

BUT 3
Parcours
Informatique

R6.Real 05 D v. avanc 



Nom de l tudiant : Tony ZHENG

Promotion : 2023 - 2024

Enseignant R f rent : Laurent Giustignano
Enseignant R f rent : Manel GouiderLaurent Giustignano
143 Avenue de Versailles, 75016 Paris

Étape 1 – état des lieux :

Vérification du bon fonctionnement du serveur HTTP avec une modification dans src/server.js.

Utilisation de Postman pour valider les méthodes GET/POST sur le point d'extrémité

<http://localhost:3000/blockchain>.

Pour avoir de bonne base je dirais.

Étape 2 – apprenons à lire :

Création du fichier blockchain.json dans le dossier data pour stocker la blockchain.

Lecture du contenu de blockchain.json avec le module fs/promises et retour des valeurs au format JSON lors de l'appel GET /blockchain.

Assez facile.

Étape 3 – une brique après l'autre :

Développement de la fonction createBlock() pour ajouter des blocs dans le fichier.

Utilisation du module uuid pour générer l'identifiant du bloc.

Mise en place de la méthode getDate() dans src/divers.js pour obtenir le timestamp au format requis.

Remplissage des champs nom et don avec les valeurs transmises lors de la requête POST.

Les blocs sont ajoutés au fichier blockchain.json sans implémentation de la fonctionnalité de hachage.

c'était la partie la plus compliqué, notamment des problèmes de syntaxe “= = =” n'est pas égale a “= =” 😭

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

Création de la fonction findLastBlock() pour récupérer le dernier bloc de la chaîne.

Utilisation de la classe Hash du module crypto pour calculer la valeur de hachage SHA256 d'une chaîne de caractères.

Intégration du champ hash dans chaque bloc en calculant la valeur de hachage du bloc précédent.

Le nouvel enregistrement est retourné au client.

Rien de très compliqué.

