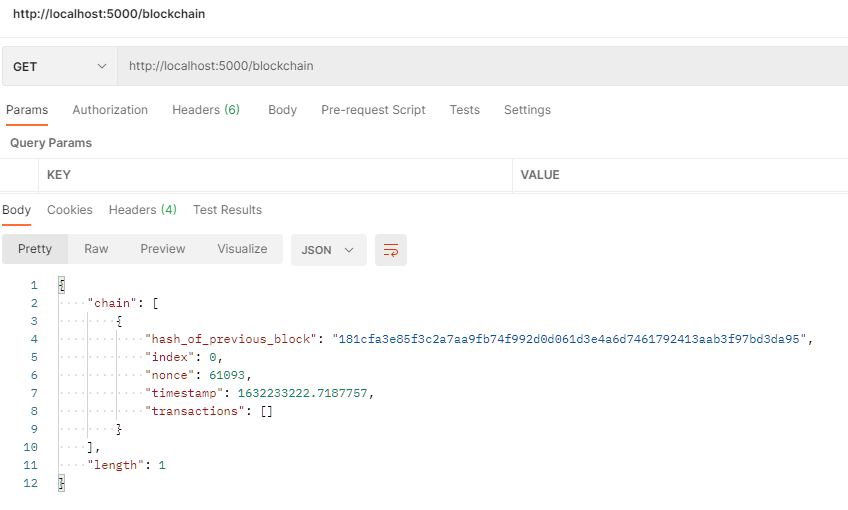
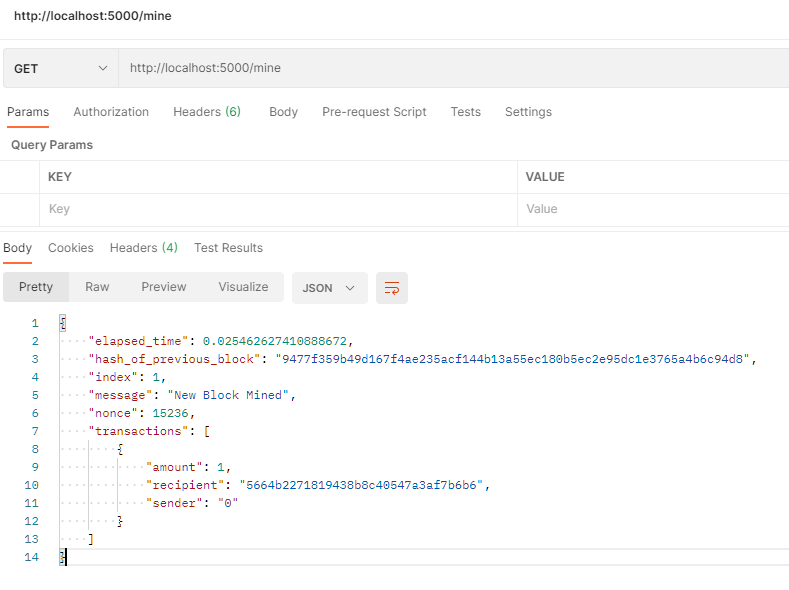
**GET request to observe blockchain.**



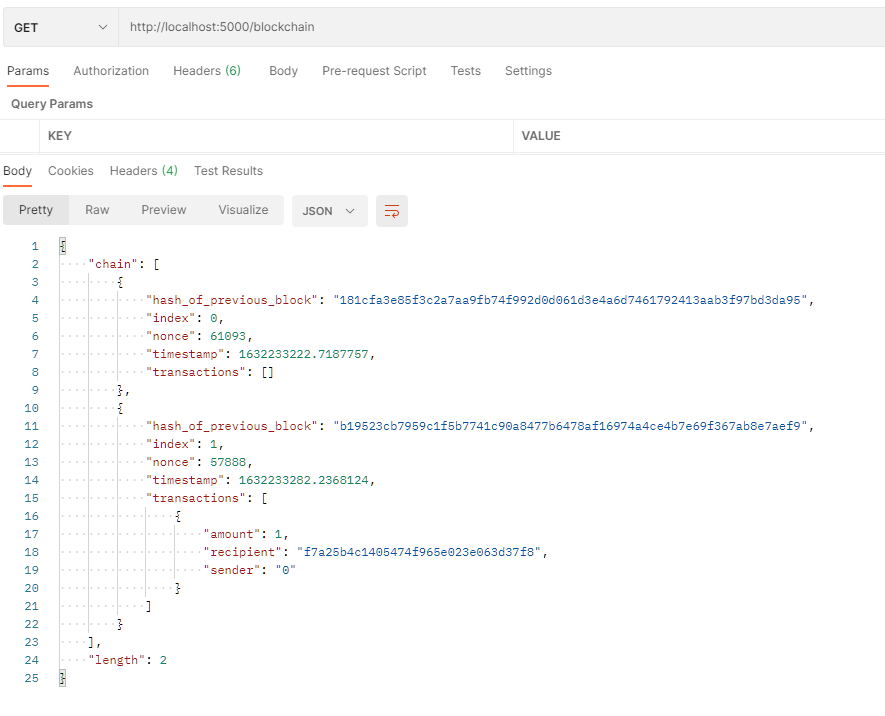
**Mining the Blockchain.**



**What do you observe? Describe your observations.**

Observed that there is a transaction from “sender” = 0 with tha value of 1 for 5664b2271819438b8c40547a3af7b6b6, and this is the block reward given to the miner who successfully mined the block. The nonce that matches the required difficulty level is 15236, and the index increased by 1.

**You now have a new block mined on the blockchain.Take a look at your blockchain**



The new block is appended to the full chain with index 1.There is a hash\_of\_previous\_block metadata which is the hash generated from concatenating hash\_of\_previous\_block, index, nonce and transactions of the previous block.

**Difficulty is set to 4 zeroes now. Implement a timing function on mining. Now change the difficulty to 5,6, 8,10 zeroes and observe time taken to mine a block. Describe your observations and the reason for it.**

4 zeroes



5 zeroes



6 zeroes



8 zeroes

Block unsuccessfully mined after 15 minutes.

10 zeroes

Block unsuccessfully mined after 15 minutes.

The observation is that it becomes exponentially longer for the content\_hash, which is generated by passing the index, previous block hash, transactions and nonce through a sha256 hashing function, to match a longer length of zeroes for the difficulty target.

**Transactions:**

Add a transaction to a block. Transfer 0.5 coin from the recipient of the reward in Section 3.

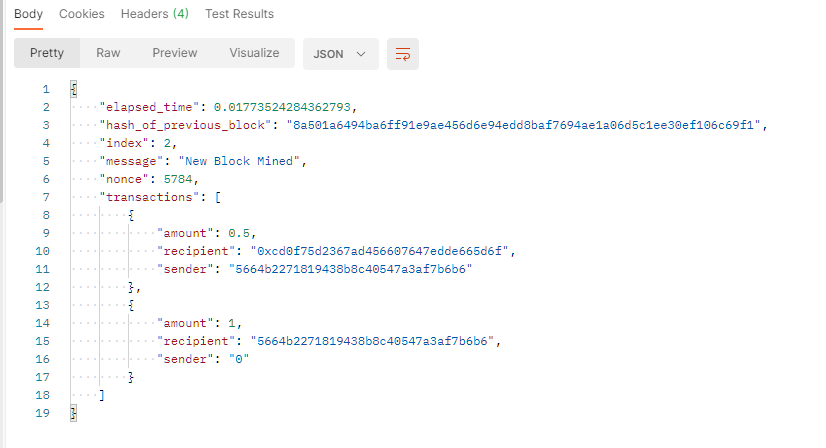
Sender address: 0x5664b2271819438b8c40547a3af7b6b6

Receiver address: 0xcd0f75d2367ad456607647edde665d6f

*curl -X POST -H 'Content-Type:application/json' -d '{"sender":"5664b2271819438b8c40547a3af7b6b6", "recipient":"cd0f75d2367ad456607647edde665d6f","amount":0.5}' http://localhost:5000/transactions/new*

[](http://localhost:5000/transactions/new%22*)

Mining block 2 using Postman.



Appended to full blockchain.



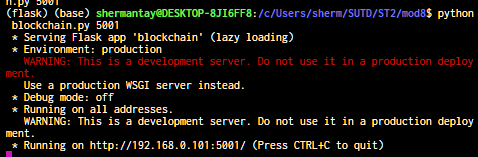
**Block 2 has now been mined and it contains two transactions – describe both of them and their meaning.**

The first transaction is the transaction we have made, which was for 0.5 tokens from 0x5664b2271819438b8c40547a3af7b6b6 to 0xcd0f75d2367ad456607647edde665d6f.

The second transaction is the block reward, which is the reward of 1 token for 0x5664b2271819438b8c40547a3af7b6b6, the miner who successfully mined the block.

Instantiate a new node running at port 5001

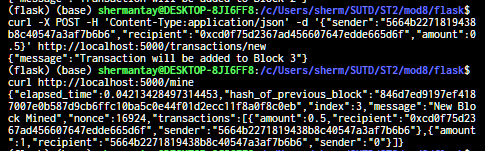
*Python3 blockchain.py 5001*



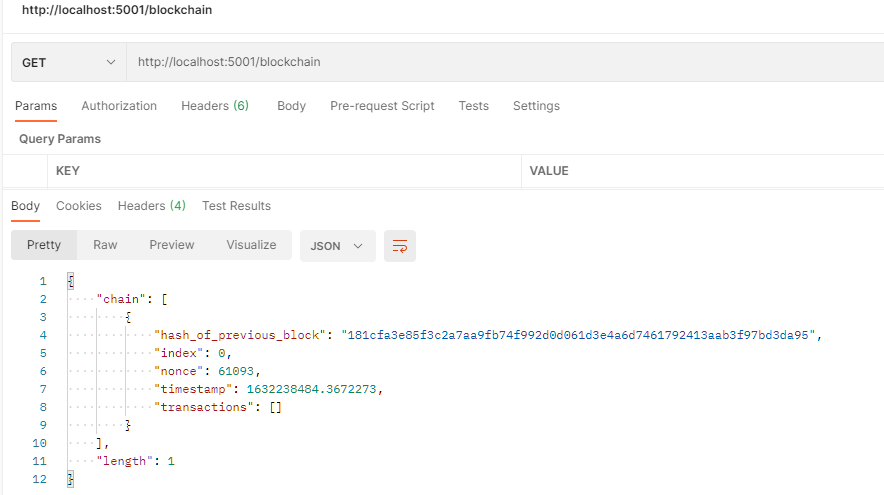
Add a new transaction (to be added to block 3) and mine it.

*curl -X POST -H 'Content-Type:application/json' -d '{"sender":"5664b2271819438b8c40547a3af7b6b6", "recipient":"cd0f75d2367ad456607647edde665d6f","amount":0.5}'* [*http://localhost:5000/transactions/new*](http://localhost:5000/transactions/new)

*Curl http://localhost:5000/mine*



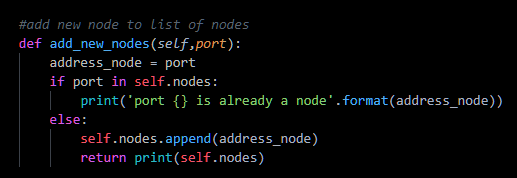
**Add a new transaction to the blockchain running at port 5000. Mine a new block. What do you observe at blockchain running on port 5001?**



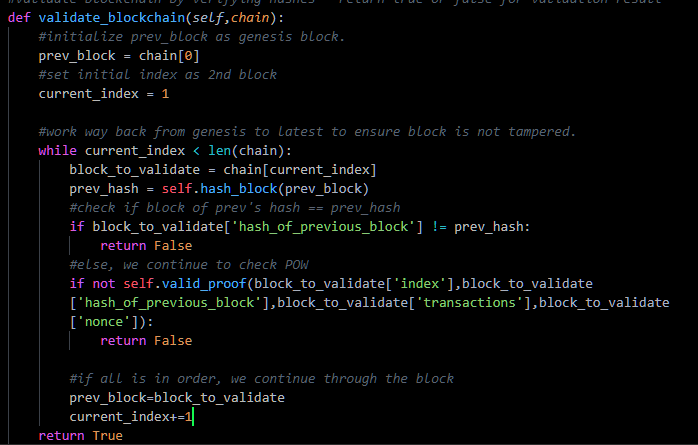
On blockchain running on port 5001, the block length is only 1, while the block length running on port 5000 is of length 3. This is because both blocks are not synchronized, and require a method to apply consensus.

**You will need to write functions to add new nodes, validate the blockchain by verifying hashes and synchronize nodes by updating blockchain on neighbour nodes.**

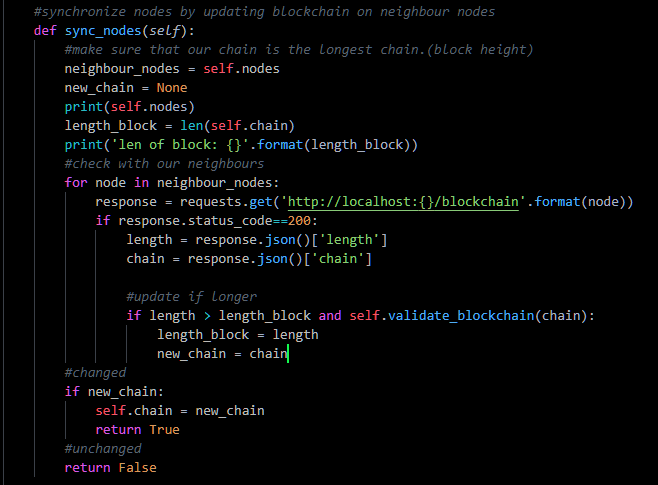
**Functions**

****

This function takes in new ports and appends it to the list initialized at the start, called nodes.

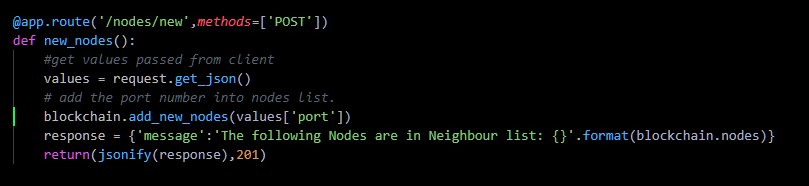
****

This function validates that a blockchain has not been tampered with. This is hashing the genesis block (prev hash) and comparing it to the one on the block being validated. Once that passes, we can validate the hash. This continues until we reach the latest block of the blockchain.

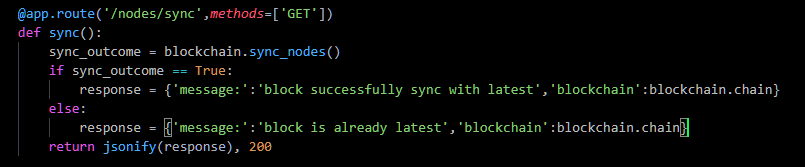


This function runs through the nodes list and updates the chain with the longest valid blockchain within the list.

**Routes**

Route to add new node to list

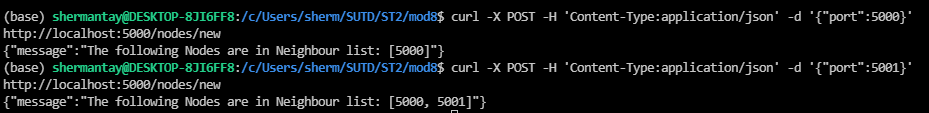
Route to sync nodes



**Output**

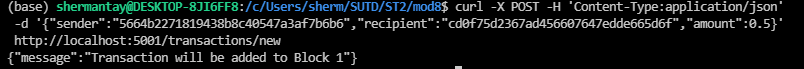
Add nodes to node list:

*curl -X POST -H 'Content-Type:application/json' -d '{"port":5001}' http://http://localhost:5000/nodes/new*



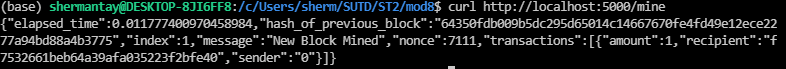
Add transaction on localhost:5000.

*curl -X POST -H 'Content-Type:application/json' -d '{"sender":"5664b2271819438b8c40547a3af7b6b6", "recipient":"cd0f75d2367ad456607647edde665d6f","amount":0.5}' http://localhost:5001/transactions/new*

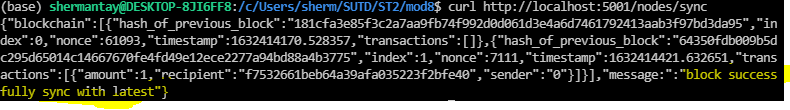


Mine transaction on localhost:5000

*curl http://localhost:5000/mine*



Synchronize localhost:5001 with blockchain



Observed output on Postman



We can see that the block as observed from localhost:5001 shows the most updated blockchain, where the latest block that was mined by localhost:5000 is being shown.