Sec01

Vue + Springboot 연동하기

작성자: 강태경



# Service 에서의 단위 테스트

Service 단에서 단위 테스트를 진행할때도 역시 DB 접속, WAS 구동을 끊고 테스트를 진행함, 테스트 진행을 위해 가짜 객체를 사용( Mockito 프레임 워크 사용 )

* Service 단위 테스트 :   
  @Mock @InjectMocks 가짜 객체를 사용해서 테스트 진행
* Service 테스트 시 Repository 나 Controller 에 모두 독립적이어야 함  
   Given(테스트 사전 조건이 주어짐),  
   When(실제 테스트가 진행됨),  
   Then(테스트 결과를 알려줌, assertThat()) 순서로 나는 간단하게 이해
* **@Mock** : 가짜 걕체를 생성하는 어노테이션, Service 클래스 단위 테스트는 주로 리파지토리 가짜 객체를 생성함
* **@InjectMocks** : Service 함수 테스트 실행을 위해 Service 객체에 가짜 객체를 넣음
* verify(리포지토리객체, times(실행횟수)).함수() : 싫행이 몇번 되는지 검증
* assertThat().isEqualTo(결과예측) : 결과 점검하는 함수, assertJ 라이브러리,  
   다양한 체이닝 함수가 있음 : .isEqualTo(), isGreatThan(), isLessThan() 등

# Repository 에서의 단위 테스트

Repository 단에서 단위 테스트를 진행할때는 DB 접속을 해서 수행하며, 내장 H2 DB 또는 외장 DB를 사용해 테스트 할 수 있음

@DataJpaTest : 테스트 후 자동 롤백 실행시키는 어노테이션으로 Repository 단위 테스트에 사용함

@AutoConfigureTestDatabase(replace = AutoConfigureTestDatabase.Replace.NONE)

: (NONE) 외부 DB 접속 후 단위 테스트 수행하는 어노테이션 ( .properties 설정으로 테스트 수행 )

, 기본 설정은 H2 스트링부트 내장 DB를 사용해 테스트 수행하게 되어 있음

assertThat() : 결과 점검하는 함수, assertJ 라이브러리,

다양한 체이닝 함수가 있음 : .isEqualTo(), isGreatThan(), isLessThan() 등