Cahier des Charges : Développement d'une API REST pour une Application de Gestion de Bibliothèque

Table des matières

Objectif

Développer une API RESTful en utilisant Node.js et Express pour gérer les opérations d'une bibliothèque en ligne. L'API doit permettre la gestion des livres, des auteurs, des emprunteurs et des emprunts.

Contexte

L'application de gestion de bibliothèque doit permettre à l'administrateur de la bibliothèque de gérer les collections de livres et d'auteurs, ainsi que les emprunts effectués par les utilisateurs (emprunteurs).

Fonctionnalités de l'API

L'API doit offrir les fonctionnalités suivantes :

Gestion des livres

- Ajouter un nouveau livre
- Mettre à jour les informations d'un livre
- Supprimer un livre
- Obtenir la liste de tous les livres
- Obtenir les détails d'un livre par son ID

Gestion des auteurs

- Ajouter un nouvel auteur
- Mettre à jour les informations d'un auteur
- Supprimer un auteur
- Obtenir la liste de tous les auteurs
- Obtenir les détails d'un auteur par son ID

Gestion des emprunteurs

- Ajouter un nouvel emprunteur
- Mettre à jour les informations d'un emprunteur
- Supprimer un emprunteur
- Obtenir la liste de tous les emprunteurs
- Obtenir les détails d'un emprunteur par son ID

Gestion des emprunts

- Enregistrer un nouvel emprunt
- Mettre à jour un emprunt (ex. date de retour)
- Supprimer un emprunt
- Obtenir la liste de tous les emprunts
- Obtenir les détails d'un emprunt par son ID

Exigences Techniques

Technologies

- Node.js pour le serveur backend
- Express pour la gestion des routes et des middlewares
- MongoDB (ou une autre base de données NoSQL) pour la persistance des données
- Mongoose pour la modélisation des données

Endpoints API

Livres

POST /books : Créer un nouveau livre

GET /books : Obtenir tous les livres

GET /books/:id : Obtenir un livre par ID

PUT /books/:id : Mettre à jour un livre par ID

DELETE /books/:id : Supprimer un livre par ID

Auteurs

POST /authors : Créer un nouvel auteur

GET /authors : Obtenir tous les auteurs

GET /authors/:id : Obtenir un auteur par ID

PUT /authors/:id : Mettre à jour un auteur par ID

DELETE /authors/:id : Supprimer un auteur par ID

Emprunteurs

POST /borrowers : Créer un nouvel emprunteur

GET /borrowers : Obtenir tous les emprunteurs

GET /borrowers/:id : Obtenir un emprunteur par ID

PUT /borrowers/:id : Mettre à jour un emprunteur par ID

DELETE /borrowers/:id : Supprimer un emprunteur par ID

Emprunts

POST /loans : Enregistrer un nouvel emprunt

GET /loans : Obtenir tous les emprunts

GET /loans/:id : Obtenir un emprunt par ID

PUT /loans/:id : Mettre à jour un emprunt par ID

DELETE /loans/:id : Supprimer un emprunt par ID

Contraintes

Validation des Données

Utiliser des middlewares de validation pour vérifier les données envoyées aux endpoints.

Gestion des Erreurs

Implémenter une gestion centralisée des erreurs pour capturer et répondre avec des messages d'erreur appropriés.

Sécurité

Utiliser des mécanismes d'authentification et d'autorisation pour protéger certaines routes (par exemple, la création et la suppression de livres, auteurs, emprunteurs).

Étapes de Développement

Initialisation du Projet

Initialiser le projet Node.js.

Installer les dépendances nécessaires (Express, Mongoose, etc.).

Configuration de la Base de Données

Configurer la connexion à la base de données MongoDB.

Création des Modèles Mongoose

Définir les schémas et modèles Mongoose pour les livres, auteurs, emprunteurs et emprunts.

Implémentation des Routes et Contrôleurs

Implémenter les routes pour chaque entité (livres, auteurs, emprunteurs, emprunts).

Créer les contrôleurs pour gérer la logique métier de chaque route.

Validation et Gestion des Erreurs

Ajouter des middlewares de validation pour les données d'entrée.

Implémenter une gestion centralisée des erreurs.

Sécurité

Mettre en place l'authentification et l'autorisation.

Livrables

- Code source de l'API.
- Documentation de l'API.
- Script de configuration de la base de données.
- Fichiers de configuration pour le déploiement.
- Rapport de tests.

Conclusion

Cet exercice vous permettra de comprendre les concepts avancés de la création d'une API RESTful, y compris la gestion des relations entre les données, la validation des données, la sécurité, et la documentation de l'API.