

[문제 1] 게시판을 Restful API 형태로 설계할 때 아래와 같이 설계하려고 한다.
요구사항에 맞게 문제를 풀시오.

기능	HTTP 메소드	URL
전체글보기	GET	/bbs
글 상세보기	GET	/bbs/{articleId}
글쓰기	POST	/bbs
글수정	PUT	/bbs/{articleId}
글삭제	DELETE	/bbs/{articleId}

[요구사항]

1. 위의 API 설계를 기반으로 Spring Boot 의 Controller 클래스를 생성한다.
2. 각 메소드에는 @GetMapping, @PostMapping, @PutMapping, @DeleteMapping 어노테이션을 활용한다.
3. 메소드 이름은 기능을 명확히 드러내도록 작성한다.
4. 예시로 간단한 출력문(ex. System.out.println("글쓰기!");)을 넣어도 좋다.
5. Controller 클래스의 이름은 **BbsController** 로 작성한다.
6. 메소드의 매개변수와 반환타입은 상황에 맞게 지정한다. 글 상세보기의 경우 @PathVariable Long articleId 를 활용할 수 있다.
7. **패키지명**: com.sec01.controller

[문제 2] 다음 조건을 만족하는 JPA 실습 문제를 작성하시오.

[요구사항]

항목	내용
Entity 생성	Member 엔티티를 작성한다. - id (Long, PK, Auto Increment) - username (String, not null) - email (String) - createdAt (LocalDateTime, 자동 생성) - toString 메소드를 오버라이딩하여 객체의 상태를 출력한다.
Repository 작성	Member 엔티티를 기준으로 Repository 인터페이스 를 작성한다. - JpaRepository 를 상속받는다.
Service 작성	MemberService 클래스를 작성한다. - createMember(String username, String email) 메소드를 구현한다. - 서비스는 트랜잭션으로 관리되어야 한다.
DB 설정	내장형 H2 데이터베이스 를 사용하여 구현한다.
디렉토리 구조 예시	src/main/java/com/sec02/entity/Member.java src/main/java/com/sec02/repository/MemberRepository.java src/main/java/com/sec02/service/MemberService.java src/test/java/com/sec02/repository/MemberRepositoryTest.java
패키지명	com.sec02

MemberRepositoryTest.java 로 테스트를 실행한다.

_ 간단한 회원 저장/조회 테스트를 작성

```
1. package com.sec02.repository;
2. @DataJpaTest
3. class MemberRepositoryTest {

4.     @Autowired
5.     private MemberRepository memberRepository;

6.     @Test
7.     void testSaveAndFindById() {
8.         // given
9.         Member member = new Member();
10.        member.setUsername("testuser");
11.        member.setEmail("testuser@example.com");
12.        member.setCreatedDate(LocalDateTime.now());

13.        // when
14.        Member savedMember = memberRepository.save(member);
15.        Optional<Member> foundMember =
            memberRepository.findById(savedMember.getId());

16.        // then
17.        assertThat(foundMember).isPresent();
18.        assertThat(foundMember.get().getUsername()).isEqualTo("testuser");
19.        assertThat(foundMember.get().getEmail()).isEqualTo("testuser@example.co
            m");
20.    }
21. }
```

[문제 3] 아래의 조건을 만족하는 **JPA 기반의 게시판 엔티티**를 작성하시오.

[요구사항]

항목	내용
Entity 생성	Article 엔티티를 작성한다. - id (Long, PK, Auto Increment) - title (String, not null) - content (String, not null) - createdAt (LocalDateTime, 자동 생성) - toString 메소드를 오버라이딩하여 객체의 상태를 출력한다.
Repository 작성	Article 엔티티를 기준으로 Repository 인터페이스 를 작성한다. - JpaRepository 를 상속받는다.
Service 작성	ArticleService 클래스를 작성한다. - createArticle(String title, String content) 메소드를 구현한다. - 서비스는 트랜잭션으로 관리되어야 한다.
DB 설정	내장형 H2 데이터베이스 를 사용하여 구현한다.
디렉토리 구조 예시	src/main/java/com/sec03/entity/Article.java src/main/java/com/sec03/repository/ArticleRepository.java src/main/java/com/sec03/service/ArticleService.java
패키지명	com.sec03

추가 코드로 테스트를 실행한다.

- | |
|---|
| <ol style="list-style-type: none">1. ArticleRepositoryTest 라는 이름의 테스트 클래스를 작성한다.2. @DataJpaTest 를 활용하고, 글 저장/조회 테스트를 작성한다.3. 패키지명: com.sec03.repository |
|---|