

## About Me

10개월 간의 백엔드 인턴으로 앱 출시 및 레거시 코드 리팩토링  
다수의 프로젝트를 통해 테스트, 성능 개선 경험 및 클린코드 지향

PT 동아리에서 1년간 중고등학생 대상 컴퓨터공학 전공 강연에 도전  
IT 창업 동아리 멧쟁이사자처럼, SOPT 활동을 통해 사람들과 활발하게 소통 및 협업  
SAA 취득 및 클라우드 커뮤니티 AUSG 활동하며 클라우드 분야의 기술 습득

## Certificate

- AWS Certified Solutions Architect-Associate
- 정보처리기사
- SQLD
- ADSP

## Activities

- **Awards** : 졸업 프로젝트 전체 3등 우수상, SOPT 서버 파트 MVP, 장기 해커톤 APPJAM 대상(30기), 최우수상(31기)
- **IT 창업 동아리 멧쟁이사자처럼 9기** : Django 공부 내용 공유 ( [Blog](#) ) → 현재 조회수 총합 약 4000회
- **전공강연연합동아리 COM.MA 15, 16기** : 중고등학생을 대상으로 한 전공 강연 및 멘토링 활동
- **IT 창업 동아리 SOPT 30, 31기** : Haptic, Zooc 어플리케이션 백엔드 개발, 31기 운영팀 (네트워킹 행사 기획 및 운영)
- **클라우드 커뮤니티 AUSG 7기** : 빅챗 참여
- **꾸준한 스터디** : 우아한 스터디 2023 겨울 시즌 ( [Notion](#) ), 대규모 설계 시스템 ( [GitHub](#) ), 객체 지향 스터디

## Work Experience

2022.12 - 2023.10

 [다운로드](#)

### Tech

Java

SpringBoot, Spring JPA

QueryDsl

MySQL, AWS RDS

Nginx

AWS EC2, S3, CodeDeploy

### FISOLUTION

#### 연구 개발 부서 백엔드 개발자

##### 운영개선

- 시니어 개발자가 없는 상황에서 아이러빗의 백엔드 개발자 역할을 맡아 타 파트와 활발하게 소통하고 개발 경험이 거의 없는 동료들을 이끌며 주체적으로 레거시 코드를 리팩토링하는 경험
- 서버에 에러가 났을 때 고객이 직접 전화로 문의하여 해결하던 업무에 비효율성을 느껴 에러가 발생했을 때 자동으로 슬랙으로 에러 원인을 전송하여 빠른 원인 파악을 통해 30분 이상 걸리던 CS 시간을 단축 ( [BLOG](#) )
- 배포 직전 수정 사항이 많아 Github Action 활용하여 CI 구축, CodeDeploy 활용하여 CD 구축하여 기능이나 수정 사항을 빠르게 반영, nginx를 활용하여 무중단 배포환경 및 로드밸런싱 환경 구성 ( [BLOG](#) )
- 기존에 노션으로 작성하던 API 명세서를 수동으로 수정하는 과정에서 누락하는 부분이 생겨 Swagger로 변경하여 코드와 동기화된 API 문서를 생성 자동화
- 로컬 개발 환경부터 개발 서버, 운영 서버까지 local, dev, prod 세 가지 profile을 구성하여 관리, 각각의 환경에서 db와 instance를 독립적으로 운영
- 회원 탈퇴, 이용 제한 및 영구 정지 기능, 유저 차단 기능 관련 기획 회의에 직접 참여해 함께 정책을 정하고 요구사항에 맞게 개발, 관련 데이터를 @Scheduled 로 일정 기간 지나면 데이터 삭제하도록 관리해 db 비용 최소화
- 앱 삭제 후 재설치해서 실행 시 iOS에서 로그인 하지 않았음에도 푸시 알림을 받는 문제를 해결하기 위해 active 필드 활용해 해당 deviceId의 토큰 비활성화, 앱 pushNotification 동작 방식에 대한 이해도 높임


##### 클린 코드 적용

- @Data 어노테이션 제거, 불변성 유지를 위해 @Getter로 변경
- 불필요한 @OneToMany 필드 삭제, @ManyToOne 단방향 관계 지향

## Projects

2023.10 - 2024.2

 [GitHub](#)

 [배포 URL](#)

### Tech

Java, SpringBoot, Spring JPA

Docker

QueryDsl

MySQL, AWS RDS

Nginx

AWS EC2, CloudWatch

Redis

Grafana

### Sum:nail

고객이 원하는 네일샵을 빠른 시간 안에 찾도록 네일샵을 큐레이션 하는 서비스

- Repository의 JPA 의존성을 끊어내어 fake를 만들고 service 객체를 소형 테스트할 수 있게 하였고 테스트코드 유지보수 비용을 고려, 중형 테스트와 혼합 사용하여 코드 안정성 및 확장성 증가 ([BLOG](#))
- 최근 검색어를 dateTime 기준으로 정렬하여 가져오는 로직의 테스트가 어려워 의존성 역전을 통해 ClockHolder를 만들, RecentSearch를 생성할 때 TestClockHolder를 주입해 Testability 향상
- 메인 화면에서 자주 조회되는 네일샵 데이터와 히트율이 높은 해시 태그 전체 데이터를 Redis로 캐싱 적용, nGrinder로 1분간 Vuser 10명으로 설정하여 성능 테스트 결과 TPS 수치 각각 2배, 26.5배 개선 및 응답시간 각각 2배, 28.2배 개선 ([BLOG](#))
- MVP 개발 중 기획 변경이 잦아 GitHub Action 활용하여 CI 구축, Docker-Compose 활용하여 CD 구축하여 기능이나 수정 사항을 빠르게 반영 ([BLOG](#))
- 서비스 출시 일정에 맞춰 CloudWatch 활용하여 CPU/디스크/메모리 사용률 모니터링 시스템 설계 및 구축, Grafana로 시각화하고 알람 조건 설정하여 출시 이후 안정적인 서버 운영 준비 ([BLOG](#))
- 기존 Rest template로 구현되어 있는 구글 로그인을 Feign Client를 사용하여 LoC 76.5%감소 ([BLOG](#))

2023.04 - 2023.11

 [GitHub](#)

### Tech

Java, SpringBoot, Spring JPA

Docker

QueryDsl

MySQL, AWS RDS

Nginx

AWS EC2, S3, CodeDeploy

### BeachCombing

AI로 분석하는 해양환경봉사 기록 서비스

- 순차적 알림 발송 방식의 한계를 극복하기 위해 Spring Event 사용하여 비동기 처리, nGrinder로 1분간 Vuser 10명으로 설정하여 성능 테스트 결과 TPS 수치 1.5배 개선
- 사용자가 청소하려는 해변 근처에 있는지 인증하는 기능 구현을 위해 RayCasting 알고리즘 사용, 실제 사용자의 위치와 해변의 꼭짓점 좌표값을 비교하여 사용자가 해변 내부에 있는지 여부 판별
- 흩어져 있는 domain/dto 생성 로직을 모으기 위해 각 domain/dto에 정적 팩토리 메서드를 적용 ([GitHub](#))
- 반복되는 코드를 줄이기 위해 해당 도메인의 Repository에 대해 save, update, delete는 해당 도메인의 Service에서만 하도록 리팩토링하고, 예외 처리 기능을 모두 메서드로 분리 ([GitHub](#))
- 랭킹을 조회하는데 있어서 ID, 포인트 등의 파라미터를 유연하게 처리하기 위해 QueryDSL을 활용하여 다양한 요청에 동적으로 대응 정렬하고 페이징하여 필요한 정보를 반환
- GitHub Issue 템플릿 6개 세분화, 작은 PR 단위 컨벤션을 수립하여 개발할 팀 내 커뮤니케이션 개선, 깔끔한 깃 커밋 히스토리 관리를 위해 스쿼시 머지 활용

2022.10 - 2023.03

 [GitHub](#)

### Tech

Kotlin

SpringBoot, Spring JPA

MySQL, AWS RDS

AWS EC2

Redis

### 먹집사

나만의 집밥 도우미, 먹집사

- Recipe 관련, Ingredient 관련 service들간의 높은 의존성을 개선하기 위해 Facade pattern 적용해 관련 서비스 간 협력은 Facade Layer에서 하도록 개선
- 만료 시간 설정을 위해 refresh token을 redis에 저장하여 jwt 토큰에 대한 보안성 유지
- mysql db 설계 및 API 50%개발 및 카카오, 애플 로그인 api구현