

14장. 파일 업로드 / 다운로드

1. 파일 업로드의 개요

(1) 파일 업로드에 필요한 요소

- ① 파일을 입력받는 폼의 method 및 enctype타입 :
=> method="post" enctype="multipart/form-data"
- ② 업로드 할 파일이 저장될 폴더
=> storage 폴더
- ③ 파일을 업로드하고 폼을 분석하는 라이브러리
=> cos.jar

(2) 파일 업로드형 폼의 형태

- ① <form>태그의 method가 "post"이고, enctype이 "multipart/form-data"
- ② <input>태그의 type속성의 값을 "file"로 지정
 - * 사용 예
`<input type="file" name="selectfile">`
- ③ 작성 방법
 - 방법1 : <form> 태그의 속성을 사용하는 방법
`<form name="formName" method="post" enctype="multipart/form-data">`
 - 방법2 : <input type="submit">에서 별도의 속성을 지정하는 방법
`<input type="submit">`에 HTML5에서 새로 추가된 formenctype, formmethod속성을 사용
 - * 사용 예
`<input type="submit" formmethod="post", formenctype="multipart/form-data" value="업로드 전송" >`

(3) Eclipse에서는 가상폴더, 실제폴더가 따로 있다.

가상폴더

- D:\JSP_ycs\workspace\file\WebContent\storage

실제폴더

- D:\JSP_ycs\workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\file\storage

2. 파일 업로드 라이브러리 - cos.jar

- 파일을 업로드하고 폼으로부터 넘어오는 데이터를 얻어내는데 사용
- 파일을 업로드하면 서버상의 특정 폴더에 저장되고, 업로드한 파일 정보는 데이터베이스 테이블에 저장

(1) 다운로드 및 파일 배치

- 사이트 : <http://www.servlets.com/cos>
=> cos-26Dec2008.zip 다운로드
=> 압축을 풀면, cos-26Dec2008/lib/cos.jar 가 있다.
- cos.jar 파일을 해당 프로젝트의 라이브러리 폴더([프로젝트]-[Webcontent]-[WEB-INF]-[lib])에 넣어서 사용

(2) MultipartRequest 클래스

- 파일 업로드 및 폼 요소를 처리
- 생성자

```
MultipartRequest( javax.servlet.http.HttpServletRequest request, //request 객체
                  java.lang.String saveDirectory,           //파일 업로드할 폴더
                  int maxPostSize,                          //업로드 할 파일의 최대크기
                  java.lang.String encoding,                 //인코딩 방식
                  FileRenamePolicy policy)                  //같은 파일 덮어쓰기 방지 설정
```
- 객체 생성의 예

```
MultipartRequest upload = new MultiPartRequest(request, fileSave, 1024*5, "utf-8", new
DefaultFileRenamePolicy( ));
```

(3) 메소드

- `getContentType(java.lang.String name) : java.lang.String`
업로드된 파일의 콘텐츠 타입을 반환하는 것으로, 업로드된 파일이 없으면 null을 리턴
- `getFile(java.lang.String name) : java.io.File`
서버 상에 업로드된 파일을 File 객체 타입으로 리턴한다. 업로드된 파일이 없다면 null을 리턴
- `getFileNames() : java.util.Enumeration`
폼의 요소들 중 <input type="file">로 된 파라미터들을 받아서 Enumeration 타입의 객체를 리턴
- `getFilesystemName(java.lang.String name) : java.lang.String`
사용자가 업로드한 파일명을 리턴한다. 서버상에 업로드된 파일명이 리턴
- `getOriginalFileName(java.lang.String name) : java.lang.String`
사용자가 업로드한 파일의 원래 파일명을 리턴한다.
파일명이 중복될 경우 이름이 변경되므로, 변경되기 전의 원래 파일명이 리턴
- `getParameter(java.lang.String name) : java.lang.String`
name에 해당하는 파라미터의 값을 리턴
- `getParameterNames() : java.util.Enumeration`
폼의 요소들 중 <input type="file">아닌 파라미터들을 Enumeraton 객체 타입으로 리턴
- `getParameterValues(java.lang.String name) : java.lang.String[]`
하나의 파라미터에 대해 여러 개의 값을 가지는 <input type="checkbox">와 같은 경우에
파라미터의 값을 얻어내기 위해 사용

예제1. 파일 업로드

Dynamic Web Project : 14-file

JSP File : WebContent/fileForm.jsp
WebContent/fileUpload.jsp
WebContent/fileDownload.jsp

Folder : storage
: WebContent/storage

<실행 결과>

- 파일 업로드

The screenshot shows a web browser window at <http://localhost:8080/file/file>. The form has a title field with 'test' and a content area with 'testing'. Below the content area, there is a file selection button and a '파일업로드' (File Upload) button. The output area shows a success message '업로드 완료!!' (Upload complete!!) and a list of file details:

- 제목 : test
- 내용 : testing
- 파일 : [Chrysanthemum.jpg](#) Chrysanthemum.jpg
- 크기 : 879394
- 파일 : null null
- 크기 : 0

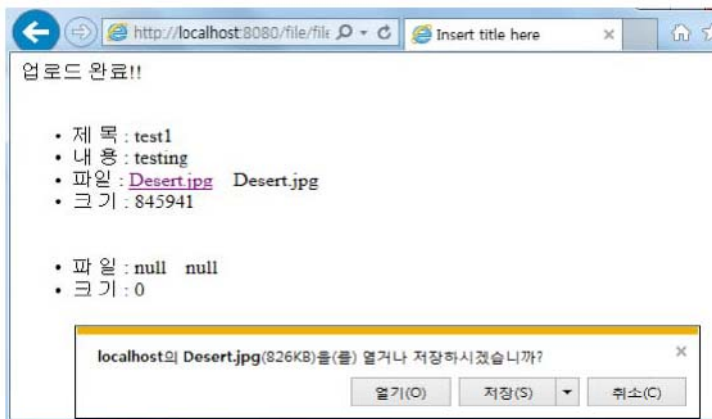
The Eclipse IDE console shows the following output:

```
Tomcat v8.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_71\bin\javaw.exe (2016. 3. 6. 오전 10:41:11)  
공고: Starting ProtocolHandler [ajp13-0009]  
3월 06, 2016 10:41:14 오전 org.apache.catalina.startup.Catalina start  
정보: Server startup in 1155 ms  
실제폴더 = D:\jsp_ee\workspace\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\file\storage
```

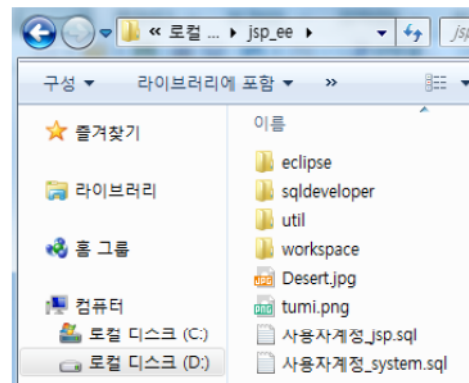
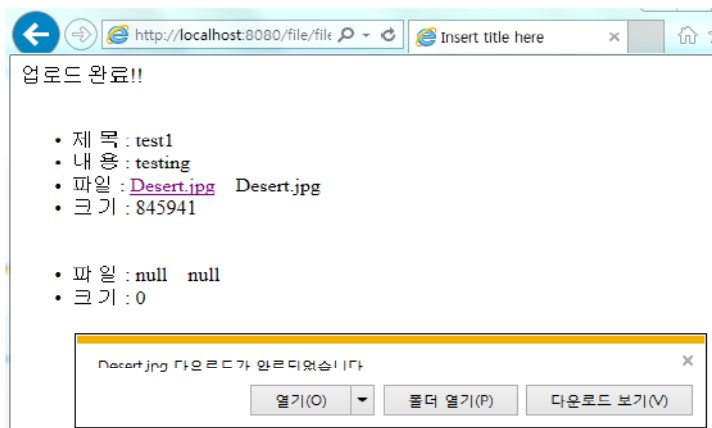
- 파일 다운로드

The screenshot shows the same web browser window, but the file selection button now shows 'C:\Users\Public\Pictures\Sample Pictures\Desert.jpg'. The '파일업로드' button is still present. The output area shows a success message '업로드 완료!!' (Upload complete!!) and a list of file details:

- 제목 : test1
- 내용 : testing
- 파일 : [Desert.jpg](#) Desert.jpg
- 크기 : 845941
- 파일 : null null
- 크기 : 0



Tomcat v8.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_71\bin\javaw.exe (2016. 3. 7. 오후 5:00:24)
 실제폴더 = D:\jsp_ee\workspace\metadata\plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\file\storage
 다운로드할 파일 : D:\jsp_ee\workspace\metadata\plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\file\storage\Desert.jpg



15장. EL/JSTL

- <% %> <%= %>를 사용하지말고 EL/JSTL를 쓰기를 권장
- 자바는 자바대로, 웹은 웹으로 분리

1. EL (Expression Language)

- EL은 JSTL에 소개된 내용으로 JSP 2.0에 추가된 기능이며 JSP의 기본문법을 보완하는 역할을 한다

(1) EL에서 제공하는 기능

- JSP의 네 가지 기본 객체가 제공하는 영역의 속성 사용
- 집합 객체에 대한 접근 방법 제공
- 수치 연산, 관계 연산, 논리 연산자 제공
- 자바 클래스 메소드 호출 기능 제공
- 표현 언어만의 기본 객체 제공
표기법 : `${ expr }`

(2) java 표현식과 el 표현식 비교

- 데이터 전달

`http://localhost:8080/el_jstl/elInput.jsp?x=5&y=7`

- java 표현식

```
int x = Integer.parseInt(request.getParameter("x"));
int y = Integer.parseInt(request.getParameter("y"));
```

- el 표현식

```
${param['x'] }
${param['y'] }
```

```
${param['x'] } + ${param['y'] } = ${param['x'] + param['y']}
```

2. JSTL (JSP Standard Tag Library)

(1) 사용 라이브러리

- 사이트 : `http://tomcat.apache.org/`
 - Impl: `taglibs-standard-impl-1.2.5.jar`
 - Spec: `taglibs-standard-spec-1.2.5.jar`
 - EL: `taglibs-standard-jstlel-1.2.5.jar`
- Eclipse에서 project의 WebContent\WEB-INF\lib에 복사

(2) JSTL core

- 출력 태그 : `<c:out>`
- 변수 설정 및 삭제 태그 : `<c:set>`, `<c:remove>`
- 예외 처리 태그 : `<c:catch>`
- 조건 처리 태그 : `<c:if>`, `<c:choose>`, `<c:when>`, `<c:otherwise>`
- 반복 처리 태그 : `<c:forEach>`, `<c:forTokens>`
- 페이지 처리 태그 : `<c:import>`, `<c: redirect>`, `<c:url>`, `<c:param>`

(3) 그 외 라이브러리

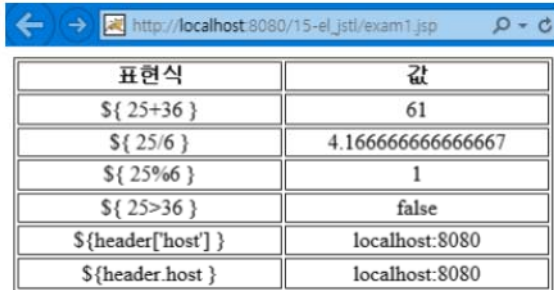
`fmt`, `xml`, `sql`

예제1.

Dynamic Web Project : 15-el_jstl

JSP File : WebContent/exam1.jsp

<실행 결과>



표현식	값
<code>\${ 25+36 }</code>	61
<code>\${ 25/6 }</code>	4.166666666666667
<code>\${ 25%6 }</code>	1
<code>\${ 25>36 }</code>	false
<code>\${header['host']}</code>	localhost:8080
<code>\${header.host}</code>	localhost:8080

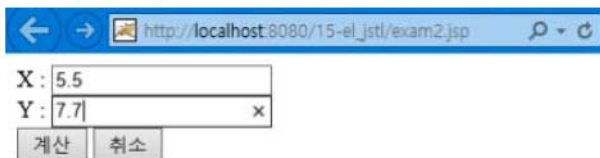
예제2.

Dynamic Web Project : 15-el_jstl

JSP File : WebContent/exam2.jsp

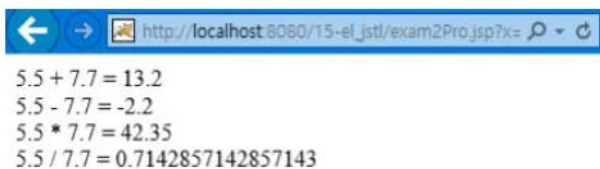
WebContent/exam2Pro.jsp

<실행 결과>



X :

Y :



5.5 + 7.7 = 13.2
5.5 - 7.7 = -2.2
5.5 * 7.7 = 42.35
5.5 / 7.7 = 0.7142857142857143

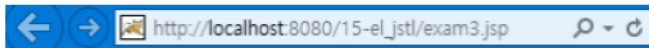
예제3.

Dynamic Web Project : 15-el_jstl

JSP File : WebContent/exam3.jsp

Library File : WebContent/WEB-INF/lib/taglibs-standard-impl-1.2.5.jar
WebContent/WEB-INF/lib/taglibs-standard-jstlel-1.2.5.jar
WebContent/WEB-INF/lib/taglibs-standard-spec-1.2.5.jar

<실행 결과>



**** 변수 설정 ****

나의 이름은 홍길동입니다.
내 나이는 25살 입니다.
나의 키는 입니다.

**** 변수 삭제 ****

나의 이름은 입니다.
내 나이는 25살 입니다.
나의 키는 입니다.

**** forEach ****

1 2 3 4 5 6 7 8 9 10
1~10 까지의 합 = 55

10 20 30 40 50

**** 문자열 분리 ****

소나타
그렌저
아우디
링컨
페라리

**** if ****

청소년

**** 다중 if ****

빨강 초록 파랑 보라 하늘

16장. include와 페이지 이동

1. include 하기

* directive

<%@ page %>

<%@ include file=""%> => 먼저 끌고 와서 컴파일함

<%@ taglib %>

* jsp 태그의 include

<jsp:include page=" " />=> 컴파일한 후 끌고 옴

*

import (지시 : 알려만 주는 것)

include (포함)

예제1. 다른 파일을 include 하기

- jsp파일에 다른 jsp 파일을 포함하기

Dynamic Web Project : 16-include_forward

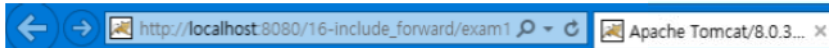
JSP File : WebContent/exam1.jsp
WebContent/today.jsp
WebContent/image.jsp

Folder : image
: WebContent/image

Image File : WebContent/image/cupra.jpg

<실행 결과>

- today.jsp 파일에 String name = "또치"; 있을 때



HTTP Status 500 - Unable to compile class for JSP:

type Exception report

message Unable to compile class for JSP:

description The server encountered an internal error that prevented it from fulfilling this request.

exception

org.apache.jasper.JasperException: Unable to compile class for JSP:

An error occurred at line: 12 in the jsp file: /today.jsp

Duplicate local variable name

9: </head>

10: <body>

11: <%

12: String name = "또치";

13: %>

14:

15: 오늘 날짜 : <%=new Date() %>

- today.jsp 파일에 String name = "또치"; 없을 때



**** include directive ****

오늘 날짜 : Sun Jul 17 12:29:36 KST 2016

**** include JSP tag ****



image.jsp name = 또치

image.jsp name = 홍길동

2. 페이지 이동

① sendRedirect

A페이지에서 B페이지 이동

데이터는 이동되지 않는다.

URL는 A페이지에서 요청하면 B페이지로 바뀌는것을 볼 수 있다

② forward(*.jsp → *.jsp)

RequestDispatcher(*.java → *.jsp)

A페이지와 B페이지가 같은 request, response 사용한다.(데이터 공유)

URL는 A페이지가 보이지만 결과화면은 B페이지가 보인다.

* JSP 내장 객체

page < request < session < application

* JSTL 내장 객체

pageScope < requestScope < sessionScope < applicationScope

예제2. 페이지 이동하기

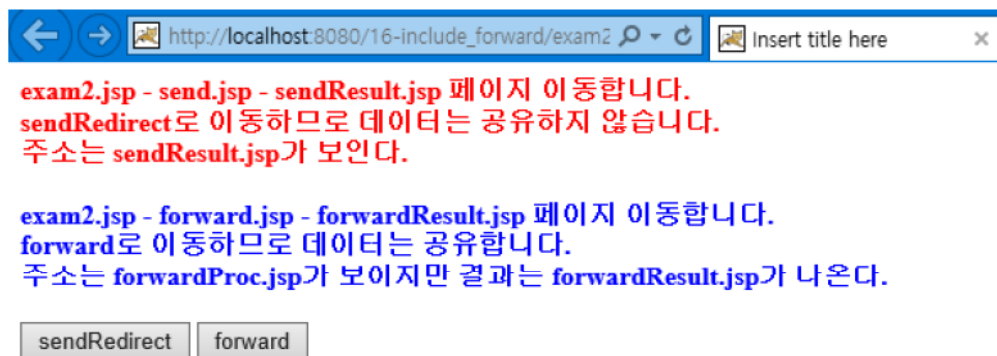
- sendRedirect 이용해서 페이지 이동하기
- forward 이용해서 페이지 이동하기

Dynamic Web Project : 16-include_forward

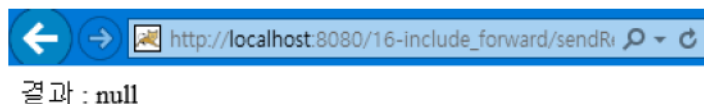
JSP File : WebContent/exam2.jsp
WebContent/send.jsp
WebContent/sendResult.jsp
WebContent/forward.jsp
WebContent/forwardResult.jsp

<실행 결과>

- exam2.jsp



- send.jsp, sendResult.jsp



- forward.jsp, forwardResult.jsp

