

14장. 파일 업로드

파일업로드를 위한 기본 폼

1. 웹 브라우저를 통해서 파일을 전송하기 위해서 폼 구성
 - 1) form 태그의 속성들 중 input 태그가 있는데 이들 중 type 속성의 속성값이 file인 `<input type="file">` 태그는 파일을 선택할 수 있는 창을 제공
 - 2) 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>`
2. `<form method="post">`의 형태로 전송한 폼에 담겨진 파라미터들은 request 객체를 통해서 해당되는 값을 얻음
3. 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 동작.
 - ▶ enctype="multipart/form-data"로 전송한 폼에 담겨진 파라미터들에 대한 이름과 값을 얻어내고
`<input type="file">`로 지정된 파일을 서버상의 한 폴더에 업로드하기 위해 특별한 컴포넌트가 필요.
4. cos.jar를 사용
 - ① cos.jar 파일 다운로드 <http://www.servlets.com>
 - ② cos-05Nov.zip파일의 압축 해제
cosWlib폴더 안에 있는 cos.jar 파일을 톰캣홈 lib폴더에 복사
 - ③ 안전한 서비스를 위해 cos.jar파일을 톰캣홈Wcommonswlib폴더에 복사
 - ④ 폴더에 업로드될 파일을 모아놓을 저장소인 fileSave폴더를 만듦

파일 업로드하고 폼 데이터를 분석 도구

★ 파일업로드 및 폼요소처리 MultipartRequest 클래스

- ① COS 라이브러리에서가장핵심적인역할을하는클래스로 파일 업로드 담당
- ② 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 Enumeraton으로 반환
java.lang.String[]	getParameterValues(java.lang.String name):주어진 이름에 대한 값을 스트링 배열로 반환. 파라미터가 전송되지 않았을 때는 null을 반환

폼 데이터 분석 및 파일 업로드 구현

1. 파일을 선택할 수 있는 fileSelect.jsp 페이지를 코딩하고 저장
`<form name="fileForm" method="post" enctype="multipart/form-data" action="fileUpload.jsp">`
2. 폼 데이터를 분석하고 파일 업로드를 구현할 fileUpload.jsp 페이지를 코딩하고 저장
 - ▶ `multi = new MultipartRequest(request
 ,realFolder
 ,maxSize
 ,encType
 ,new DefaultFileRenamePolicy());`

fileSelect.jsp 예시

```
<%@ page contentType="text/html; charset=euc-kr"%>
<html>
  <head>
    <title>파일 업로드 예제 </title>
    <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
  </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 예시1

```
<%@page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%@page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
import="java.util.*,java.io.*"%>
<!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");
    int maxSize = 5 * 1024 * 1024;
    String fileSave = "/fileSave";
    String realPath = getServletContext().getRealPath(fileSave);
    MultipartRequest multi = new MultipartRequest(request,realPath, maxSize,"utf-8", new
    DefaultFileRenamePolicy());
    // Enumeration en = multi.getFileNames();
    // while(en.hasMoreElements()) {
    //input 태그의 속성이 file인 태그의 name 속성값 :파라미터이름
    //String name = (String)en.nextElement();
    //서버에 저장된 파일 이름
    String name1 = "uploadFile";
    String filename = multi.getFilesystemName(name1);
    //전송전 원래의 파일 이름
    String original = multi.getOriginalFileName(name1);
```

fileUpload.jsp 예시2

//전송된 파일의 내용 타입

```
String type = multi.getContentType(name1);
```

//전송된 파일속성이 file인 태그의 name 속성값을 이용해 파일객체생성

```
File file = multi.getFile(name1);
```

```
out.println("파라미터 이름 : " + name1 + "<br>");
```

```
out.println("실제 파일 이름 : " + original + "<br>");
```

```
out.println("저장된 파일 이름 : " + filename + "<br>");
```

```
out.println("파일 타입 : " + type + "<br>");
```

```
if(file!=null){
```

```
    out.println("크기 : " + file.length() + "<br>");
```

```
}
```

```
String name = multi.getParameter("name");
```

```
String title = multi.getParameter("title");
```

```
%>
```

```
작성자 : <%=name %><p>
```

```
제목   : <%=title %>
```

```
</body>
```

```
</html>
```