

길을 찾는 개발자 최서인입니다.

(+82) 00-0000-000 | befine324@gmail.com | [inseo24 \(inseo24\) · GitHub](#)

길이 보이지 않는 것은 찾지 않기 때문일지 모릅니다. 무슨 일이든 된다는 마음과 반드시 되게 할 수 있다는 자신감을 갖고 불안한 마음을 갖지 않으려고 노력합니다. 낙관적, 긍정적으로 생각합니다.
다양한 기술을 빠르게 학습하여 개발에 적용할 수 있으며 학습에 AI 도구를 적극적으로 활용하는 편입니다.

기술

Language	Java, Kotlin, Python
Framework	Spring Boot, FastAPI
Infra	AWS(S3, SES, EKS), MySQL, Docker
Tools	Slack, Git, Bitbucket, Jira, Confluence

경력

2022년 8월 - 현재

Maxst- 선임

- 플랫폼 개발팀(2022.08 - 2023.12)
 - Java/Kotlin, Spring 기반 통합 계정 관리 및 알림 서비스 설계, 120개+ API 개발 및 운영
 - 통합 로그인 및 알림 기능을 REST API로 제공
 - [개발자 문서](#) 11개 작성 및 고객사 문의 대응
- 메타버스 개발팀(2024.1-현재)
 - Python + FastAPI 기반 디지털 트윈 서비스의 서버 설계 및 API 개발
 - Pytest 기반 Test 도입 - Unit Test 및 Integration Test 작성
 - Argo CD와 Bitbucket Pipeline을 통해 자동으로 서비스가 배포되도록 인프라 구축
 - AWS S3, RDS, IAM, VPC, EC2, EKS, ECR, Route53, ExternalDNS 등 사용

2021년 12월 - 2022년 7월

업파이 - 사원

- API 개발 및 유지보수
- Kotlin + Spring Boot 기반 모놀리식 레거시 서비스를 Microservice로 마이그레이션

프로젝트 상세

알림 서버 개발(2022.08-2023.12)

Kotlin, Spring Boot, Kafka, MySQL, AWS(SES, SNS), Firebase

- **요구사항**
 - **공통**
 - 문자, 이메일, 푸시에 대한 기본적인 알림 기능 제공
(웹, 안드로이드, iOS에 대한 서비스 제공)
 - **사내**
 - 자사 플랫폼 서비스를 위해 외부 솔루션과 연동된 Push API 필요
 - MSA 내 다른 서비스들이 이벤트 형태로 알림 요청을 전달할 수 있는 구조 필요
 - **외부**
 - SDK를 통해 외부 개발자에게 Push API 제공 -> 문서화 필요
- **역할**
 - **공통**
 - 테스트를 위해 간단한 웹, 안드로이드, iOS 프로젝트를 개발해 푸시 테스트 진행
 - **사내**
 - 다른 서비스는 Kafka를 통해 이벤트를 전송하고 알림 서버는 자신에게 온 이벤트만 받아 해당 이벤트를 비동기로 처리될 수 있도록 로직 구성
 - **외부**
 - 외부 개발자를 위한 [푸시 설정 및 API 문서](#) 작성
- **결과**
 - 사용자에게 다양한 기기, 환경에서의 일관된 서비스 경험 제공
 - 느슨한 결합으로 확장성 및 유지보수성 증가

사내 인증 서버 개발(2022.12-2023.12)

Java, Spring Boot, Keycloak, OAuth2.0, OpenID, MySQL

- **요구사항**
 - 사내 SSO 구축 목적 -> 웹 애플리케이션에서의 인증/권한 코드 없이 처리 필요
 - 커스텀을 위해 Keycloak SPI(Service Provider Interface) 구현 필요
 - 권한 부여 및 관리를 위한 RBAC 정책 수립 및 구현 필요
- **역할**
 - Keycloak REST API를 연동해 클라이언트 설정 및 로그인, 회원관리에 필요한 기능 개발
 - 토큰 커스텀을 위해 Keycloak SPI를 구현해 기능 개발
 - 각 리소스 서버의 리소스 정리 및 해당 리소스에 대한 Role 정의
- **결과**
 - 하나의 아이디로 모든 사내 시스템에 로그인 할 수 있게 되어 편의성 증가
 - SPI 구현을 통해 구현의 확장성 및 유연성 증가
- **참고:** [Keycloak 학습용 개인 레포지터리](#)

외부 인증 및 프로필 서버 개발(2022.12-2023.12)

Java, Spring Boot, Spring Authorization Server, OAuth2.0, MySQL, Neo4j, Kafka, Redis

요구사항

- 인증 서버
 - 플랫폼에 사용할 OAuth2.0 기반 소셜 로그인 서비스 개발 필요
 - 스펙에 따라 Public 클라이언트와 Confidential 클라이언트 지원 필요
 - 사용자 로그인을 통한 Authorization Code Flow 및 PKCE 대응 필요
 - 애플리케이션에서 직접 호출을 위한 Client Credential 지원 필요
 - [패스포트 로그인 SDK](#)를 통해 외부 개발자에게 인증/인가 API 제공 -> 문서화 필요
 - OpenID Connect 지원 및 로그인 세션 대신 사용 가능한 ID 토큰 제공
- 프로필 서버
 - 인증 서버가 변경될 수 있으니 그에 대응이 가능한 프로필 서버의 별도 구축 필요
 - 외부 회원 가입 프로세스 포함 프로필 관리 기능 요구
 - 구글, 카카오 로그인과 같이 콘솔 설정을 통해 OAuth2.0 로그인 지원 기능 필요

역할

- 인증 서버
 - 인증 관리에 필요한 클라이언트, 권한, 동의 등의 데이터 설계 및 개발
 - ID Token을 이용한 로그아웃 구현(RP-Initiated Logout)
 - 외부 개발자를 위한 [패스포트 로그인](#) API, 가이드 문서 작성
 - 직접적인 고객사 문의 대응
 - Discord, Discourse, 이메일 등 유입되는 문의 대응 및 외부 출장 경험
- 프로필 서버
 - 회원가입 프로세스를 포함한 회원 관리 20+ API 개발
 - 개발자 콘솔 내 애플리케이션 관련 설정(Client Type, Redirect URI 등)에 관한 데이터 설계 및 20+ API 개발
 - 여러 서비스와의 통신을 위해 직접 호출 혹은 Kafka를 통해 이벤트를 전송하는 방식의 로직 구현
 - 인터페이스로 인증 서버와의 통신을 추상화
 - 사용 중이던 인증 서버를 다른 인증 서버로 변경하는 작업 수행
 - 필요한 데이터 마이그레이션 및 변환 작업(pbkdf2 -> bcrypt) 진행

결과

- 인증 서버
 - 다양한 유즈케이스에 대한 인증/인가 지원으로 유연성 증가
 - 30건 이상의 사내/외 개발 문의 대응
- 프로필 서버
 - 외부 개발자가 콘솔과 SDK를 통해 로그인을 구현할 수 있도록 기여
 - 인증 서버 변경에도 서비스가 안정적으로 제공되도록 호환성 유지

디지털 트윈 서비스 서버 개발(2024.01-2024.04)

Python, FastAPI, MySQL, AWS(S3, SES), Poetry, Pydantic V2, Pytest

- **요구사항**
 - 회원가입을 비롯한 회원관리 서비스 API 필요
 - 폴더를 관리하기 위해 안정적으로 계층형 데이터를 다룰 수 있는 API 필요
 - 자사에서 직접 회원가입하는 경우와 소셜 로그인 서비스 모두에 대해 동일한 인증을 제공할 필요성이 증가
- **역할**
 - 회원 관리 및 폴더 관리 20+ API 개발
 - 폴더와 같은 트리 구조는 별도로 클로저 테이블을 사용하여 모든 관계를 저장
 - 폴더 생성/이동에 대해 depth 제한 정책을 적용하는 로직 추가
 - Unit Test 및 Integration Test 적용, 테스트 커버리지 70% 목표
 - 인증 미들웨어 구현 - 미들웨어 인스턴스를 의존성 주입 받아 사용자 인증 정보 검증
 - 소셜 로그인 서비스 토큰을 검증하고 서버에서 새로운 토큰을 발행하도록 구성
- **결과**
 - 일관된 인증 체계를 적용하고, 클로저 테이블을 이용한 유연성 및 확장성 증가
 - 테스트 자동화를 통한 안정적인 비즈니스 로직 제공(테스트 커버리지 77% 달성)

인프라 구축(2024.01-2024.04)

AWS(EKS, ECR, IAM, VPC, ALB, Route53, EC2, RDS), Kubernetes, ExternalDNS, Helm, ArgoCD, Bitbucket Pipeline, GitVersion

- **요구사항**
 - 프론트 및 서버 배포 목적의 EKS Cluster 구축 필요
 - PR이 반영됨과 동시에 CI/CD가 자동으로 이뤄질 필요성 증가
 - 배포 후 명령어 기반이 아닌 UI로 관리 필요성
- **역할**
 - AWS EKS 클러스터 구축(VPC, IAM, RDS 등 관련 AWS Service 설정)
 - ExternalDNS 셋업을 통해 동적으로 Route53에 레코드가 관리되도록 적용
 - 프론트 및 서버 도커라이징 - 이미지 축소를 위해 멀티 스테이지 빌드 적용
 - Nextjs, FastAPI 앱 모두 약 70~77% 감소(500MB -> 90MB)
 - ArgoCD 설정 및 Helm Chart가 포함된 Repository와 ArgoCD 연동
 - PR이 반영되면 Pipeline을 통해 배포되도록 구성, 시멘틱 버저닝을 통해 빌드 버전 관리
 - 변경된 태그를 ArgoCD에서 감지하여 자동으로 해당 이미지를 이용해 배포
- **결과**
 - 배포 자동화로 각 담당자가 필요할 때마다 즉시 배포 가능
 - 모든 리소스를 한 Repository에서 선언적으로 관리하여 유지보수성 및 확장성 증가
 - ExternalDNS 설정으로 동적으로 레코드 등록하며 vendor 종속성 줄임