

# 김민지 Minji Kim

Backend Developer / Data Engineer

 [GitHub](#)  [Blog](#)

## Summary

장애 대응부터 성능 개선, 자동화까지 시스템의 지속 가능성과 안정성에 집중해온 백엔드 개발자입니다. 보험 마이데이터 도메인을 기반으로 Java와 Node.js(TypeScript)를 활용해 유연하고 견고한 API 및 배치 시스템을 설계·구현 해왔습니다. 도메인 지식을 함께 공부하며 데이터의 맥락을 이해하는 것을 중요하게 생각하고, 이를 바탕으로 Airflow 기반 데이터 파이프라인을 직접 설계·운영하며 안정적인 데이터 흐름과 서비스 품질 개선에 기여한 경험이 있습니다.

개발은 결국 문제를 정확히 정의하고 끝까지 해결해내는 일이라 생각합니다. 커뮤니케이션을 통해 문제를 함께 이해하는 것을 추구하고, 해결 과정 전반에 책임감을 가지고 일합니다.

## Work Experience

### Aijinet

R&D Center Backend Chapter / Backend Developer

2022.02 - 현재 • Seoul, South Korea

- Spring Boot 기반 마이데이터 보험업권 관련 API 개발 및 어드민 페이지 개발
- Airflow 및 GCP 기반의 병렬 처리 스케줄링 시스템 설계 및 운영 자동화
- 보험 약관·보험료 등 정형·비정형 데이터를 대상으로 한 ETL 파이프라인 재구축 및 운영
- Bodoc 앱의 보험 진단·추천 기능을 위한 엔진 로직 설계 및 Express.js 기반 API 서버 개발
- PostgreSQL, Nest.js 기반 SaaS 아키텍처 도입 및 보험료 계산 API 개발
- Spring Boot 기반 B2B 보장분석 리포트 페이지 개발

### Aijinet

DataLabs / Data Researcher

2021.08 - 2022.01 • Seoul, South Korea

- KOICD 질병코드 간 연관성 분석 및 매핑을 통한 약관 데이터와 보상 내용 정제 작업
- 보험 공식 데이터를 기반으로 한 고객 보유 상품의 가성비 분석 로직 기획 및 엔진 개발 지원
- 보험료 및 공사실 데이터 수집을 통한 엔진 메타데이터 구성 및 Pandas 기반 가공

## Projects

### 마이데이터 보험업권 통계치 추출 자동화 시스템 고도화

고객사 신규 요청사항에 따른 로직 변경 및 최대 6,000건 건 데이터 처리 성능 고도화

2025.09

- 추출 조건 변경으로 인한 데이터 처리 병목을 해결하기 위해 Airflow DAG 구조 전면 재설계
- Polars 및 Parquet 도입하여 연산 효율과 I/O 성능 개선, 처리 속도 약 50% 단축
- 데이터를 chunk 단위로 분산 저장·로딩하는 구조로 변경하여 메모리 부하 해소 및 파이프라인 안정성 확보
- Pandera 기반 스키마 검증 로직을 도입하여 데이터 유입 시 타입 안정성과 유효성 보장
- B2B 고객사별 통계치 추출 작업을 자동화하여 수작업 제거 및 운영 안정성 확보

[Python](#) [Polars](#) [Pandera](#) [Airflow](#) [GCS](#) [Slack API](#)

### 마이데이터 숨은 보험금 찾기

마이데이터 연동 시 숨은 보험금 찾기 신규 기능 추가

2025.08

- Spring Boot 기반 숨은 보험금 조회 API 구현 (숨은 보험금 목록, 연동 결과 조회 등)
- 기획·FE 팀과의 협업을 통해 숨은 보험금 조회 플로우 및 화면 노출 조건 정의
- 마이데이터 연동 여부, 숨은 보험금 유무 케이스 기반의 응답 포맷 설계

[Java](#) [Spring Boot](#)

### B2B SaaS 보장분석 서비스 개발

RabbitMQ 기반 메시지 처리 및 보장 리포트 템플릿 자동화

2025.07 - 2025.09

- 판매채널에 따른 Thymeleaf 템플릿 엔진 기반 보장 분석 리포트 HTML 바인딩
- 정적 HTML 페이지 생성 후 S3 업로드 자동화
- 요구사항 기반 DLQ 를 포함한 메시지 큐 서버 구성 및 장애 대응 구조 설계 기여

[Java](#) [Spring Boot](#) [RabbitMQ](#) [Thymeleaf](#) [S3](#)

### Amplitude 사용자 정보 일괄 처리 시스템 개발

Airflow 기반 병렬 처리 ETL 파이프라인 설계 및 GCS 연동 구조 개발

2025.06

- expand 기반 병렬 처리 시스템 신규 구축 및 최적화로 6만 건 데이터 처리 시간 84% 개선
- Task와 GCS 간 매핑 구조를 설계하여 에러 발생 시 추적 가능성 향상 및 운영 효율 증대
- Pydantic 기반 입력 데이터 검증 로직을 통해 타입 안정성과 유효성 보장
- Amplitude API 호출 시 Exponential Backoff 로직 적용으로 장애 대응력 및 재시도 안정성 확보

[Python](#) [Pydantic](#) [Airflow](#) [GCS](#) [Amplitude API](#) [Slack API](#)

### 전사 KPI 대시보드 데이터 적재 자동화

전사 KPI 시각화를 위한 일별 누적 데이터 ETL 파이프라인 설계 및 자동화

2025.05

- Airflow 기반 DAG 설계 및 MySQL-Google Sheet API 연동을 통한 데이터 적재 자동화
- Google Calendar API 연동하여 공휴일 여부 판단
- Task branch를 통해 DAG 실행 흐름을 제어로 리소스 사용 최적화
- 통합 트랜잭션 처리 방식 도입으로 데이터 적재의 일관성과 안정성 확보
- MSSQL 기반 스키마 설계 및 Looker Studio 시각화 환경 구성

[Python](#) [Airflow](#) [MSSQL](#) [MySQL](#) [Google Calendar API](#) [Google Sheet API](#) [Slack API](#)

### 보험료 기반 추천 데이터 자동화 시스템 구축

Airflow 도입 및 보험료 기반 추천 점수 생성을 위한 대용량 데이터 ETL 자동화

2025.03 - 2025.04

- Airflow 도입 및 GCS 연동을 포함한 ETL 파이프라인 자동화 DAG 설계
- 약 150만 건의 API 요청을 비동기 처리로 구현, 5분 이내 처리 완료하여 생산성 향상
- PostgreSQL 기반 스키마 설계 및 구축
- 내부 엔진 빌드 시 최신 메타데이터 자동 참조 구조로 배포 효율 및 데이터 신뢰도 강화
- Slack API 연동을 통한 실시간 모니터링 체계 구축으로 운영 안정성 확보

[Python](#) [sqlalchemy](#) [Airflow](#) [GCS](#) [PostgreSQL](#) [Slack API](#)

### Bodoc 어드민 페이지 보안 강화 시스템 개발

금융감독원 보안 가이드 대응을 위한 인증 및 계정 보안 고도화

2025.02

- Spring Security 기반 마이페이지 API 설계 및 개발
- 비밀번호 인증 및 변경 로직 구현으로 보안 정책 구현
- JUnit 기반 단위 테스트 커버리지 96% 달성 및 Redoc 문서화

[Java](#) [Spring Boot](#) [Spring Security](#) [JUnit](#) [MySQL](#) [Redoc](#)

### 통계 데이터 생성을 위한 배치 시스템 구축

MySQL Event Scheduler 기반 통계 데이터 자동 추출 및 최신화 체계 구축

2025.01 - 2025.02

- 통계 처리 전용 스키마 설계 및 일괄 처리(Batch) 자동화 구조 구현
- 기존 운영 스키마 분석을 바탕으로 인덱스 재설계 및 최적화 적용
- 인덱스 적용을 통해 통계 데이터 추출 속도 약 90% 개선
- 기존 수동 통계 작업을 주 단위 자동화하여 데이터 최신성 및 운영 효율성 향상

[MySQL](#) [Event Scheduler](#)

### 보험 약관 크롤링 시스템 재구축

반복적인 수동 크롤링 API 요청 기반 수집 작업 자동화 및 데이터 파이프라인 재구축

2024.04 - 2024.12

- 전체 ETL 파이프라인 재설계 및 오류 복구 가능한 구조 구현
- HTML 파싱 및 API 응답 기반 이중 데이터 수집 방식 적용으로 크롤링 범용성 향상
- 월간 오류 발생률 142건에서 15건으로 89% 감소
- Exponential backoff 기반 요청 재시도 체계 강화로 운영 안정성 확보
- 전체 처리 기간 7일에서 3일로 단축하여 수집 주기 개선
- 분기 단위의 약관 업데이트를 주 단위로 Cron 스케줄링하여 운영 효율성을 91% 향상

[Python](#) [sqlalchemy](#) [GCS](#) [PostgreSQL](#) [Django ORM](#) [Slack API](#)

### 앱 백엔드 서버 마이그레이션 지원

유지보수성과 구조적 일관성을 위한 Laravel → Spring Boot 기반 백엔드 API 마이그레이션 수행

2024.08 - 2024.09

- Laravel 기반 백엔드 서버를 Spring Boot 환경으로 특정 API를 마이그레이션하여 시스템 일관성 기여
- 복잡한 보험 도메인 구조를 sealed interface permits 기반으로 정의하여 인터페이스 재사용률 향상
- 중복 로직 제거 및 JPA Repository 기반 쿼리 구조 개선을 통해 공통 모듈화 진행
- 인터페이스 및 쿼리 재사용을 통해 유사 API 개발 소요 시간 5일 → 3일로 단축 (약 40% 감소)

[Java](#) [Spring Boot](#) [MySQL](#) [PHP](#)

### 간편보험료 기반 보험 상품 비교 시스템 개발

보험설계사들의 수기 설계만 작성을 자동화하는 보험 비교 엔진 구축

2024.01 - 2024.03

- Nest.js 기반 보험료 계산 및 데이터 제어 API 설계 및 성능 최적화
- 보험설계사의 상품 비교 소요 시간 80% 단축, 계약 전환율 4%p 향상
- TypeORM 도입 및 쿼리 최적화를 통해 응답 속도 90% 개선
- 데이터 검증용 API 추가 개발로 정확성 및 서비스 신뢰도 확보

[TypeScript](#) [Node.js](#) [Nest.js](#) [MongoDB](#) [GCP Cloud Run](#) [Redoc](#) [Memcached](#) [Python](#)

### SaaS 보험 엔진 고도화 및 개발



TypeScript 기반 보험 진단·추천 엔진 고도화 및 개발

2022.05 - 2025.07

- Bodoc 앱 진단·추천 탭의 비즈니스 요구사항 반영 및 안정적 유지보수 수행
- 건강 연동 여부에 따라 엔진 엔드포인트 분기 처리하여 건강 맞춤형 보험 결과 제공 로직 구현
- GCP Cloud Run 환경에서 B2C/B2B 서버 분리 및 Github branch 체계 구축
- 환경별 배포 파라미터를 반영한 GCP Cloud Run 배포 명령어를 스크립트화하여 배포 절차 자동화
- 자동화 문서화 도구 Redoc 도입으로 API 문서 관리 생산성 향상
- 메타데이터 캐싱을 위한 Memcached 적용으로 초기 로딩 성능 30% 개선
- 문자열 유사도 기반 Python 스크립트를 도입하여 상품 메타데이터 갱신 업무 시간 70% 단축, 성공률 4%p 개선
- 비개발 부서 대상 엔진 시뮬레이션 API 설계로 정책 변경 테스트 환경 제공
- 증권사 마이데이터 기반 보장 분석 기능에 TypeGuard 및 암호화 토큰 기능 설계 참여
- 마이데이터 및 신용정보원 보험 데이터 비교 분석하여 진단·추천 엔진 로직 기획 및 구현 참여

[Node.js](#) [TypeScript](#) [Express.js](#) [MongoDB](#) [Redoc](#)

## Skills

<b>&lt;/&gt; Languages &amp; Runtimes</b> Python, Java, TypeScript, JavaScript, Node.js	<b>🔗 Frameworks &amp; Libraries</b> Spring Boot, Nest.js, Express.js, Django
<b>☁ Infrastructure &amp; Databases</b> GCP, Airflow, MySQL, PostgreSQL, MSSQL, MongoDB, Docker	<b>🔧 Tools &amp; IDEs</b> Bash, Zsh, Git, Gitlab, Github, Jira, Confluence, Notion, IDEA / PyCharm / DataSpell / Datagrip / Webstorm
<b>Education</b>	<b>Certificates</b>
<b>세종대학교</b> 경영학과   Business Analytics 융합연계진공 2017.02 - 2022.02 • Seoul, South Korea <ul style="list-style-type: none"><li>Total GPA of 3.84 / 4.5</li><li>Major GPA of 3.94 / 4.5</li></ul>	 정보처리기사 2024.06  TEPS 358점 2024.09
<b>Awards</b>	
<b>지역 데이터를 활용한 데이터 분석 및 시각화 "광진구 시장 활성화 프로젝트"</b> 대학총장상 2021.12 • 세종대학교/서울특별시 <a href="#">Tableau</a> <a href="#">Python</a> <a href="#">Geopandas</a>	
<b>대학 팀프로젝트 경진대회, "삼성전자 주식 예측 보고서"</b> 최우수상 2020.02 • 세종대학교 <a href="#">Python</a> <a href="#">scikit-learn</a>	
<b>대학생 영화 수요 예측 경진대회, "개봉 전후 영화 흥행 예측 분석을 통한 의사결정"</b> 장려상 2020.01 • 소니전자 <a href="#">Tableau</a> <a href="#">Python</a> <a href="#">scikit-learn</a>	
<b>전국 대학생·일반인 고조선 독서논술대회</b> 2위 은상 2019.10 • 거래열살리기국민운동본부	
<b>교내 독서경시대회 "사소한 것들"</b> 3위 2018.10 • 세종대학교	
<b>전국 대학생·일반인 고조선 독서논술대회</b> 장려상 2018.10 • 거래열살리기국민운동본부	
<b>신입생 입시 합격수기 공모전</b> 최우수상 2017.04 • 세종대학교	