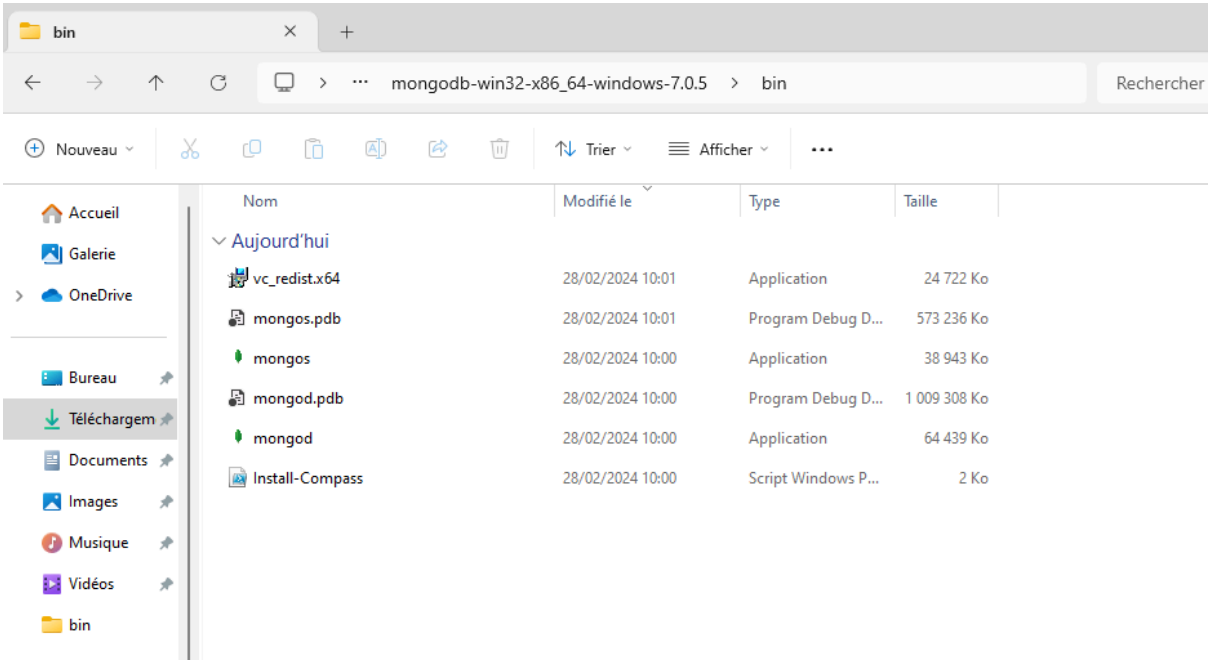
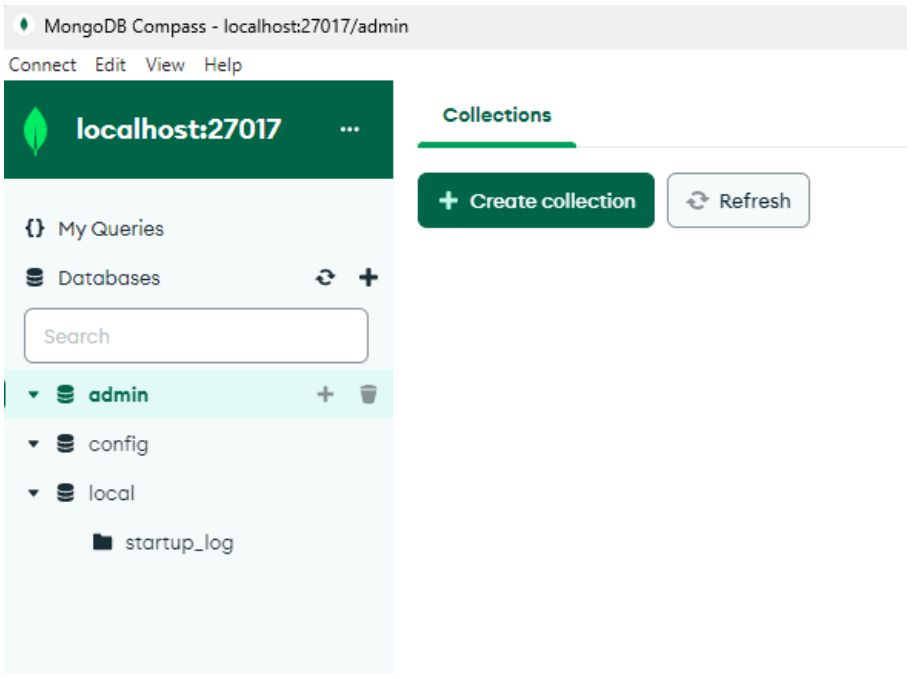
Mehdi
Zaoui

TP5

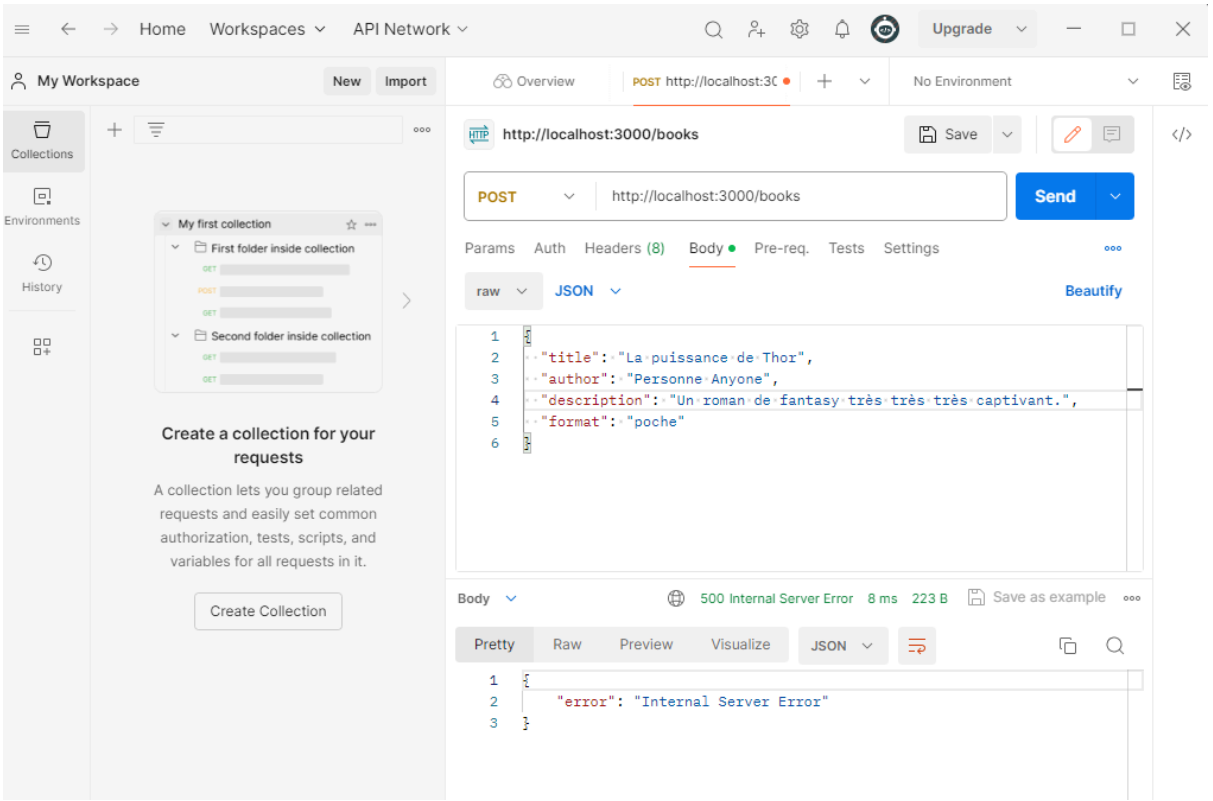
Au début, j'ai dû installer MongoDB sur ma machine en suivant la documentation officielle. C'était un peu délicat au début car je devais m'assurer de suivre toutes les étapes correctement. Mais après avoir suivi les instructions pas à pas, j'ai réussi à obtenir MongoDB fonctionnel sur ma machine. Ensuite, en utilisant l'outil mongosh, j'ai vérifié que les bases de données par défaut étaient bien présentes, ce qui était un soulagement car cela signifiait que MongoDB fonctionnait correctement.



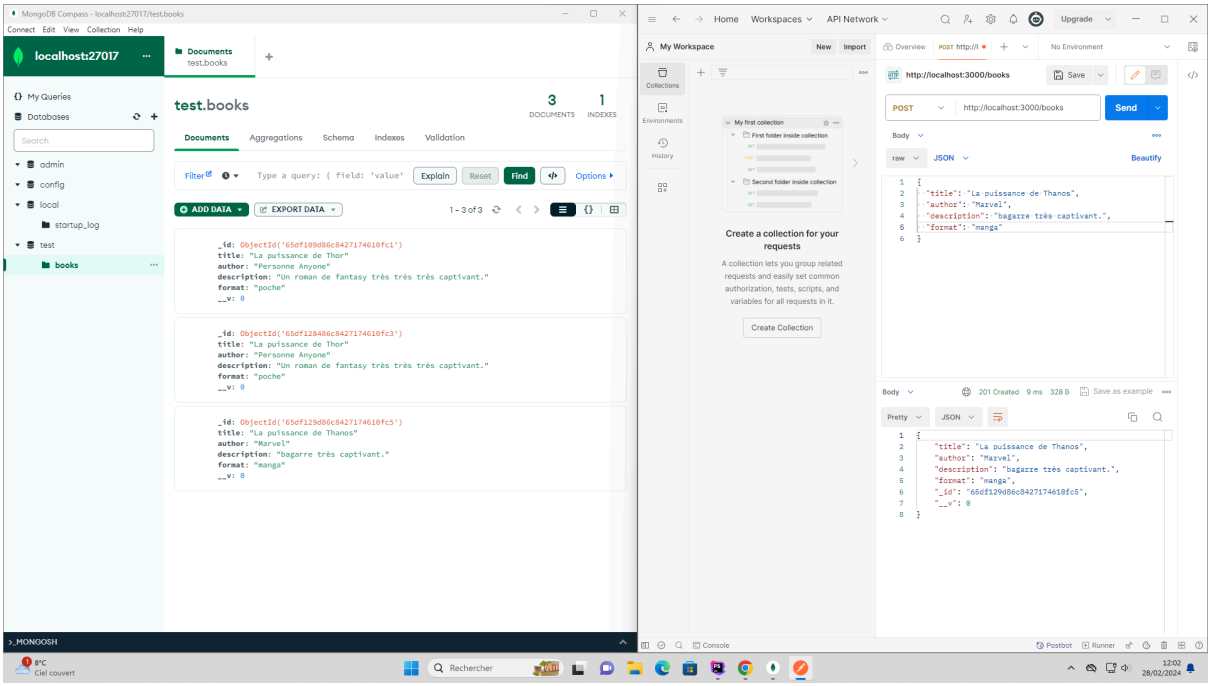


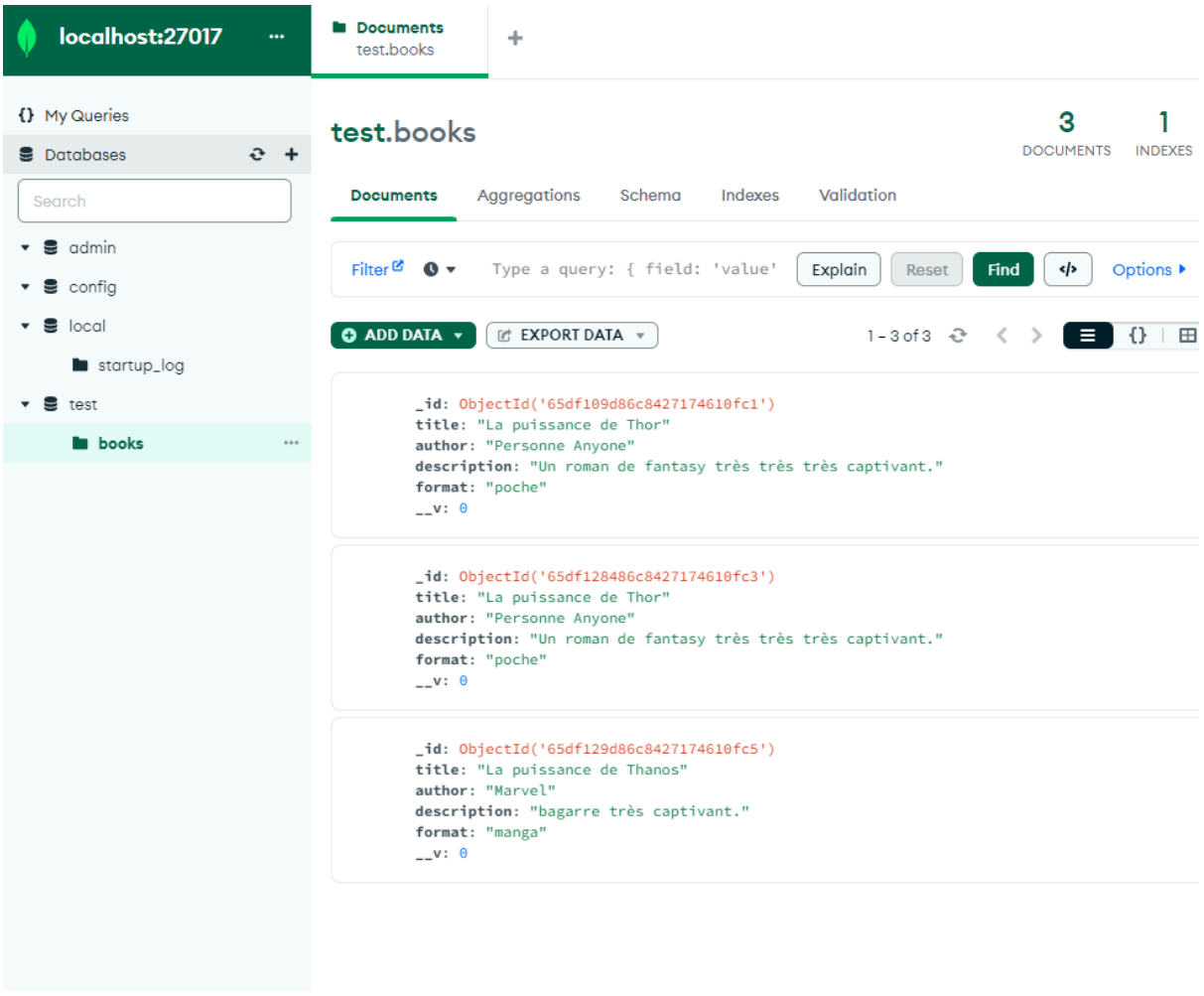
L'utilisation de MongoDB Compass pour explorer les bases de données était assez simple. J'ai ouvert l'application et j'ai pu facilement visualiser les bases de données par défaut ainsi que toute autre base de données que j'avais créée précédemment. C'était une étape relativement facile par rapport à l'installation initiale de MongoDB.

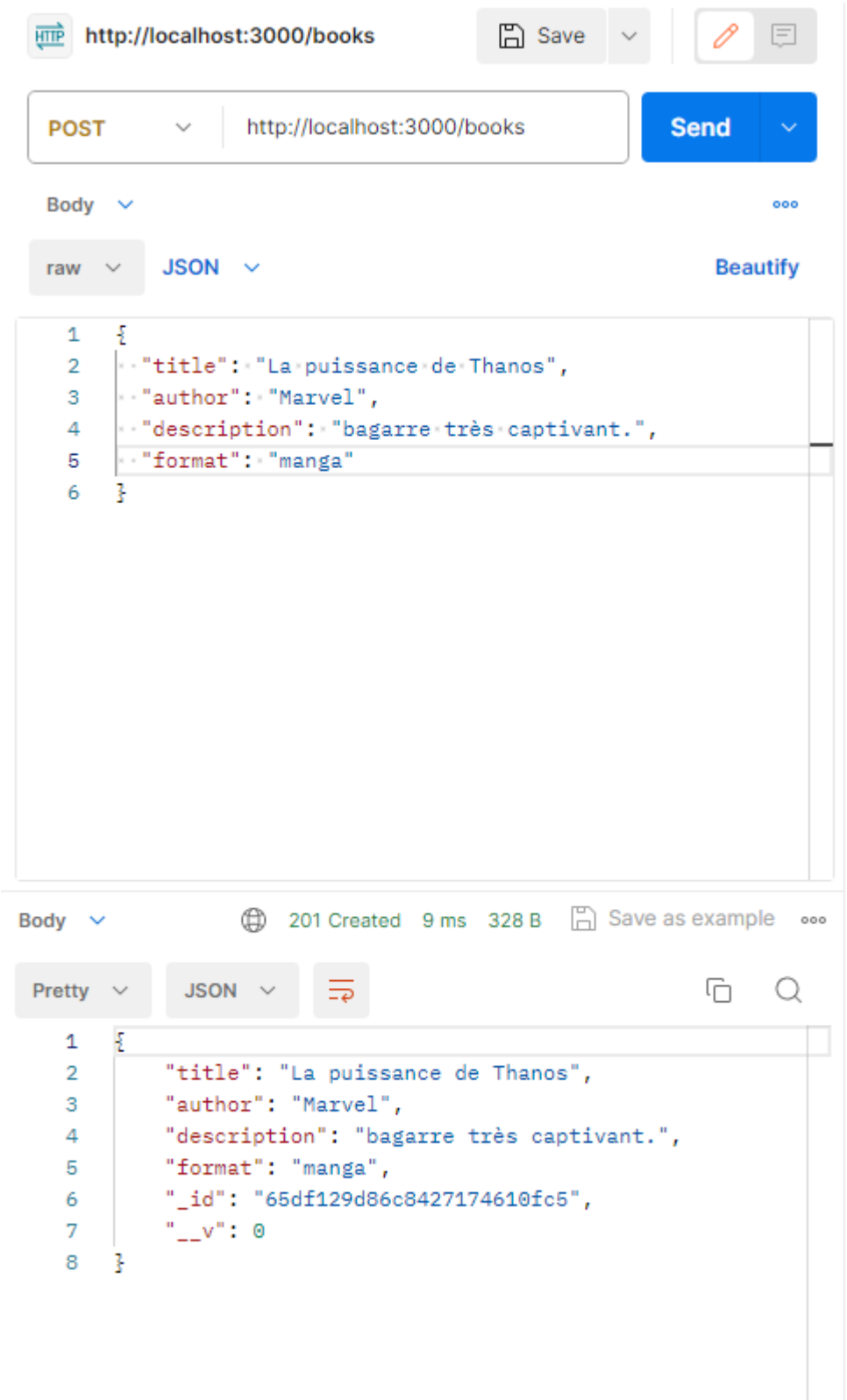
L'étape suivante consistait à connecter MongoDB à mon application Fastify et à définir un modèle de données pour les livres. C'était un peu compliqué car je devais m'habituer à la syntaxe de Mongoose et comprendre comment configurer la connexion à la base de données dans Fastify. Après avoir consulté la documentation et fait quelques essais, j'ai réussi à établir la connexion et à définir le modèle de données.



Tester l'API avec Postman était une étape amusante mais j'ai rencontré un problème lors de la connexion à mon serveur Fastify. Je n'avais pas sécurisé la connexion avec la commande openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem car le docker bug énormément sur les machines de l'iut. C'était un peu frustrant car je pensais que j'avais mal configuré quelque chose. Après avoir vérifié mes paramètres de connexion, j'ai finalement résolu le problème et j'ai réussi à ajouter un nouveau livre avec succès via Postman.




localhost:27017
Documents
test.books
My Queries
Databases
Search
admin
config
local
startup_log
test
books
test.books
3
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value'
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
1 - 3 of 3
_id: ObjectId('65df109d86c8427174610fc1')
title: "La puissance de Thor"
author: "Personne Anyone"
description: "Un roman de fantasy très très très captivant."
format: "poche"
__v: 0
_id: ObjectId('65df128486c8427174610fc3')
title: "La puissance de Thor"
author: "Personne Anyone"
description: "Un roman de fantasy très très très captivant."
format: "poche"
__v: 0
_id: ObjectId('65df129d86c8427174610fc5')
title: "La puissance de Thanos"
author: "Marvel"
description: "bagarre très captivant."
format: "manga"
__v: 0



La validation des données avec des schémas JSON était probablement l'étape la plus difficile. J'ai trouvé un peu difficile de définir les schémas JSON pour chaque route et de filtrer correctement les données de sortie. Après avoir consulté la documentation de Fastify et fait plusieurs tentatives, j'ai finalement réussi à implémenter la validation des données avec succès.

En résumé, bien que j'aie rencontré quelques obstacles tout au long de ces étapes, j'ai réussi à les surmonter grâce à la persévérance et à la recherche active de solutions. Chaque défi rencontré m'a permis d'apprendre quelque chose de nouveau sur Fastify et MongoDB.