

# Backend Developer Portfolio

## 소개

저는 토이 프로젝트를 통해 개발 역량을 꾸준히 쌓아가는 백엔드 개발자 안효성입니다.

실무에서 접한 기술들을 단순히 사용하는 것에 그치지 않고, 직접 설계하고 구현해보며 깊이 있는 이해를 추구합니다. MSA 운영 경험을 바탕으로 제로베이스부터 시스템을 구축해보고, 나아가 대규모 트래픽을 처리하는 시스템 아키텍처까지 시야를 넓혀가고 있습니다.

## 프로젝트 목록

#	프로젝트	핵심 키워드	상태
1	spike-order	시스템 아키텍처, 대량 트래픽, CQRS, SAGA	🛠 개발 진행중
2	msa-order-claim	MSA, Outbox 패턴, Kafka, 분산 추적	✓ 완료
3	ASAP	OAuth2, 게임피케이션, 무중단 배포	✓ 완료
4	ASAP-api	JWT, Stateless 인증, RESTful API	✓ 완료

## 1. spike-order

MSA를 넘어 시스템 아키텍처 전체의 큰 그림을 보기 위한 토이 프로젝트

### ★ 프로젝트 개요

배달앱에서 한 가게에 이벤트로 인해 주문이 대량으로 몰리는 상황을 가정하고, 초당 수천 건의 주문을 안정적으로 처리할 수 있는 시스템 아키텍처를 설계했습니다.

### ● 기술적 핵심 포인트

영역	핵심 내용
핫패스 최소화	유저 응답에 필수적인 것만 동기 처리, Outbox 패턴으로 비동기 이벤트 발행
CQRS	읽기/쓰기 분리, Replica DB + Elasticsearch + Redis 캐시 조합
SAGA 패턴	분산 환경에서 보상 트랜잭션을 통한 데이터 일관성 보장
장애 격리	Circuit Breaker, Timeout, Fallback으로 외부 시스템 장애 전파 방지
실시간 처리	Apache Flink + RocksDB State Backend로 스트림 파이프라인 구성
캐시 전략	Singleflight 패턴으로 Cache Stampede 방지, Redis Pub/Sub 캐시 무효화

### ❖ 기술 스택

Spring Boot Spring Data JPA Kafka Redis Elasticsearch Flink Resilience4j Eureka Keycloak

### ❖ 상세 정보

- GitHub: [spike-order Repository](#)

## 2. msa-order-claim

| 운영만 해왔던 **MSA** 구조를 제로베이스부터 직접 구축해보고 싶어 시작한 토이 프로젝트

### ❖ 프로젝트 개요

오픈마켓 개발 경험을 떠올려 해당 도메인을 접목하여, 주문 생성 → 재고 확인 → 결제 처리 → 주문 승인/취소가 이벤트 기반으로 비동기 처리되는 마이크로서비스 시스템을 구축했습니다.

### ● 기술적 핵심 포인트

영역	핵심 내용
<b>Transactional Outbox 패턴</b>	DB 트랜잭션과 이벤트 발행의 원자성 보장, At-least-once 전달
<b>Saga 패턴 (Choreography)</b>	서비스 간 이벤트 기반 분산 트랜잭션 관리
멱등성 보장	ProcessedMessage 테이블로 중복 메시지 처리 방지
분산 추적	Zipkin + Micrometer Tracing으로 서비스 간 요청 흐름 추적
동시성 제어	FOR UPDATE SKIP LOCKED 로 Outbox 폴링 시 동시성 이슈 해결
서비스 디스커버리	Eureka 기반 동적 서비스 등록/조회

### ▣ 모듈 구성

```
msa-order-claim/
├── api-gateway/          # API 게이트웨이 (Circuit Breaker)
├── discovery/            # Eureka 서버
├── config-server/         # 중앙 집중식 설정 관리
├── order-service/         # 주문 서비스
├── inventory-service/    # 재고 서비스
├── payment-service/       # 결제 서비스
└── outbox-relay/          # Outbox → Kafka 이벤트 릴레이
└── common-lib/             # 공통 라이브러리
```

### ❖ 기술 스택

Java 21 Spring Boot 3.3 Spring Cloud Kafka PostgreSQL Resilience4j Eureka Zipkin

### ❖ 상세 정보

- GitHub: [msa-order-claim Repository](#)

## 3. ASAP

| 동기부여를 줄 수 있는 재미난 서비스가 뭐가 있을까 고민하다 만들게 된 프로젝트

### ❖ 프로젝트 개요

사용자가 하루 두 번(오전/오후) 'Took'을 던지며 미션을 수행하고, 성공 시 티어 포인트를 획득하여 랭킹에 도전하는 게임미피케이션 기반 습관 형성 서비스입니다.

### ● 기술적 핵심 포인트

영역	핵심 내용
<b>OAuth2 소셜 로그인</b>	Google, Naver, Kakao 통합 인증, Provider별 사용자 정보 매핑

게이미피케이션	7단계 티어 시스템(Bronze~Master), 실시간 랭킹, 포인트 보상/패널티
<b>QueryDSL</b>	타입 안전 동적 쿼리, 서브쿼리를 활용한 복잡한 업데이트 처리
복합 키 설계	@EmbeddedId로 email + registrationId 복합키 구현
스케줄 기반 정산	Spring Scheduler로 미션 성공/실패 자동 판정
무중단 배포	NGINX + Blue-Green 배포, 포트 스위칭 자동화

## ▣ 미션 시스템

미션	기간	성공 보상	실패 패널티
Tick 1Day	1일	+10 pt	-10 × 티어배수
Tick 3Day	3일	+60 pt	-60 × 티어배수
Quick 5Day	5일	+150 pt	-150 × 티어배수
Quick 7Day	7일	+280 pt	-280 × 티어배수

## ❖ 기술 스택

Java 11 Spring Boot 2.1 Spring Security OAuth2 QueryDSL JPA MariaDB NGINX Jenkins AWS

## ❖ 상세 정보

- GitHub: [ASAP Repository](#)

## 4. ASAP-api

ASAP 서비스의 백엔드와 프론트엔드를 분리하여 백엔드 코드만 구성한 프로젝트

### ❖ 프로젝트 개요

기존 ASAP의 세션 기반 인증을 **JWT** 기반 **Stateless** 인증으로 전환하여, 웹/모바일 앱/외부 서비스 등 다양한 클라이언트에서 통합된 API를 사용할 수 있도록 설계했습니다.

### ● 기술적 핵심 포인트

영역	핵심 내용
<b>JWT</b> 인증	HS256 서명 알고리즘, 토큰 만료 처리 (발급 후 10분)
<b>Vendor</b> 기반 인증	클라이언트(업체)별 SecretKey 관리로 API 접근 제어
<b>Spring Security Filter</b>	JWTAuthorizationFilter로 토큰 검증 파이프라인 구현
<b>Stateless</b> 설계	서버에 세션 저장 없이 토큰만으로 인증 처리
<b>Swagger JWT</b> 지원	API 문서에서 JWT 인증 테스트 가능

## ▣ JWT 검증 플로우

- 헤더 스키마 체크 (Bearer 토큰 확인)
- JWT Payload에서 업체 코드(jti) 추출
- 업체 유효성 검증 (Vendor Enum)
- SecretKey로 JWT 서명 검증
- 토큰 유효 기간 검증

## ■ ASAP vs ASAP-api 비교

구분	ASAP (v1)	ASAP-api (v2)
인증 방식	OAuth2 + Session	JWT Token
상태 관리	Stateful (세션)	Stateless (토큰)
클라이언트	웹 브라우저	웹, 모바일, 외부 서비스
템플릿 엔진	Mustache	없음 (순수 API)

## ▣ 기술 스택

Java 11 Spring Boot 2.1 Spring Security jjwt JPA MariaDB NGINX Travis CI AWS

## ◊ 상세 정보

- GitHub: [ASAP-api Repository](#)

## ▣ 기술 키워드 종합

### Architecture & Patterns

MSA CQRS SAGA Outbox Pattern Event-Driven Blue-Green Deployment

### Spring Ecosystem

Spring Boot Spring Cloud Spring Security Spring Data JPA QueryDSL

### Messaging & Streaming

Apache Kafka Apache Flink Redis Pub/Sub

### Database & Cache

PostgreSQL MariaDB Redis Elasticsearch

### Resilience & Observability

Resilience4j Circuit Breaker Zipkin Micrometer Tracing

### Authentication

OAuth2 JWT Keycloak

### Infrastructure

AWS (EC2, RDS, S3, CodeDeploy) NGINX Docker Jenkins Travis CI

## >Contact

- Email: [allsdfgh88@gmail.com](mailto:allsdfgh88@gmail.com)
- LinkedIn: [www.linkedin.com/in/%ED%95%91%EC%84%9C-%EA%B0%80-b5b3b7393](https://www.linkedin.com/in/%ED%95%91%EC%84%9C-%EA%B0%80-b5b3b7393)

- GitHub: <https://github.com/hyot88>