

- ④ <jsp:useBean> : 주어진 스코프에서 자바 객체 나 인스턴스를 id와 연결시킨다.
액션은 주어진 id로 기존의 **객체를찾다**가 발견하지 못하면
객체를 생성한다. Jsp:useBean액션은 하나의 JavaBean을 jsp와
연관시키는 데 쓰인다. 적어도 하나의 type과 class는 존재해야 하며,
class와 beanName이 동시에 유효하지는 않다.

형태1 : body가 없는 경우

```
<jsp:useBean id="변수이름" scope="page | request | session | application"  
class="빈클래스이름"/>
```

형태2 body가 있는 경우

```
<jsp:useBean id="name" scope="page | request | session | application" class="  
빈클래스이름"/>
```

body

*이부분에는 스크립트렛이나 <jsp:setProperty ..>태그를 사용하여
생성된 객체의 속성값 설정.*

```
</jsp:useBean>
```

id : 주어진 스코프에서 객체 인스턴스를 식별하기 위한 이름. 대소문자 구별
scope : 레퍼런스가 사용될 수 있는 범위(page, request, session, application)
class : 패키지 이름까지 모두 기술.

예1) <jsp:useBean id="member" class="shop.member.MemberBean" />

→ MemberBean member = new MemberBean ();

예2)<jsp:useBean

id="member"

class="shop.member.MemberBean"

scope="application">

<jsp:setProperty name=" member" property="name" value="김경호" />

<jsp:setProperty name=" member" property="address" value="경기성남" />

</jsp:useBean>

→ MemberBean member = new MemberBean ();

member.setName("김경호");

member.setAddress("경기성남");

④-①<jsp:setProperty>:jsp:setProperty액션은 빈에 속성 값을 설정할 때 사용한다.

④-②<jsp:getProperty>:jsp:getProperty는객체의 속성을 문자열 타입으로 변환해서
out변수를 이용해서 출력하는 작업을 한다.

memberBeanForm.jsp

```
<html>
<body>
<form method=POST action=memberBean.jsp>

이름 <input type=text name=name><br>
주소 <input type=text name=addr> <br><br>

<input type=submit value=전송>
<input type=reset value=취소>
</form>
</body>
</html>
```

MemberBean.java

```
package japana;

public class MemberBean{

    private String name = "";
    private String address = "";

    public void setName(String name) {
        this.name = toKor(name);
    }
    public String getName(){
        return name;
    }
    public void setAddr(String addr){
        this.address = toKor(addr);
    }
    public String getAddr() {
        return address;
    }
}
```

memberBean.jsp

```
<%@page contentType="text/html; charset=EUC_KR" import="japana.NameBean"%>

<jsp:useBean id="member" class="japana.NameBean" scope="page" />
<jsp:setProperty name="mybean" property="name" />
<jsp:setProperty name="mybean" property="addr" />
<HTML>
<HEAD><TITLE>setProperty Test</TITLE></HEAD>
<BODY>
<CENTER><H2>setProperty와 getProperty 테스트 </H2></CENTER>
<hr><br>
setProperty 테스트<br><br>
<%
    out.println("이름:" + member.getName());
    out.println("<BR>");
    out.println("주소:" + member.getAddr());
%>
<hr><br>
getProperty 테스트 <br><br>

이름: <jsp:getProperty name=" member " property="name" /><br>
주소: <jsp:getProperty name=" member " property="addr" />
</BODY>
</HTML>
```

- ⑤<jsp:param> : jsp:param액션은 이름-값 쌍의 형태를 통해서 다른 액션들에 추가적인 정보를 제공하는 데 쓰임.

* JSP 내장(디폴트) 객체

1. 정의

JSP에서 선언하지 않고 사용할 수 있는 객체
컨테이너에 의해 미리 선언된 자바클래스의 참조 변수
스크립트릿이나 표현식에서 사용할 수 있음

2. 주요 활용

- HTML폼에서 입력한 값을 가지고 올 때.
- 세션 관련 작업을 처리할 때.
- 스크립트릿에서 브라우저 출력이 필요할 때.
- 사용자 요청을 다른페이지로 전달할 때.
- 현재 JSP에 대한 각종 정보를 알고자 할 때.

2. 내장객체종류

표>내장객체종류 와 클래스타입

객체	클래스
page	javax.servlet.jsp.HttpJspPage
config	javax.servlet.jsp.ServletConfig
request	javax.servlet.http.HttpServletRequest
reponse	javax.servlet.http.HttpServletResponse
out	javax.servlet.jsp.JspWriter
session	javax.servlet.http.HttpSession
application	javax.servlet.ServletContext
pageContext	javax.servlet.jsp.PageContext
exception	java.lang.Throwable

① request 객체

클래스: javax.servlet.http.HttpServletRequest

주요 기능:

- 사용자 요청(request)과 관련된 정보를 처리할 때 사용.
- 주로 HTML 폼을 통해 전달된 값을 가져올 때 사용함.

② response 객체

클래스: javax.servlet.http.HttpServletResponse

주요 기능:

- 사용자 요청에 대한 응답 처리
- 페이지 전환

page_control.jsp

```
<%@ page contentType="text/html; charset=euc-kr" %>

<HTML>
<HEAD><TITLE> </TITLE></HEAD>
<BODY>
<H2>forward, sendRedirect 테스트</H2>
<HR>
<form method=post action=forward.jsp>
    forward action : <input type=text name=username>
    <input type=submit value="확인">
</form>

<form method=post action=sendRedirect.jsp>
    response.sendRedirect : <input type=text name=username>
    <input type=submit value="확인">
</form>
</BODY>
</HTML>
```

forward.jsp

```
<% request.setCharacterEncoding("euc-kr"); %>
<jsp:forward page="page_control_end.jsp">
    <jsp:param name="tel" value="000-000-0000" />
</jsp:forward>
```

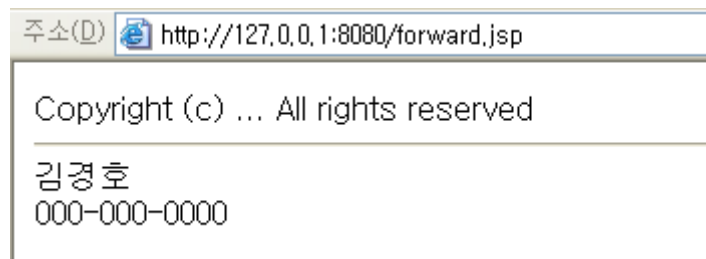
sendRedirect.jsp

```
<% response.sendRedirect("page_control_end.jsp"); %>
```

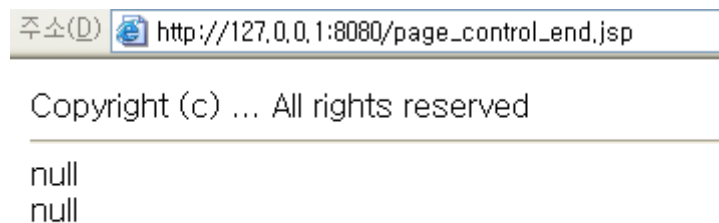
page_control_end.jsp

```
<%@ page contentType="text/html; charset=euc-kr" %>
<HTML>
<HEAD><TITLE> </TITLE></HEAD>
<BODY>
Copyright (c) ... All rights reserved
<HR>
<%= request.getParameter("username") %> <BR>
<%= request.getParameter("tel") %>
</BODY>
</HTML>
```

forward 결과



sendRedirect 결과



③ out 객체

클래스: javax.servlet.jsp.JspWriter

주요 기능:

- 스크립트릿에서 브라우저 출력, 버퍼 컨트롤

④session 객체

클래스 : javax.servlet.http.HttpSession

주요 기능:

- 클라이언트와의 지속적인 연결 유지를 위한 세션 처리
- 세션은 하나의 애플리케이션 내에서만 유효함.
- 두 개의 서로 다른 애플리케이션에서는 쿠키를 사용해야 함.
- 세션은 단순한 쿠키 대용이 아님.

session.jsp

```
<%@ page contentType="text/html; charset=euc-kr" %>
<HTML>
<HEAD><TITLE> </TITLE></HEAD>
<BODY>
<center>
<H2>session 예제 </H2>
<HR>
<%
    // isNew() 메서드를 이용해 최초세션설정을 확인하고 있다.
    if(session.isNew()) {
        out.println("<script> alert('세션이 해제되어 다시설정합니다.')
</script>");
        session.setAttribute("login","ok");
    }
%>
1. 세션 ID : <%= session.getId() %> <BR>
2. 세션 유지시간 : <%= session.getMaxInactiveInterval() %> <BR>
3. login 세션의 설정값 : <%= session.getAttribute("login") %> <BR>
</form>
</BODY>
```

⑤config 객체

클래스: javax.servlet.http.ServletConfig

주요기능:

- 애플리케이션에서 공유할수 있는 초기화 변수 접근
- web.xml 에 초기화 파라미터가 설정되어야 함.
- 애플리케이션의 ServletContext객체를 가져옴.

⑥application 객체

클래스: javax.servlet.http.ServletContext

주요기능:

- 웹 애플리케이션(컨텍스트) 에 관련된 정보를 제공
- 개발자를 위한 서버 정보
- 서버의 각종 자원 정보
- 로그 관련 정보
- application scope의 속성 관련 정보

application.jsp

```
<%@ page contentType="text/html;charset=euc-kr"
    import="java.io.*"
%>
<HTML>
<HEAD><TITLE> </TITLE></HEAD>
<BODY>
<center>
<H2>application 예제</H2>
<HR>
1. 서버 정보 : <%= application.getServerInfo() %> <BR>
2.   서블릿   API   버전정보   :   <%=   application.getMajorVersion()
+ ". "+ application.getMinorVersion() %> <BR>
3. application.jsp 파일의 실제경로 :<%= application.getRealPath("application.jsp") %>
<BR><HR>
setAttribute 로 username 변수에 "홍길동" 설정<P>
<%
    application.setAttribute("username","홍길동");
    application.log("username=홍길동");
%>
<a href="application_result.jsp">확인하기</a>
</center>
</BODY>
</HTML>
```


application_result.jsp

```
<%@ page contentType="text/html;charset=euc-kr"
import="java.io.*"
%>
<HTML>
<HEAD><TITLE> </TITLE></HEAD>
<BODY>
<center>
<H2>application 예제</H2>
<HR>
username 에 설정된 값은 : <%= application.getAttribute("username") %>
</center>
</BODY>
</HTML>
```

<<로그파일>>

자바코드: application.log("username= 신명숙 ");

위치: C:\Program Files\Apache Software Foundation\Tomcat 5.5\logs

파일이름: localhost.2007-02-09.txt

내용:

```
2007. 2. 9 오전 1:23:38 org.apache.catalina.core.ApplicationContext log
정보: username=신명숙
```

⑦page 객체 :

클래스: HttpJspBase

주요 기능:

- JSP 파일에서 자기자신을 참조할때 사용함.
- <%@page language=java...%>인 경우 크게 의미는 없음.

⑧pageContext 객체

클래스 : javax.servlet.jsp.PageContext

주요 기능:

- include, forward 액션의 실제구현에 응용됨.
- 내장객체에 대한 프로그램적인 접근방법 제공

⑨ exception 객체

클래스 : java.lang.Throwable

주요 기능:

- page 지시어에 isErrorPage="true" 로 지정된 에러 페이지에서 사용
- 현재 호출된 에러 정보 제공.