

Dans cet exercice, il vous est demandé de créer une API NodeJS qui sera testée par un site internet qui effectuera des appels sur divers endpoints dont voici la liste :

POST /api/agent/register

Paramètres : numAgent, grade, password

Retour : l'api renverra l'_id de l'agent en cas de succès.

Précision : le numAgent est un entier, il doit être unique.

// POST /api/agent/login

Paramètres : numAgent, grade, password

Retour : l'api renverra l'attribut : message : "Agent introuvable" en cas d'erreur d'authentification.

Elle renverra l'attribut : numAgent : le numero de l'agent et le token en cas de réussite.

Précision : le login d'un agent sera effectué en vérifiant son numéro d'agent, son grade et son password.

// POST /api/agent/update

Paramètre : grade

Retour : l'api renverra l'attribut : message : "Pas autorisé" si l'agent n'existe pas en base.

Elle renverra l'attribut : success : "Agent modifié" en cas de réussite.

Précision : un agent ne peut modifier que ses informations, pas celles des autres agents.

// POST /api/intervention

Paramètres : motif, lieu (optionnel), date

Retour : l'api renverra l'attribut : success : "Intervention enregistrée" en cas de réussite.

Précision : lorsque l'on ajoute une intervention dans l'API, elle sera ajoutée automatiquement à l'agent qui effectue l'ajout. Il faut donc être loggé pour y accéder.

// GET /api/intervention

Précision : cette route renverra la liste des interventions de l'agent connecté. Il faut donc être loggé pour y accéder.

// GET /api/intervention/all

Précision : cette route listera toutes les interventions stockées

// DELETE /api/intervention/:id

Retour : l'api renverra l'attribut : message : "Ce n'est pas votre intervention" si l'agent tente de modifier une intervention qui ne lui appartient pas.

Elle renverra : success : "Intervention supprimée" en cas de réussite.

Précision : seul un agent connecté peut supprimer ses propres interventions

Une fois votre API terminée :

Veuillez créer un repository public sur github.

Si vous savez utiliser git, pusher votre travail dessus.

Si vous ne savez pas, un simple drag and drop de vos fichiers sur votre repository fonctionnera.

Ne postez pas le dossier node_modules.

Ensuite, vous allez mettre votre application en production via Render.com.

- Créez un compte sur render.com
- Choisissez "New Web Service"
- Copier / coller votre Public Git Repository dans le champ prévu et faites "continue"
- Donnez un Name à votre projet, mettez "npm install" au lieu de "yarn" dans build command et renseignez "node server" dans start command.

Le serveur se déploiera.

Si vous faites un nouveau commit sur votre repository, il vous faudra cliquer sur manual deploy -> clear build cache & deploy.

Votre api sera alors en ligne !

Pour que je puisse la tester, rdv sur : api.agence-pixi.fr

Rentrez votre nom/prénom et l'url de votre render.com indiquée en haut à gauche.