ALX PRODEV PROGRAMME

E-commerce Backend Project

Robust, Scalable, and Secure API Platform

Presented by: Ebunilo Igwilo
ebuniloigwilo@gmail.com

github.com/ebunilo/alx-project-nexus

www.ecommerce.igwilo.com/swagger/

Project Vision

The Challenge

Build a production-ready e-commerce backend that handles:

- 10,000+ products with variants
- 100+ concurrent users
- Secure payment processing
- Real-time inventory management

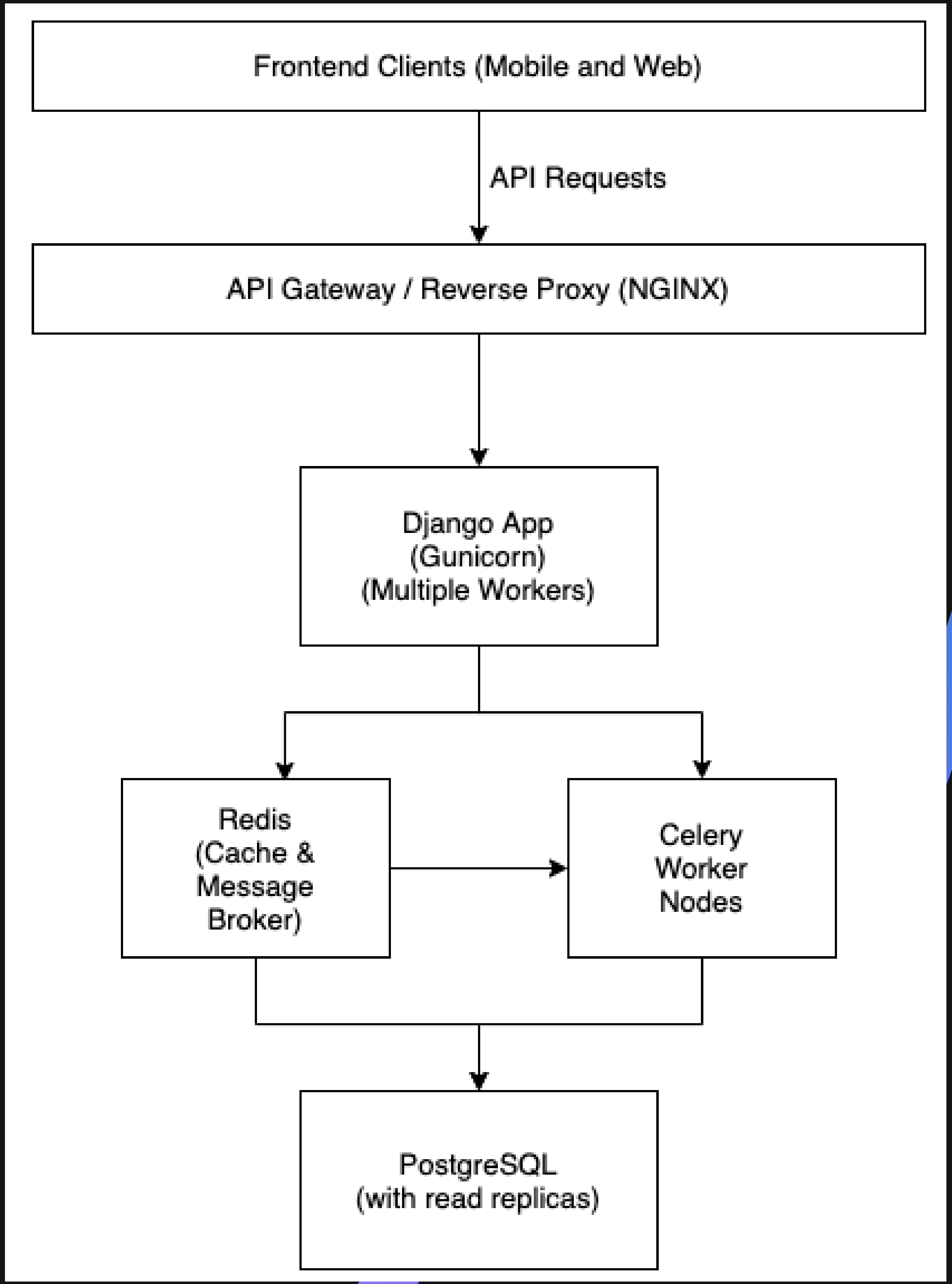
The Solution

A microservices-inspired Django monolith with:

- RESTful API design
- JWT authentication
- PostgreSQL + Redis stack
- Paystack payment integration

System Architecture

High-Level Architecture


Frontend Clients (Mobile and Web)
API Requests
API Gateway / Reverse Proxy (NGINX)
Django App (Gunicorn) (Multiple Workers)
Redis (Cache & Message Broker)
Celery Worker Nodes
PostgreSQL (with read replicas)

Technology Stack

Backend Framework

- Django 4.2+ - Battle-tested Python framework
- Django REST Framework - Robust API development
- PostgreSQL 14 - Relational database
- Redis 7 - Caching & message broker

Development & Deployment

- Docker & Docker Compose - Containerization
- Gunicorn - WSGI HTTP Server
- Nginx - Reverse proxy & load balancing

Authentication & Security

- JWT (SimpleJWT) - Stateless authentication
- bcrypt - Secure password hashing
- CORS headers - Cross-origin resource sharing

Payment Processing

- Paystack API - African-focused payments
- Webhook handling - Real-time payment verification

ER Diagram

Core Entity Groups:

- User Management - Customers, Admins, Profiles
- Product Catalog - Products, Categories, Variants
- Order Processing - Orders, Payments, Shipments
- Business Intelligence - Reviews, Analytics, Inventory

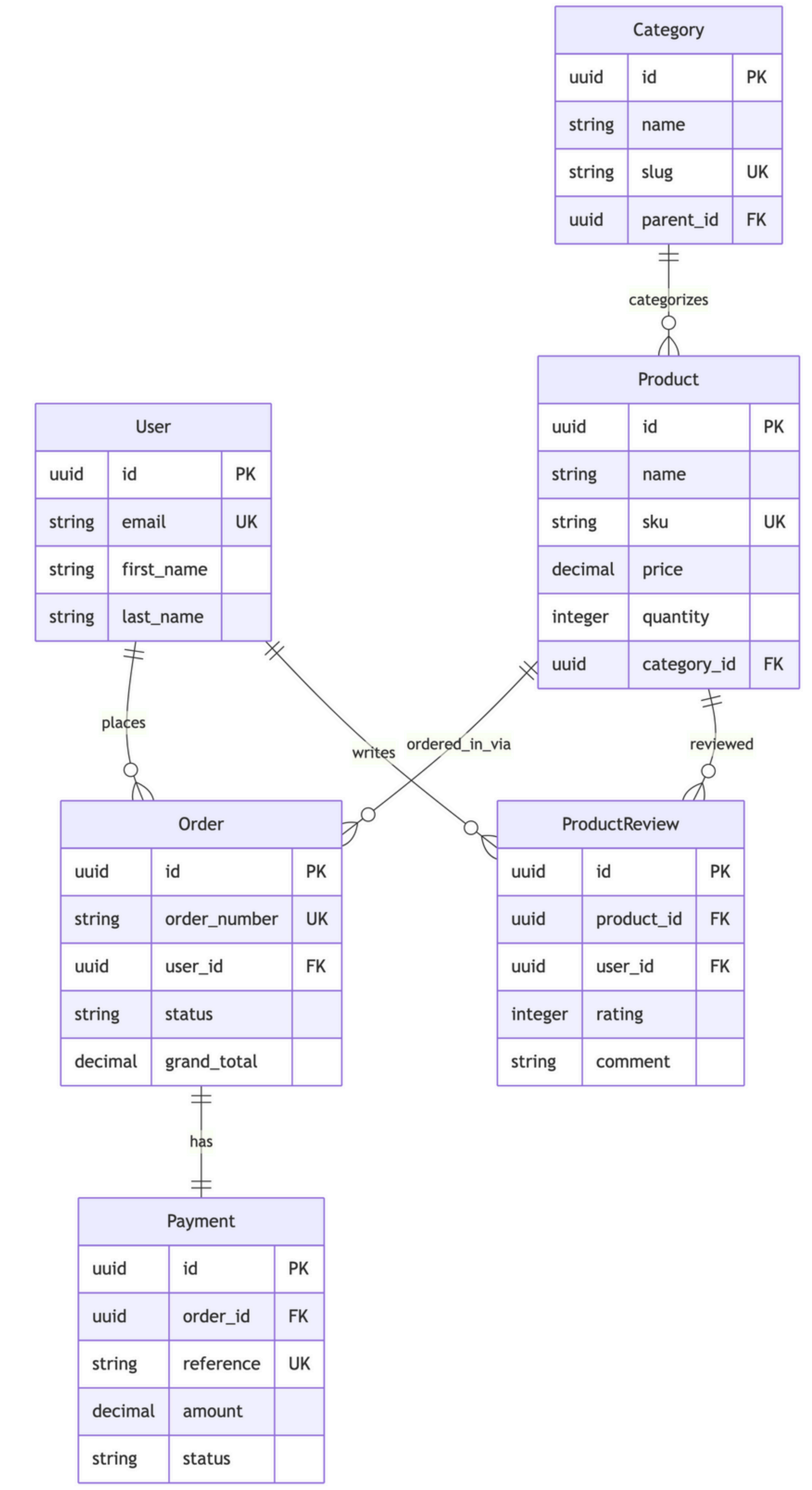
Key Design Principles:

✅ Normalized to 3NF - Eliminates data redundancy

✅ Indexed for Performance - Fast query execution

✅ Audit Trails - Track all critical operations

✅ Soft Deletes - Data recovery capability



Category		
uuid	id	PK
string	name	
string	slug	UK
uuid	parent_id	FK

User		
uuid	id	PK
string	email	UK
string	first_name	
string	last_name	

Product		
uuid	id	PK
string	name	
string	sku	UK
decimal	price	
integer	quantity	
uuid	category_id	FK

Order		
uuid	id	PK
string	order_number	UK
uuid	user_id	FK
string	status	
decimal	grand_total	

ProductReview		
uuid	id	PK
uuid	product_id	FK
uuid	user_id	FK
integer	rating	
string	comment	

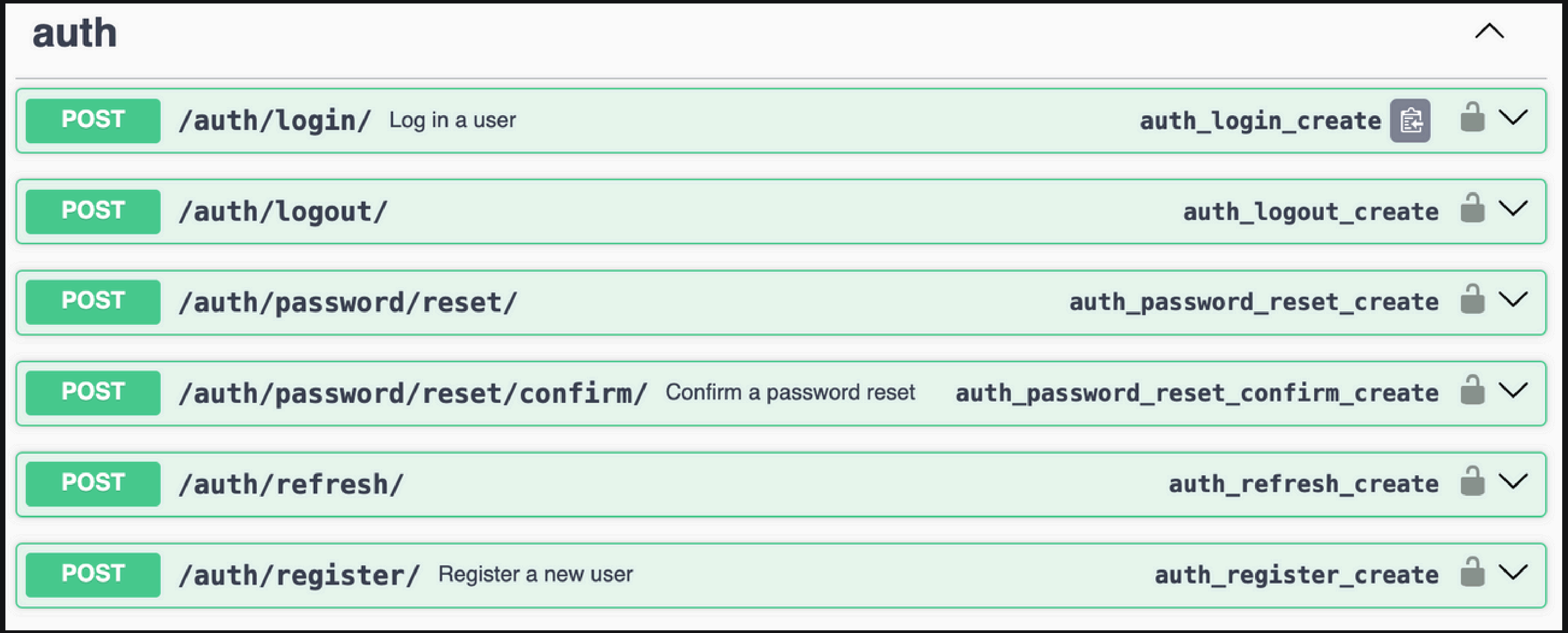
Payment		
uuid	id	PK
uuid	order_id	FK
string	reference	UK
decimal	amount	
string	status	

APIs:

Authentication & User Mgt.

User Registration & Login

- POST /api/auth/register/
- POST /api/auth/login/
- POST /api/auth/refresh/
- POST /api/auth/password/reset/



auth

POST /auth/login/ Log in a user auth_login_create

POST /auth/logout/ auth_logout_create

POST /auth/password/reset/ auth_password_reset_create

POST /auth/password/reset/confirm/ Confirm a password reset auth_password_reset_confirm_create

POST /auth/refresh/ auth_refresh_create

POST /auth/register/ Register a new user auth_register_create

Security Features

- JWT tokens with refresh mechanism
- Rate limiting 1000 requests/hour for authenticated users)
- Email verification for new accounts
- Password complexity enforcement
-

Profile Management

- GET /api/users/profile/
- PUT /api/users/profile/
- GET /api/users/addresses/
- POST /api/users/addresses/

Product Catalog API

Product Discovery

- GET /api/products/ # List with pagination
- GET /api/products/{slug}/ # Single product
- GET /api/products/search/ # Advanced search
- GET /api/products/categories/ # Category tree

Performance Optimizations:

- Full-text search using PostgreSQL tsvector
- Redis caching of search results
- Eager loading of related data
- Query optimization with proper indexes

Advanced Search Capability

```
{
  "q": "wireless headphones",
  "min_price": 50,
  "max_price": 500,
  "category": "electronics",
  "attributes": {"color": ["black",
"white"]},
  "sort_by": "price_asc",
  "page": 1,
  "page_size": 20
}
```

Shopping Experience API

Cart Management

- GET   /api/cart/   # View cart
- POST   /api/cart/items/   # Add item
- PUT   /api/cart/items/{id}/   # Update quantity
- DELETE /api/cart/items/{id}/   # Remove item

Checkout Flow

- POST   /api/cart/checkout/shipping/ # Set shipping
- POST   /api/cart/checkout/payment/   # Set payment
- POST   /api/cart/checkout/confirm/   # Confirm order

Real-time Calculaltions

- Dynamic pricing with tax calculation
- Shipping cost based on weight & destination
- Discount application with coupon codes
- Inventory validation before checkout

Order & Payment Processing

Order Management

- GET /api/orders/ # User's orders
- GET /api/orders/{id}/ # Order details
- POST /api/orders/{id}/cancel/ # Cancel order
- GET /api/orders/{id}/tracking/ # Shipping status

Payment Integration

- POST /api/payments/initialize/ # Start payment
- GET /api/payments/verify/{ref}/# Verify payment
- POST /api/payments/webhook/ # Paystack webhook
- POST /api/payments/refund/ # Process refund

Payment Methods

💳 Credit/Debit Cards (Visa, Mastercard)

🏦 Bank Transfers

📱 Mobile Money (MTN, Airtel, etc.)

💵 Cash on Delivery

Business Intelligence

Real-time Dashboards

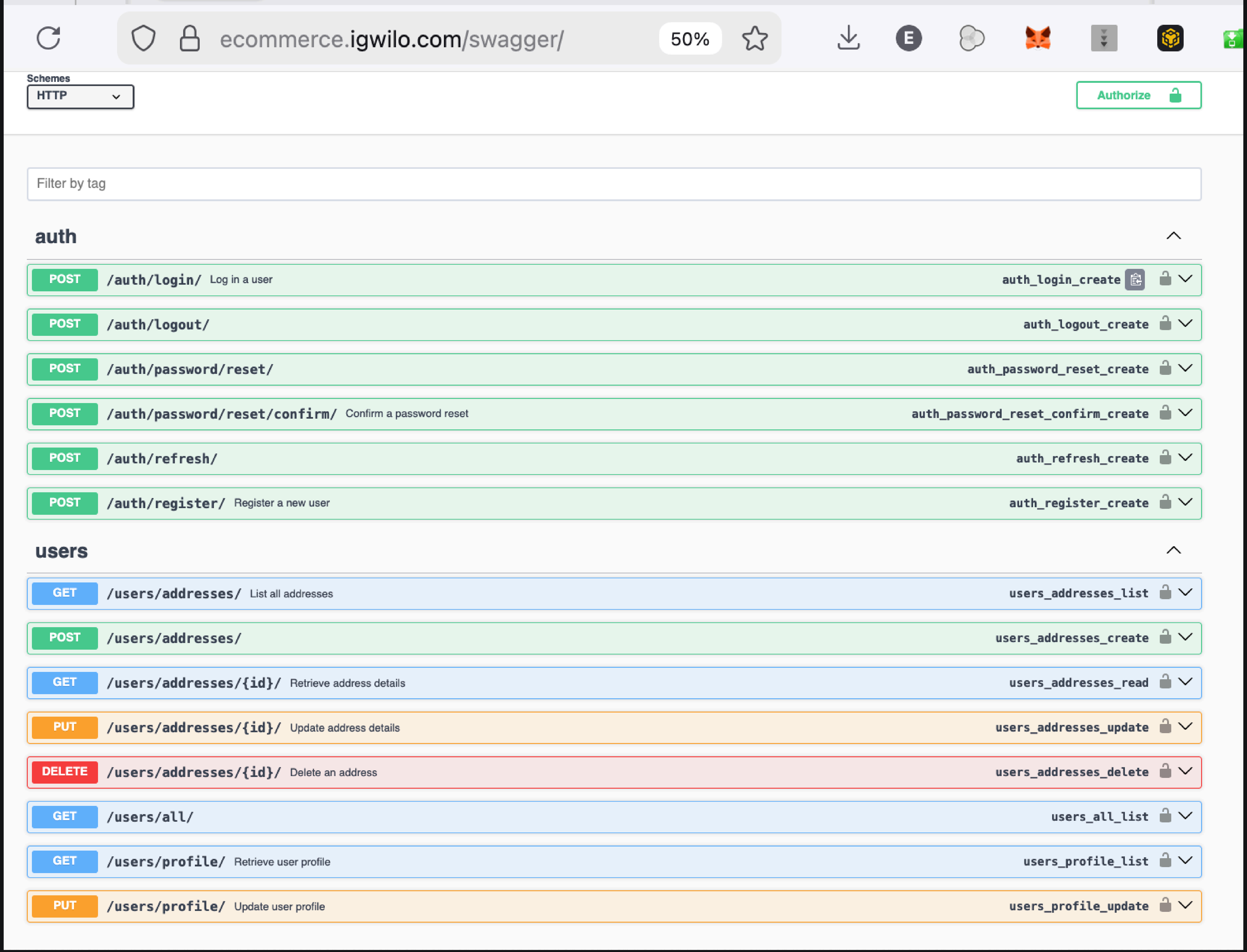
- # Analytics endpoints
- GET /api/analytics/dashboard/ # Overview
- GET /api/analytics/sales/ # Sales trends
- GET /api/analytics/products/ # Product performance
- GET /api/analytics/customers/ # Customer insights

Key Metrics Tracked

- Revenue - Daily, weekly, monthly
- Conversion rate - Cart to purchase
- Customer acquisition - New vs returning
- Inventory turnover - Stock movement
- Product ratings - Customer satisfaction

Automated Reporting

- Daily sales reports via email
- Low stock alerts
- Abandoned cart notifications
- Performance degradation alerts
-



ecommerce.igwilo.com/swagger/

50%

Schemes

HTTP

Authorize

Filter by tag

auth

POST /auth/login/ Log in a user auth_login_create

POST /auth/logout/ auth_logout_create

POST /auth/password/reset/ auth_password_reset_create

POST /auth/password/reset/confirm/ Confirm a password reset auth_password_reset_confirm_create

POST /auth/refresh/ auth_refresh_create

POST /auth/register/ Register a new user auth_register_create

users

GET /users/addresses/ List all addresses users_addresses_list

POST /users/addresses/ users_addresses_create

GET /users/addresses/{id}/ Retrieve address details users_addresses_read

PUT /users/addresses/{id}/ Update address details users_addresses_update

DELETE /users/addresses/{id}/ Delete an address users_addresses_delete

GET /users/all/ users_all_list

GET /users/profile/ Retrieve user profile users_profile_list

PUT /users/profile/ Update user profile users_profile_update