

👋 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

1. IM, HR 인사시스템 개발 및 운영

구분	내용
프로젝트명	IM, HR 인사시스템 개발 및 운영
기간	2018.02 - 현재
시작 배경과 과제	<ul style="list-style-type: none"><li>· 낡은 액티브X 기반의 핵심 인사 시스템을 웹 표준으로 전환하기 위해 프로젝트 시작.</li><li>· 해외 네트워크 불안정으로 인한 데이터 유실 및 전송 지연 문제 해결 필요.</li><li>· 이로 인한 계정 생성 지연, 업무 병목 해소 및 데이터 신뢰성 확보를 목표.</li></ul>
기술적 접근과 의사결정	<ul style="list-style-type: none"><li>· 안정성 향상을 위한 레이어드 아키텍처 기반 시스템 구축. 역할 분리가 명확해 초급 개발자에게 유리했고, 데이터 무결성 확보에 중점.</li><li>· 월 7.5만 건의 대량 데이터 안정적 처리를 위한 MD5 체크섬 기반 자동 검증 및 재전송 로직 구현. 이를 통한 운영 개입 최소화.</li><li>· 초기 MyBatis에서 개발 속도 향상을 위해 JPA로 전환. 반복적인 JDBC 코드 감소 및 생산성 향상.</li></ul>

구분	내용
	<ul style="list-style-type: none"> <li>· 외부 데이터 제공 시 RESTful API 대신 Querydsl 활용. 복잡한 조건의 데이터에 대한 유연한 대응 및 개발 생산성 확보.</li> </ul>
주요 성과와 교훈	<ul style="list-style-type: none"> <li>· 7년간 데이터 인식 오류 5건 이하로 감소 및 월 7.5만 건 데이터 안정적 처리.</li> <li>· 대규모 시스템에서 명확한 책임 분리를 가진 레이어드 아키텍처의 중요성 체감.</li> <li>· MyBatis → JPA 전환 과정에서 각 ORM의 장단점 및 전환 시 고려사항에 대한 깊은 이해.</li> <li>· 불안정한 환경에서 데이터 신뢰성을 지키는 방어적 설계와 자동화된 검증 메커니즘의 중요성 깨달음.</li> </ul>

## 2. 협력사 계정 생성 및 관리 시스템 개발 및 운영

구분	내용
프로젝트명	협력사 계정 생성 및 관리 시스템 개발 및 운영
기간	2021.03 - 현재
시작 배경과 과제	<ul style="list-style-type: none"> <li>· 수작업으로 48시간 이상 소요되던 계정 생성 프로세스 자동화를 통한 비효율 개선 필요.</li> <li>· 계정 만료 관리 미흡으로 인한 보안 취약점 해결 과제.</li> </ul>
기술적 접근과 의사결정	<ul style="list-style-type: none"> <li>· 애자일 방식으로 가장 시급한 문제부터 빠르게 해결 후, 피드백을 반영하며 점진적 시스템 확장. (계정 일시정지, 권한 설정 등)</li> <li>· Java/Spring 기반 RESTful API를 통한 계정 생성-만료 전 과정 자동화.</li> <li>· 모듈화, 확장 가능한 아키텍처에 대한 지속적인 고민 및 AES256 암호화, 유연한 권한 관리 체계 도입.</li> <li>· 애자일 방식을 통한 요구사항 변화에 대한 빠른 대응 및 시스템 고도화.</li> </ul>
주요 성과와 교훈	<ul style="list-style-type: none"> <li>· 계정 생성 시간 50% 단축 (48시간 → 24시간) 및 담당자 수동 업무 월 20건 이상 감소.</li> <li>· 협력사 개발자의 포털 사용 경험 개선 및 긍정적 피드백 수신.</li> <li>· 변화가 잦은 비즈니스 환경에서 애자일 방식의 효과 체감.</li> <li>· 점진적 기능 확장 속에서 시스템 견고성을 유지하는 방법 실전.</li> </ul>

## 3. 패스워드 통합 관리 시스템 개발 및 운영

구분	내용
프로젝트명	패스워드 통합 관리 시스템 개발 및 운영
기간	2022.08 - 현재
시작 배경과 과제	<ul style="list-style-type: none"> <li>· 13개 분산 시스템의 패스워드를 개별적으로 변경해야 하는 직원 불편 해소 필요.</li> <li>· 반복적인 패스워드 문의로 인한 IT 헬프데스크의 심각한 업무 부담 감소 필요.</li> <li>· 단 한 번의 요청으로 전 과정을 처리하는 통합 시스템 구축을 통해 사용자 편의성 및 운영 효율 개선 목표.</li> </ul>
기술적 접근과 의사결정	<ul style="list-style-type: none"> <li>· 13개 시스템 동시 패스워드 변경을 위한 비동기 처리(async/await) 아키텍처 도입. (성능 병목 및 의존성 문제 해결)</li> <li>· 단일 요청을 받아 13개 시스템에 동시에 비동기적으로 변경 요청을 발송하도록 설계.</li> <li>· 비동기 환경의 데이터 일관성을 위한 서비스별 로컬 트랜잭션 설정 및 실패 시 메시지 큐를 활용한 재처리 로직 적용.</li> <li>· 특정 시스템 장애가 전체 프로세스 중단으로 이어지지 않도록 설계.</li> </ul>
주요 성과와 교훈	<ul style="list-style-type: none"> <li>· 패스워드 분실 요청 70% 감소 및 IT 헬프데스크 문의 월 30건 이상 감소.</li> <li>· 분산 환경에서 비동기 아키텍처가 가진 유연성, 확장성, 성능의 강력한 이점 이해.</li> <li>· 분산 트랜잭션 상황의 데이터 일관성 문제와 그 기술적 해결의 중요성 체감.</li> </ul>

## 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 등 을 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 기여**하고 싶습니다. 감사합니다.