Code -

```
import hashlib
import datetime
import json
from flask import Flask, jsonify

class Blockchain:
    def __init__(self):
        self.chain = []
        self.create_block(proof=1, previous_hash='0')

    def create_block(self, proof, previous_hash):
        block = {
            "index": len(self.chain) + 1,
            "timestamp": str(datetime.datetime.now()),
            "proof": proof,
            "previous_hash": previous_hash
        }

        self.chain.append(block)
        return block

    def get_previous_block(self):
        return self.chain[-1]

    def proof_of_work(self, previous_proof):
        new_proof = 1
        check_proof = False

        while check_proof == False:
            new_hash = hashlib.sha256(str(new_proof**2 - previous_proof**2).encode()).hexdigest()

            if new_hash[:4] == "0000":
                check_proof = True
            else:
                new_proof += 1

        return new_proof

    def hash(self, block):
        encoded_block = json.dumps(block, sort_keys=True).encode()
        return hashlib.sha256(encoded_block).hexdigest()

    def is_chain_valid(self, chain):
        previous_block = chain[0]
        block_index = 1

        while block_index < len(chain):
            block = chain[block_index]
            if block['previous_hash'] != self.hash(previous_block):
                return False

            previous_proof = previous_block['proof']
            proof = block['proof']
            hash_operation = hashlib.sha256(str(proof**2 - previous_proof**2).encode()).hexdigest()

            if hash_operation[:4] != '0000':
                return False
            previous_block = block
            block_index += 1

        return True

app = Flask(__name__)

blockchain = Blockchain()

@app.route('/mine_block', methods=['GET'])
def mine_block():
    previous_block = blockchain.get_previous_block()
    previous_proof = previous_block['proof']
    proof = blockchain.proof_of_work(previous_proof)
    previous_hash = blockchain.hash(previous_block)
    block = blockchain.create_block(proof, previous_hash)

    response = {'message': 'A block is MINED',
                'index': block['index'],
                'timestamp': block['timestamp'],
                'proof': block['proof'],
                'previous_hash': block['previous_hash']}

    return jsonify(response), 200

@app.route('/get_chain', methods=['GET'])
def display_chain():
    response = {'chain': blockchain.chain,
                'length': len(blockchain.chain)}
    return jsonify(response), 200
```

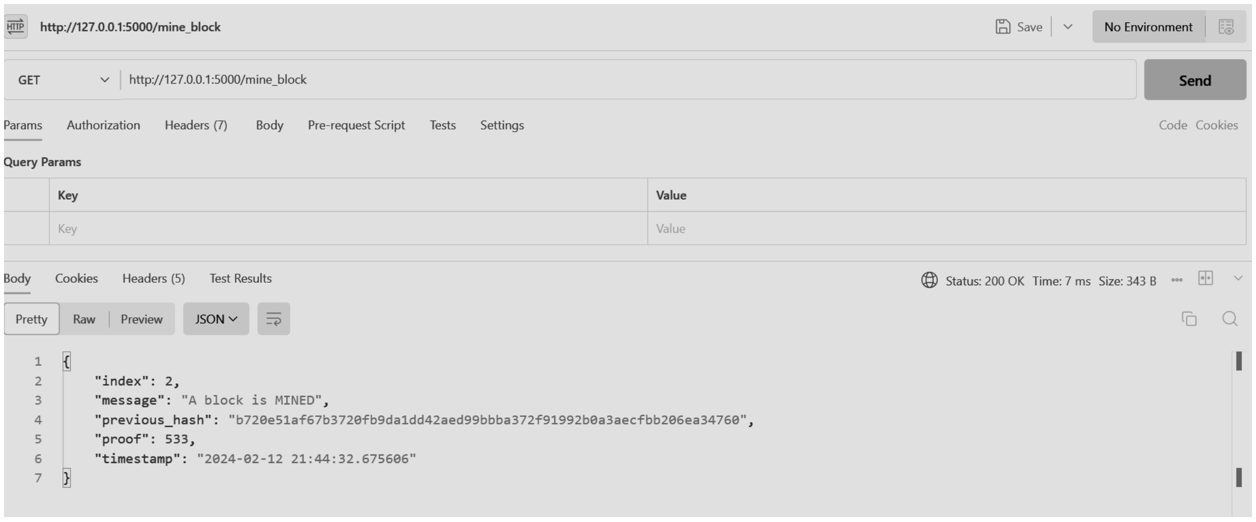
```
@app.route('/valid', methods=['GET'])
def valid():
    valid = blockchain.is_chain_valid(blockchain.chain)

    if valid:
        response = {'message': 'The Blockchain is valid.'}
    else:
        response = {'message': 'The Blockchain is not valid.'}
    return jsonify(response), 200

app.run(host='127.0.0.1', port=5000)
```

Output-

a. Mine Block.

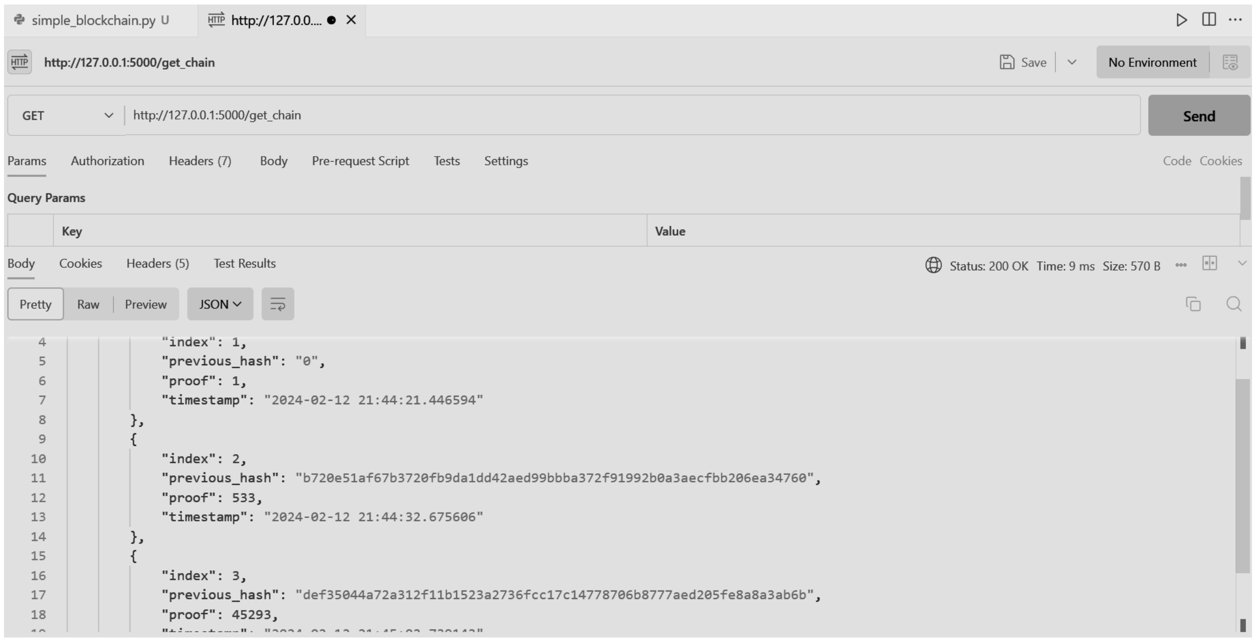


GET http://127.0.0.1:5000/mine_block

Status: 200 OK Time: 7 ms Size: 343 B

```
{
    "index": 2,
    "message": "A block is MINED",
    "previous_hash": "b720e51af67b3720fb9da1dd42aed99bbba372f91992b0a3aecfbb206ea34760",
    "proof": 533,
    "timestamp": "2024-02-12 21:44:32.675606"
}
```

b. Print Chain.



simple_blockchain.py U | http://127.0.0....

http://127.0.0.1:5000/get_chain

GET http://127.0.0.1:5000/get_chain

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

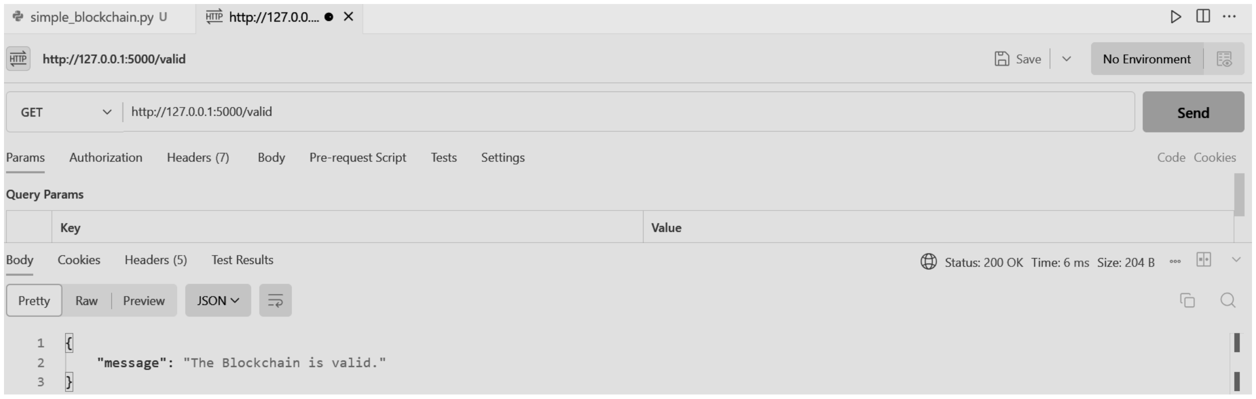
Key	Value

Body Cookies Headers (5) Test Results   Status: 200 OK Time: 9 ms Size: 570 B

Pretty Raw Preview JSON

```
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2024-02-12 21:44:21.446594"
        },
        {
            "index": 2,
            "previous_hash": "b720e51af67b3720fb9da1dd42aed99bbba372f91992b0a3aecfbb206ea34760",
            "proof": 533,
            "timestamp": "2024-02-12 21:44:32.675606"
        },
        {
            "index": 3,
            "previous_hash": "def35044a72a312f11b1523a2736fcc17c14778706b8777aed205fe8a8a3ab6b",
            "proof": 45293,
```

c. Valid Chain.



simple_blockchain.py U | http://127.0.0....

http://127.0.0.1:5000/valid

GET http://127.0.0.1:5000/valid

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

Key	Value

Body Cookies Headers (5) Test Results   Status: 200 OK Time: 6 ms Size: 204 B

Pretty Raw Preview JSON

```
{
    "message": "The Blockchain is valid."
}
```