

Évaluation – Développement d'une API Node.js (3e année Dev App)

Contexte :

Vous travaillez sur la création d'un micro réseau social. Une base de code vous est fournie, comprenant une API Express avec les fonctionnalités suivantes déjà développées :

- Inscription (POST /auth/signin)
- Connexion (POST /auth/login)
- Affichage des utilisateurs (GET /user)
- Middleware d'authentification avec JWT

Votre objectif est de compléter cette API en ajoutant la gestion des messages et des émotions.

Travail demandé :

1. Gestion des posts (CRUD) :

- GET /posts : liste de tous les messages
 - GET /posts/:id : détail d'un message
 - POST /posts : publier un message
 - PUT /posts/:id : modifier un message (seulement si vous en êtes l'auteur)
 - DELETE /messages/:id : supprimer un message (seulement si vous en êtes l'auteur)
- Chaque message doit contenir : authorId, content, createdAt, updatedAt. L'utilisateur peut aussi ajouter une image dans un post s'il le souhaite, mais ce n'est pas obligatoire.

2. Gestion des émotions (ajout/retrait de réaction) :

- POST /messages/:id/emotions : Ajoute ou remplace une émotion ('like', 'love', 'thumpup', etc.) pour un utilisateur donné.
 - DELETE /messages/:id/emotions : Supprime l'émotion de l'utilisateur sur ce message.
- Un utilisateur ne peut avoir qu'une seule émotion par message.

Contraintes techniques :

- Vous devez respecter l'architecture MVC (fichiers séparés : routes, contrôleurs, modèles).
- Vous devez utiliser Sequelize avec une base de données MySQL.
- Toutes les routes de messages et émotions doivent être protégées par le middleware d'authentification.
- Vous devez gérer les erreurs (requêtes invalides, accès non autorisé, etc.).

Vous pouvez vous servir des exercices/projets réalisés en cours, et internet. Mais pas d'IA génératives.