

TP API mi-cours

Travail Pratique : Création d'une API de Collection de Livres

Objectif :

Développer une API RESTful pour gérer une collection de livres, en utilisant TypeScript et la librairie Express.

Prérequis :

- Langage : TypeScript
- Librairie : Express
- Base de donnée Postgres
- Le projet ne doit contenir que le code nécessaire aux fonctionnalités demandées.

Partie 1 : Mise en Place

Configuration Initiale :

Pour démarrer votre application, vous devrez :

1. Commencer d'abord par initialisé votre base de donnée et la connexion à celle-ci.
2. l'application devra être démarré par un npm run start ou start:dev
3. Pour le développement rien ne vous empêche d'utiliser nodemon pour gagner du temps.

Partie 2 : Fonctionnalités CRUD

Gestion des Livres :

- **Ajout** : Il doit être possible d'ajouter une collection de livres avec les attributs suivants : id, nom, nombre de tomes, auteur, prix unitaire, et une note de 1 à 5. Les livres peuvent être fictifs mais il en faut un minimum de 10 dans votre collection. (bonus)Vous pouvez ajouter la date de parution du livre(bonus)
- **Lecture** : Récupérer les informations d'une collection via son identifiant de base de données.
- **Mise à jour** : Permettre la mise à jour du nombre de tomes ou de la note d'une collection. La modification d'autres champs est considérée comme un bonus.
- **Suppression** : Il doit être possible de supprimer une collection.

Partie 3 : Listing et Recherches

Tri :

Permettre le tri des livres par :

- Nombre de tomes
- Note
- Prix

Filtre :

Ajouter des filtres pour :

- Prix (inférieur/supérieur à un certain montant)
- Nombre de tomes (inférieur/supérieur à un certain nombre)

Partie 4 : Fonctionnalités Avancées (bonus)

Filtres Avancés :

- **Par Nom** : Filtrer les livres dont le nom contient une chaîne spécifique, sans nécessité d'une correspondance exacte.

Pagination

- implémenter la pagination dans votre listing.

Points d'Attention :

- Architecture des routes et cohérence avec les principes REST.
- Choix judicieux des verbes HTTP pour chaque action.
- Conformité du rendu avec les spécifications fournies.
- Qualité du code, lisibilité et maintenabilité.
- format de la date

Rendu :

- **Code Source** : Fournir le code source complet de l'application (les librairies du node_modules ne sont pas requisent)
- **Export de la Base de Données** : Inclure à la fois le schéma et les données de la base de données.
- **Instructions de Démarrage** : Pour lancer votre application, il doit être nécessaire de :
 1. Lancer une base de données avec les bons identifiants et votre export de base de données.
 2. Exécuter `npm install` à la racine du projet.
 3. Démarrer l'application avec `npm run start` .

Assurez-vous de suivre ces instructions pour développer une API fonctionnelle et bien structurée. Bonne chance !