

## 👋 저를 소개합니다

안녕하세요. 11년차 백엔드 개발자 한형섭입니다.

저는 Java 기반의 시스템을 개발하고 운영하며, **기술을 통해 문제를 해결하고 시스템의 안정성을 높이는 과정**에 몰입해왔습니다. 특히 인사(HR, IM) 시스템을 다루면서, 복잡한 요구사항을 기술적으로 풀어내고 실제 운영 환경에서 발생할 수 있는 문제들을 예측하여 대응하는 데 저만의 강점을 갖게 되었습니다.

- **기술을 대하는 저의 관점:** 저는 기술 스택을 단순히 나열하기보다, **각 기술의 강점과 한계를 이해하고 문제 해결에 적합한 도구를 선택**하는 것을 중요하게 생각합니다. 예를 들어, [IM, HR 시스템] 프로젝트에서는 초기 MyBatis를 사용하다가 이후 JPA로 전환하며 생산성을 높였던 경험처럼, 상황과 목표에 맞춰 유연하게 기술을 적용하는 것을 추구합니다. 또한, [IM, HR 시스템] 프로젝트에서 Querydsl이 RESTful API를 대신해 외부 연계 자료 제공에 더 효과적이라고 판단했던 것처럼, 정해진 방식에 얽매이지 않고 최적의 길을 찾는 데 노력합니다.
- **꾸준함과 성장에 대한 믿음:** 새로운 기술이나 변화에 대해 열린 마음을 가지고 꾸준히 배우고 성장하는 것을 좋아합니다. 제가 맡은 시스템이 안정적으로 운영되고, 그 안에서 저와 동료들이 함께 성장하는 것이 개발자로서의 보람이라 생각합니다.

## 🌟 프로젝트 경험: 고민과 문제 해결

### 1. IM, HR 인사시스템 개발 및 운영 (2018.02 - 현재)

- **시작 배경과 당면 과제:** 이 프로젝트는 회사 핵심인 인사 데이터를 다루는 시스템으로, **오래된 액티브X 기반 시스템을 웹 표준 환경으로 전환**해야 하는 필요성에서 시작되었습니다. 동시에 기존 시스템이 가진 근본적인 문제, 즉 해외 네트워크의 불안정성으로 인한 데이터 유실이나 전송 지연이 잦았다는 점을 해결해야 했죠. 이는 신규 계정 생성 지연과 업무 병목으로 이어지는 문제였습니다. 우리팀에게는 웹 표준 전환과 더불어, 불안정한 네트워크 환경에서도 데이터의 신뢰성을 확실히 보장해야 하는 과제가 주어졌습니다.
- **기술적 접근과 핵심 의사결정 (레이어드 아키텍처 및 ORM 전환 경험):** 프로젝트는 **탄탄한 레이어드 아키텍처**를 기반으로 견고하게 구축되었습니다. 이 아키텍처는 각 계층의 역할을 명확히 분리하여 안정성을 높이고, 초급개발자들이 많은 환경에서 유리했습니다. 저는 이 구조 안에서 데이터 무결성 확보에 중점을 두었습니다.

월 약 7.5만 건의 대량 데이터를 불안정한 해외 네트워크에서 안정적으로 처리하기 위해, 단순히 재전송을 넘어 **MD5 체크섬 기반의 자동 데이터 검증 및 재전송 로직을 설계하고 구현**했습니다. 이는 시스템의 **자율 복구 능력**을 높여 운영 개입을 최소화하는 결정이었습니다.

데이터 처리 방식에 있어서는, 프로젝트 초반에는 **\*\*MyBatis\*\***를 활용하여 SQL 쿼리를 직접 관

리했습니다. 이는 복잡한 DB 구조에 유연하게 대응하고 쿼리 성능을 미세하게 제어하는 데 장점이 있었습니다. 하지만 시간이 지나면서 현업으로부터 요구사항이 다양해지고 개발 속도를 높일 필요했습니다. 그래서 데이터 접근의 효율성을 높이기 위해 **JPA로 변경작업을 진행**했습니다. JPA 도입을 통해 반복적인 JDBC 코드를 줄여 생산성을 향상시킬 수 있었습니다.

또한, 외부 시스템으로의 데이터 제공 요구가 많았는데, 이때 RESTful API 형태의 개별 연동 방식이 아닌, **Querydsl을 변경, 활용하여 자료를 효율적으로 제공**했습니다. Querydsl은 복잡한 조건에 따른 데이터 추출 요구사항에 유연하게 제공하고, 데이터 가공을 간소화하기 위한 이유였습니다. 이는 코드 기반으로 유연한 쿼리를 작성할 수 있게 하여 개발 생산성을 확보할 수 있었습니다.

- **주요 성과와 배운 점:** 이러한 노력 덕분에 7년이 넘는 기간 동안 데이터 인식 오류를 5건 이하로 줄였고, 월 약 7.5만 건의 대량 데이터를 안정적으로 처리했습니다. 이 경험을 통해 **레이어드 아키텍처의 견고함과 명확한 책임 분리가 대규모 시스템의 안정적인 운영에 얼마나 중요한지**를 체감했습니다. 더불어, 프로젝트 생애주기 동안 **기술 스택을 전환(MyBatis → JPA)하며 얻은 경험은 각 ORM의 장단점과 전환 시 고려사항에 대한 깊은 이해**를 주었습니다. 또한, 불안정한 외부 환경에서도 데이터의 신뢰성을 지켜내는 방어적인 설계와 자동화된 검증 메커니즘의 중요성을 깊이 있게 이해하게 되었습니다.

## 2. 협력사 계정 생성 및 관리 시스템 개발 및 운영 (2021.03 - 현재)

- **프로젝트 배경과 당면 과제:** 협력사 계정 생성 및 관리가 수작업으로 이루어져 **계정 생성에 여러 과정이 있어 48시간 이상 걸리고 담당자 업무 부담이 매우 컸습니다.** 계정 만료 관리 미흡으로 보안 취약점도 존재하여, 이 비효율을 개선하고자 했습니다.
- **기술적 접근과 핵심 의사결정:** 이 프로젝트는 **애자일(Agile) 방식**으로 진행되었습니다. 초기에는 가장 시급한 '수작업 계정 생성 병목' 해결에 집중하여 핵심 기능을 빠르게 구현했고, 이후 사용자 피드백을 반영하며 시스템을 점진적으로 확장했습니다. (계정 일시정지, 권한 설정, 이용 시스템 구분 등 목적별 로직 추가)

Java/Spring 기반 백엔드의 RESTful API를 기본으로 설계하여 계정 생성부터 권한 부여, 만료 처리 등을 자동화했습니다. 기능을 확장함에 따라, **모듈화와 확장 가능한 아키텍처에 대해 지속적으로 고민**했습니다. 민감한 계정 정보 보호를 위해 **AES256 암호화 도입과 유연한 권한 관리 체계를 단계적으로 강화**한 것은 중요한 결정이었습니다. 애자일 방식을 통해 변화하는 요구사항에 빠르게 반영하며 시스템을 고도화할 수 있었습니다.

- **주요 성과와 배운 점:** 현재 계정 생성 시간을 50% 단축(48h → 24h)했고, 담당자 수동 처리 업무를 월 20건 이상 줄였습니다. 협력사 개발자들의 포털 접근/활용 경험도 개선되며 긍정적인 피드백을 받았습니다. 이 경험을 통해 **애자일 방식이 복잡하고 변화가 잦은 비즈니스 환경에 효과적임**을 체감했습니다. 또한, **점진적인 기능 확장 속에서도 시스템의 견고함을 유지하는 방법**에 대해 실천할 수 있었습니다.

## 3. 비밀번호 통합 관리 시스템 개발 및 운영 (2022.08 - 현재)

- **프로젝트 배경과 당면 과제:** 회사 내 13개의 분산된 시스템이 있었고, 직원들이 비밀번호를 잊

어버리면 각 시스템마다 개별적으로 변경해야 하는 불편함이 있었습니다. 이는 직원들의 불편함을 야기할 뿐만 아니라, IT 헬프데스크의 반복적인 문의로 인한 업무 부담이 심각했습니다. 저는 이 복잡한 과정을 단 한 번의 요청으로 처리할 통합 시스템을 구축하여 사용자 편의성과 IT 운영 효율을 개선하고자 했습니다.

- **기술적 접근과 핵심 의사결정 (분산 환경, 비동기 처리):** 13개 분산 시스템에 동시 패스워드 변경을 효율적으로 처리하는 것은 제 기술적 도전이었습니다. 단순히 C#(그 당시 오토에버의 암호 변경 C#으로 되어있는 모듈이 필수로 들어가야함)에는 백엔드 로직으로 각 시스템을 순차적으로 호출하는 방식은 성능 병목과 시스템 간의 강한 의존성을 유발할 것이 명확했습니다. 이러한 문제를 해결하고 시스템의 확장성과 사용자 경험을 고려하여 '**비동기 처리(async/await) 아키텍처를 도입**하기로 결정했습니다.

시스템에서 단 한 번의 패스워드 변경 요청을 받으면, 13개 시스템들에게 동시에 발송해 비동기적으로 패스워드를 업데이트하도록 설계했습니다. 이 과정에서 비동기 환경에서의 **\*\*데이터 일관성 유지 (Eventual Consistency)\*\***를 위해 각 서비스별 로컬 트랜잭션을 설정했고, **실패 시의 재처리 메커니즘**으로는 메시지 재발송 큐를 활용하는 등 로직을 적용하여 시스템 간의 일관성을 주었습니다. 특정 시스템의 장애가 전체 패스워드 변경 프로세스를 멈추지 않도록 설계하는 것이 중요했습니다.

- **주요 성과와 배운 점:** 패스워드 분실 요청이 70% 감소했고, IT 헬프데스크 문의도 월 30건 이상 줄었습니다. 이 프로젝트를 통해 **\*\*시스템 환경에서 비동기 처리 아키텍처가 갖는 강력한 이점 (유연성, 확장성, 성능)\*\***을 명확히 깨달았습니다. 또한, 분산 트랜잭션 상황에서 발생할 수 있는 데이터 일관성 문제에 대한 깊이 있는 이해와 이를 해소하기 위한 기술적 고민의 중요성을 체감했습니다.

---

## 끝으로...

- **함께 자라나는 개발 문화:** 저는 개인의 성장뿐만 아니라 팀 전체의 성장을 중요하게 생각합니다. 동료들과의 코드 리뷰를 통해 서로의 지식을 공유하며, 이런 활동들이 효율적인 개발 문화와 팀의 역량 강화로 이어진다고 생각합니다.
- **멈추지 않는 배움의 즐거움:** 운영 통해 얻은 경험은 저에게 새로운 기술에 대한 관심을 갖게 했습니다. 현재 관심 있는 기술 분야(디자인패턴, 아키텍처, AI, 대용량처리 등)에 깊이 관심을 가지고 온라인 강좌 수강 등을 통해 끊임없이 지식을 확장하고 있습니다. 항상 더 나은 해결책을 찾기 위해 배우는 개발자로 남고 싶습니다.

---

## 마무리하며

지난 11년간 쌓아온 백엔드 개발 및 시스템 운영 노하우는 귀사에 큰 가치를 더할 수 있다고 생각합니다. 특히 타인의 대한 배려와 경험을 바탕으로, **안정적인 서비스를 함께 만들어가고, 팀의 기술적 성장에 기여하고** 싶습니다. 감사합니다.