원티드

Server Platform Engineer.

높은 퍼포먼스의 유연한 소프트웨어를 개발하는 엔지니어입니다 👨🏻‍💻

압도적인 결과 산출을 위해 끊임없이 고민하고 개선하는 자세가 제가 갖추어 나가야할 스탠스라고 생각합니다. 서버 퍼포먼스 향상을 위해 기여했으며 **TPS 300%향상 JVM Warm UP, 캐시, 인덱싱을 통한 레이턴시 60~80% 단축 그리고 고가용성을 위한 마이크로서비스 전환**등의 활동으로 기여했습니다.

배움과 성장은 함께, 조직의 성장을 적극 도모합니다 💪

조직의 성장을 위해 항상 지식을 공유하는 시간을 가집니다. 높은 퍼포먼스를 위한 조직내의 역량을 강화하기 위하여 극도의 투명함과 신뢰의 문화를 추구하고 협업하며 교육청과 공동사업을 진행하며 **+3,000명의 유저**를 보유한 서비스를 개발/운영, **200명의 DAU**를 보유한 서비스를 운영중입니다.

About

0000. 00. 00

email@gmail.com

https://twitter.com/hellowanted

https://github.com/hellowanted

Skills

나의 업무 스타일과 강점, 스킬을 요약하며 포트폴리오를 시작해 보세요.

- Java, Kotlin, Spring Frameworks
- MySQL, Redis, MongoDB
- AWS, Kubernetes, Prometheus, Grafana, Loki, Kafka, Terraform 외 여러 오픈소스

NNN 서비스

2021. 03 ~ present

GitHub ↗ Site ↗ Docs ↗

어떤 프로젝트였는지
이미지를 보여주면 인사 담당자가
이해하기 쉬워요!

개요 📖

G 기숙사 고등학교의 교내 기숙사 관리 서비스입니다. 2222년도부터 운영된 서비스 입니다. NNN명의 DAU를 보유하고 있으며 서비스 전체 사용 유저(학생, 졸업생, 선생님) 수 NNN+ 이상을 보유하고 있고 매년 약 NNN명 신규 유저가 생기는 서비스입니다. 도토리는 앞으로 전국 단위의 기숙사 관리 SaaS 플랫폼이라는 목적을 달성하기 위하여 계속해서 발전하고 있습니다.

역할 🙋🏻‍♂️

- V2 - 서버 엔지니어로서 RestfulAPI 개발, 성능 개선, 모니터링 환경 구축, CI/CD
- V3 - 마이크로서비스아키텍처 전환(설계, 구축, 개발), 마이크로서비스 통합 환경 모니터링, RESTful / gRPC API 개발, 배치 애플리케이션 개발

Skills 🛠️

- Java, Kotlin, Spring Boot, Spring Batch, Spring Webflux, Spring Cloud
- AWS, Kubernetes, Kafka, Prometheus, Grafana, Loki, Helm Charts, ArgoCD
- MySQL, Redis

애플리케이션 개발 (V2,V3) 💻

서비스 애플리케이션 API 개발

예약 시스템, 음악 신청 시스템, 유저 정보 관리 시스템의 서비스를 개발했습니다. REST/gRPC 기반의 API를 개발했습니다. 각 마이크로서비스의 요구사항에 맞추어 MVC 기반의 동기 Blocking API와 순간적으로 많은 트래픽이 발생하는 서비스(예약 시스템, 서버간 통신 시스템)들은 **WebFlux, Coroutine** 기반의 비동기 Non-Blocking 기술로 처리량을 향상시켰습니다. **TPS 300%향상**

Gateway 미들웨어 개발 GitHub ↗ Docs ↗

자세한 내용이 궁금할 때 눌러볼 수 있는 링크를 넣어두는 것도 좋아요.

프로젝트 전/후 비교

프로젝트 진행 전 어떤 문제가 있었는지, 이를 어떻게 해결했는지 보여주기 좋은 구조예요.

AS-IS

모든 서비스들의 요청을 한 url에서 endpoint로 구분하기 위해 게이트웨이 서버를 개발했으며, 서비스 요청의 프록시 역할을 하는 gateway 서버에 jwt기반 인증 시스템에서 request한 유저의 id, 권한값으로 유저를 식별하기 위한 시스템이 필요했습니다.

TO-BE

요청으로 함께 들어온 jwt 값을 파싱하여 header에 담아 포워딩하도록하는 **미들웨어를 구현**했습니다. 이후 gateway를 거친 다른 서비스들도 header 정보를 통해 유저 정보를 식별할 수 있게 되었습니다.

분산 트랜잭션 관리, 이벤트 소싱

AS-IS

각각의 마이크로서비스간에 모든 서비스가 직접적으로 통신하는 구조의 애플리케이션 구성을 가지고 있어, 서비스간 **강결합**이 발생하게 되었습니다.

TO-BE

이러한 문제를 해결하기 위하여 **이벤트 소싱**을 통해 데이터를 처리하도록 개선했습니다. 메세지 큐 서비스로 데이터의 안정성과 높은 처리량을 위하여 Kafka를 채택했습니다. Kafka와 spring actuator를 사용하여 실시간 config 정보 업데이트와 분산 트랜잭션 관리를 통해 데이터 일관성을 보장하도록 구현했습니다.

슬로우 쿼리 개선, Indexing

Docs ↗

AS-IS

도토리 서비스의 페이징이 적용된 여러 부분의 페이징 검색 쿼리들이 매우 늦은 처리시간을 가진 슬로우 쿼리인 것을 확인했습니다. (유저 정보 리스트, 공지사항)

TO-BE

발견한 슬로우 쿼리(페이징 조회 부분)를 **커버링 인덱스, 프로젝션**을 통해 성능을 개선할 수 있었습니다. 적용 후 레이턴시 **794ms → 211ms로 약 73% 개선되었습니다.**

예약 시스템 동시성 문제 해결

Docs ↗

AS-IS

심야 자습, 안마의자 예약 시스템에서 최대 인원의 수를 초과하게 되는 동시성 문제들이 발생했습니다.

TO-BE

In-memory 저장소인 Redis와 RDB 중에서 처리량을 더욱 향상시키기 위해 더 빠른 Redis를 채택해 구현하는 것이 데이터의 양과 고려했을 때 최적이라고 판단되었습니다. Redis Client Redission을 통해 분산락을 구현했으며 TTL 설정을 통해 일정 시간내에 락을 처리하지 못하면 실패되도록 구현했으며, 데이터 일관성을 위해 트랜잭션이 커밋된 이후 락을 반납하도록 구현했습니다.

효율적인 Redis 활용을 위한 메모리 관리

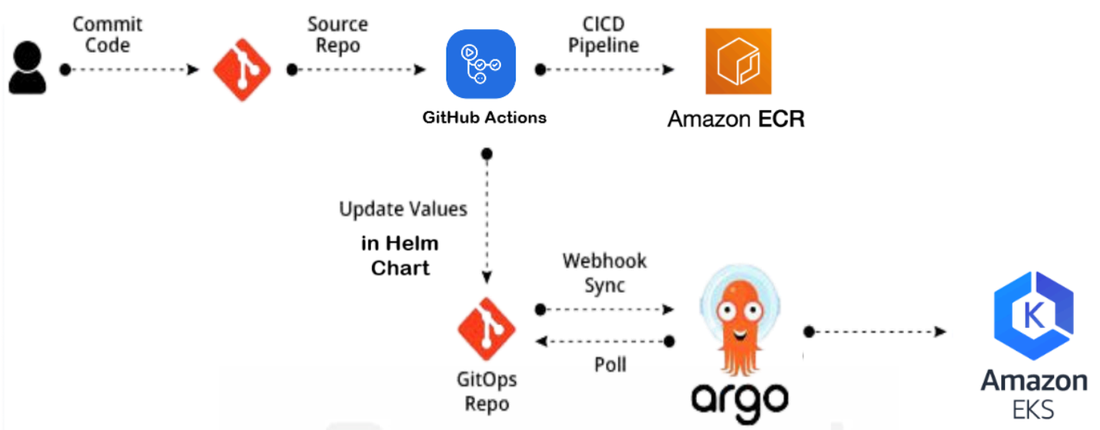
Docs ↗

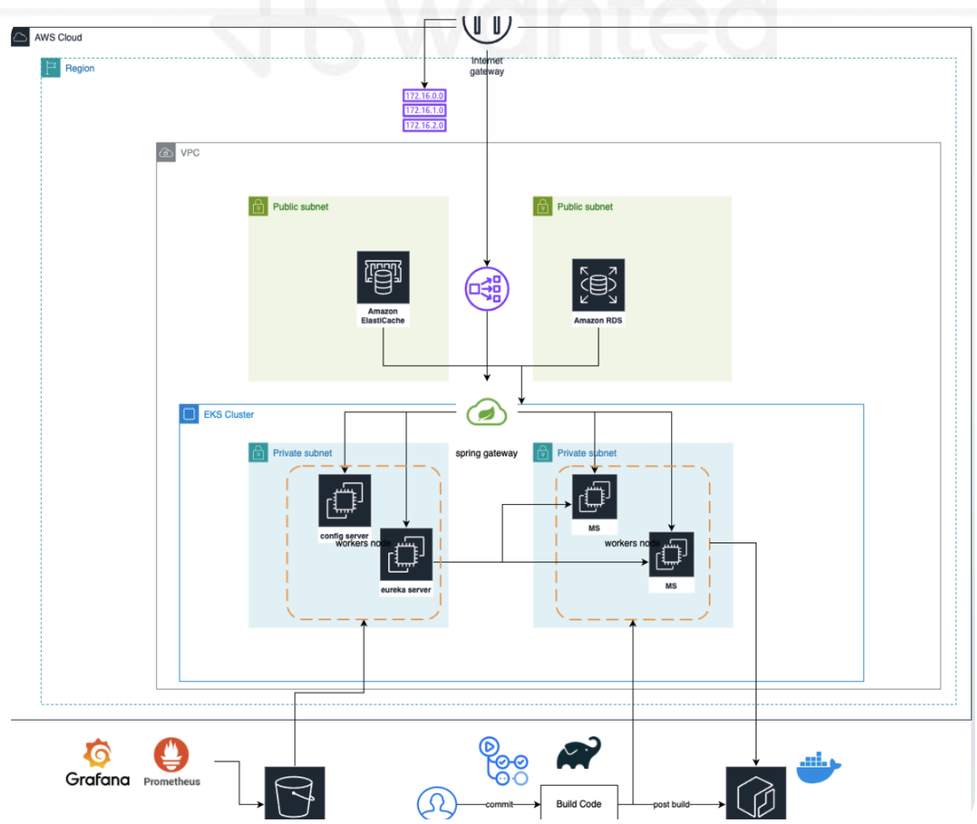
사전에 Redis 환경을 더욱 최적화하여 처리량을 개선하기 위해 여러가지 옵션들을 설정했습니다. Redis의 max-memory를 설정하여 운영체제의 가상 메모리인 swap 영역까지 사용하는 것을 방지했으며, cache eviction 정책을 고려하여 적용된 애플리케이션 특성에 맞는 **LFU 알고리즘**을 적용하여 캐시 메모리 성능을 개선했습니다.

마이크로서비스 전환(V3) 🙌

EKS 환경 구축 GitHub ↗ Docs ↗

도토리 프로젝트의 고가용성 확보와 안정적인 운영을 위하여 마이크로서비스로 서비스를 전환 했습니다. K8s 리소스를 **Helm Charts로 템플릿화** 하여 중복을 제거하였으며 **ArgoCD**로 각 애플리케이션들을 클러스터로 배포합니다.



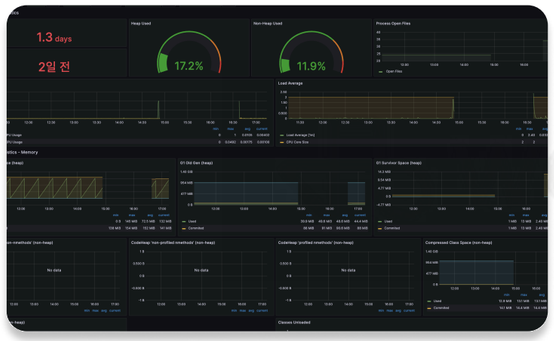


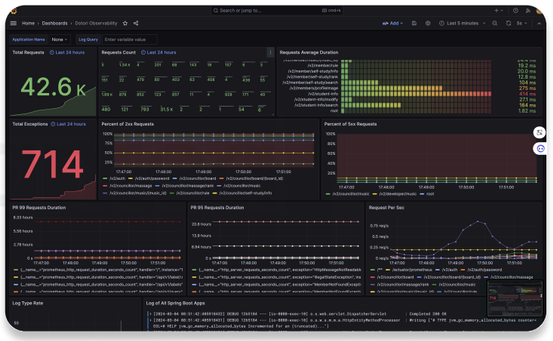
Node 가용성 문제 개선 Docs ↗

변동되는 프로젝트 수에 맞추어 자유롭게 Node를 스케일 인/아웃 할 수 있도록 구성해야 했습니다. CA는 ASG의 Desired 사이즈를 조정하여 노드를 프로비저닝하기 때문에, **스케일 인/아웃 반응속도가 느리다는 문제점이 존재했습니다. (downtime 발생)** 그리고 이러한 문제를 해결하기위해 AWS 리소스 의존성없이 **JIT 배포가 가능한 Karpenter를 입하여 문제를 해결**했습니다. Desired 사이즈를 조정하지 않고 신속하게 노드가 생성됩니다.

모니터링

모니터링 스택으로 다양한 메트릭, 대시보드 커스텀의 이점등을 고려하여 오픈소스 모니터링툴 Grafana를 채택, Loki와 Prometheus 그리고 Spring Actuator, Promtail과 함께 적용하여 모니터링합니다.





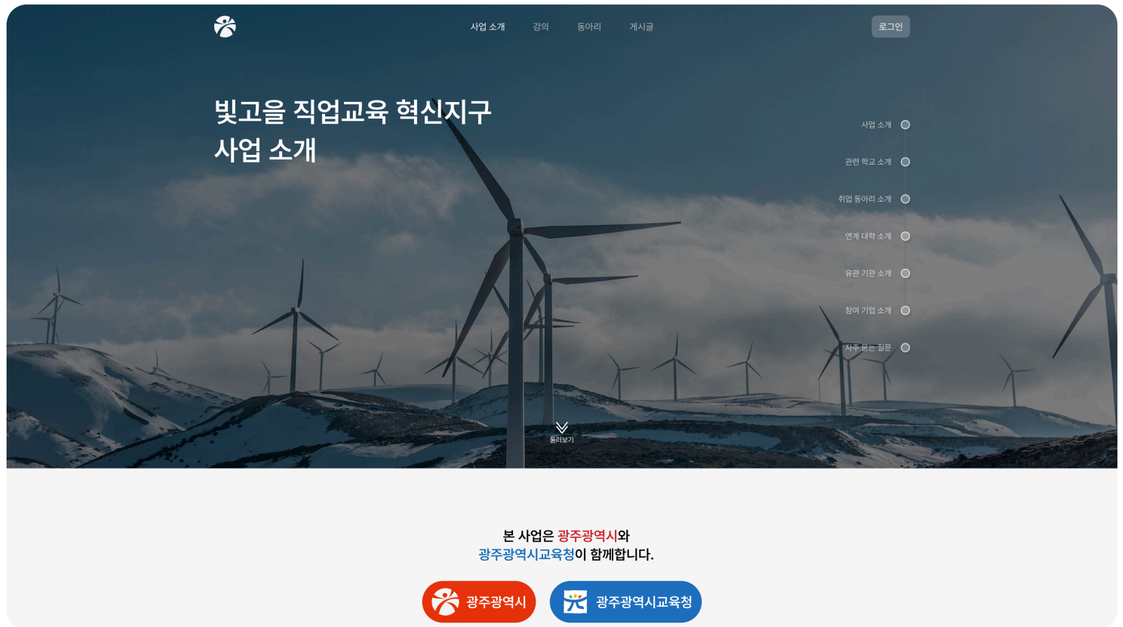
회고

마이크로서비스로 전환하면서 신경써야할 점이 매우 많다는 것을 느꼈습니다. 여러 오픈소스들을 적용하면서 가파른 러닝커브와 팀원들과 협업하면서 관리 방법을 인수인계 하는 과정에서 커뮤니케이션과 문서화가 매우 중요하다고 느끼게 되었습니다. 극도의 투명함의 문화를 추구하게 된 계기이자 여러 새로운 기술들을 도입해 볼 수 있어 매우 즐거웠고, 좋은 경험이었습니다. 또한 실시간으로 프로젝트를 운영하기에 여러 문의 사항들과 이슈 제보가 끊임없이 나타났기 때문에 유연한 소프트웨어 개발을 추구하게 된 계기가 있었고 다양한 문제 상황들을 겪고 해결할 수 있었던 프로젝트였습니다.

NNN NNNN NNNN

2023. 09 ~ 2024. 03

GitHub ↗ Site ↗



개요 📰

G고등학교 학교기업 동아리와 교육청과 함께 진행한 취업 동아리 활동관리, 지원 서비스입니다. NN개교의 특성화 고등학교와 N개교의 대학, NN명의 대학교수, 멘토, 광주 지역 기업체 강사들이 해당 서비스를 사용하며 총 유저 NNNN+명을 보유한 서비스입니다.

역할 🙋🏻‍♂️

- ✦ 서버 엔지니어로서 애플리케이션 I/O 레이턴시 최적화, 쿼리 최적화, 배치 애플리케이션 개발, RESTful API 개발, 멀티모듈 아키텍처 설계, 구성
- ✦ 인프라 엔지니어로서 AWS 아키텍처 설계, GitHub Actions CI/CD 구축, Terraform을 통한 인프라 스펙 선언, ELK 기반 모니터링, 비용 절감을 위한 인스턴스 매니저 개발

Skills 🛠️

- ✦ Kotlin, Spring Boot, Spring Batch, Shell Script
- ✦ AWS, Docker, Terraform, GitHub Actions, Datadog
- ✦ MySQL, Redis, Tibero

Persistable, 쿼리 성능 최적화 ⚙️

GitHub ↗

문제

spring data jpa 사용시, insert, update 작업시 **추가로 select 쿼리문이 발생**합니다. 발생하는 이유는 jpa repository save 내부 구현에서 persist()가 아닌 merge()가 실행됩니다.AbstractEntityInformation.isNew()를 통해 확인하는 내부 로직에서 PK가 nullable 하다는 다소 애매모호한 조건 검사도 함께 진행됩니다.

```
Hibernate:
    select
        entity.id as id1_3_0_,
        entity as name2_3_0_
    from
        entity entity
    where
        entity.id=?
Hibernate:
    insert
    into
        entity
        (name, id)
    values
```

해결

모든 엔티티가 상속받는 BasePkEntity 클래스를 생성했으며 해당 베이스 엔티티는 Persistable이라는 인터페이스를 상속합니다. pk의 타입은 UUID이며 isNew()함수를 재정의 하여 엔티티의 생성 여부를 판별합니다. +추가적으로 hibernate proxy 객체일때의 엔티티의 동등성 검사를 위하여 equals를 오버라이드 했습니다. 이후 **추가 select 쿼리는 발생하지 않았습니다.**

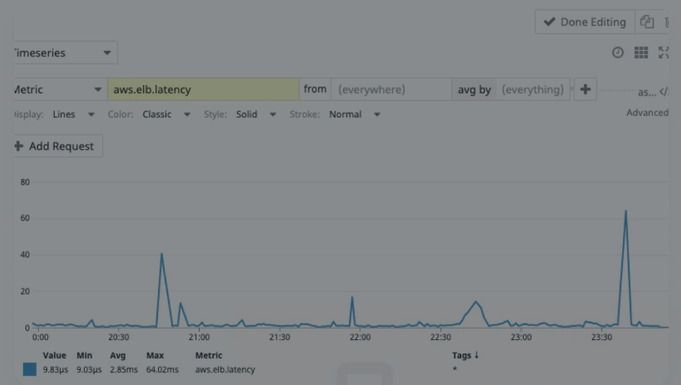
```kotlin
@MappedSuperclass
abstract class BaseUUIDEntity(

    @Id
    @GeneratedValue(generator = "uuid2")
    @GenericGenerator(name = "uuid2", strategy = "uuid2")
    @Column(columnDefinition = "BINARY(16)", nullable = false)
    @get:JvmName(name = "getIdentifier")
    open var id: UUID = UUID(0, 0)
) : BaseTimeEntity(), Persistable<UUID> {

    @Column(name = "ulid", updatable = false, unique = true)
    var ulid: String? = ULIDGenerator.generateULID()

    override fun isNew(): Boolean = (id == UUID(0,0)).also { new ->
        if(new) id = UUID.randomUUID()
    }

    override fun getId(): UUID? = id
```

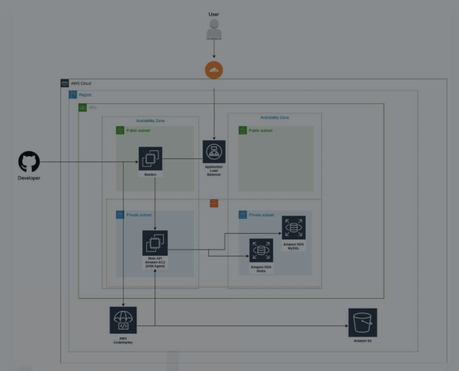
```kotlin
override fun equals(other: Any?): Boolean {
    if (other == null) {
        return false
    }

    if (other !is HibernateProxy && this::class != other::class) {
        return false
    }

    return id == if (other is HibernateProxy) {
        other.hibernateLazyInitializer.identifier
    } else {
        (other as BaseUUIDEntity).id
    }
}

override fun hashCode() = Objects.hashCode(id)
```

인프라 엔지니어로서 기여 🚧

서버 인프라 구성

프로젝트 특성상 교육청 서버를 사용합니다. 그렇기에 애플리케이션 테스트를 위한 개발 서버를 설계했습니다. AWS를 사용하여 애플리케이션을 배포했으며 CodeDeploy, GitHub Actions를 기반으로 CI/CD를 적용하였으며. **datadog를 통해 메트릭을 모니터링**했습니다.





Terraform 인프라 스펙 선언 GitHub ↗

IaC 기술을 사용한 유지보수 측면에서의 이점을 고려하여 Terraform을 선택해 프로젝트의 **AWS 인프라 스펙을 선언** 했습니다.

AWS 인스턴스 매니저 개발 GitHub ↗

AWS CLI 기반 인스턴스 중지 매니저를 Shell Script로 개발했습니다. 운영 서버는 교육청 온프레미스 환경에 배포되기에, 테스트용 개발서버만 AWS로 구성하였습니다. 그렇기에 비용 절감을 위하여 새벽에는 인스턴스를 작동하지 않도록 개발했습니다.이 후 새벽에는 테스트가 진행되지 않는 개발 서버에 적용하여 서버 비용을 **38% 절감**했습니다.

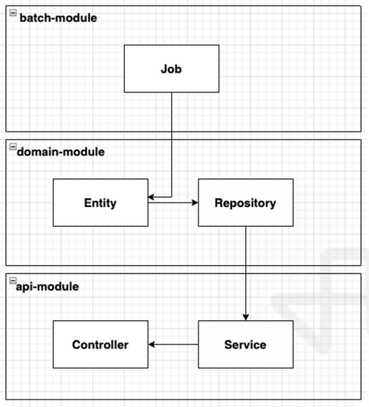
성과는 데이터 기반으로

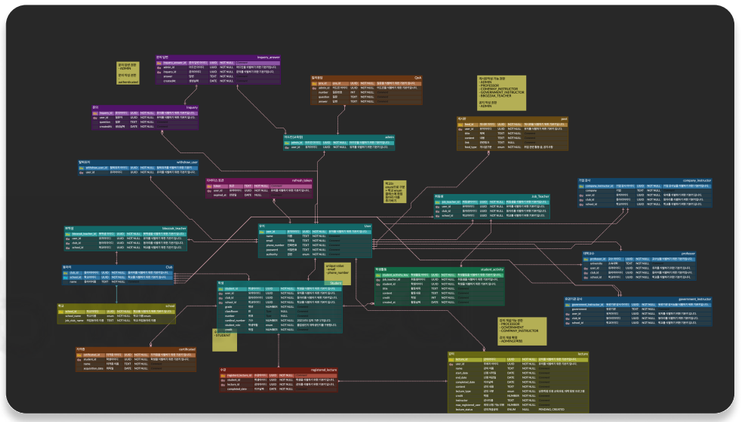
프로젝트 배경, 목적, 담당 업무,
성과를 데이터 기반으로 요약하세요!

프로젝트 아키텍처 설계 🖇️

모듈화/ERD

직관적이고 빠른 개발 속도를 위해서 레이어드 아키텍처 구성으로 아키텍처를 설계했습니다. [Presentation | Application | Infrastructure | Persistence] 계층으로 이루어지는 아키텍처를 설계하면서 각 모듈별 의존성을 낮추고 Controller - Service - Repository 형식의 API 모듈과 Batch Application의 모듈이 같은 Entity를 사용한다는 점에서 패키지별 상호 의존성들을 낮추기 위하여 모듈화했습니다. **batch-module, api-module 그리고 domain-module로 분리된 아키텍처 구성도**를 가집니다.





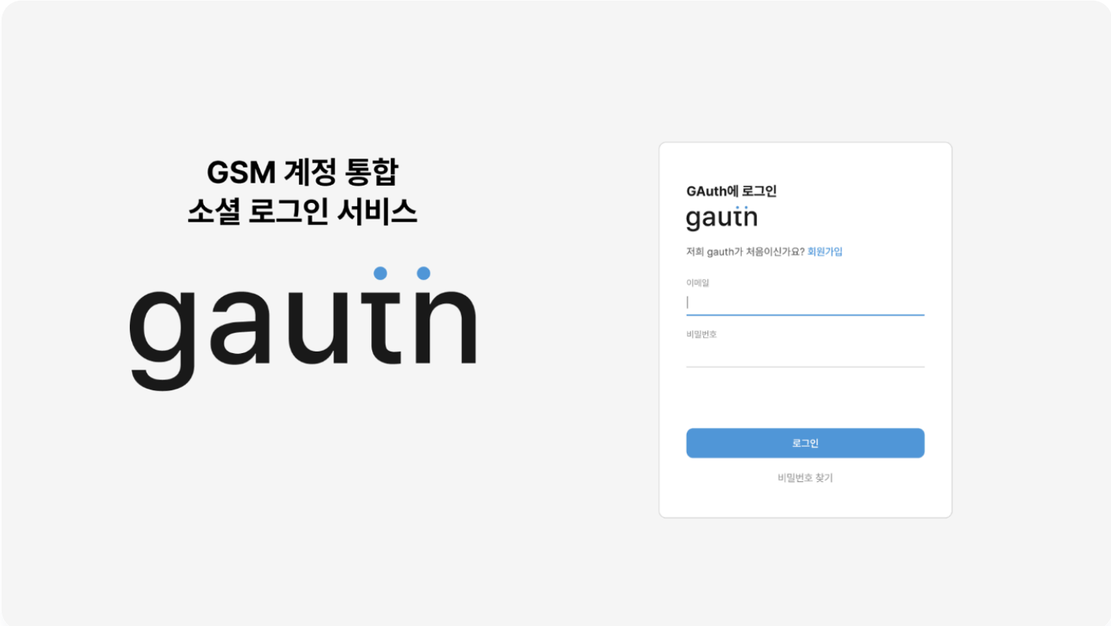
회고

0부터 시작하면서 NNN측도 처음 소프트웨어 관련 프로젝트를 함께 하는지라 부족한 문서화와 전체 팀원들의 요구사항 파악이 부족했던 점, 커뮤니케이션의 어려움등으로 많은 경험을 한 프로젝트 였습니다. 그러나 저희 개발팀 내에서는 데일리 스크럼, 애자일, MVP, 스프린트 등의 방식의 프로젝트 협업을 통해 부족한 시간 대비 꽤나 거대한 프로젝트였음에도 성공적으로 배포를 완료했고 NNNN만원 지원을 받았던 대규모 프로젝트였습니다. 모르는 사람, 다른 직군의 사람들과 협업을 하면서 여러 시야에서 프로덕트를 바라볼 수 있게 되었습니다.

GAuth

2022.07 ~ present

GitHub ↗ Site ↗



개요 📰

G고등학교 교내 계정 통합 소셜 로그인 서비스입니다. 교내 계정 관리, OAuth OpenAPI, SDK, OAuth Project API를 제공중인 서비스입니다. 현재 저희 서비스는 NNN명 이상의 학생, 선생님, 졸업생 유저를 보유하고있고 매년 70명의 유저가 늘어나고있습니다.

역할 🙋🏻

- ✦ OAuth 인증/인가 플로우 설계, 개발
- ✦ 애플리케이션 API, Open API 관련 API 설계/개발, 성능 개선
- ✦ JVM 환경에서 간단하게 GAuth를 사용할 수 있게 도와주는 SDK 개발

Skills ⚒️

- ✦ Kotlin, Spring Boot
- ✦ AWS, Docker, Terraform, GitHub Actions
- ✦ MySQL, Redis

OAuth 플로우 설계/개발 ⚒️

OAuth 서비스를 사용하여 다른 교내 프로젝트내에서 GAuth 계정을 이용하기 위해 제공되어야할 인증/인가 플로우를 구상하고 그에 맞추어 Open API, SDK를 개발했습니다.

1. https://gauth.co.kr/login?client_id=(클라ID)&redirect_uri=(리다이렉트URI)에 유저가 접속한다.
   → GAuth 로그인 페이지

2. GAuth 사용하는 서비스를 통해, 로그인 성공시 (redirectURI)?code=(코드) 로 리다이렉트된다.

3. 리다이렉트된곳(클라이언트)에서 받은 코드를 사용해 https://server.gauth.co.kr/user에 POST method로 body에 code, clientId, clientSecret을 담아서 보내면 유저정보를 받을 수 있다

GAuth SDK 개발

GitHub ↗

GAuth Open API를 사용하여 인증/인가 작업을 진행하는 코드를 작성하는 것이 매우 복잡하여 JVM 환경의 언어에서 간단하게 개발할 수 있게 도와주는 SDK를 Kotlin으로 개발했습니다.

회고

프로젝트 도메인을 분석하면서 OAuth라는 도메인을 분석하기 위해 색다른 방식의 생각과 고민들을 했습니다. 도메인 특성상 보안적인 요소가 매우 중요했으며 유연하게 학생 정보를 동기화하기 위해 어떤 솔루션을 만들 수 있을까 하며 여러가지 방법을 시도해보고 팀원들과 회의해봤던 시간이였고, 실시간으로 고객의 불편함과 이슈에 대응하여 빠르게 개선하고 변화하는 소프트웨어를 개발해볼 수 있었던 좋은 경험이였습니다.

Other Projects

Hi - HomeBase Interface

2022. 08 ~ present | Server Developer

G고등학교 교내 학습 공간 홈베이스 예약, 관리 서비스
Contribute: 서버 API 개발, 동시성 핸들링, 헥사고날 아키텍처 설계, N + 1 해결
현재까지 지속적으로 서비스 운영/개선중 **DAU 80**

선도 - 선생님 도우미 서비스

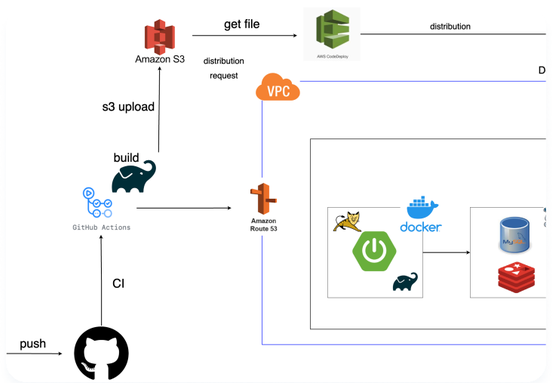
2023. 09 ~ 2024. 04 | Server Developer

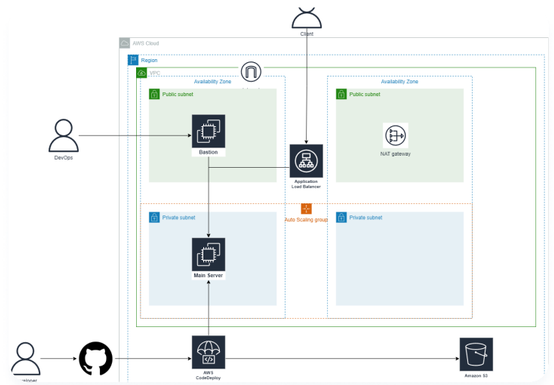
G고등학교 교과, 비교과, 담임선생님 도우미 서비스
Contribute: wee클래스 1:1 채팅 기능 개발, Java WebFlux - NonBlocking API 개발
http api test 작성, Docker, AWS EC2 모놀리식 서비스 배포, 운영

Go API - GSM Open API

2022. 11 ~ 2023. 03 | Server Developer

G고등학교 교내 정보 Open API 서비스
Contribute: api key 발급, 인증/인가 기능 개발, 교내 정보 Open API 개발
AWS EC2 배포, Code Deploy + GitHub Action CI/CD 구축





Others 🏝️

estudy

2022. 06 ~ present

지식을 습득하고, 메타인지 활성화를 위하여 꾸준히 기록장을 작성중입니다.
서버, 백엔드 그리고 더 넓게 나아가 소프트웨어 학문 그 자체를 학습하고 기록합니다.
GitHub Star +200

rks-java

2024. 04 ~ present

CLI 환경에서 Redis의 주요 사용 통계를 분석하고 모니터링하기위해서 개발한 CLI Tool

Open Sources.

오픈소스 프로젝트 개선을 위해 기여한 경험이 있습니다.

경험을 기반으로
나의 역량을 어필해요.

- 2023.09.19-spring-kafka: sample 결과 출력 문서 개선
- 2023.09.21-spring-framework: webflux 확장함수 null값 처리 개선
- 2023.09.23-spring-boot: Actuator HealthMetric 조건 검사 개선
- 2023.09.23-armeria: Type 클래스 확장 함수명 네이밍 개선
- 2024.01.06-Exposed: SQL, update Statement 개선
- 2024.01.19-http-auth-parser: 문서 예제 오류 fix
- 2024.01.30-kotlin-jdsl: 산술 연산자 CEILING 스펙 정의, 테스트 코드 작성
- 2024.03.08-cglib: Spring DI, Enhancer private method 관련 이슈 트래킹

Certificated

- 정보처리기능사 - NNNN 정기기능사 제 N회
- SQL 개발자 - NNNN 제 N회

Awards

- (2222) NNN 역량 인증대회 우수상 - G학교
- (2223) 제 NN회 NN 해커톤 우수상
- (2224) NN 연합 해커톤 AWS 기업상
- (2222) 삼성 창업놀이터 70만원 투자 유치

Presentation

Youtube ↗

컨퍼런스 스피커로 세 번 발표해본 경험이 있습니다. KotlinConf'2024 in South Korea 에서 NNNNNN 주제로 발표했으며, NNN Concerence 1, 2회 컨퍼런스에서도 기술, 성장에 관련한 주제로 발표를 진행했습니다.

- 2023 제 1회 NNN 컨퍼런스 스피커 활동 (Spring Boot 근데 코틀린을 곁들인)
- 2023 NNN Conference 스피커 활동 (개발자로서 미친 성장을 이루어내는 방법)
- KotlinConf'24 Global in South Korea 스피커 활동 (NNNNNNNNNNNN)

Thank You