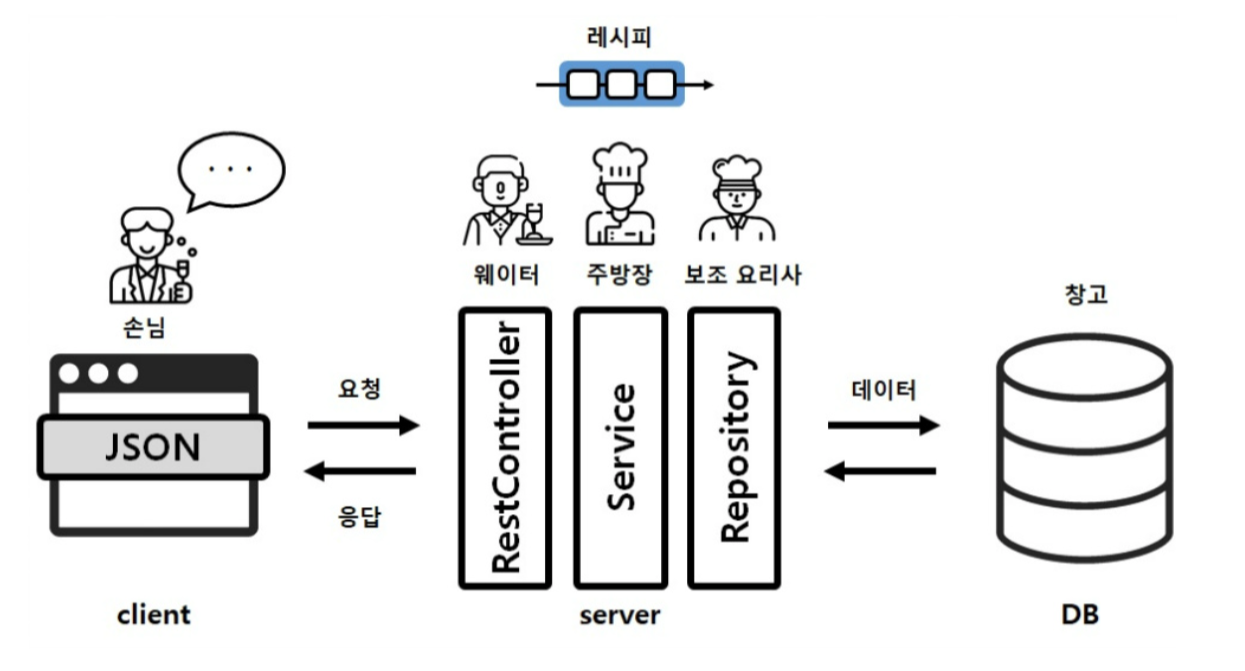
서비스 계층과 트랜잭션

서비스는 컨트롤러와 리파지터리의 사이에 위치하는 계층. 서버의 핵심 기능을 처리하는 순서를 총괄



트랜잭션 : 모두 성공되어야 하는 일련의 과정을 의미

서비스 계층 만들기

게시글 조회 요청 개선하기

단일 게시글 조회 요청 개선하기

게시글 생성 요청 개선하기

게시글 수정 요청 개선하기

게시글 삭제 요청 개선하기

트랜잭션 맛보기

1. 게시판에 데이터 3개를 한꺼번에 생성요청
2. 데이터를 DB에 저장하는 과정에서 의도적으로 오류 발생시키기
3. 어떻게 롤백되는지 확인

@RequestBody 어노테이션은 Post 요청 시 본문에 실어 보내는 데이터를 transactionsTest() 메소드의 매개변수로 받아오는 역할을 진행

컨트롤러는 웨이터 처럼 요청을 받고 결과만 반환 실제 작업은 서비스가 진행

1분퀴즈

transaction 이란 모두 성공되어야 하는 일련의 과정으로 쪼갤 수 없는 업무 처리의 최소 단위이다. transaction이 선언된 코드 내부에서 만약 실행이 실패한다면 변경된 데이터를 모두 이전 값으로 되돌리는데, 이러한 동작을 롤백이라 한다.

마무리

1. 서비스 : Controller와 repository 사이에 위치하는 계층으로 서버의 핵심 기능을 처리하는 순서를 총괄합니다. 클라이언트가 요청을 보내면, 이를 컨트롤러가 받아 서비스로 전달하고, 서비스는 정해진 코드 흐름을 따라 처리를 진행한다. 이때 처리에 필요한 데이터는 repository가 DB에서 가져와 서비스로 반환된다. 이렇게 대부분의 웹 서비스는 Controller와 repository 사이에 서비스 계층을 두어서 역할을 분업화한다.

2. 트랜잭션 : 모두 성공되어야 하는 일련의 과정으로, 쪼갤 수 없는 업무 처리의 최소 단위를 의미. 트랙잭션은 보통 서비스 단에서 관리

3. 롤백 : 트랜잭션 내부에서 실행에 실패하면 지금까지 수행한 것을 모두 폐기하고 진행 초기 단계로 되돌리는데, 이를 롤백이라고 한다.

4. @Service : 해당 어노테이션이 선언된 클래스를 서비스로 인식해 서비스 객체를 생성한다. 컨트롤러는 DI (의존성 주입)하는 방식으로 서비스 객체 사용이 가능

5. @Transactional : 해당 어노테이션이 선언된 클래스 또는 메서드를 트랜잭션으로 묶는다. 트랜잭션으로 묶인 클래스나 메서드는 처음부터 끝까지 완전히 실행되거나, 아예 실행되지 않거나 둘 중 하나로 동작한다. 이는 중간에 실패한다면 롤백되어 처음 상태로 되돌아가기 때문이다.