Loaner - Logiciel de gestion des emprunts de matériel

Présentation

Loaner est une application web permettant de gérer l'emprunt de matériel informatique dans le temps.

L'application utilise les technologies et frameworks suivantes : - NodeJS - NPM (Node Package Manager) - Express - SQLite - VueJS

Pour les tests et la vérification de la qualité du code, les technologies suivantes sont utilisées : - **Mocha / Chai** : Pour les tests unitaires sur le backend - **Cypress** : Pour les tests de bout en bout - **Supertest** : Pour les tests sur les routes et les contrôleurs dans le backend - **ESLint** : Pour la vérification de la qualité de la syntaxe - **NYC** : Pour vérifier le taux de couverture des tests unitaires

Le dépôt Git est accessible ici.

Documents importants

Voici une liste des documents importants pour la compréhension du projet : - *Cahier de Specifications_V1.pdf* : Cahier de spécifications du projet fourni au client - *Cahier de test.docx* : Cahier de recettes crée avec Squash-TM compilant tous les tests du backend - *backend_test_reports/mochawesome.html* : Rapport d'exécution des TU du backends généré avec l'outil mocha-awesome - *Rapport.pdf* : Rapport sur les points qui ont été importants sur le projet -

Structuration du dépôt

Le dépôt est composé de 3 branches : - **Branche main** : Branche sur laquelle se trouve tout les documents utiles et la version production du logiciel - **Branche frontend** : Contient la partie web du projet ainsi que des tests de bout en bout avec Cypress - **Branche backend** : Contient la partie serveur du projet faisait fonctionner l'API REST avec un base de données; Contient des tests effectuées avec Mocha/Chai et Supertest

Arborecence du projet

Voici la liste des fichiers indispensable au bon fonctionnement du projet ainsi que leur utilité : - **database.db** : Fichier contenant la base de données à format de SQLIte3 - **dist/** : Dossier contenant la partie backend du projet (serveur et API REST) - **dist/util/** : Dossier contenant des programmes utilitaires - **dist/config.json** : Fichier de configuration du serveur (*cf. partie Configuration*) - **init_db.sql** : Fichier contenant les instructions SQL permettant de créer la base de données initiale - **package.json** : Fichier utilisé par NPM, contient les informations sur le projet et la liste de ses dépendances - **package-lock.json** : Fichier interne à NPM - **static/** : Dossier contenant la partie frontend du projet

Mise en place du logiciel

Installation des dépendances

Une fois les fichiers de ce dépôt récupéré, il est nécessaire de faire télécharger les dépendances du projet par NPM en exécutant la commande suivante :

```
npm install
```

Création de la base de données

Une fois cela fait, il vous sera ensuite possible de créer une base de données initiale pour le projet. La méthode la plus simple consiste à exécuter cette commande :

```
npm run create-db
```

Cette commmande va lire le contenu du fichier *init_db.sql* et va créer une fichier de base de données *database.db* à partir de celui-ci.

Si vous souhaitez créer la base de données à partir de votre propre fichier SQL, exécutez la commande suivante :

```
node dist/util/create_db.js <fichier SQL> --exec [--log-queries]
```

Il existe un fichier *empty_db.sql* permettant de créer une base de données vide avec un seul utilisateur administrateur à l'intérieur.

Il est aussi possible de générer un mot de passe afin de l'insérer dans la base de données avec la commande :

```
node dist/util/hash_password.js
```

De la même façon, il possible de générer un token de connexion JWT avec la commande :

```
node dist/util/gen_token.js
```

Configuration

Dans le dossier *dist/* ce trouve le fichier de configuration du serveur *config.json* se présentant sous la forme suivante :

```
{
    "serverPort": 3000,
    "dbFile": "database.db",
    "useStatic": true,
    "jwtSecret": "*Phrase secrete pour la génération du token JWT*",
    "hashSaltRounds": 10
}
```

Voici une description pour chacun des paramètres du fichier :

Paramètre	Valeur par défaut	Description
serverPort	3000	Port sur lequel écoute le serveur (attention à ne pas mettre un numéro de port nécessitant les privilèges administrateurs)
dbFile	database.db	Fichier contenant la base de données
useStatic	true	Permet de savoir si le serveur fonctionne en mode normal ou en mode "back-end uniquement" (à laisser à tel quel)
jwtSecret	*Phrase secrète*	Phrase secrete pour la génération du token JWT
hashSaltRounds	10	Nombre de tours effectué par l'algorithme de hachage BCrypt lors de la génération du "sel" des mots de passes

Lancement

Vous pouvez lancer le serveur en appelant NodeJS et en lui précisant le fichier d'entrée de l'application avec la commande :

```
node dist/server.js
```

Il existe aussi un raccourci avec la commande :

```
npm start
```

Une fois cela fait, il ne vous reste plus qu'à attendre que le message suivant s'affiche, vous confirmant que l'application est bien lancée :

> Server is listening on port **numéro de port du serveur**

Utilisation

Une fois votre serveur lancé, il ne vous reste plus qu'à vous rendre à l'adresse suivante :

```
http://<adresse du serveur>:<port du serveur>
```

Pour vous connecter, il existe 2 utilisateurs créés par défaut :

Prénom Nom	Email	Mot de passe	Adminstrateur ?
Lilian Bethus	lilianb@mail.fr	fromage	Oui
Milan Pasquereau	mpsqr@mail.fr	bourbe	Non

Il existe aussi un lot de 3 catégories contenant au total 3 matériels différents : **Téléphones** : - Samsung Galaxy S1000 - Huawei P80

Ordinateurs : - Acer Pro Max

Tablettes : *Catégorie vide*