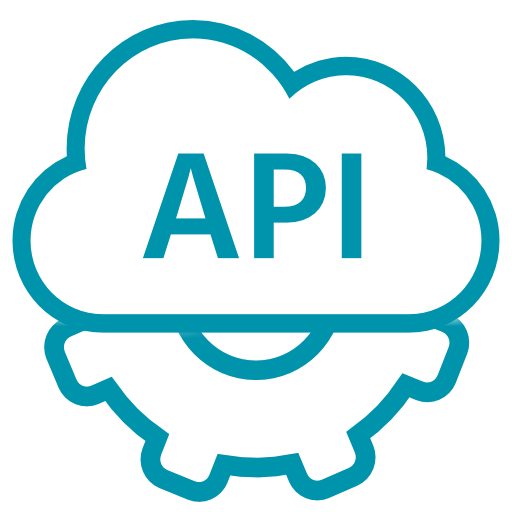
Logotipo, Icono

Descripción generada automáticamente

**ECOLE TECHNIQUE  
ECOLE DES MÉTIERS LAUSANNE**

API REST

P\_WEB 295



Joachim Berchel & Thomas Nardou

MID2 - 2024

02.02.2024 – 15.03.2024

Antoine Mveng

Vennes - ETML

Table des matières

[1. Introduction 3](#_Toc160267989)

[2. Analyse 3](#_Toc160267990)

[2.1 Planification 3](#_Toc160267991)

[2.2 API REST 3](#_Toc160267992)

[createBooks 3](#_Toc160267993)

[deleteBooks 3](#_Toc160267994)

[findAllBooks 3](#_Toc160267995)

[findBooksBycategory 3](#_Toc160267996)

[findBooksById 3](#_Toc160267997)

[findBooksByTitle 3](#_Toc160267998)

[updateBooks 3](#_Toc160267999)

[login 3](#_Toc160268000)

[2.3 Base de données 4](#_Toc160268001)

[2.4 Analyse de la structure du code 4](#_Toc160268002)

[2.5 Schéma d’interaction backend - frontend 4](#_Toc160268003)

[3. Réalisation 4](#_Toc160268004)

[3.1 Authentification et gestion des rôles 4](#_Toc160268005)

[3.2 Sécurité 4](#_Toc160268006)

[3.3 Technique 5](#_Toc160268007)

[4. Test 6](#_Toc160268008)

[5. Conclusion 6](#_Toc160268009)

[5.1 Gestion du code 6](#_Toc160268010)

[5.2 Conclusion générale 6](#_Toc160268011)

[5.3 Conclusion personnelle 6](#_Toc160268012)

[5.4 Critiques 6](#_Toc160268013)

[6. Webographie / Bibliographie / Glossaire 6](#_Toc160268014)

[7. Utilisation d’IA 6](#_Toc160268015)

# Introduction

Comprend une brève explication du projet (½ page minimum)

L’objectif de ce projet est de réaliser le backend d’une application avec Node, ExpressJs et une base de données MySQL. Le travail s’effectue en groupe de 2. Chacun doit avoir travaillé sur chaque aspect du projet (programmation, rapport). Le projet s’effectue du 2 février 2024 jusqu’au 15 mars 2024. Le projet est lié au module C295 « Réaliser le backend pour des applications ».

TODO

# Analyse

## Planification

Planification de toutes les tâches à réaliser : Vous devez utiliser un outil permettant au chef de projet de voir l’évolution des tâches (GitHub projet ou Trello).

La planification des tâches se fait sur Trello : « https://trello.com/invite/b/1XpbzDPe/ATTIe16af295e76a2196a850280222a1304897A962C5/papi »

## API REST

### createBooks

POST localhost :3000/api/books/add

### deleteBooks

DELETE localhost :3000/api/books/remove/1

### findAllBooks

GET localhost :3000/api/books

### findBooksBycategory

GET localhost :3000/api/books/category/manga

### findBooksById

GET localhost :3000/api/books/1

### findBooksByTitle

GET localhost :3000/api/books/title/monster

### updateBooks

PUT localhost :3000/api/books/1

### login

Contiendra une analyse de l’API REST présentant pour chaque route :

Le verbe http, l’URI et si nécessaire le json ou autre information à transmettre (1 page minimum)

Contiendra une analyse de la base de données à réaliser (MCD, MLD, MPD) (1 page minimum)

Contiendra une analyse de la structure du code qui sera effectuée (Organisation du code, …) (1 page minimum)

Schéma de l’architecture représentant les interactions entre le frontend et le backend tout en situant les différents composants (API REST, la base de données, l’ORM, etc.)

## Base de données

Une image contenant texte, capture d’écran, diagramme, Post-it

Description générée automatiquement

## Analyse de la structure du code

## Schéma d’interaction backend - frontend

# Réalisation

## Authentification et gestion des rôles

Comprend une explication de l'algorithme utilisé pour gérer l’authentification et la gestion des rôles (1 page minimum)

## Sécurité

Comprend une explication des mesures prises pour les aspects de sécurité. (1 page minimum)

## Technique

Comprend une explication sur chaque fonctionnalité technique demandée.

* Un ensemble de routes permettant la gestion des livres, des catégories, des utilisateurs, etc.

Chaque route s’effectue sur un fichier module javascript à part. Tous les fichiers sont dans le même dossier « routes »

|  |  |
| --- | --- |
| createBooks.mjs | Route permettant de rajouter des livres à l’API. |
| deleteBooks.mjs | Route permettant de supprimer des livres de l’API. |
| findAllBooks.mjs | Route permettant d’afficher tous les livres de l’API. |
| findBooksByCategory.mjs | Route permettant d’afficher les livres de l’API selon leur catégorie, la catégorie est choisie par l’utilisateur. |
| findBooksById.mjs | Route permettant d’afficher les livres de l’API selon leurs id, l’id est choisie par l’utilisateur. |
| findBooksByTitle.mjs | Route permettant d’afficher les livres de l’API selon leur titre, le titre est choisi par l’utilisateur. |
| login.mjs | Route permettant la connexion à l’API |
| updateBooks.mjs | Route permettant la modification des livres de l’API. |

Un fichier « helper.mjs » est stocké dans le même répertoire, il contient des fonctions utiles à aux routes.

|  |  |
| --- | --- |
| success | Permet d’écrire l’état de la requête et de l’afficher à l’utilisateur. |
| getUniqueId | Génère un nouvel id, l’id sera le plus grand id +1. |
| getBook | Retrouve un livre grâce à son id |
| removeBook | Supprime un livre grâce à son id. |
| updateBook | Met à jour un livre grâce à son id. |

* Une validation de toutes les données fournies par le consommateur de l’API
* Une gestion des statuts http (200, 3xx, 4xx, 5xx) et des erreurs.

200 pour les succès – 404 si l’URL n’est pas correcte – 500 s’il y a une erreur depuis la base de données.

* Une recherche sur les livres, catégories, etc.
* Un système d’authentification basé sur les jetons JWT
* Une documentation Swagger la plus complète possible
* Des tests de votre API avec Insomnia ou Postman
* Des tests automatisés avec vitest
* Une intégration continue dans github actions
* Une « dockerisation » du backend

# Test

Comprend une explication des tests réalisés (test automatique) (1 page minimum)

Les tests ont été effectués avec vitest.

# Conclusion

## Gestion du code

Comprend une explication de l’organisation du groupe quant à la gestion du code avec git/GitHub.

GitHub a été utilisé lors de ce projet afin de gérer le code.

## Conclusion générale

Comprend une conclusion générale sur le projet (½ page minimum)

## Conclusion personnelle

Comprend une conclusion personnelle sur le projet (½ page minimum)

## Critiques

Comprend une critique constructive sur la planification du projet (½ page minimum)

# Webographie / Bibliographie / Glossaire

* Support de cours (Step 1 à Step 14).

# Utilisation d’IA

Pour le moment nous n’avons pas utilisé d’intelligence artificielle, tel que ChatGPT, pour nous aider lors de ce projet.