

# Examen API NodeJS

## Énoncé :

Une entreprise de vente de [insérer quelque chose] décide de se mettre au numérique, et fait appel à vous pour développer la partie back-end de leurs application. Ce back-end devra servir pour un site internet, une appli mobile et des bornes disponibles en boutique à la fois. Tout naturellement vous décidez de créer une API REST sur laquelle pourront interagir les trois clients. Vous décidez d'utiliser les technologies NodeJS & MySQL.

Suite à de nombreuses réunions avec le client, vous notez les spécifications suivantes :

Les produits sont caractérisés par les attributs suivants :

- Un titre
- Un prix
- Une description
- Le nombre de référence en stock actuellement
- Des étiquettes (tags)

Trois types d'utilisateurs pourront interagir sur l'appli :

## Les visiteurs

Ils pourront parcourir la liste des produits et ainsi consulter leurs titres, prix et étiquettes. Si quelque chose attire leur oeil, ils peuvent consulter une page plus détaillée avec leur description en plus des infos mentionnées précédemment.

La liste des produits devra être paginée en suivant la même stratégie que celle vue en cours. Les autres ressources n'auront pas besoin de l'être.

Si un produit n'est pas en stock il n'est pas proposé aux clients.

Ils peuvent également filtrer les produits pour n'afficher que ceux qui comporte les catégories fournies dans la requête

Ils peuvent également décider de se créer un compte en fournissant un mot de passe et une adresse mail, et bien sûr de se connecter à celui-ci.

## **Les clients**

Ils ont la possibilité d'ajouter des produits à leur panier. Bien entendu, ils peuvent sélectionner une quantité à ajouter.

Dans leur panier ils peuvent retirer des produits, et modifier les quantités.

Une fois satisfait de leur panier, ils peuvent le valider en fournissant une adresse de livraison.

Une fois le panier validé, une commande est créée et le stock des produits est mis à jour.

Les utilisateurs peuvent consulter l'ensemble de leurs commandes, avec notamment le montant totale de celles-ci.

Lorsqu'ils se connectent depuis la borne, leurs sessions sont temporaires et ne durent qu'une heure. Pour les autres plateformes, les sessions durent 30j.

## **Les administrateurs**

Ils peuvent ajouter, supprimer et modifier les produits à volonté.

Ils peuvent également administrer les étiquettes disponibles afin de pouvoir les ajouter aux produits

*Pour des raisons de simplicité, les comptes des administrateurs sont directement créés en BDD.*

## **Pistes d'amélioration**

*Ces spécifications sont purement optionnelles, et ne serviront qu'à avoir des points bonus.*

- Envoi d'un email au client lors de la validation de sa commande.
- Dans la liste des produits, les produits sont triés par popularité (score à calculer en fonction du nombre de commande comprenant ce produit, puis du nombre de visite sur la page de détail du produit dans les 3 derniers mois).
- Lier le panier non pas au compte utilisateur, mais à sa session en cours (au Token JWT). De cette façon, il pourra avoir un panier sur son smartphone, et un autre sur son ordinateur.

## Consignes

Le code devra être **clair, organisé** et **commenté** (en anglais ou en français).

Le rendu se fera sur Github dans un repo public, ou un repo privé en invitant "gabrielley68".

C'est sûr la version du repo le **23 janvier à 17h** qui sera faite la correction.

Cependant, vous êtes invités à faire des modifications jusqu'au **03 février à 08h DANS UNE NOUVELLE BRANCHE.**

Ces modifications permettront d'obtenir des points bonus, permettant aux gens insatisfait de leur travail de se rattraper.

L'utilisation de module non vu en cours est autorisé, mais devra être **justifiée**.