**Blockchain Lab Experiment 2**

**Shashwat Tripathi**

**D20A Roll No: 64**

**Aim:** Create a blockchain using python

**Practical:**

**Step - 1 : Construct Merkle Tree** [Shashwat 64 Blockchain Lab2 Step1](https://colab.research.google.com/drive/1cYdV_J0RnpkLpDb9D91ZY9cIliuvDbDH?usp=sharing)

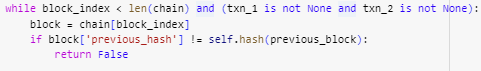
**Step - 2 : Run a Blockchain with one node** [Shashwat 64 Blockchain Lab2 Step2](https://colab.research.google.com/drive/1bwfq01JhifRA8LDS4x3Aw3j2PyODISiG?usp=sharing)

1. Make a copy of this Google Colab Notebook
2. Try to solve the errors in given Program
3. After successful execution of the Program in Colab Notebook.

Add a method, create\_Transactions



Mine the block only when the transaction list is not null.



Remove the transactions from the list of transactions before mining.

(The transactions are removed from the transaction pool list before mining).



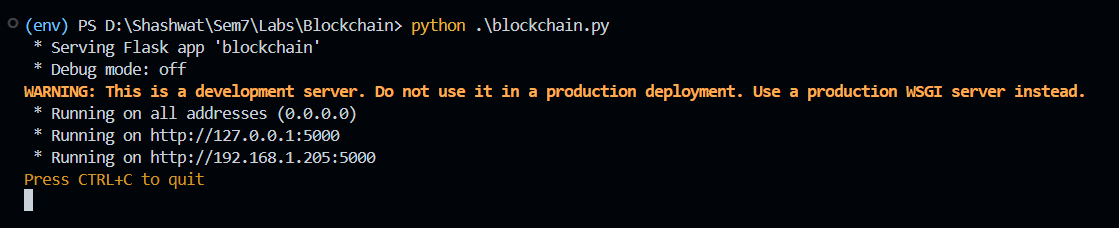
Modify the method, proof\_of\_work() to search for the golden nonce(new\_proof variable in the below code)

Cryptographic Puzzle is to have “000” leading zeros in the Block Hash



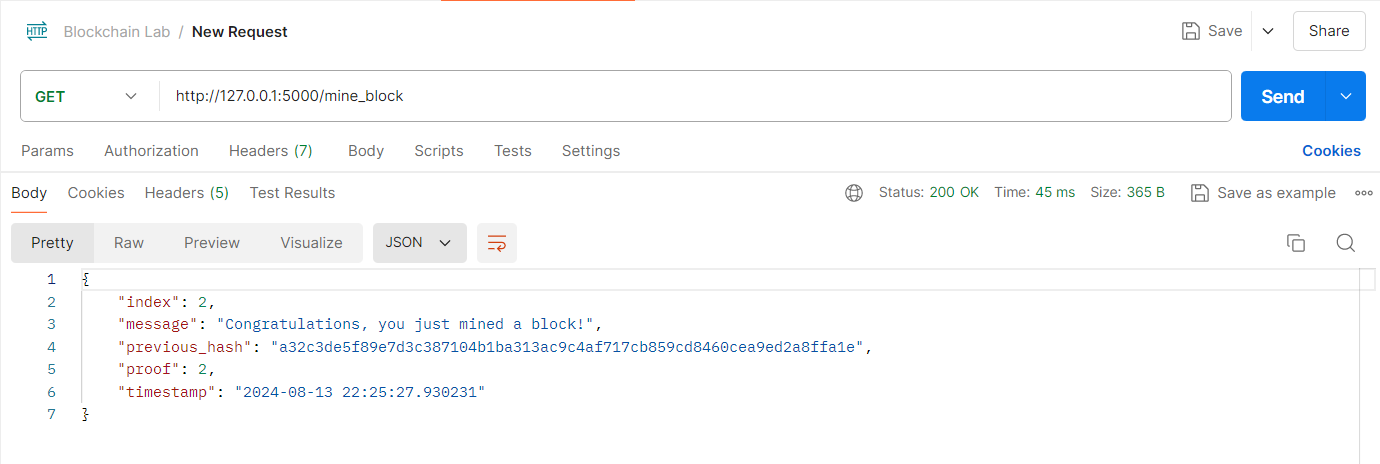
1. Download the code - blockchain.py
2. Update the code to incorporate the changes in step 3 to the code in step 4.
3. Follow the steps in Manual to demonstrate the working of Blockchain using Flask and Postman:

* Execute the blockchain.py in virtual environment

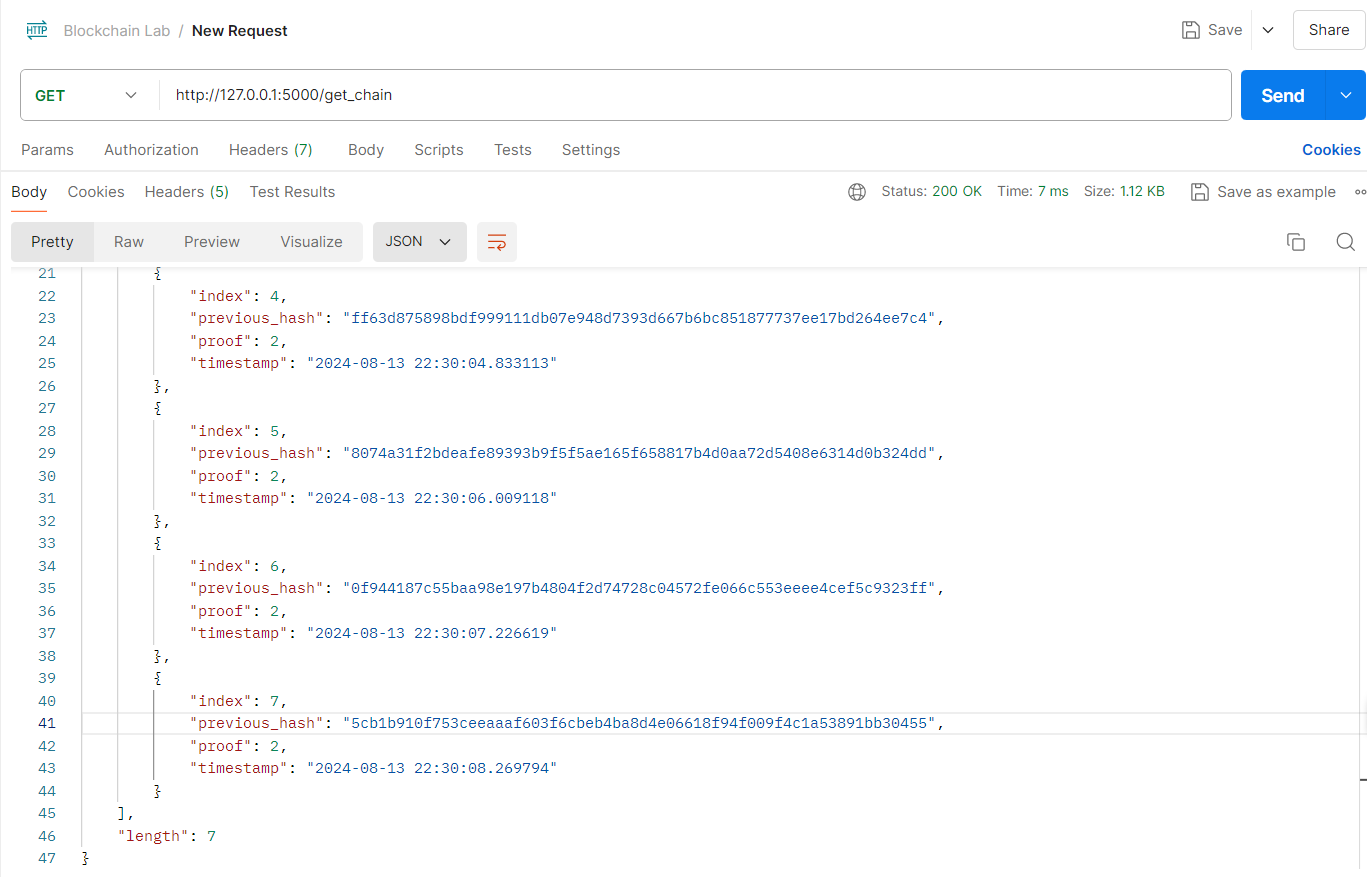


* Open the postman to run the following commands

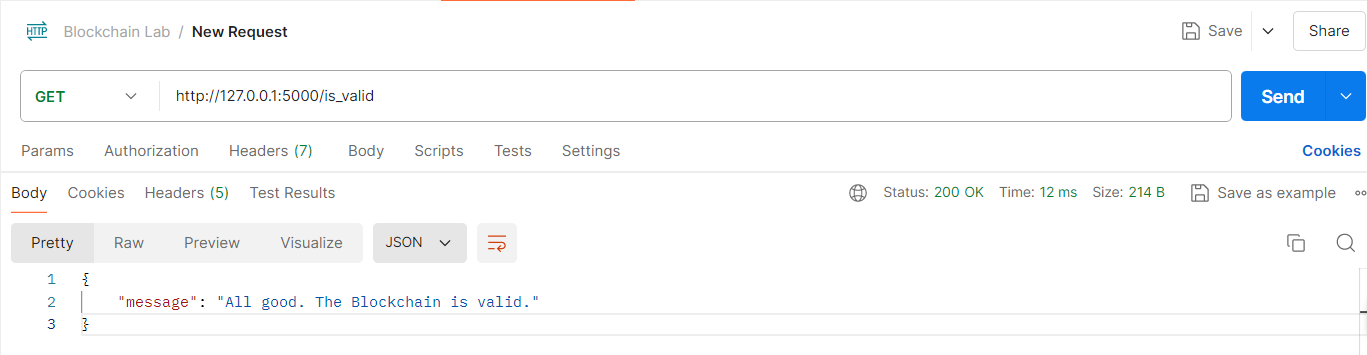
1. Mine the blocks



1. Get the chain



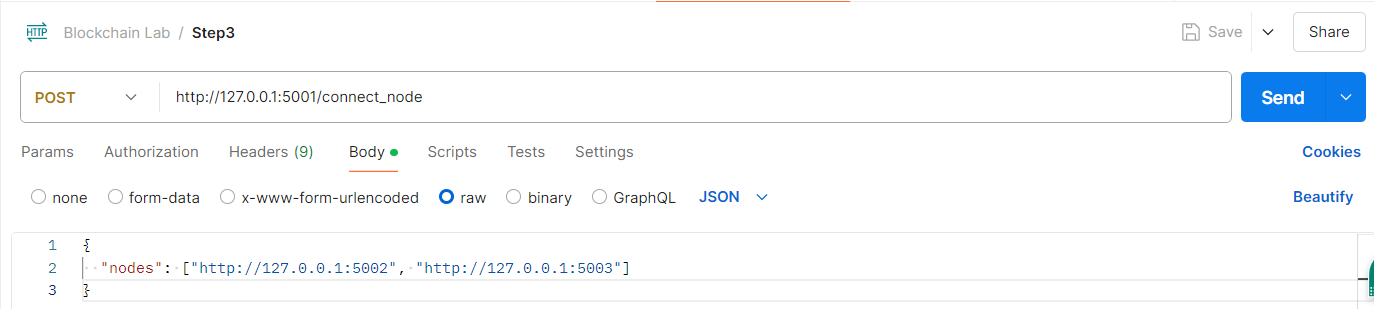
1. Check whether blockchain is valid

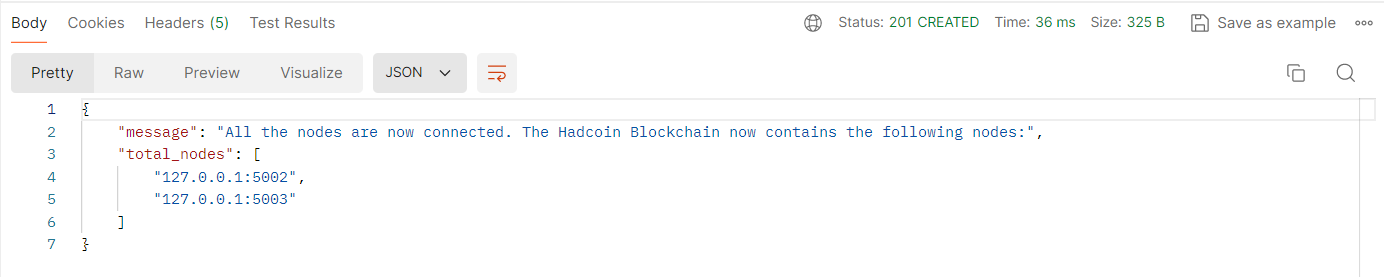


**Step - 3 : Run a Blockchain with 3 peer nodes**

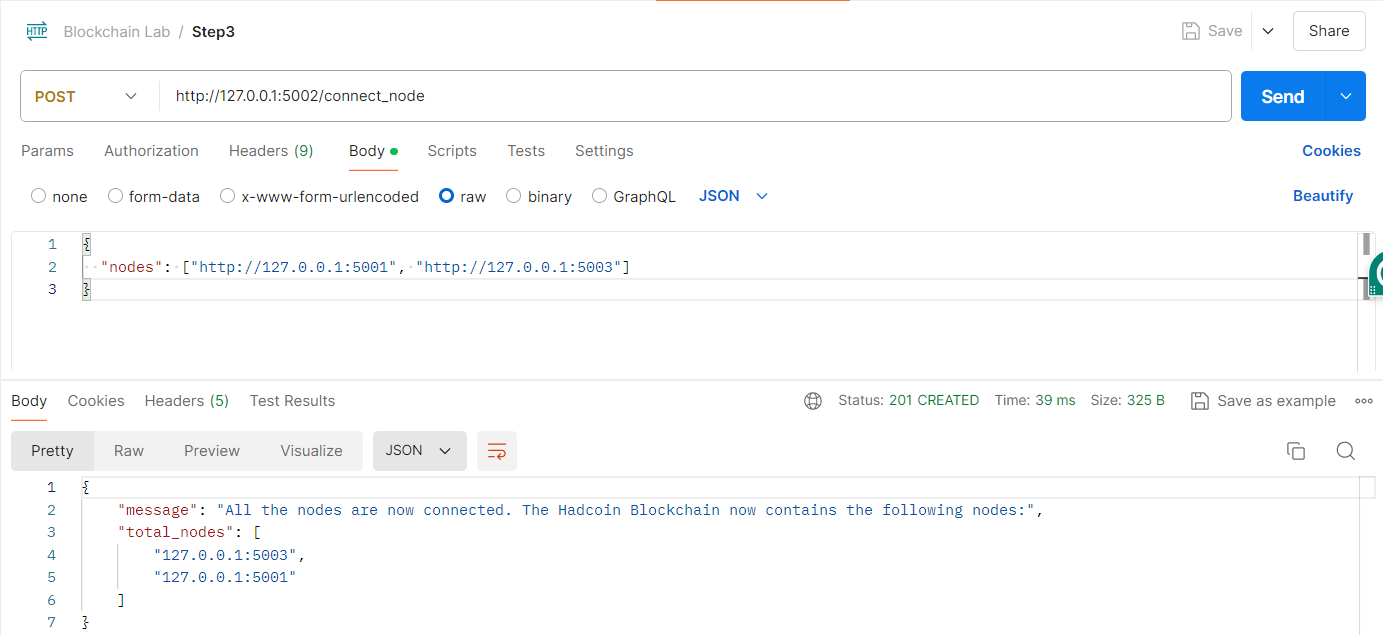
1. Download the code from folder, Lab\_2\_Step\_2
2. Install requests in the virtual environment created in Step 1. (Follow the instructions)
3. Run the files - hadcoin\_node\_5001.py, hadcoin\_node\_5002.py, hadcoin\_node\_5003.py in 3 different terminals.
4. Open Postman, from each node - invoke connect\_node() and pass the peers as POST requests.

Connecting nodes 5002 & 5003 to 5001

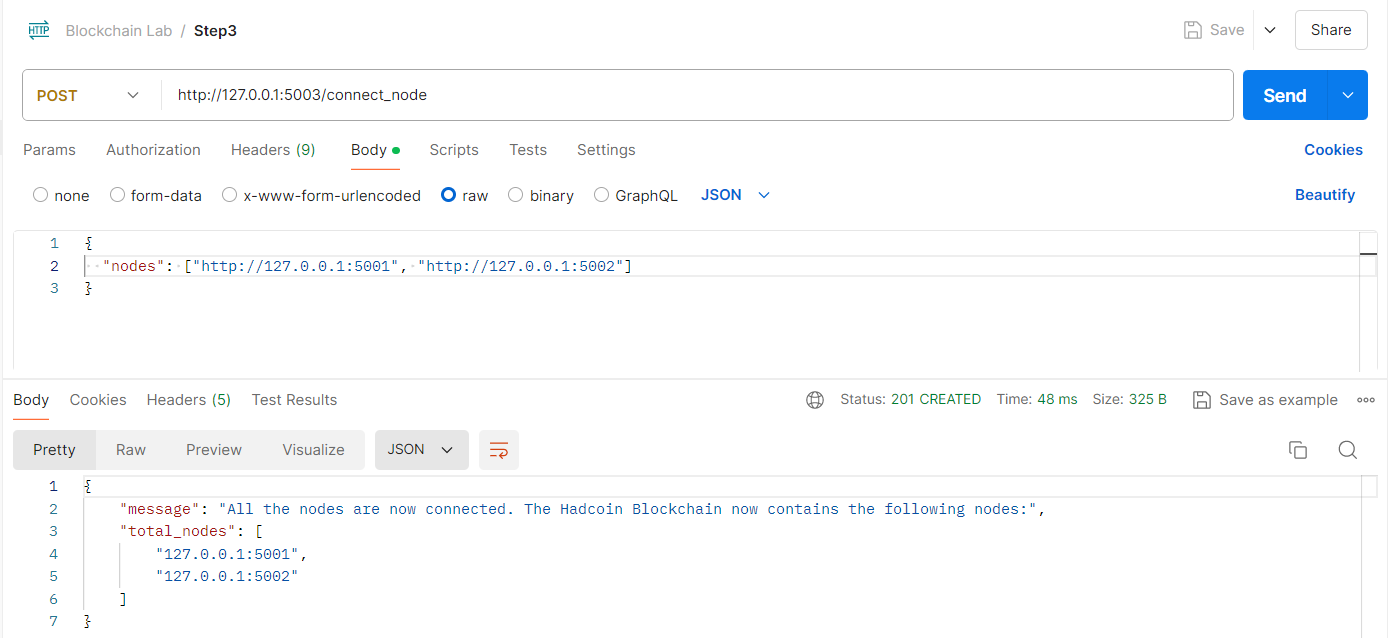




Connecting 5002 to 5001 & 5003

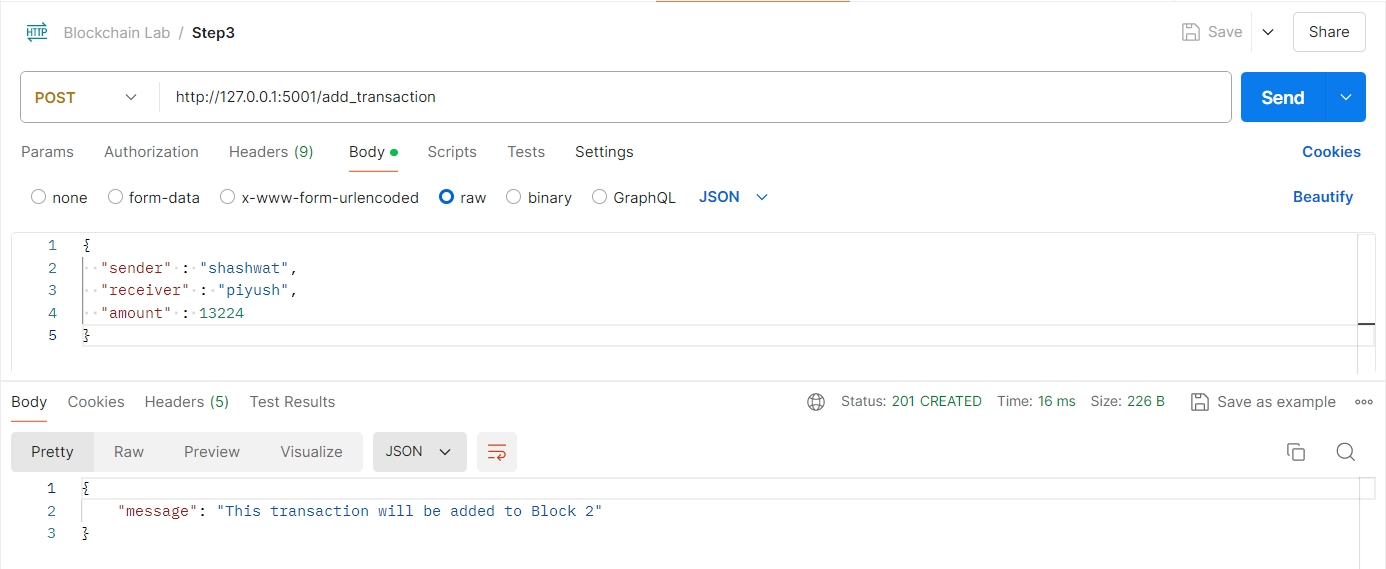


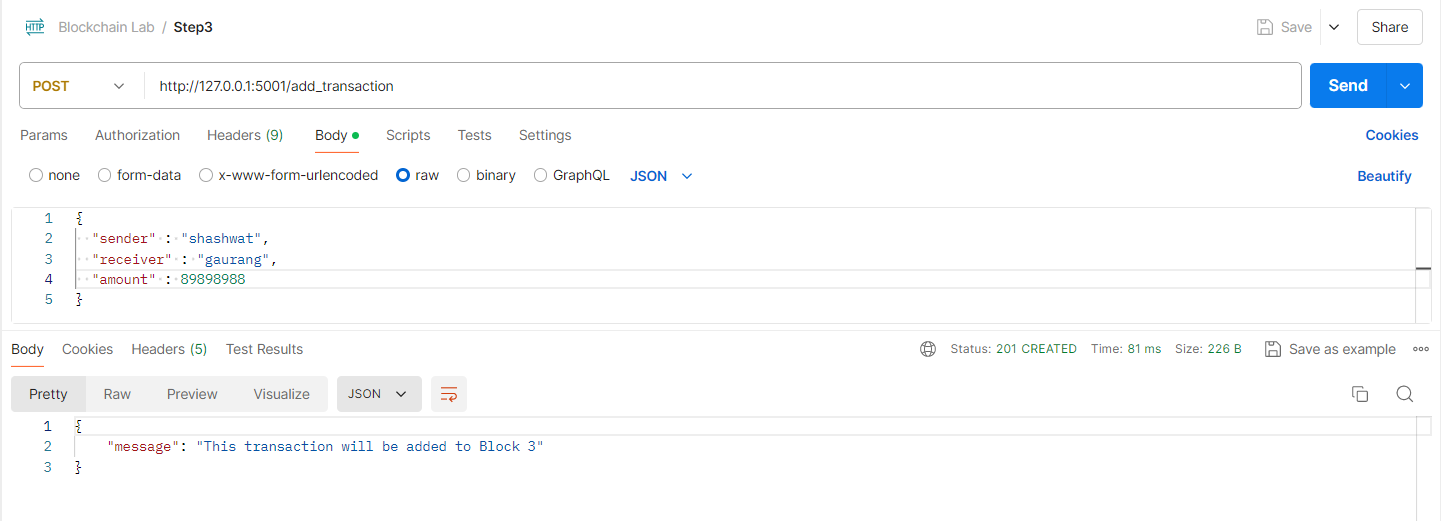
Connecting 5003 to 5001 & 5002

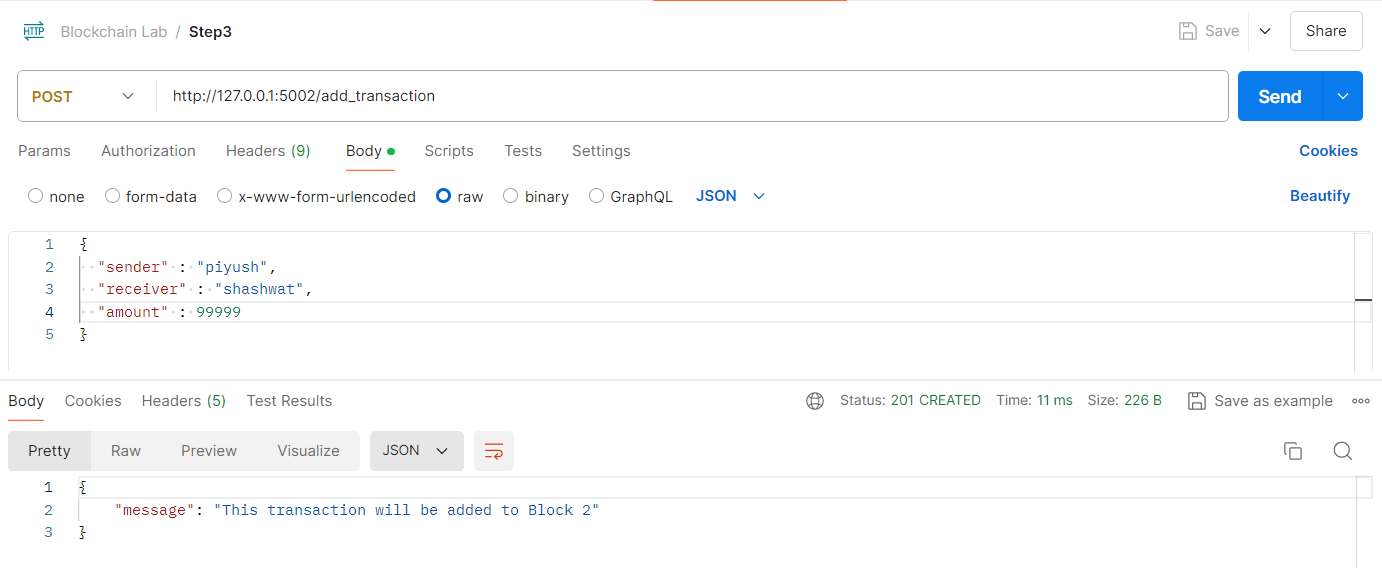


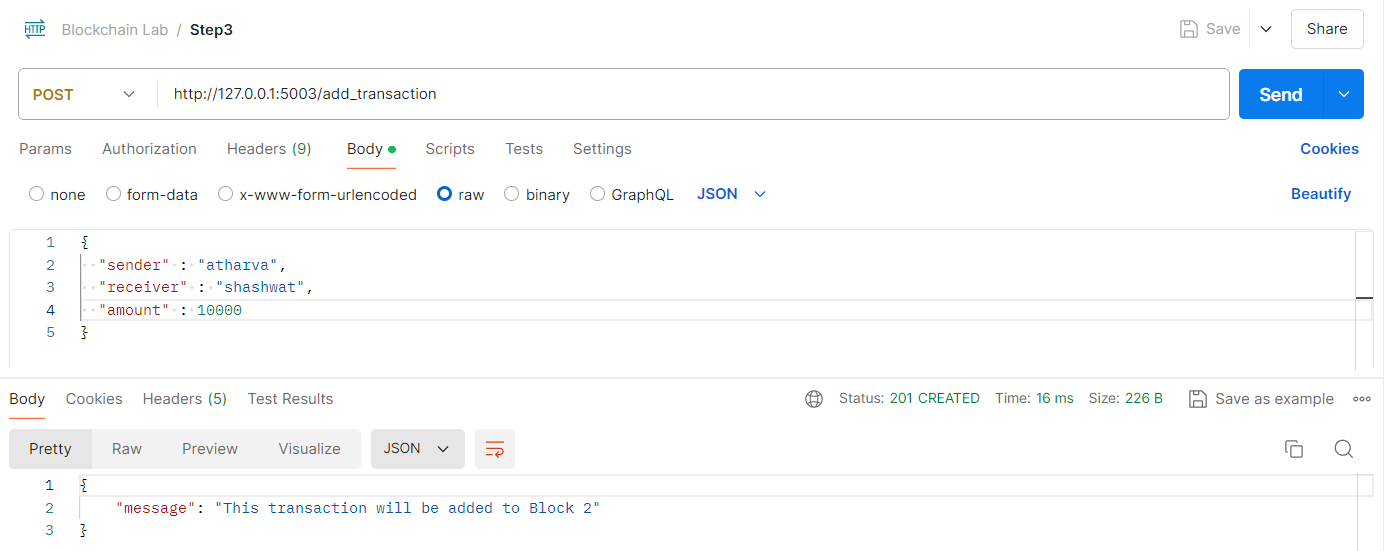
1. Perform the following functions

Add Transactions - invoke add\_transactions() as a POST request.

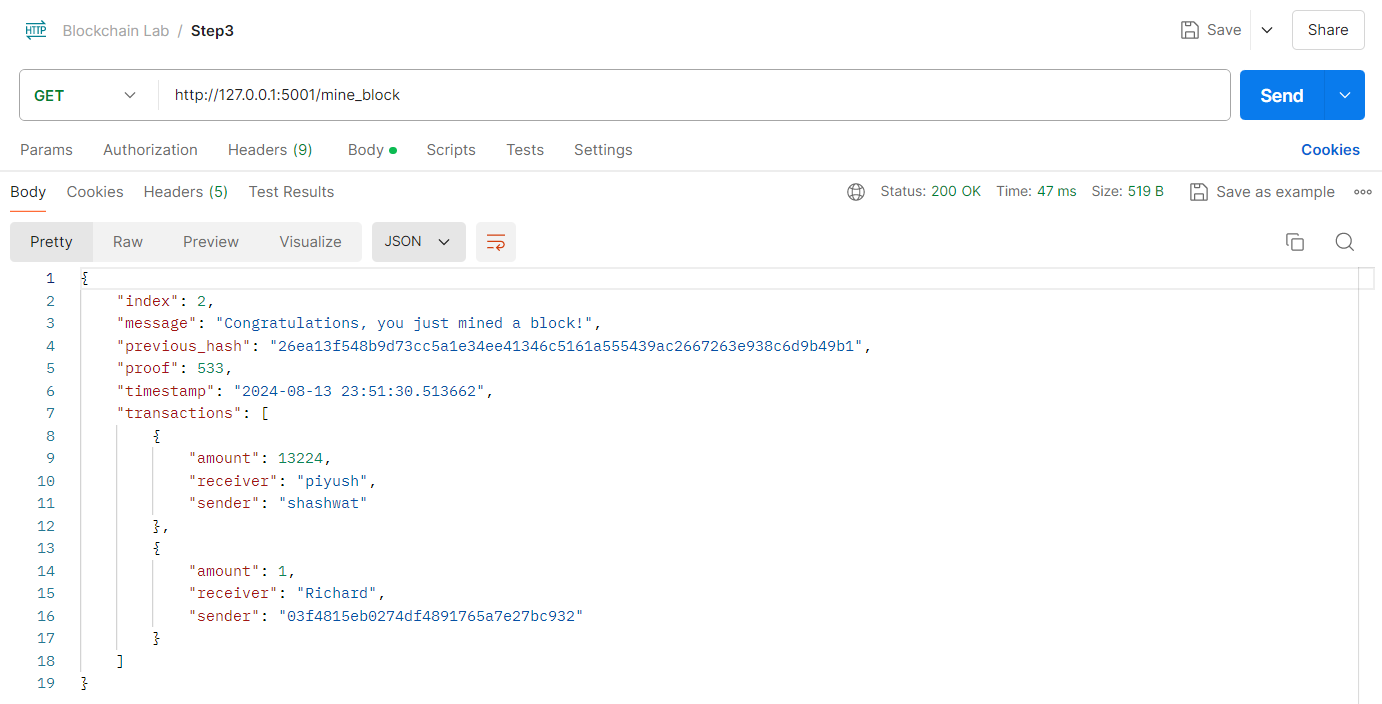


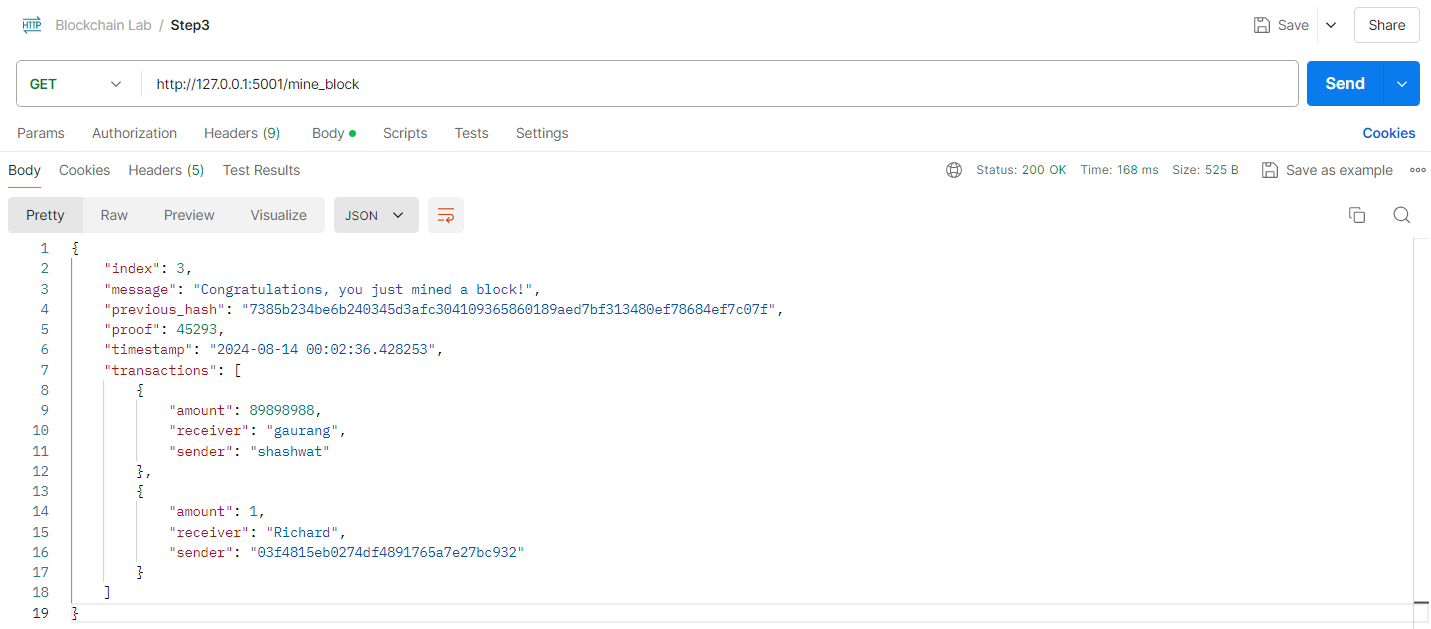


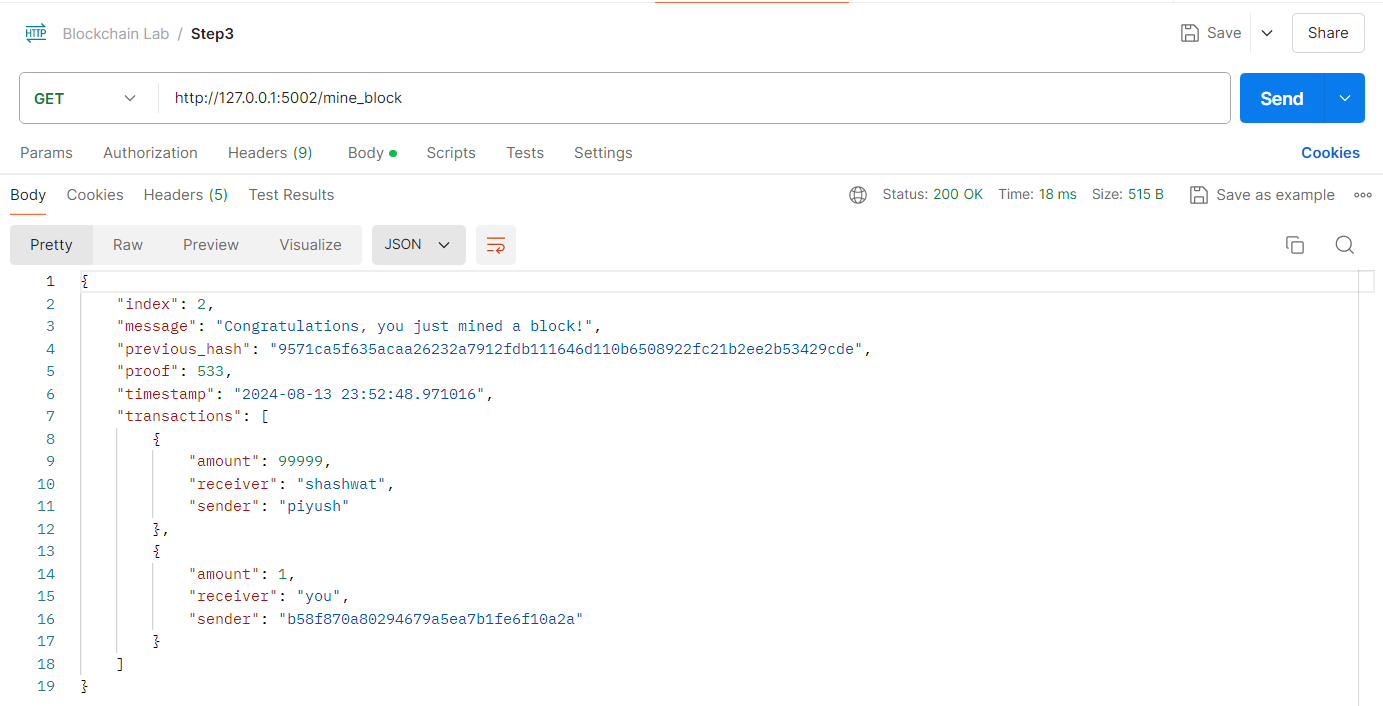


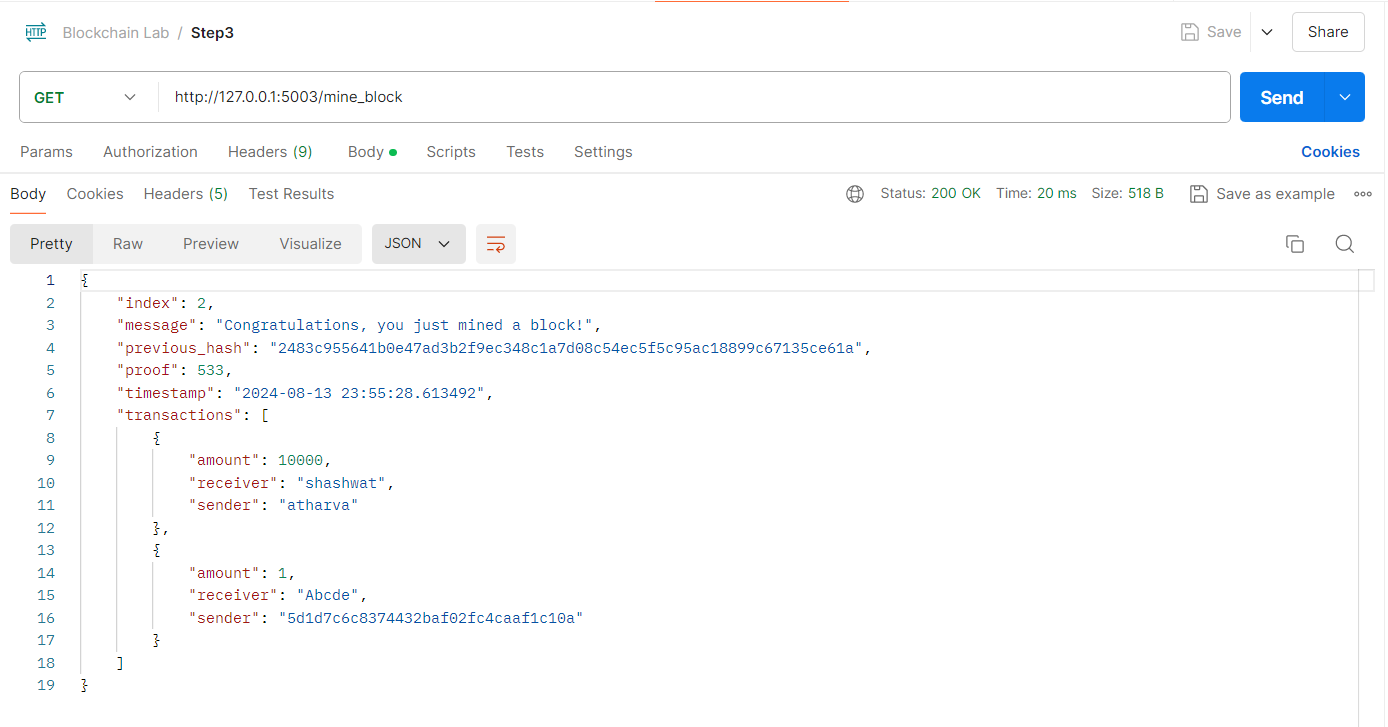


Mining - mine\_block()

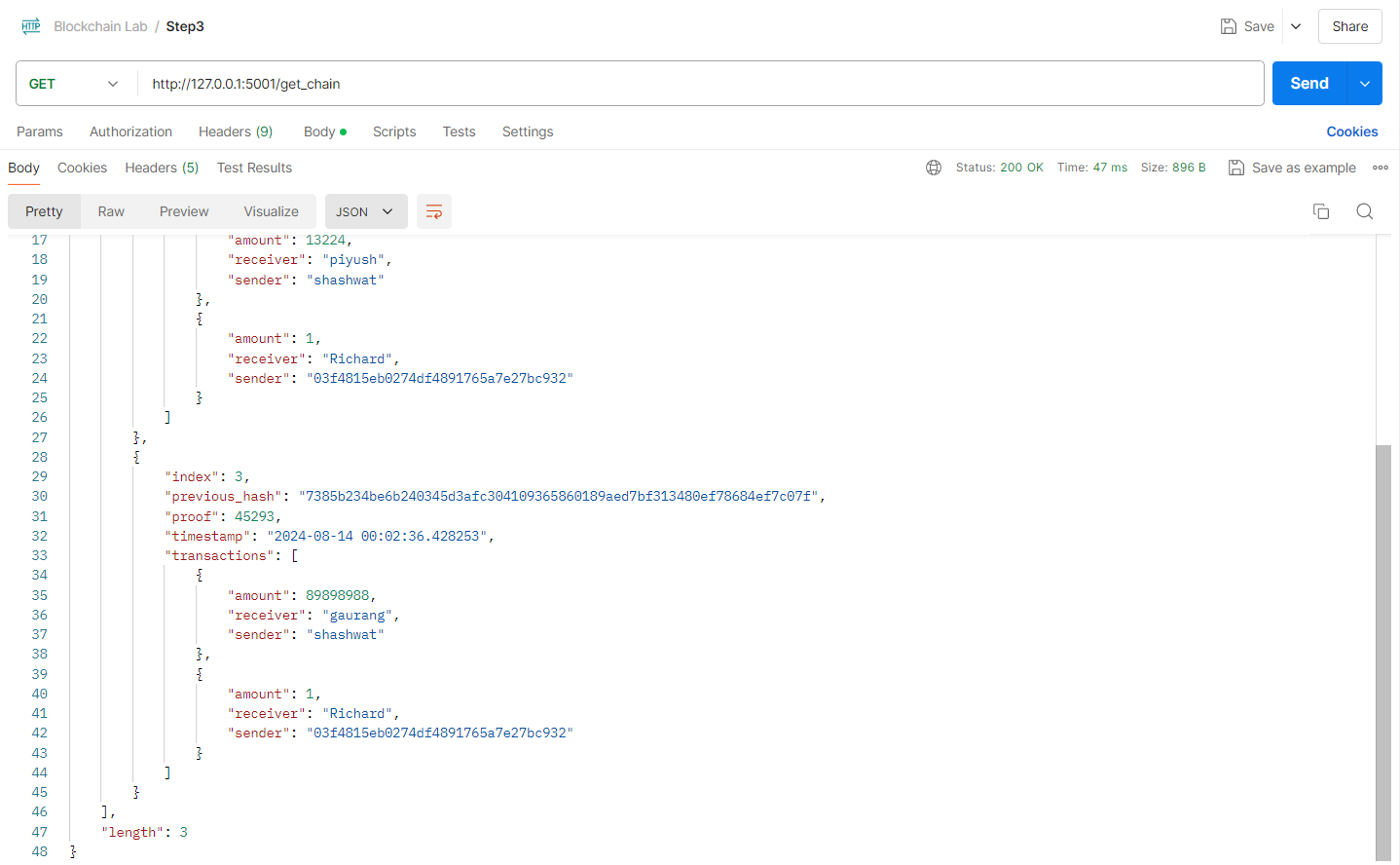


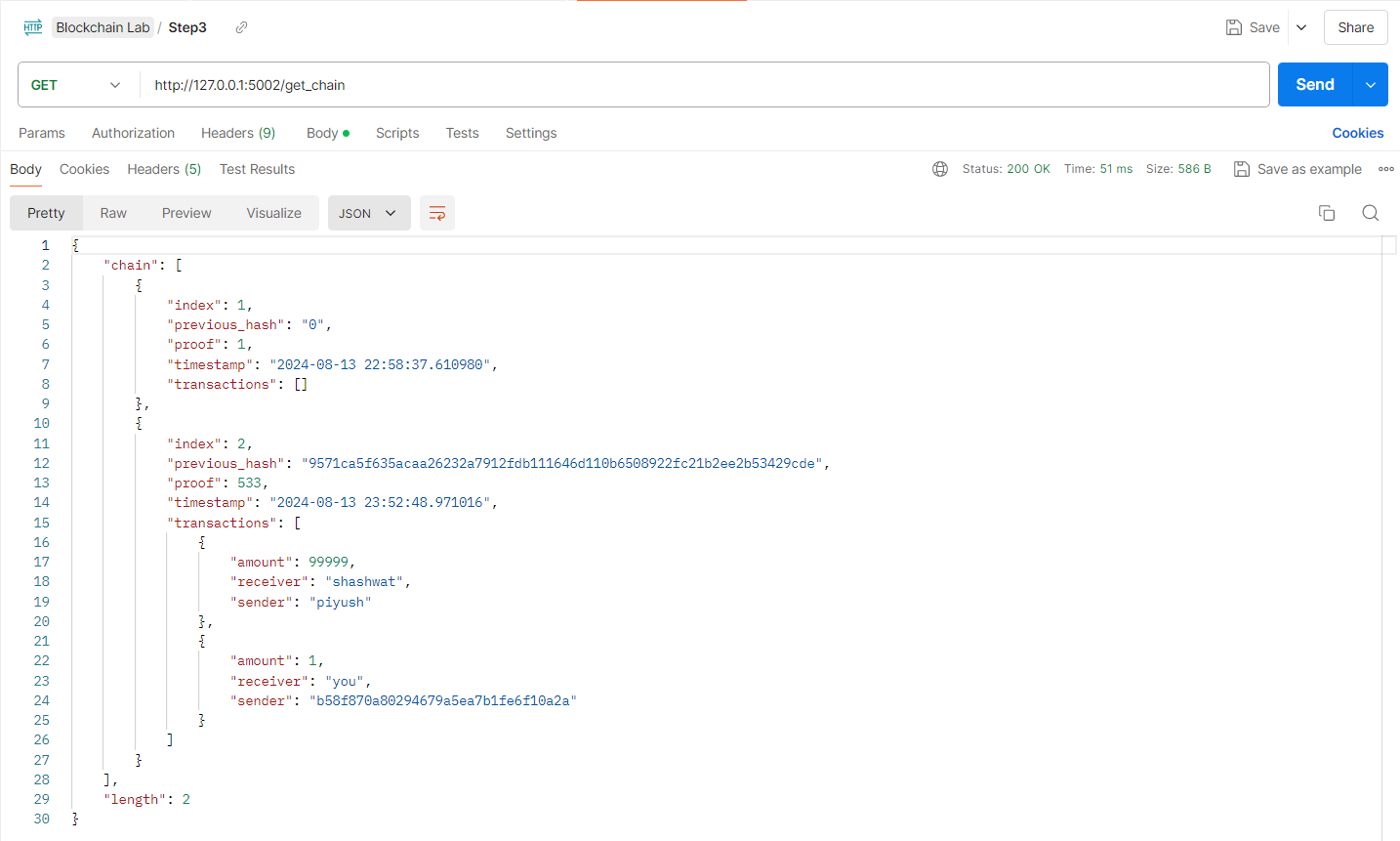


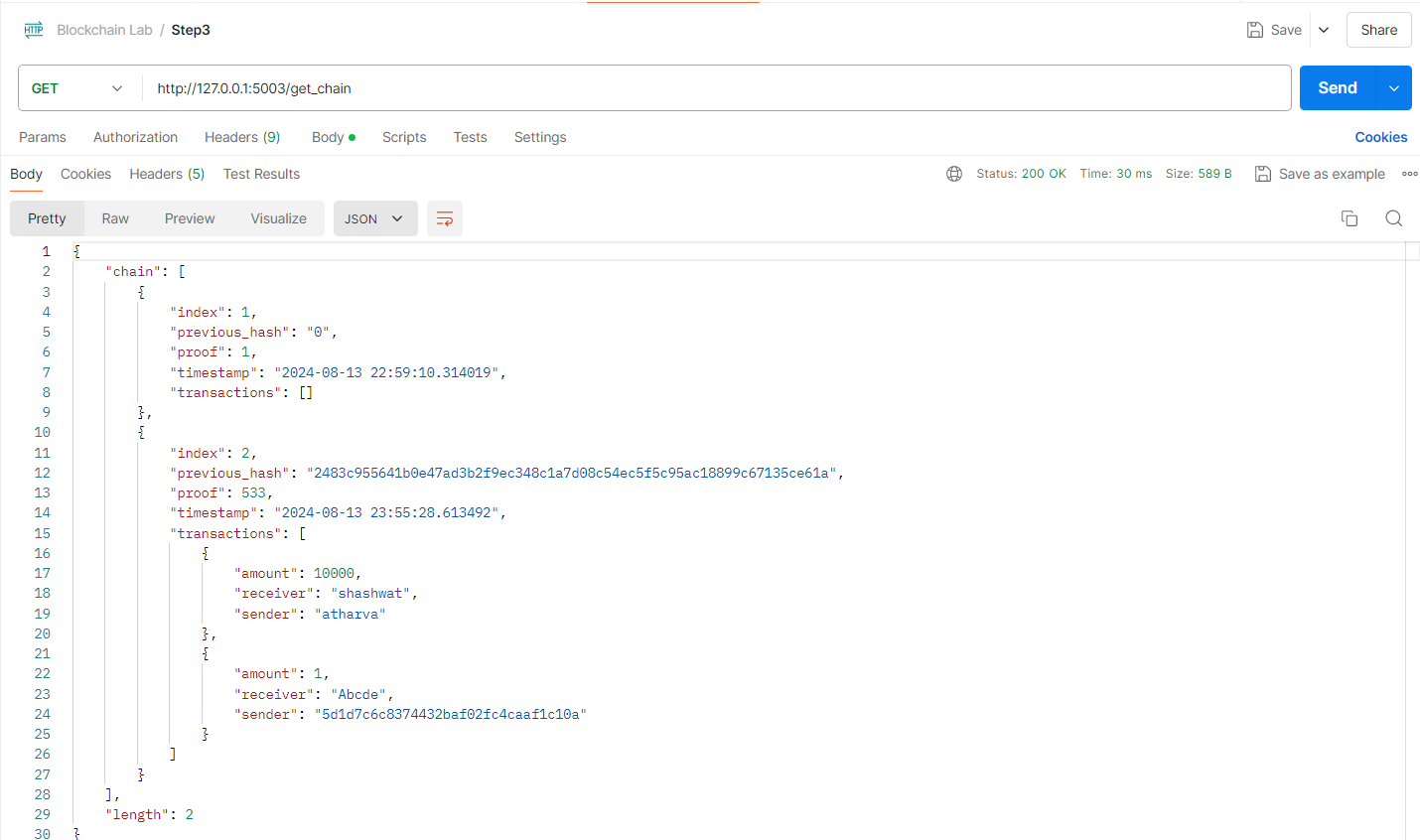




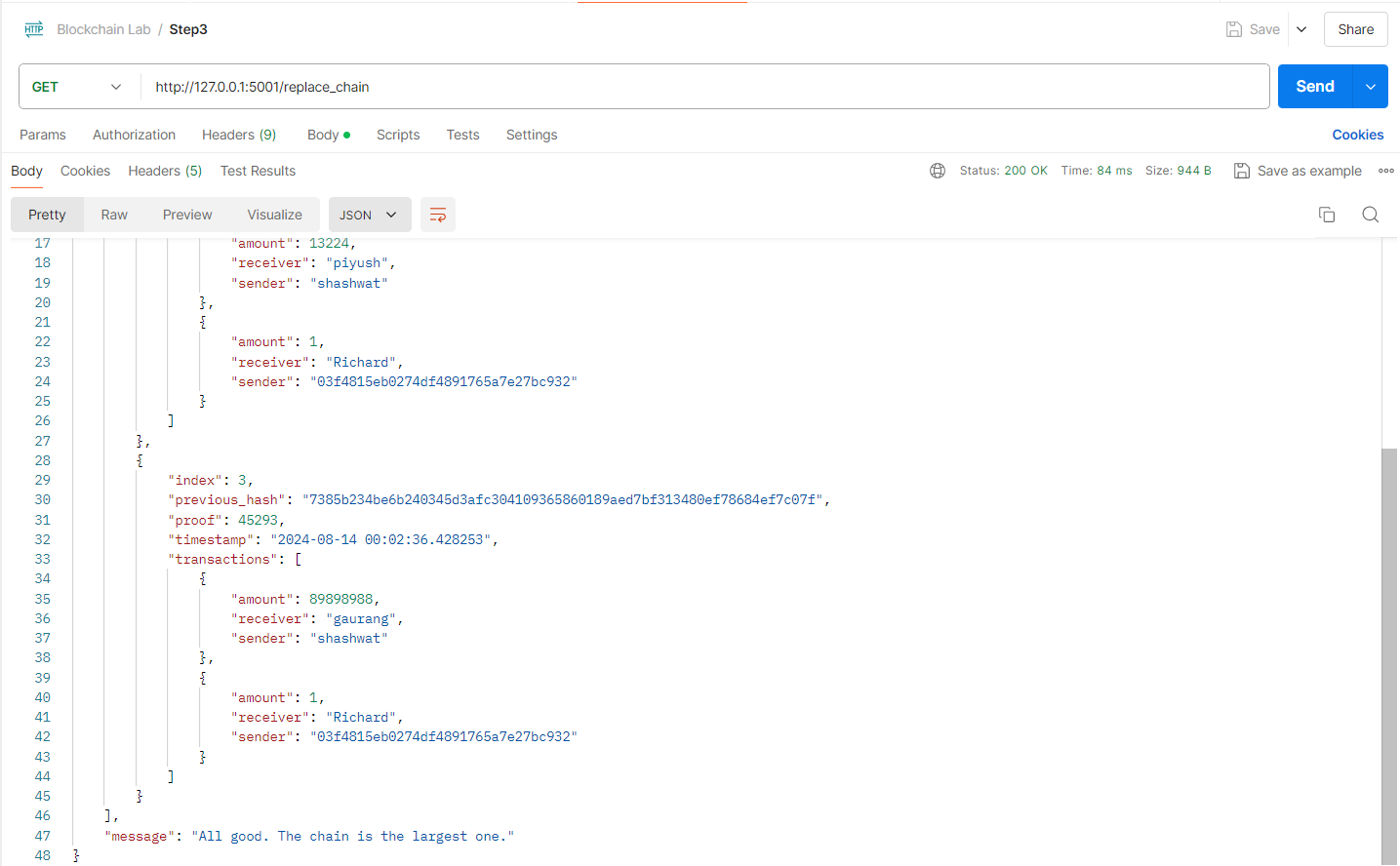
Fetch the chain - get\_chain()







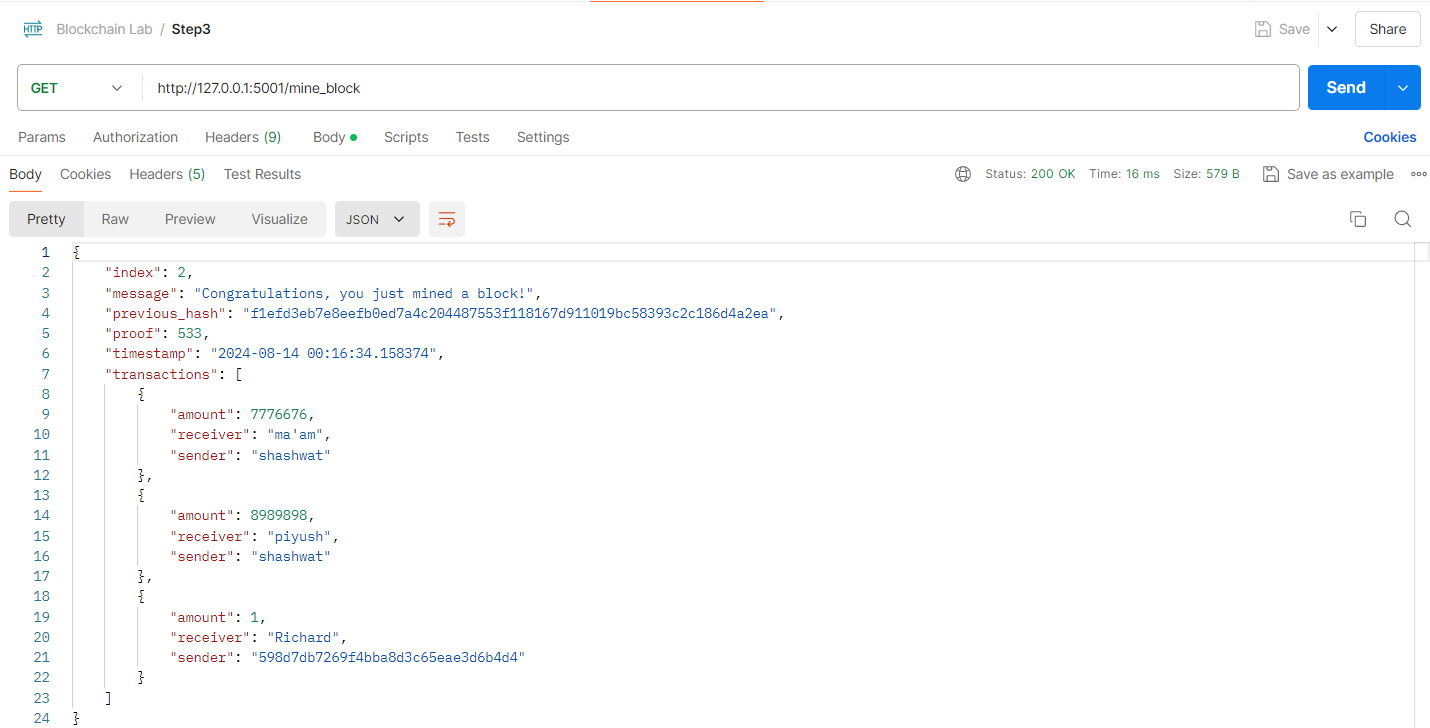
Replace the longest chain - replace\_chain()



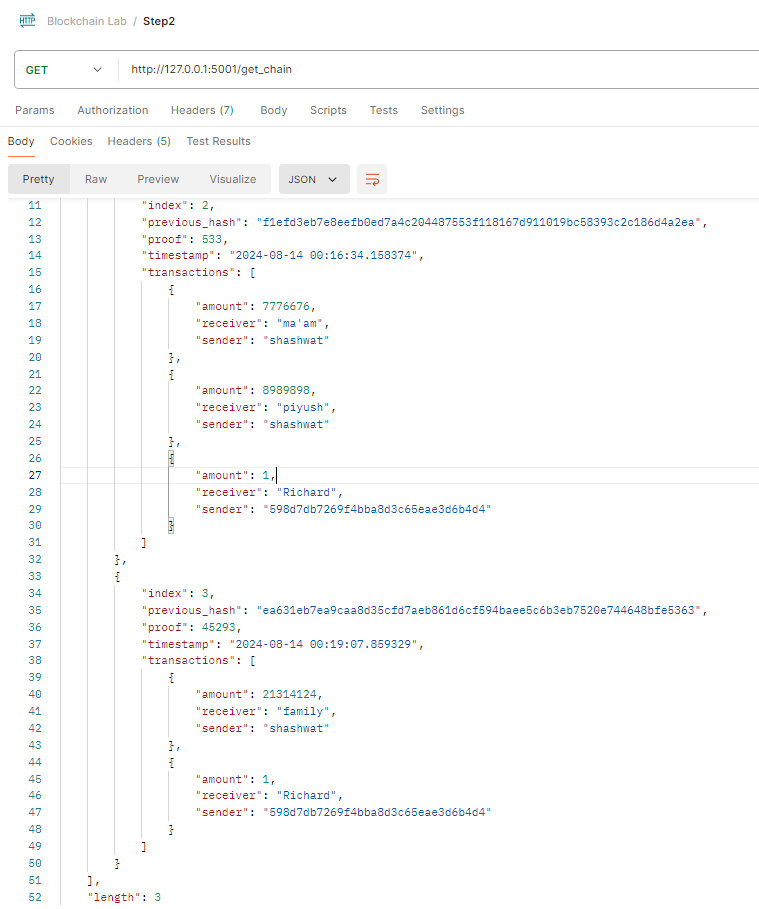
1. Modify the code such that transactions are removed after they are added to the block.

Modify the code such that transactions are removed after they are added to the block.

Added 2 txns and mined them



Now we will add another txn and mine the block, but this time when we call get\_chain(), it should only show the latest txn in last block, all the previous txns in above block only.



The txns of shashwat -> piyush and shashwat -> ma’am did not come into 3rd block. This is the because once they are mined, they are removed from the pool.

**Conclusion:** Thus, we have successfully created a blockchain using python and interacted using multiple nodes in the same chain.