

# 신주용 | 프로젝트 포트폴리오형 이력서

## 연락처

- Contact: 010-8392-1580
- Email: sea71511@gmail.com
- GitHub: <https://github.com/SJY0917032>
- Blog: <https://sjy0917032.github.io/>

## 프로필

OTA, O2O 도메인에서 4년차 백엔드 엔지니어입니다. 71개국 / 400만 대 규모 해외 렌터카 인벤토리 시스템을 설계·운영했고, Google Sheets로 돌아가던 스타트업 업무를 NestJS 기반으로 전환하는 작업을 주도하고 있습니다. 비즈니스 요구를 기술 구조로 옮기고, 운영에서 측정 가능한 성과를 내는 걸 중요하게 생각합니다.

## 핵심 키워드

도메인경험 · 프로젝트 · 임팩트

## 포트폴리오 개요

- 경력 연차: 백엔드 4년차 (OTA/O2O 도메인)
- 강점: 문제 구조화, 도메인 모델링, 운영 자동화, 비용 최적화
- 일하는 방식: 문제 정의 → 기술 설계 → 지표 기반 검증

## 주요 프로젝트

### 인수증 발행 시스템 이관

- 소속/도메인: 뽕득 (2025.04 ~)
- 배경: GAS 기반 동기 파이프라인이라 업체 하나가 실패하면 뒤 업체 전부 발송이 멈추는 구조였습니다. 실패 알림도 없어서 다음 날 업체 전화로 알게 되는 상황이 월 2~3회 반복됐습니다.
- 기여: 업체별 SQS 메시지 독립 처리로 전환. 한 업체 실패가 다른 업체로 전파되지 않음
- 기술: NestJS, SQS, Lambda(Python), NHN Cloud SMS
- 임팩트: 발송 실패 월 2~3회 → 0건 / 수기 작업 완전 제거

### TMS 레거시 개편

- 소속/도메인: 뽕득 (2025.04 ~)
- 배경: 이전 담당자 퇴사로 API 문서도 테스트도 없는 TMS 연동 코드를 인수했습니다. 타임아웃이 나도 재시도가 없어서 운영팀이 직접 TMS 콘솔에 들어가 수기로 처리하는 일이 월 3~5회, 건당 30분씩 발생했습니다.
- 기여: 인터페이스 추상화 + MockAdapter 분리로 로컬 테스트 환경 확보
- 기술: NestJS, MySQL, Slack Webhook
- 임팩트: 수기 대응 월 3~5회 → 0회 / 장애 인지 시점: 다음 날 → 즉시

### 결제 자동화 및 PG 추상화

- 소속/도메인: 뽕득 (2025.04 ~)
- 배경: API가 없는 CMS라 담당부서와 개발팀이 매월 3번씩 수기업무를 통해 CMS 데이터를 갱신하고 있었습니다. 데이터가 늘어날수록 수기 작업 시간이 늘어났고, 휴먼에러가 지속 발생하여 이를 PG 연동을 통한 정기결제 시스템으로 변경 했습니다.

- 기여: 토스페이먼츠 빌링키 기반 정기결제로 자동화
- 기술: NestJS, 토스페이먼츠, 이니시스
- 임팩트: 월 3회 수기 작업 제거 / PG 추가 시 Adapter 1개 구현으로 완료

## NestJS 어드민 시스템화

- 소속/도메인: 뽀득 (2025.04 ~)
- 배경: 운영 업무 전체가 Google Sheets 기반이었습니다. 시트 수십 개가 서로 참조 걸려 있어서 하나 바꾸면 다른 시트가 깨지고, 누가 데이터를 바꿨는지 추적할 방법이 없었습니다.
- 기여: 팀·파트·인원 3계층 RBAC 설계. 도메인별로 NONE/READ/WRITE/MANAGE 4단계 권한 부여
- 기술: NestJS, TypeORM, MySQL, RBAC
- 임팩트: 실패율/수기 대응을 줄이고 운영 속도를 개선

## AWS 인프라 개편

- 소속/도메인: 뽀득 (2025.04 ~)
- 배경: 인수중·TMS 이관 완료 후 불필요해진 전용 Windows EC2와 RDBMS를 운영 종료하고 메인 RDS로 통합했습니다.
- 기여: 도메인 설계, API 구현, 배포까지 엔드투엔드로 수행
- 기술: AWS EC2, RDS
- 임팩트: 연간 약 400만 원 인프라 비용 절감

## 기타

- 소속/도메인: 뽀득 (2025.04 ~)
- 배경: 운영 중 반복되는 병목과 수기 프로세스가 존재
- 기여: **업체 데이터 동기화**: Google Sheets → DB 주간 수기 동기화를 NestJS Cron + Sheets API로 자동화
- 기술: NestJS, TypeScript
- 임팩트: 실패율/수기 대응을 줄이고 운영 속도를 개선

## 해외 공급사 API 통합 — GOTAR

- 소속/도메인: 팀오투 (카모아) (2022.07 ~ 2025.03)
- 배경: 공급사마다 API 형식이 달랐고(REST / SOAP), 기존 모놀리스 구조에서는 공급사 하나가 느려지면 서버 전체가 영향을 받았습니다. 해외 공급사가 계속 늘어날 예정이라 구조를 바꿔야 했습니다.
- 기여: 도메인별(차종·차량·업체·지점·주문·예약) 공통 인터페이스(GOTAR 스키마)를 정의하고 npm private 패키지로 배포. 래핑 서버와 코어 서버가 같은 타입을 참조해서 타입 불일치를 원천 차단
- 기술: NestJS, TypeScript, ECS
- 임팩트: 7개국 → 71개국 / 400만 대

## DocumentDB 스토리지 비용 91% 절감

- 소속/도메인: 팀오투 (카모아) (2022.07 ~ 2025.03)
- 배경: AWS 비용을 항목별로 보다가 DocumentDB가 월 140만 원 나오는 걸 발견했습니다. 하루 최대 1,500건 예약 규모에 비해 과한 비용이었고, 원인은 서비스 초기부터 쌓인 수년치 데이터를 그대로 두고 있었던 것이었습니다.
- 기여: 사업부에 보존 기간 확인 후 TTL Index로 1개월 초과 데이터 자동 만료
- 기술: AWS DocumentDB, TTL Index
- 임팩트: 월 140만 → 12만 원 (91% 절감) / 데이터 유실 없음

## 렌터카+호텔 패키지 — 보상 트랜잭션

- 소속/도메인: 팀오투 (카모아) (2022.07 ~ 2025.03)
- 배경: 렌터카 예약할 때 호텔도 같이 팔자는 사업 요구였습니다. 기존에 렌터카 예약과 결제가 하나의 DB 트랜잭션에 묶여 있어서 호텔 도메인을 끼울 수 없었습니다.
- 기여: 사가 패턴은 두 단계짜리 흐름에 오케스트레이터까지 만드는 건 과하다고 판단해서 보상 트랜잭션 선택

- 기술: NestJS, MySQL, REST Webhook, DLQ
- 임팩트: 패키지 상품 출시 / 호텔 매출 내 패키지 비중 60%

## 지점 마스터 시스템

- 소속/도메인: 팀오투 (카모아) (2022.07 ~ 2025.03)
- 배경: 6개 공급사의 지점 데이터가 전부 달랐습니다. 같은 인천공항인데 "Incheon Airport", "ICN Int'l", "인천공항 T1"으로 제각각 들어오니 통합 검색이나 가격 비교가 안 됐습니다.
- 기여: 마스터 지점 테이블 + 공급사별 매핑 테이블 구조로 6개 공급사 데이터를 정규화. 신규 공급사는 매핑만 추가 하면 됨
- 기술: NestJS, MySQL, OpenAI Batch API
- 임팩트: 타깃 지역 일 평균 예약 200 → 400건

## 기술 스택

- Backend: Node.js, NestJS, TypeScript
- Database: MySQL, MongoDB, Redis, PostgreSQL
- Infrastructure: AWS, GCP

## 오픈소스/대외 기여

### Claw-Empire — AI 에이전트 오케스트레이션 플랫폼

- 소개: GreenSheep01201/claw-empire (<https://github.com/GreenSheep01201/claw-empire>) (★106) — AI 코딩 에이전트를 가상 회사처럼 오케스트레이션하는 프로젝트. 직접 쓰다가 불편한 부분을 고쳐서 PR을 올렸습니다.
- 기여 포인트: #23 미팅 프롬프트 토큰 절감 / #26 안전 자동 업데이트 시스템 (Merged)

## 학력

- 코드캠프 Node.js 백엔드 부트캠프 | 2022.03 - 2022.07