



Projet MongoDB / NodeJS

Sujet & Objectif

Digischool vous propose de mettre en pratique vos compétences récemment acquises à travers un projet complet. Vous allez devoir réaliser une **API REST complète** en utilisant **Express.js** et **MongoDB**, incluant tests unitaires et options avancées selon votre progression.

Pour cela, vous avez à votre disposition le fichier `digischool.sql` qui représente une base de données classique. Vous pouvez vous en inspirer pour modéliser vos données dans MongoDB.

Livrables attendus

Un dépôt GitHub contenant :

- Le code source complet et fonctionnel
- Les tests unitaires
- Un fichier `README.md` décrivant :
 - Le projet
 - Les dépendances
 - Les instructions pour lancer l'API
 - (Optionnel) La documentation Swagger
 - (Optionnel) La mise en place de sécurité basique
 - (Optionnel) Les étapes Docker
 - (Optionnel) L'authentification JWT

Contenu du projet

1. Initialisation du projet

- Création d'un projet Node.js
- Mise en place d'Express
- Mise en place de la structure du projet (routes, controllers, models)

2. Base de données – MongoDB

- Installation et configuration de MongoDB local **ou** utilisation de **MongoDB Atlas**
- Connexion à la base au démarrage de l'application
- OPTIONNEL (car sans schéma par défaut) : Création de schémas (DTO) et Models (DAO)

3. Développement de l'API REST

Créer une API comprenant au minimum un CRUD complet :

- **Create**
- **Read** (liste et détail)
- **Update**
- **Delete**

Endpoints et ressources libres (ex. : utilisateurs, produits, cours, etc.)

Conseil : regardez la doc mongodb. (ex. : findOneAndUpdate)

4. Tests unitaires (Jest)

- Tests des fonctions/services
- Tests des routes

Partie optionnelle (bonus)

5. Documentation Swagger

- Mise en place de Swagger pour documenter l'API
- Génération de la documentation à partir des annotations de code

6. Mise en place de sécurité basique

- Utilisation de Helmet pour sécuriser les headers HTTP
- Configuration de CORS pour restreindre l'accès à l'API
- Limitation du nombre de requêtes (rate limiting)

7. Docker

- Création d'un Dockerfile
- Mise en place d'un docker-compose (API + MongoDB)
- Vérification du bon fonctionnement en conteneur

8. Authentification JWT

- Création d'un système d'inscription / connexion
- Génération de tokens JWT
- Protection de certaines routes