



Econobots - Installation

Dépendances système

L'application web a besoin que deux éléments soient présents sur le système sur lequel le backend est lancé :

- Node.js
- Mongo DB

Architecture du projet

Le projet contient quatre dossiers :

- *backend* : contenant le code serveur, basé sur le framework *Node.js Express*
- *frontend* : contenant le code frontend, basé sur **React**
- *scripts* : contenant tous les scripts nécessaires au lancement de l'application
- *docs* : contenant la documentation

Seuls les dossiers *backend* et *scripts* vont nous intéresser ici.

Dossier backend

Ce dossier contient un fichier important pour la configuration : *config.js*. On y trouve un certain nombre de paramètres qu'il est possible d'éditer.

Voici une liste des plus importants, concernant le backend lui-même :

- *server.ssl* permet de lancer le backend en https ou non
- *server.port* permet de spécifier le port d'écoute du backend
- *database.admin.password* : permet de définir le mot de passe du compte *admin* créé par défaut dans la base de données

Puis concernant le robot :

- *robot.ip* : permet de spécifier l'IP du robot (afin que le backend puisse faire des appels à l'API REST)
- *robot.port.rest* : permet de spécifier le port d'écoute du serveur exposant l'API REST
- *robot.port.camera* : permet de spécifier le port d'écoute du système de caméra du robot

Puis concernant le pilotage :

- *pilot.time* : permet de spécifier le temps imparti au pilote lors de l'activation du mode manuel
- *pilot.recordingTime* : permet de spécifier le temps maximal d'enregistrement vidéo

Dossier scripts

Ce dossier contient tous les scripts permettant le démarrage de l'application :

- *compile.sh* : télécharge les dépendances du backend et du frontend de l'application puis compile une version de production du frontend et la place où il faut dans les dossiers du backend
- *generate-self-signed-ssl-certificate.sh* : génère un certificat SSL auto-signé (seulement utile lorsque le paramètre *server.ssl* du backend est à *true*)

- *init-database.sh* : crée (supprime si existant) la base de données nécessaire au fonctionnement du backend, puis y ajoute l'utilisateur *admin*.
- *start.sh* : démarre une instance du backend en arrière plan en utilisant *nohup* afin de détacher le processus du terminal
- *all.sh* : exécute les quatre autres scripts dans le bon ordre

Il est bien entendu possible de modifier le script *start.sh* afin de lancer la plate-forme d'une autre façon (avec Docker par exemple, ou autre)