

Développement Avancé - Compte rendu TP1

Étape 1 – état des lieux

Pour récupérer le projet, je l'ai *fork* sur GitHub et cloner sur ma machine.

Pour ce qui est des messages à afficher en console, j'ai ajouté un `console.log()` dans les cas des méthodes GET ou POST avec notamment une petite mention ("[INFO]") et un message pour dire que la requête a bien aboutie.

Je n'ai pas rencontré de difficultés sur cette partie.

Étape 2 – apprenons à lire

J'ai ajouté dans un dossier `data` le fichier `blockchain.json` et l'ai initialisé avec l'objet `{"message" : "Bonjour à tous"}` pour le tester. Ensuite, j'ai consulté le site <https://nodejs.org/api/fs.html> pour savoir comment lire dans un fichier Json et coder la fonction `findBlocks()`.

Je n'ai pas non plus eu de difficultés particulières.

Étape 3 – une brique après l'autre

Pour la méthode `createBlock()`, j'ai commencé par récupérer les blocs avec la méthode `findBlocks()`, puis j'ai créé un nouveau bloc à insérer dans la liste des blocs (le fichier Json contenant à présent un tableau vide (`[]`)). Ce nouveau bloc contient notamment deux choses différentes du sujet : pour *nom*, si la chaîne est vide ou nulle, il devient "MystérieuxInconnu" ; pour *don*, pareil, il devient "0". Finalement, on intègre le nouveau bloc aux autres et on réécrit sur le fichier Json.

J'ai fait des tests avec Postman pour vérifier si la fonction marchait.

J'ai eu des problèmes pour intégrer mon nouveau bloc, parce qu'au début, je n'avais pas nettoyé mon fichier Json.

Étape 4 – « Vers l’infini et au-delà ! »

Coder la fonction *findLastBlock()* a été très rapide et simple. J’ai juste une petite interrogation sur le renvoi de la fonction : je ne savais pas si elle renvoyait bien une Promesse.

Ensuite pour le hachage, je n’avais pas compris que *createHash()* renvoyait un hash et que celui-ci devait être utilisé pour générer nos hachage, je l’utilisais directement comme étant le hash en question. Une fois cela compris, j’ai pu simplement utiliser le hash. Par ailleurs, pour le premier bloc, j’ai haché avec le paramètre “*genesis*” comme celui-ci ne reposait pas sur le hash du bloc précédent.

Mis à part le problème énoncé, je n’ai pas eu trop de mal à passer cette étape.

Pour aller plus loin...

Pour les fonctions facultatives, j’ai codé la fonction *findBlock(id)* avec un filtre qui ne garde que les *id* des blocs et qui vérifie si l’*id* est identique à celui passé en paramètre de fonction.

Je n’ai pas codé la seconde fonction.