

서비스 계층과 트랜잭션

12.1 서비스와 트랜잭션의 개념

서비스(service)

- 컨트롤러와 리파지터리 사이에 위치하는 계층
- 서버의 핵심 기능(비즈니스 로직)을 처리하는 순서를 총괄

트랜잭션(transaction)

- 모두 성공해야 하는 일련의 과정
- 쪼갤 수 없는 업무 처리의 최소 단위
- 롤백(rollback) : 트랜잭션이 실패로 돌아갈 경우 진행 초기 단계로 돌리는 것

12.2 서비스 계층 만들기

REST 컨트롤러에 서비스 계층을 추가해 컨트롤러, 서비스, 리파지터리의 역할 분업하기

- ArticleApiController 내 모든 코드를 주석 처리
- 객체 주입된 리파지터리를 서비스로 바꾸기
- com.example.firstproject에 service 패키지 생성
- ArticleService 클래스 생성
- @Service 어노테이션 사용
 - @Service : 해당 클래스를 서비스로 인식해 스프링 부트에 서비스 객체를 생성
- 리파지터리와 협업하도록 articleRepository 필드 추가, @Autowired 통해 객체 주입

1) 게시글 조회 요청 개선하기

모든 게시글 조회 요청 개선하기

- index() 메서드 주석 풀기
- index() 메서드에서 return 문의 articleRepository.findAll() 삭제, articleService.index() 메서드 호출
- ArticleService 클래스에 index() 메서드 생성
 - return articleRepository.findAll(); 문을 작성해 DB에서 조회한 결과를 반환

단일 게시글 조회 요청 개선하기

- show() 메서드 주석 풀기

- show() 메서드에서 return 문의 articleRepository.findById(id).orElse(null)을 삭제, articleService.show(id) 호출
- ArticleService 클래스에 show() 메서드 생성
 - DB에서 id로 조회한 결과를 반환, 없으면 null을 반환하도록 return문 작성

2) 게시물 생성 요청 개선하기

- create() 메서드 주석 풀기
- POST 요청이 들어오면 리파지처리가 아닌 서비스가 처리하도록 수정
 - dto.toEntity() 삭제, articleService.create() 호출, 괄호 안에 dto 넣기
 - create() 메서드로 얻은 객체 이름을 created로 수정
- return 문에서 받은 created가 null이 아니면 good 요청, 그렇지 않으면 bad 요청 보내도록 함
 - good 요청 → ResponseEntity 상태는 OK, 본문에는 created
 - bad 요청 → ResponseEntity 상태는 BAD_REQUEST, 본문은 없으므로 빌드
 - create() 메서드 반환형을 ResponseEntity<Article>로 수정
- ArticleService 클래스에 create() 메서드 생성
 - dto를 엔티티로 변환해 article 객체에 저장
 - 리파지터리에 article을 DB에 저장하도록 함
- create() 메서드에서 article 객체에 id가 존재한다면 null을 반환하는 코드 추가

3) 게시물 수정 요청 개선하기

- update() 메서드 주석 풀기 및 코드 복사
- 컨트롤러가 컨트롤러의 업무만 처리하도록 수정
 - update() 메서드 호출, 매개변수로 id와 dto 전달, updated 객체로 반환값을 받아 저장
 - updated에 내용이 존재 → ResponseEntity의 상태에는 OK, 본문에는 updated
 - updated에 내용이 존재x → ResponseEntity의 상태에는 BAD_REQUEST, 본문은 없으므로 빌드
- ArticleService 클래스에 update() 메서드 생성
 - 복사해둔 update() 메서드 코드 붙여넣기
 - 로깅할 수 있게 @Slf4j 어노테이션 추가
 - 잘못된 요청 처리 부분 코드의 return 문을 null 반환하도록 수정
 - 업데이트 부분 코드의 return 문을 updated 반환하도록 수정

4) 게시물 삭제 요청 개선하기

- delete() 메서드 주석 풀기 및 코드 복사
- 삭제 요청이 오면 컨트롤러는 요청 받아 결과만 응답하는 역할 하도록 수정

- ArticleService의 delete(id) 메서드 호출, deleted 객체로 받아옴
- deleted에 내용이 존재 → ResponseEntity의 상태에는 NO_CONTENT, 본문은 빌드
- deleted에 내용이 존재x → ResponseEntity의 상태에는 BAD_REQUEST, 본문은 빌드
- ArticleService 클래스에 delete() 메서드 생성
 - 복사해둔 delete() 메서드 코드 붙여넣기
 - 잘못된 요청 처리 부분 코드의 return 문을 null 반환하도록 수정
 - 대상 삭제하는 부분 코드의 return 문을 target 반환하도록 수정