

API부터 UI까지 연결하는 개발자

빠른 UI, 견고한 API, 실용적인 시스템을 설계합니다. API 설계와 실시간 협업에 집중합니다.



ABOUT ME

나에 대하여

소개

- AI/RAG에 집중하는 풀스택 엔지니어
- 실시간 협업 도구 구축을 즐김
- 신뢰성과 UX를 중요시

핵심 역량

- End-to-end 책임감 (UI → 인프라)
- API 설계 + 데이터 모델링
- Docker/AWS/Nginx로 DevOps

주요 성과

- CRDT 기반 라이브 리뷰 도구 구축
- AI 기반 글쓰기 플랫폼 출시
- AWS에 풀스택 앱 배포

연락하기

이메일: hy1x1mn@gmail.com

GitHub: github.com/hy2min

LinkedIn: linkedin.com/in/hy2min



기술 스택

분야별로 그룹화된 기술들입니다. 각 기술은 실제 프로젝트에서 사용한 경험을 반영합니다.

Frontend

TypeScript

React

Zustand

React Query

WebRTC

Backend

Spring Boot

FastAPI

REST APIs

OpenAPI/Swagger

WebSocket

PostgreSQL

MySQL

Redis

DevOps & Cloud

Docker

Docker Compose

Nginx

AWS EC2/S3

GitHub Actions

CI/CD

AI & Data

OpenAI API

LangChain

RAG

Vector DB

Pinecone

🤝 Collaboration

Notion

Slack

Figma

Jira

Ouroboros • OpenAPI Specification Management & Mock Server

OpenAPI 3.1.0 기반 REST API 명세서 관리 및 Mock Server 라이브러리.

The screenshot shows the Ouroboros API management tool's interface. On the left, there's a sidebar with tabs for REST and WebSocket, and buttons for All, Mock, and Completed. Below that is a tree view of controllers: TEST-CONTROLLER (selected), COMPARE-CONTROLLER, CHAT-CONTROLLER, and AUTH. The TEST-CONTROLLER node has a POST /api/test endpoint selected. The main area is the 'API Spec' tab, which displays the Method & URL (POST /api/test), Tags (test-controller), Request and Response sections (both currently empty), and a Body section with fields: email (string), password (string), name (string), age (integer), and phone (string). The body is defined with application/json type. At the bottom right are 'Edit' and 'Delete' buttons.

🔗 프로젝트 링크

GitHub: <https://github.com/whitesnakegang/ouroboros> **Demo:** <https://ouroboros.co.kr>



프로젝트 개요

Java Spring Boot + React 기반의 API 명세서 작성부터 Mock Server, 테스트, 성능 분석까지 지원하는 통합 개발 도구. OpenAPI 3.1.0 스페 준수 및 자동 검증 기능 제공.



기술 스택

Java

Spring Boot

React



담당 역할

- React + TypeScript 기반 웹 UI 전면 개발 및 아키텍처 설계
- OpenAPI 명세서 작성/편집 인터페이스 및 실시간 미리보기 기능 구현
- Mock Server 테스트 인터페이스 및 Request/Response 스키마 관리 UI 개발
- Call Trace 성능 추적 시각화 및 트리 구조 자동 확장 기능 구현
- 코드 스니펫 생성 기능 및 다국어 지원 인증 처리
- WebSocket/STOMP 명세서 관리 UI 및 실시간 상태 동기화 구현
- JSON 에디터 통합 및 사용자 경험 최적화



트러블슈팅

1. Call Trace 트리 구조 경로 탐색 및 확장

문제 Call Trace 모달에서 특정 메서드를 클릭했을 때, 해당 span이 트리 깊숙이 있어 사용자가 수동으로 부모 노드들을 모두 확장해야 하는 UX 문제가 발생했습니다.

원인 중첩된 트리 구조에서 특정 노드를 찾고, 루트부터 해당 노드까지의 모든 경로를 자동으로 확장하는 로직이 없었습니다.

해결 DFS 알고리즘을 구현한 `findSpanPath` 함수를 개발하여 대상 노드까지의 전체 경로를 자동으로 찾고, React state로 관리하여 모든 부모 노드를 한 번에 자동 확장하도록 구현했습니다. `initialExpandedSpanId` 변경 시에만 실행하여 성능을 최적화했습니다.

근거: front/src/features/spec/components (Call Trace 관련 컴포넌트), DFS 알고리즘 구현

2. FormDataBodyForm 무한 업데이트 루프 해결

문제 FormDataBodyForm 컴포넌트에서 `Maximum update depth exceeded` 에러가 발생하고 브라우저가 응답하지 않는 문제가 있습니다.

원인 useEffect에서 `formData`가 변경될 때마다 `onChange`를 호출하고, `onChange`가 부모 컴포넌트의 상태를 변경하여 다시 `value` prop으로 전달되었습니다.

해결 useRef를 사용하여 이전 값을 추적하는 `prevValueRef`를 구현했습니다. 실제로 값이 변경되었을 때만 업데이트하도록 조건을 추가하고, `handleFormDataChange`에서 `useEffect` 대신 직접 `onChange`를 호출하도록 변경했습니다.

근거: front/src/features/testing/components/RequestBodyForm.tsx, FormDataBodyForm 컴포넌트, useRef를 활용한 상태 관리 최적화

3. Node.js 전용 라이브러리 브라우저 호환성 문제

문제 코드 스니펫 생성 기능을 위해 `openapi-snippet` 라이브러리를 사용했으나, 브라우저에서 `Module 'stream' has been externalized`, `global is not defined` 등의 에러가 발생했습니다.

원인 `openapi-snippet`은 Node.js 환경을 가정하고 설계된 라이브러리로, Node.js 전용 모듈(`stream`, `string_decoder`, `qs` 등)에 의존합니다.

해결 `vite-plugin-node-polyfills`를 설치하고 Vite 설정에 추가하여 Node.js 모듈을 브라우저에서 사용할 수 있도록 폴리필을 제공했습니다.

근거: front/vite.config.ts, front/src/features/spec/components/CodeSnippetPanel.tsx, vite-plugin-node-polyfills 통합

OtterReview • Live Code Review & Whiteboard

Yorkie CRDT + tldraw + WebRTC for real-time collaboration and reviews.

The screenshot shows the OtterReview application interface. At the top, there is a navigation bar with a logo, user information, and buttons for connecting to a repository and logging out. The main title is "코드 리뷰 워크스페이스" (Code Review Workspace). Below the title, a subtitle says "Repository와 Pull Request를 효율적으로 관리하세요".

The interface is divided into two main sections:

- Repository:** Shows two public repositories: "PageOn" and "reviewOtter", both managed by "otterreview".
- Pull Request:** Shows three pull requests:
 - "accessTokenGenerate 메서드 수정" by otterreview/PageOn, created by kidreamx 5 days ago. Status: 병합 가능, 리뷰 0.
 - "[FEATURE]: Swagger 설정 및 카카오 유저 정보 수..." by otterreview/PageOn, created by kidreamx 1 minute ago. Status: 병합 가능, 리뷰 0.
 - "[FEATURE]: 다이어리 엔티티 및 서비스 구현 ..." by otterreview/PageOn, created by kangboom 1 day ago. Status: 병합 가능, 리뷰 0.

🔗 프로젝트 링크

GitHub: <https://github.com/hy2min/otterreview>



프로젝트 개요



React (Vite) + Zustand + Yorkie + tldraw; Spring Boot backend. Live code sharing, whiteboarding, chat, and secure sessions.



기술 스택

React

Zustand

Yorkie

tldraw

OpenVidu

STOMP

Spring Boot



담당 역할

- CRDT 연동/문서 키 관리(attach/detach 안정화)
- 화이트보드 도구 구성(tldraw)
- 오디오 룸 토큰/세션 만료 복구
- 채팅(WS) + 보안헤더/코르스 정리



트러블슈팅

1. 연결 실패(음성 채팅) - WebRTC / Audio Chat

문제 세션 연결 불가, 재시도 루프

원인 OpenVidu 토큰 연결 지연·타임아웃

해결 Promise.race 기반 10s 타임아웃, 재시도/에러 상태 관리 추가

근거: fix: 음성채팅 연결 에러 (815fd912, 2025-08-13), useWebRTC.js 타임아웃 로직

2. 참가자 목록 동기화 누락

문제 트랙 소실/유저명 미표시

원인 connectionCreated/Destroyed 이벤트 처리 부재

해결 세션 이벤트 리스너 추가, connectedParticipants 상태 동기화

근거: useWebRTC.js 이벤트 핸들러 추가

Drug Service • RAG-based Medication Guide

LLM + RAG chatbot for safe medication information and contraindications.

The screenshot shows a mobile application interface for 'AI Pharmacy Consultation'. On the left, there's a profile section with fields for age (27), gender (Female), and current medications (Aspirin, Ibuprofen, Metformin). A message from the bot says: '안녕하세요! 궁금한 점을 입력해 주세요. 친절하게 답변드리겠습니다.' (Hello! Please enter your question. I will answer it politely.) The user responds with '[타이레놀정500mg] [가스모틴정] 함께 복용해도 될까요?' (Is it okay to take Tylenol 500mg and Gasomotin together?). The bot replies with '네' (Yes). Below the messages, there are buttons for saving the profile and a microphone icon for voice input.

🔗 프로젝트 링크

GitHub: https://github.com/hy2min/drug_service



프로젝트 개요



*Builds an index from public drug datasets with retrieval chains and reference citations.
Backend framework pluggable.*



기술 스택

LLM

RAG

Vector DB

FastAPI/Django

OpenAI



담당 역할

- RAG 파이프라인 설계(인덱싱·검색·답변 체인)
- 의약품 데이터 파싱/정규화
- 안전성/금기/상호작용 가이드 출력 포맷



트러블슈팅

1. 한글 PDF 파싱 인코딩

문제 EUC-KR 문서 파싱 오류

원인 전처리 미흡

해결 UTF-8 강제 변환·예외 처리 파이프라인

근거: backend/requirements.txt(PDF 관련), 데이터 소스 comprehensive_drug_safety_docs.csv

2. RAG 검색 지연

문제 질의 응답 지연

원인 임베딩/벡터 쿼리 대기

해결 Top-K 축소·LRU 캐시·요약 단계 캐시

근거: backend/app.py 구조

Orakgarak • Vocal Analysis & AI Covers

Interactive vocal analysis and AI cover generation with community features.



🔗 프로젝트 링크

GitHub: <https://github.com/hy2min/orakgarak>

📋 프로젝트 개요

“ React (Vite + Tailwind + MUI) frontend, Spring Boot backend. Features include segment analysis, AI cover generation, likes/bookmarks, and deployment via Docker Compose. ”

🛠️ 기술 스택

React

Tailwind

MUI

Framer Motion

Zustand

Spring Boot



담당 역할

- 보컬 음역대 분석 게임(Phaser.js) 로직 및 시각화 구현
- AI 커버 생성 UI/비교모드 및 이미지 프리셋 로직 개발
- 앨범 CRUD, 피드·댓글·좋아요 API 연동 및 프론트 UI 구성
- Spring Boot + Docker Compose 배포 환경 구성 및 모니터링 설정



트러블슈팅

1. 음성 분석 지연

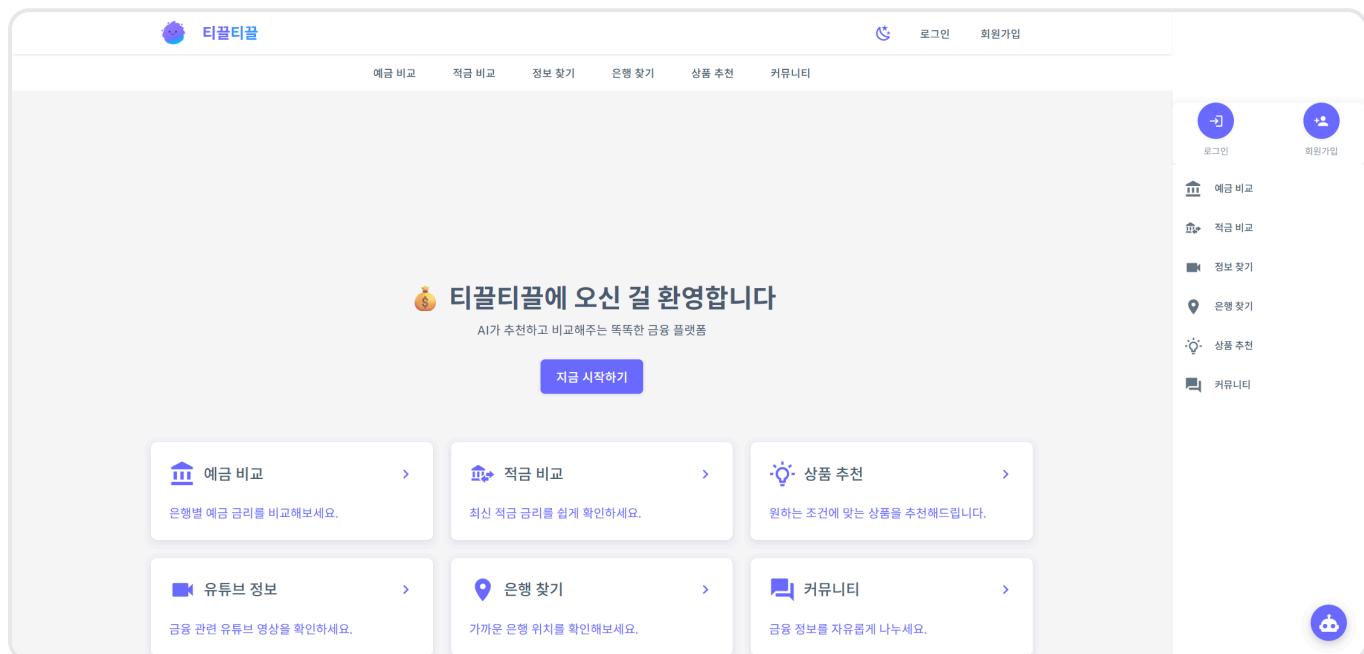
문제	분석 대기 시간 과다
원인	Python voice_analysis 파이프라인 단일 처리
해결	비동기 큐·배치 처리 분리 제안, 결과 폴링형 API 제안

근거: python/voice_analysis/requirements.txt, python/main.py

제공된 아카이브에 .git 이력 부재 → 커밋 로그 직접 확인 불가. 코드 구조/의존성 근거로 정리.

TikkleTikkle • Finance Recommender & Community

FSS API-based product recommendations, Kakao Maps local community, and an AI chatbot.



🔗 프로젝트 링크

GitHub: <https://github.com/hy2min/tikkletikkle> **Demo:** <http://tikkle.r-e.kr>

📋 프로젝트 개요

Django backend with React frontend, deployed to EC2 with Docker + Nginx. Includes recommendation flows, validation, and map-based discovery.

🛠️ 기술 스택

Django

React

Docker

Nginx

AWS EC2

OpenAI

Kakao Maps



담당 역할

- 금융 추천 로직 및 외부 API 연동
- 커뮤니티/지도 기능 구현
- AI 챗봇 프롬프트·플로우 구성
- 배포 파이프라인 일부 구성 및 CORS/환경변수 정리



트러블슈팅

1. 모델 입력 스케일 불일치

문제 예측치 분산 과대

원인 전처리 불일관

해결 입력 스케일러 고정·검증 단계 추가

근거: final_pjt_back/requirements.txt(ML 스택), 금융 입력 변수 특성

2. 로컬/배포 환경 차이

문제 브랜치별 동작 차이

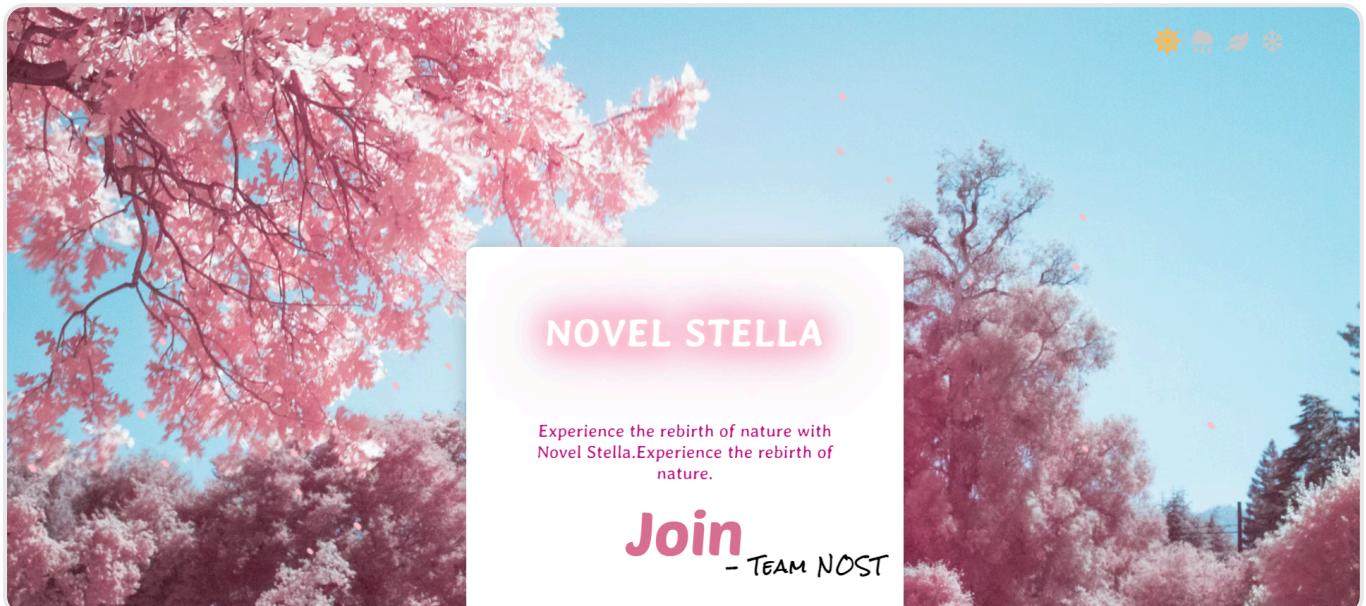
원인 master(배포) vs release/local(로컬) 분기

해결 .env 프로필 분리·CI 환경 행렬

근거: README.md(브랜치 안내)

NOST • AI Writing Community

AI suggests outlines and variations with translation support and community sharing.



🔗 프로젝트 링크

GitHub: https://github.com/hy2min/nost_service

☰ 프로젝트 개요

“ React + Django + PostgreSQL. Uses LangChain/OpenAI/DALLE/DeepL. Handles content creation flow and collaboration features.

🛠️ 기술 스택

React

Zustand

Django

PostgreSQL

AWS

LangChain

담당 역할

- 생성 파이프라인 설계 및 프롬프트 엔지니어링
- 피드/댓글 등 커뮤니티 흐름 설계
- 다국어(번역) 유틸/컴포넌트 정리

트러블슈팅

1. 대용량 텍스트 저장 성능

문제	본문 저장 시 응답 지연
원인	단일 모델 직접 저장
해결	본문 외부 스토리지 분리(S3)-메타데이터 DB 저장 제안

근거: backend/README.md 구조, 장문 생성 서비스 특성

2. 이미지 생성 실패율

문제	이미지 실패/429
원인	생성 API 쿼터 초과
해결	큐-콜다운-리트라이, 사용자별 레이트리밋

근거: frontend 비동기 호출 구조, 생성형 워크플로우