05/03/2024

Tiago Rodrigues Sousa, Evin Paramanathan

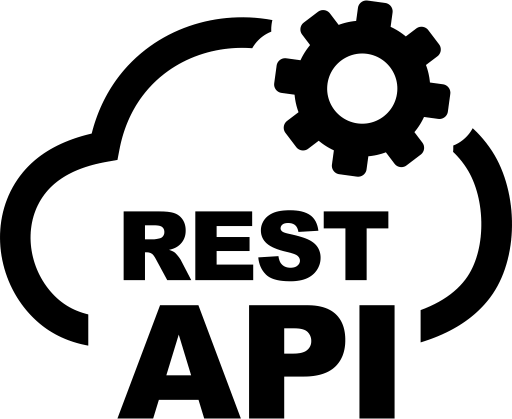
ETML – Vennes

Chef de projet : Antoie Mveng

Salle : B11

Rapport de projet

P\_Web\_C295



Rapport de projet

Table des matières

[Introduction 2](#_Toc160535605)

[Analyse 2](#_Toc160535606)

[1. Planification 2](#_Toc160535607)

[2. Verbe http 2](#_Toc160535608)

[3. Base de données 3](#_Toc160535609)

[4. Structure du code 4](#_Toc160535610)

[5. Schéma de l’architecture 5](#_Toc160535611)

[Réalisation 5](#_Toc160535612)

[1. Authentification et la gestion des rôles 5](#_Toc160535613)

[2. Sécurité 5](#_Toc160535614)

[3. Fonctionnalité technique 5](#_Toc160535615)

[a) Routes : 5](#_Toc160535616)

[b) Swagger 5](#_Toc160535617)

[c) Docker 5](#_Toc160535618)

[Test 5](#_Toc160535619)

[ChatGPT 5](#_Toc160535620)

[Conclusion 6](#_Toc160535621)

[1. Gestion du code 6](#_Toc160535622)

[2. Général 6](#_Toc160535623)

[3. Personnelle 6](#_Toc160535624)

[d) Tiago 6](#_Toc160535625)

[e) Evin 6](#_Toc160535626)

[4. Planification du projet 6](#_Toc160535627)

[Webographie 6](#_Toc160535628)

[1. Docker : 6](#_Toc160535629)

# Introduction

L'objectif de ce projet est de mettre en pratique le module ICT C295, qui consiste à réaliser le backend pour des applications. Pour ce faire, nous devons mettre en place une API qui permet de partager notre passion pour la lecture. Nous travaillons en binôme pour faire ce projet.

Pour réaliser ce projet, nous disposons d'un PC, de VS Code, d'un accès à Internet, de supports de cours, ainsi que de technologies telles que Node.js, npm, Express.js, Sequelize, MySQL, Swagger et Insomnia. Nous avons à notre disposition 24 périodes pour faire ce projet.

Avant d’aller dans les détails regardons juste qu’est-ce que c’est une API.

API est l’abréviation d’application programming interface ou interface de programmation d’application en français, une api est une interface logicielle qui permet de relier d’un service a un autre ou d’un logiciel a un autre. Cela permet d’échanger des données et des fonctionnalités

# Analyse

## Planification

La planification des taches à réaliser pour le projet a été fait sur Trello, ça permet de créer un tableau et de l’adapter pour faire un suivit du projet plus personnalisé.

<https://trello.com/invite/b/CKSDCYuq/ATTI9eb058082f456ec9f8f1b74df1999ac35786C719/pweb295>

## Verbe http

|  |  |  |
| --- | --- | --- |
| Verb | Action | Utiliser pour |
| GET |  |  |
| POST | Créer |  |
| PUT | Mettre à jours |  |
| DELETE | Supprimer |  |

## Base de données

La base de données est constitué de 6 tables, une table catégorie pour les différents catégorie de livre , une table Ecrivain avec le nom et prénom des écrivains, une table éditeur avec son nom, une table commentaire avec une note et un commentaire, une table utilisateur avec son pseudo, Mot de passe, sa date d’entrée sur le site web, le nombre d’ouvrage qu’il a proposé, et le nombre de commentaire qu’il a écrit et pour finir il y a la tables Ouvrage avec le titre du livre, le nombre de page, un résumé, l’année de l’édition, une image de la couverture(lien vers une image), un Extrait du livre et pour finir une moyenne d’appréciation du livre.

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

Description générée automatiquement

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

## Structure du code

Le projet (partie code) est organisé de cette façon : dans le dossier API qui est la racine de l’API il y a un gitignore, les packages json et un dossier src, à l’intérieur de celui-ci il y a un fichier app.mjs et 4 dossiers.

Premièrement il y a le dossier auth contient deux fichiers pour faire le système d’authentification auth.mjs et private\_key.mjs.

Ensuite il y a le dossier db qui contient aussi deux fichiers, mocks-livre.mjs et sequelize.mjs

L’avant dernière dossier est le dossier models qui contient 6 fichiers, chaque fichier est une table de la base données : donc il y a les fichiers suivant t\_categorie.mjs, t\_commentaire.mjs, t \_ecrivain.mjs, t\_editeur.mjs, t\_ouvrage.mjs et t\_utilisateur.

Pour finir il y a le dossier routes qui contient le plus de fichier. Elle contient toutes les fonctionnalités de l’API donc il y a 12 fichiers dans ce dossier voici la liste de fichiers :

addCategorie.mjs, addCommentaires.mjs, addLivre.mjs, deleteLivre.mjs, getAllLivreCategorie.mjs, getCategorie.mjs, getCommantaire.mjs, getLivre.mjs, getLivre.mjs, getLivreId.mjs, helper.mjs, login.mjs et pour finir updateLivre.mjs.

Une image contenant texte, capture d’écran, menu, Police

Description générée automatiquement

## Schéma de l’architecture

# Réalisation

## Authentification et la gestion des rôles

## Sécurité

## Fonctionnalité technique

### Routes :

Livre :

|  |  |
| --- | --- |
| addLivre |  |
| deleteLivre |  |
| getAllLivresCategorie |  |
| getLivre |  |
| getLivreId |  |
| updateLivre |  |

Catégorie :

|  |  |
| --- | --- |
| addCategorie |  |
| getCategorie |  |

Commentaire :

|  |  |
| --- | --- |
| addCommentaires |  |
| getCommentaire |  |

Login :

|  |  |
| --- | --- |
| Login |  |

### Swagger

### Docker

# Test

# ChatGPT

Pour la partie code chatGPT nous a servis à comprendre des détails qu’on ne savait pas ou qu’on ne comprenait pas. ChatGPT nous a aussi servit à créer des données factices pour les insérer dans la base de données.

Et du côté du rapport ChatGPT nous a seulement aider à corriger des fautes d’orthographe et de grammaire.

# Conclusion

## Gestion du code

## Général

Le projet est plutôt abouti toutes les fonctionnalités créer dans le dossier marche, mais le il n’est pas fini à 100% car la charge de travail est plutôt élevée par rapport au nombre de périodes que nous disposons (24 périodes) et aussi pour le nombre de personnes par groupe.

## Personnelle

### Tiago

### Evin

## Planification du projet

Pour la planification nous avons utilisé Trello comme dit dans le point 1 dans analyse, et cela nous a permis de décortiquer le projet pour mieux s’organiser, et vers la fin du projet quand on a compris que ça va être compliqué de tout finir, le fait d’avoir découper le projet auparavant nous a permis de nous réorganiser pour faire les points les plus important pour atteindre notre objectif.

# Webographie

## Docker :

<https://welovedevs.com/fr/articles/docker-node-api/>