

Exercice : Gestion de Films pour un Cinéma

Vous avez été engagé pour développer une application de gestion de films pour un cinéma local. La base de données doit stocker des informations sur les films tels que le titre, la description, le réalisateur et la date de sortie.

1. Mise en place du Projet :

- Créez un nouveau projet Node.js.
- Installez les dépendances nécessaires : `express` , `mongoose` .

2. Définir le Modèle de Film :

- Créez un fichier `movie.js` pour définir le schéma et le modèle Mongoose pour les films.
- Incluez au moins les propriétés suivantes : `title` , `description` , `director` , `releaseDate` .

3. Connexion à la Base de Données :

- Dans votre fichier principal (`app.js`), établissez une connexion à une base de données MongoDB locale en utilisant Mongoose.
- Assurez-vous que la connexion est réussie avant de passer à l'étape suivante.

4. Création d'un Nouveau Film :

- Créez une route pour ajouter un nouveau film.
- Utilisez la méthode `create` de Mongoose pour insérer un film dans la base de données.

5. Affichage de Tous les Films :

- Créez une route pour afficher tous les films disponibles.
- Utilisez la méthode `find` de Mongoose pour récupérer la liste des films depuis la base de données.
- Affichez ces films dans la réponse HTTP.

6. Recherche d'un Film par Titre :

- Ajoutez une route pour rechercher un film par son titre.
- Utilisez la méthode `findOne` de Mongoose pour récupérer un film en fonction du titre spécifié dans les paramètres de la requête.

7. Mise à Jour d'un Film :

- Ajoutez une route pour mettre à jour les informations d'un film existant.
- Utilisez la méthode `findOneAndUpdate` de Mongoose pour trouver et mettre à jour le film en fonction de son ID.

8. Suppression d'un Film :

- Créez une route pour supprimer un film en fonction de son ID.
- Utilisez la méthode `findByIdAndDelete` de Mongoose.

9. Tests :

- Utilisez un outil comme Postman ou CURL pour tester chacune des fonctionnalités CRUD de votre application.
- Vérifiez que les opérations sont correctement exécutées et que la base de données est mise à jour conformément aux actions effectuées.

10. Bonus :

- Ajoutez des validations au niveau du schéma Mongoose pour garantir que certaines propriétés sont obligatoires ou respectent certaines conditions.
- Ajoutez des gestionnaires d'erreurs pour traiter les cas où une opération échoue.

Ressources :

- [Documentation Mongoose](#)

Conseils :

- N'hésitez pas à consulter la documentation de Mongoose pour plus de détails sur les méthodes et les options disponibles.
- Assurez-vous d'organiser votre code de manière modulaire avec des fichiers distincts pour le modèle, les routes, etc.

- Testez fréquemment chaque étape pour vous assurer que tout fonctionne correctement.

Bon codage !