

TP5

Étape 1 – C'est la base

Étant donné que MongoDB était déjà installé sur mon PC, j'ai directement vérifié son bon fonctionnement. Après avoir démarré le service MongoDB, je me suis connecté via mongosh et exécuté la commande `show dbs` pour vérifier la présence des bases de données par défaut (admin, config, local).

Pour faciliter l'interaction avec la base de données, j'ai ajouté MongoDB dans PhpStorm via l'outil intégré de base de données, confirmant ainsi que le service fonctionnait correctement.

Étape 2 – C'est la base

J'ai créé un projet Node.js en initialisant npm et en installant les dépendances nécessaires :

- `npm init -y`
- `npm install fastify fastify-plugin mongoose dotenv fastify-secure-session fastify-helmet fastify-cors`

Ensuite, j'ai généré un certificat auto-signé pour activer HTTPS :

- `openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes`

Puis, j'ai mis en place le serveur Fastify dans `server.js` pour activer HTTPS et la connexion à MongoDB.

Après avoir créé un fichier `.env` contenant l'URI de connexion à MongoDB, j'ai testé l'exécution du serveur via :

- `node server.js`

Le serveur a bien démarré sur `https://localhost:3000` et la connexion MongoDB a été confirmée dans la console.

Étape 3 – Les routes

J'ai créé un dossier `models` contenant le fichier `Book.js`, qui définit le schéma Mongoose, j'ai ensuite créé le fichier `routes.js` pour définir les routes de l'API et relié chaque route aux handlers situés dans `controllers/bookController.js`. Les routes principales sont :

- `POST /books` : Ajouter un livre
- `GET /books` : Récupérer tous les livres
- `GET /books/:id` : Récupérer un livre spécifique
- `PUT /books/:id` : Mettre à jour un livre
- `DELETE /books/:id` : Supprimer un livre