

About Me



개인의 성장과 회사의 발전을 동시에 추구하는 백엔드 개발자 김현기입니다.
저는 기존 문제들을 더 **효율적이고 긍정적인 방식**으로 해결하고자 노력하며,
비용 절감, 인프라 개선, 좋은 커뮤니케이션 등 다양한 분야에서 회사의 성장에 기여하고자 합니다,
좋은 코드 작성뿐만 아니라, 팀 문화에 **긍정적인 영향을 주는 동료**가 되고자 합니다
이를 통해 **회사와 함께 성장하는 것**이 제 목표입니다.
감사합니다.

경력

패션 산업에 특화된 ERP 솔루션을 서비스합니다.
이 서비스를 통해 패션 브랜드들이 자원을 더 효과적으로 관리할 수 있도록 돕고 있습니다.
erp의 특성 상 WMS, CRM, SCM 등 다양한 부분의 여러 도메인을 경험하고
서비스 품질 향상을 위해 많은 노력을 기울였습니다.


Java Spring Framework Oracle Mybatis Gitlab Jenkins Scouter AWS CentOS

Work Experience (업무 경험)

SGU
IT 2팀 사원
2022.08 ~ 2024.04

- **ERP 와 Groupware 간 전자결재 연동 API 기능 개발**
 - 주요 기능을 표준화하고 전략 패턴을 적용해 전자결재 상태 관리 및 승인 프로세스 일관성을 확보.
 - 전자결재 표준을 추상화된 공통 인터페이스를 모듈화함으로써 개발 효율을 극대화하고 예상 개발 시간을 기존 6주에서 2주로 단축시킴.
- **통계성 데이터 처리 개선**
 - 브랜드 및 품목별 연간 매출, 시즌별 실적 목표 등 대규모 데이터를 위한 매우 복잡하고 느린 쿼리가 다수 존재.
 - 뷰 테이블과 인덱스, 로컬 캐시를 활용해 복잡성을 해소하고 평균 **15000ms(15초)** 이상 소요되던 쿼리를 **5000ms(5초)** 이하로 단축.
- **API 처리 속도 개선**
 - 비효율적인 데이터 조회 중복 문제를 해결하기 위해 레거시 코드에 페이지네이션을 도입.
 - 페이지 처리를 통해 기존 API 응답 속도가 **800ms(0.8초)** 이상 소요되던 쿼리를 **450ms(0.45초)** 이하로 단축.
- **CRM 서비스에서 네이티브 쿼리 개선**
 - CRM에서 계속 추가되는 연관 정보 노출 요청으로 인해 속도 저하 발생.
 - 기존 subquery를 분석하여 join으로 해소, 기존 **3500ms(3.5초)** 이상 소요되던 쿼리를 **800ms(0.8초)** 이하로 단축.
- **커뮤니티 서비스 리팩토링**
 - 매장 간, 본사 간, 매장 본사 간, MD 간 등 여러 목적의 커뮤니티들이 추가될수록 중복된 서비스가 많아지는 문제 발생.
 - 추상 팩토리 패턴을 활용해 중복된 서비스를 제거하고 추상화된 서비스를 통해 중복된 코드를 제거하며 유지보수성을 높임.
- **공통 기능 리팩토링 구현**
 - 다수의 서비스에서 사용되는 공통 코드의 과부하 문제를 해결하기 위해 각 서비스별로 필요한 기능만 포함하도록 모듈을 분리.
 - 이를 통해 공통 코드를 모듈화하고, 중복 코드를 제거하여 유지보수성과 재사용성을 높임.
- **WMS to ERP 물류 연동 API 개발**
 - 자사의 WMS와 타사 ERP 시스템 간 **물류 데이터 연동 API**를 개발.
 - **Optimistic Locking**를 통한 데이터 정합성 유지 이를 통해 물류 데이터의 정확성을 확보.

프로젝트

 술을 리뷰하고 평가할 수 있는 *spring boot* 기반의 프로젝트입니다.
실제 서비스를 하기 위한 프로젝트로 진행중인 프로젝트입니다.
총 5명의 팀으로서 진행하며, *back-end*, *devops*, *infra*를 담당합니다.

JAVA 17	SpringBoot 3.0	Spring Security	Spring Data JPA	MySql	github Action	Docker	Restdocs	Junit5	QueryDSL	AWS ECS	AWS ECR
AWS Route53	AWS S3	AWS CloudFront	Grafana	Prometheus							


bottle-note

2024-01 ~ 진행중

<https://github.com/bottle-note>



- DSL을 통한 다이나믹 쿼리 처리 DSL을 통한 다이나믹 쿼리 처리
 - JPA를 활용한 쿼리 작성 시 여러 조건을 고려해야하는 문제 발생
 - 변동 사항이 많은 검색/정렬 조건을 처리하기 위해 도입 후 팀원에게 공유 전파
- 모니터링 환경 구축
 - *spring boot actuator/ micrometer*를 활용한 모니터링 환경 구축
 - on-premise 환경에서의 *prometheus*, *grafana*를 활용한 모니터링 환경 구축
- 배포 파이프 라인 구축
 - *github action*을 활용한 CI/CD 파이프라인 구축
 - *ecs on faragate*를 활용한 *aws* 인프라 환경을 구축
- *github projects*를 도입
 - 일정, 이슈, *chart* 등 여러 정보를 관리해야하지만 비용적 문제로 인해 노션등의 비용 발생 틀을 고민
 - 개인적으로 사용해본 *github projects*를 활용하여 프로젝트 관리의 효율성 증가
- 정적분석을 활용한 소수인원의 단점 보완
 - 팀원이 각각의 개성으로 인해 코드 품질이 일정하지 않은 문제 발생
 - *Qudana*, *SonarQube*를 활용한 정적 분석과 *CheckStyle*을 통한 코드 품질 향상

 *petLink*는 반려동물 용품 구매와 동시에 펀딩에 참여할 수 있는 플랫폼을 구상하고 진행한 *spring boot* 기반 사이트 프로젝트입니다.

JAVA 17	SpringBoot 3.0	Spring Security	Spring Data JPA	MySql 8.0	Redis	QueryDSL	Naver Cloud Platform	AWS EC2	Ubuntu
Object Storage	GitHub-action	Docker	Restdocs(with openapi)	Junit5					

petLink

2023-03 ~ 2023-08

<https://f-lab-edu.github.io/petLink/>

- 신뢰성과 관리 편의성을 위한 테스트 기반의 API 명세 구현
 - *RestDocs*와 *Swagger-UI*를 활용한 테스트 기반의 신뢰성 API 명세 구현
 - **Open API Spec**을 통한 *Swagger UI*를 활용한 관리 편의성 증가
- 중복되지 않는 키를 만들면서 효율성을 극대화
 - 주문번호를 생성하는 기능 구현 중 중복을 방지하려면 DB에 접근하는 문제 발생
 - 중복되지 않으면서 다중화 환경에서도 활용가능하고 성능면에서 효율적인 **스노우플레이크** 방식 구현
- 효율적인 배포를 위한 CI/CD 구축
 - 매번 수작업으로 jar 배포해야하는 반복성 문제 발생
 - *GitHub Actions*와 *Docker*를 활용한 안정적인 PR, 자동화된 서버 배포 환경을 구축.
- 동시성 해결을 위한 Redis 분산 락 서버 구현
 - 클라이언트가 동시에 펀딩에 참여해 재고 감소 로직 수행 시 재고 동시성 문제 발생.
 - DB락, 서버 제어등 다양한 방식들과 트레이드 오프를 비교 분석 **Redis 분산락 서버 구현**을 통한 문제 해소.
- *Junit5*를 활용한 테스트 기반의 개발
 - 테스트 기반의 개발을 위해 *Junit5*를 활용한 테스트 코드 작성
 - *Mock*를 활용한 단위 테스트 코드 작성
- 효율적인 쿼리를 위한 *QueryDSL* 활용
 - JPA를 활용한 쿼리 작성 시 여러 조건을 고려해야하는 문제 발생
 - **QueryDSL**을 활용한 동적 쿼리 작성으로 효율적인 쿼리 작성

대외활동

DDD 세레나데 6기

nextstep에서 진행하는 DDD 과정
2024.04 ~ 진행중

- 레거시 프로젝트를 DDD로 점진적 리팩터링하는 경험
- 도메인에 대해 깊이 이해하고 도메인 주도 설계를 경험
- DDD에 대한 지식과 이벤트 스토밍부터 이벤트 소싱, CQRS, 코드리뷰 스킬등 다양한 개발 스킬을 경험.

TDD with Java

nextstep에서 진행하는 TDD 과정
2023.9 ~ 2023.11

- 테스트 기반으로 리팩토링을 통해 클린 코드를 경험하는 과정
- 테스트 코드의 장점과 테스트 주도 개발의 장점을 경험
- 클린코드의 필요성과 리팩토링의 방법등을 학습

F-LAB

Back-End 멘토링 과정
2023.02 ~ 2023.08

- Back-end 시니어 개발자와 함께하는 백엔드 멘토링 과정
- 실무에서 경험하는 다양한 문제 해결과정을 경험
- 내부적으로 진행되는 다양한 해커톤에 참여

클라우드 AWS 구축/운영

시나리오형 기반 클라우드 환경 활용
과정
2022.10 ~ 2022.12

- 클라우드 환경 구축 및 운영을 경험
- AWS EC2, EKS, CloudFormation, CloudWatch, IAM 등을 경험

교육

한양사이버대학교

컴퓨터공학과
2016.03 ~ 2023.03 (졸업)

- 컴퓨터공학과
- 리눅스, DB, 소프트웨어 공학 등등의 분야를 학습

KH education

스프링 웹 기반 교육과정
2022.02 ~ 2022.08

- spring 기반의 웹 개발 교육과정