

JSP 3-2

JSP 3-1

JSP - 어플리케이션, 예외 처리, 자바 빈

어플리케이션 (Application)

웹 어플리케이션 전체에서 공유되는 정보를 저장하는 객체
서버가 종료되기 전까지 데이터가 유지됨

주요 메소드

메소드	설명
<code>setAttribute(String name, Object value)</code>	어플리케이션에 값 저장
<code>getAttribute(String name)</code>	저장된 값 불러오기
<code>removeAttribute(String name)</code>	저장된 값 삭제

```

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   <title>Insert title here</title>
8 </head>
9 <body>
10 <!--
11   어플리케이션 객체를 사용하여
12   웹 브라우저에 방문 시 방문자 수 출력하기
13   -->
14
15 <h2>웹 브라우저에 방문 시 방문자 수 출력하기</h2>
16
17 <%
18   int cnt = 0;
19   if(application.getAttribute("count") != null )
20   {
21     cnt = (int)application.getAttribute("count");
22   }
23   cnt++;
24   application.setAttribute("count", cnt);
25 %>
26
27 <%= cnt %>
28
29 </body>
30 </html>

```

초기 방문자 null.

=> cnt ++ 한 뒤
어플리케이션 추가.



' 세로고침 '



이제 count가 null이 아니므로
cnt에 값 넣고
cnt ++.

그리고 세로고침으로
어플리케이션에 추가한다.

← → localhost:8090/JSP_Basic/practice07/visitEx.jsp

웹 브라우저에 방문 시 방문자 수 출력하기

1

← → localhost:8090/JSP_Basic/practice07/visitEx.jsp

웹 브라우저에 방문 시 방문자 수 출력하기

6

! 예외 페이지

사용자 친화적인 에러 페이지를 제공하기 위해 직접 예외 페이지 작성 가능

- 대표 에러 코드:
 - 400: 잘못된 요청 (페이지 없음)
 - 500: 서버 내부 오류 (자바 실행 중 오류)
 - 200: 정상 처리

예외 처리 설정 (web.xml)

```

<error-page>
  <error-code>500</error-code>
  <location>/error/500.jsp</location>
</error-page>

```

- `<%@ page isErrorPage="true" %>` : 해당 페이지가 에러 처리용임을 명시
- `<% response.setStatus(200); %>` : 에러 페이지이지만 정상 상태로 응답


```

error.jsp  errorPage.jsp ×
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2    pageEncoding="UTF-8"%>
3
4
5  <%@ page isErrorPage="true" %>
6  <%-- 현재 페이지는 에러 페이지라는 것을 알림 --%>
7
8
9  <%
10     response.setStatus(200);
11     //상태는 정상이다
12
13     %>
14
15  <!DOCTYPE html>
16  <html>
17  <head>
18  <meta charset="UTF-8">
19  <title>Insert title here</title>
20  </head>
21  <body>
22  <h2>예외가 발생</h2>
23  <hr>
24
25  <%= exception.getMessage() %>
26  </body>
27  </html>

```

← → ↻ ⓘ localhost:8090/JSP_Basic/step09/error.jsp



예외가 발생

/ by zero

Web.xml로 설정하기.

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4
5
6
7
8 <!--
9   본래 페이지에서 제공이 예정된 경우, 대 사용자에게 보여줄
10  에러페이지를 처리할 jsp 페이지의 경로를 지정
11  -->
12
13 <!DOCTYPE html>
14 <html>
15 <head>
16 <meta charset="UTF-8">
17 <title>Insert title here</title>
18 </head>
19 <body>
20 <%
21   int i = 10 / 0;
22 //에러 발생
23 %>
24 </body>
25 </html>
```

→ 페이지마다 명시하지 않아도 된다.
(web.xml에서)

```
10 <welcome-file>default.html</welcome-file>
11 </welcome-file-list>
12
13 <error-page>
14   <error-code>404</error-code>
15   <location>/practice08/error404.jsp</location>
16 </error-page>
17
18 <error-page>
19   <error-code>500</error-code>
20   <location>/practice08/error500.jsp</location>
21 </error-page>
22
23
24 </web-app>
```

→ 해당 에러뜨면 ☐ 3 이동해준다.

```
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 <title>Insert title here</title>
8 </head>
9 <body>
10 <h2>404 에러페이지 보기</h2>
11 <a href="error1004.jsp">error1004.jsp</a>
12 </body>
13 </html>
```

localhost:8090/JSP_Basic/practice08/error1004.jsp

error404 페이지 입니다

자바 빈 (Java Bean)

JSP에서 데이터를 효율적으로 다루기 위한 클래스 객체
속성(필드) + 기능(메소드)으로 구성된 클래스

💡 빈 = 데이터 객체

1. 자바빈 클래스 생성 (속성과 기능 포함)
2. JSP에서 `<jsp:useBean>` 등 액션 태그로 활용

자바 빈 관련 액션태그

◆ `<jsp:useBean>`

```
<jsp:useBean id="Student" class="com.java.jsp.Student" scope="page" />
```

속성	설명
id	JSP에서 객체 접근 시 이름
class	자바빈 클래스 전체 경로
scope	객체 유효 범위 지정 (page , request , session , application)

범위	설명
page	해당 JSP 내에서만 사용
request	요청 처리 완료 전까지 유지
session	브라우저 종료 전까지 유지
application	서버 종료 전까지 유지 (모든 사용자 공유)

◆ <jsp:setProperty>

```
<jsp:setProperty name="Student" property="name" value="홍길동"/>
```

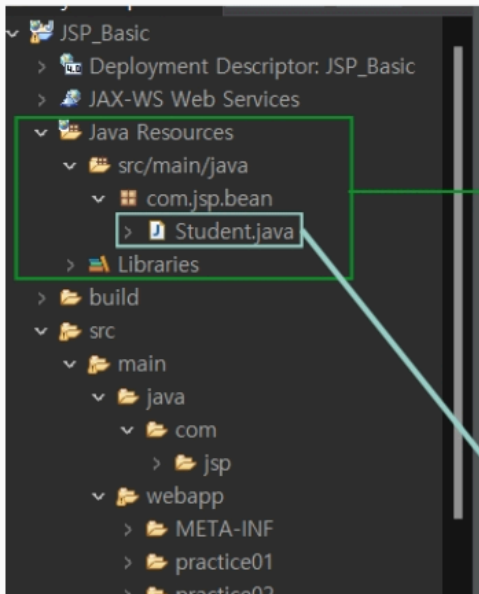
- name : 자바빈 이름
- property : 속성명
- value : 설정할 값

◆ <jsp:getProperty>

```
<jsp:getProperty name="Student" property="name"/>
```

자바 빈 규약

- 반드시 **패키지**에 포함되어 있어야 함 (default package X)
- 기본 **생성자** 존재 필요
- 멤버 변수는 **private**
- 접근을 위한 **getter** , **setter** 메소드 필요



여기다가 java 파일생성

```

1 package com.jsp.bean;
2
3 public class Student {
4
5
6 // 자바빈 규약
7 // 1. 빈이 패키지와 되어 있어야한다
8 // (default package가 아닌 지정된 패키지에 지정되어 있어야한다)
9 //
10 // 2. 기본 생성자를 반드시 가지고 있어야한다
11 // (getter / setter로 필드값을 설정할 목적으로 만들어졌으므로 기본 생성자를 가지고 있어야한다)
12 // 3. 빈의 변수의 접근자는 private으로 선언된다
13 // 4. 빈의 변수에 접근하기 위한 public 접근자인 getter / setter 메소드가 존재해야한다
14
15 private int studentID;
16 private String name;
17 private int age;
18 private String major;
19
20 public Student() {
21
22 }
23
24 public int getStudentID() {
25     return studentID;
26 }
27
28 public void setStudentID(int studentID) {
29     this.studentID = studentID;
30 }
31
32 public String getName() {

```

```

<jsp:useBean id="student" class="com.jsp.bean.Student" scope="page" />

<%--
id: jsp태그에서 자바빈 객체에 접근할 때 사용하는 이름
Student student = new Student(); 에서 student에 해당한다고 보면된다

class: 패키지 이름을 포함한 자바빈 클래스의 완전한 이름을 입력
scope: 자바빈 객체가 저장될 영역을 지정
(page, request, session, application)

name: 자바빈 객체의 이름을 명시
property: 속성 이름을 명시
value: 해당 속성값을 설정
--%>

<jsp:setProperty name="student" property="studentID" value="202401"/>
<jsp:setProperty name="student" property="name" value="홍길동"/>
<jsp:setProperty name="student" property="age" value="20"/>
<jsp:setProperty name="student" property="major" value="컴퓨터공학"/>

화면: <jsp:getProperty name="student" property="studentID"/><br>
이름: <jsp:getProperty name="student" property="name"/><br>
나이: <jsp:getProperty name="student" property="age"/><br>
전공: <jsp:getProperty name="student" property="major"/><br>
</body>
</html>

```

값 넣기.

값 불러오기



localhost:8090/JSP_Basic/step10/bean.jsp



학번: 202401

이름: 홍길동

나이: 20

전공: 컴퓨터공학