

파일 업로드

COS 라이브러리

❖ COS 라이브러리

- cos.jar
 - com.oreilly.servlet의 약자
 - <http://www.servlets.com/cos>
- 메이븐 의존 모듈
 - pom.xml

```
<dependencies>
  <dependency>
    <groupId>servlets.com</groupId>
    <artifactId>cos</artifactId>
    <version>05Nov2002</version>
  </dependency>
</dependencies>
```

파일 업로드

❖ 파일 전송을 위한 form

- 속성 설정
 - method="post"
 - enctype="multipart/form-data"
- 파일 지정 태그
 - <input>의 type="file" 지정

```
<form name = "사용자지정" method = "post" enctype = "multipart/form-data">  
  <input type = "file" name = "사용자지정">  
</form>
```

❖ 01_upload.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <form action="upload.do" method="post" enctype="multipart/form-data">
        글쓴이 : <input type="text" name="name"><br>
        제 &nbsp;    목 : <input type="text" name="title"><br>
        파일 지정하기 : <input type="file" name="uploadFile"><br>
        <input type="submit" value="전송">
    </form>
</body>
</html>
```

파일 업로드

❖ MultipartRequest

- cos의 파일 업로드 처리 담당
- 생성자

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

파일 업로드

❖ MultipartRequest

- 생성자 매개변수

매개 변수	설명
request	MultipartRequest와 연결할 request 객체
saveDirectory	서버 측에 저장될 경로
maxPostSize	최대 파일 크기
encoding	파일의 인코딩 방식 (파일 이름이 한글일 경우 매개 변수 값을 utf-8로 준다.)
policy	파일 중복 처리를 위한 매개 변수, policy는 중복 처리를 해주는 매개 변수로서 'b.bmp' 파일을 업로드 하였는데 다시 같은 파일을 업로드 할 경우 'b1.bmp' 등으로 자동으로 파일 중복 처리를 해준다. 매개 변수 값으로는 'new DefaultFileRenamePolicy()'를 사용한다.

파일 업로드

❖ MultipartRequest

- 주요 메서드

메소드명	설명
getParameterNames()	폼에서 전송된 파라미터의 이름을 Enumeration 타입으로 리턴한다.
getParameterValues()	폼에서 전송된 파라미터들을 배열로 받아온다.
getParameter()	객체에 있는 해당 파라미터의 값을 가져온다.
getFileName()	파일을 여러 개 업로드 할 경우 그 값들을 Enumeration 타입으로 리턴한다.
getFileSystemName()	서버에 실제로 업로드 된 파일의 이름을 의미한다.
getOriginalFileName()	클라이언트가 업로드한 파일의 원본 이름을 의미한다.
getContentType()	업로드 파일의 콘텐츠 타입을 얻을 때 사용한다.
getFile()	서버에 업로드 된 파일의 정보를 객체로 얻어낼 때 사용한다.

파일 업로드

❖ MultipartRequest

- 업로드 디렉토리 준비
 - WebContent/upload
 - 이클립스 프로젝트에서의 디렉토리와 실제 파일 업로드는 서비스되는 디렉토리는 다름
 - 실제 경로 알아내기

```
ServletContext context = getServletContext();  
String uploadFilePath = context.getRealPath(savePath);
```


파일 업로드

❖ MultipartRequest

- 업로드 디렉토리 준비
 - WebContent/upload
 - 이클립스 프로젝트에서의 디렉토리와 실제 파일 업로드는 서비스되는 디렉토리는 다름
 - 실제 경로 알아내기

```
ServletContext context = getServletContext();  
String uploadFilePath = context.getRealPath(savePath);
```

```
<workspace>\.metadata\.plugins\org.eclipse.wst.server.core\  
tmp0\wtpwebapps\web-study-10\upload
```

파일 업로드

❖ MultipartRequest

- 업로드 처리

```
import com.oreilly.servlet.MultipartRequest;
import com.oreilly.servlet.multipart.DefaultFileRenamePolicy;

String savePath = "upload"; //<- 여기를 바꿔주면 다운받는 경로가 바뀜
int uploadFileSizeLimit = 5 * 1024 * 1024; // 최대 업로드 파일 크기 5MB로 제한
String encType = "UTF-8";

MultipartRequest multi = new MultipartRequest(
    request, // request 객체
    uploadFilePath, // 서버상의 실제 디렉토리
    uploadFileSizeLimit, // 최대 업로드 파일 크기
    encType, // 인코딩 방법
    // 동일한 이름이 존재하면 새로운 이름이 부여됨
    new DefaultFileRenamePolicy());
```

파일 업로드

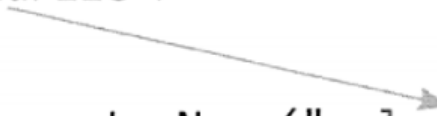
❖ MultipartRequest

○ 업로드 처리

- `enctype="multipart/form-data"` 방식에서는 `request.getParameter()`로 파라미터 추출 불가
- `MultipartRequest` 객체의 메서드로 추출

```
<input type="file" name="uploadFile">
```

```
String fileName = multi.getFilesystemName("uploadFile");
```



❖ UploadServlet.java

```
@WebServlet("/upload.do")
public class UploadServlet extends HttpServlet {
    :
    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {

        request.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charset=UTF-8");

        PrintWriter out = response.getWriter();
        // 여기를 바꿔주면 다운받는 경로가 바뀜
        String savePath = "upload";
        // 최대 업로드 파일 크기 5MB로 제한
        int uploadFileSizeLimit = 5 * 1024 * 1024;
        String encType = "UTF-8";

        ServletContext context = getServletContext();
        String uploadFilePath = context.getRealPath(savePath);
        System.out.println("서버상의 실제 디렉토리 :");
        System.out.println(uploadFilePath);
    }
}
```

❖ UploadServlet.java

```
try {
    MultipartRequest multi =
        new MultipartRequest(request,    // request 객체
            uploadFilePath,            // 서버상의 실제 디렉토리
            uploadFileSizeLimit,        // 최대 업로드 파일 크기
            encType,                    // 인코딩 방법
            // 동일한 이름이 존재하면 새로운 이름이 부여됨
            new DefaultFileRenamePolicy());

    // 업로드된 파일의 이름 얻기
    String fileName = multi.getFilesystemName("uploadFile");
    if (fileName == null) { // 파일이 업로드 되지 않았을때
        System.out.print("파일 업로드 되지 않았음");
    } else { // 파일이 업로드 되었을때
        out.println("<br> 글쓴이 : " + multi.getParameter("name"););
        out.println("<br> 제 &nbsp;   목 : " +
            multi.getParameter("title"););
        out.println("<br> 파일명 : " + fileName);
    } // else
} catch (Exception e) {
    System.out.print("예외 발생 : " + e);
} // catch
}
```

파일 업로드

❖ 다중 파일 업로드

```
<form action="upload2.do" method="post" enctype="multipart/form-data">  
  1. 파일 지정하기 : <input type="file" name="uploadFile01"><br>  
  2. 파일 지정하기 : <input type="file" name="uploadFile02"><br>  
  3. 파일 지정하기 : <input type="file" name="uploadFile03"><br>  
  <input type="submit" value="전송" >  
</form>
```

파일 업로드

❖ 다중 파일 업로드

- MultipartRequest

- getFileNames()로 업로드 파일 목록을 Enumeration 반환

```
Enumeration files = multi.getFileNames();
```

메소드	설명
boolean hasMoreElements()	데이터(element)가 존재한다면 true를 반환하고 없으면 false를 반환한다.
E nextElement()	데이터(element)를 얻어낸다.

```
while (files.hasMoreElements()) {  
    String file = (String) files.nextElement();  
    System.out.println(file);  
}
```

파일 업로드

❖ 다중 파일 업로드

- MultipartRequest

```
while (files.hasMoreElements()) {  
    String file = (String) files.nextElement();  
    String file_name = multi.getFilesystemName(file);  
    out.print("<br> 업로드된 파일명 : " + file_name);  
}
```

- 파일명 중복으로 새로운 파일명이 배정된 경우
 - 원본 파일명, 서버에서의 새로운 파일명을 모두 저장

```
while (files.hasMoreElements()) {  
    String file = (String) files.nextElement();  
    //중복된 파일을 업로드할 경우 파일명이 바뀐다.  
    String ori_file_name = multi.getOriginalFileName(file);  
    out.print("<br> 원본 파일명 : " + ori_file_name);  
}
```


❖ 02_upload.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <form action="upload2.do" method="post" enctype="multipart/form-data">
        1. 파일 지정하기 : <input type="file" name="uploadFile01"><br>
        2. 파일 지정하기 : <input type="file" name="uploadFile02"><br>
        3. 파일 지정하기 : <input type="file" name="uploadFile03"><br>
        <input type="submit" value="전송">
    </form>
</body>
</html>
```

❖ MultiUploadServlet.java

```
@WebServlet("/upload2.do")
public class MultiUploadServlet extends HttpServlet {
    :
    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException {

        request.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charset=UTF-8");

        PrintWriter out = response.getWriter();
        String savePath = "upload";
        int uploadFileSizeLimit = 5 * 1024 * 1024;
        String encType = "UTF-8";
        ServletContext context = getServletContext();
        String uploadFilePath = context.getRealPath(savePath);
```

❖ MultiUploadServlet.java

```
try {
    MultipartRequest multi = new MultipartRequest(request,
                                                    uploadFilePath, uploadFileSizeLimit, encType,
                                                    new DefaultFileRenamePolicy());

    Enumeration files = multi.getFileNames();
    while (files.hasMoreElements()) {
        String file = (String) files.nextElement();

        String file_name = multi.getFilesystemName(file);
        // 중복된 파일을 업로드할 경우 파일명이 바뀐다.
        String ori_file_name = multi.getOriginalFileName(file);

        out.print("<br> 업로드된 파일명 : " + file_name);
        out.print("<br> 원본 파일명 : " + ori_file_name);
        out.print("<hr>");
    }
} catch (Exception e) {
    System.out.print("예외 발생 : " + e);
} // catch
}
```