Store API Documentation

Last Updated: May 13, 2025

Author: Kalpesh Bhad

Base URL: http://localhost:3000/api/v1

# 1. Project Overview

The Store API is a RESTful service built with Express and MongoDB, providing secure and scalable endpoints for managing users, products, and categories. It includes features like user authentication, CRUD operations, validation, Swagger API documentation, and tests.

# 2. Setup Instructions

1. Clone the repository.  
2. Install dependencies with `npm install`.  
3. Create a `.env` file and configure MongoDB URI, JWT secret, and other settings.  
4. Run `npm start` to start the server.  
5. Access Swagger docs at `/api-docs`.

# 3. Folder Structure

- /api

- /constants

- /controllers

- /middlewares

- /routes

- /validators

- /configs

- /helpers

- /loaders

- /migration

- /models

- /services

- /tests

- app.js

- Dockerfile

- docker-compose.yml

- sample.env

- README.md

# 4. API Authentication

Authentication is handled via JWT. Routes requiring authentication use a middleware to verify tokens.

# 5. API Endpoints Summary

Refer to Swagger UI at `/api-docs` for complete documentation.

# 6. Validations

All incoming payloads are validated using Joi. Custom messages are used from `errorMessages.js`.

# 7. Testing

Unit tests validate individual functions, while integration tests check full request-response cycles.

Use `jest` and `supertest` for testing.

# 8. Trade-offs & Considerations

- Did not use `mongodb-memory-server` to avoid issues with CI/CD compatibility.  
- Assumed all product-category relationships are one-to-many.  
- Used `mongoose-populate` to simplify related data fetching.  
- Opted for route definition in an array format to centralize logic.  
- Swagger auto-generation integrated directly in routes.

# 9. Future Enhancements

- Add pagination and filtering to products and categories.  
- Implement rate limiting.  
- Add user role management.  
- Add email notifications.

# 10. Credits & License

Developed by Kalpesh Bhada.  
MIT License.