

# File Upload & Download

524730-1  
2020년 봄학기  
6/3/2020  
박경신

## 파일 전송을 위한 FORM 설정

### □ 파일 업로드 (File Upload)를 위한 JSP 페이지

- 웹 브라우저에서 서버로 파일을 전송하기 위해 JSP 페이지에 폼 태그를 작성할 때 몇 가지 중요한 규칙
  1. form 태그의 method 속성은 반드시 **POST** 방식으로 설정
  2. form 태그의 enctype 속성은 반드시 **multipart/form-data**로 설정
  3. form 태그의 action 속성은 파일 업로드를 처리할 JSP 파일로 설정
  4. 파일 업로드를 위해 input 태그의 **type** 속성을 **file**로 설정
    - 만약 여러 파일을 업로드하려면 2개 이상의 input 태그를 사용하고 name 속성에 서로 다른 값을 설정

```
<form action="..." method="POST" enctype="multipart/form-data">
...
<input type="file" name="file1" />
</form>
```

## File Upload

### □ 파일 업로드 (File Upload)

- 웹 브라우저에서 서버로 파일을 전송하여 서버에 저장하는 것
- 서버로 업로드할 수 있는 파일
  - ▣ 텍스트 파일, 바이너리 파일, 이미지 파일, 문서 등 다양한 유형이 있습니다.
- 웹 브라우저에서 서버로 파일을 전송하기 위해 JSP 페이지에 폼 태그 사용
- 전송된 파일을 서버에 저장하기 위해 오픈 라이브러리를 이용

## MultipartRequest를 이용한 파일 업로드

### □ MultipartRequest

- 웹 페이지에서 서버로 업로드되는 파일 자체만 다루는 클래스.
- 웹 브라우저가 전송한 multipart/form-data 유형과 POST 방식의 요청 파라미터 등을 분석한 후 일반 데이터와 파일 데이터를 구분하여 파일 데이터에 접근
- 한글 인코딩 값을 얻기 쉽고, 서버의 파일 저장 폴더에 동일한 파일명이 있으면 파일명을 자동으로 변경
- 오픈 라이브러리 cos.jar를 배포 사이트에서 직접 다운로드해서 사용
  - ▣ 배포 사이트: <http://servlets.com/cos/>
  - ▣ 다운로드 파일: cos-26Dec2008.zip
  - ▣ JSP 페이지에 page 딕레티브 태그의 import 속성을 사용하여 패키지 com.oreilly.servlet.\*을 설정

NOTE... 오픈 라이브러리 cos.jar 설치 방법

❶ 배포 사이트에서 cos-26Dec2008.zip 파일을 다운로드합니다.  
❷ cos-26Dec2008.zip 파일의 압축을 풉니다.  
❸ 압축을 푸 파일의 lib 폴더에 있는 cos.jar 파일을 찾습니다.  
❹ 웹 애플리케이션의 /WebContent/WEB-INF/lib/ 폴더에 cos.jar 파일을 포함합니다. 또는 서버 뿐만 아니라 /common/lib/ 폴더와 자바 JDK의 lib 폴더에 cos.jar 파일을 포함합니다.

## Multipart/form-data로 전송된 데이터 형식

```
[multipart/form-data; boundary=-----7d9202102082c]
--7d9202102082c
Content-Disposition: form-data; name="text1"
텍스트
-----7d9202102082c
Content-Disposition: form-data; name="file1"; filename="set-javab-home.bat"
Content-Type: application/octet-stream
set JAVA_HOME=c:\java\jdk1.6.0_07
set PATH=%JAVA_HOME%\bin;%PATH%
-----7d9202102082c
Content-Disposition: form-data; name="file2"; filename=""
Content-Type: application/octet-stream
-----7d9202102082c--
```

일반 파라미터

파일 데이터

## FileUpload API

- ▣ multipart/form-data로 전송된 데이터 처리
- ▣ <http://commons.apache.org/fileupload/>에서 다운로드
  - commons-fileupload-1.4.jar
- ▣ 추가 필요 라이브러리
  - Commons IO
  - <http://commons.apache.org/io/> 사이트에서 다운로드
  - commons-io-2.6.jar

## FileUpload API를 이용한 업로드 데이터 처리

```
// 1. multipart/form-data 여부 확인
boolean isMultipart = ServletFileUpload.isMultipartContent(request);
if (isMultipart) {
    // 2. 메모리나 파일로 업로드 파일 보관하는 FileItem의 Factory 설정
    DiskFileItemFactory factory = new DiskFileItemFactory();
    // 3. 업로드 요청을 처리하는 ServletFileUpload 생성
    ServletFileUpload upload = new ServletFileUpload(factory);
    // 4. 업로드 요청 파싱해서 FileItem 목록 구함
    List<FileItem> items = upload.parseRequest(request);
    Iterator<FileItem> iter = items.iterator();
    while (iter.hasNext()) {
        FileItem item = iter.next();
        // 5. FileItem이 폼 입력 항목인지 여부에 따라 알맞은 처리
        if (item.isFormField()) { // 텍스트 입력인 경우
            String name = item.getFieldName();
            String value = item.getString("euc-kr");
        } else { // 파일 업로드인 경우
            String name = item.getFieldName();
            String fileName = item.getName();
            String contentType = item.getContentType();
            boolean isInMemory = item.isInMemory();
            long sizeInBytes = item.getSize();
        }
    }
}
```

## 업로드 한 파일 처리 방법

- ▣ FileItem.write(File file) 메서드를 사용하는 방법
- ▣ FileItem.getInputStream() 메서드로 구한 입력 스트림으로부터 바이트 데이터를 읽어와 FileOutputStream을 사용해서 파일에 출력하는 방법
- ▣ FileItem.get() 메서드로 구한 바이트 배열을 FileOutputStream을 사용해서 파일에 출력하는 방법

## 자료실 - 업로드 한 파일 저장 방식

### ▣ 업로드 한 파일 저장 방식 두 가지

- DB의 BLOB에 저장
- 파일 시스템에 저장
  - ▣ 이 경우 저장할 때 사용하는 파일의 이름은 현재 시간 등의 값을 이용해서 생성
    - 개시글 번호와 첨부된 순서 번호를 함께 사용하기도 함
  - ▣ 실제 파일 이름은 DB에 저장

## 자료실 - 다운로드의 구현

### ▣ 다운로드 구현 시 고려 사항

- 응답 컨텐츠 타입은 application/octet-stream
- Content-Disposition 헤더로 파일명 지정
  - ▣ 파일명 설정시 ISO-8859-1로 인코딩 변환해서 설정

```
<%
    String fileName = new String(item.getFileName().getBytes("euc-kr"), "iso-8859-1");
    response.setContentType("application/octet-stream");
    response.setHeader("Content-Disposition",
        "attachment; filename=" + fileName + "UTF-8");
%>
```

### ▣ 실제 파일 전송

- response.getOutputStream()으로 구한 OutputStream에 파일 데이터 출력