

Projet NodeJs

Delivecrous

Geoffrey Auzou, Maxime Frémeaux



Conception du projet	3
Avantages et inconvénients de l'utilisation de Node et Express	4
Liste des avantages NodeJs :	4
Liste des inconvénients NodeJS :	4
Liste des avantages Express :	4
Liste des inconvénients Express :	4
Présentation du code et choix	5
Conclusion	6

Conception du projet

Nous avons choisi de rester sur le sujet initial afin de développer notre projet nommé "Delivecrous". Cette application permet de passer une commande depuis son smartphone avec des services de restauration partenaires ; le tout de manière sécurisée grâce au JsonWebToken.

La démarche d'une utilisation par un client est la suivante. Supposons que le client soit déjà enregistré dans la base de données. Le client devra se connecter grâce à un POST, ce qui aura pour effet d'envoyer un token Access nécessaire pour toute manipulation sur l'application.

Le client pourra ainsi afficher tous les plats disponibles via un **GET /dishes**. L'intégralité des plats enregistrés sur l'application sera alors affichée. On peut également afficher un plat spécifique via son id, toujours avec le verbe **GET /dish/:id**.

Le client aura bien entendu un panier afin de compléter sa commande, il lui sera donc possible d'enregistrer un plat dans son panier, via l'id du plat sélectionné, grâce à une méthode **POST /cart/:id**. Si le client n'est pas satisfait de son choix, il aura la possibilité de supprimer un plat, toujours via son id, en utilisant la méthode associée à **DELETE /cart/:id**.

Le client pourra à tout moment, afficher le panier courant via un **GET /cart**.

Afin de permettre au client de finaliser sa commande et d'être livré au bon endroit, il devra au préalable fournir une adresse (rue, code postal et ville) de par l'utilisation de la requête **PUT /cart_validation**.

Avantages et inconvénients de l'utilisation de Node et Express

Liste des avantages NodeJs :

- NodeJs ne nécessite d'utiliser qu'un seul langage de programmation à la fois, aussi bien pour le front-end que le backend
- Temps d'exécution très rapide avec JS, avec un avantage supplémentaire via le moteur v8 de google qui le rend encore plus rapide, car il peut compiler JS directement dans le code machine.
- Profite des avantages du RPM (gestion de packages)
- Solution idéale pour les applications Web en temps réel
- Déploiement facile d'une application Web
- Programme mono-thread

Liste des inconvénients NodeJS :

- L'interface API n'est pas stable, évolution constante, ce qui implique des problèmes de compatibilité.
- Mauvais dans les calculs lourds
- Mauvais avec les bases de données relationnelles, peu développées.
- Modèle de programmation asynchrone seulement.

Liste des avantages Express :

- Framework minimaliste et léger, gain de performances
- Extrêmement robuste et ne prend pas le dessus sur les fonctionnalités natives de NodeJs
- Très flexible

Liste des inconvénients Express :

- Peu de contraintes et de cadre qu'il devient très facile de partir dans tous les sens et d'aboutir sur une architecture difficile à maintenir.

Présentation du code et choix

Le code se décompose en plusieurs parties :

- La génération des données initiales est un peu particulière et donc à part dans l'architecture du projet. Elle permet d'insérer dans la base de données la liste des plats existants, d'initialiser le panier, et la liste des clients déjà inscrits. Nous sommes partis sur un total de 5 plats et 3 clients.
- La majeure partie du code, c'est-à-dire le programme principal, se trouve à la racine de notre dossier **src**, qui contient toutes les sources qui vont suivre. Ce fichier JS contient l'ensemble des requêtes back-end que nous allons pouvoir utiliser. Pour chacune, l'utilisation du *jsonWebToken* est obligatoire (sauf bien entendu pour **login**). Chaque requête contient le type d'accès ainsi que la route pour obtenir ce qu'on souhaite.
- Chaque méthode nécessite un token valide afin d'avoir un retour du serveur, dans le cas contraire, il retourne une erreur de validité. Une fonction *generateAccessToken* s'occupera de générer le token via l'username client et établira une durée de validité de 300s. De plus, une fonction *verifyToken* s'occupera de vérifier la validité d'un token avant d'effectuer une quelconque action sur le serveur. Dans le cas contraire, une erreur générée par JWT sera retournée au client. Le tout fonctionnera via un *TOKEN_SECRET* stocké sur le serveur.
C'est dans cette partie du code que nous lançons de manière infinie notre serveur, et précisons le port d'écoute. Les modèles que nous avons utilisés se trouvent dans le dossier **models**. Ils définissent les champs pour chaque type de donnée, que ce soit un Client, un Plat ou un Panier. Les schémas vont nous permettre, entre autres, d'inclure un type dans un autre type : c'est le cas de Panier, qui contient une liste de plats.
- Le dossier **postman** contient le fichier de configuration de l'environnement de test pour le développeur. Il suffit de l'importer directement dans votre application Postman. Vous retrouverez alors toutes les routes du projet.
- Le dossier **images** contient simplement les ressources que nous avons utilisées dans la documentation.
- La documentation, qui se trouve à la racine absolue du projet, dans le fichier **README.md**, contient toutes les informations nécessaires afin de tester notre projet.
- Le fichier **package-lock.json** contient la liste des dépendances et de leur version.
- Le dossier **node_modules** contient toutes les dépendances et sous dépendances.
- Le fichier **package.json** contient des informations générales sur l'application, et la liste des dépendances directes (celles qui sont explicitement demandées comme **express**, **mongoose**...)

Conclusion

Ce projet a été pour notre équipe une expérience intéressante et enrichissante dans le développement web orienté backend. Étant donné que nous ne faisons pas de NodeJs en entreprise, cela nous a permis d'apprendre les conceptions de base de NodeJs et Express avec plusieurs frameworks et librairies très utiles.

De plus, nous n'avons pratiquement jamais fait de projet web orienté backend durant nos études, ce qui nous a permis d'apprendre des conceptions et comprendre comment certaines choses fonctionnent, ce qui nous sera utile sur les projets à venir.