



Production-ready Java backends, **generated** from a SQL schema.

Hexagonal architecture for global structure — **vertical slicing** so every entity is a self-contained unit, not chaos across packages. A **production-ready, 95–100% tested backend** delivered in days, not months. Deploy as a **vertical-sliced monolith** or **vertical-sliced microservices** from the same schema. No lock-in.

Java 25

Spring Boot 4.1

Hexagonal

Vertical slicing

OpenAPI

95–100% tests

BUILD IT BY HAND

Two mid-level developers, from a blank page.

- × **6–9 months** before the structural layer is shippable
- × Weeks on scaffolding & test plumbing — **before one business rule**
- × **20–40% coverage** under deadline; rarely measured systematically
- × “Hexagonal” on paper — **no vertical slices**, layers shared across entities
- × Dev + AI: **different architecture every sprint**

GENERATE IT WITH ALVEOFORGE

Guided end-to-end — same SQL, same output, every time.

- ✓ **Days, not months** — live Swagger preview before payment
- ✓ **Hexagonal + vertical slicing** — ports & adapters **inside each entity slice**
- ✓ Team on **domain logic from day one** — not scaffolding
- ✓ **95–100% test coverage** — methods, lines & branches
- ✓ **Large generated test suite** — all green at handoff

TIME TO PRODUCTION-READY

Months → **Days**

TEST COVERAGE

20–40% → **95–100%**

ARCHITECTURE

Hexagonal · vertical slices

SCALE

300+ tables · 1,000 → microservices

INDUSTRIAL PATTERNS — FROM YOUR DDL, NOT FROM PROMPTS

DATA MODEL

- PostgreSQL DDL: schemas, FK, composite PK, UUID, ENUM
- Table **inheritance** (INHERITS / JOINED)
- Relations **1-n · n-n · 1-1** · join tables
- Multi-schema tables · identical names across schemas resolved
- Paginated lists · nested reads · bulk operations

ARCHITECTURE & STORES

- **Hexagonal** — domain isolated from database & HTTP
- **Vertical slicing** — one slice per entity; without it, hexagonal packages bleed together
- Each slice: domain · application · infrastructure — self-contained
- Vertical-sliced **monolith** or **microservices** — same schema
- SQL + optional Mongo · Redis **cache** · transactional outbox

QUALITY ASSURANCE

- Unit & component tests per entity
- Database integration tests (SQL & Mongo)
- Full HTTP contract test suite
- End-to-end tests with Docker (PostgreSQL + Mongo)
- Email flows verified in tests · HTML reports included

WHAT YOU RECEIVE

COMMERCIAL

- **Try before you buy** — live Swagger; every endpoint testable pre-payment
- Private repo — code **100% yours** · no vendor lock-in
- **30-day author support** after handoff
- **Regenerate** when your SQL changes — deterministic output

TECHNICAL DELIVERY

- **95–100% coverage report** + HTML test reports with the delivery
- Runnable **hexagonal, vertical-sliced monolith** or **microservices** — same schema, your choice
- Setup & security **documentation** included
- Validated at **300+ tables** scale · up to **1,000 → microservices**

WHAT STAYS WITH YOU

A **production-ready backend** — not a starter kit or Spring Initializr: hexagonal structure with a **vertical slice per entity**, runnable and deployable. Business rules and third-party integrations are yours. Row-level authorization stays with your team — authenticated CRUD by default, not per-row policy unless you extend it.

Guided delivery: schema review together → align modules & stores → live preview → private repo when you're satisfied.

Your schema is **confidential** — used only to build your backend. NDA on request.

Hexagonal structure, vertical slices — **the same output, every build.**