

Documentation Technique - API de Gestion de Marketplace

1 Introduction

L'API de gestion de marketplace permet aux utilisateurs de **s'inscrire, vendre des produits, passer des commandes et gérer les transactions**. L'API suit une architecture REST et est sécurisée avec **JWT**.

2 Technologies utilisées

- **Backend** : Node.js, Express.js
- **Base de données** : MongoDB Atlas
- **ORM** : Mongoose
- **Authentification** : JSON Web Tokens (JWT)
- **Sécurité** : Helmet, Express-Rate-Limit, Express-Mongo-Sanitize, XSS-Clean, HPP
- **Documentation** : Swagger UI
- **Déploiement** : Railway

3 Installation & Configuration

Prérequis

- Node.js installé
- Un compte MongoDB Atlas
- Un compte Railway
- Un éditeur de code (VS Code recommandé)

Installation

Cloner le projet
git clone <https://github.com/Bernadoadk/marketplace-api>

```
cd marketplace-api
```

```
# Installer les dépendances  
npm install
```

Configuration de l'environnement

Créer un fichier `.env` à la racine et ajouter :

```
PORT=5000
```

```
MONGO_URI=mongodb+srv://<user>:<password>@cluster0.mongodb.net/marketplace
```

```
JWT_SECRET=supersecretkey
```

Structure du Projet

```
marketplace-api/  
├── config/  
│   └── swagger.js  
├── controllers/  
│   ├── authController.js  
│   ├── productController.js  
│   └── orderController.js  
├── middleware/  
│   └── authMiddleware.js  
├── models/  
│   ├── User.js  
│   ├── Product.js  
│   └── Order.js  
├── routes/  
│   ├── authRoutes.js  
│   ├── productRoutes.js  
│   └── orderRoutes.js  
├── server.js  
├── package.json  
└── .env
```

5 Endpoints de l'API

Authentification

Méthode	Endpoint	Description
POST	/api/auth/register	Inscription d'un utilisateur
POST	/api/auth/login	Connexion d'un utilisateur
GET	/api/auth/verify	Vérification d'un token JWT

Gestion des Produits

Méthode	Endpoint	Description
POST	/api/products	Ajouter un produit (vendeur)
GET	/api/products	Récupérer tous les produits
GET	/api/products/:id	Récupérer un produit par ID
PUT	/api/products/:id	Modifier un produit (vendeur)
DELETE	/api/products/:id	Supprimer un produit (vendeur)

Gestion des Commandes

Méthode	Endpoint	Description
POST	/api/orders	Passer une commande (acheteur)
GET	/api/orders	Récupérer les commandes de l'utilisateur
PUT	/api/orders/:id	Modifier le statut d'une commande (vendeur/admin)
DELETE	/api/orders/:id	Supprimer une commande (admin)

6 Sécurité & Validation

- **Validation des entrées** : Express Validator
- **Protection contre injections NoSQL** : Express-Mongo-Sanitize
- **Protection contre XSS** : XSS-Clean
- **Limitation des requêtes** : Express-Rate-Limit

- **Authentification sécurisée** : JWT avec middleware

7 Documentation API avec Swagger

- Documentation interactive disponible sur : /api-docs
- Authentification Bearer Token activée pour les routes protégées.

8 Déploiement

Avec Railway

1 Créer un projet sur Railway et connecter le repo GitHub **2** Ajouter les variables d'environnement (MONGO_URI, JWT_SECRET) **3** Lancer le déploiement et récupérer l'URL de l'API

9 Tests & Améliorations Futures

✓ Tests unitaires avec Jest & Supertest (à implémenter) ✓ Ajout d'une intégration de paiement future (Stripe, Flutterwave...) ✓ Amélioration des rôles et permissions (ex: Super Admin)

 Développé par : Bernado ADIKPETO 