

<<게시글 -> 댓글 관리를 구현하는 로직을 설계하려고 한다. >>

[문제 1] 게시글 관리를 위한 REST API 를 구현하려는 내용이다. 아래 요구사항에 맞게 문제를 푸시오.

항목	내용
Entity 생성	<ul style="list-style-type: none">- Post 엔티티를 작성한다.- id (Long, PK, Auto Increment)- title (String, not null)- content (String)- toString() 메서드를 오버라이딩하여 객체 상태를 출력한다.
Repository 작성	<ul style="list-style-type: none">- Post 엔티티를 기반으로 Repository 인터페이스를 작성한다. - JpaRepository 를 상속받는다.
Controller 작성	<ul style="list-style-type: none">- PostController 클래스를 작성한다.- CRUD 메서드 (createPost, getAllPosts, updatePost, deletePost)를 구현한다.
DB 설정	<ul style="list-style-type: none">- 내장형 H2 데이터베이스를 사용하여 구현한다.
디렉토리 구조 예시	<pre>src/main/java/com/sec01/entity/Post.java src/main/java/com/sec01/repository/PostRepository.java src/main/java/com/sec01/controller/PostController.java</pre>
패키지명	com.sec01
테스트 코드	src/test/java/com/sec01/controller/PostControllerTest.java 에 테스트 클래스를 작성하여, CRUD 동작을 검증한다.

[문제 2] 게시글에 대한 댓글 관리를 위한 REST API 를 구현하려는 내용이다. 아래 요구사항에 맞게 문제를 푸시오.

항목	내용
Entity 생성	<ul style="list-style-type: none">- Comment 엔티티를 작성한다.- id (Long, PK, Auto Increment)- content (String)- post (Post, ManyToOne 관계)- toString() 메서드를 오버라이딩한다.
Repository 작성	<ul style="list-style-type: none">- Comment 엔티티를 기반으로 Repository 인터페이스를 작성한다.- JpaRepository 를 상속받는다.
Controller 작성	<ul style="list-style-type: none">- CommentController 클래스를 작성한다.- 댓글 CRUD 메서드 (addComment, getComments, updateComment, deleteComment)를 구현한다.- URL: /posts/{postId}/comments
DB 설정	<ul style="list-style-type: none">- 내장형 H2 데이터베이스를 사용하여 구현한다.
디렉토리 구조 예시	<pre>src/main/java/com/sec02/entity/Comment.java src/main/java/com/sec02/repository/CommentRepository.java src/main/java/com/sec02/controller/CommentController.java</pre>
패키지명	com.sec02
테스트 코드	<pre>src/test/java/com/sec02/controller/CommentControllerTest.java</pre> <p>a 에 테스트 클래스를 작성하여, 댓글 API 동작을 검증한다.</p>

[문제 3] **Spring Security** 를 활용해 사용자 인증 및 권한을 설정하려는 내용이다. 아래 요구사항에 맞게 문제를 푸시오.

항목	내용
Security 설정	<ul style="list-style-type: none">- SecurityConfig.java 를 작성한다.- InMemoryUserDetailsManager 를 사용하여 사용자(user/user123, admin/admin123)를 생성한다.- /admin 은 ADMIN 만 접근 가능, /posts/**는 인증된 사용자만 접근 가능하게 설정한다.
Controller 작성	<ul style="list-style-type: none">- SecurityController 를 작성한다.- /admin 경로에서 "관리자 페이지!" 메시지를 출력한다.
디렉토리 구조 예시	src/main/java/com/sec03/security/SecurityConfig.java src/main/java/com/sec03/controller/SecurityController.java
패키지명	com.sec03
테스트 코드	<ul style="list-style-type: none">- src/test/java/com/sec03/security/SecurityConfigTest.java 에 Security 설정을 테스트하는 코드를 작성한다.