

2025년 상반기 K-디지털 트레이닝

# 비즈니스 계층 (심화1)

---

[KB] IT's Your Life

- ✓ 다음과 같은 조건을 가지는 BoardDTO 클래스를 정의하세요
  - 패키지: org.scoula.board.dto
  - BoardVO에 대한 DTO 객체
  - BoardVO로부터 DTO 객체를 생성하는 팩토리 메서드 of() 정의
  - BoardVO로 변환하는 toVo() 메서드

- ✓ RootConfig의 컴포넌트 스캔 패키지에 `org.scoula.board.service`를 추가하세요.

- ✓ Board에 대한 CRUD를 수행하는 메서드를 가지는 BoardService 인터페이스를 정의하세요.
  - 패키지: org.scoula.board.service

- ✓ **BoardService 인터페이스의 구현체 BoardServiceImpl 클래스를 정의하세요.**
  - 멤버변수 BoardMapper는 생성자 주입으로 받음
  - 실제 메서드 구현은 추후 진행 예정
  - 테스트 클래스 BoardServiceTest의 기본 골격 작성

- ✓ 게시글 목록을 추출하는 getList() 메서드를 정의하고, 단위 테스트를 진행하세요.

- ✓ 게시글 하나를 추출하는 `get(Long no)` 메서드를 정의하고, 단위 테스트를 진행하세요.
  - Optional을 이용하여 지정한 번호의 글이 없는 경우 `NoSuchElementException` 발생

- ✓ 게시글 하나를 생성하는 `create(BoardDTO)` 메서드를 정의하고, 단위 테스트를 진행하세요.



- ✓ 게시글 하나를 수정하는 `update(BoardDTO)` 메서드를 정의하고, 단위 테스트를 진행하세요.

- ✓ 게시글 하나를 삭제하는 delete(Long no) 메서드를 정의하고, 단위 테스트를 진행하세요.

2025년 상반기 K-디지털 트레이닝

# 프레젠테이션(웹) 계층의 CRUD 구현 (심화2)

---

[KB] IT's Your Life

- ✓ **BoardController 클래스의 기본 모양을 정의하세요.**
  - 패키지: org.scoula.board.controller
  - BoardService는 생성자 주입을 통해 설정
  - 공통 url: /board
  
- ✓ **ServletConfig에서 컴포넌트 스캔에 탐지되도록 설정하세요.**

- ✓ 컨트롤러 테스트 클래스 BoardControllerTest 클래스의 기본 구조를 정의하세요.
  - 매 테스트마다 MockMvc 객체가 초기화되도록 구성

✓ 목록 요청을 처리하는 `getList()` 메서드를 추가하고, 단위 테스트를 진행하세요.

○ GET 요청

- url: `/board/list`
- Model 구성
  - 속성명 : `list`
  - 속성값: service의 `getList()`로 목록
- View 이름: `board/list`

✓ 새 글 등록 요청을 처리하는 create() 메서드를 추가하고, 단위 테스트를 진행하세요.

- GET 요청

- url: /board/create
- view 이름: board/create

- POST 요청

- url: /board/create
- 파라미터: BoardDTO
- 처리: service.create()로 실제 등록
- view 이름 : redirect:/board/list

✓ 글 상세 보기 요청을 처리하는 get() 메서드를 추가하고, 단위 테스트를 진행하세요.

### ○ GET 요청

- url: /board/get
- 파라미터
  - no: 글 번호
- Model 구성
  - 속성명 : board
  - 속성값: service.get()으로 BoardDTO 획득
- View 이름: board/get

### ○ 수정의 GET 요청 처리와 동일

- url: /board/update



✓ 글 수정 요청을 처리하는 update() 메서드를 추가하고, 단위 테스트를 진행하세요.

- GET 요청

- url: /board/update
- BoardController의 get()에서 처리

- POST 요청

- url: /board/update
- 파라미터: BoardDTO
- 처리: service.update()로 실제 수정
- view 이름 : redirect:/board/list

✓ 삭제 요청을 처리하는 delete() 메서드를 추가하고, 단위 테스트를 진행하세요.

- GET 요청 없음
- POST 요청
  - url: /board/delete
  - 파라미터: no - 삭제할 글 번호
  - 처리: service.delete()로 실제 삭제
  - view 이름 : redirect:/board/list