



제12장

파일 업로드

1. 파일 업로드의 개요

파일 업로드(file upload)

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

1. 파일 업로드의 개요

파일 업로드를 위한 JSP 페이지

1. 웹 브라우저에서 서버로 파일을 전송하기 위해 JSP 페이지에 폼 태그를 작성할 때 몇 가지 중요한 규칙

```
<form action = "JSP 파일" method="POST" enctype="multipart/form-data">  
  <input type="file" name="요청 파라미터 이름">  
</form>
```

- ❶ form 태그의 method 속성은 반드시 POST 방식으로 설정
- ❷ form 태그의 enctype 속성은 반드시 multipart/form-data로 설정
- ❸ form 태그의 action 속성은 파일 업로드를 처리할 JSP 파일로 설정
- ❹ 파일 업로드를 위해 input 태그의 type 속성을 file로 설정
 - 만약 여러 파일을 업로드 하려면 2개 이상의 input 태그를 사용하고 name 속성에 서로 다른 값을 설정

enctype은 전송 데이터 형식 설정을 하는 부분이다.

1. application/x-www-form-urlencoded

디폴트 값. 서버로 전송되기 전에 url-encode 된다는 뜻.

2. multipart/form-data

파일 받으면서 설정해 준다. 이미지나 파일을 서버로 전송할 경우 이 방식을 사용한다.

3. text/plain

인코딩을 하지 않은 문자 그대로 상태를 전송한다는 의미.

1. 파일 업로드의 개요

파일 업로드 처리 방법

1. 단순한 자바 코드로 작성하여 처리할 수 없어 오픈 라이브러리인 cos.jar나 commonsfileupload.jar를 사용해야 함

파일 업로드 처리 방법

| 종류 | 특징 | 필요한 라이브러리 |
|-----------------------|-----------------|--|
| MultipartRequest 이용하기 | 가장 간단한 방법 | cos.jar |
| 아파치 API 이용하기 | 편리하고 강력한 API 제공 | commons-fileupload.jar commons-io.jar |

cos.jar : 웹 to 웹으로 전송해야되는 간단한 데이터, 진행상태 표시 구현이 필요없는 파일 업로드 기능 구현시 매우 간단하게 구현 가능한 장점을 가짐, 소스가 제공되므로 개발자 편의에 따라 소스를 수정하여 사용할 수도 있음. 업데이트가 잘 안됨.

Apache commons upload : 일반적인 파일 업로드 기능 외에 파일명이나 저장되어야 할 위치에 대한 조작 등 개발 의도에 따른 소스 구현이 좀 더 유연함. 업로드 진행상태를 알 수 있는 프로그레스바 구현을 ProgressListner를 통해 쉽게 구현 할 수 있음.

2. MultipartRequest를 이용한 파일 업로드

MultipartRequest

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

NOTE_ 오픈 라이브러리 cos.jar 설치 방법

- ① 배포 사이트에서 cos-26Dec2008.zip 파일을 다운로드합니다.
- ② cos-26Dec2008.zip 파일의 압축을 풉니다.
- ③ 압축을 푼 파일의 lib 폴더에 있는 cos.jar 파일을 찾습니다.
- ④ 웹 애플리케이션의 /WebContent/WEB-INF/lib/ 폴더에 cos.jar 파일을 포함합니다. 또는 서버 톰캣의 /common/lib/ 폴더와 자바 JDK의 lib 폴더에 cos.jar 파일을 포함합니다.

2. MultipartRequest를 이용한 파일 업로드

MultipartRequest 클래스 생성

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

MultipartRequest 생성자의 매개 변수

| 매개변수 | 설명 |
|---------------|--|
| request | Request 내장 객체를 설정합니다. |
| saveDirectory | 서버의 파일 저장 경로를 설정합니다. |
| maxPostSize | 파일의 최대 크기(바이트 단위)를 설정합니다. 최대 크기를 초과하면 IOException이 발생합니다. |
| encoding | 인코딩 유형을 설정합니다. |
| policy | 파일명 변경 정책을 설정합니다. saveDirectory에 파일명이 중복되는 경우 덮어쓰기 여부를 설정하며, 설정하지 않으면 덮어씁니다. |

2. MultipartRequest를 이용한 파일 업로드

[MultipartRequest 클래스 생성 예]

```
<%@ page import="com.oreilly.servlet.*"%>
<%@ page import="com.oreilly.servlet.multipart.*"%>
...(생략)...

MultipartRequest multi = new MultipartRequest(request,
    "C:\\upload", 5*1024*1024, "utf-8",
    new DefaultFileRenamePolicy())
```

2. MultipartRequest를 이용한 파일 업로드

MultipartRequest 메소드

1. 웹 브라우저에서 전송되는 요청 파라미터 중

- 일반 데이터는 `getParameter()` 메소드로 값을 받음
- 파일의 경우 `getFileNames()` 메소드를 이용하여 데이터를 받음

2. MultipartRequest 메소드의 종류

| 메소드 | 유형 | 설명 |
|---|-----------------------|--|
| <code>getContentType(String name)</code> | String | 업로드된 파일의 콘텐츠 유형을 반환합니다. 업로드된 파일이 없으면 null을 반환합니다. |
| <code>getParameter(String name)</code> | String | 요청 파라미터 이름이 name인 값을 전달받습니다. |
| <code>getParameterNames()</code> | java.util Enumeration | 요청 파라미터 이름을 Enumeration 객체 타입으로 반환합니다. |
| <code>getFile(String name)</code> | java.io File | 서버에 업로드된 파일에 대한 파일 객체를 반환합니다. 업로드된 파일이 없으면 null을 반환합니다. |
| <code>getFileNames()</code> | java.util Enumeration | 폼 페이지에 input 태그 내 type 속성 값이 file로 설정된 요청 파라미터의 이름을 반환합니다. |
| <code>getFileSystemName(String name)</code> | String | 사용자가 설정하여 서버에 실제로 업로드된 파일명을 반환합니다. 파일명이 중복되면 변경된 파일명을 반환합니다. |
| <code>getOriginalFileName(String name)</code> | String | 사용자가 업로드한 실제 파일명을 반환합니다. 파일명이 중복되면 변경 전의 파일명을 반환합니다. |

3. Commons-FileUpload를 이용한 파일 업로드

Commons-FileUpload

1. 파일 업로드 패키지
2. 서버의 메모리상에서 파일 처리가 가능하도록 지원
3. 오픈 라이브러리 commons-fileupload.jar, commons-io.jar 파일을 배포 사이트에서 직접 다운로드해서 사용
 - 배포 사이트: <http://commons.apache.org/downloads/>
 - 다운로드 파일: commons-fileupload-1.3.3-bin.zip, commons-io-2.6-bin.zip
4. JSP 페이지에 page 디렉티브 태그의 import 속성을 사용하여 패키지 org.apache.commons.fileupload.*를 설정

NOTE_ 오픈 라이브러리 Commons-FileUpload.jar, commons-io.jar 설치 방법

- ① 배포 사이트에서 commons-fileupload-1.3.3-bin.zip와 commons-io-2.6-bin.zip 파일을 다운로드합니다.
- ② 다운로드한 두 파일의 압축을 풉니다.
- ③ 압축을 푼 두 파일의 commons-fileupload-1.3.3-bin, commons-io-2.6-bin 폴더에 있는 commons-fileupload-1.3.2.jar, commons-io-2.6.jar 파일을 찾습니다.
- ④ 웹 애플리케이션의 /WebContent/WEB-INF/lib 폴더에 commons-fileupload-1.3.2.jar, commons-io-2.6.jar 파일을 포함합니다. 또는 서버 톰캣의 /common/lib 폴더와 자바 JDK의 lib 폴더에 commons-fileupload-1.3.2.jar, commons-io-2.6.jar 파일을 포함합니다.

3. Commons-FileUpload를 이용한 파일 업로드

1. DiskFileUpload 클래스의 메소드

| 메소드 | 유형 | 설명 |
|--|----------------|---|
| setRepositoryPath(String repositoryPath) | void | 업로드된 파일을 임시로 저장할 디렉터리를 설정합니다. |
| setSizeMax(long sizeMax) | void | 최대 파일의 크기를 설정합니다. |
| setSizeThreshold(int sizeThreshold) | void | 메모리상에 저장할 최대 크기를 설정합니다. |
| parseRequest(HttpServletRequest req) | List<FileItem> | multipart/form-data 유형의 요청 파라미터를 가져옵니다. |

2. FileItem 클래스의 메소드

| 메소드 | 유형 | 설명 |
|----------------------------|---------|---|
| isFormField() | boolean | 요청 파라미터가 파일이 아니라 일반 데이터인 경우 true를 반환합니다. |
| getFieldName() | String | 요청 파라미터의 이름을 얻어옵니다. |
| getString() | String | 기본 문자 인코딩을 사용하여 요청 파라미터의 값을 얻어옵니다. |
| getString(String encoding) | String | 설정된 문자 인코딩을 사용하여 요청 파라미터의 값을 얻어옵니다. |
| getName() | String | 업로드된 파일(경로 포함)의 이름을 얻어옵니다. |
| getSize() | long | 업로드된 파일의 크기를 얻어옵니다. |
| get() | byte[] | 업로드된 파일을 바이트 배열로 얻어옵니다. |
| isInMemory() | boolean | 업로드된 파일이 메모리에 저장된 상태인 경우 true를 반환하고, 임시 디렉터리에 저장된 경우 false를 반환합니다. |
| delete() | void | 파일과 관련된 자원을 삭제합니다. 메모리상에 저장된 경우 할당된 메모리를 반환하고, 임시 파일로 저장된 경우 파일을 삭제합니다. |
| write() | void | 파일과 관련된 자원을 저장합니다. |
| getContentType() | String | 웹 브라우저가 전송하는 콘텐츠 유형을 반환하고, 정의되어 있지 않은 경우 null을 반환합니다. |

감사합니다.

