

WA#LE

S T U D I O



와플스튜디오 Backend Seminar

Instructor: 강지혁

2022.11. 01.(화) 20:00

Session 3

Table of Contents

- [과제 리뷰](#)
- 리팩토링
 - 좋은 코드란 무엇일까?
 - 컴포넌트의 책임을 적절히 분배하기
 - 쿼리 튜닝
- 테스트
 - JUnit5 & Mockk
 - Test Doubles

Assignment 2 Review

Step 0. 인증 구현하기

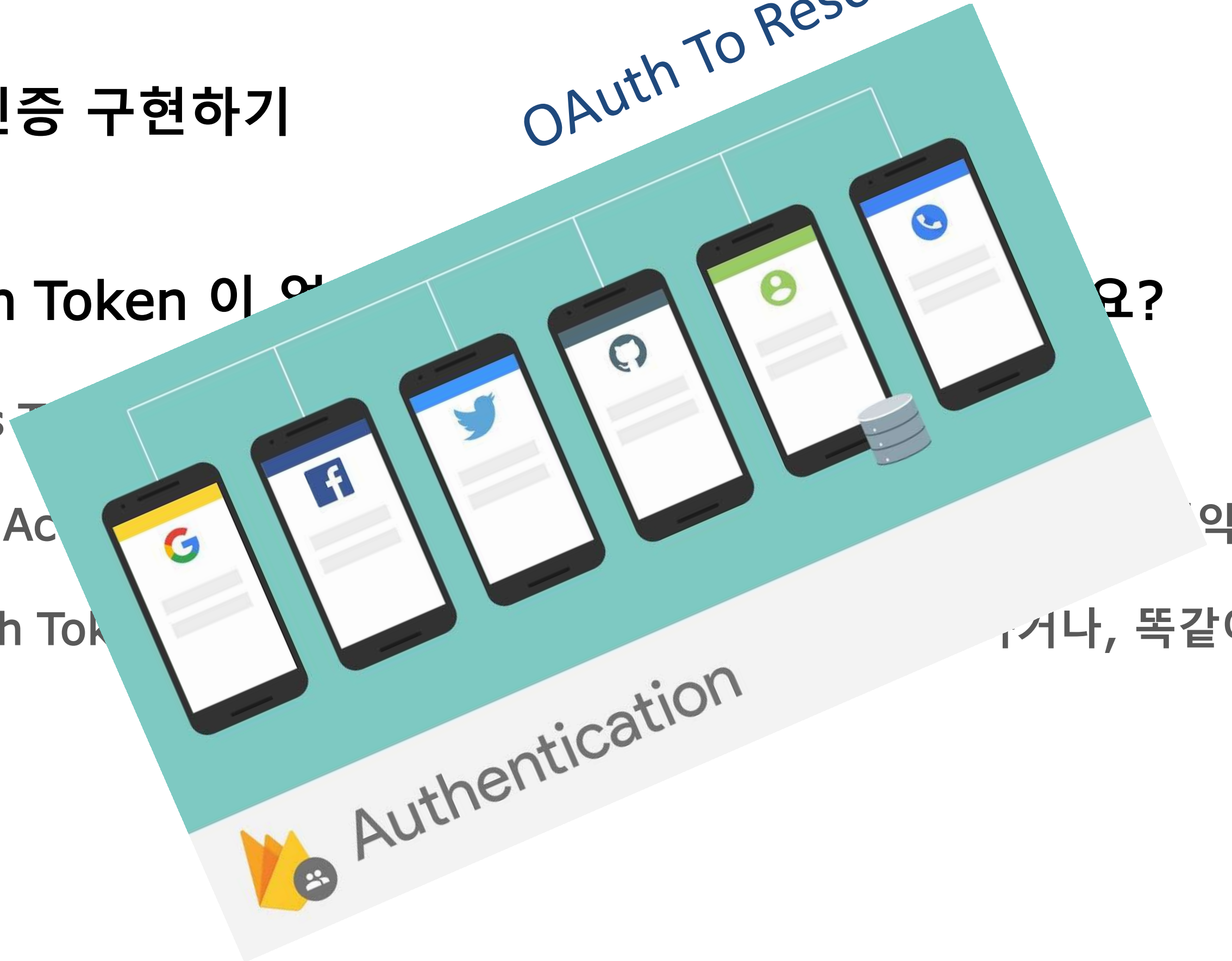
- Refresh Token 이 없

- Access Token

- 따라서 Access Token

- Refresh Token (없거나, 똑같이 탈취 위험이 있거나..)

OAuth To Rescue



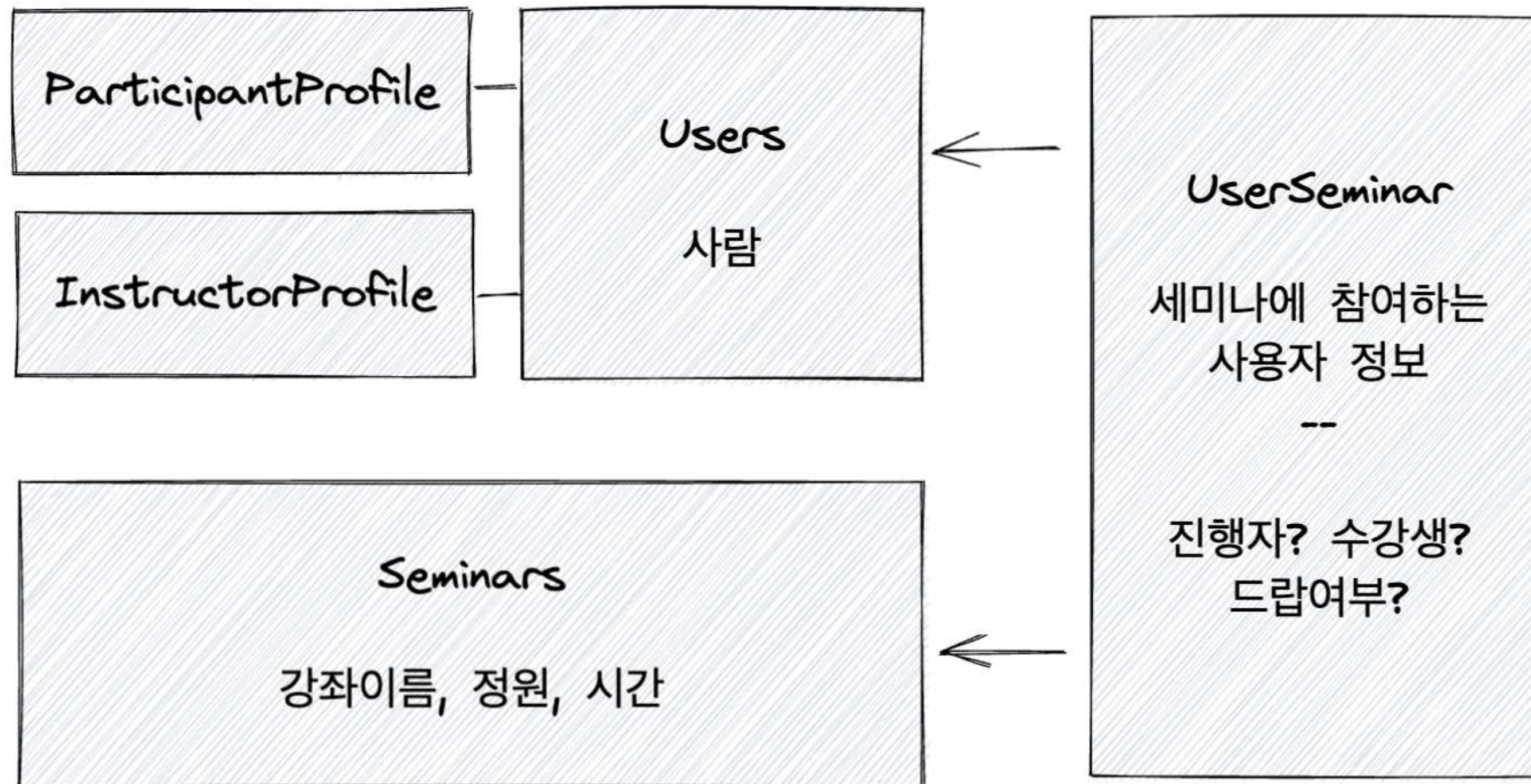
요?

약

거나, 똑같이 탈취 위험이 있거

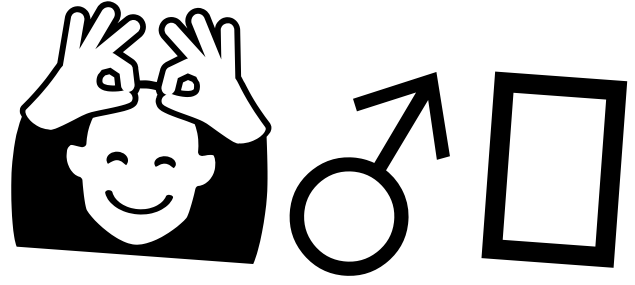
Assignment 2 Review

Step 1. 도메인 설계하기 : 비즈니스 요구사항과 연관관계



Assignment 2 Review

Step 2. 요구사항 잘 읽고 각 잡기

- 과제를 하다 보면..
 - 이런 경우는 어떻게 해야 하지?
 - 알아서 하라는 건가?!
- 판단과 실행은 개발하는 여러분의 몫
 - 어쩌면 실력만큼이나 중요한 역량
 - 애매할 때 : 바로 질문하기 vs 알아서 처리하기
 - 문제를 모를 때 : 찾아보기 vs 아는 범위 내에서 해결하기

Assignment 2 Review

Step 3. 로직 실제로 구현하기

- 어떤 점이 가장 어려웠나요?
 - Ex) 사소한 코드 디자인 : 에러는 언제 어떻게 발생시켜야 할까?
- 어떻게 프로젝트를 발전시키는 것이 베스트일까요?
 - Ex) 패키지 구조는 어떻게 관리할까? 어떤 순서로 짜야 효율적일까?
- 기술적인 고민은 어떤 것들이 있을까요?
 - Ex) 응답 시간을 어떻게 줄일 수 있을까? 성능적인 문제로는 어떤 것이 있을까?

Refactoring : Query Tuning

- 쿼리 수를 신경 쓰자
- JPA의 N+1 문제
 - Fetch Join은 만능일까?
 - Memory ⇔ Speed 사이의 Trade-Off
 - Pagination을 통해 적절히 의사소통 할 수 있다
 - Fetch Join, Pagination의 사용 예시 확인해보기
 - 가급적이면 QueryDSL 통해 사용하자

Refactoring : Clean Code

읽기 쉬운 코드가 왜 중요할까?

- 보통의 프로젝트는 오래, 같이 만들기 때문
- 생산성이 더 중요할 수도, 성능이 더 중요할 수도 있다
- 예시로 살펴보아요

협업을 돕는 수많은 수단들

- OOP, REST API
- Linter
- [Git-Flow](#)

Testing : Junit

JVM의 테스트 프레임워크

Spring과 잘 붙는다

Given / When / Then 형식으로 작성해보자

@SpringBootTest 어노테이션

- Junit, Spring Test, assert, Mockito 등 테스트 의존성 통합 제공
- 스프링 컴포넌트가 아니기 때문에, @Autowired 붙여주자
- [관련 ref](#)

Testing : Mockk & Test Doubles

Test Doubles : Mock, Spy, Stub 등, 테스트를 위한 컴포넌트들 [\(ref\)](#)

Test Double

스턴트 대역 (Stunt Double)에서 파생한 말로, 객체를 테스트 하는 상황에서, 의존하는 객체들을 대신해 동작하는 대역 객체들을 일컫는 말입니다.

- Mock Group : 외부로 나가는 상호작용을 모방
 - DB에 값을 쓰는 것처럼, 외부 의존성에 변경을 만드는 경우 사용합니다.
 - mock
 - spy (handwritten mock)
 - spy 객체의 경우 단순 mock 기능 뿐만 아니라, elapsed time , logging 등 우리가 원하는 동작을 추가적으로 사용하게끔 만들기도 합니다.
- Stub Group : 내부로 들어오는 상호작용을 모방
 - DB에서 값을 읽는 것처럼, 부작용이 없는 경우 사용합니다.
 - dummy (값에 의미 X)
 - stub (시나리오에 따라 원하는 값을 반환)
 - fake (아직 존재하지 않는 의존성을 대체할 때 씬, 그냥 통상적으로 우리가 아는 stub)

테스트 작성 방법 : Black Box vs White Box

- 예제와 함께 살펴보아요

Testing : Mockk & Test Doubles

@WebMvcTest

- API 요청을 받아서, 어떻게 처리하는지 까지만 (Interceptor 등) 확인하고 싶을 때

@DataJpaTest

- 데이터 접근 레이어 (쿼리 수행) 정도만 확인하고 싶을 때

실제 앱이 점점 무거워지게 되면, 앱을 전체 로딩하는 것이 테스트에 따라 낭비일 수 있다

(시간 & 메모리)

Next Assignment

- 조원의 코드에 블랙박스 테스트 작성하기
 - 조원들의 PR 브랜치에 들어가서, SeminarService에 대한 테스트를 작성해주세요!
 - 테스트를 작성하면서, 틀리는 부분이 있다면 작성자에게 리포트해주세요.
 - 그 외의 피드백도 자유롭게 주고받아 주세요. 모든 피드백을 반영할 필요는 없어요
 - 협업이 빠그덕 거리더라도, 구글 설문을 통해 감안할 테니 유익한 시간이 되었으면 좋겠습니다.
- 이와 같은 과정을 통해
 1. 과제 2 스펙에 적합하고
 2. 실제 동료의 의견을 받아들여 협업에 더 유리해진 코드를 확인해봅시다

조별 생존신고

- 이번 과제를 제출하지 않으신 분들도 많으셔서,
조별 Peer Review 가 원활히 이루어지지 않을 수 있을 듯 합니다.
따라서 제출하신 분들에 한해 2인 1조를 구성해보았습니다.
- PR # 기준
 - 81, 82
 - 83, 85
 - 86, 87
 - 88, 89
 - 90, 95
 - 92, 93
 - 94, 84

Next Seminar

- 배포하기
 - Docker & Kubernetes
 - AWS EC2, Load Balancers
- 더 다양한 기술들
 - Caching
 - Non Blocking I/O
 - NoSQL

Q&A

WA#LE
STUDIO

WA#LE
STUDIO