

이력서

김인철

Backend Engineer

Phone | 010-8857-3039

Email | ickim1218@gmail.com

Github | <http://github.com/kimich1218>

Blog | <https://velog.io/@kimich1218>

Summary

- **Spring Boot, FastAPI, Node.js** 를 활용하여 다양한 서버 아키텍처를 설계 및 구현 능력을 보유하고 있습니다.
- **Docker** 를 이용한 애플리케이션의 컨테이너화 및 **AWS EC2** 를 활용한 클라우드 기반 배포 전략을 통해 효율적인 개발 및 운영 워크플로우를 마련할 수 있습니다.
- 프론트엔드, 백엔드, 그리고 DevOps 를 아우르는 종단간(End-To-End) 프로젝트 수행을 통해, 소프트웨어 개발의 전 과정에 대한 깊은 이해와 경험을 쌓았습니다.

Tech Skills

- **Backend:** FastAPI, Spring Boot, Node.js, Flask
- **Database:** MySQL, PostgreSQL, Redis
- **DevOps:** Docker, AWS EC2, AWS S3

Projects

Voilio - 영상공유 및 소셜 네트워킹 플랫폼

1. 기술 스택

Spring Boot, JPA, QueryDSL, AWS S3, AWS EC2, Docker, JUnit, Mockito, Redis, Nginx

2. 주요 업무 및 성과

- **Spring Boot 와 JPA 를 활용한 RESTful API 설계 및 구현:** 사용자, 게시판, 댓글 등의 기본적인 CRUD 기능을 포함하여, 동적인 콘텐츠 관리 및 사용자 인터랙션 기능을 제공하는 RESTful API 를 설계 및 구현.
- **QueryDSL 을 이용한 복잡한 데이터 조회 최적화:** 복잡한 데이터 조회 요구 사항에 대해 QueryDSL 을 활용하여 동적 쿼리를 생성하고 성능 최적화를 달성. 특히, 키워드 기반 검색 및 카테고리 필터링 기능에서 QueryDSL 을 활용하여 효율적인 데이터 처리를 구현.
- **N+1 쿼리 문제 해결:** JPA 와 QueryDSL 을 결합하여 N+1 쿼리 문제를 해결. 특히, 게시판 목록 조회 및 게시글 상세 조회에서 발생할 수 있는 N+1 쿼리 문제를 fetch join 과 QueryDSL 을 통해 효과적으로 최적화.
- **실시간 채팅 기능 구현:** WebSocket 과 Redis 를 활용하여 실시간 채팅 기능을 구현. 사용자 간의 실시간 통신을 위해 WebSocket 프로토콜을 적용하고, 메시지 교환을 위한 메시지 브로커로 Redis 의 Pub/Sub 기능을 사용.
- **AWS S3 와 EC2 를 활용한 파일 저장 및 서버 배포:** 멀티미디어 콘텐츠(동영상, 이미지)의 안정적인 저장을 위해 AWS S3 를 사용하고, EC2 인스턴스에 애플리케이션을 배포하여 높은 가용성 및 확장성을 확보.
- **Docker 를 이용한 컨테이너화 및 배포 자동화:** 개발 및 배포 환경의 일관성을 유지하고, 배포 과정의 효율성을 개선하기 위해 Docker 컨테이너를 적용. Nginx 를 통한 리버스 프록시 설정으로 로드 밸런싱 및 보안 강화.
- **JUnit 과 Mockito 를 사용한 단위 및 통합 테스트:** 코드의 신뢰성을 보장하기 위해 JUnit 과 Mockito 를 사용하여 주요 기능에 대한 단위 및 통합 테스트를 수행. 지속적인 테스트를 통해 버그 발견 및 수정 시간 단축.
- **Spring RestDocs 를 활용한 API 문서 자동화:** 개발자와 최종 사용자가 API 를 손쉽게 파악하고 활용할 수 있도록, 상호작용이 가능한 API 문서를 자동 생성하여 제공

RUAlone – 혼밥러들을 위한 위치 기반 동행 찾기 어플리케이션

1. 기술 스택

Spring Boot, PostgreSQL (PostGIS), JPA, Jib

2. 주요 업무 및 성과

- **PostGIS 를 활용한 위치 기반 검색 기능 구현:** 사용자의 현재 위치를 기준으로 주변 1.5km 이내에 있는 게시글을 조회할 수 있는 기능을 PostGIS 를 이용하여 구현. 이를 통해 실시간 상호작용을 촉진.
- **동적 위치 기반 쿼리 최적화:** 사용자의 실시간 위치 데이터를 활용하여 동적으로 게시글을 검색하는 과정에서 발생할 수 있는 성능 문제를 PostGIS 의 공간 쿼리 최적화 기능을 이용하여 해결.
- **보안 및 인증 구현:** JWT 기반 인증 시스템을 구현하여, API 접근 시 보안을 강화하고 사용자 인증 프로세스를 효율적으로 관리.
- **Jib 를 사용한 컨테이너화:** Gradle 빌드 도구와 Jib 플러그인을 사용하여 Dockerfile 없이 Java 애플리케이션을 컨테이너 이미지로 직접 빌드
- **코드 스타일 자동화 및 일관성 유지:** GitHub Actions 를 활용하여 google-java-format 을 자동 실행하는 CI/CD 워크플로우를 구성 및 코드베이스에 push 나 pull request 이벤트가 발생할 때마다 트리거되어, Java 코드 스타일을 자동으로 포맷팅