

# 파일 업로드

강사 : 강병준

# 파일업로드를 위한 기본 폼

- ❖ 웹 브라우저를 통해서 파일을 전송하기 위해서 폼 구성
  - form 태그의 속성들 중 input 태그가 있는데 이들 중 type 속성의 속성값이 file인 `<input type="file">` 태그는 파일을 선택할 수 있는 창을 제공
  - form 태그의 속성들 중 method의 속성값은 "post"를 enctype의 속성값은 "multipart/form-data"을 사용
    - ✓ `<form name="formName" method="post" enctype="multipart/form-data">`  
`<input type="file" name="selectfile ">` `</form>`
- ❖ `<form method="post">`의 형태로 전송한 폼에 담겨진 파라미터들은 request 객체를 통해서 해당되는 값을 얻음
- ❖ form 엘리먼트에는 enctype을 multipart/form-data라는 attribute를 기재해야 하는데 이를 기재하지 않으면 웹 서버로 데이터를 전송할 때 파일 내용을 전송하는 것이 아니라 파일의 경로명만 전송되며 이 속성이 제대로 동작하도록 하려면 form의 전송 방식이 post 방식이어야만 동작합니다.
  - enctype="multipart/form-data"로 전송한 폼에 담겨진 파라미터들에 대한 이름과 값을 얻어내고 `<input type="file">`로 지정된 파일을 서버상의 한 폴더에 업로드하기 위해서 특별한 컴포넌트가 필요.
- ❖ cos.jar를 사용
  - cos.jar 파일 다운로드 <http://www.servlets.com>
  - cos-05Nov.zip파일의 압축 해제
    - ✓ cos\lib폴더 안에 있는 cos.jar 파일을 톰캣홈 lib폴더에 복사
  - 안전한 서비스를 위해 cos.jar파일을 톰캣홈\commons\lib폴더에 복사
  - 폴더에 업로드될 파일을 모아놓을 저장소인 fileSave폴더를 만듦

# 파일 업로드하고 폼 데이터를 분석 도구

## ❖ 파일업로드 및 폼요소처리 MultipartRequest 클래스

- COS 라이브러리에서가장핵심적인역할을하는클래스로 파일 업로드 담당
- MultipartRequest의생성자

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

인자	설명
request	MultipartRequest와 연결될 request 객체를 의미한다.
saveDirectory	서버 측에 파일이 실질적으로 저장될 경로를 의미한다.
maxPostSize	한번에 업로드 할 수 있는 최대 파일 크기를 의미한다.
encoding	파일의 인코딩 방식을 의미한다.
policy	파일 이름 중복 처리를 위한 인자를 의미한다.

# MultipartRequest 클래스의 메소드

Return type	Method
java.lang.String	getContentType(java.lang.String name):업로드된 파일의 콘텐츠 타입을 반환. 업로드된 파일이 없으면 null을 반환한다.
java.io.File	getFile(java.lang.String name):서버 상에 업로드된 파일의 파일객체를 반환. 업로드된 파일이 없다면 null을 반환한다.
java.util.Enumeration	getFileNames():폼요소 중 input 태그속성이 file로 된 파라미터의 이름들을 반환. upload된 파일이 없으면 비어있는 Enumeration을 반환한다.
java.lang.String	getFileName(java.lang.String name):사용자가 지정해서 서버 상에 실제로 업로드된 파일명을 반환한다.
java.lang.String	getOriginalFileName(java.lang.String name):사용자가 지정해서 서버 상에 업로드된 파일명을 반환. 이 때의 파일명은 파일중복을 고려한 파일명 변경전의 이름을 말한다.
java.lang.String	getParameter(java.lang.String name):스트링으로 주어진 이름에 대한 값을 반환. 값없이 파라미터가 전송되었거나, 해당되는 이름의 파라미터가 전송이 안 되었을 경우 null을 반환한다.
java.util.Enumeration	getParameterNames():모든 파라미터 이름을 Enumeration으로 반환한다.
java.lang.String[]	getParameterValues(java.lang.String name):주어진 이름에 대한 값을 스트링 배열로 반환. 파라미터가 전송되지 않았을 때는 null을 반환한다.

# 폼 데이터분석 및 파일 업로드 구현

## ❖ 파일을 선택할 수 있는 fileSelect.jsp 페이지를 코딩하고 저장

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

## ❖ 폼 데이터를 분석하고 파일 업로드를 구현할 fileUpload.jsp 페이지를 코딩하고 저장

- `multi = new  
MultipartRequest(request,realFolder,maxSize,encType,  
new DefaultFileRenamePolicy());`

# fileSelect.jsp

```
<%@ page contentType= "text/html; charset=euc-kr"%>
<html>
<head>
<title>파일 업로드 예제</title>
<meta http-equiv= "Content-Type"
      content="text/html; charset=utf-8">
</head>
<body>
<form name= "fileForm" method="post"
      enctype="multipart/form- data" action="fileUpload.jsp">
  작성자: <input type= "text" name="user"><br>
  제 목: <input type= "text" name="title"><br>
  파일명: <input type= "file" name="uploadFile"><br>
  <input type= "submit" value="파일 올리기"><br>
</form>
</body>
</html>
```

# fileUpload.jsp – 1

```
<%@page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%@page import="com.oreilly.servlet.MultipartRequest,java.io.File"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><head><meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Insert title here</title></head><body>
<%
    request.setCharacterEncoding("utf-8");
    String fileSave = "/fileSave";
    String realPath = application.getRealPath(fileSave);
    int maxSize = 5 * 1024 * 1024; // 5M
    MultipartRequest mr = new MultipartRequest(request,realPath,
        maxSize,"utf-8",new DefaultFileRenamePolicy());
    String title = mr.getParameter("title");
    String writer = mr.getParameter("writer");
    String fileName = mr.getFileName("fileName");
    String oriName = mr.getOriginalFileName("fileName");
    File file = new File(realPath+"/"+fileName);
    int fileSize = (int)file.length();
%>
```



# fileUpload.jsp -2

**<h2>파일 업로드 결과</h2>**

**제목 : <%=title %><p>**

**작성자 : <%=writer %><p>**

**파일명 : <%=fileName %><p>**

**원래명 : <%=oriName %><p>**

**크기 : <%=fileSize %><p>**

**그림 :<br>**

****

**</body>**

**</html>**



## Style.css

```
a:link,a:visited,a:active {  
    text-decoration : none;  
}  
a:hover {  
    text-decoration : underline;  
    color : red;  
}  
TABLE, TD,TR, TH{  
    font-size:9pt;  
}  
p{  
    font-size:9pt;  
}
```

# 쇼핑물테이블

**member**

**id v12, passwd v12,  
name v20, address v50  
tel v20, reg\_date date**

**create table member(id varchar2(12) primary key,  
passwd varchar2(12) not null, name varchar2(20) not null,  
address varchar2(50), tel varchar2(20) not null,  
reg\_date date);**

**manager**

**managerId v12  
managerPasswd v12  
insert into manager("master",'1234');**

**create table manager (managerId varchar2(12) primary key,  
managerPasswd varchar2(12) not null);  
insert into manager values ('master','1234');  
commit;**

**book**

**book\_id pk number(10) 자동증가**

**book\_kind v3, book\_title v100**

**book\_price number(10), book\_count number(5)**

**author v20**

**publishing\_com v30**

**publishing\_date v15**

**book\_image v16 default 'nothing.jpg'**

**book\_content v(500)**

**discount\_rate number(3)**

**reg\_date date**

**create sequence book\_seq start with 1  
increment by 1 nocycle nocache;**

**create table book (book\_id number(10) primary key,  
book\_kind varchar2(3) not null, book\_title varchar2(100) not null,  
book\_price number(10) not null, book\_count number(5) not null,  
author varchar2(20),  
publishing\_com varchar2(30), publishing\_date varchar2(15),  
book\_image varchar2(16) default 'nothing.jpg',  
book\_content varchar2(500), discount\_rate number(3),  
reg\_date date);**

**bank**  
**account varchar2(30)**  
**bank varchar2(16)**  
**name varchar2(12)**

**create table bank (account varchar2(30) primary key,  
bank varchar2(16) not null, name varchar2(12) not null);**

**cart**  
**cart\_id number(10) pk 자동증가**  
**buyer varchar2(12), book\_id number(10)**  
**book\_title varchar2(100), buy\_price number(10)**  
**buy\_count number(5)**  
**book\_image varchar2(16) default 'nothing.jpg'**

**create sequence cart\_seq start with 1  
increment by 1 nocycle nocache;**

**create table cart (cart\_id number(10) primary key,  
buyer varchar2(12) not null, book\_id number(10) not null,  
book\_title varchar2(100), buy\_price number(10) not null,  
buy\_count number(5) not null,  
book\_image varchar2(16) default 'nothing.jpg');**

**buy**  
**buy\_id** number(12),     **buyer** v12  
**book\_id** v12,            **book\_title** v100  
**buy\_price** number(10)  
**buy\_count** number(5)  
**book\_image** v16 default 'nothing.jpg'  
**account** v50  
**deliveryName** v10  
**deliveryTel** v20  
**deliveryAddress** v80  
**sanction** v10 default '상품준비중'  
**buy\_date** date;

**create table buy** (**buy\_id** number primary key ,  
**buyer** varchar2(12) not null, **book\_id** varchar2(12),  
**book\_title** varchar2(100), **buy\_price** number(10) not null,  
**buy\_count** number(5) not null,  
**book\_image** varchar2(16) default 'nothing.jpg',  
**buy\_date** date,  
**account** varchar2(50), **deliveryName** varchar2(12),  
**deliveryTel** varchar2(20), **deliveryAddress** varchar2(80),  
**sanction** varchar2(12) default '준비중'  
);

# 쇼핑몰

## 관리자 영역

1. Table : member, manager, book, bank, cart, buy
2. 자바빈 – ShopBookDataBean, ShopBookDBBean

## 3. Jsp

- . 관리자 메인 : managerMain.jsp
- . managerLoginForm.jsp(로그인 화면)
- . managerLoginPro.jsp: 로그인 처리
- . 로그아웃 : managerLogout.jsp
  
- . 책 등록하는 메인 : bookRegisterForm.jsp
- . 책 등록처리 : bookRegisterPro.jsp
- . 등록된 상품 확인 : bookList.jsp
- . 등록된 상품을 수정하기 위한 폼 : bookUpdateForm.jsp
- . 등록된 상품을 수정 처리 : bookUpdatePro.jsp
- . 등록상품 삭제 : bookDeleteForm.jsp , bookDeletePro.jsp
- . 주문상표 List : orderedList.jsp

## **사용자 영역**

**1.자바빈 – CustomerDataBean.java,**

## **2.Jsp**

- . 쇼핑몰 메인 : shopMain.jsp**
- . 모듈페이지 : top.jsp, left.jsp , bottom.jsp**
- . 신간목록 : introList.jsp**
- . 쇼핑몰 목록 : list.jsp**
  
- . 책 등록하는 메인 : bookRegisterForm.jsp**
- . 책 등록처리 : bookRegisterPro.jsp**

## **장바구니 관리**

- . 자바빈 : CartDataBean.java, CartDBBean.j네**
- . 장바구니 목록추가 : cartInsert.jsp**
- . 장바구니 : cartList.jsp**
- . 수정 : updateCart.jsp , cartListDel.jsp**

## **구매관리**

- . Bean : BuyDataBean.java, BuyDBBean.java**
- . 구매폼 : buyForm.jsp, 목록추가 : buyPro.jsp**
- . 구매목록 표시 : buyKList.jsp**