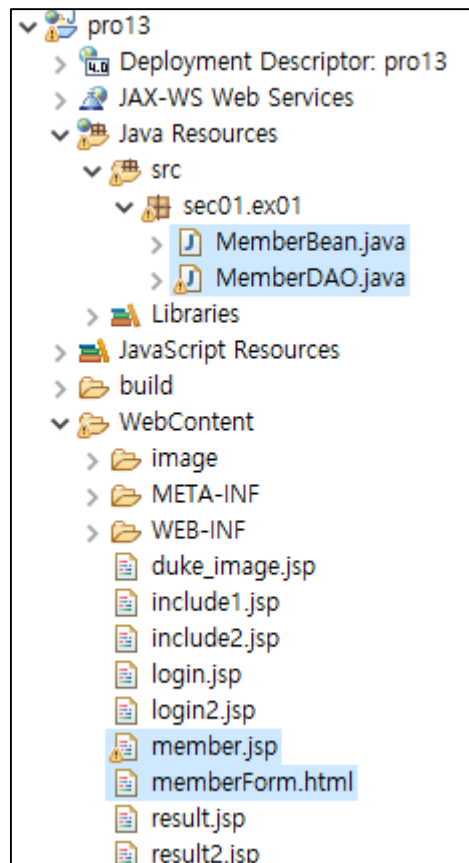
자바 빈(Bean) 특징

- 속성의 접근 제한자는 private임
- 각 속성(attribute, property)은 각각의 setter/getter를 가짐
- setter/getter 이름의 첫 글자는 반드시 소문자이어야 함
- 인자 없는 생성자를 반드시 가지며 다른 생성자도 추가할 수 있음

13.3 useBean, setProperty, getProperty

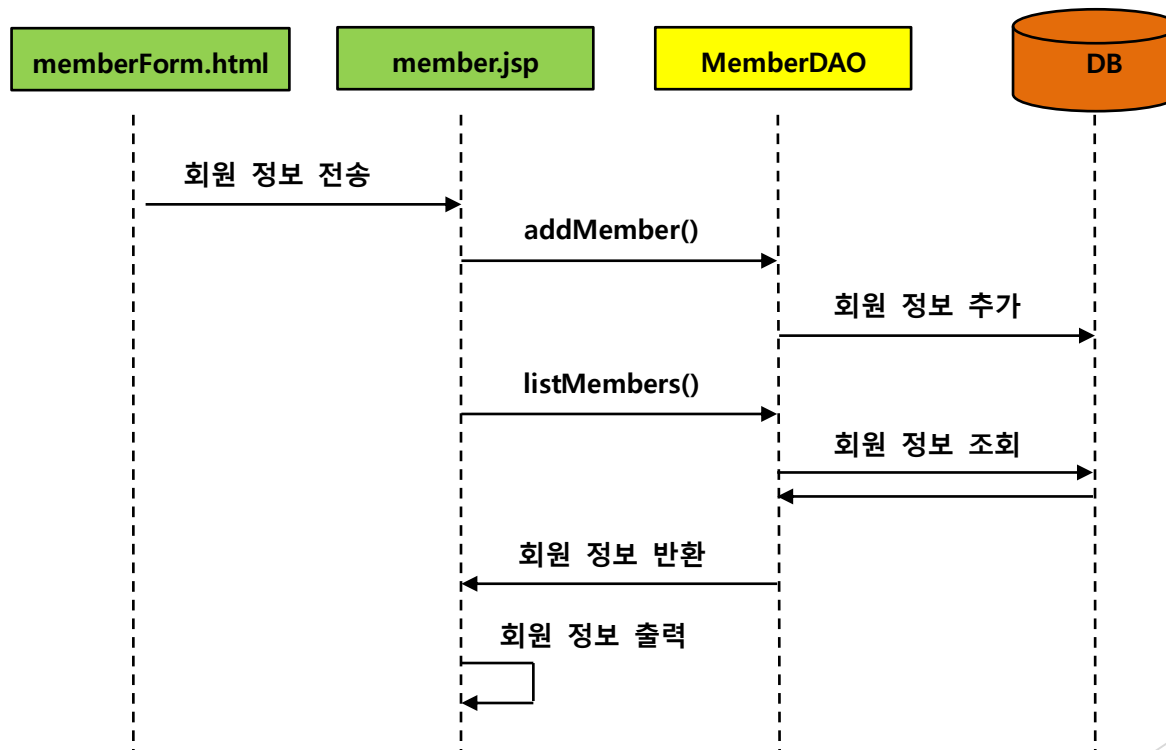
자바 빈(Been)을 이용한 회원 조회 실습

1. sec01.ex01 패키지를 생성하고 MemberBean, MemberDAO 클래스를 추가합니다. 그리고 실습 파일 member.jsp와 memberForm.html을 생성합니다.



13.3 useBean, setProperty, getProperty

- 회원 정보 등록 및 조회과정



13.3 useBean, setProperty, getProperty

2. 회원 정보를 저장하는 MemberBean 클래스를 만들기 전에 다음의 회원 테이블을 봅시다.

회원 정보 저장 테이블의 각 컬럼명들

```
1  --회원 테이블 생성
2  create table t_member(
3      id varchar2(10) primary key,
4      pwd varchar2(10),
5      name varchar2(50),
6      email varchar2(50),
7      joinDate date default sysdate
8  );
```

13.3 useBean, setProperty, getProperty

3. 회원 테이블을 참고해 MemberBean 클래스를 작성합니다.

코드 13-8 pro13/src/sec01/ex01/MemberBean.java

```
package sec01.ex01;
```

```
...
```

```
public class MemberBean {
```

```
    private String id;  
    private String pwd;  
    private String name;  
    private String email;  
    private Date joinDate;
```

회원 테이블의 컬럼 이름과 동일하게
이름과 자료형을 선언합니다.

```
    public MemberBean() {
```

```
    }
```

인자가 네 개인 생성자를 추가합니다.

```
    public MemberBean(String id, String pwd, String name, String email) {  
        this.id = id;  
        this.pwd = pwd;  
        this.name = name;  
        this.email = email;  
    }
```

```
    public String getId() {  
        return id;  
    }
```

getter/setter를 추가합니다.

```
    public void setId(String id) {  
        this.id = id;  
    }
```

```
    public String getPwd() {
```

13.3 useBean, setProperty, getProperty

4. 회원 가입창에서 회원 정보를 입력한 후 member.jsp로 전송하도록 memberForm.html을 작성합니다.

코드 13-9 pro13/WebContent/memberForm.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>회원 가입창</title>
<body>
  <form method="post" action="member.jsp">
    <h1 style="text-align:center">회원 가입창</h1>
    <table align="center">
      <tr>
        <td width="200">
          <p align="right">아이디
        </td>
        <td width="400"><input type="text" name="id"></td>
      </tr>
      <tr>
        <td width="200">
          <p align="right">비밀번호
        </td>
        <td width="400"><input type="password" name="pwd"></td>
      </tr>
      <tr>
        <td width="200">
```

13.3 useBean, setProperty, getProperty

5. member.jsp를 다음과 같이 작성합니다.

코드 13-10 pro13/WebContent/member.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    import="java.util.*, sec01.ex01.*" pageEncoding="UTF-8"%>

<%
    request.setCharacterEncoding("UTF-8");
%>

<%
    String id=request.getParameter("id");
    String pwd = request.getParameter("pwd");
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    MemberBean m = new MemberBean(id, pwd, name, email);
    MemberDAO memberDAO=new MemberDAO();
    memberDAO.addMember(m);
    List membersList = memberDAO.listMembers();
%>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>회원 목록창</title>
```

한글 인코딩을 설정합니다.

전송된 회원 정보를 가져옵니다.

MemberBean 객체를 생성한 후 회원 정보를 속성에 설정합니다.

회원 정보를 테이블에 추가합니다.

전체 회원 정보를 조회합니다.

13.3 useBean, setProperty, getProperty

```
</head>
<body>
  <table align="center" width="100%">
    <tr align="center" bgcolor="#99ccff">
      <td width="7%">아이디</td>
      <td width="7%">비밀번호</td>
      <td width="5%">이름</td>
      <td width="11%">이메일</td>
      <td width="5%">가입일</td>
    </tr>
  <%
    if(membersList.size()==0) {
  %>
    <tr>
      <td colspan="5">
        <p align="center"><b><span style="font-size:9pt;">
          등록된 회원이 없습니다.</span></b></p>
      </td>
    </tr>
  <%
  }else{
    for( int i = 0; i < membersList.size(); i++ ) {
      MemberBean bean = (MemberBean) membersList.get(i);
  %>
```

for 반복문을 이용해 membersList의 저장된 MemberBean 객체를 한 개씩 가져온 후 각 속성의 회원 정보를 다시 getter를 이용해 출력합니다.

13.3 useBean, setProperty, getProperty

```
<tr align="center">
  <td>
    <%= bean.getId() %>
  </td>
  <td>
    <%= bean.getPwd() %>
  </td>
  <td>
    <%= bean.getName() %>
  </td>
  <td>
    <%= bean.getEmail() %>
  </td>
  <td>
    <%= bean.getJoinDate() %>
  </td>
</tr>
<%
  } // end for
} // end if
%>
```

13.3 useBean, setProperty, getProperty

6. MemberDAO를 다음과 같이 작성합니다.

코드 13-11 pro13/src/sec01/ex01/MemberDAO.java

```
package sec01.ex01;

...

public class MemberDAO
{
    private Connection con;
    private PreparedStatement pstmt;
    private DataSource dataFactory;

    public MemberDAO()
    {
        try
        ...
```

13.3 useBean, setProperty, getProperty

```
public List listMembers()
{
    List list = new ArrayList();
    try
    {
        con = dataFactory.getConnection();
        String query = "select * from t_member order by joinDate desc ";
        System.out.println("prepareStatement: " + query);
        pstmt = con.prepareStatement(query);
        ResultSet rs = pstmt.executeQuery();
        while (rs.next())
        {
            String id = rs.getString("id");
            String pwd = rs.getString("pwd");
            String name = rs.getString("name");
            String email = rs.getString("email");
            Date joinDate = rs.getDate("joinDate");
            MemberBean vo = new MemberBean();
            vo.setId(id);
            vo.setPwd(pwd);
            vo.setName(name);
        }
    }
}
```

회원 정보를 최근 가입일 순으로 조회합니다.

조회한 회원 정보를 MemberBean 객체의 속성에 저장한 후 다시 ArrayList에 저장합니다.

13.3 useBean, setProperty, getProperty

```
        vo.setEmail(email);  
        vo.setJoinDate(joinDate);  
        list.add(vo);  
    }  
    rs.close();  
    pstmt.close();  
    con.close();  
} catch (Exception e)  
{  
    e.printStackTrace();  
}  
return list;  
}
```

13.3 useBean, setProperty, getProperty

```
public void addMember(MemberBean memberBean)
{
    try
    {
        Connection con = dataFactory.getConnection();
        String id = memberBean.getId();
        String pwd = memberBean.getPwd();
        String name = memberBean.getName();
        String email = memberBean.getEmail();
        String query = "insert into t_member";
        query += " (id,pwd,name,email)";
        query += " values(?,?,?,?)";
        System.out.println("prepareStatement: " + query);
        pstmt = con.prepareStatement(query);
        pstmt.setString(1, id);
        pstmt.setString(2, pwd);
        pstmt.setString(3, name);
        pstmt.setString(4, email);
        pstmt.executeUpdate();
        pstmt.close();
    }
}
```

MemberBean 객체에 저장된 회원 정보를 전달합니다.

getter를 이용해 회원 정보를 가져옵니다.

회원 정보를 추가합니다.

13.3 useBean, setProperty, getProperty

7. <http://localhost:8090/pro13/memberForm.html>로 요청하여 회원 정보를 입력한 후 가입하기를 클릭합니다.

회원 가입창

아이디

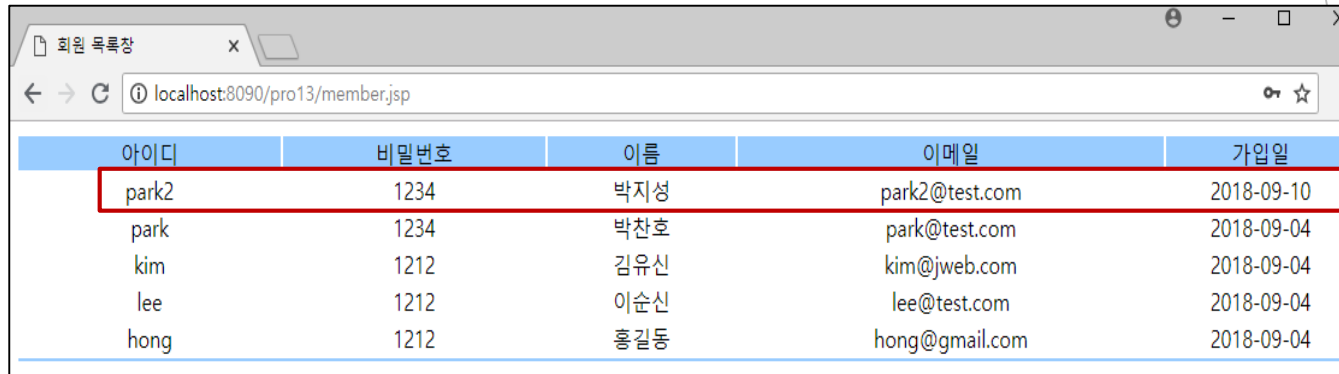
비밀번호

이름

이메일

13.3 useBean, setProperty, getProperty

8. 새 회원이 추가된 후 다시 회원 정보를 조회하여 목록으로 출력하는 것을 볼 수 있습니다.



아이디	비밀번호	이름	이메일	가입일
park2	1234	박지성	park2@test.com	2018-09-10
park	1234	박찬호	park@test.com	2018-09-04
kim	1212	김유신	kim@jweb.com	2018-09-04
lee	1212	이순신	lee@test.com	2018-09-04
hong	1212	홍길동	hong@gmail.com	2018-09-04

13.3 useBean, setProperty, getProperty

- 13.3.2 유즈빈 액션 태그를 이용한 회원 정보 조회 실습

유즈빈 액션 태그

➤ JSP 페이지에서 자바 빈을 대체하기 위한 태그

유즈빈 액션 태그 형식

```
<jsp:useBean id="빈 이름" class="패키지 이름을 포함한 자바 빈 클래스이름" [scope="접근범위"] />
```

id : JSP 페이지에서 자바 빈 객체에 접근할 때 사용할 이름

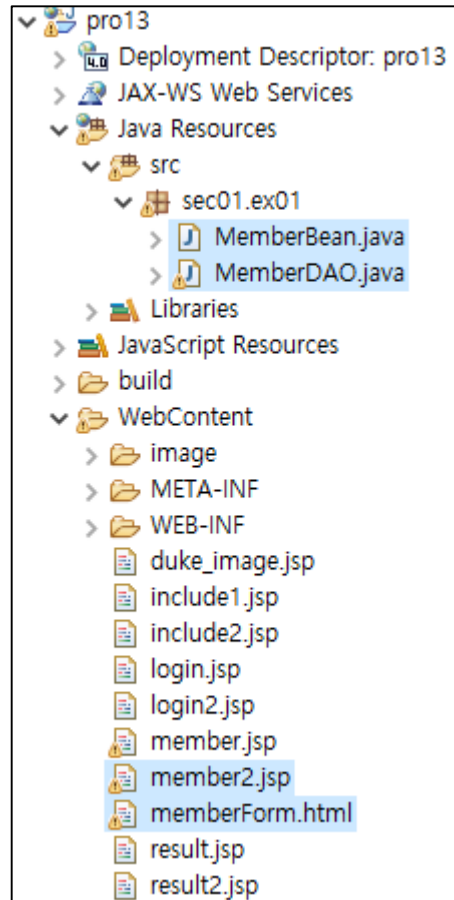
class : 패키지 이름을 포함한 자바 빈 이름

scope: 자바빈에 대한 접근 범위 (page, request, session, application)

13.3 useBean, setProperty, getProperty

- 유즈빈 액션 태그 실습

1. 다음과 같이 실습 파일 member2.jsp를 추가합니다.



13.3 useBean, setProperty, getProperty

2. member2.jsp를 다음과 같이 작성합니다.

코드 13-12 pro13/WebContent/member2.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    import="java.util.*, sec01.ex01.*" pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
%>
<jsp:useBean id="m" class="sec01.ex01.MemberBean" scope="page" />
<%
    String id=request.getParameter("id");
    String pwd = request.getParameter("pwd");
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    // MemberBean m = new MemberBean(id, pwd, name, email);
%>
```

useBean에 속성 값을 설정하기 전에 한글 인코딩 작업을 합니다.

유즈빈 액션 태그로 id가 m인 MemberBean 객체를 만듭니다.

자바 코드에서 MemberBean 객체를 생성하지 않습니다.

13.3 useBean, setProperty, getProperty

```
m.setId(id);  
m.setPwd(pwd);  
m.setName(name);  
m.setEmail(email);
```

useBean에 전송된 회원 정보를 설정합니다.

```
MemberDAO memberDAO=new MemberDAO();  
memberDAO.addMember(m);  
List membersList = memberDAO.listMembers();
```

회원 정보를 추가한 후
목록으로 출력합니다.

%>

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="UTF-8">  
  <title>회원 목록창</title>  
</head>  
<body>  
  <table align="center" width="100%">  
    <tr align="center" bgcolor="#99ccff">  
      <td width="7%">아이디</td>  
      <td width="7%">비밀번호</td>  
      <td width="5%">이름</td>  
      <td width="11%">이메일</td>  
      <td width="5%">가입일</td>  
    </tr>  
<%  
  if( membersList.size()==0) {  
%>  
  ...
```

13.3 useBean, setProperty, getProperty

3. 마찬가지로 `http://localhost:8090/pro13/memberForm.html`로 요청하여 회원 가입창에 회원 정보를 입력하고 가입하기를 클릭합니다.

회원 가입창

아이디

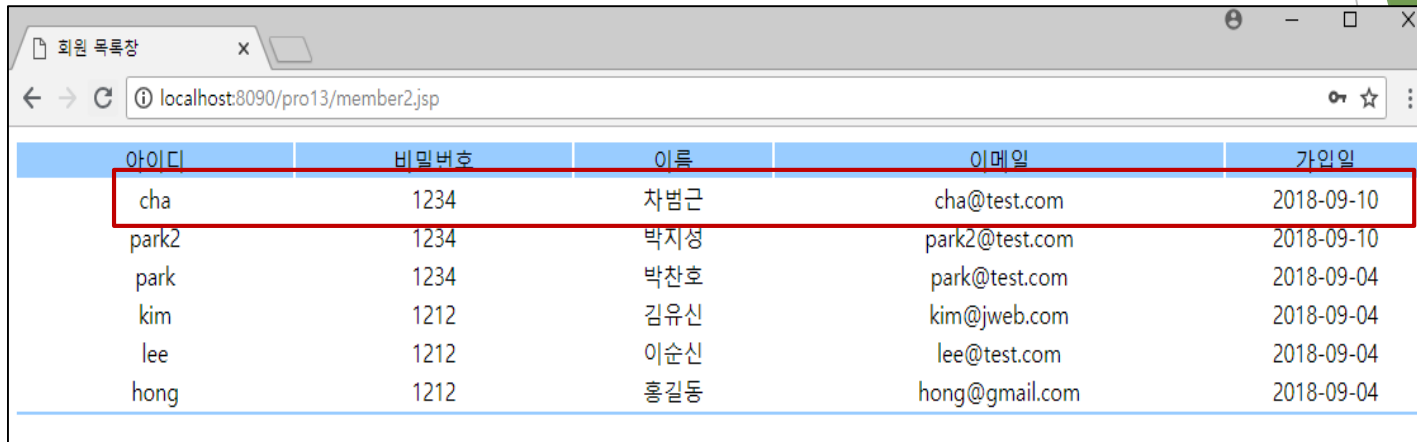
비밀번호

이름

이메일

13.3 useBean, setProperty, getProperty

4. 자바 빈을 사용했을 때와 마찬가지로 추가된 새 회원과 함께 회원 목록을 출력합니다.



A screenshot of a web browser window displaying a table of member information. The browser's address bar shows the URL `localhost:8090/pro13/member2.jsp`. The table has five columns: 아이디 (ID), 비밀번호 (Password), 이름 (Name), 이메일 (Email), and 가입일 (Join Date). The first row, containing the data for user 'cha', is highlighted with a red rectangular border. The other rows are not highlighted.

아이디	비밀번호	이름	이메일	가입일
cha	1234	차범근	cha@test.com	2018-09-10
park2	1234	박지성	park2@test.com	2018-09-10
park	1234	박찬호	park@test.com	2018-09-04
kim	1212	김유신	kim@jweb.com	2018-09-04
lee	1212	이순신	lee@test.com	2018-09-04
hong	1212	홍길동	hong@gmail.com	2018-09-04

13.3 useBean, setProperty, getProperty

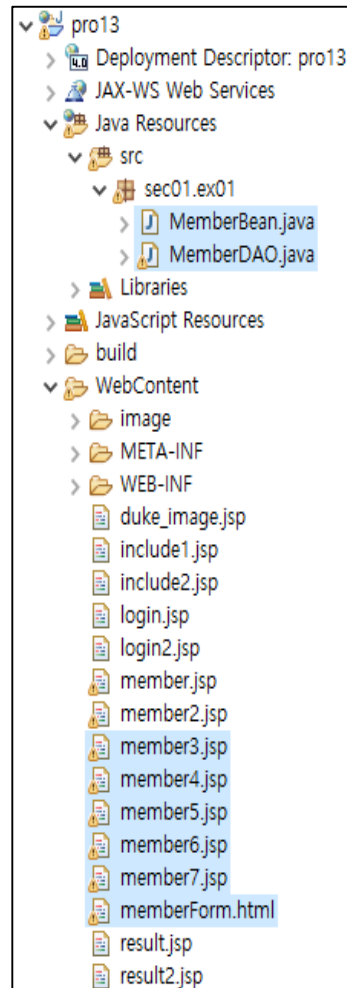
- 13.3.3 setProperty/getProperty 액션 태그를 이용한 회원 정보 조회 실습

setProperty와 getProperty 태그의 특징

이름	정의	형식
setProperty	useBean의 속성에 값을 설정하는 태그	<pre><jsp:setProperty name="자바 빈 이름" property="속성 이름" value="값" /></pre> <ul style="list-style-type: none">- name: <jsp:useBean> 액션 태그의 id 속성에 지정한 이름- property: 값을 설정할 속성 이름- value: 속성에 설정할 속성 값
getProperty	useBean의 속성 값을 얻는 태그	<pre><jsp:getProperty name="자바 빈 이름" property="속성 이름" /></pre> <ul style="list-style-type: none">- name: <jsp:useBean> 액션 태그의 id 속성에 지정한 이름- property: 값을 얻을 속성 이름

13.3 useBean, setProperty, getProperty

1. 다음과 같이 실습 파일 member3~7.jsp를 준비합니다.



13.3 useBean, setProperty, getProperty

2. member3.jsp에서 <jsp:useBean> 액션 태그로 생성된 빈에 대해 <jsp:setProperty> 액션 태그를 이용해 빈의 속성을 설정합니다.

코드 13-13 pro13/WebContent/member3.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    import="java.util.*, sec01.ex01.*" pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
%>

<jsp:useBean id="m" class="sec01.ex01.MemberBean" scope="page"/>
<jsp:setProperty name="m" property="id"
    value='<%= request.getParameter("id") %>' />
<jsp:setProperty name="m" property="pwd"
    value='<%= request.getParameter("pwd") %>' />
<jsp:setProperty name="m" property="name"
    value='<%= request.getParameter("name") %>' />
<jsp:setProperty name="m" property="email"
    value='<%= request.getParameter("email") %>' />
<%
```

useBean에 속성 값을 설정하기 전에 한글 인코딩 작업을 합니다.

유즈빈 액션 태그를 이용해 유즈빈을 만듭니다.

회원 가입창에서 전송된, 동일한 이름의 매개변수에 해당되는 유즈빈 속성에 전송된 값을 설정합니다.

13.3 useBean, setProperty, getProperty

```
/*
String  id=request.getParameter("id");
String  pwd = request.getParameter("pwd");
String  name = request.getParameter("name");
String  email = request.getParameter("email");

m.setId(id);
m.setPwd(pwd);
m.setName(name);
m.setEmail(email)
*/

MemberDAO memberDAO=new MemberDAO();
memberDAO.addMember(m);
List  membersList = memberDAO.listMembers();
%>

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>회원 목록창</title>
</head>
...
```

자바 코드를 이용해 빈에 설정할
필요가 없습니다.

13.3 useBean, setProperty, getProperty

3. 실행 결과는 자바 빈을 사용했을 때와 같습니다.
4. 이번에는 회원 가입창에서 전달된 회원 정보를 <jsp:setProperty> 액션 태그를 이용해 유즈 빈의 속성에 좀 더 편리하게 설정하는 방법을 알아보겠습니다.

```
9=<table align="center">
10=<tr>
11=<td width="200"><p align="right">아이디</td>
12=<td width="400"><input type="text" name="id"></td>
13=</tr>
14=<tr>
15=<td width="200"><p align="right">비밀번호</td>
16=<td width="400"><input type="password" name="pwd"></td>
17=</tr>
18=<tr>
19=<td width="200"><p align="right">이름</td>
20=<td width="400"><p><input type="text" name="name"></td>
21=</tr>
22=<tr>
23=<td width="200"><p align="right">이메일</td>
24=<td width="400"><p><input type="text" name="email"></td>
25=</tr>
26=<tr>
27=<td width="200"><p>&nbsp;</p></td>
28=<td width="400">
29=<input type="submit" value="가입하기">
30=<input type="reset" value="다시입력">
31=</td>
```

MemberBean의 속성명과
동일하게 설정합니다.

13.3 useBean, setProperty, getProperty

5. member4.jsp를 다음과 같이 작성합니다.

코드 13-14 pro13/WebContent/member4.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    import="java.util.*, sec03.ex02.*" pageEncoding="UTF-8"%>
<%
request.setCharacterEncoding("UTF-8");
%>
<jsp:useBean id="m" class="sec01.ex01.MemberBean" scope="page"/>
<jsp:setProperty name="m" property="id" param="id" />
<jsp:setProperty name="m" property="pwd" param="pwd" />
<jsp:setProperty name="m" property="name" param="name"/>
<jsp:setProperty name="m" property="email" param="email" />
<%
    MemberDAO memberDAO=new MemberDAO();
    memberDAO.addMember(m);
    List membersList = memberDAO.listMembers();
%>
...
```

회원 가입창에서 전달된 매개변수 이름과
속성 이름이 같으면 같은 이름으로 값을 설
정합니다

13.3 useBean, setProperty, getProperty

6. member5.jsp를 다음과 같이 작성합니다.

코드 13-15 pro13/WebContent/member5.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    import="java.util.*, sec01.ex01.*" pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
%>
<jsp:useBean id="m" class="sec01.ex01.MemberBean" scope="page"/>
<jsp:setProperty name="m" property="id" />
<jsp:setProperty name="m" property="pwd" />
<jsp:setProperty name="m" property="name" />
<jsp:setProperty name="m" property="email" />
<%
    MemberDAO memberDAO=new MemberDAO();
    memberDAO.addMember(m);
    List membersList = memberDAO.listMembers();
%>
...

```

회원 가입창에서 전달받은 매개변수 이름이 일치하는 useBean 속성에 자동으로 값을 설정해 줍니다.

13.3 useBean, setProperty, getProperty

7. member6.jsp를 다음과 같이 작성합니다.

코드 13-16 pro13/WebContent/member6.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    import="java.util.*,sec01.ex01.*"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
%>
```

```
<jsp:useBean id="m" class="sec01.ex01.MemberBean" scope="page"/>
```

```
<jsp:setProperty name="m" property="*" />
```

전송된 매개변수 이름과 빈 속성을 비교한
후 동일한 빈에 값을 자동으로 설정합니다.

```
<%
    MemberDAO memberDAO=new MemberDAO();
    memberDAO.addMember(m);
    List membersList = memberDAO.listMembers();
%>
...
```

13.3 useBean, setProperty, getProperty

8. 마지막으로 member7.jsp를 다음과 같이 작성합니다.

코드 13-17 pro13/WebContent/member7.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    import="java.util.*, sec01.ex01.*" pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
%>
<jsp:useBean id="m" class="sec01.ex01.MemberBean" scope="page"/>
<jsp:setProperty name="m" property="*" />
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>회원 목록창</title>
</head>
...
```

13.3 useBean, setProperty, getProperty

```
<table align="center" width="100%">
  <tr align="center" bgcolor="#99ccff">
    <td width="7%">아이디</td>
    <td width="7%">비밀번호</td>
    <td width="5%">이름</td>
    <td width="11%">이메일</td>
  </tr>
  <tr align="center">
    <td>
      <jsp:getProperty name="m" property="id" />
    </td>
    <td>
      <jsp:getProperty name="m" property="pwd" />
    </td>
    <td>
      <jsp:getProperty name="m" property="name" />
    </td>
    <td>
      <jsp:getProperty name="m" property="email" />
    </td>
  </tr>
  <tr height="1" bgcolor="#99ccff">
    ...
  </tr>
</table>
```

〈jsp:getProperty〉 태그를 이용해
useBean 속성 값에 접근합니다.

13.3 useBean, setProperty, getProperty

9. <http://localhost:8090/pro13/memberForm.html>로 요청하여 회원 가입창에 회원 정보를 입력한 후 가입하기를 클릭합니다.

회원 가입창

아이디

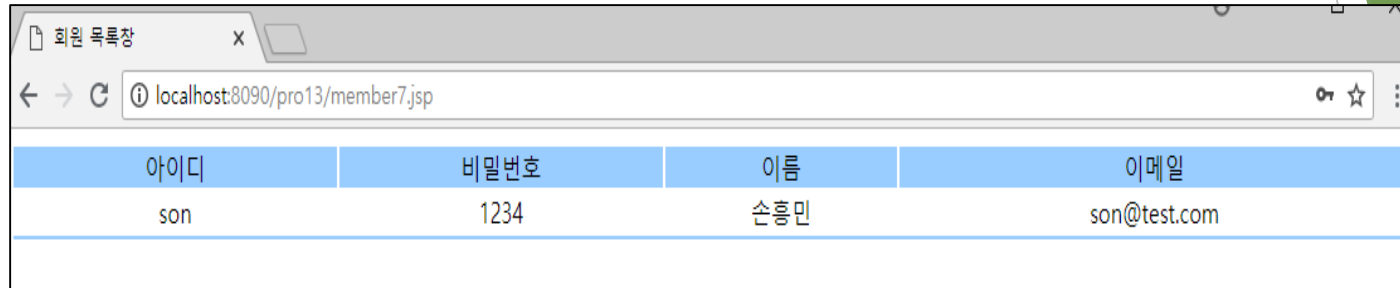
비밀번호

이름

이메일

13.3 useBean, setProperty, getProperty

10. 그러면 전송된 회원 정보를 `<jsp:getProperty>` 액션 태그를 이용해 출력합니다.



A screenshot of a web browser window. The address bar shows 'localhost:8090/pro13/member7.jsp'. The page displays a table with four columns: '아이디' (ID), '비밀번호' (Password), '이름' (Name), and '이메일' (Email). The table contains one row of data for a user named 'son' with password '1234', name '손흥민', and email 'son@test.com'.

아이디	비밀번호	이름	이메일
son	1234	손흥민	son@test.com