Une image contenant texte, Police, Graphique, calligraphie

Description générée automatiquement

**BUT 3 INFORMATIQUE**

**Parcours A FI**

**Réalisations d'applications**

**RAPPORT TP5**

**Albums et Mongoose**

**Prénom et Nom de l’étudiant :** Akash Selvaratnam

**Groupe :** 303

**Promotion :** 2023-2024

# Sommaire

Table des matières

[Sommaire 2](#_Toc160376406)

[Étape 1 – C’est la base 3](#_Toc160376407)

[Étape 2 – C’est la base 3](#_Toc160376408)

[Étape 3 – Les routes 5](#_Toc160376409)

# Étape 1 – C’est la base

Pour la première étape, j’avais déjà installé MongoDB Community sur mon ordinateur, je me suis donc tout simplement connecté a ma base de données MongoDB avec la commande **mongosh** puis j’ai lancé la commande **show dbs** afin de voir l’ensemble des bases de données existantes.

Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 1 : Terminale mongosh avec l'ensemble des bases de données existante

# Étape 2 – C’est la base

Pour la deuxième étape, j’ai installé les modules **Fastify** et **mongoose** comme on peut le voir sur le fichier **package.json.**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 2 : Package.json (dépendances)

Ensuite, j’ai créé un nouveau fichier que j’ai nommé **databases.js** ou je me connecte à ma base de données **MongoDB** dans une fonction asynchrone que j’exporte.

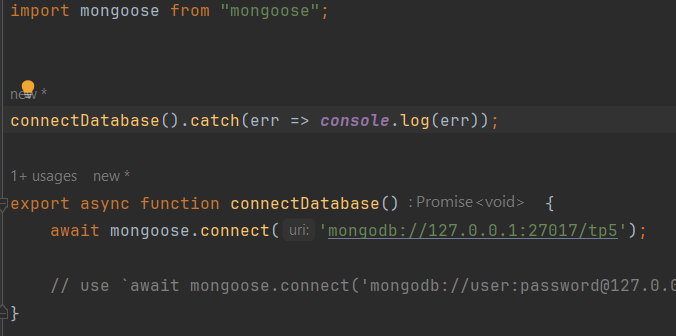


Figure 3 : databases.js

J’ai créé un nouveau fichier nommé **model.js** pour pouvoir créer un nouveau document que j’appellerai **Livre**. Pour cela, j’ai créé un nouveau schéma avec la méthode **Shema** proposé par **Mongoose**. Ce schéma contient 4 différents paramètres (**le titre et l’auteur** qui est de type **String** et qui est requis, **la description** qui est également de type **String** et le **format** qui est de type **String** et avec **comme valeur de défaut poche**, les trois valeurs acceptées pour le format sont **poche, manga et audio**).

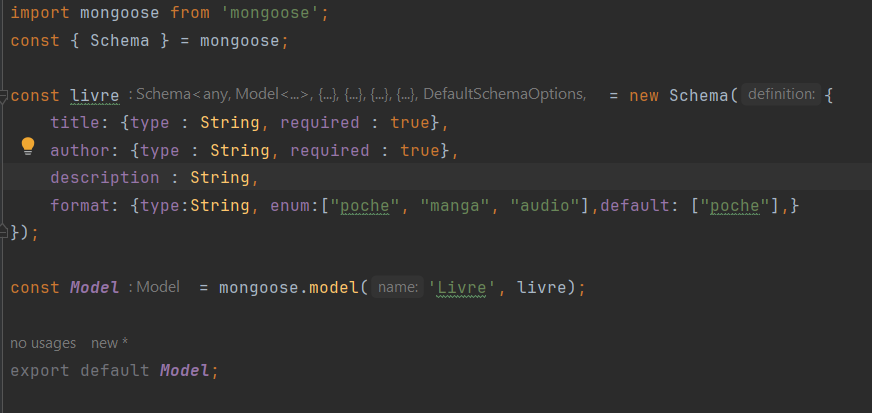


Figure 4 : Schéma du Livre

Ensuite, j’ai créé le modèle associé au **schéma du Livre**, ce modèle, je l’ai **nommé Livre** puis **j’exporte mon Model**.

# Étape 3 – Les routes

Pour l’étape 3, j’ai défini l’ensemble des routes de l’application pour les 4 opérations suivantes, l’ajout d’un livre (**POST**), la suppression d’un livre (**DELETE**), la mise à jour d’un livre (**PUT**) et l’obtention des livres dans la base de données (**GET**).

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 5 : Route permettant de récupérer l'ensemble des livres

Ensuite, j’ai créé 4 différentes méthodes, **une première méthode permettant d’ajouter un livre**, pour cela, je me connecte à ma base de données puis je réalise les opérations nécessaires pour ajouter un nouveau livre dans mon document puis je me déconnecte de ma base de données et **je retourne** **le titre, le nom de l’auteur, la description et le format du livre que j’ai ajouté avec un code status 200.**

Pour cette première méthode, j’ai réalisé un **schéma JSON d’entrée** avec les **propriétés title (requis), author (requis), description et format.**

J’ai également réalisé un **schéma JSON de réponse/sortie** avec les propriété **title (requis), author (requis), description et format (requis).**



Figure 6 : Méthode permettant d'ajouter un livre

Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 7 : Route de la méthode permettant d'ajouter un livre

Pour la méthode permettant de **récupérer l’ensemble des livres**, je me suis tout d’abord connecté à ma base de données puis j’ai réalisé **la méthode find()** proposé par le **module mongoose** sur tous les éléments de ma base de données, je me déconnecte ensuite de ma base de données et je **retourne un tableau JSON** qui contient **le titre, le nom de l’auteur, la description et le format du livre que j’ai récupéré avec un code status 200.**

J’ai également réalisé un **schéma JSON de réponse/sortie** avec les propriété **title (requis), author (requis), description et format (requis).**

**Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement**

Figure 8 : Méthode permettant de récupère l'ensemble des livre de la base de données

****

Figure 9 : Route permettant de récupérer l'ensemble des livres

J’ai réalisé ensuite une méthode permettant de **mettre à jour les informations d’un livre**, je me connecte à ma base de données puis j’exécute la méthode **findOneAndUpdate()** avec **l’id du livre** qu’on souhaite mettre à jour et **les données qu’on souhaite modifier**, il s’agit d’une méthode proposée par le **module mongoose** et je me déconnecte ensuite de ma base de données. Pour finir, je retourne un objet JSON qui contient **le titre, le nom de l’auteur, la description et le format du livre avec un code status 200.**

Pour cette méthode, j’ai réalisé un **schéma JSON d’entrée** avec la propriété **id (requis).**

J’ai également réalisé un **schéma JSON de réponse/sortie** avec les propriété **title (requis), author (requis), description et format (requis).**

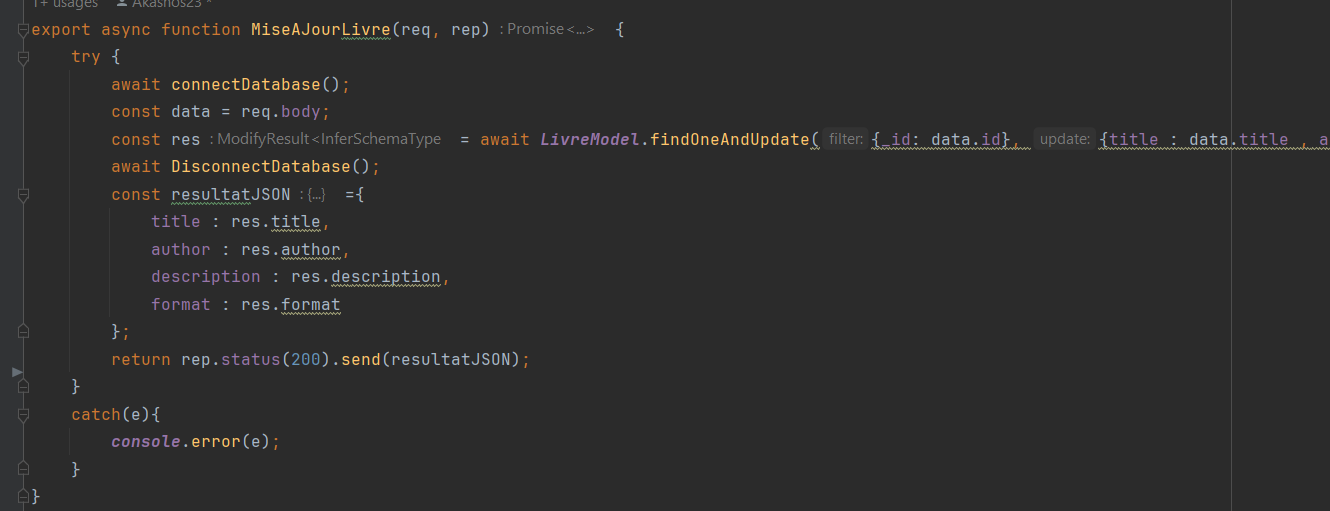
****

Figure 10 : Méthode permettant de mettre à jour un livre

Une image contenant texte, capture d’écran

Description générée automatiquement

Figure 11 : Route permettant de mettre à jour un livre

Pour finir, j’ai réalisé une méthode permettant de **supprimer un livre**, tout d’abord, je me connecte à ma base de données puis j’exécute la **fonction findOneAndDelete()** avec comme **paramètre l’id du livre** que je souhaite supprimer puis je me déconnecte de la base de données et **je retourne un objet JSON** qui contient **le titre, le nom de l’auteur, la description et le format du livre avec un code status 200.**

Pour cette méthode, j’ai réalisé un **schéma JSON d’entrée** avec la propriété **id (requis).**

J’ai également réalisé un **schéma JSON de réponse/sortie** avec les propriété **title (requis), author (requis), description et format (requis).**

**Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement**

Figure 12 : Méthode permettant de supprimer un livre



Figure 13 : Route permettant de supprimer un livre