

TP4 : Création d'une Blockchain en Python

Abdelkader Ouared

October 27, 2024

1 Introduction

Ce TP a pour objectif de vous initier aux blockchains en développant une implémentation en Python.

2 Prérequis

- Python 3.8 ou supérieur.
- Bibliothèque pycryptodome.

3 Installation

Créez un fichier `requirements.txt` contenant :

```
pycryptodome
```

Installez les dépendances avec :

```
pip3 install -r requirements.txt
```

4 Structure du Projet

Organisez votre projet comme suit :

```
my_blockchain_project/  
    node/  
        __init__.py  
        block.py  
        utils.py  
    main.py  
    requirements.txt
```

5 Implémentation

5.1 Fichier `utils.py`

```
from Crypto.Hash import SHA256  
  
def calculate_hash(data: bytes) -> str:  
    h = SHA256.new()  
    h.update(data)  
    return h.hexdigest()
```

5.2 Fichier block.py

```
import json
from node.utils import calculate_hash

class Block:
    def __init__(self, timestamp: float, transaction_data: str, previous_block=None):
        self.transaction_data = transaction_data
        self.timestamp = timestamp
        self.previous_block = previous_block

    @property
    def previous_block_cryptographic_hash(self):
        return self.previous_block.cryptographic_hash if self.previous_block else ""

    @property
    def cryptographic_hash(self) -> str:
        block_content = {
            "transaction_data": self.transaction_data,
            "timestamp": self.timestamp,
            "previous_block_cryptographic_hash": self.previous_block_cryptographic_hash
        }
        return calculate_hash(json.dumps(block_content, indent=2).encode('utf-8'))
```

5.3 Fichier main.py

```
import time
from node.block import Block

genesis_block = Block(timestamp=time.time(), transaction_data="Genesis_Block")
second_block = Block(timestamp=time.time(), transaction_data="Block_2", previous_block=genesis_block)

print("Genesis_Block_Hash:", genesis_block.cryptographic_hash)
print("Second_Block_Hash:", second_block.cryptographic_hash)
```

6 Exécution

Pour lancer le projet, exécutez :

```
python3 main.py
```

7 Conclusion

Vous avez créé une blockchain simple en Python, comprenant la création de blocs et l'utilisation de hachages pour garantir l'intégrité des données.

Devoirs (Homework)

Instructions : Créez un bloc contenant des informations sur les patients. Par exemple :

```
patient_block = Block(timestamp=time.time(), transaction_data="Patient_Information: John_Doe, Age: 30, Condition: Healthy", previous_block=genesis_block)
print("Patient_Block_Hash:", patient_block.cryptographic_hash)
```

Objectif : Expérimentez avec l'ajout de différents blocs contenant des informations variées sur les patients.