01
010
110 VanillaByte

주식회사 바닐라바이트

VanillaOne

본질에 집중합니다. 단 하나의 완성된 시작점

Vanilla One은 엔터프라이즈 통합 풀스택 개발 스위트로 단순한 코드 템플릿을 넘어서 시스템 통합 개발의 레시피를 제공합니다.

Enterprise Solution powered by Next.js + Spring Boot + PostgreSQL w/ vLLM + Gemma4

Vanilla One: 단 하나의 완성된 시작점

엔터프라이즈 통합 풀스택 개발 스위트

Next.js 16.x | Spring Boot 4.x | PostgreSQL

왜 Vanilla One 인가?

- **소스코드 파편화 종결:** 전자정부 표준 프레임워크 기반의 일관된 개발 방법론 제시
- **초고속 데이터 처리:** 대용량 데이터에 대해 검증된 컴포넌트와 데이터 연동 표준화
- **AI 가속 개발:** Swagger 연동 소스 자동 생성 및 LLM(로컬 모델, Ollama, vLLM) 지원 도구 제공

Enterprise Operational Excellence

비즈니스 연속성과 운영 효율을 위한 통합 관리 환경

권한 및 조직/사용자 관리

RBAC Admin

- RBAC 기반 정교한 권한 생성 및 관리
- 메뉴-권한 매핑을 통한 유연한 접근 제어
- 조직도 기반의 체계적인 사용자 관리 시스템

운영 최적화 파운데이션

Built-in Functionalities

- 글로벌 서비스를 위한 다국어 적재 시스템
- 전사 공통 코드 및 첨부파일 관리 표준화
- 공공 데이터 또는 기간계 시스템 데이터 동기화 표준

지능형 배치 자동화

Batch Runner

- 공공데이터 수집 및 실시간 동기화 자동화
- 배치 실행 상태 조회 및 즉시 실행 기능
- 대용량 데이터 처리에 최적화된 스케줄링

로그 분석 어시스트

Log Visualizer

- API 및 배치 수행 로그 데이터의 직관적 시각화 지원
- 신속한 문제 파악을 위한 고급 검색 엔진
- AI 연동을 통한 분석 및 장애 징후 탐지

High Performance Frontend (Next.js 16.x)

엔터프라이즈급 UI 성능과 일관된 개발 경험의 완성

프론트엔드 핵심 특기사항

- **초고속 대용량 데이터그리드:** 10만 건 이상의 로우를 지연 없이 렌더링하고 실시간 수정/조회가 가능한 고성능 그리드로써 Vanilla Table 컴포넌트 탑재
- **전자정부 표준 가이드 준수:** 트렌디한 기술셋(Next.js)과 안정적인 공공 표준 방법론을 결합하여 유지보수성 극대화
- **Production-Ready 자산:** 대형 프로젝트에서 검증된 커스텀 훅(Hooks), 유틸리티 함수 및 테마 시스템 내장
- **완벽한 DX(개발자 경험):** Storybook 기반 컴포넌트 명세와 엄격한 ESLint 설정으로 팀원 간 코드 파편화 원천 차단

Robust & Scalable Backend (Spring Boot 4.x)

고성능 마이크로서비스 아키텍처와 엔터프라이즈급 보안 프레임워크

백엔드 핵심 특기사항

- **MSA 기반 확장성:** 서비스별 독립적 배포 및 확장이 유연하게 가능한 마이크로서비스 구조
- **Spring Security & JWT:** RBAC 기반의 정교한 권한 제어와 표준화된 토큰 인증 체계
- **QueryDSL & JPA:** 대용량 데이터 쿼리 최적화 및 타입 안정성이 보장된 데이터 레이어
- **자동화된 API 명세:** Swagger(OpenAPI 3) 연동을 통한 실시간 문서화 및 프론트엔드 협업 최적화

Intelligent Development Workflow

AI와 자동화가 결합된 차세대 엔터프라이즈 개발 패러다임

생산성 극대화 전략

- **OpenAPI / Swagger 자동 연동:** 백엔드 설계와 동시에 프론트엔드 API 모듈 소스 100% 자동 생성
- **Storybook 표준 가이드:** 컴포넌트 명세화로 개발 파편화를 방지하고 실시간 테스트 환경 제공
- **AI 기반 코딩 가속기:** LLM(Ollama, Gemini, Claude 등 고객사 사용 API에 연결) 연동 CLI로 보일러플레이팅 시간 제로화
- **엄격한 DX:** 팀원 모두가 한 사람이 만든 결과물처럼 클린 코드를 유지하는 표준 설정

Vanilla One 도입 효과 분석

기존 개발 방식 대비 생산성 및 품질 혁신 지표

도입 지표	기존 개발 방식	Vanilla One 도입 후
초기 아키텍처 셋업	인프라, 백엔드, 프론트 셋업에 수주 소요	인프라 가이드로 단 몇 시간 만에 완료
프론트엔드 품질	개발자 역량에 따른 성능 파편화 및 렌더링 저하	표준 가이드 및 데이터그리드로 균일한 초고속 성능 확보
유지보수 및 확장성	버전 업그레이드 및 파편화로 유지보수 비용 폭증	국내 업계 표준 준수 및 MSA 구조로 용이한 확장
인터페이스 조율	API 규격서 작성 및 소통 리소스 낭비	Swagger 기반 연동 소스 자동 생성으로 조율 제로

Vanilla One 도입 사례

다양한 산업군의 비즈니스 가치를 극대화한 성공적인 구축 케이스

GM Warehouse System

물류 시스템 및 재고 관리 등 입출력 최적화

부품 재고 관리 및 발주/주문 시스템 통합:
APS, PMS, AOMS, IWS 등 부품 창고 시스템의 PowerBuilder 기반 레거시 애플리케이션을 웹 애플리케이션으로 통합했습니다.

물류 리드타임 단축:
바쁜 업무 중 조회 및 입력이 수월하고 빠르게 동작하며 기존 C/S 대비 오류 없이 즉각적으로 반응하는 사용자 인터페이스를 구축하여 고객사 업무 생산성 증대를 이끌어 낼 수 있었습니다.

유지보수 비용 절감:
Vanilla One 표준 개발 방법론 도입으로 백엔드팀과 프론트엔드팀 각각 일관되고 매뉴얼화된 코드 템플릿 적용으로 유지보수 난이도를 낮추고 작업 시간 단축할 수 있었습니다.

AirZeta 항공유 관리시스템

항공유 구매 및 정산 프로세스 자동화

급유 및 정산 업무 효율화:
Excel 문서 기반 수작업 위주의 급유 입력 및 발주, 정산, 추정 등을 하나의 시스템에 통합 및 연계하여 고객사의 업무 효율을 드라마틱하게 끌어올릴 수 있었습니다.

ESG, SAF 등 항공유 규제 선제적 대응:
탄소배출량 산정용 원천 데이터 확보 및 보고 체계 마련

개발 공수 및 유지보수 비용 절감:
Vanilla One의 아키텍쳐 구조 및 개발 방법론 적용으로 백엔드와 프론트엔드의 개발 연속성을 가속할 수 있었으며, 공통 모듈과 단위업무를 효율적으로 분리함으로써 개발기간을 상당히 단축했습니다.

* 각 사례는 Vanilla One의 표준 컴포넌트와 프레임워크 및 함께 제공하는 개발 방법론을 기반으로 구축되었습니다.