

Devoir Surveillé

Énoncé :

Créer un serveur express hébergeant une API Rest avec expressJS. Le serveur persiste les données via une BDD noSQL ou relationnelle. Le couple (API/web app) doit permettre aux utilisateurs de faire du CRUD (Create, Read, Update, Delete) sur un **thème de votre choix**.

Ce qui est demandé dans le projet :

- ✓ 4 verbes ou endpoints CRUD (3 pts)
- ✓ Une gestion des utilisateurs (login, mot de passe) (3 pts)
- ✓ Implémenté vous-même sans passer par une librairie
 - Les mots de passe doivent être hachés. Vous pouvez utiliser une librairie ou réutiliser du code existant pour la fonction de hachage.
 - A l'authentification, un token (chaînes de caractères aléatoires) est associée à l'utilisateur
 - A chaque requête CRUD, le client passe le token dans un header Authorization de valeur Bearer TOKEN
 - Authorization: **Bearer** TOKEN
 - A la déconnexion, le serveur supprime le token de la BDD
 - Attention : ne pas faire comme en PHP (\$_session, formulaire en \$_post, etc...)
- ✓ Le projet doit être hébergé et accessible sur internet (par exemple alwaysdata propose gratuitement un hébergement nodejs, Render, **Netlify**, Vercel, **Github pages**, Firebase Hosting) (3 pts)
- ✓ Une gestion des erreurs correcte (par exemple : interdiction d'ajout de doublons) (3 pts)
- ✓ Structurer votre application en plusieurs microservices indépendants (ex : service utilisateur, service produit, etc.). Gérer la communication entre microservices, vous pouvez utiliser un broker de messages comme RabbitMQ (ex : un service envoie une notification après une action, via une file de messages). (4pts)
- ✓ Un fichier compatible avec l'extension Rest Client qui permet de tester l'API (1 pts)
- ✓ **Fonctions en plus**, Intégration des APIs public déjà existant dans votre projet (3 pts)
 - Introduire l'API choisie (Description – Fonctionnement – Cas d'utilisations – Exemple pratique)
 - Exemples d'APIs publiques : (OpenWeather API – Google Maps API - NewsAPI/GNewsAPI – Stripe API/Paypal API – Github API – OpenAI API– Twilio API/SendGrid API - Fake store API/DummyJSON - YouTube Data API/Spotify API - Trello API/Notion API - Google Books API - Open Data APIs (data.gouv.fr) -TMDB -)
 - [Voir la liste complète des APIs publiques](#)
- ✓ Le thème est au choix (exemple : Système de gestion des emprunts d'une librairie (déjà traité en cours), Application e-commerce gérant les produits et leurs commandes (déjà traité en cours), Application de gestion de notes,)

Rendu attendu

- ✓ Le code source du projet. Vous avez le choix entre :
 - L'url github de votre projet (si le projet est privé, me donner l'accès pour cloner) et faire la mention
 - Le zip minimal du projet (sans node_modules). Il faut que j'arrive à lancer le projet en local avec un "npm install" puis un "npm start"

Devoir | « Créer une application Cloud native »

- Le nom du zip est : NomPrenom1_NomPrenom2_AppCloudNative
- ✓ L'url de votre site hébergé et fonctionnel
- ✓ Présentation ppt pour expliquer l'essentiel du projet.

Points d'attention/ conseils

- ✓ Documentez bien votre projet, vos choix techniques, et fournissez un schéma si nécessaire.