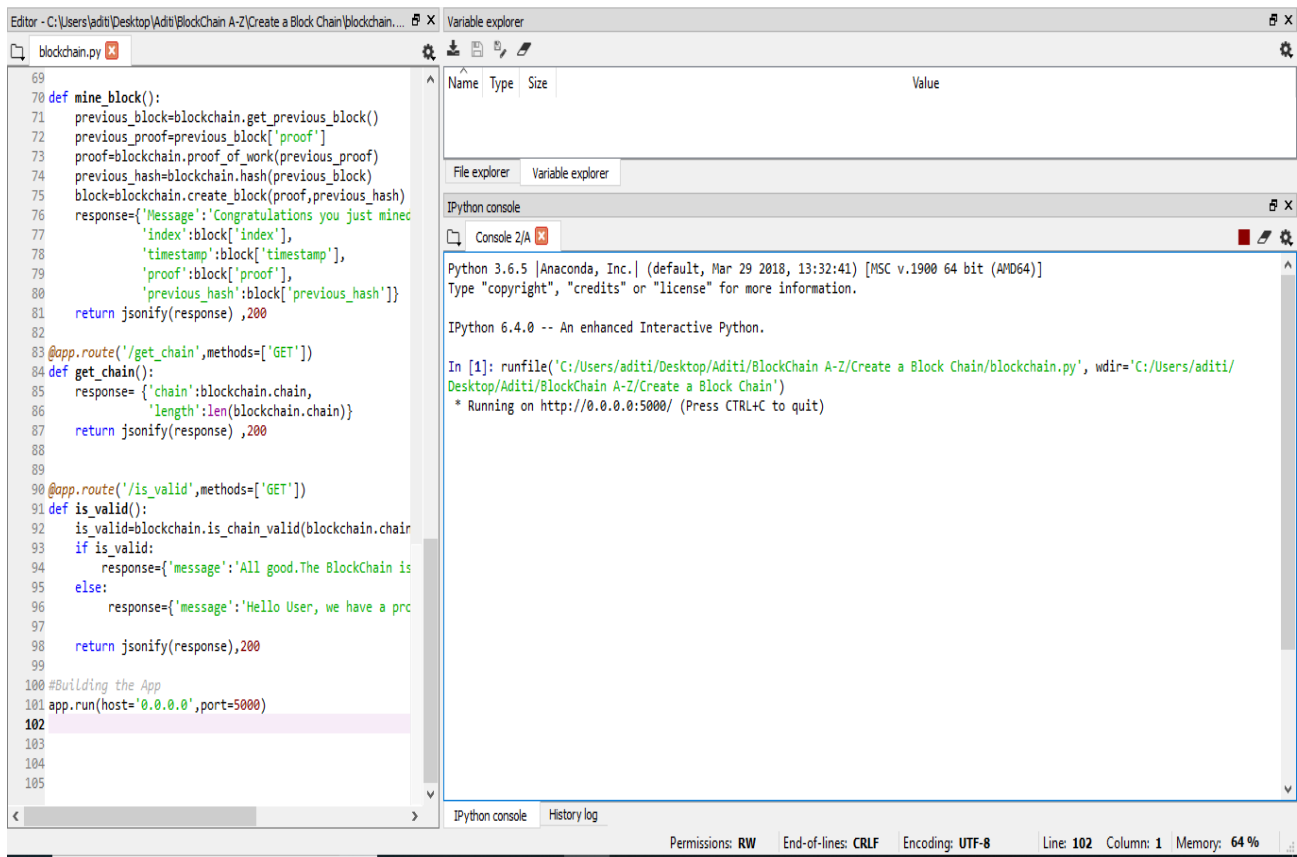


1. Running Code in IDE



The screenshot shows a Python IDE with a file named `blockchain.py` open. The code defines a simple blockchain with functions for mining a block, getting the chain, and validating it. The IPython console is running the script, showing the output of the `mine_block` function.

```
69 def mine_block():
70     previous_block=blockchain.get_previous_block()
71     previous_proof=previous_block['proof']
72     proof=blockchain.proof_of_work(previous_proof)
73     previous_hash=blockchain.hash(previous_block)
74     block=blockchain.create_block(proof,previous_hash)
75     response={'Message': 'Congratulations you just mined a block',
76              'index':block['index'],
77              'timestamp':block['timestamp'],
78              'proof':block['proof'],
79              'previous_hash':block['previous_hash']}
80     return jsonify(response),200
81
82 @app.route('/get_chain',methods=['GET'])
83 def get_chain():
84     response= {'chain':blockchain.chain,
85               'length':len(blockchain.chain)}
86     return jsonify(response),200
87
88 @app.route('/is_valid',methods=['GET'])
89 def is_valid():
90     is_valid=blockchain.is_chain_valid(blockchain.chain)
91     if is_valid:
92         response={'message':'All good.The BlockChain is valid'}
93     else:
94         response={'message':'Hello User, we have a problem'}
95     return jsonify(response),200
96
97 #Building the App
98 app.run(host='0.0.0.0',port=5000)
99
100
101
102
103
104
105
```

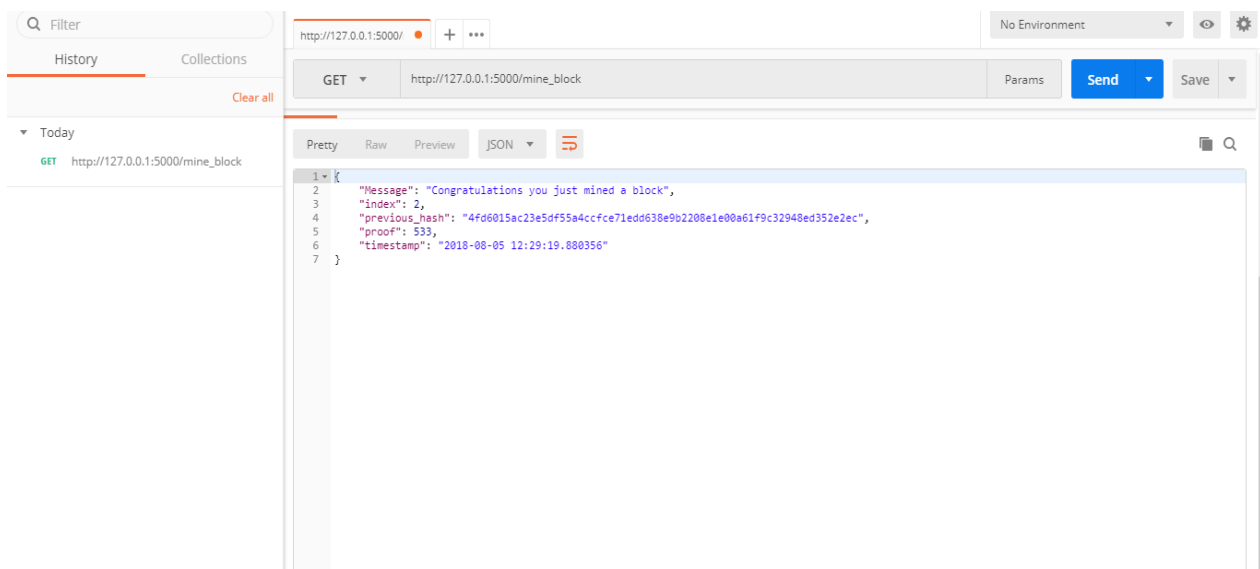
IPython console output:

```
Python 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.

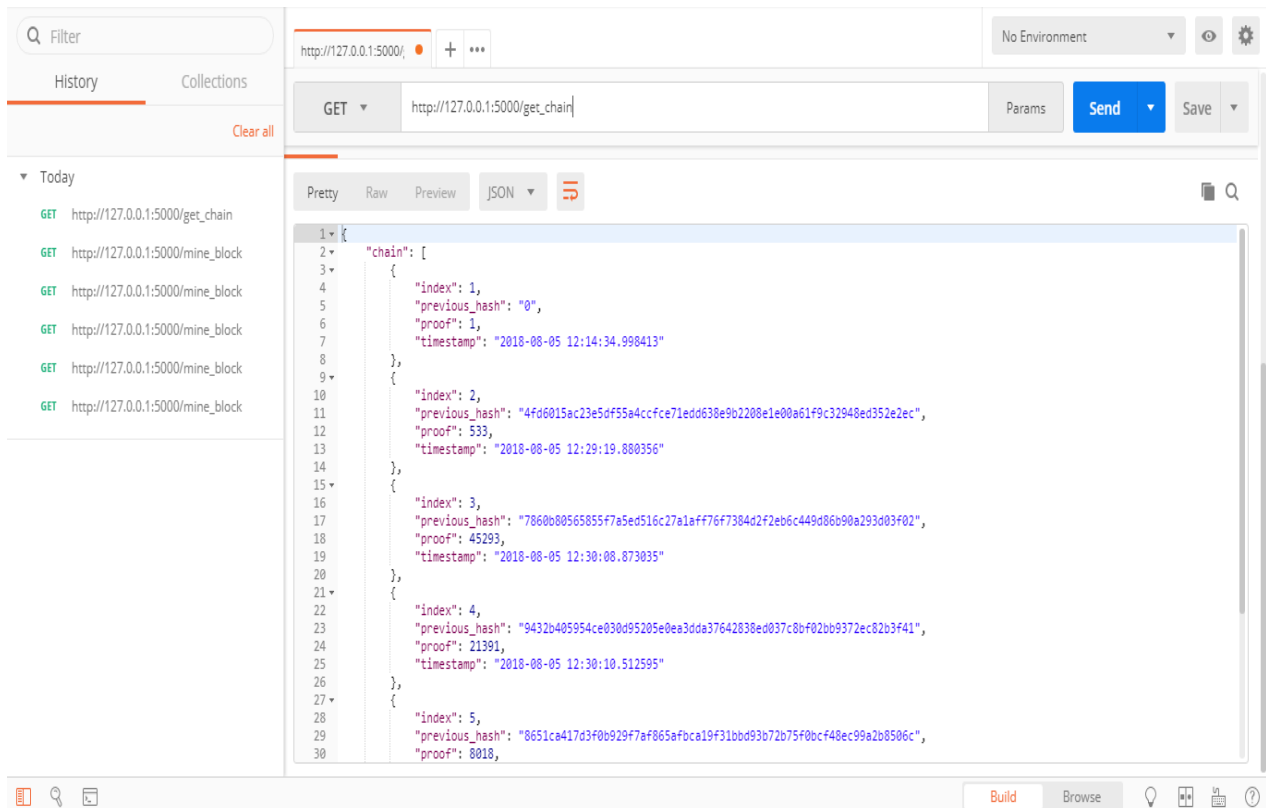
IPython 6.4.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/aditi/Desktop/Aditi/BlockChain A-Z/Create a Block Chain/blockchain.py', wdir='C:/Users/aditi/Desktop/Aditi/BlockChain A-Z/Create a Block Chain')
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

2. Mining a block



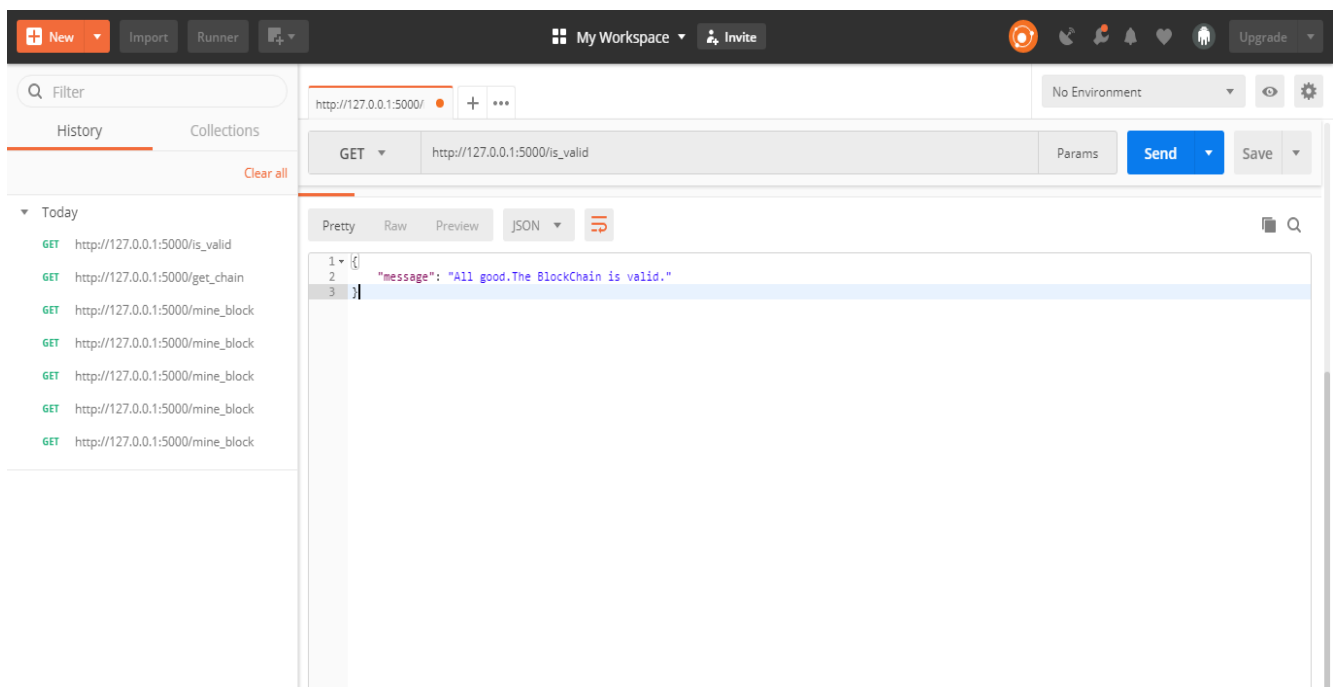
3. Getting the Chain



GET http://127.0.0.1:5000/get_chain

```
1 {
2   "chain": [
3     {
4       "index": 1,
5       "previous_hash": "0",
6       "proof": 1,
7       "timestamp": "2018-08-05 12:14:34.998413"
8     },
9     {
10      "index": 2,
11      "previous_hash": "4fd6015ac23e5df55a4ccfce71eddd630e9b2208e1e00a61f9c32948ed352e2ec",
12      "proof": 533,
13      "timestamp": "2018-08-05 12:29:19.880356"
14     },
15     {
16      "index": 3,
17      "previous_hash": "7860b80565855f7a5ed516c27a1aff76f7384d2f2eb6c449d86b90a293d03f02",
18      "proof": 45293,
19      "timestamp": "2018-08-05 12:30:08.873035"
20     },
21     {
22      "index": 4,
23      "previous_hash": "9432b405954ce030d95205e0ea3dda37642838ed037c8bf02bb9372ec82b3f41",
24      "proof": 21391,
25      "timestamp": "2018-08-05 12:30:10.512595"
26     },
27     {
28      "index": 5,
29      "previous_hash": "8651ca417d3f0b929f7af865afbc19f31bbd93b72b75f0bcf48ec99a2b8506c",
30      "proof": 8018,
```

4. Checking if Chain is Valid



GET http://127.0.0.1:5000/is_valid

```
1 {
2   "message": "All good. The Blockchain is valid."
3 }
```