

2025년 상반기 K-디지털 트레이닝

File Upload

[KB] IT's Your Life

✓ File upload

- 첨부파일을 위한 tbl_board_attachment 테이블 생성
- board의 bno에 대한 FK 설정

```
DROP TABLE IF EXISTS tbl_board_attachment;
```

```
CREATE TABLE tbl_board_attachment (  
  no INTEGER AUTO_INCREMENT PRIMARY KEY,  
  filename VARCHAR(256) NOT NULL,      -- 원본 파일 명  
  path VARCHAR(256) NOT NULL,          -- 서버에서의 파일 경로  
  content_type VARCHAR(56),            -- content-type  
  size INTEGER,                        -- 파일의 크기  
  bno INTEGER NOT NULL,                -- 게시글 번호, FK  
  reg_date DATETIME DEFAULT now(),  
  CONSTRAINT FOREIGN KEY(bno) REFERENCES tbl_board(no)  
);
```

✓ 주의 사항

- 업로드 된 파일을 저장할 때 파일 명 중복 발생 가능
- 유일한 파일명이 되도록 해야함.
- 방법
 - UUID를 사용
 - 기존 파일명에 timestamp를 추가
- 다운로드를 대비해서 원본 파일명을 저장 해야함

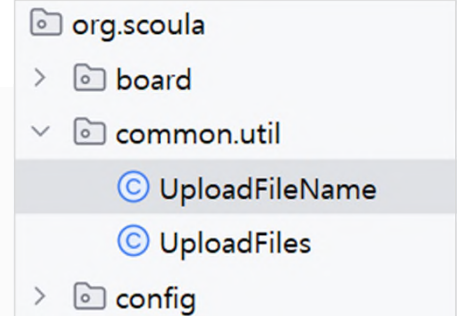
UploadFileName.java

```
package org.scoula.common.util;

public class UploadFileName {
    public static String getUniqueName(String filename) {

        int ix = filename.lastIndexOf(".");
        String name = filename.substring(0, ix);           // 파일명 추출
        String ext = filename.substring(ix+1);            // 확장명 추출

        return String.format("%s-%d.%s", name, System.currentTimeMillis(), ext);
    }
}
```



UploadFiles.java

```
package org.scoula.common.util;

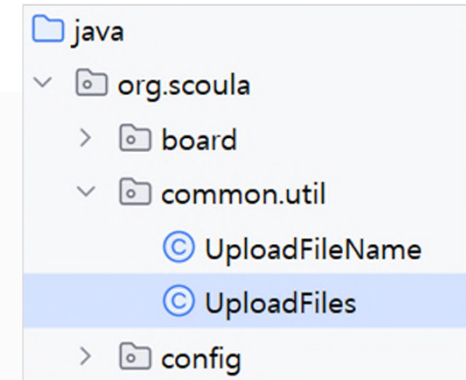
import org.springframework.web.multipart.MultipartFile;

import java.io.File;
import java.io.IOException;

public class UploadFiles {
    public static String upload(String baseDir, MultipartFile part) throws IOException {

        // 기본 디렉토리가 있는지 확인, 없으면 새로 생성
        File base = new File(baseDir);
        if(!base.exists()) {
            base.mkdirs();                // 중간에 존재하지 않는 디렉토리까지 모두 생성
        }

        String fileName = part.getOriginalFilename();
        File dest = new File(baseDir, UploadFileName.getUniqueName(fileName));
        part.transferTo(dest);            // 지정한 경로로 업로드 파일 이동
        return dest.getPath();           // 저장된 파일 경로 리턴
    }
}
```



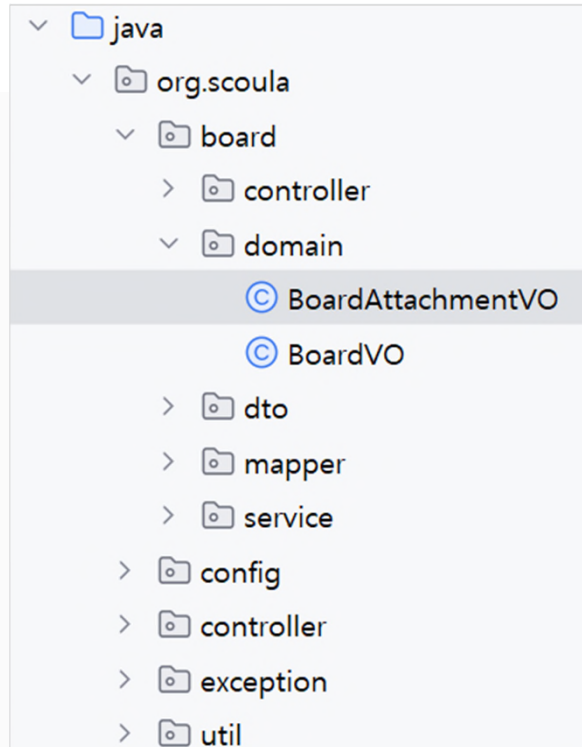
BoardAttachmentVO.java

```
package org.scoula.board.domain;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import org.springframework.web.multipart.MultipartFile;

import java.util.Date;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
public class BoardAttachmentVO {
    private Long no;        //
    private Long bno;       // FK: Board의 no
    private String filename; // 원본 파일명
    private String path;     // 서버에 저장된 파일 경로
    private String contentType; // 파일 mime-type
    private Long size;       // 파일의 크기
    private Date regDate;    // 등록일
```



BoardAttachmentVO.java

// 팩토리 메서드

public static BoardAttachmentVO of(MultipartFile part, Long bno, String path) { // path: 업로드된 파일 경로

```
    return builder()
        .bno(bno)
        .filename(part.getOriginalFilename())
        .path(path)
        .contentType(part.getContentType())
        .size(part.getSize())
        .build();
```

```
}
```

```
}
```

BoardMapper.java

```
package org.scoula.board.mapper;
...

public interface BoardMapper {
    ...

    public void createAttachment(BoardAttachmentVO attach);

    public List<BoardAttachmentVO> getAttachmentList(Long bno);

    public BoardAttachmentVO getAttachment(Long no);

    public int deleteAttachment(Long no);
}
```


resources::BoardMapper.xml

```
<insert id="createAttachment">
    insert into tbl_board_attachment(filename, path, content_type, size, bno)
    values(#{filename}, #{path}, #{contentType}, #{size}, #{bno})
</insert>

<select id="getAttachmentList" resultType="org.scoula.board.domain.BoardAttachmentVO">
    select * from tbl_board_attachment
    where bno = #{bno}
    order by filename
</select>

<select id="getAttachment" resultType="org.scoula.board.domain.BoardAttachmentVO">
    select * from tbl_board_attachment
    where no = #{no}
</select>

<delete id="deleteAttachment">
    delete from tbl_board_attachment
    where no = #{no}
</delete>
```

✓ BoardVO에 BoardAttachmentVO 연결

○ BoardVO.java

```
@Data
public class BoardVO {
    ...

    private List<BoardAttachmentVO> attaches;

    private Date regDate;
    private Date updateDate;
}
```

- 1) join 처리 → resultMap 구성
- 2) setAttaches() 호출로 처리

BoardDTO.java

```
...
public class BoardDTO {
    private Long no;
    private String title;
    private String content;
    private String writer;
    private Date regDate;
    private Date updateDate;

    // 첨부 파일
    private List<BoardAttachmentVO> attaches;

    List<MultipartFile> files = new ArrayList<>(); // 실제 업로드된 파일(Multipart) 목록
```

BoardDTO.java

```
// VO --> DTO 변환
public static BoardDTO of(BoardVO vo) {
    return BoardDTO.builder()
        ...
        .attaches(vo.getAttaches())
        .regDate(vo.getRegDate())
        .updateDate(vo.getUpdateDate())
        .build();
}

// DTO --> VO 변환
public BoardVO toVo() {
    return BoardVO.builder()
        ...
        .attaches(attaches)
        .regDate(regDate)
        .updateDate(updateDate)
        .build();
}
}
```

org/scoula/board/mapper/BoardMapper.xml

join

```
<resultMap id="attachmentMap" type="org.scoula.board.domain.BoardAttachmentVO">
  <id column="ano" property="no"/>
  <result column="bno" property="bno"/>
  <result column="filename" property="filename"/>
  <result column="path" property="path"/>
  <result column="contentType" property="contentType"/>
  <result column="size" property="size"/>
  <result column="a_reg_date" property="regDate"/>
</resultMap>
```

```
<resultMap id="boardMap" type="org.scoula.board.domain.BoardVO">
  <id column="no" property="no"/>
  <result column="title" property="title"/>
  <result column="content" property="content"/>
  <result column="writer" property="writer"/>
  <result column="reg_date" property="regDate"/>
  <result column="update_date" property="updateDate"/>
  <collection property="attaches" resultMap="attachmentMap"/>
</resultMap>
```

```
<select id="get" resultMap="boardMap">
  select b.*, a.no as ano, a.bno, a.filename, a.path,
    a.content_type, a.size, a.reg_date as a_reg_date
  from tbl_board b
    left outer join tbl_board_attachment a
      on b.no = a.bno
  where b.no = #{no}
  order by filename
</select>
```

BoardService.java

```
public interface BoardService {  
    ...  
  
    public BoardAttachmentVO getAttachment(Long no);  
  
    public boolean deleteAttachment(Long no);  
}
```

BoardServiceImpl.java

```
public BoardServiceImpl implements BoardService {  
    private final static String BASE_DIR = "c:/upload/board";  
    ...  
  
    // 2개 이상의 insert 문이 실행될 수 있으므로 트랜잭션 처리 필요  
    // RuntimeException인 경우만 자동 rollback.  
    @Transactional  
    @Override  
    public void create(BoardDTO board) {  
        log.info("create....." + board);  
  
        BoardVO boardVO = board.toVo();  
        mapper.create(boardVO);  
  
        // 파일 업로드 처리  
        List<MultipartFile> files = board.GetFiles();  
        if(files != null && !files.isEmpty()) {  
            upload(boardVO.getNo(), files);  
        }  
    }  
}
```

BoardServiceImpl.java

```
private void upload(Long bno, List<MultipartFile> files) {  
    for(MultipartFile part: files) {  
        if(part.isEmpty()) continue;  
        try {  
            String uploadPath = UploadFiles.upload(BASE_DIR, part);  
            BoardAttachmentVO attach = BoardAttachmentVO.of(part, bno, uploadPath);  
            mapper.createAttachment(attach);  
        } catch (IOException e) {  
            throw new RuntimeException(e); // @Transactional에서 감지, 자동 rollback  
        }  
    }  
}
```


BoardServiceImpl.java

```
...  
  
// 첨부파일 한 개 얻기  
@Override  
    public BoardAttachmentVO getAttachment(Long no) {  
        return mapper.getAttachment(no);  
    }  
  
// 첨부파일 삭제  
@Override  
    public boolean deleteAttachment(Long no) {  
        return mapper.deleteAttachment(no) == 1;  
    }  
}
```

✓ 파일 업로드

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

views/board/create.jsp

중요!!!!

```
...
<div >
  <form method="post" enctype="multipart/form-data">
    <div>
      <label>제목</label>
      <input name="title" class="form-control">
    </div>

    <div>
      <label>작성자</label>
      <input name="writer" class="form-control">
    </div>

    <div>
      <label>첨부파일</label>
      <input type="file" class="form-control-file border" multiple name="files"/>
    </div>

    <div>
      <label>내용</label>
      <textarea class="form-control" name="content" rows="10"></textarea>
    </div>
  </form>
</div>
...
```

2025년 상반기 K-디지털 트레이닝

File Download

[KB] IT's Your Life

- ✓ 상세 보기에서 첨부파일 목록 보여주기
 - 다운로드 링크 전시

📁 파일 업로드 테스트

👤 admin

🕒 2023-12-07

📄 백엔드.mm (631 Bytes)

📄 컨트롤러에서 직접 인증하기.pdf (1.2 MB)

파일 업로드 테스트

☰ 목록

✎ 수정

🗑 삭제

✓ 첨부파일 목록 보여주기

```
@Data
public class BoardDTO {
    ...

    private List<BoardAttachmentVO> attaches;
    ...
}
```

📄 파일 업로드 테스트

👤 admin

🕒 2023-12-07

📄 백엔드.mm (631 Bytes)

📄 컨트롤러에서 직접 인증하기.pdf (1.2 MB)

파일 업로드 테스트

☰ 목록

✎ 수정

🗑 삭제

✓ Java로 파일 크기 포맷 출력 메서드

○ 1225957 → 1.225MB

```
public class UploadFiles {  
    ...  
  
    public static String getFormatSize(Long size) {  
        if (size <= 0)  
            return "0";  
        final String[] units = new String[] { "Bytes", "KB", "MB", "GB", "TB" };  
        int digitGroups = (int) (Math.log10(size) / Math.log10(1024));  
        return new DecimalFormat("#,##0.#").format(size / Math.pow(1024, digitGroups)) + " " + units[digitGroups];  
    }  
}
```

BoardAttachmentVO.java

```
public class BoardAttachmentVO {
    private Long no;        // PK
    private Long bno;        // FK: Board.no
    private String filename; // 원본 파일명
    private String path;     // 서버에 저장된 파일 경로
    private String contentType; // 파일 mime-type
    private Long size;       // 파일의 크기
    private Date regDate;    // 등록일

    public static BoardAttachmentVO of(MultipartFile part, Long bno, String path) {
        return builder()
            .bno(bno)
            .filename(part.getOriginalFilename())
            .path(path)
            .contentType(part.getContentType())
            .size(part.getSize())
            .build();
    }

    public String getFileSize() {
        return UploadFiles.getFormatSize(size);
    }
}
```


views/board/get.jsp

```
...
<h1 class="page-header my-4"><i class="far fa-file-alt"></i> ${board.title}</h1>

<div class="d-flex justify-content-between">
  <div><i class="fas fa-user"></i> ${board.writer}</div>
  <div>
    <i class="fas fa-clock"></i>
    <fmt:formatDate pattern="yyyy-MM-dd" value="${board.regDate}"/>
  </div>
</div>

<div class="text-end">
  <c:forEach var="file" items="${board.attaches}">
    <div class="attach-file-item">
      <a href="/board/download/${file.no}" class="file-link">
        <i class="fa-solid fa-floppy-disk"></i>
        ${file.filename} (${file.fileSize})<br>
      </a>
    </div>
  </c:forEach>
</div>

<hr>
...
```

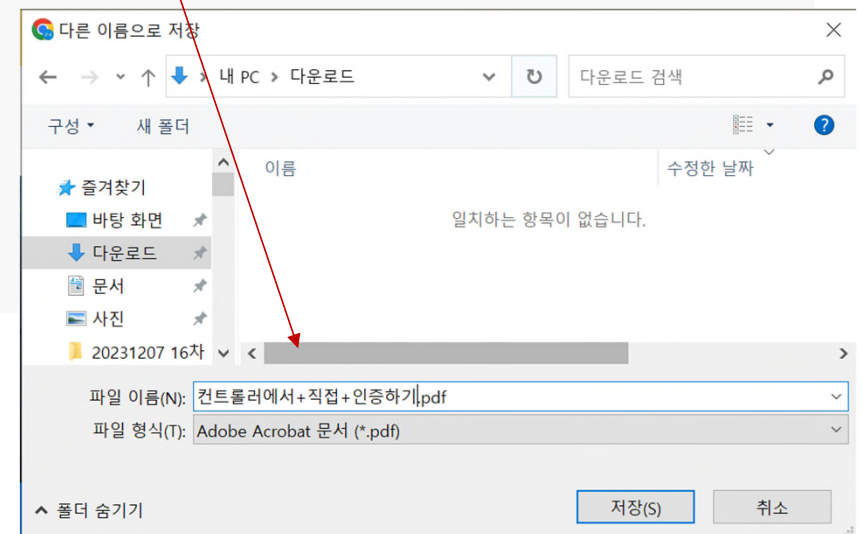
파일 다운로드 링크

✓ File download

- c:/upload/board 경로는 웹 url로 표현할 수 없음
 - 직접 링크를 걸 수 없음
- Controller에서 download 요청을 받았을 때 직접 파일을 읽어서 처리
 - url: /board/download/{no}
 - 경로변수: url상에 있는 변수
 - @PathVariable("no")로 추출
- 다운로드
 - 응답 헤더로 파일 첨부 정보를 설정
 - HttpServletResponse에 binary stream을 얻어 직접 write

UploadFiles.java

```
public class UploadFiles {  
    ...  
  
    public static void download(HttpServletResponse response, File file, String orgName) throws Exception {  
  
        response.setContentType("application/download");  
        response.setContentLength((int)file.length());  
  
        String filename = URLEncoder.encode(orgName, "UTF-8"); // 한글 파일명인 경우 인코딩 필수  
        response.setHeader("Content-disposition", "attachment;filename=\"" + filename + "\"");  
  
        try(OutputStream os = response.getOutputStream();  
            BufferedOutputStream bos = new BufferedOutputStream(os)) {  
            Files.copy(Paths.get(file.getPath()), bos);  
        }  
    }  
}
```



BoardController.java

```
@GetMapping("/download/{no}")
@ResponseBody // view를 사용하지 않고, 직접 내보냄
public void download(@PathVariable("no") Long no, HttpServletResponse response) throws Exception {

    BoardAttachmentVO attach = service.getAttachment(no);

    File file = new File(attach.getPath());

    UploadFiles.download(response, file, attach.getFilename());
}
```