**Documentation API REST — Bibliothèque en ligne**

**Base URL**

bash

CopierModifier

http://localhost:3000/api

**1. Authentification**

**1.1. Enregistrement d’un utilisateur**

* **URL :** /users/register
* **Méthode :** POST
* **Description :** Crée un nouvel utilisateur.
* **Corps de la requête (JSON) :**

json

CopierModifier

{

"username": "exemple",

"password": "motdepasse"

}

* **Réponse (201) :**

json

CopierModifier

{

"message": "Utilisateur créé avec succès"

}

**1.2. Connexion (login)**

* **URL :** /users/login
* **Méthode :** POST
* **Description :** Connecte un utilisateur et retourne un token JWT.
* **Corps de la requête (JSON) :**

json

CopierModifier

{

"username": "exemple",

"password": "motdepasse"

}

* **Réponse (200) :**

json

CopierModifier

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."

}

* **Utilisation :** Copier ce token pour accéder aux routes protégées.

**2. Gestion des livres**

Toutes les routes ci-dessous nécessitent un **token JWT** dans l'en-tête Authorization sous la forme :

php-template

CopierModifier

Bearer <token>

**2.1. Ajouter un livre**

* **URL :** /livres
* **Méthode :** POST
* **Description :** Ajouter un nouveau livre dans la bibliothèque.
* **Headers :**
  + Authorization: Bearer <token>
* **Corps de la requête (JSON) :**

json

CopierModifier

{

"titre": "Titre du livre",

"auteur": "Nom de l'auteur",

"resume": "Résumé du livre"

}

* **Réponse (201) :**

json

CopierModifier

{

"message": "Livre ajouté avec succès",

"livre": {

"\_id": "64f998abc1234d56e7890123",

"titre": "Titre du livre",

"auteur": "Nom de l'auteur",

"resume": "Résumé du livre"

}

}

**2.2. Lister tous les livres avec pagination**

* **URL :** /livres
* **Méthode :** GET
* **Description :** Récupère la liste des livres, avec pagination.
* **Headers :**
  + Authorization: Bearer <token>
* **Paramètres Query :**
  + page (optionnel, défaut = 1) — numéro de la page
  + limit (optionnel, défaut = 10) — nombre de livres par page
* **Exemple :**

bash

CopierModifier

/livres?page=2&limit=5

* **Réponse (200) :**

json

CopierModifier

{

"page": 2,

"limit": 5,

"totalLivres": 23,

"totalPages": 5,

"livres": [

{

"\_id": "64f998abc1234d56e7890123",

"titre": "Livre 6",

"auteur": "Auteur 6",

"resume": "Résumé du livre 6"

},

// autres livres...

]

}

**3. Middleware d’authentification (pour info)**

* Le middleware authMiddleware vérifie la présence d’un token JWT valide dans l’en-tête Authorization.
* Si le token est valide, la requête est autorisée à continuer.
* Sinon, l’API répond avec une erreur 401 ou 403.

**4. Erreurs courantes**

| **Code** | **Description** | **Cause possible** |
| --- | --- | --- |
| 400 | Mauvaise requête | Données manquantes ou mal formatées |
| 401 | Non autorisé | Token manquant ou invalide |
| 404 | Ressource non trouvée | Livre ou utilisateur inexistant |
| 500 | Erreur serveur | Problème interne côté serveur |

**5. Exemple d’en-têtes avec Postman**

* Dans **Headers**, ajoute :

| **Clé** | **Valeur** |
| --- | --- |
| Authorization | Bearer eyJhbGciOiJIUzI1NiIs... |