Introduction

- Project Overview
- Architecture

Project Overview

OnMarket Landing API is a comprehensive e-commerce backend framework built with Node.js and Express.js. It provides a complete marketplace solution with multi-vendor support, affiliate marketing, payment processing, order management, and real-time chat functionality.

Key Capabilities

- **Multi-vendor Marketplace**: Support for multiple stores with individual management
- **Product Management**: Complete CRUD operations with variants, categories, and inventory
- **Order Processing**: Full order lifecycle from cart to delivery tracking
- **Payment Gateway**: Integrated with Xendit for multiple payment methods
- **Affiliate Marketing**: 3-level referral system with commission tracking
- **Shipping Integration**: GoSend instant delivery and other courier services
- **Real-time Chat**: WebSocket-based customer-seller communication
- **Virtual Accounts**: Automated payment collection via bank virtual accounts
- **Review System**: Product reviews with media uploads
- **CMS Features**: Content management for banners, categories, and promotions

---

Tech Stack

Core Technologies

- **Runtime**: Node.js
- **Framework**: Express.js v4.21.0
- **Language**: JavaScript (ES6+)

Databases

- **PostgreSQL**: Primary relational database (via Sequelize ORM v6.37.3)
- **MongoDB**: Document storage for chats, affiliate data, and caching (via Mongoose v8.6.3)
- **Redis**: Caching and session management (v4.7.0)

Key Dependencies

Web Framework & Middleware

- `express` - Web application framework
- `cors` - Cross-origin resource sharing
- `helmet` - Security headers
- `morgan` - HTTP request logger
- `express-validator` - Request validation
- `passport` - Authentication middleware

Data Management

- `sequelize` - PostgreSQL ORM
- `mongoose` - MongoDB ODM
- `redis` - Redis client
- `pg` - PostgreSQL driver

Authentication & Security

- `jsonwebtoken` - JWT token generation
- `jwt-decode` - JWT token decoding
- `passport-http` - HTTP authentication strategy
- `helmet` - Security middleware

File Processing

- `multer` - File upload handling
- `minio` - Object storage (S3-compatible)
- `sharp` - Image processing
- `ffmpeg-static` & `fluent-ffmpeg` - Video processing
- `bwip-js` - Barcode generation

Document Generation

- `pdfkit` - PDF generation
- `pdfmake` - Advanced PDF creation
- `exceljs` - Excel file generation

External Integrations

- `xendit-node` - Payment gateway integration
- `axios` - HTTP client
- `nodemailer` - Email sending

- `socket.io` - WebSocket for real-time features

Task Scheduling

- `node-cron` - Cron job scheduling
- `bull` - Job queue management

Utilities

- `dayjs` - Date manipulation
- `lodash` - Utility functions
- `nanoid` - Unique ID generation
- `uuid` - UUID generation
- `ajv` - JSON schema validation
- `cli-color` - Console color output
- `dotenv` - Environment variable management

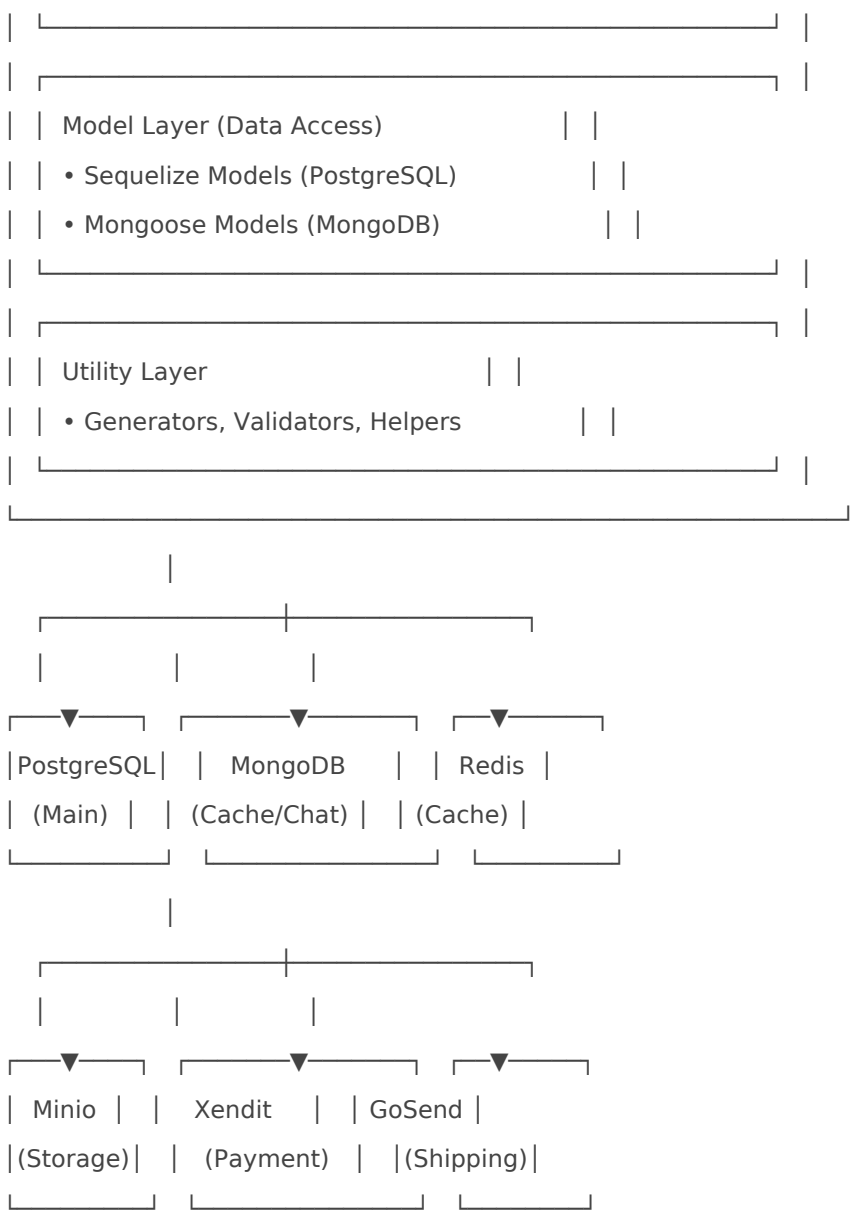
API Documentation

- `@scalar/express-api-reference` - Interactive API documentation

Architecture

Application Architecture





Request Flow

1. **Client Request** → Express server receives HTTP/WebSocket request
2. **Middleware** → Authentication, validation, logging
3. **Router** → Route matching and handler selection
4. **Controller** → Business logic execution
5. **Service/Model** → Data persistence and external API calls
6. **Response** → JSON response or WebSocket event emission