

Rapport pédagogique – TP Node.js + Express + MongoDB

1. Introduction

Ce projet consiste à créer une API REST complète en Node.js en respectant une architecture MVC. L'application gère des produits via un CRUD complet connecté à MongoDB.

2. Architecture du projet

L'arborescence respecte le modèle MVC :

- server.js : point d'entrée
- /models/product.js : définition du schéma Mongoose
- /controllers/product.controller.js : logique métier
- /routes/product.routes.js : définition des routes
- .env : configuration

3. server.js

- Importation des modules Express, CORS, dotenv et mongoose.
- Configuration du middleware express.json() pour lire le JSON dans les requêtes.
- Activation de CORS pour les requêtes multi-origines.
- Connexion à MongoDB via mongoose.connect().
- Montage des routes /products.
- Démarrage du serveur sur le port défini dans .env.

4. Modèle Product (product.js)

Le modèle définit les champs suivants :

- name (String, requis)
- description (String)
- price (Number, requis)
- quantity (Number, requis, valeur par défaut : 0)
- createdAt (Date, valeur par défaut : maintenant)

5. Contrôleur Product (product.controller.js)

Il contient 5 fonctions :

- createProduct : crée un produit (POST)
- getAllProducts : récupère tous les produits (GET)
- getProductById : récupère un produit par ID (GET :id)
- updateProduct : met à jour un produit (PUT :id)
- deleteProduct : supprime un produit (DELETE :id)

Chaque fonction :

- utilise async/await
- gère les erreurs via try/catch
- renvoie uniquement du JSON
- respecte les bons codes HTTP (201, 200, 404, 400)

6. Routes (product.routes.js)

Les routes REST pointent vers les fonctions du contrôleur :

- POST /products
- GET /products
- GET /products/:id
- PUT /products/:id
- DELETE /products/:id

7. Tests et fonctionnement

Les tests avec curl ont validé :

- création d'un produit
- récupération de tous les produits
- récupération par ID
- mise à jour
- suppression

Le CRUD complet est fonctionnel et connecté à MongoDB.

8. Conclusion

Ce projet respecte totalement les consignes du TP :

- architecture propre
- séparation MVC
- usage correct de Mongoose
- endpoints REST standards
- réponses JSON conformes
- bonne gestion des erreurs et codes HTTP