

DELIV'CROUS

Rapport projet d'API NodeJS et MongoDB



Clarence Claux

INSA Hauts-De-France

INSA

INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
HAUTS-DE-FRANCE

Table des matières

1- Contexte.....	2
A- Reprise du sujet	2
B- Outils utilisés et leurs avantages	2
2- Problèmes rencontrés.....	3
3- Explication du code.....	4
A- Dossier Config	4
B- Dossier Controller	5
C- Dossier Models	5
D- Postman_Export.....	5
E- Routes	6
F- Autres	6

1- Contexte

A- Reprise du sujet

DelivCrous est une application mobile pour une épicerie. Celle-ci permettra de se créer un panier, y ajouter ses aliments, fruits, etc... Afin de se faire livrer à domicile. L'objectif du sujet était de s'occuper de l'API utilisée par cette application :

- Création d'utilisateur
- Gestion des aliments
- Gestion d'un panier

Il était également possible de réaliser de façon facultative :

- Identification via JSON Web Token
- Récupérer des aliments à partir d'un texte fournit

B- Outils utilisés et leurs avantages

Ce projet se construit autour du module de programmation Web, et plus particulièrement autour du Node, Express ainsi que MongoDB.

Node JS est utilisé comme moteur dans le projet, on ajoutera à cela le framework Express qui permet de construire plus rapidement et aisément un serveur mais également les routes pour celui-ci.

L'utilisation de MongoDB nous permet de construire rapidement et simplement une base de données. En effet, orientée autour du No SQ, MongoDB supprime quasiment les prérequis de la création de base de données comme par exemple l'architecture ou la conception avancée et fastidieuse des schémas de données.

L'utilisation de ces outils rends la solution proposée dans ce projet très portative et très répliquative. En effet, une fois le projet cloné il suffit simplement d'installer les dépendances via une simple commande puis une fois la base de données installée, importer les données via un fichier JSON.

2- Problèmes rencontrés

Le développement Web n'est pas quelque chose que je fais habituellement. Je travaille au centre de recherches de Vallourec et ma mission principale est la conversion de routines VBA en python ainsi que la mise en place d'un nouveau système de Workflow (Apache Airflow) pour l'automatisation d'une chaîne de calcul de simulation numérique. Le Web était quelque chose que j'essayais d'éviter car je n'étais pas très à l'aise et je n'avais pas spécialement appris grand-chose au part avant. Ce module a été pour moi une opportunité de me remettre réellement au Web et j'avoue avoir apprécié réaliser ce projet.

Évidemment, ce projet ne fut pas sans encombre, bien que celles-ci soient dû aux bonus, il me tenait à cœur de les compléter.

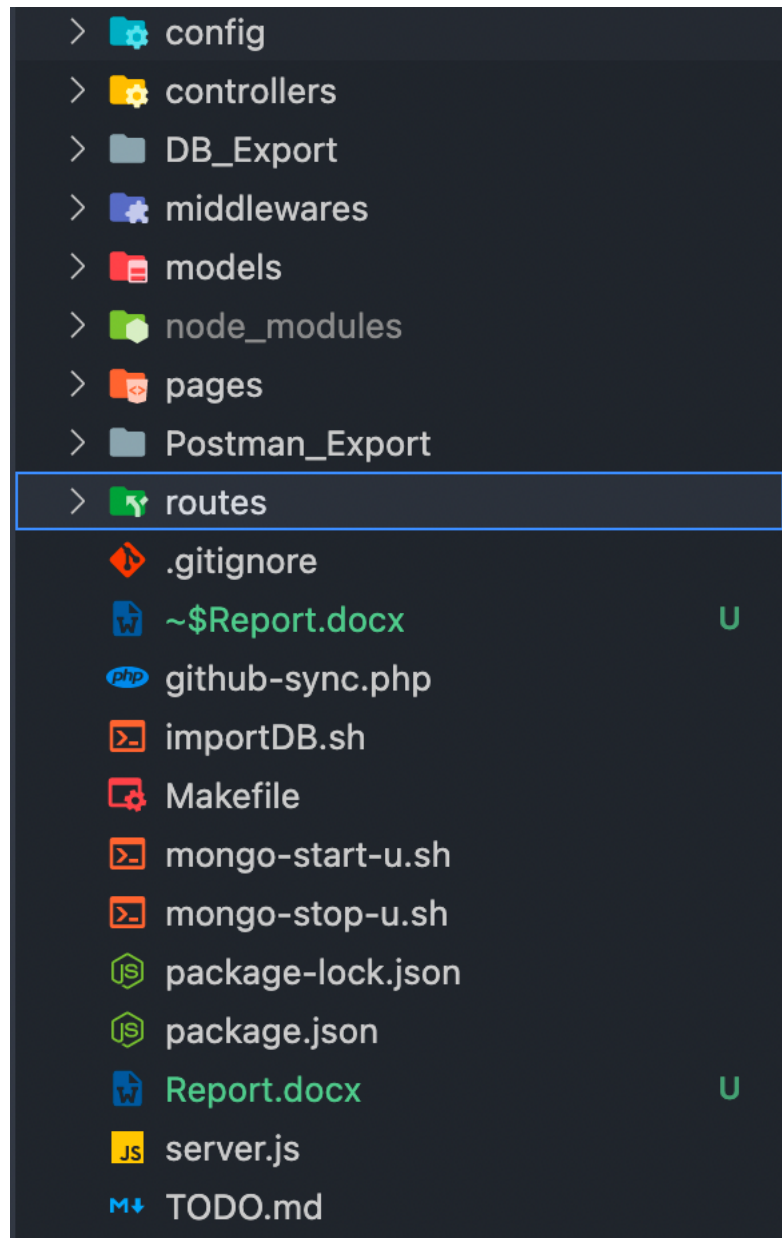
Mon premier problème fut la création et l'utilisation des JSON Web Token, en effet cela m'a poussé à reconstruire la totalité de la structure du projet afin de rendre celle-ci plus présentable.

Mon second problème fut l'exportation de l'API sur un VPS afin de rendre celle-ci disponible partout, les problèmes sont surtout liés au réseau comme par exemple la gestion des ports etc. Cette partie n'est toujours pas résolu mais cela changera sûrement ce qui pourrait expliquer des commits après la date butoir (si ceux-ci ne sont pas pris en compte pour la note cela est totalement normal, voyez cela comme un projet personnel).

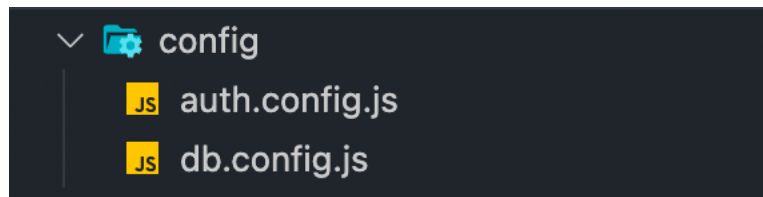
J'ai développé un petit script pour importer la base de données rapidement, celui-ci fonctionne cependant les données créer semble ne pas fonctionner lors de la connexion. Cela peut-être dû à l'encryption des mots de passe.

3- Explication du code

Le projet est structuré de la façon suivante :



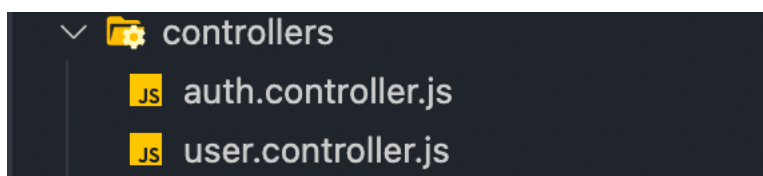
A- Dossier Config



Le fichier « auth.config.js » permet de stocker la clé d'encryption pour les tokens et les mots de passe.

Le fichier « db.config.js » comporte quant à lui les paramètres pour la base de données MongoDB.

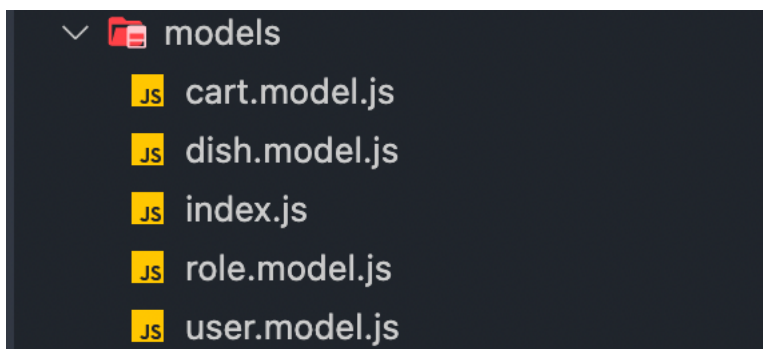
B- Dossier Controller



Le fichier « auth.controller.js » permet l'inscription et la connexion des utilisateurs. Les mots de passes sont encryptés via le module « bcrypt » afin de garantir la sécurité de ceux-ci et les JSON Web Token sont générés et valides pour 24h (nous sommes tolérants).

Le fichier « user.controller.js » permet quant à lui de vérifier les droits liés à l'utilisateur connecté (ou non). En effet, chaque utilisateur possède un rôle qui lui permet d'effectuer ou non certaines actions.

C- Dossier Models



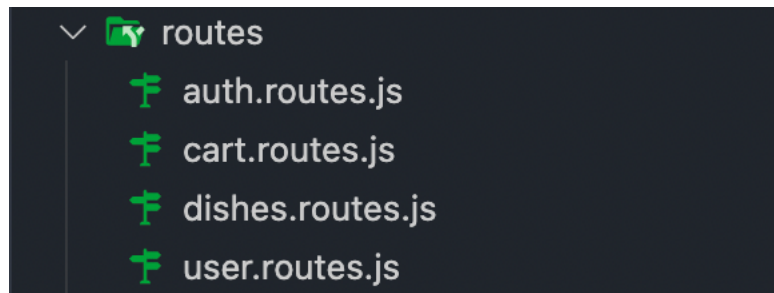
Le fichier « index.js » permet de référencer tous les modèles du projet, ce qui simplifie l'import de ceux-ci dans le projet.

Les autres fichiers sont quant à des modèles.

D- Postman_Export

Je ne pense pas qu'il soit réellement nécessaire d'en dire plus.

E- Routes



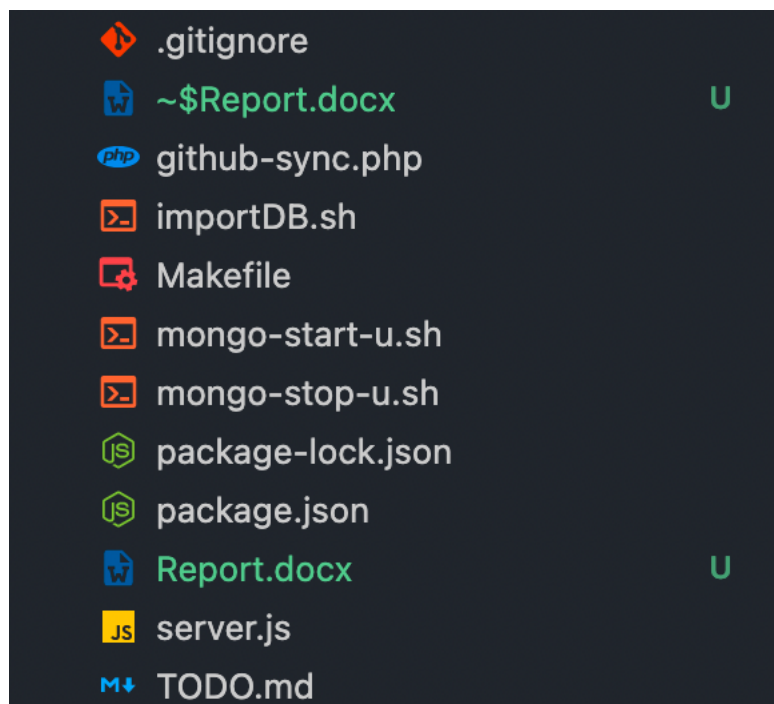
Le fichier « auth.routes.js » comporte les routes liées à l'authentification.

Le fichier « cart.routes.js » comporte les routes liées au panier.

Le fichier « dishes.routes.js » comporte les routes liées aux aliments.

Le fichier « user.routes.js » comporte les routes liées aux utilisateurs.

F- Autres



« .gitignore » pour ne pas exporter les modules node.

« importDB.sh » importer les données exportées dans le DB_Export dans la base de données DelivCrous.

« github-sync.php » pour synchroniser le serveur sur le VPS automatique lorsque le git est mis à jour.

« Makefile » utilisation de raccourcis pour lancer le projet.

« package.json » fichier maître du projet car celui-ci comporte les paramètres et les dépendances du projet.

« server.js » fichier JS principal qui permet de lancer le serveur, se connecter à la base de données, créer les rôles de base et prendre en compte les routes.

4- Conclusion

Ce projet m'aura permis de reprendre le développement web sur de bonnes bases. Il m'aura permis également d'apprendre beaucoup en faisant les parties bonus du projet ou encore en tentant de le rendre disponible sur un serveur.

Il m'aura également permis de travailler une première fois avec le No SQL, ce qui me sera utile au sein de mon entreprise puisque nous migrons nos données sur AWS (S3 et MariaDB).