

@SpringBootTest: 스프링 컨테이너와 테스트 함께 실행

@Transactional: 테스트 케이스에 이게 있으면 테스트 전에 트랜잭션 시작, 테스트 완료 후에 항상 롤백. 이렇게 하면 DB에 데이터가 남지 않아 다음 테스트에 영향을 주지 않음.

스프링 JdbcTemplate

순수 Jdbc와 동일한 환경설정을 하면 된다.

스프링 JdbcTemplate과 MyBatis 같은라이브러리는 JDBC API에서본반복코드를대부분 제거해준다. 하지만 SQL은직접작성해야한다.

JPA

JPA는기존의반복코드는물론이고, 기본적인 SQL도 JPA가직접만들어서실행해준다.

JPA를사용하면, SQL과데이터중심의설계에서객체중심의설계로패러다임을전환을할수있다.

JPA를사용하면개발생산성을크게높일수있다.서본반복코드를대부분 제거해준다. 하지만 SQL은직접작성해야한다.

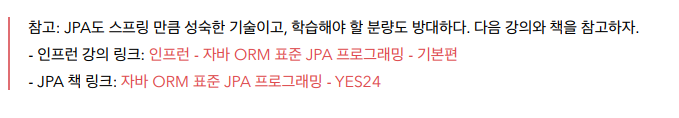
spring-boot-starter-data-jpa는 내부에 jdbc관련라이브러리를 포함한다. 따라서 jdbc는 제거해도 된다.

스프링부트에 JPA 설정 추가: resources/application.properties

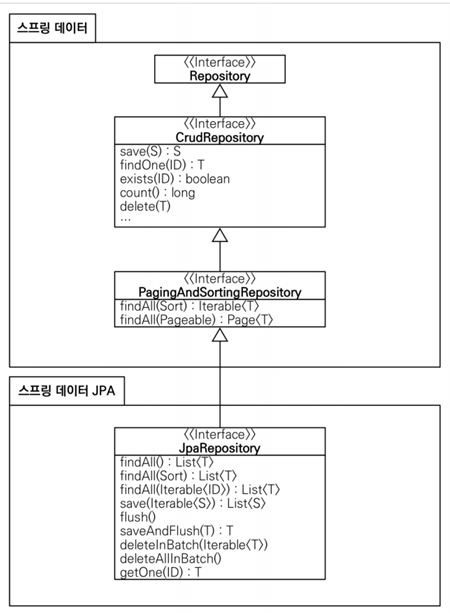
\*주의\*: 스프링부트 2.4부터는 spring.datasource.username=sa를 꼭 추가해주어야 한다. 그렇지 않으면 오류가 발생한다.

show-sql: JPA가생성하는 SQL을 출력한다.

ddl-auto: JPA는 테이블을 자동으로 생성하는 기능을 제공하는데 none를 사용하면 해당 기능을 끈다. create를 사용하면 엔티티정보를 바탕으로 테이블도 직접 생성해준다.



스프링 데이터 JPA  
스프링 부트와 JPA만 사용해도 개발 생산성이 정말 많이 증가하고, 개발해야 할 코드도 확연히 줄어듭니다. 여기에 스프링 데이터 JPA를 사용하면, 기존의 한계를 넘어 마치 마법처럼, 리포지토리에구현클래스없이 인터페이스만으로 개발을 완료할 수 있습니다. 그리고 반복 개발해 온 기본 CRUD 기능도 스프링 데이터 JPA가 모두 제공합니다.  
스프링 부트와 JPA라는 기반 위에, 스프링 데이터 JPA라는 환상적인 프레임워크를 더하면 개발이 정말 즐거워집니다. 지금까지 조금이라도 단순하다 반복이라 생각했던 개발코드들이 확연하게 줄어듭니다. 따라서 개발자는 핵심 비즈니스로 직을 개발하는데, 집중할 수 있습니다.  
실무에서 관계형데이터베이스를 사용한다면 스프링 데이터 JPA는 이제 선택이 아니라 필수입니다.



AOP  
AOP가 필요한 상황  
모든 메서드의 호출시간을 측정하고 싶다면?  
공통 관심사항(cross-cutting concern) vs 핵심 관심사항(core concern) 회원가입 시간, 회원 조회시간을 측정하고 싶다면?

