Calliclès Bazolo
302

TP1 Blockchain

Étape 1

ajout de `console.log("Bonjour a tous j'èspère que vous allez bien c'est Romain et on se retrouve aujourd'hui pour le traditionnel debrief d'après Match
")` dans server.js

```
try {
    switch (endpoint) {
        case 'GET:/blockchain':
            results = await liste(req, res, url)
            console.log("Bonjour a tous j'èspère que vous allez bien c'est Romain et on se retrouve  aujourd'hui pour le traditionnel debrief d'après Match")
            break
        case 'POST:/blockchain':
            results = await create(req, res)
            break
        default :
            res.writeHead(404)
    }
```

Résultat :

```
Restarting 'src/server.js'
Bonjour a tous j'èspère que vous allez bien c'est Romain et on se retrouve  aujourd'hui pour le traditionnel debrief d'après Match
```

Étape 2

Création du fichier **blockchain.json** dans un répertoire **data** :

```
{"message" : "Bonjour à tous","penser" :  "Aurevoir tout le monde"}
```

Suite à cela j'ai mis à jour le chemin dans le **blockchainStorage.js** pour pouvoir lire le fichier, j'ai aussi par la suite mis à jour la fonction findblocks :

```
/* Chemin de stockage des blocks */
const path : string  = "./data/blockchain.json"

/**
 * Mes définitions
 * @typedef { id: string, nom: string, don: number, date: string,hash: string} Block
 * @property {string} id
 * @property {string} nom
 * @property {number} don
 * @property {string} date
 * @property {string} string
 *
 */

/**
 * Renvoie un tableau json de tous les blocks
 * @return {Promise<any>}
 */
2 usages   Laurent Giustignano *
export async function findBlocks() : Promise<any>  {
    return JSON.parse(await readFile(path, "utf-8"));
}
```

Résultat :

```
{
    "message": "Bonjour à tous",
    "penser": "Aurevoir tout le monde"
}
```

Étape 3

J'ai mis en place la fonction `createBlock()` pour ajouter les blocs dans le fichier. J'ai commencé par récupérer les blocs existants en les mettant dans un tableau. Ensuite, j'ai rajouté le nouveau bloc à la fin de ce tableau.

```
export async function createBlock(contenu) : Promise<...>  {
    let blocks : any[]  = [];
    const block = await findBlocks();
    blocks.push(block)
    const newBlock : {...}  = {
        id: uuid(),
        date: getDate(),
        nom: contenu.nom,
        don: contenu.don
    };
    blocks.push(newBlock);
    await writeFile(path, JSON.stringify(blocks, replacer: null, space: 2));
    return blocks;
}
```

Résultat :

```
[
  {
    "message": "Bonjour à tous",
    "penser": "Aurevoir tout le monde"
  },
  {
    "id": "b554f5d5-86cc-427f-ac42-0c58813f180e",
    "date": "Fri Jan 26 2024 11:30:46 GMT+0100 (heure normale d'Europe centrale)"
  }
]
```

Étape 4

Dans l'étape 4, j'ai ajouté la fonction `findLastBlock()` pour obtenir le dernier bloc de ma chaîne blockchain, simplifiant ainsi sa gestion. J'ai aussi mis à jour la fonction `createBlock()` en calculant désormais le hachage SHA256 du bloc précédent et en l'ajoutant à chaque nouveau bloc créé.

```
export async function findLastBlock() : Promise<...>  {
    const blocks = await findBlocks();
    return blocks.length > 0 ? blocks[blocks.length - 1] : null;
}

/**
 * Creation d'un block depuis le contenu json
 * @param contenu
 * @return {Promise<Block[]>}
 */

2 usages  Laurent Giustignano *
export async function createBlock(contenu) : Promise<...>  {
    let blocks : any[]  = [];
    const block = await findBlocks();
    blocks.push(block)
    const newBlock : {...}  = {
        id: uuid(),
        date: getDate(),
        nom: contenu.nom,
        don: contenu.don
    };
    const lastBlock : {id: string, nom: string, don:... | null  = await findLastBlock();
    if (lastBlock) {
        const data : string  = JSON.stringify(lastBlock);
        const hash : Promise<ArrayBuffer>  = crypto.createHash('sha256').update(data).digest( algorithm: 'hex');
        newBlock.hash = hash;
    }

    blocks.push(newBlock);
    await writeFile(path, JSON.stringify(blocks,  replacer: null,  space: 2));
    return newBlock;
}
```

Résultat :


```
[
  [
    {
      "message": "Bonjour à tous",
      "penser": "Aurevoir tout le monde"
    },
    {
      "id": "b554f5d5-86cc-427f-ac42-0c58813f180e",
      "date": "Fri Jan 26 2024 11:30:46 GMT+0100 (heure normale d'Europe centrale)"
    }
  ],
  {
    "id": "bd022e61-c356-4fc8-9e31-f326686e4f59",
    "date": "Fri Jan 26 2024 11:50:23 GMT+0100 (heure normale d'Europe centrale)",
    "hash": "f1d647794b8ea243de2baba9b4c6d08bf86684e188b4d31e4b2780f8bcfedbf6"
  }
]
```