

## COMPTE RENDU TRAVAUX PRATIQUE – SUJET 1

### Etape 1 :

J'ai configuré le serveur avec le module **node:http** et ajouté des **console.log** dans le fichier **server.js** pour suivre les appels. J'ai utilisé la fonction **createServer()** pour démarrer le serveur et écouter les requêtes. La requête **GET /blockchain** fonctionne mais ne renvoie rien, car aucun fichier JSON valide n'est encore configuré. La requête **POST /blockchain** ne fonctionne pas pour l'instant.

### Etape 2 :

J'ai créé le fichier **blockchain.json** dans le dossier **data** avec un message initial pour tester. Ensuite, j'ai utilisé les fonctions suivantes de la bibliothèque **node:fs/promises** :

- **readFile** : Pour lire le fichier JSON contenant la blockchain.
- **JSON.parse** : Pour convertir le contenu JSON du fichier en objet JavaScript.

J'ai implémenté la fonction **findBlocks()** pour lire le fichier et renvoyer son contenu. En testant avec **GET /blockchain**, la réponse affiche correctement les données du fichier.

### Etape 3 :

Pour ajouter un nouveau bloc, j'ai utilisé :

- **uuidv4** (de la bibliothèque **uuid**) : Pour générer un **id** unique pour chaque bloc.
- **writeFile** (de **node:fs/promises**) : Pour sauvegarder les données mises à jour dans **blockchain.json**.
- **getDate()** (fonction personnalisée dans **divers.js**) : Pour obtenir la date et l'heure au format demandé.

J'ai implémenté la fonction **createBlock()** qui ajoute un bloc avec ces informations. En testant avec **POST /blockchain**, les blocs sont bien ajoutés dans le fichier JSON.

### Etape 4 :

Pour le calcul des hash, j'ai utilisé :

- **createHash** (de la bibliothèque **node:crypto**) : Pour générer un hash SHA256 basé sur les données du dernier bloc.
- **JSON.stringify** : Pour convertir les données du dernier bloc en chaîne de caractères avant de les hacher.

J'ai implémenté la fonction **findLastBlock()** pour récupérer le dernier bloc et calculer le champ **hash** de chaque nouveau bloc. J'ai vérifié les résultats avec des outils externes. Les nouveaux blocs incluent désormais des hashes valides.

**Pour aller plus loin :**

Je pourrais utiliser les fonctions suivantes :

- **createHash** (de **node:crypto**) : Pour vérifier l'intégrité de la blockchain avec **verifBlocks()**.
- **findBlock()** : Pour rechercher un bloc spécifique en utilisant son **id**.