



황수영 | Back-end

홍익대학교 실내건축학/컴퓨터공학 (복수전공)



@sooyoungh



sooyoung.h8@gmail.com



sooyoungh.github.io



010-8922-7426

Introduce

사람들에게 유용한 서비스를 만들어 주는 개발의 매력에 빠져 꾸준히 공부하고 있습니다. **커뮤니케이션이 잘 되고 협업**하고 싶은 개발자, **꾸준히 배우고 성장**하는 개발자, 그리고 **사용자 관점에서 고민하며 항상 개선점을 찾는** 개발자가 되고자 아래와 같은 노력을 하고 있습니다.

- **사용자 관점에서 생각하며 개발합니다.** 사용자에게 가장 중요하거나 자주 사용하는 기능을 위주로 성능을 개선합니다. 예를 들어, 어떤 서비스에서 다른 사용자를 검색하는 과정이 항상 필요한 경우, 해당 기능을 우선적으로 개선합니다. 또한 졸업 프로젝트에서 직접 사용자 시연 후, 50개의 피드백을 받았습니다. 또한 **개선점을 항상 고민하며 개발**합니다. **조회 속도, CI/CD 반영 속도를 개선**하고 **CS 지식을 활용**하며 개발합니다.

- **블로그 글을 통해 논리적으로 공부한 내용과 생각을 정리**합니다. 단순 공부한 내용에 그치지 않고 프로젝트에 적용하거나 해당 기술을 사용하는 이유/과정/장단점/이후에 더욱 개선해볼 점 등의 논리 구조를 가지고 글을 정리합니다. 또한 가독성과 이해를 위해 이미지화합니다.

- 적극적으로 커뮤니케이션합니다. 협업할 때, **다른 팀원들과 적극적으로 공유**하고 **문서화물을 활용**하여 소통합니다. 또한 팀스터디를 통해 지식을 공유하며 성장합니다.

Projects

P : 문제 **S** : 해결 : 관련 글 포스팅

Remember Me

2023.02.01~2023.03.31

디자인, AI, 모바일, 백엔드 각 1명
총 4인 협업 중 백엔드 담당

- 소개 : **치매 예방**을 위한 **추억 회상** 및 **퀴즈** 어플리케이션
- 기능 : JWT 로그인, 추억 알림(FCM), 퀴즈, 커뮤니티 기능 등
- 기술 스택 : Java Spring Boot, Gradle, Spring Data JPA, MySQL



☑ 인덱스로 조회 속도 개선

- P** 사용자 검색 시, 불필요한 Full Scan으로 속도 저하
해당 프로젝트의 사용자 검색(조회) 기능에서는 모든 column 데이터가 필요하지 않고, 아이디/닉네임/프로필 이미지만 필요하였기에 커버링 인덱스 생성을 고려함.
성능 비교를 위해 커버링 인덱스와 보조 인덱스를 생성하여 조회 쿼리 속도 비교함. (100만건 기준)

결과	Full Scan	보조 인덱스	커버링 인덱스
Query Cost	1037.65	0.71	0.45
쿼리 실행 시간	0.2153s	0.004s	0.003s

- S** 커버링 인덱스가 가장 속도가 개선되었으나, **보조 인덱스와의 속도 차이가 크지 않고 해당 테이블의 update가 많은 점을 고려하여 최종적으로는 보조 인덱스로 속도 개선함.**

개념 글 : DB 인덱싱 알아보기(클러스터형, 보조, 커버링)

실전 글 : 커버링 인덱스 생성하여 조회 속도 비교하기

☑ FCM 푸시 알림 비동기로 구현

- **FCM 푸시 알림을 사용하여 전체 사용자에게 추억 회상 알림 일괄 전송**(9시/18시)
이때, 기본적으로 FCM의 HTTP 프로토콜은 동기 방식이기 때문에 동시에 전체 사용자에게 알림을 보내기엔 비효율적이라 개선이 필요. 따라서 **FCM 비동기 방식으로 구현**하여 사용자 의 정상 수신을 확인할때까지 기다리지 않고, 계속 다른 사용자에게 알림 보내도록 구현.
- 개념 글 : 동기 vs 비동기, 블로킹 vs 논블로킹 차이 정리글
- 실전 글 : FCM 푸시 알림 비동기로 구현하기

✓ Github Actions, Docker로 CI/CD 구축 및 반영 속도 개선

- 소개 : Docker + Github Actions로 CI/CD 환경 구축
[Spring Boot와 MySQL을 Docker-compose로 연동하여 배포.](#)
- P 매 커밋마다 CI/CD 반영되는데 약 2분 후반 ~3분 정도 소요
- S Gradle 의존성 캐싱, Docker Image 캐싱하여 속도 개선.
불필요한 파일은 워크플로우에서 제외, jobs 분리하여 병렬 실행.
CI/CD 반영 속도 평균 **180s에서 100s으로 약 1분** 속도 개선.
- 📖 : [Github Actions CI/CD 속도 개선하기](#)
- CI/CD 구조도 참고 : [Remember Me 레포지토리](#)

✓ JPA N+1 문제 - fetch join으로 해결

- P 일대다 관계에서 N+1 쿼리 문제 발생
Post와 Comment(일대다)관계를 **지연 로딩으로 설정해주었으나 N+1 문제 발생.**
Post 조회 시 연관 엔티티인 Comment를 함께 조회하면, Post 개수만큼 추가적으로 Comment를 조회하는 불필요한 쿼리가 발생하여 성능 저하 이슈 발생.
- S 각 Post마다 Comment 총 개수만 알면 되기 때문에 JPQL fetch join으로 해결.
(페이징이 필요없고, 또 다른 연관 엔티티가 없었기에 fetch join로 해결 가능)
- 📖 : [JPA N+1 문제 상황별 해결하기\(fetch join, Batch Size, DTO\)](#)

뮤랑이(MUic & Emotion)

2022.09.29~2022.11.25 (8주)
프론트, 백엔드 각 1명
총 2인 협업 중 백엔드 담당

- 소개 : 표정 분석 후 음악 추천 웹사이트
- 기능 : 구글 OAuth 로그인, 표정 분석 후 음악 추천, 감정 색상 달력 기록 등
- 기술 스택 : Java Spring Boot, Gradle, Spring Data JPA, QueryDSL, MySQL AWS(EC2, RDS, ACM, ELB), JavaScript, Python
- 학부 졸업 프로젝트 장려상, 전시장에서 시연 후 **약 50개의 피드백** 수집



✓ AWS 로드 밸런서(ELB)로 HTTPS 배포

- AWS ACM을 통해 **SSL 인증서 발급 후, AWS ELB(로드밸런서)에 등록하여 HTTPS로 배포.** (해당 프로젝트에서 웹캠을 사용하기에 보안을 위해 SSL 필요)
- 📖 : [HTTPS로 배포하기 with AWS](#)

CAFEMATE

2021.07.27~2021.08.31 (5주)
프론트, 백엔드 각 2명
총 4인 협업 중 백엔드 담당

- 소개 : 카페 추천 웹사이트
- 기능 : 추천 카페 조회, 좋아요, 댓글 기능 등
- 기술 스택 : Java Spring Boot, Gradle, JUnit4, Spring Data JPA, MySQL, AWS(EC2, RDS), React
- 기획부터 배포까지 진행한 첫 번째 팀 프로젝트로, DB/API 문서화, 깃허브
- 📖 : [Index로 조회 성능 개선하기 with MySQL](#)



Study & Bloging

- #Java, Spring Boot, CS 지식 공부 정리 마인드맵
- #로그인과 세션
- #JPA N+1 문제 해결 방법 비교
- #CS 팀스터디 레포지토리
- #구글 OAuth 동작 방식
- #JWT 동작 방식
- #HTTPS로 배포하기(AWS)
- #DB 동시성 문제 - 낙관락 해결기
- #상속보단 컴포지션
- #토비의 스프링
- #Java의 Call By Value