

# 탁찬홍 Backend Engineer (신입)

lim70kr38@gmail.com · [github.com/Takch02](https://github.com/Takch02) · [velog.io/@takch02/posts](https://velog.io/@takch02/posts)

## Introduction

장애 시나리오를 설계 단계부터 상정하고, 문제를 수치로 재현해 해결하는 백엔드 엔지니어입니다.

XRPL 결제 유실 방지를 위해 Outbox 패턴으로 이벤트 정합성을 확보했고, 'NAFAL' 프로젝트에서 입찰 집중 구간의 DB 락을 Redisson 분산 락으로 분리해 DB 커넥션 고갈을 해결했습니다.

또한, Apache/Amoro · MapStruct 오픈소스 기여 및 MCP 기반 서버 진단 에이전트로 카카오 PlayMCP 공식 승인을 받았습니다.

## Open Source

**Apache/Amoro** [\[PR 링크\]](#) 2026.03 ~ 2026.04

- Mixed-Hive 비파티션 테이블 테스트 추가 과정에서 Iceberg - Hive 간 데이터 불일치 버그 발견
- 비파티션 테이블에 부분 덮어쓰기 시 Hive 가 일부 데이터만 인식하는 문제 확인 후 UpdateHiveFiles 클래스에 location consistency 검증 로직 추가로 수정

**MapStruct** [\[PR 링크\]](#) 2026.04 ~ 2026.05

- ZonedDateTime · OffsetDateTime · Instant 간 LocalDateTime 변환 시 불필요한 XML 타입 경유 버그 수정
- 8 가지 암묵적 타입 변환 구현 및 테스트 케이스 추가, 공식 문서화까지 기여

## Skills

**[Core]** Java / Spring Boot / Spring Data JPA / QueryDSL / MySQL / Redis-Redisson

**[Familiar]** TypeScript / NestJS / MongoDB

**[Test & Tool]** K6 / Docker / AWS EC2-RDS-S3 / GitHub Actions

## Project

'NAFAL' 경매 플랫폼 - 해커톤 최우수상 및 개인 리팩토링 [\[GITHUB\]](#) [\[BLOG\]](#)

일정 : 2025.11 ~ 2026.03 (대용량 데이터셋 환경에서 시스템 고도화 및 리팩토링) / 2025.08 (해커톤 참여 및 최우수상)

기술 스택 : JAVA 17, Spring Boot (3.4.5), MySQL(8.0), JPA, QueryDSL, Redis, AWS, GitHub Actions

참여인원 : 해커톤 (6인: BE 2, UX/UI 1, PM 3) — 입찰 도메인 담당 / 리팩토링 (개인)

서비스 : 기업 폐기 예정 자산 재판매 플랫폼

테스트 환경 : K6 부하테스트 / DB (t2.medium), App (t2.medium), Redis (t2.small) / 입찰 데이터 300만건 가정

**대규모 트래픽 집중 시 DB 커넥션 풀 고갈 방지를 위한 분산 락 도입**

- 경매 마감 임박 시 비관적 락으로 인한 DB 커넥션 점유 유지를 해결하고자, 락 메커니즘을 Redis 로 분리하는 Redisson 분산 락 도입 (경합 집중 시  $O(N^2)$  위험이 있는 낙관 락 제외)
- 부하 테스트(K6 최대 200VU) 기준 입찰 응답 속도 59.7% 개선(1,649ms → 665ms) 및 입찰 요청의 500 에러 5% 해결로 경매 마감 임박 구간 입찰 실패 해소

## 비동기 처리를 통한 Deadlock 제거와 데이터 정합성 보장

- 입찰-유저 트랜잭션 간 cross lock Deadlock 발생, 참여도 데이터 유실 발생을 해결하기 위해 TransactionalEventListener 로 트랜잭션 경계 분리 및 비동기 아키텍처 구축
- 이벤트 유실 방지를 위해 Claim Check 패턴을 도입하여 Event Store 기반의 복구 파이프라인 구현
- 100 스레드 동시 요청 테스트에서 Deadlock 미발생 및 참여도 데이터 정합성 확보

## 인프라 증설 없이 쿼리 튜닝으로 Timeout(p95 30s) 장애 개선

- 300 만 건 환경에서 인덱스 부재 및 Count 쿼리 병목으로 발생한 Timeout 을 해결하고자 Covering Index & Deferred Join 적용
- 불필요한 Count 쿼리를 Page → Slice 구조로 전환 및 Redis Caching 으로 최적화
- (K6 ramping-vus 최대 100VU 부하) 조회 응답 99.2% 개선(30s → 62ms)

## 무역 파트너 매칭 및 XRPL 에스스로 결제 B2B 플랫폼 [\[GITHUB\]](#) [\[배포링크\]](#)

일정 : 2026.04 ~ 진행 중

기술 스택 : TypeScript, NestJS, MongoDB Atlas, Redis, XRPL, Docker, Railway, GitHub Actions, Claude Code

참여인원 : 4명 (대표(기획), BE 1, FE 1, UI/UX 1) / 팀 리드(BE 단독) · API 스펙 및 기술 방향 결정, 파트너사 기술 미팅 주도

서비스 : 무역거래 바이어-셀러 검색 및 매칭 후 XRPL 무역대금 결제 플랫폼

## 외부 네트워크(XRPL) 지연 환경에서 안정적인 비동기 결제 처리 구조 구축

- XRPL 네트워크 지연(약 3~5s) 문제를 Redis Bull Queue 비동기 처리로 전환하여 결제 API 응답 속도 3,163ms → 14ms 로 단축
- 비동기 처리 과정에서 이벤트 유실 문제를 막기 위해 Outbox 이벤트 기록을 결제 상태 변경과 동일 트랜잭션으로 관리
- MQ 장애 및 실패/타임아웃 상황에 대비해 재시도 지수 백오프(최대 10 회, 5s) 및 Recovery Scheduler 구현
- MongoDB Change Stream 단절, 재시도 소진 등 5 가지 장애 시나리오 E2E 테스트에서 복구 검증

## 도메인 특화 키워드 매칭 중심의 Hybrid Search 구축 및 검색 품질 개선

- 고차원 벡터 검색(1536 차원)만으로 잡지 못하는 도메인 고유 키워드 누락 문제를 해결하고자 키워드 병렬 검색 시스템 도입
- 점수 스케일 차이로 인한 랭킹 역전 문제를 RRF(Reciprocal Rank Fusion) 알고리즘 적용 및 페널티 튜닝으로 해결
- 동일 쿼리 테스트 기준 관련 파트너 포함율을 기존대비 약 9.4 배 개선(0.67/10 → 6.3/10 기준)

## Spring 서버 진단 MCP 'Server Doctor' – 카카오 PlayMCP 공식 승인/배포 [\[GITHUB\]](#) [\[BLOG\]](#) [\[배포링크\]](#)

일정 : 2026.01 ~ 2026.03

기술 스택 : JAVA 21, Spring Boot (3.4.2), Spring AI, Python, MySQL(8.0), JPA, AWS, GitHub Actions

참여인원 : 개인

서비스 : Spring 서버의 Log/Metrics를 로그수집기(Forwarder)가 추출하여 LLM을 통해 자동 진단하는 MCP

테스트 환경 : K6 부하테스트 / 로컬 환경 (DB-App 동일 머신)

## 인프라 보안 확보를 위한 Sidecar 기반 Push 아키텍처로 사용자 서버 보안성 확보

- Log/Metrics 수집을 위해 외부 포트(9090) 개방하는 Pull 방식 로그 유출 리스크를 방지하고자 내부 Forwarder 를 Sidecar 형태로 배치한 Push 방식 채택
- 외부 포트 노출 차단 및 .env, application.yaml, docker-compose.yml 설정만으로 기존 서버와 독립적인 로그 수집 구조 확보

## 대량 로그 유입 시 발생하는 DB 트래픽 병목 해소 및 Deadlock 제거

- 장애 복구 및 네트워크 불안정으로 동일 로그가 유입될 때 발생하는 중복 키 S-Lock → X-Lock 승격 Deadlock 과 단건 Insert 의 네트워크 병목 인지
- 단건 처리를 JDBC Bulk Insert 로 전환하여 처리량을 개선하고, @Retryable 을 적용한 간헐적 Deadlock 자동 복구 구조 구현
- 기존 대비 TPS 19043 → 32080(68.9% 향상), 응답 시간 249.32ms → 85.45ms(65.3% 개선) 및 재시도 기반 복구 구조로 장애 전파 최소화

## 시스템 연쇄 장애 방어를 위한 유량제어 및 생존성 확보

- 통신 장애 시 일시적인 무한 재시도로 인해 발생하는 메모리 고갈(OOM) 및 사용자 서버 다운타임 리스크를 방지하고자 Queue, Jitter, Timeout(1 분) 기반의 백오프 정책 도입
- 재시도 과정에서 발생하는 중복 데이터 적재를 막기 위해 SHA-1 기반의 멍등성 키와 ON DUPLICATE KEY UPDATE 구문을 활용한 중복 삽입 방지 구조 구현
- 동일 이벤트 재처리 테스트 기준 중복 데이터 미발생 검증

## Education

---

**경북대학교** 글로벌소프트웨어융합전공(복수전공) / 건설방재공학과(본전공)

기간 : 2021.03 ~ 현재 (2027년 2월 졸업예정, 6/23부터 풀타임 가능)

**SQLD** 취득일 : 2025.12