Développement Avancé

TP1 - Hachage - Wania Jean-Pierre (302)

Thèmes

- Les promises
- Le module node :fs
- Le module node :crypto

Dans cette première étape, j'ai commencé par cloner le projet depuis GitHub et j'ai ouvert le fichier src/server.js. J'ai ajouté une instruction pour vérifier le bon fonctionnement du serveur HTTP en affichant un message identifiable dans la console. Ensuite, j'ai utilisé des outils comme Postman pour valider les méthodes GET/POST sur http://localhost:3000/blockchain et confirmer le fonctionnement du serveur.

Pour la deuxième étape, j'ai travaillé sur le développement de la fonction findBlocks() dans le fichier src/blockchainStorage.js. Cette fonction était chargée de récupérer l'ensemble de la blockchain à partir du fichier blockchain.json et de la retourner au client sous forme de JSON. J'ai créé le dossier data à la racine du projet et ajouté un fichier blockchain.json avec un contenu JSON de test. En utilisant le module fs/promises, j'ai implémenté la lecture asynchrone du fichier JSON et géré les erreurs si le fichier n'existait pas.

Dans la troisième étape, j'ai développé la fonction createBlock() pour ajouter de nouveaux blocs dans le fichier. J'ai généré l'ID avec la fonction uuidv4() du module uuid et implémenté la méthode getDate() dans src/divers.js pour obtenir le timestamp au format demandé. J'ai également rempli les champs nom et don avec les valeurs transmises lors de la requête POST.

Enfin, dans la quatrième étape, j'ai écrit la fonction findLastBlock() pour retourner le dernier bloc de la chaîne. En utilisant la classe Hash du module crypto, j'ai calculé la valeur de hachage SHA-256 d'une chaîne de caractères et ajouté le champ hash à chaque bloc en calculant la valeur de hachage SHA-256 du bloc précédent.

Bilan
Globalement, ce premier TP était très simple. Une bonne introduction à node.js et à son fonctionnement.