



황수영 | Back-end

E-mail



sooyoung.h8@gmail.com

Github



@sooyoungh

Blog



https://sooyoungh.github.io

Introduce

사람들에게 유용한 서비스를 만들어 주는 개발의 매력에 빠져 꾸준히 공부하고 있습니다. 커뮤니케이션이 잘 되고 협업하고 싶은 개발자, 그리고 CS 지식을 활용하여 개선점을 항상 고민하는 개발자가 되고자 아래와 같은 노력을 하고 있습니다.

- ◆ 지식 공유를 통해 성장합니다. 팀 스터디와 팀 프로젝트를 하며 배운 내용은 꾸준히 기술 블로그에 업로드합니다.
- ◆ 항상 개선점을 생각하며 개발합니다. 조회 성능, CI/CD 반영 속도를 개선하고 CS 지식을 활용하면서 개발합니다.
- ◆ 문서화를 통해 공유합니다. 문서화물을 적극 활용하여 팀원들과 적극적으로 커뮤니케이션합니다.

Skill

Java Spring Boot MySQL JUnit Spring Data JPA QueryDSL AWS(EC2, RDS) Python

Experience

Remember Me

2023.02.01~(진행중)
디자인, AI, 모바일, 백엔드 각 1명
총 4인 협업 중 백엔드 담당

- 소개 : 치매 예방을 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
- Docker + Github Actions CI/CD 환경 구축, CI/CD 속도 개선
- FCM 푸시 알림 비동기로 구현, 인덱스 조회 속도 개선, DB 동시성 문제 낙관적 락 해결



글또

2023.02~진행중

- 소개 : 글쓰는 개발자들 커뮤니티, 주기적인 글 작성 및 공유

우테코 프리코스

2022.11

- 소개 : 구현 과제, 클린 코드, 테스트 코드, 커밋 규칙 학습 및 피어 리뷰



뮤랑이(MUsic & 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
- HTTPS로 배포, 학부 졸업 프로젝트 장려상, 전시장에서 시연 후 약 50개의 피드백 수집



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, 테스트 학습
- MySQL Index를 사용하여 조회 성능 개선



Education

2023.01~2023.02

삼성 SDS 알고리즘 특강

2016.03~2023.02

홍익대학교 실내건축학과, 컴퓨터공학과 (복수전공)

저는 성능 개선을 고민하며 개발합니다.

인덱스로 조회 속도 개선

(Remember Me 프로젝트)

- 문제1 : 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 고려.
성능 비교를 위해 커버링 인덱스와 보조 인덱스를 생성하여 조회 쿼리 속도 비교.
⚡ 결과 : Full Scan(87ms) -> 보조 인덱스(42ms) -> 커버링 인덱스(39ms)로 개선.
- 해결 : 커버링 인덱스가 가장 속도가 개선되었으나, 보조 인덱스와의 속도 차이가 크지 않고 커버링 인덱스 트리 생성 비용이 훨씬 큰 것을 고려하여 보조 인덱스를 생성하여 속도 개선.
- 관련 글 포스팅1 : DB 인덱싱 알아보기(클러스터형, 보조, 커버링)
- 관련 글 포스팅2 : 커버링 인덱스 생성하여 조회 속도 비교하기

(CAFEMATE 프로젝트)

- 문제2 : 주소 컬럼(ex. 연남동)으로 카페 정보 조회할 때, Full Scan으로 속도 저하
- 해결 : MySQL Index를 사용하여 보조 인덱스 생성
주소 컬럼을 보조 인덱스로 사용하여 평균 34ms에서 10ms로 속도 개선
- 관련 글 포스팅 : Index로 조회 성능 개선하기 with MySQL

Github Actions 속도 개선

(Remember Me프로젝트)

- 문제 : 매 커밋마다 CI/CD 반영되는데 약 2분 후반 ~3분 정도 소요
- 해결 : Gradle 의존성 개선, Docker Image 캐싱하여 속도 개선.
불필요한 파일은 워크플로우에서 제외, jobs 분리하여 병렬 실행.
평균 180s에서 100s으로 약 1분 속도 개선
- 관련 글 포스팅 : Github Actions CI/CD 속도 개선하기

JPA N+1 문제

- fetch join으로 해결

(Remember Me프로젝트)

- 문제 : 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.
- 해결 : 각 Post마다 Comment 총 개수만 알면 되기 때문에 JPQL fetch join으로 해결.
(페이징이 필요없고, 또 다른 연관 엔티티가 없었기에 fetch join로 해결 가능)
- 관련 글 포스팅 : JPA N+1 문제 상황별 해결하기(fetch join, Batch Size, DTO)

저는 CS 지식을 적용하여 개발합니다.

FCM 푸시 알림 비동기로 구현

(Remember Me프로젝트)

- 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 구현하여 사용자의 정상 수신을 확인할때까지 기다리지 않고, 계속 다른 사용자에게 알림 보내도록 구현.
- 관련 글 포스팅 : 동기 vs 비동기, 블로킹 vs 논블로킹 차이 정리글
- 관련 글 포스팅 : FCM 푸시 알림 비동기로 구현하기

DB 동시성 문제

- 낙관적 락으로 해결

(Remember Me프로젝트)

- 같은 가족끼리는 추억 데이터를 공유함. 따라서 여러 사용자가 같은 추억 데이터를 동시에 수정할 경우 DB 동시성 문제가 발생함. 무결성이 크게 중요한 기능이 아니고, 충돌이 많이 발생하지 않으므로 비관락이 아닌 낙관락 사용.
엔티티에 JPA의 @Version 어노테이션 필드를 추가하여 낙관락으로 해결.
- 관련 글 포스팅 : DB 동시성 문제 - 낙관락으로 해결하기

저는 팀원들과 항상 소통하며 개발합니다.

디자이너, 프론트, 백엔드 팀원 간 커뮤니케이션 툴 적극 활용

(Remember Me 프로젝트)

- 기획 단계에서는 팀원들과 Whimsical로 함께 충분히 대화하며 프로토타입 제작 이후부터는 더 정확한 공유를 위해 [Figma로 변경 사항을 버전 관리](#)하여 미스 커뮤니케이션 최소화하고 협업을 효율적으로 진행.

참고 : [프로젝트\(Remember Me\) Figma 레포지토리](#)

- API 문서는 Postman과 Swagger로 공유

참고 : [프로젝트\(Remember Me\) API 문서 Postman](#)

Spring 예외 처리 전략 사용

(Remember Me 프로젝트)

- 프론트 개발자에게 보다 [정확한 에러 상황을 전달](#)해주기 위해 Spring 예외 처리 전략 (@ExceptionHandler, @RestControllerAdvice) 사용.
- 상황별 상태 코드 지정 및 구체적인 에러메시지 전달.
- 관련 글 포스팅 : [스프링의 예외 처리 전략, @ExceptionHandler 알아보기](#)

Git 커밋 규칙 및 클린코드 학습

(우테코 프리코스)

- 클린 코드, Git 커밋 규칙 학습 및 피어 코드 리뷰
- 관련 Repository : [우테코 프리코스 5기 레포지토리](#)

저는 실제 배포 후 네트워크를 공부하며 개발합니다.

AWS 로드 밸런서(ELB)로 HTTPS 배포

(MUrangE 프로젝트)

- 소개 : AWS ACM을 통해 [SSL 인증서 발급 후 AWS ELB\(로드밸런서\)에 등록하여 HTTPS로 배포](#). (해당 프로젝트에서 웹캠을 사용하기에 보안을 위해 SSL 필요)
- 관련 글 포스팅 : [HTTPS로 배포하기 with AWS](#)

Github Actions, Docker로 CI/CD 구축

(Remember Me 프로젝트)

- 소개 : Docker + Github Actions로 CI/CD 환경 구축 [Spring Boot와 MySQL을 Docker-compose로 연동하여 배포](#). MySQL은 GCS에 배포하고, EC2에 Docker 사용하여 배포.
- CI/CD 구조도 참고 : [Remember Me 레포지토리](#)

저는 내부 동작을 충분히 이해하고 개발합니다.

기술 블로그 및 팀스터디

- Java, Spring Boot, CS 지식들을 약 1년 간 꾸준히 공부 후 개인 블로그에 포스팅 스스로 이해한 것을 남들에게 쉽고 간단하게 설명해야 충분히 습득한 것이라 생각하여 [꾸준히 블로그 작성 및 팀스터디 진행](#). 시각화하여 쉽게 이해되도록 작성.
- 기술 블로그 : <https://sooyoungh.github.io/>
- 팀스터디 레포지토리 : <https://github.com/Cs-Java-Interview/Tech-Interview>

구글 OAuth 로그인/JWT 로그인

- 프로젝트에 새로운 기술 적용 전, 자료조사 후 충분히 이해하고 기술 적용. 이를 통해 어려가 발생하였을 때, 원인을 더욱 빠르게 파악 가능.
- 구글 OAuth 관련 글 포스팅 : [OAuth 동작 방식 알아보기](#)
- JWT 관련 글 포스팅 : [JWT 동작 방식 알아보기](#)