

7. 자바빈(JavaBean)

자바빈의 개요

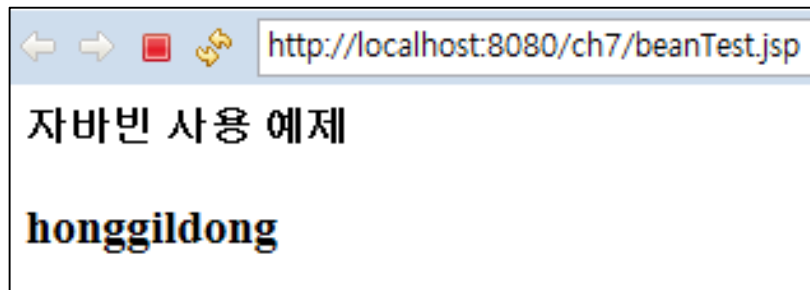
- ▶ JSP 페이지의 디자인 부분과 비즈니스 로직 부분을 분리하여 비효율적인 부분을 지원하기 위해 제공되는 기능의 형태
- ▶ 복잡한 JSP 코드들을 줄이고 프로그램 재사용성을 증가시킴
- ▶ 사용 이유
 - ▶ 디자인과 비즈니스 로직 부분을 분리하기 위함
- ▶ 장점
 - ▶ JSP 페이지가 복잡한 자바 코드로 구성되는 것을 피함
 - ▶ JSP 페이지에는 HTML 코드와 쉽고 간단한 자바 코드만을 구성
 - ▶ 한번 작성된 자바 빈즈를 여러 응용 프로그램에서 재사용이 가능
 - ▶ 프로그램의 개발 기간도 단축됨

자바빈의 개요

- ▶ 자바빈의 설계 규약
 - ▶ 멤버 변수마다 별도의 get/set 메소드가 존재해야 함
 - ▶ get 메소드는 파라미터가 존재하지 않아야 함
 - ▶ set 메소드는 반드시 하나 이상의 파라미터가 존재해야 함
 - ▶ 빈즈 컴포넌트의 속성은 반드시 읽기 또는 쓰기가 가능해야 함
 - ▶ get 메소드와 set 메소드를 구현해야 함
 - ▶ 읽기 전용일 경우 get 메소드만 정의 가능함
 - ▶ 생성자는 파라미터가 존재하지 않아야 함
 - ▶ 인자 없는 생성자는 반드시 있어야 함
 - ▶ 멤버 변수의 접근 제어자는 `private`이며, 각 set/get 메소드의 접근 제어자는 `public`으로 정의 되어야 함

자바빈의 개요

- ▶ 자바빈의 기본 작성 예제 (p.247~249)
 - ▶ beanTest.jsp 파일을 실행하면 BeanTest 자바빈 클래스를 beantest라는 이름의 객체를 생성함
 - ▶ beantest 객체에 있는 name 값을 출력함



BeanTest.java 파일

```
1 package ch7;
2
3 public class BeanTest {
4     private String name="honggildong";
5
6     public String getName(){
7         return name;
8     }
9
10    public void setName(String name){
11        this.name = name;
12    }
13 }
```

beanTest.jsp 파일

```
<jsp:useBean id="beantest" class="ch7.BeanTest" scope="page" />
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<b>자바빈 사용 예제</b>
<h3><%=beantest.getName() %></h3>
</body>
</html>
```

JSP에서 자바빈 사용하기

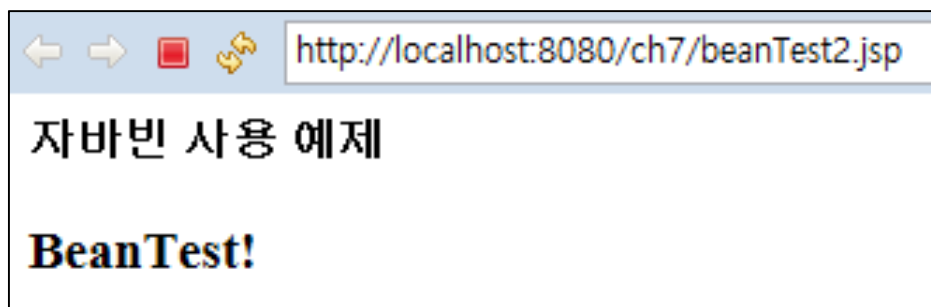
- ▶ <jsp:useBean /> 태그
 - ▶ 자바빈 객체를 생성하기 위한 액션 태그
 - ▶ 사용법

```
<jsp:useBean id="자바빈이름" class="자바빈 클래스명" scope="사용범위" />
```

- ▶ <jsp:setProperty /> 태그
 - ▶ 자바빈 클래스의 속성 값을 설정하기 위한 태그
 - ▶ 사용법

```
<jsp:setProperty name="자바빈 이름" property="속성명" value="설정할 속성 값" />
```

beanTest2.jsp 파일



```
<jsp:useBean id="beantest" class="ch7.BeanTest" scope="page" />
<jsp:setProperty property="name" name="beantest" value="BeanTest!" />
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<b>자바빈 사용 예제</b>
<h3><%=beantest.getName() %></h3>
</body>
</html>
```

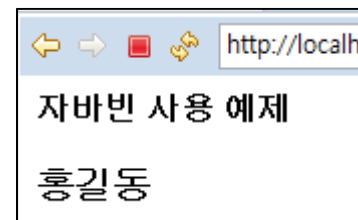
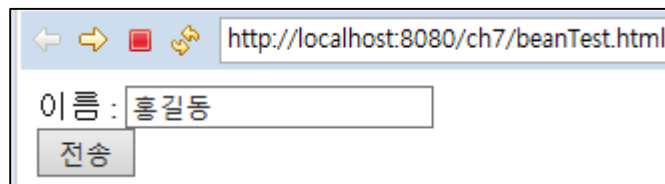
```
public void setName(String name){
    this.name = name;
}
```

JSP에서 자바빈 사용하기

- ▶ `<jsp:setProperty />` 태그
 - ▶ 클라이언트에서 전송되어 오는 파라미터 값을 속성 값으로 할당하는 경우
 - ▶ value 대신 param 사용

```
<jsp:setProperty name="자바빈 이름" property="속성명" param="파라미터명" />
```

- ▶ 파라미터의 이름이 설정하려는 객체의 이름과 같으면 생략 가능함
- ▶ 예(beanTest.html, beanTest3.jsp)
 - ▶ 이름을 입력 받아 beanTest3.jsp 페이지로 name 속성 값을 전송하고 전송되어 온 name 속성 값을 출력함



beanTest.html 파일

```
<body>
<form action="beanTest3.jsp">
    이름 : <input type="text" name="name"> <br>
    <input type="submit" value="전송">
</form>
</body>
```

beanTest3.jsp 파일

```
4 <jsp:useBean id="beantest" class="ch7.BeanTest" scope="page" />
5 <jsp:setProperty property="name" name="beantest" param="name"/>
6 <% request.setCharacterEncoding("utf-8"); %>
7<html>
8<head>
9 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
0 <title>Insert title here</title>
1 </head>
2<body>
3 <b>자바빈 사용 예제</b>
4 <h3><%=beantest.getName() %></h3>
5 </body>
```

JSP에서 자바빈 사용하기

- ▶ `<jsp:setProperty />` 태그
 - ▶ 클라이언트에서 전송되어 오는 파라미터 이름이 빈객체의 속성명과 모두 같다면 한번에 속성명을 할당할 수 있음

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

- ▶ 예(beanTest2.java, beanTest2.html, beanTest4.jsp)
 - ▶ 이름을 입력 받아 beanTest4.jsp 페이지로 속성 값들을 전송하고 자바빈을 사용하여 속성 값들을 출력함

beanTest2.html

이름 :

주소 :

이메일 주소 :

생년월일 : 연도 -월 -일

2017년 10월 ▼

일	월	화	수	목	금	토
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

beanTest4.jsp

자바빈 속성 값 출력

이름 : 홍길동
주소 : 부산시
이메일 주소 : a@a.n
생년월일 : 2017-10-22

beanTest2.java

```
package ch7;

public class BeanTest2 {
    private String name;
    private String addr;
    private String email;
    private String birthday;
    public String getName(){
        return name;
    }
    public void setName(String name){
        this.name = name;
    }
    public String getAddr(){
        return addr;
    }
    public void setAddr(String addr){
        this.addr = addr;
    }
    public String getEmail(){
        return email;
    }
    public void setEmail(String email){
        this.email = email;
    }
    public String getBirthday(){
        return birthday;
    }
    public void setBirthday(String birthday){
        this.birthday = birthday;
    }
}
```

beanTest4.jsp

```
<% request.setCharacterEncoding("utf-8"); %>
<jsp:useBean id="beantest" class="ch7.BeanTest2" scope="page" />
<jsp:setProperty property="*" name="beantest"/>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h3> 자바빈 속성 값 출력</h3>
이름 : <%=beantest.getName() %><br>
주소 : <%=beantest.getAddr() %><br>
이메일 주소 : <%=beantest.getEmail() %><br>
생년월일 : <%=beantest.getBirthday() %><br>
</body>
</html>
```

JSP에서 자바빈 사용하기

- ▶ `<jsp:getProperty />` 태그
 - ▶ 자바빈 클래스의 속성 값을 가져오기 위한 태그
 - ▶ 사용법

```
<jsp:getProperty name="자바빈 이름" property="속성명" />
```

- ▶ 예(beanTest5.jsp)
 - ▶ beanTest3.jsp를 복사하여 beanTest5.jsp 생성

```
14 <h3><%=beantest.getName() %></h3>  
15 <h3><jsp:getProperty name="beantest" property="name" /></h3>  
16 </body>
```

자바빈 사용 예제

honggildong

honggildong

자바빈을 이용한 회원 폼 작성

- ▶ JoinBean.java, joinForm.jsp, joinChk.jsp (p.262~267)

회원정보 입력

아이디 :

비밀번호 :

이름 :

성별 : ☒ 남 ☐ 여

나이 :

이메일주소 :

아이디 : hong123
비밀번호 : 1234
이름 : 홍길동
성별 : 1
나이 : 21
이메일주소 : hong@a.n

joinChk.jsp

```
<% request.setCharacterEncoding("utf-8"); %>
<jsp:useBean id="join" class="ch7.JoinBean" />
<jsp:setProperty property="*" name="join"/>

~

<table>
  <tr>
    <td><b>아이디 : </b></td>
    <td><jsp:getProperty name="join" property="id"/></td>
  </tr>
  <tr>
    <td><b>비밀번호 : </b> </td>
    <td><jsp:getProperty name="join" property="pass"/></td>
  </tr>
  <tr>
    <td><b>이름 : </b></td>
    <td><jsp:getProperty name="join" property="name"/></td>
  </tr>
  <tr>
    <td><b>성별 : </b></td>
    <td><jsp:getProperty name="join" property="sex"/></td>
  </tr>
  <tr>
    <td><b>나이 : </b></td>
    <td><jsp:getProperty name="join" property="age"/></td>
  </tr>
  <tr>
    <td><b>이메일주소 : </b></td>
    <td><jsp:getProperty name="join" property="email"/></td>
  </tr>
</table>
```


8. 세션과 쿠키

세션(session)

▶ 세션 객체 메소드

메소드	설명
setAttribute(String attrname, Object attrValue)	세션 영역에 속성을 생성함
removeAttribute(String attrName)	파라미터로 지정된 이름의 속성을 제거함
getAttribute(String attrName)	지정된 이름의 속성 값을 반환함
getId()	클라이언트의 세션ID 값을 반환함
setMaxInactiveInterval(int seconds)	세션의 유지 시간을 설정함
getMaxInactiveInterval()	세션의 유지시간을 반환함
invalidate()	현재의 세션 정보들을 모두 제거함

세션(session)

▶ 예제 (p.275~276)



sessionTest.jsp

```
<%  
    String name;  
    if (session.getAttribute("name") != null) {  
        name=(String) session.getAttribute("name");  
    }else{  
        name="세션 값 없음.";  
    }  
%>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>Insert title here</title>  
</head>  
<body>  
<h2>세션 테스트</h2>  
<input type="button" onclick="location.href='sessionSet.jsp'" value="세션 값 저장">  
<input type="button" onclick="location.href='sessionDel.jsp'" value="세션 값 삭제">  
<input type="button" onclick="location.href='sessionInvalidate.jsp'" value="세션 초기화">  
<h3><%=name %></h3>  
</body>  
</html>
```

sessionSet.jsp

```
<body>
<%session.setAttribute("name","Session Test!"); %>
<script>
location.href="sessionTest.jsp";
</script>
</body>
```

sessionDel.jsp

```
<body>
<%session.removeAttribute("name"); %>
<script>
location.href="sessionTest.jsp";
</script>
</body>
```

sessionInvalidate.jsp

```
<body>
<%session.invalidate(); %>
<script>
location.href="sessionTest.jsp";
</script>
</body>
```

쿠키(Cookie)

- ▶ 클라이언트 측에서 관리되는 정보
- ▶ 브라우저 종료 후에도 지정된 생존 기간 동안 유지됨
- ▶ 보안적으로 취약하므로 보안이 필요한 정보는 세션을 사용함
- ▶ 로그인 시 아이디 저장 => 쿠키 사용

쿠키(Cookie)

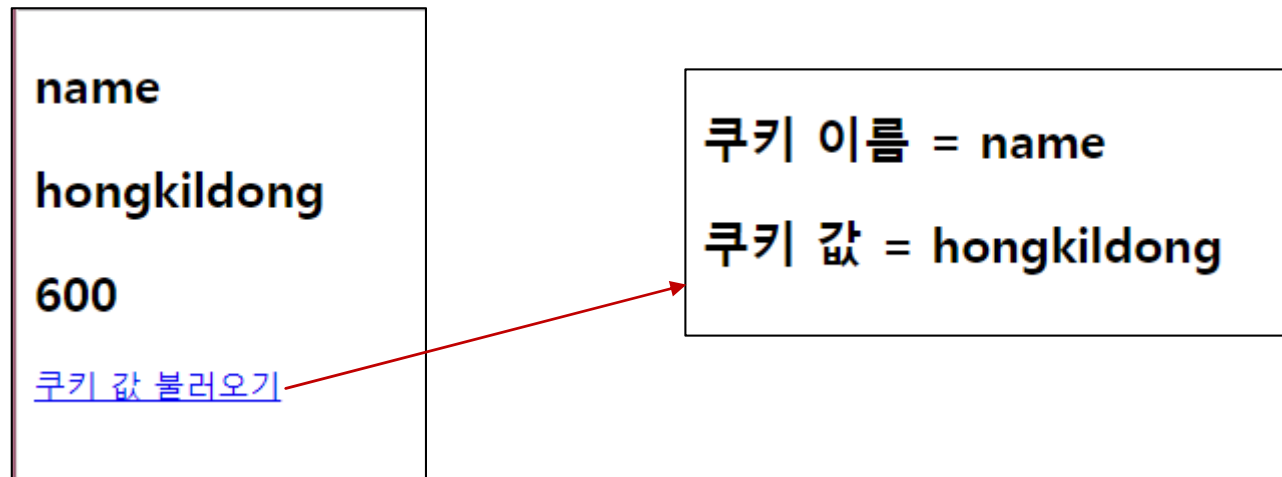
- ▶ 서블릿 API를 이용한 쿠키 설정
 - ▶ 객체 생성 및 클라이언트 전송

```
Cookie cookie = new Cookie(name, value);  
response.addCookie(cookie);
```

속성	설명
setValue(String value)	쿠키 값을 설정함
setMaxAge(int seconds)	쿠키 만료 기간을 지정함
getValue()	쿠키 값을 얻어옴
getMaxAge()	쿠키 만료 기간을 얻어옴
getName()	쿠키 이름을 얻어옴

쿠키(Cookie)

▶ 쿠키 사용 예(p.282~283)



cookieTest.jsp

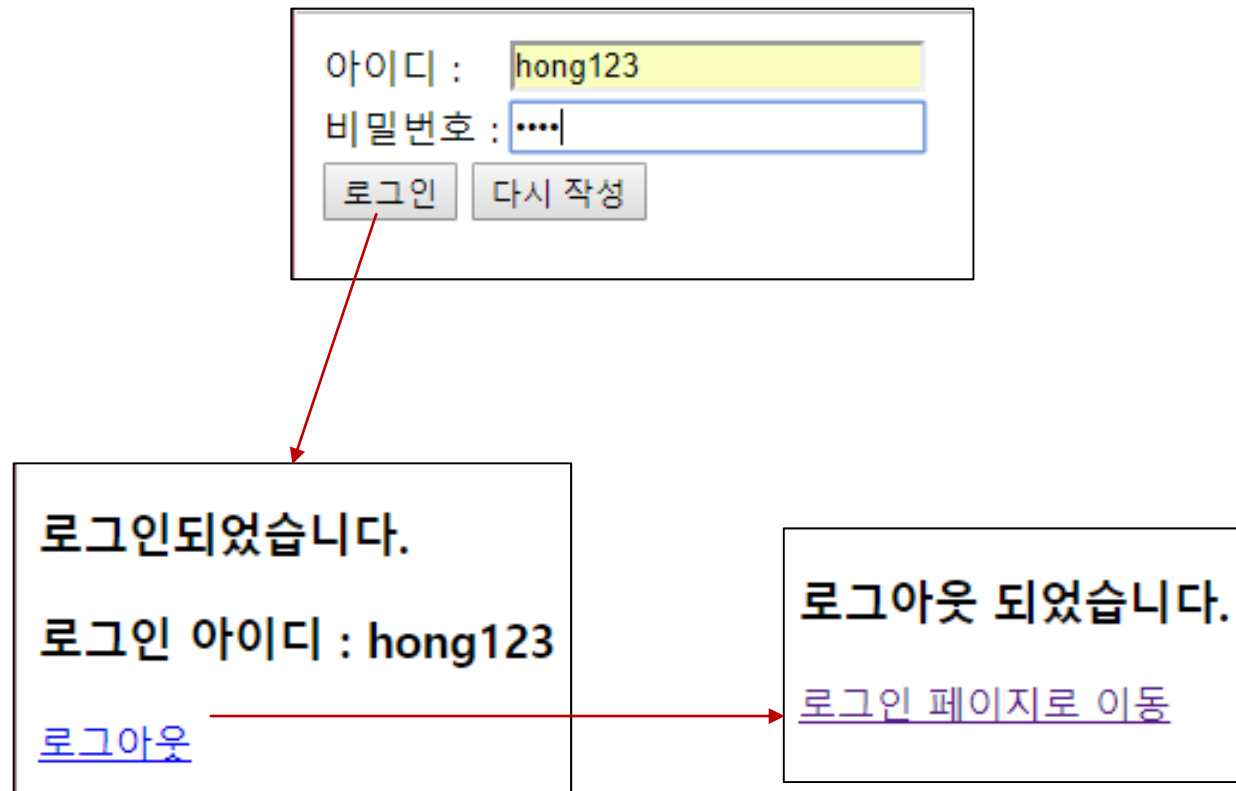
```
<%  
    Cookie cookie=new Cookie("name", "hongkildong");  
    cookie.setMaxAge(600);  
    response.addCookie(cookie);  
%>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>Insert title here</title>  
</head>  
<body>  
<h2><%=cookie.getName() %></h2>  
<h2><%=cookie.getValue() %></h2>  
<h2><%=cookie.getMaxAge() %></h2>  
<a href="cookieTest2.jsp">쿠키 값 불러오기</a>  
</body>  
</html>
```

cookieTest2.jsp

```
<%  
    String name="";  
    String value="";  
    String cookie = request.getHeader("Cookie");  
  
    if(cookie!=null){  
        Cookie cookies[]=request.getCookies();  
  
        for(int i=0;i<cookies.length;i++){  
            if(cookies[i].getName().equals("name")){  
                name=cookies[i].getName();  
                value=cookies[i].getValue();  
            }  
        }  
    }  
%>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>Insert title here</title>  
</head>  
<body>  
<h2>쿠키 이름 = <%=name %></h2>  
<h2>쿠키 값 = <%=value %></h2>  
</body>
```

세션과 쿠키를 이용한 예제

- ▶ 세션을 이용한 로그인 정보 유지(p.285~287)



sessionLogin2.jsp

```
<%session.setAttribute("id",request.getParameter("id")); %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=
<title>Insert title here</title>
</head>
<body>
<h3>로그인되었습니다.</h3>
<h3>로그인 아이디 : <%= (String) session.getAttribute("id") %></h3>
<a href="sessionLogout.jsp">로그아웃</a>
</body>
```

sessionLogout.jsp

```
<body>
<%session.removeAttribute("id"); %>
<h3>로그아웃 되었습니다.</h3>
<a href="sessionLogin1.jsp">로그인 페이지로 이동</a>
</body>
```

세션과 쿠키를 이용한 예제

- ▶ 쿠키를 이용한 사용자 화면 설정 정보 유지 (p.290~292)

안녕하세요. 이것은 쿠키 예제입니다.

☒ 한국어 페이지 보기 ☐ 영어 페이지 보기

안녕하세요. 이것은 쿠키 예제입니다.

☐ 한국어 페이지 보기 ☒ 영어 페이지 보기

Hello. This is Cookie example.

☐ 한국어 페이지 보기 ☒ 영어 페이지 보기

cookieExample1.jsp

```
<%  
    String language="korea";  
    String cookie = request.getHeader("Cookie");  
  
    if(cookie!=null){  
        Cookie cookies[]=request.getCookies();  
  
        for(int i=0;i<cookies.length;i++){  
            if(cookies[i].getName().equals("language")){  
                language=cookies[i].getValue();  
            }  
        }  
    }  
%>  
  
~  
  
<body>  
    <%if(language.equals("korea")){%>  
    <h3>안녕하세요. 이것은 쿠키 예제입니다.</h3>  
    <%}else{%>  
    <h3>Hello. This is Cookie example.</h3>  
    <%}%>  
  
    <form action="cookieExample2.jsp" method="post">  
        <input type="radio" name="language" value="korea"  
        <%if(language.equals("korea")){%>checked<%}%>>한국어 페이지 보기  
        <input type="radio" name="language" value="english"  
        <%if(language.equals("english")){%>checked<%}%>>영어 페이지 보기  
        <input type="submit" value="설정">  
    </form>  
</body>
```

cookieExample2.jsp

```
<%  
    Cookie cookie=new Cookie("language",request.getParameter("language"));  
    cookie.setMaxAge(60*60*24);  
    response.addCookie(cookie);  
%>  
<script>  
location.href="cookieExample1.jsp"  
</script>  
</body>
```

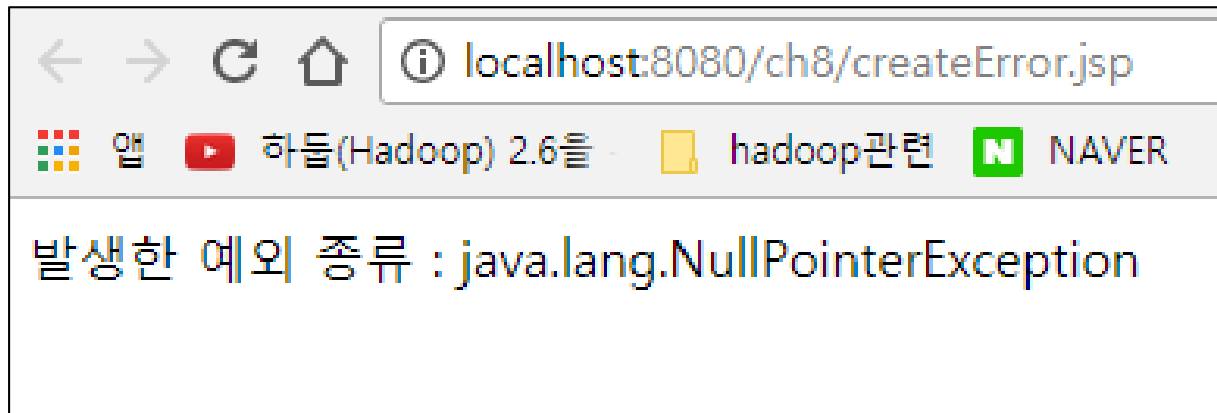
9. 예외처리

page 지시자의 errorPage 속성 사용

▶ 사용법

```
<%@ page errorPage = "errorProcessing.jsp" %>
```

▶ 사용 예(p.297~298)



createError.jsp

```
<%@ page errorPage="errorProcessing.jsp" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    String name = request.getParameter("name");
    if(name == null){
        throw new NullPointerException();
    }
%>
</body>
```

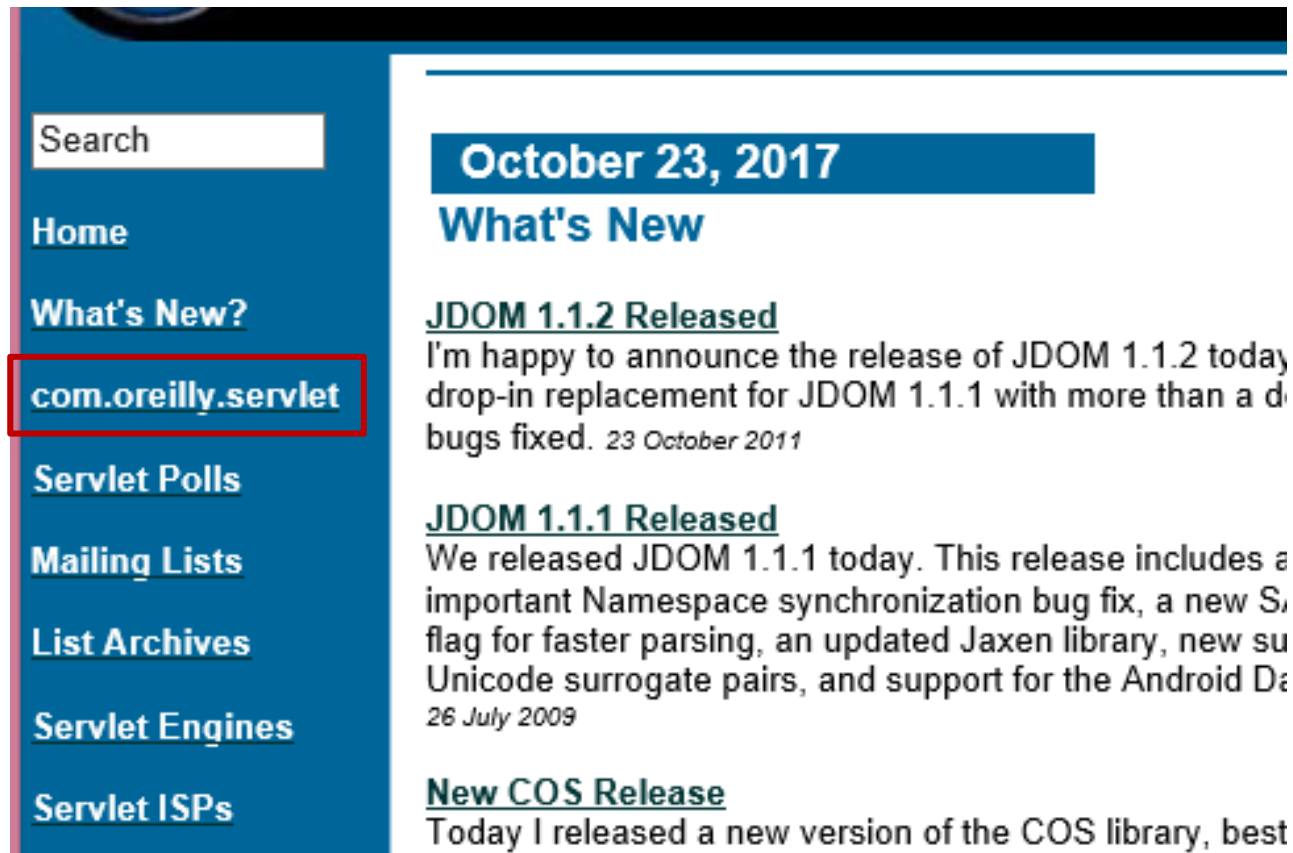
errorProcessing.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page isErrorPage="true" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://w
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    발생한 예외 종류 : <%=exception.getClass().getName() %>
    <!--
    사용 브라우저가 IE 일 경우는 예러 페이지 크기가 513바이트 이상 되어야 인식된다. 513바이트 이상을 주석으로 만들어 주자
    사용 브라우저가 IE 일 경우는 예러 페이지 크기가 513바이트 이상 되어야 인식된다. 513바이트 이상을 주석으로 만들어 주자
    사용 브라우저가 IE 일 경우는 예러 페이지 크기가 513바이트 이상 되어야 인식된다. 513바이트 이상을 주석으로 만들어 주자
    사용 브라우저가 IE 일 경우는 예러 페이지 크기가 513바이트 이상 되어야 인식된다. 513바이트 이상을 주석으로 만들어 주자
    -->
</body>
</html>
```

10. 파일 업로드

업로드 모듈 COS 라이브러리

- ▶ <http://www.servlets.com> 에서 다운로드



The screenshot shows the homepage of www.servlets.com. On the left is a blue sidebar with a search bar and several links. The link [com.oreilly.servlet](#) is highlighted with a red rectangular box. The main content area on the right has a blue header bar with the date 'October 23, 2017' and the title 'What's New'. Below this, there are three news items, each with a title and a brief description.

Search

[Home](#)

[What's New?](#)

[com.oreilly.servlet](#)

[Servlet Polls](#)

[Mailing Lists](#)

[List Archives](#)

[Servlet Engines](#)

[Servlet ISPs](#)

October 23, 2017

What's New

JDOM 1.1.2 Released
I'm happy to announce the release of JDOM 1.1.2 today drop-in replacement for JDOM 1.1.1 with more than a dozen bugs fixed. *23 October 2011*

JDOM 1.1.1 Released
We released JDOM 1.1.1 today. This release includes a important Namespace synchronization bug fix, a new S flag for faster parsing, an updated Jaxen library, new surrogate Unicode surrogate pairs, and support for the Android DOM. *26 July 2009*

New COS Release
Today I released a new version of the COS library, best

업로드 모듈 COS 라이브러리

- ▶ 좌측 메뉴의 com.oreilly.servlet 클릭
- ▶ 하단의 cos-26Dec2008.zip 파일 다운

Download

This is a .zip readable by "jar", newer releases are at the top.
To be notified when new versions release, [subscribe here](#).
Be sure to check out the [FAQ](#) and [Javadocs](#) below.

Version	Comments
cos-26Dec2008.zip	File upload improvements: <ul style="list-style-type: none">• Added support for Servlets 2.4 and Java 5.• Added an ExceededSizeException type to make catching easier.• Added support for EBCDIC machines.• Added a workaround for browsers that send Content-Length of -1.• Added a workaround for Opera missing parameter names.

업로드 모듈 COS 라이브러리

- ▶ 다운로드 한 파일의 압축을 푼 후 lib 폴더 내의 cos.jar 파일을 이클립스 라이브러리 폴더에 추가함
 - ▶ ch10 디렉터리의 WebContent\WEB-INF\lib 폴더에 cos.jar를 복사하여 붙여 넣음
- ▶ MultipartRequest 클래스
 - ▶ COS 라이브러리에서 파일 업로드를 담당하는 클래스
 - ▶ 파일 업로드를 담당하는 생성자와 여러 가지 메소드를 포함함

```
MultipartRequest(javax.servlet.http.HttpServletRequest request,  
                 java.lang.String saveDirectory,  
                 int maxPostSize,  
                 java.lang.String encoding,  
                 FileRenamePolicy policy)
```

업로드 모듈 COS 라이브러리

▶ MultipartRequest 클래스 메소드

메소드	설명
getParameterNames()	전송타입이 file이 아닌 파라미터들의 이름을 Enumeration 타입으로 반환함
getParameterValues()	폼에서 전송된 파라미터 값들을 배열로 받아옴
getParameter()	request 객체에 있는 지정된 이름의 파라미터 값을 가져옴
getFileNames()	파일을 여러 개 업로드할 경우 타입이 file인 파라미터 이름들을 Enumeration 타입으로 반환함
getFileSystemName()	서버에 실제로 업로드된 파일의 이름을 반환함
getOriginalFileName()	클라이언트가 업로드한 파일의 원본 이름을 반환함
getContentType()	업로드된 파일의 마임 타입을 반환함
getFile()	서버에 업로드된 파일 객체 자체를 반환함

파일 업로드 구현

▶ 파일 업로드(p.312~321)

파일 업로드 폼

올린 사람 :

제목 :

파일명1 :

파일명2 :

파일 업로드 폼

올린 사람 :

제목 :

파일명1 :

파일명2 :

[업로드 확인 및 다운로드 페이지 이동](#)

올린 사람 : 홍길동
제목 : 파일업로드
파일명1 : [uptest2.txt](#)
파일명2 : [uptest1.txt](#)

fileUploadForm.jsp

```
<body>
<form action="fileUpload.jsp" method="post" enctype="multipart/form-data">

~

<tr>
  <td>파일명1 : </td><td><input type="file" name="fileName1" id = "fileName1"></td>
</tr>
<tr>
  <td>파일명2 : </td><td><input type="file" name="fileName2" id = "fileName2"></td>
</tr>
<tr>
  <td colspan=2 align=center><input type="submit" value="전송"></td>
</tr>
</table>
</form>
```

fileUpload.jsp

```
<%@ page import="com.oreilly.servlet.MultipartRequest" %>
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy" %>
<%@ page import="java.util.*" %>
<%
    String uploadPath=request.getRealPath("/upload");

    int size = 10*1024*1024;
    String name="";
    String subject="";
    String filename1="";
    String filename2="";
    String origfilename1="";
    String origfilename2="";

    try{
        MultipartRequest multi=new MultipartRequest(request,
            uploadPath,
            size,
            "UTF-8",|
            new DefaultFileRenamePolicy());

        name=multi.getParameter("name");
        subject=multi.getParameter("subject");

        Enumeration files=multi.getFileNames();

        String file1 =(String)files.nextElement();
        filename1=multi.getFilesystemName(file1);
        origfilename1= multi.getOriginalFileName(file1);
```

```

        String file2 =(String)files.nextElement();
        filename2=multi.getFilesystemName(file2);
        origfilename2=multi.getOriginalFileName(file2);

    }catch (Exception e){
        e.printStackTrace();
    }
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<form name="filecheck" action="fileCheck.jsp" method="post">
    <input type="hidden" name="name" value="<%=name%>">
    <input type="hidden" name="subject" value="<%=subject%>">
    <input type="hidden" name="filename1" value="<%=filename1%>">
    <input type="hidden" name="filename2" value="<%=filename2%>">
    <input type="hidden" name="origfilename1" value="<%=origfilename1%>">
    <input type="hidden" name="origfilename2" value="<%=origfilename2%>">
</form>
<a href="#" onclick="javascript:filecheck.submit()">업로드 확인 및 다운로드 페이지 이동</a>
</body>
</html>

```

fileCheck.jsp

```
<%
    request.setCharacterEncoding("UTF-8");
    String name=request.getParameter("name");
    String subject=request.getParameter("subject");
    String filename1=request.getParameter("filename1");
    String filename2=request.getParameter("filename2");
    String origfilename1=request.getParameter("origfilename1");
    String origfilename2=request.getParameter("origfilename2");
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
출린 사람 : <%=name %><br>
제목 : <%=subject %><br>
파일명1 : <a href="file_down.jsp?file_name=<%=filename1 %>"><%=origfilename1 %></a><br>
파일명2 : <a href="file_down.jsp?file_name=<%=filename2 %>"><%=origfilename2 %></a><p>
</body>
```

file_down.jsp

```
<%@page import="java.net.URLEncoder"%>
<%@ page import="java.io.File"%>
<%@ page import="java.io.*"%>

~

<body>
<%
    String fileName = request.getParameter("file_name");

    String savePath = "upload";
    ServletContext context = getServletContext();
    String sDownloadPath = context.getRealPath(savePath);
    String sFilePath = sDownloadPath + "\\ " + fileName;
    byte b[] = new byte[4096];
    FileInputStream in = new FileInputStream(sFilePath);
    String sMimeType = getServletContext().getMimeType(sFilePath);
    System.out.println("sMimeType>>>" + sMimeType);

    if (sMimeType == null)
        sMimeType = "application/octet-stream";

    response.setContentType(sMimeType);
    String agent = request.getHeader("User-Agent");
    boolean ieBrowser = (agent.indexOf("MSIE") > -1) || (agent.indexOf("Trident") > -1);
```

file_down.jsp

```
if (ieBrowser) {
    fileName = URLEncoder.encode(fileName, "UTF-8").replaceAll("\\\\+", "%20");
} else {
    fileName = new String(fileName.getBytes("UTF-8"), "iso-8859-1");
}

response.setHeader("Content-Disposition", "attachment; filename= " + fileName);

ServletOutputStream out2 = response.getOutputStream();
int numRead;

while ((numRead = in.read(b, 0, b.length)) != -1) {
    out2.write(b, 0, numRead);
}
out2.flush();
out2.close();
in.close();
```

}%>

</body>

The background features abstract geometric shapes in various shades of gray and white. On the left, a tall, narrow, light gray triangle points downwards. On the right, there is a complex arrangement of overlapping triangles and polygons in darker and lighter gray tones, creating a layered, architectural effect. The central text is positioned between these two main graphic elements.

11. 이미지 섬네일

섬네일 이미지 만들기

- ▶ JAI(Java Advanced Imaging) API 다운로드 및 설치
 - ▶ <http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-java-client-419417.html>

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux Auto-installation for Java Plug-in	2.51 MB	jai-1_1_2_01-linux-i586-jar.zip
Linux CLASSPATH Install	2.48 MB	jai-1_1_2_01-lib-linux-i586.tar.gz
Linux JDK Install	2.54 MB	jai-1_1_2_01-lib-linux-i586-jdk.bin
Linux JRE Install	2.54 MB	jai-1_1_2_01-lib-linux-i586-jre.bin
Solaris SPARC Auto-installation for Java Plug-in	18.04 MB	jai-1_1_2_01-solaris-sparc-jar.zip
Solaris SPARC CLASSPATH Install	18.07 MB	jai-1_1_2_01-lib-solaris-sparc.tar.gz
Solaris SPARC JDK Install	18.15 MB	jai-1_1_2_01-lib-solaris-sparc-jdk.bin
Solaris SPARC JRE Install	18.14 MB	jai-1_1_2_01-lib-solaris-sparc-jre.bin
Solaris x86 Auto-installation for Java Plug-in	2.65 MB	jai-1_1_2_01-solaris-i586-jar.zip
Solaris x86 CLASSPATH Install	2.62 MB	jai-1_1_2_01-lib-solaris-i586.tar.gz
Solaris x86 JDK Install	2.70 MB	jai-1_1_2_01-lib-solaris-i586-jdk.bin
Solaris x86 JRE Install	2.70 MB	jai-1_1_2_01-lib-solaris-i586-jre.bin
Windows Auto-installation for Java Plug-in	5.22 MB	jai-1_1_2_01-windows-i586-jar.zip
Windows CLASSPATH Install	5.29 MB	jai-1_1_2_01-lib-windows-i586.exe
Windows JDK Install	5.29 MB	jai-1_1_2_01-lib-windows-i586-jdk.exe
Windows JRE Install	5.29 MB	jai-1_1_2_01-lib-windows-i586-jre.exe

섬네일 이미지 만들기

- ▶ JAI(Java Advanced Imaging) API 다운로드 및 설치
 - ▶ 다운로드 후 설치함
 - ▶ 설치 폴더의 lib 폴더에서 jai_codec.jar 파일과 jai_core.jar 파일을 복사하여 WebContent\WEB-INF\lib 디렉토리에 붙여 넣음

썸네일 이미지 만들기

- ▶ 썸네일 이미지 예제 (p.343~345)
 - ▶ cos.jar 파일 라이브러리에 추가
 - ▶ image 폴더 WebContent에 추가

썸네일 이미지 폼 예제

이미지 파일 : E:\D\소영\부경대\2017-2\ [찾아보기...](#)

[전송](#)

-원본 이미지-



-썸네일 이미지-



thumbnail.jsp

```
<%@ page import="java.awt.Graphics2D" %>
<%@ page import="java.awt.image.renderable.ParameterBlock" %>
<%@ page import="java.awt.image.BufferedImage" %>
<%@ page import="javax.media.jai.JAI" %>
<%@ page import="javax.media.jai.RenderedOp" %>
<%@ page import="javax.imageio.ImageIO" %>
<%@ page import="com.oreilly.servlet.MultipartRequest" %>
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy" %>
<%@ page import="java.util.*" %>
<%@ page import="java.io.*" %>
<%
    ServletContext context = request.getServletContext();
    String imagePath=context.getRealPath("image");

    int size = 1*1024*1024 ;
    String filename="";

    try{
        MultipartRequest multi= new MultipartRequest(request,
            imagePath,
            size,
            "euc-kr",
            new DefaultFileRenamePolicy());

        Enumeration files=multi.getFileNames();

        String file =(String)files.nextElement();
        filename=multi.getFilesystemName(file);
    }
}
```

```

    }catch(Exception e){
        e.printStackTrace();
    }

    ParameterBlock pb=new ParameterBlock();
    pb.add(imagePath+"/"+filename);
    RenderedOp rOp=JAI.create("fileload",pb);

    BufferedImage bi= rOp.getAsBufferedImage();
    BufferedImage thumb=new BufferedImage(100,100,BufferedImage.TYPE_INT_RGB);

    Graphics2D g=thumb.createGraphics();
    g.drawImage(bi,0,0,100,100,null);

    File file=new File(imagePath+"/sm_"+filename);
    ImageIO.write(thumb,"jpg",file);

```

}%>

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
-원본 이미지-<br>
<p>
-썸네일 이미지-<br>

</body>

```