

2025년 상반기 K-디지털 트레이닝

게시판 - 백엔드

[KB] IT's Your Life

✓ 다음 경로에 대한 접근 권한을 설정 하세요.

- /api/board

- POST, PUT, DELETE 요청 : 인증 요구

SecurityConfig.java

```
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
    ...  
    @Override  
    public void configure(HttpSecurity http) throws Exception {  
        ...  
  
        http  
            .authorizeRequests()  
                ...  
                .antMatchers(HttpMethod.POST, "/api/board/**").authenticated()  
                .antMatchers(HttpMethod.PUT, "/api/board/**").authenticated()  
                .antMatchers(HttpMethod.DELETE, "/api/board/**").authenticated()  
                .anyRequest().permitAll();  
  
        http.httpBasic().disable()    // 기본 HTTP 인증 비활성화  
            .csrf().disable()        // CSRF 비활성화  
            .formLogin().disable()    // formLogin 비활성화    관련 필터 해제  
            .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS); // 세션 생성 모드 설정  
    }  
    ...  
}
```

✔ 첨부 파일 테이블에 다음 제약조건을 가지도록 다시 정의하세요.

○ 게시글 삭제 시 연관 첨부 파일 자동 삭제

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

```
CREATE TABLE tbl_board_attachment
```

```
(
```

```
    no            INTEGER AUTO_INCREMENT PRIMARY KEY, -- PK
```

```
    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) ON DELETE CASCADE
```

```
);
```

- ✓ 게시글을 추출할 때 첨부 파일 목록이 추가되도록 정의하세요.

BoardMapper.java

...

```
public interface BoardMapper {  
    List<BoardVO> getList();  
    BoardVO get(Long no);  
    void create(BoardVO board);  
    int update(BoardVO board);  
    int delete(Long no);  
  
    void createAttachment(BoardAttachmentVO attach);  
    List<BoardAttachmentVO> getAttachmentList(Long bno);  
    BoardAttachmentVO getAttachment(Long no);  
    int deleteAttachment(Long no);  
}
```

BoardMapper.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-
mapper.dtd">

<mapper namespace="org.scoula.board.mapper.BoardMapper">
  <select id="getList" resultType="BoardVO">
    <![CDATA[
      select * from tbl_board order by no desc
    ]]>
  </select>
```

BoardMapper.xml

```
<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>
```


BoardMapper.xml

```
<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
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>

<insert id="create">
    insert into tbl_board (title, content, writer)
    values (#{title}, #{content}, #{writer})
    <selectKey resultType="Long" keyProperty="no" keyColumn="no" order="AFTER">
        SELECT LAST_INSERT_ID()
    </selectKey>
</insert>
```

BoardMapper.xml

```
<update id="update">
  update tbl_board set
    title = #{title},
    content = #{content},
    writer = #{writer},
    update_date = now()
  where no = #{no}
</update>

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

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="BoardAttachmentVO">
    select * from tbl_board_attachment where bno = #{bno}
    order by filename
</select>

<select id="getAttachment" resultType="BoardAttachmentVO">
    select * from tbl_board_attachment
    where no = #{no}
</select>

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

✓ **게시판 서비스에 다음 기능을 추가하세요.**

- 추가, 수정 시 첨부파일 업로드 처리
- 첨부파일 하나 추출 기능
- 첨부파일 하나 삭제 기능

BoardService.java

```
package org.scoula.board.service;
...

public interface BoardService {
    List<BoardDTO> getList();

    BoardDTO get(Long no);

    BoardDTO create(BoardDTO board);

    BoardDTO update(BoardDTO board);

    BoardDTO delete(Long no);

    BoardAttachmentVO getAttachment(Long no);

    boolean deleteAttachment(Long no);
}
```

BoardServiceImpl.java

```
package org.scoula.board.service;
...

@Log4j2
@Service
@RequiredArgsConstructor
public class BoardServiceImpl implements BoardService {
    private final static String BASE_DIR = "c:/upload/board";

    private final BoardMapper mapper;

    @Override
    public List<BoardDTO> getList() {
        log.info("getList.....");
        return mapper.getList().stream()
            .map(BoardDTO::of)
            .toList();
    }
}
```

BoardServiceImpl.java

```
@Override
public BoardDTO get(Long no) {
    log.info("get....." + no);
    BoardDTO board = BoardDTO.of(mapper.get(no));

    log.info("===== " + board);
    return Optional.ofNullable(board)
        .orElseThrow(NoSuchElementException::new);
}
```

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.from(part, bno, uploadPath);
            mapper.createAttachment(attach);
        } catch (IOException e) {
            throw new RuntimeException(e);
            //log.error(e.getMessage());
        }
    }
}
```


BoardServiceImpl.java

```
@Transactional // 2개 이상의 insert 문이 실행될 수 있으므로 트랜잭션 처리 필요
@Override
public BoardDTO 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);
    }
    return get(boardVO.getNo());
}
```

BoardServiceImpl.java

```
@Override
public BoardDTO update(BoardDTO board) {
    log.info("update..... " + board);
    BoardVO boardVO = board.toVo();
    log.info("update..... " + boardVO);

    mapper.update(boardVO);

    // 파일 업로드 처리
    List<MultipartFile> files = board.GetFiles();
    if(files != null && !files.isEmpty()) {
        upload(board.getNo(), files);
    }

    return get(board.getNo());
}
```

BoardServiceImpl.java

```
@Override
public BoardDTO delete(Long no) {
    log.info("delete...." + no);
    BoardDTO board = get(no);

    mapper.delete(no);
    return board;
}

@Override
public BoardAttachmentVO getAttachment(Long no) {
    return mapper.getAttachment(no);
}

@Override
public boolean deleteAttachment(Long no) {
    return mapper.deleteAttachment(no) == 1;
}

}
```

✔ 다음과 같이 BoardController를 정의하세요.

- RestController 구성
- 첨부파일 다운로드 및 삭제 처리

HTTP 메서드	URL	메서드명	리턴
GET		getList()	ResponseEntity<List<BoardDTO>>
GET	/no	get()	ResponseEntity<BoardDTO>
POST		create()	ResponseEntity<BoardDTO>
PUT	/no	update()	ResponseEntity<BoardDTO>
DELETE	/no	delete()	ResponseEntity<BoardDTO>
GET	/download/no	download	void
DELETE	/deleteAttachment/no	deleteAttachment	ResponseEntity<Boolean>

BoardController.java (Swagger 코드 생략)

```
package org.scoula.board.controller;
..
@RestController
@RequestMapping("/api/board")
@RequiredArgsConstructor
@Log4j2
public class BoardController {

    private final BoardService service;

    @GetMapping("")
    public ResponseEntity<List<BoardDTO>> getList() {
        return ResponseEntity.ok(service.getList());
    }

    @GetMapping("/{no}")
    public ResponseEntity<BoardDTO> getById(@PathVariable Long no) {
        return ResponseEntity.ok(service.get(no));
    }
}
```

BoardController.java (Swagger 코드 생략)

```
@PostMapping("")
public ResponseEntity<BoardDTO> create(BoardDTO board) {
    return ResponseEntity.ok(service.create(board));
}

@PutMapping("/{no}")
public ResponseEntity<BoardDTO> update(@PathVariable Long no, BoardDTO board) {
    return ResponseEntity.ok(service.update(board));
}

@DeleteMapping("/{no}")
public ResponseEntity<BoardDTO> delete(@PathVariable Long no) {
    return ResponseEntity.ok(service.delete(no));
}
```

****주의****
Multipart 인코딩이므로
@RequestBody 붙이지 않음

BoardController.java (Swagger 코드 생략)

```
@GetMapping("/download/{no}")
public void download(@PathVariable Long no, HttpServletResponse response) throws Exception {
    BoardAttachmentVO attachment = service.getAttachment(no);
    File file = new File(attachment.getPath());
    UploadFiles.download(response, file, attachment.getFilename());
}

@GetMapping("/deleteAttachment/{no}")
public ResponseEntity<Boolean> deleteAttachment(@PathVariable Long no) throws Exception {
    return ResponseEntity.ok(service.deleteAttachment(no));
}
}
```