Exigence de l’API

Ce document détaille les besoins de l’API requis pour le bon fonctionnement du front-end.

Spécification de l’API

	Point d'accès	**Request body (le cas échéant)**	**Type de réponse attendu**	**Fonction**
POST	/api/auth/signup	{ email: string, password: string }	{ message: string }	Hachage du mot de passe de l'utilisateur, ajout de l'utilisateur à la base de données.
POST	/api/auth/login	{ email: string, password: string }	{ userId: string, token: string }	Vérification des informations d'identification de l'utilisateur ; renvoie l’_id de l'utilisateur depuis la base de données et un token web JSON signé (contenant également l'_id de l'utilisateur).
GET	/api/books	-	Array of books	Renvoie un tableau de tous les livres de la base de données.
GET	/api/books/:id	-	Single book	Renvoie le livre avec l’_id fourni.
GET	/api/books/bestrating	-	Array of books	Renvoie un tableau des 3 livres de la base de données ayant la meilleure note moyenne.
POST	/api/books	{ book: string, image: file }	{ message: String } **Verb**	Capture et enregistre l'image, analyse le livre transformé en chaîne de caractères, et l'enregistre dans la base de données en définissant

				correctement son ImageUrl. Initialise la note moyenne du livre à 0 et le rating avec un tableau vide. Remarquez que le corps de la demande initiale est vide ; lorsque Multer est ajouté, il renvoie une chaîne pour le corps de la demande en fonction des données soumises avec le fichier.
PUT	/api/books/:id	EITHER Book as JSON OR { book: string, image: file }	{ message: string }	Met à jour le livre avec l'_id fourni. Si une image est téléchargée, elle est capturée, et l'ImageUrl du livre est mise à jour. Si aucun fichier n'est fourni, les informations sur le livre se trouvent directement dans le corps de la requête (req.body.title, req.body.author, etc.). Si un fichier est fourni, le livre transformé en chaîne de caractères se trouve dans req.body.book. Notez que le corps de la demande initiale est vide ; lorsque Multer est ajouté, il renvoie une chaîne du corps de la demande basée sur les données soumises avec le fichier.
DELETE	/api/books/:id	-	{ message: string }	Supprime le livre avec l'_id fourni ainsi que l'image associée.
POST	/api/books/:id/rating	{ userId: String, rating: Number }	{ message: string }	Définit la note pour le user ID fourni. La note doit être comprise entre 0 et 5. L'ID de l'utilisateur et la note doivent être ajoutés ou retirés du tableau "rating". Cela permet de garder une trace de leurs notations, et les empêche de noter le même livre plusieurs fois. La note moyenne "averageRating" doit être tenue à jour.

API Errors
Les erreurs éventuelles doivent être renvoyées telles qu'elles sont produites, sans modification ni ajout. Si nécessaire, utilisez une nouvelle Error().

API Routes
Toutes les routes pour les livres doivent disposer d'une autorisation (le token est envoyé par le front-end avec l'en-tête d'autorisation « Bearer »). Avant qu'un utilisateur puisse apporter des modifications à la route livre (book), le code doit vérifier si le user ID actuel correspond au user ID du livre. Si le user ID ne correspond pas, renvoyer « 403: unauthorized request ». Cela permet de s'assurer que seul le propriétaire d'un livre puisse apporter des modifications à celui-ci.

Models

User {
email : String - adresse e-mail de l'utilisateur [unique]
password : String - mot de passe haché de l'utilisateur
}

Book {
userId : String - identifiant MongoDB unique de l'utilisateur qui a créé le livre
title : String - titre du livre
author : String - auteur du livre

```
    imageUrl : String     - illustration/couverture du livre
    year: Number    - année de publication du livre
    genre: String    - genre du livre
    ratings : [
      {
        userId : String     - identifiant MongoDB unique de l'utilisateur qui a noté le livre
        grade : Number     - note donnée à un livre
      }
    ]         - notes données à un livre
    averageRating : Number     - note moyenne du livre
}
```

Sécurité

- Le mot de passe de l'utilisateur doit être haché.
- L'authentification doit être renforcée sur toutes les routes livre (book) requises.
- Les adresses électroniques dans la base de données sont uniques, et un plugin Mongoose approprié est utilisé pour garantir leur unicité et signaler les erreurs.
- La sécurité de la base de données MongoDB (à partir d'un service tel que MongoDB Atlas) ne doit pas empêcher l'application de se lancer sur la machine d'un utilisateur.
- Les erreurs issues de la base de données doivent être remontées.