

Chapter

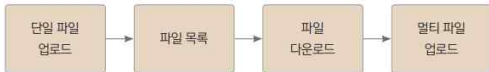
13

파일 업로드 및 다운로드

■ 학습 목표

- 파일을 첨부할 수 있는 자료실형 게시판을 만들기 위해 파일을 업로드하고 다운로드하는 기능을 구현해보겠습니다.
- 파일 업로드를 위한 <form> 태그를 통한 전송 방식 및 인코딩 방식, 파일이 저장될 폴더의 물리적 경로 설정 등을 학습합니다.
- 파일 다운로드를 위한 I/O 및 응답 헤더 설정 등도 학습합니다.

■ 학습 순서



■ 파일 업로드

- Tomcat 에서 제공하는 Part API를 사용해서 파일업로드를 구현
- 구현 진행 절차
 - 화면(폼) 작성
 - 데이터베이스 테이블 준비
 - DTO와 DAO 작성
 - 모두를 연동해 파일 업로드 완성

■ 작성폼

- <form> 의 method 속성은 post, enctype 속성은 multipart/form-data

```
<form method="post" enctype="multipart/form-data" action="업로드 처리 파일 경로">  
  <input type="file" name="input 이름" />  
</form>
```

■ 작품

```

<h3>파일 업로드</h3>
<span style="color: red;">${errorMessage }</span>
<form name="fileForm" method="post" enctype="multipart/form-data"
      action="UploadProcess.do" onsubmit="return validateForm(this);"> ❷
  제목 : <input type="text" name="title" /><br /> ❸
  카테고리(선택사항) :
    <input type="checkbox" name="cate" value="사진" checked />사진
    <input type="checkbox" name="cate" value="과제" />과제
    <input type="checkbox" name="cate" value="워드" />워드
    <input type="checkbox" name="cate" value="음원" />음원 <br /> ❹
  첨부 파일 : <input type="file" name="attachedFile" />
    <input type="submit" value="전송하기" /> ❺
</form>

```

[Note] 첨부파일의 name속성은 **ofile**로 해야합니다.

FileUploadMain.jsp

❷ 파일 전송을 위한 <form> 태그

❸ 입력을 위한 <form> 하위 태그들. 특히 첨부파일의 경우 type속성을 file로 설정

파일 업로드

제목 :

카테고리(선택사항) : ☒ 사진 ☐ 과제 ☐ 워드 ☐ 음원

첨부파일 : 선택된 파일 없음

■ 데이터베이스에 테이블 생성

컬럼명	데이터 타입	null 허용	키	기본값	설명
idx	number	N	기본키		일련번호, 기본키
title	varchar2(200)	N			제목
cate	varchar2(30)				카테고리
ofile	varchar2(100)	N			원본 파일명(original filename)
sfile	varchar2(30)	N			저장된 파일명(saved filename)
postdate	date	N		sysdate	등록한 날짜

```
create table myfile (
    idx number primary key,
    title varchar2(200) not null,
    cate varchar2(30),
    ofile varchar2(100) not null,
    sfile varchar2(30) not null,
    postdate date default sysdate not null
);
```

■ DTO 및 DAO 클래스 생성

MyFileDTO.java

```
package fileupload;

public class MyFileDTO {
    // 멤버 변수 ❶
    private String idx;
    private String title;      // 제목
    private String cate;      // 카테고리
    private String ofile;      // 원본 파일명
    private String sfile;      // 저장된 파일명
    private String postdate;   // 등록 날짜

    // 게터/세터 ❷
    public String getIdx() {
        return idx;
    }
}
```

■ DTO 및 DAO 클래스 생성 **MyFileDAO.java**

```
public class MyFileDAO extends DBConnPool { ❶
    // 새로운 게시물을 입력합니다.
    public int insertFile(MyFileDTO dto) { ❷
        int applyResult = 0; ❸
        try {
            String query = "INSERT INTO myfile ( "
                + " idx, title, cate, ofile, sfile) "
                + " VALUES ( "
                + " seq_board_num.nextval, ?, ?, ?, ?, ?)"; ❹

            pstmt = con.prepareStatement(query); ❺
            pstmt.setString(1, dto.getTitle()); ❻
            pstmt.setString(2, dto.getCate());
            pstmt.setString(3, dto.getOfile());
            pstmt.setString(4, dto.getSfile());

            applyResult = pstmt.executeUpdate(); ❼
        }
    }
}
```

- ❶ 커넥션풀(DBCP)을 통해 DB에 연결
- ❷ 새로운 게시물을 입력하기 위한 메서드

- ❹ insert 쿼리문 작성 및 실행

■ 파일 업로드 및 폼값 처리

- 일반적인 폼값은 request 내장 객체의 `getParameter()` 로 처리
- `type="file"` 인 경우에는 Part 클래스 사용

```
public class FileUtil {  
    // 파일 업로드  
    public static String uploadFile(HttpServletRequest req, String sDirectory)  
        throws ServletException, IOException { ❶  
        Part part = req.getPart("ofile"); ❷  
        String partHeader = part.getHeader("content-disposition"); ❸  
        String[] phArr = partHeader.split("filename="); ❹  
        String originalFileName = phArr[1].trim().replace("\\", "");  
        if (!originalFileName.isEmpty()) { ❺  
            part.write(sDirectory+ File.separator +originalFileName);  
        }  
  
        return originalFileName; ❻  
    }  
}
```

FileUtil.java

- ❶ 파일업로드를 위한 메서드
- ❷ file 타입으로 전송된 폼값을 받아 Part 객체를 생성
- ❸❹ Part 객체의 헤더값을 읽어와서 원본 파일명을 추출

partHeader=form-data; name="ofile"; filename="성유겸-요리사.jpeg"

〈input〉의 name속성값

원본파일명

■ 파일 업로드 및 폼값 처리 **FileUtil.java(계속)**

// 파일명 변경

```
public static String renameFile(String sDirectory, String fileName) {  
    String ext = fileName.substring(fileName.lastIndexOf(".")); ⑦  
    String now = new SimpleDateFormat("yyyyMMdd_HmsS").format(new Date()); ⑧  
    String newFileName = now + ext; ⑨  
    File oldFile = new File(sDirectory + File.separator + fileName);  
    File newFile = new File(sDirectory + File.separator + newFileName);  
    oldFile.renameTo(newFile); ⑩  
  
    return newFileName; ⑪  
}
```

파일명 변경을 위한 메서드

⑦ 파일명에서 확장자를 추출

⑧ “년월일_시분초” 형식의 파일명 생성

⑨ 파일명과 확장자 연결

⑩ 파일명 변경

■ 파일 업로드 및 폼값 처리 UploadProcess.java

```
@WebServlet("/13FileUpload/UploadProcess.do") ❶
@MultipartConfig( ❷
    maxFileSize = 1024 * 1024 * 1,
    maxRequestSize = 1024 * 1024 * 10
)
public class UploadProcess extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        try { ❸
            String saveDirectory = getServletContext().getRealPath("/Uploads"); ❹
            String originalFileName = FileUtil.uploadFile(req, saveDirectory); ❺
            String savedFileName = FileUtil.renameFile(saveDirectory, originalFileName);
            insertMyFile(req, originalFileName, savedFileName); ❻
            resp.sendRedirect("FileList.jsp");
        }
    }
}
```

작성폼에서 submit 했을때 요청을 받아
처리할 서블릿 클래스

❶ 요청명 매핑

❷ 파일업로드를 위한 멀티파트 설정. 첨부파일의 최대용량을 지정.

▶ maxFileSize : 개별 파일 1Mb

▶ maxRequestSize : 전체 파일 10Mb

❸ Uploads 폴더의 물리적 경로 얻어와서 파일을 업로드 및 파일명 변경

❹ DB에 insert 처리(다음 페이지)

■ 파일 업로드 및 품값 처리

UploadProcess.java(계속)

```
private void insertMyFile(HttpServletRequest req, String oFileName, String
sFileName) { ⑦
    String title = req.getParameter("title"); ⑧
    String[] cateArray = req.getParameterValues("cate");
    StringBuffer cateBuf = new StringBuffer();
    if (cateArray == null) {
        cateBuf.append("선택한 항목 없음");
    }
    else {
        MyFileDTO dto = new MyFileDTO(); ⑨
        dto.setTitle(title);
        dto.setCate(cateBuf.toString());
        dto.setOfile(oFileName);
        dto.setSfile(sFileName);

        MyFileDAO dao = new MyFileDAO(); ⑩
        dao.insertFile(dto);
    }
}
```

⑦ request 내장객체를 통해 서버로 전송된 품값을 받을 수 있음

⑧ 파일 외 품값을 받아 변수에 저장(아래 부분 생략)

⑨ 품값을 DTO에 저장

⑩ DAO객체 생성 및 insert를 위한 메서드 호출

■ 파일 업로드 및 품값 처리

- 파일 저장을 위한 Uploads 폴더의 물리적 경로가 필요한 이유?
 - 운영체제에 따라 드라이브나 경로를 표시하는 방식이 다르기 때문
- 파일의 확장자를 추출하기 위해 `lastIndexOf()` 메서드를 사용한 이유?
 - 파일명에 다음과 같이 점이 2개 이상 포함될 수 있기 때문
- 파일 업로드시 파일명을 변경하는 이유?
 - 파일명이 한글인 경우 인코딩 방식에 따라 한글 깨짐 현상이 발생할 수 있기 때문
 - 영문과 숫자의 조합은 절대 깨짐현상이 발생하지 않음
 - 파일명이 중복되는 것을 방지하기 위함
 - 예를 들어 “가을하늘.jpg”는 누구나 사용할 수 있는 파일명임

파일명.indexOf(".")

2021년.가을하늘.jpg

파일명.lastIndexOf(".")

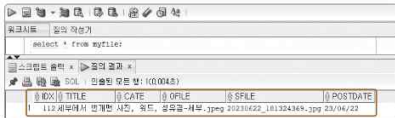
동작 확인

파일 업로드

제목 : 세부에서 변경

카테고리(선택사항) : ☒ 사진 ☐ 과제 ☒ 워드 ☐ 음원

첨부파일 : 성유경-세부.jpeg



업로드 된 파일은 이클립스에서는 확인할 수 없고 윈도우 탐색기에서 확인 가능

- C:\02Workspaces\metadata\plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\MustHaveJSP\Uploads



■ 파일 목록 출력하기

- myfile 테이블에 저장된 레코드를 기반으로 목록 출력

// 파일 목록을 반환합니다.

```
public List<MyFileDTO> myFileList() {  
    List<MyFileDTO> fileList = new Vector<MyFileDTO>(); ①  
    // 쿼리문 작성  
    String query = "SELECT * FROM myfile ORDER BY idx DESC"; ②  
    try {  
        stmt = con.createStatement(); // statement 객체 생성 ③  
        rs = stmt.executeQuery(query); // 쿼리 실행 ④  
  
        while (rs.next()) { // 목록 안의 파일 수만큼 반복 ⑤  
            // DTO에 저장  
            MyFileDTO dto = new MyFileDTO(); ⑥  
            dto.setIdx(rs.getString(1));  
            dto.setTitle(rs.getString(2));  
            dto.setPostdate(rs.getString(6));  
  
            fileList.add(dto); // 목록에 추가 ⑦  
        }  
    }  
}
```

MyFileDAO.java

① List 컬렉션 생성

② myfile 테이블의 레코드를 내림차순 정렬

③ 쿼리실행

⑤ 레코드수 만큼 반복해서 List 에 추가

■ 파일 목록 출력하기(계속)

```

<%
MyFileDAO dao = new MyFileDAO(); ❷
List<MyFileDTO> fileLists = dao.myFileList(); ❸
dao.close();
%>
<table border="1">
  <tr>
    <th>No</th><th>작성자</th><th>제목</th><th>카테고리</th>
    <th>원본 파일명</th><th>저장된 파일명</th><th>작성일</th><th></th>
  </tr>
  <% for (MyFileDTO f : fileLists) { %> ❹
    <tr>
      <td><%= f.getIdx() %></td>
      <td><%= f.getTitle() %></td>
      <td><%= f.getCate() %></td>
      <td><%= f.getOfile() %></td>
      <td><%= f.getSfile() %></td>
      <td><%= f.getPostdate() %></td>
      <td><a href="Download.jsp?oName=<%= URLEncoder.encode(f.getOfile(),
"UTF-8") %>&sName=<%= URLEncoder.encode(f.getSfile(),"UTF-8") %>">[다운로드]
</a></td> ❺
    </tr>
  <% } %>
</table>

```

FileList.jsp

❷ 레코드 인출을 위한 DAO의 메서드 호출

❹ 레코드수 만큼 반복해서 <tr>태그로 출력

❺ 다운로드 링크. 원본파일명과 저장된파일명을 매개변수로 전달 및 한글 깨짐 처리

DB에 등록된 파일 목록 보기

새 파일 등록용 링크

다운로드 링크

파일등록1 파일등록2

No	제목	카테고리	원본 파일명	저장된 파일명	작성일	
22	세부에서 번개맨	사진, 워드	성유경-세부.jpeg	20230622_181324369.jpeg	2023-06-22 18:13:25	[다운로드]

첨부파일을 서버에 저장할때는 날짜와 시간을 조합한 파일명으로 변경
따라서 파일을 다운로드 할때는 원본파일명으로 변경한 후 로컬에 저장해야 함

Download.jsp

```
<%  
String saveDirectory = application.getRealPath("/Uploads"); ❶  
String saveFilename = request.getParameter("sName");  
String originalFilename = request.getParameter("oName"); ❷  
  
try {  
    // 파일을 찾아 입력 스트림 생성 ❸  
    File file = new File(saveDirectory, saveFilename);  
    InputStream inStream = new FileInputStream(file);  
    String client = request.getHeader("User-Agent"); ❹  
    if (client.indexOf("WOW64") == -1) { ❺  
        originalFilename = new String(originalFilename.getBytes("UTF-8"),  
                                       "ISO-8859-1");  
    }  
    else {  
        originalFilename = new String(originalFilename.getBytes("KSC5601"),  
                                       "ISO-8859-1");  
    }  
}
```

❶ 폴더의 물리적경로와 매개변수로 전달된 파일명 받음

❸ 서버에 저장된 파일을 대상으로 입력 스트림 생성

❹ 웹 브라우저가 익스플로러인 경우 별도의 인코딩(KSC5601) 처리
나머지는 UTF-8로 인코딩

JSP 13.3 파일 다운로드

```
response.reset(); ❸  
response.setContentType("application/octet-stream"); ❷  
response.setHeader("Content-Disposition",  
    "attachment; filename=\"" + originalFilename + "\""); ❸  
response.setHeader("Content-Length", "" + file.length() );
```

```
// response 내장 객체로부터 새로운 출력 스트림 생성 ❶  
OutputStream outStream = response.getOutputStream();
```

```
// 출력 스트림에 파일 내용 출력 ❶  
byte b[] = new byte[(int)file.length()];  
int readBuffer = 0;  
while ( (readBuffer = inStream.read(b)) > 0 ) {  
    outStream.write(b, 0, readBuffer);  
}
```

```
// 입/출력 스트림 닫음 ❷  
inStream.close();  
outStream.close();
```

```
}  
catch (FileNotFoundException e) {  
    JSFunction.alertBack("파일을 찾을 수 없습니다.", out);  
}
```

Download.jsp(계속)

- ❶ 응답헤더 초기화
- ❷ 콘텐츠 타입을 octet-stream 으로 설정하면 파일 종류에 상관없이 무조건 다운로드 창을 띄우게 됨
- ❸ 응답 헤더 설정을 통해 원본파일명으로 변경되게 됨
- ❶ 새로운 출력 스트림을 생성해서 파일을 로컬로 저장

No	제목	카테고리	원본 파일명	저장된 파일명	작성일	
22	세부에서 번개맨 사진, 워드		성유경-세부.jpeg	20230622_181324369.jpeg	2023-06-22 18:13:25	다운로드

한꺼번에 2개 이상의 파일업로드 하기 위해서는 다음과 같은 등록폼이 필요

파일 업로드

제목 :

카테고리(선택사항) : ☒ 사진 ☐ 과제 ☐ 워드 ☐ 음원

첨부파일1 :

첨부파일2 :

...중간생략...

첨부파일10 :

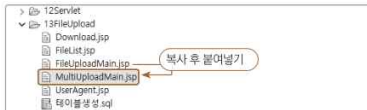
단점

- 파일을 개별적으로 선택해야 하므로 매우 불편
- `<input>` 태그의 name속성도 ofile1, ofile2와 같이 별도의 이름을 부여해야 함

우리는 **multiple** 속성을 사용

■ 작성품

- [예제 13-1] FileUploadMain.jsp 를 복사한 후 파일명을 MultiUploadMain.jsp로 변경



■ 작성품(계속)

예제 13-10] webapp/13FileUpload/MultiFileUploadMain.jsp

```

<body>
  <h3>멀티 파일 업로드(multiple 속성 추가)</h3> ❶
  <span style="color: red;">${errorMessage }</span>
  <form name="fileForm" method="post" enctype="multipart/form-data"
    action="MultipleProcess.do" onsubmit="return validateForm(this);"> ❷
    제목 : <input type="text" name="title" /><br />
    카테고리(선택사항) :
      <input type="checkbox" name="cate" value="사진" checked />사진
      <input type="checkbox" name="cate" value="과제" />과제
      <input type="checkbox" name="cate" value="워드" />워드
      <input type="checkbox" name="cate" value="음원" />음원 <br />
    첨부 파일(다중선택가능) : <input type="file" name="attachedFile" multiple /> ❸
  <br />
  <input type="submit" value="전송하기" />
</form>
</body>

```

기존 업로드 폼에서 표시한 부분만 수정

❸ 파일 다중 선택을 위해 multiple 속성 추가. ctrl 이나 드레그로 여러 파일 선택 가능

■ 파일 업로드 및 품값 처리

예제 13-11] Java Resources/fileupload/FileUtil.java

```
public static ArrayList<String> multipleFile(HttpServletRequest req, String
sDirectory)
    throws ServletException, IOException {

    ArrayList<String> listFileName = new ArrayList<>(); ❶
    Collection<Part> parts = req.getParts(); ❷
    for(Part part : parts) {
        if(!part.getName().equals("ofile")) ❸
            continue;

        String partHeader = part.getHeader("content-disposition"); ❹
        String[] phArr = partHeader.split("filename=");
        String originalFileName = phArr[1].trim().replace("\\", "");
        if (!originalFileName.isEmpty()) {
            part.write(sDirectory+ File.separator +originalFileName);
        }
        listFileName.add(originalFileName); ❺
    }
    return listFileName; ❻
}
```

upload() 메서드와 거의 유사함

❶ 여러개의 파일을 첨부하므로 파일명 저장을 위한 List 컬렉션 생성

❷ getParts()로 전송된 전체 품값을 받음

❸ 품값 중 ofile이 아니면 저장 대상이 아님

❹ 헤더값으로 파일명 추출 및 파일저장

❺ List 컬렉션에 원본 파일명 추가

■ 파일 업로드 및 품값 처리(계속)

- UploadProcess.java를 복사한 후 MultipleProcess.java로 파일명을 변경

예제 13-12] Java Resources/fileupload/MultipleProcess.java

```
try {
    String saveDirectory = getServletContext().getRealPath("/Uploads");
    ArrayList<String> listFileName = FileUtil.multipleFile(req,
        saveDirectory); ①
    for(String originalFileName : listFileName) { ②
        String savedFileName = FileUtil.renameFile(saveDirectory,
            originalFileName);
        insertMyFile(req, originalFileName, savedFileName);
    }
    resp.sendRedirect("FileList.jsp"); ③
}
catch (Exception e) { ④
```

```
private void insertMyFile(HttpServletRequest req, String oFileName, String
sFileName) { ⑤
    ... 생략 ...
}
```



① multipleFile() 메서드를 호출하여 파일 업로드

② 첨부파일의 갯수만큼 반복해서 파일명 변경 및 insert 처리

⑤ 이 부분은 변경 없음

동작확인

파일 업로드(multiple속성추가)

제목 : 2개 이상 선택하세요

카테고리(선택사항) : ☒ 사진 ☐ 과제 ☐ 워드 ☐ 음원

첨부파일 : 파일 선택 선택된 파일 없음

전송하기

파일 업로드(multiple속성추가)

제목 : 2개 이상 선택하세요

카테고리(선택사항) : ☒ 사진 ☐ 과제 ☐ 워드 ☐ 음원

첨부파일 : 파일 선택 파일 5개

전송하기

Ctrl 키를 누른 상태로 파일선택



선택한 파일의 갯수가 표시됨

동작확인(계속)

DB에 등록된 파일 목록 보기

파일등록1 파일등록2

2개 이상 선택하세요

No	제목	카테고리	원본 파일명	저장된 파일명	작성일	
28	2개 이상 선택하세요	사진	성유경-둘.jpg	20230623_15536657.jpg	2023-06-23 15:53:06	[다운로드]
27	2개 이상 선택하세요	사진	성유경-기타2.jpg	20230623_15536653.jpg	2023-06-23 15:53:06	[다운로드]
26	2개 이상 선택하세요	사진	성유경-기타1.jpg	20230623_15536646.jpg	2023-06-23 15:53:06	[다운로드]
25	2개 이상 선택하세요	사진	성유경-원투.jpg	20230623_15536642.jpg	2023-06-23 15:53:06	[다운로드]
24	2개 이상 선택하세요	사진	성유경-5살.jpg	20230623_15535503.jpg	2023-06-23 15:53:06	[다운로드]
22	세부에서 번개맨	사진, 워드	성유경-세부.jpeg	20230622_181324369.jpeg	2023-06-22 18:13:25	[다운로드]

업로드 완료시 선택한 5개의 파일
이 등록됨

gins@org.eclipse.wst.server.core#tmp0#wtpwebapps#MustHave/SPWUploads

Uploads 검색









20230622_181324369.jpeg
20230623_15535503.jpg
20230623_15536642.jpg
20230623_15536646.jpg
20230623_15536653.jpg
20230623_15536657.jpg
업로드폴더

Upload 폴더에서도 첨부된 파일
확인

■ 핵심요약

- 내용 입력을 위한 <form> 태그를 작성할 때에는 반드시 다음 2가지 속성을 지정해야 함
 - method = "post"
 - enctype = "multipart/form-data"
- 폼값을 받을 때는 type이 file인 항목과 아닌 것을 구분해야 합니다.
 - type='file'인 경우 : getPart() 메서드를 통해 Parts 객체로 받음
 - 그외의 경우 : getParameter() 메서드를 통해 String 객체로 받음
- 파일명이 한글인 경우 웹 브라우저에 출력할 때 문제가 될 수 있으므로 파일명은 영문과 숫자의 조합으로 변경하는 것이 안전
- 파일 다운로드 시에는 응답 헤더 설정을 통해 원본 파일명으로 변경한 후 저장

