



IUT de Paris - Rives de Seine
Université Paris Cité

Projet

Programmation avancée

21/01/2025

Zheng Mickaël (304)

IUT de Paris - Rives de Seine

BUT 3 Info

Étape 1 – état des lieux.....	3
Étape 2 – apprenons à lire.....	3
Étape 3 – une brique après l'autre.....	3
Étape 4 – « Vers l'infini et au-delà ! ».....	3

Étape 1 – état des lieux

J'ai fork le projet à la place de le cloner, en effet, si j'avais juste cloné le projet, je n'aurais pas pu push les commits que j'aurais fait. Aucune difficulté concernant le développement, la seule complication était le fait que j'ai cloné le projet original puis j'ai fork le projet, j'ai donc dû modifier un peu mon répertoire local pour faire en sorte d'avoir le dépôt original ainsi que mon fork.

Pour tester le bon fonctionnement du serveur http, j'ai simplement ajouter un `console.log` pour afficher un message lors d'un requête GET à la route `/blockchain`

Étape 2 – apprenons à lire

Pas de grande difficulté, je suis allé me renseigner sur l'utilisation de `readFile` dans `node:fs` et non `node:fs/promises`, alors qu'ils n'ont pas exactement le même fonctionnement. J'ai perdu un peu de temps sur cela.

Pour la fonction `findBlocks` j'ai ajouté la lecture d'un fichier json puis je le parse en objet json, pour ensuite le renvoyer.

Étape 3 – une brique après l'autre

J'ai installé la dépendance `uuid`, mais je n'ai pas trouvé de fonction `uuidv4` dans le module `uuid`, mais on m'a dit qu'il fallait utiliser `v4`.

Pour la fonction `createBlock`, j'ai utilisé le spread operator (...) pour ajouter le nouveau block à la fin des anciens blocks. De plus, je m'appuie sur `findBlocks` pour obtenir les anciens blocks. J'écrase également l'ancien fichier pour écrire le nouveau block.

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

J'ai rencontré quelque difficultés, en effet, j'ai utilisé la fonction `toString` sur le résultat de la fonction `findLastBlock`, sauf que la représentation en String d'un block est `[Object Object]`, il fallait à la place utiliser `JSON.stringify` pour pouvoir utiliser une fonction de hashage.