


Golang Clean Architecture


Production-Ready SaaS Starter Kit - JWT Auth, Multi-Tenancy & Subscription Management


PROJECT OVERVIEW


Golang Clean Architecture SaaS Template adalah starter kit production-ready untuk membangun aplikasi SaaS dengan Go, mengikuti prinsip Clean Architecture. Template ini menghilangkan kerja boilerplate selama berminggu-minggu dengan menyediakan fitur SaaS esensial secara out-of-the-box: **JWT Authentication** (access & refresh token), **Multi-Tenancy** dengan organization-first design, **Subscription Management** dengan tiered plans, **Email Verification** async, serta **UUID Primary Keys** dan **Soft Delete** untuk database design yang aman dan scalable.

KEY FEATURES

- 

JWT Authentication
Access & Refresh token, email verification
- 

Multi-Tenancy
Organization-first dengan RBAC
- 

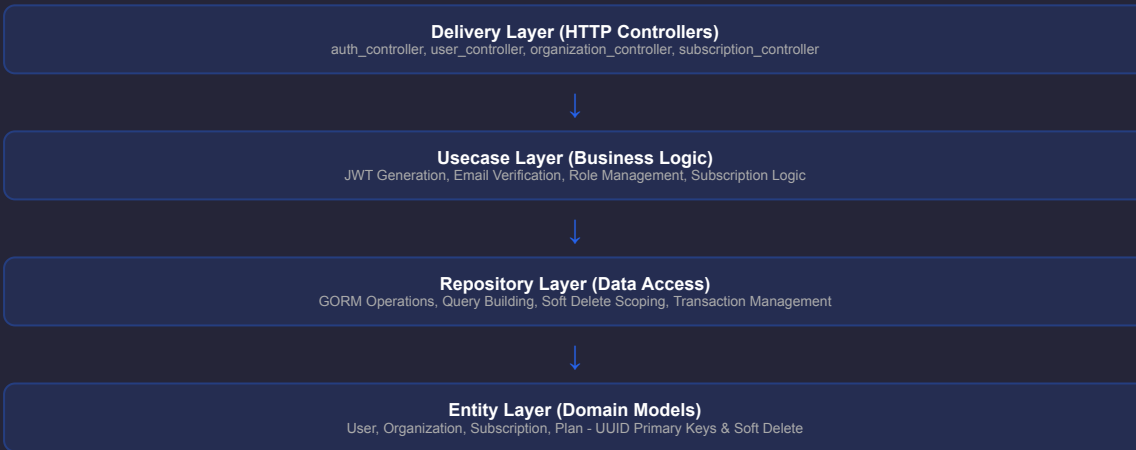
Subscription Plans
Free, Pro, Enterprise tiers
- 

59 Unit Tests
Comprehensive test coverage

TECH STACK

- Go 1.23Language
- Fiber v2Framework
- GORMORM
- MySQL 8.0Database
- JWTAuth
- DockerContainer
- ViperConfig
- LogrusLogging
- TestifyTesting

CLEAN ARCHITECTURE LAYERS



DATABASE SCHEMA

```
// Core Tables with UUID & Soft Delete
organizations (id, name, slug, logo_url)
users (id, org_id, email, password, verified)
organization_members (id, org_id, user_id, role)
plans (id, name, price, limits, features)
subscriptions (id, org_id, plan_id, status)
audit_logs (id, action, resource, old/new_value)
```

API ENDPOINTS

```
// Authentication
POST /api/v1/auth/register
POST /api/v1/auth/login
POST /api/v1/auth/refresh
POST /api/v1/auth/verify-email
// Protected Routes
GET /api/v1/users/me
GET /api/v1/organizations/:id
```

PROJECT STRUCTURE

Architecture

```
cmd/web/           - Application entry point
internal/
├── config/         - Configuration and bootstrap
├── entity/         - Domain entities (database models)
├── model/          - Request/Response DTOs
├── converter/      - Entity to DTO converters
├── repository/     - Data access layer
├── usecase/        - Business logic layer
├── delivery/       - Presentation layer
├── http/           - HTTP handlers and routes
└── middleware/    - Auth middleware

pkg/
├── jwt/           - JWT service
├── email/         - Email service with HTML templates
├── templates/     - Email HTML templates (embedded)
├── db/migrations/ - Database migration files
├── docs/          - Additional documentation
├── CONFIGURATION.md - Configuration guide
├── EMAIL_SERVICE.md - Email service & async patterns
└── TESTING.md     - Testing documentation
```

Folder Structure

Go Clean Architecture SaaS API

1.0.0 OAS 3.0

<http://openapi-prod.example.com>

Production-ready SaaS API with JWT authentication, multi-tenancy, subscription management, and email verification
[Contact API Support](#)

Servers [http://localhost:3000 - Development server](#) [Authorize](#)

Health

Health check endpoints

- GET /health Health check
- GET /ready Readiness check

Authentication

Auth endpoints (register, login, logout, refresh, verify email)

- POST /api/v1/auth/register Register new organization and user
- POST /api/v1/auth/verify-email Verify email address
- POST /api/v1/auth/resend-verification Resend verification email
- POST /api/v1/auth/login Login user
- POST /api/v1/auth/refresh Refresh access token