Rujuta Vinit Medhi - 33

Experiment: 03

Aim: Create a Cryptocurrency using Python and perform mining in the Blockchain created.

Theory:

1. Blockchain Overview

Blockchain is a **distributed and decentralized ledger** that stores information in a series of linked blocks.

Each block contains:

- Transaction data
- Timestamp
- Previous block's hash
- Its own unique hash (digital fingerprint)

Once data is recorded in a blockchain, it becomes **immutable** because altering one block would require recalculating all subsequent blocks.

2. Mining

Mining is the process of:

1. Collecting pending transactions into a block.
2. Performing a computational puzzle (Proof-of-Work) to find a valid hash.
3. Adding the new block to the blockchain.
   Broadcasting it to all connected peers.

Miners are rewarded with cryptocurrency for successfully mining a block.

3. Multi-Node Blockchain Network

In this lab, we simulate **three independent blockchain nodes** (5001, 5002, 5003).

Each node:

- Runs on a separate port.
- Maintains its own copy of the blockchain.
- Can connect with peers to share and validate blocks.

4. Consensus Mechanism

We use the **Longest Chain Rule**:

- If multiple versions of the chain exist, the **longest valid chain** is chosen.
- This ensures all nodes agree on a single transaction history.

5. Transactions & Mining Reward

Each transaction has:

- Sender
- Receiver
- Amount

When mining a block:

- Pending transactions are added to the block.
- A reward transaction is added automatically to pay the miner.

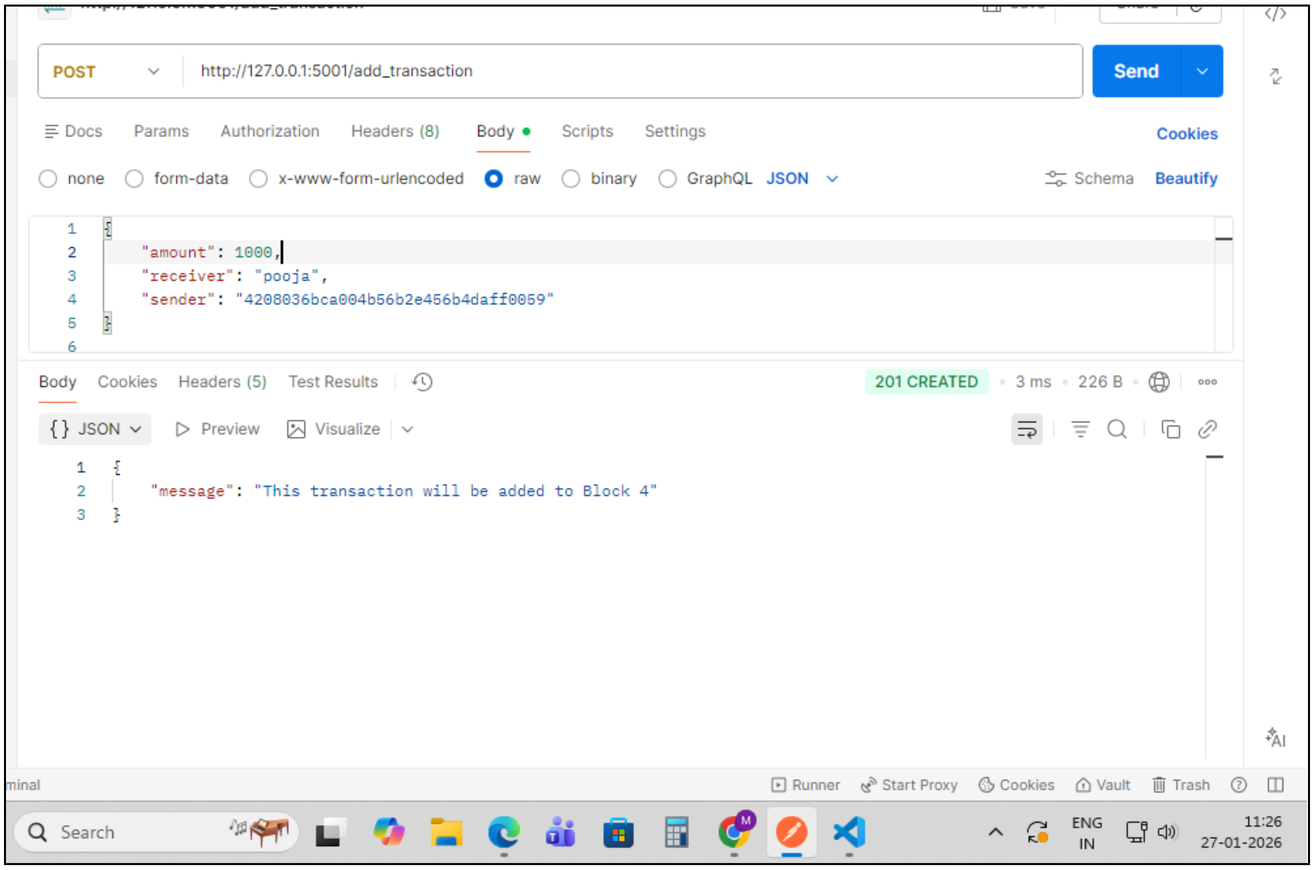
6. Chain Replacement

When /replace_chain is called:

1. Node requests chains from peers.
2. If it finds a longer and valid chain, it replaces its own.
3. This keeps the blockchain consistent across all nodes.

Implementation:

1. Add Transactions - invoke add_transactions() as a POST request.



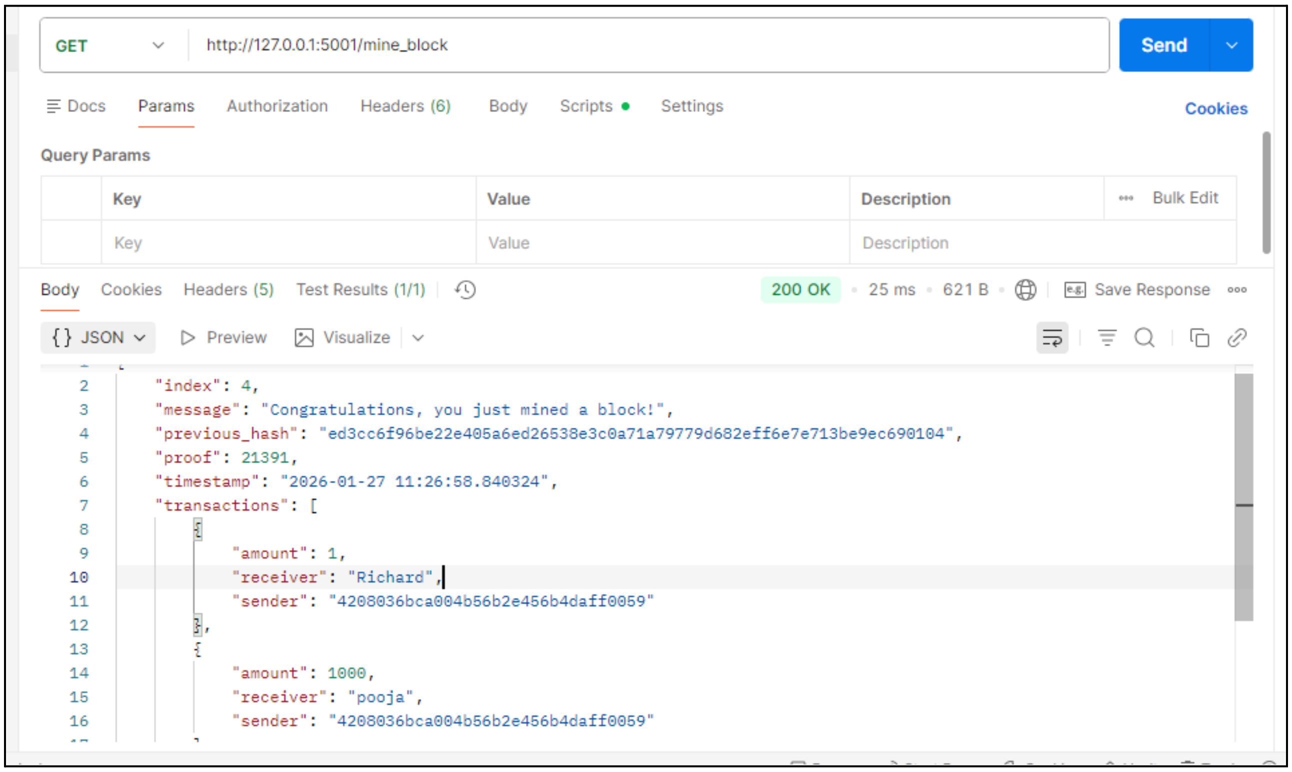
POST http://127.0.0.1:5001/add_transaction

```
{
    "amount": 1000,
    "receiver": "pooja",
    "sender": "4208036bca004b56b2e456b4daff0059"
}
```

201 CREATED · 3 ms · 226 B

```
{
    "message": "This transaction will be added to Block 4"
}
```

2. mining - mine_block()

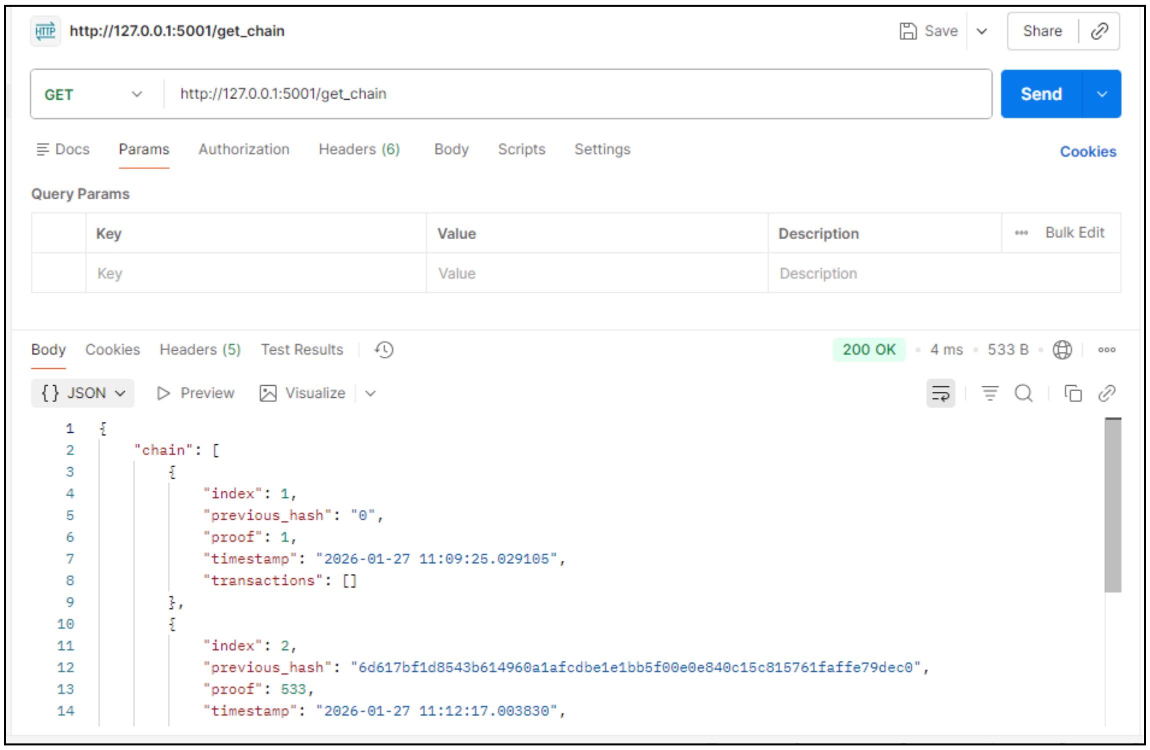


http://127.0.0.1:5001/mine_block

200 OK · 25 ms · 621 B

```
    "index": 4,
    "message": "Congratulations, you just mined a block!",
    "previous_hash": "ed3cc6f96be22e405a6ed26538e3c0a71a79779d682eff6e7e713be9ec690104",
    "proof": 21391,
    "timestamp": "2026-01-27 11:26:58.840324",
    "transactions": [
        {
            "amount": 1,
            "receiver": "Richard",
            "sender": "4208036bca004b56b2e456b4daff0059"
        },
        {
            "amount": 1000,
            "receiver": "pooja",
            "sender": "4208036bca004b56b2e456b4daff0059"
```

3.fetch the chain - get_chain(),



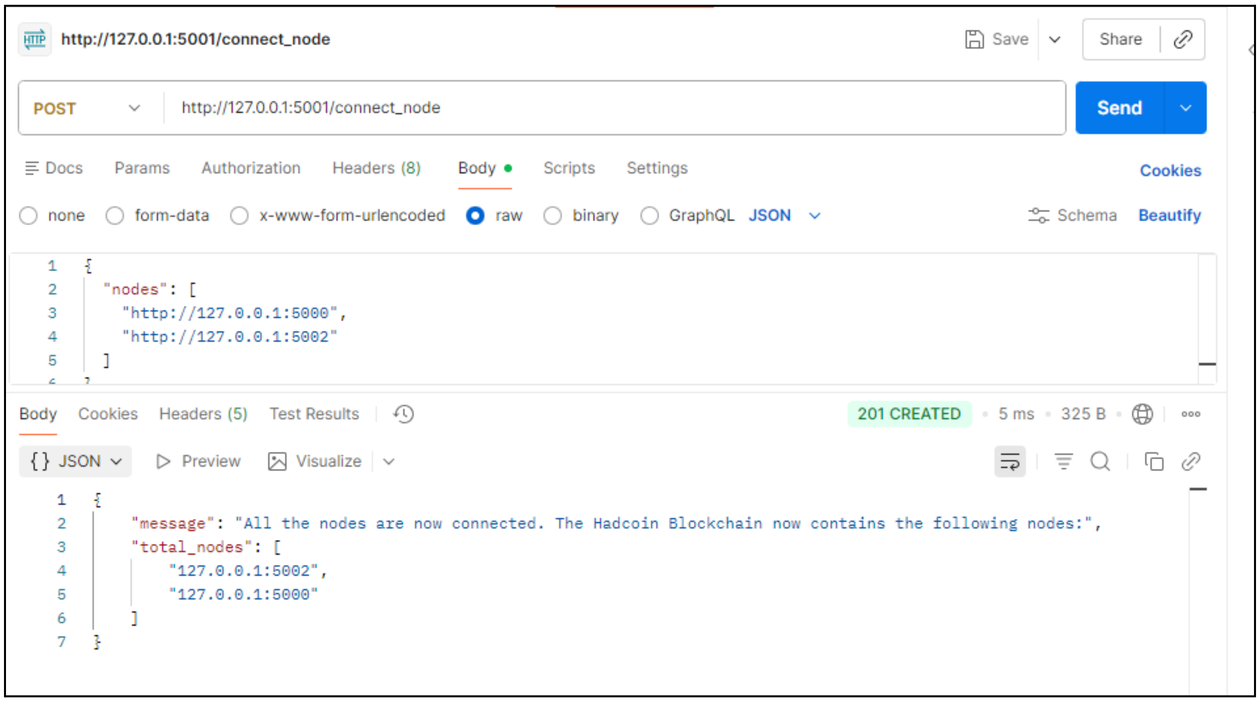
http://127.0.0.1:5001/get_chain

http://127.0.0.1:5001/get_chain

200 OK · 4 ms · 533 B

```
{
    "chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2026-01-27 11:09:25.029105",
            "transactions": []
        },
        {
            "index": 2,
            "previous_hash": "6d617bf1d8543b614960a1afcdbe1e1bb5f00e0e840c15c815761faffe79dec0",
            "proof": 533,
            "timestamp": "2026-01-27 11:12:17.003830",
```

4. node - invoke connect_node()



POST http://127.0.0.1:5001/connect_node

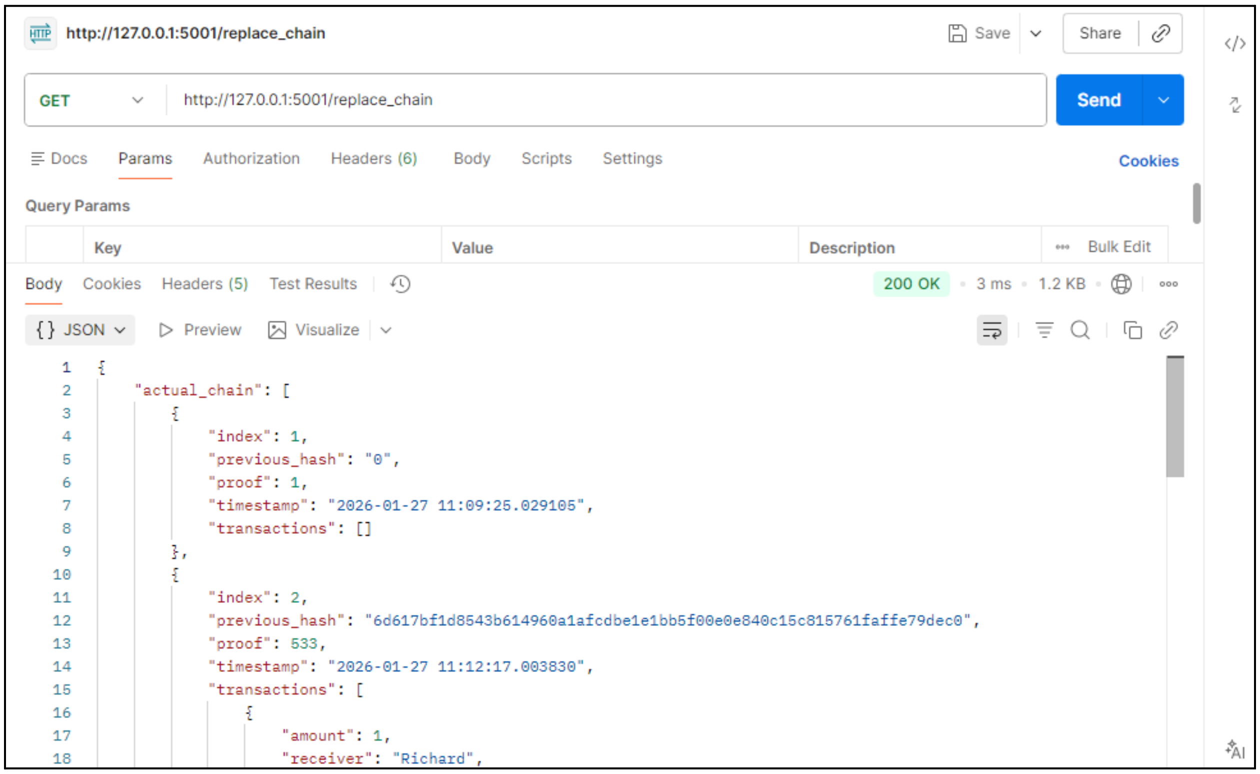
```
{
  "nodes": [
    "http://127.0.0.1:5000",
    "http://127.0.0.1:5002"
  ]
```

201 CREATED · 5 ms · 325 B

Body {} JSON

```
{
    "message": "All the nodes are now connected. The Hadcoin Blockchain now contains the following nodes:",
    "total_nodes": [
        "127.0.0.1:5002",
        "127.0.0.1:5000"
    ]
}
```

5. replace the longest chain - replace_chain()

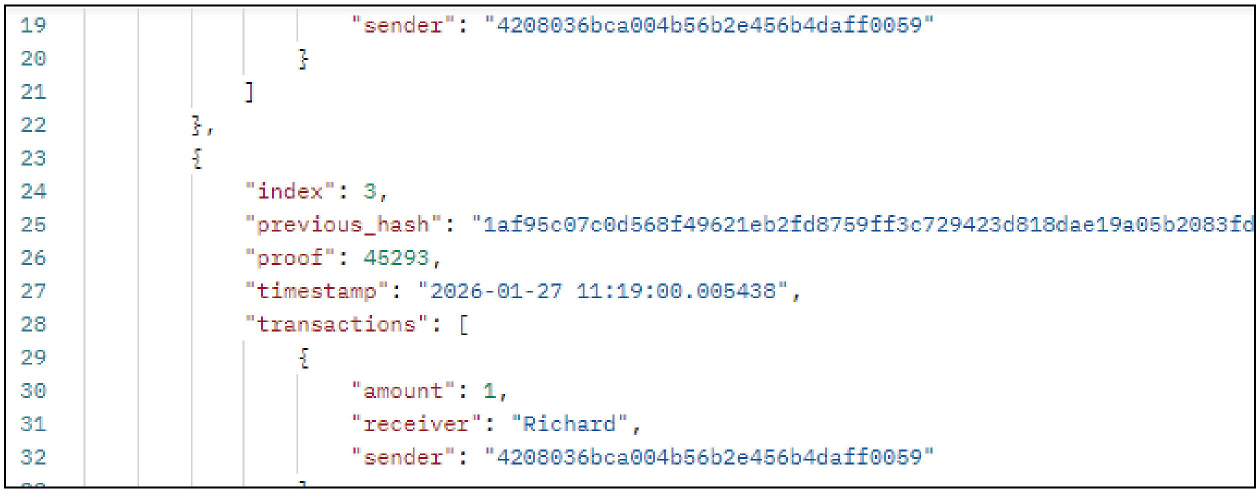


GET http://127.0.0.1:5001/replace_chain

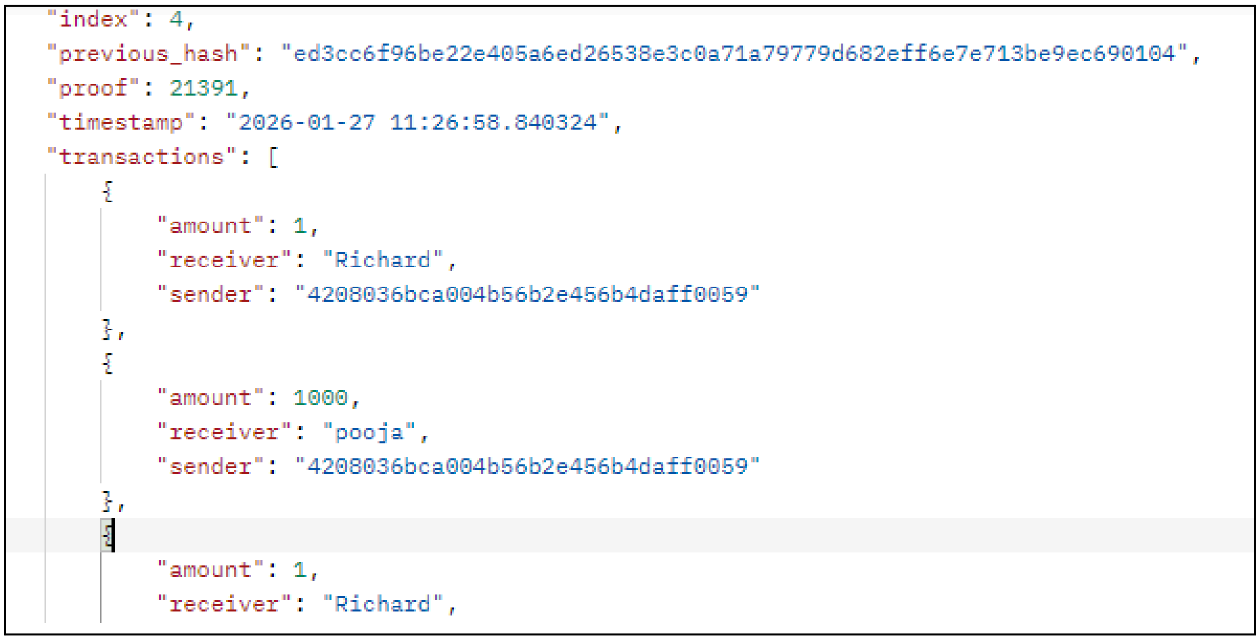
200 OK · 3 ms · 1.2 KB

Body {} JSON

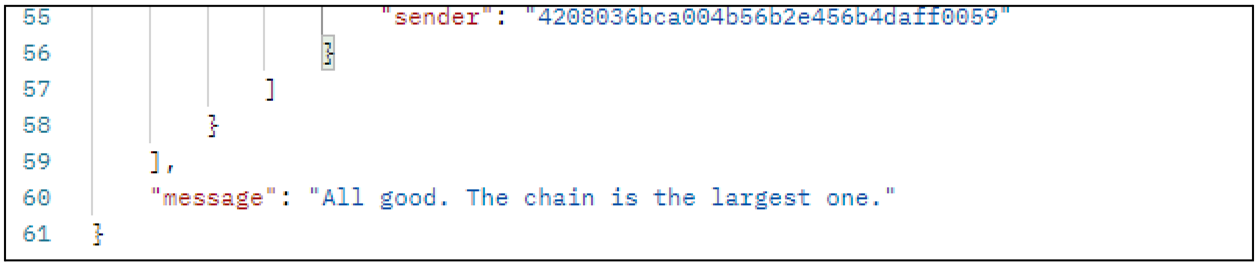
```
{
    "actual_chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2026-01-27 11:09:25.029105",
            "transactions": []
        },
        {
            "index": 2,
            "previous_hash": "6d617bf1d8543b614960a1afcdbe1e1bb5f00e0e840c15c815761faffe79dec0",
            "proof": 533,
            "timestamp": "2026-01-27 11:12:17.003830",
            "transactions": [
                {
                    "amount": 1,
                    "receiver": "Richard",
```



```
                    "sender": "4208036bca004b56b2e456b4daff0059"
                }
            ]
        },
        {
            "index": 3,
            "previous_hash": "1af95c07c0d568f49621eb2fd8759ff3c729423d818dae19a05b2083fd
            "proof": 45293,
            "timestamp": "2026-01-27 11:19:00.005438",
            "transactions": [
                {
                    "amount": 1,
                    "receiver": "Richard",
                    "sender": "4208036bca004b56b2e456b4daff0059"
```



```
"index": 4,
"previous_hash": "ed3cc6f96be22e405a6ed26538e3c0a71a79779d682eff6e7e713be9ec690104",
"proof": 21391,
"timestamp": "2026-01-27 11:26:58.840324",
"transactions": [
    {
        "amount": 1,
        "receiver": "Richard",
        "sender": "4208036bca004b56b2e456b4daff0059"
    },
    {
        "amount": 1000,
        "receiver": "pooja",
        "sender": "4208036bca004b56b2e456b4daff0059"
    },
    {
        "amount": 1,
        "receiver": "Richard",
```



```
                    "sender": "4208036bca004b56b2e456b4daff0059"
                }
            ]
        }
    ],
    "message": "All good. The chain is the largest one."
}
```

Conclusion: This experiment demonstrates the practical application of blockchain principles through the implementation of a simulated cryptocurrency. By modeling transactions, mining, and consensus across multiple nodes, it provides clear insight into how decentralized blockchain systems function. The experiment emphasizes the role of security, consensus mechanisms, and decentralization in maintaining a reliable and distributed ledger.