

정훈희

풀스택 개발자

☎ 010-5378-8920

🌐 github.com/JeongHunHui

✉ hunhui.jeong.db@gmail.com

🌐 velog.io/@jeong_hun_hui



Summary

- Spring Boot, FastAPI, Node.js로 서버를 개발하고, Docker, AWS, Github Actions로 클라우드에 서비스를 배포할 수 있습니다.
- MySQL의 실행 계획을 파악하고 적절한 쿼리 튜닝을 수행하여 성능을 향상시킨 경험이 있습니다. ([쿼리 성능 개선 과정](#))
- 서버, 클라이언트, 인프라 등 모든 분야에 대해 End To End 프로젝트 경험이 있습니다.
- 문제 해결 과정을 [블로그](#)에 꾸준히 기록하여 3년 동안 64개의 게시물을 작성했고, 총 조회수 30K를 달성했습니다.

Skills

- **Language** : Kotlin, Java, Python, JavaScript
- **Server** : **Spring Boot**(JPA, JUnit 5), FastAPI, Node.js(Express.js)
- **Data** : **MySQL**, **Redis**, MongoDB, Google BigQuery
- **Infra** : **AWS**(EC2, ALB, Route53, Cloud Front, S3), GCP, Docker, NGINX, Github Actions, K6
- **Monitoring** : Prometheus, Grafana, New Relic
- **Client** : React, Next.js, TanStack Query

Experience

소프트웨어 마에스트로 15기 2024.06 ~ 2024.11 sulmoon.io [GitHub](#)

소프트웨어 마에스트로 과정 중 개발한 서비스 [설문이용](#)은 AI를 통해 누구나 쉽게 설문조사를 만들고 관리할 수 있도록 돕는 서비스입니다. 팀의 리더를 맡아 서버, 클라이언트, 인프라 등 모든 분야에 적극적으로 기여했습니다. 대형마트와 연계해서 진행한 설문조사 이벤트를 성공적으로 수행했습니다.

사용 기술 : Spring Boot(Kotlin), MongoDB, FastAPI, AWS(EC2, ALB), Docker, Github Actions, New Relic, Next.js, TanStack Query

설문조사 관련 REST API 설계 및 구현 [설문조사 클래스 설계 및 구현 과정](#)

- 설문조사 관련 클래스 설계 및 구현, 설문조사 데이터에 적합한 NoSQL(MongoDB)을 활용하여 MongoDB 컬렉션 설계
- **Spring Boot**, **MongoDB**, Spring Data MongoDB를 활용하여 설문조사 관련 REST API를 설계 및 개발

CI/CD 파이프라인 구축을 통한 안정성 및 생산성 향상

- **Github Actions**, Ktlint, JaCoCo를 활용하여 코드 스타일 검사 및 테스트 커버리지 측정을 자동화하는 CI 파이프라인 구축
- **Github Actions**, **Docker**를 활용하여 **AWS** 환경에서 개발/운영 서버 배포 과정을 자동화하는 CD 파이프라인 구축

기업 연계 시나리오 기반 부하 테스트 진행, 성능 저하 원인 분석 및 개선 [부하 테스트 및 문제 해결 과정](#)

- 대형마트 회원 5천명을 대상으로 설문조사를 진행하기 전에 서비스 운영 안정성 검증을 위해 K6 부하 테스트를 진행
- 부하 테스트 과정에서 설문 응답 API 성능 저하 발생 → 모니터링 도구로 MongoDB 쿼리의 성능 저하 확인, 인덱스 생성을 통해 개선
- 성과 : 문제 조기 발견 및 해결, 피크타임 1시간 동안 약 2200건의 설문 참여 과정을 성공적으로 수행

설문 요약 및 설문 초안 생성 LLM 작업을 병렬 처리 하여 응답시간 개선

- **FastAPI**의 설문 생성 API에서 수행하는 두 가지 LLM 작업을 **asyncio**를 통해 병렬 처리하여 응답시간 39% 감소(41s → 25s)

JUnit5, Mockito를 활용하여 단위 테스트 작성, JaCoCo를 활용하여 테스트 커버리지 측정

- 코드 신뢰성 향상을 위해 단위 테스트 작성, 테스트 커버리지 50% 이상 유지

Next.js를 활용하여 설문 관리 페이지 및 마이 페이지 구현, TanStack Query를 통한 상태 관리 [관련 PR 링크](#)

(주)유니로보틱스 / 서버 개발팀 인턴 2022.12 ~ 2023.06 okomo.io

[okomo](https://okomo.io)는 1만 대 이상의 IoT 장비에서 데이터를 수집하는 파이프라인을 구축하여 실시간 장비 제어 및 대시보드를 제공하는 서비스입니다.

사용 기술 : Node.js(Express.js), MySQL, Redis, GCP(BigQuery), Docker, NGINX, Prometheus, Grafana

장비 관련 REST API 구현 및 개선, 쿼리 개선 과정 프로세스화를 통한 팀 내 생산성 향상

- **Express.js, MySQL**을 활용하여 장비 관리에 필요한 REST API 구현
- 장비 목록 조회 쿼리 성능 저하 발생 → Explain 쿼리로 index merge 실행 계획을 확인한 뒤 복합 인덱스를 생성하여 해결
- slow query 알람을 설정하여 쿼리 개선 과정을 프로세스화 → 쿼리 개선 과정이 간소화되어 팀 내 생산성 향상

대시보드 REST API 구현 및 대시보드 로딩속도 개선

- **Express.js** 환경에서 **BigQuery**의 IoT 장비 데이터를 조회해서 대시보드에 적합한 형태로 가공하는 API 구현
- BigQuery의 조회 성능을 개선하기 위해 장비 ID 기준 클러스터링 및 주 단위 파티셔닝 적용
- DB 접근을 줄이기 위해 변경이 적은 장비 정보 데이터와 주간 통계 데이터에 **Redis** 캐싱 적용
- 성과 : 대시보드 로딩시간 78% 감소(2.7s → 0.6s)

모니터링 및 알람 시스템 구축을 통한 서버 장애 및 부하 확인 자동화

- **Prometheus, Grafana**를 활용하여 모니터링 시스템 구축, Alertmanager, Slack을 활용하여 서버 장애 및 부하 알람 시스템 구축

Side Project

티켓월드 2025.02 ~ 진행 중 [GitHub](https://github.com)

공연의 티켓을 예매할 수 있는 서비스입니다. 다양한 문제들을 해결한 과정을 블로그에 기록하고있습니다.

사용 기술 : Spring Boot(Kotlin), JPA, MySQL, Docker, Prometheus, Grafana, New Relic, K6, React

티켓 예매 관련 REST API 설계 및 구현, 동시성 문제 해결 [중복 예매 문제 해결 과정](#) [구매 제한 수량 초과 문제 해결 과정](#)

- **Spring Boot, MySQL, JPA**를 활용하여 티켓 예매 관련 REST API 설계 및 구현
- **중복 예매 문제 발생** : 비관적 락 적용 시 락 대기시간 증가로 인한 성능 저하 발생 → 응답속도가 더 빠른 낙관적 락을 적용하여 해결
- **구매 제한 수량 초과 문제 발생** : 조회 대상과 수정 대상이 다르므로 낙관적 락 적용 불가능 → 비관적 락을 적용하여 해결, 실행 계획과 락의 범위를 분석하여 성능 저하 문제가 없음을 확인

Activity

소프트웨어 마에스트로 15기 [링크](#) 2024.04 ~ 2024.11 • 과학기술정보통신부 주관 SW 인재 양성 프로그램

우아한 스테디 [링크](#) 2023.11 ~ 2023.12 • 우아한 형제들에서 진행하는 클린코드 스테디

Techeer [링크](#) 2022.10 ~ 진행 중 • 실리콘밸리 엔지니어가 운영하는 개발자 그룹

GDSC [링크](#) 2022.08 ~ 2023.07 • 구글이 지원하는 대학생 개발 동아리

Education

한국공학대학교 / 소프트웨어전공 / 2019.03 ~ 2025.02 (졸업)