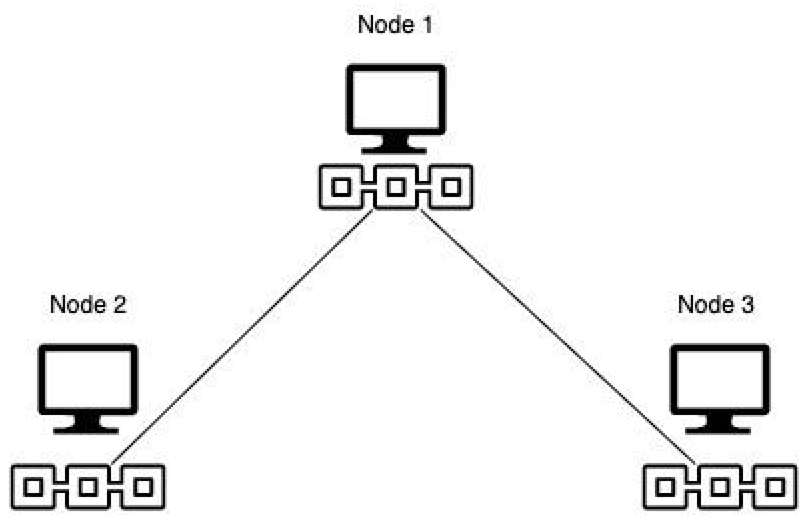
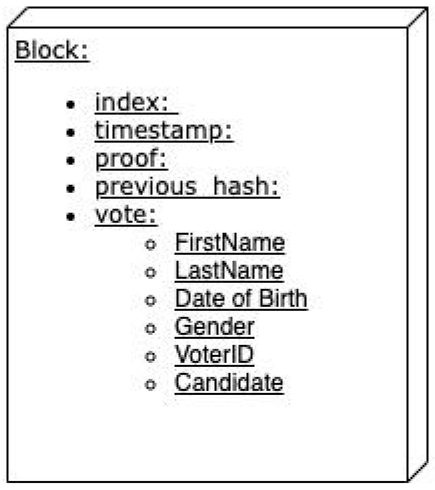
Project: "E-voting using Blockchain Technology"
Authors: Sereysathia Luy
Advisor: Omar Abuzaghleh
Description: A simple e-voting application using Blockchain Technology.
Programs used: anaconda, spyder, and postman

Overview of Application





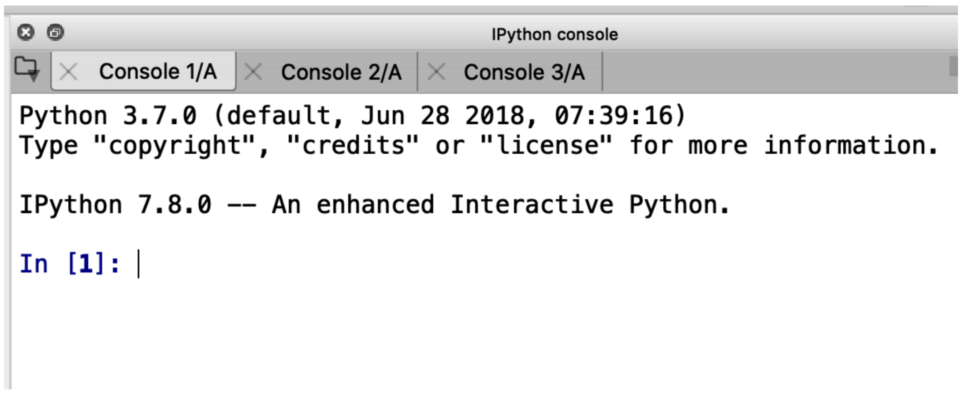
Block:

- index:
- timestamp:
- proof:
- previous_hash:
- vote:
    - FirstName
    - LastName
    - Date of Birth
    - Gender
    - VoterID
    - Candidate

Demo instruction

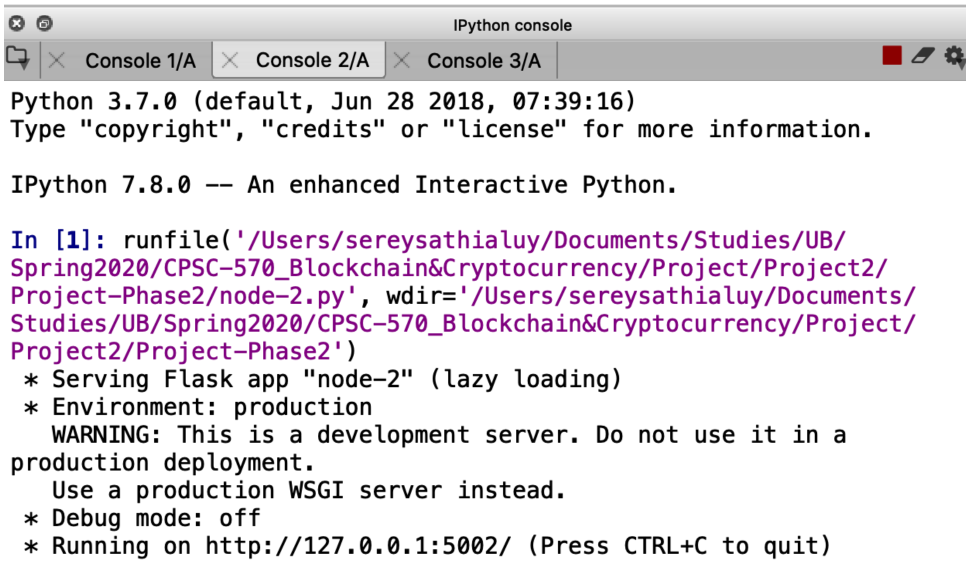
1. Running the 3 nodes:

- Open anaconda, then launch spyder
- In Spyder, open node-1.py, node-2.py, and node-3.py
- In IPython console, open 3 consoles by:
  - Right-click and click open an IPython console



```
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

In [1]:
```

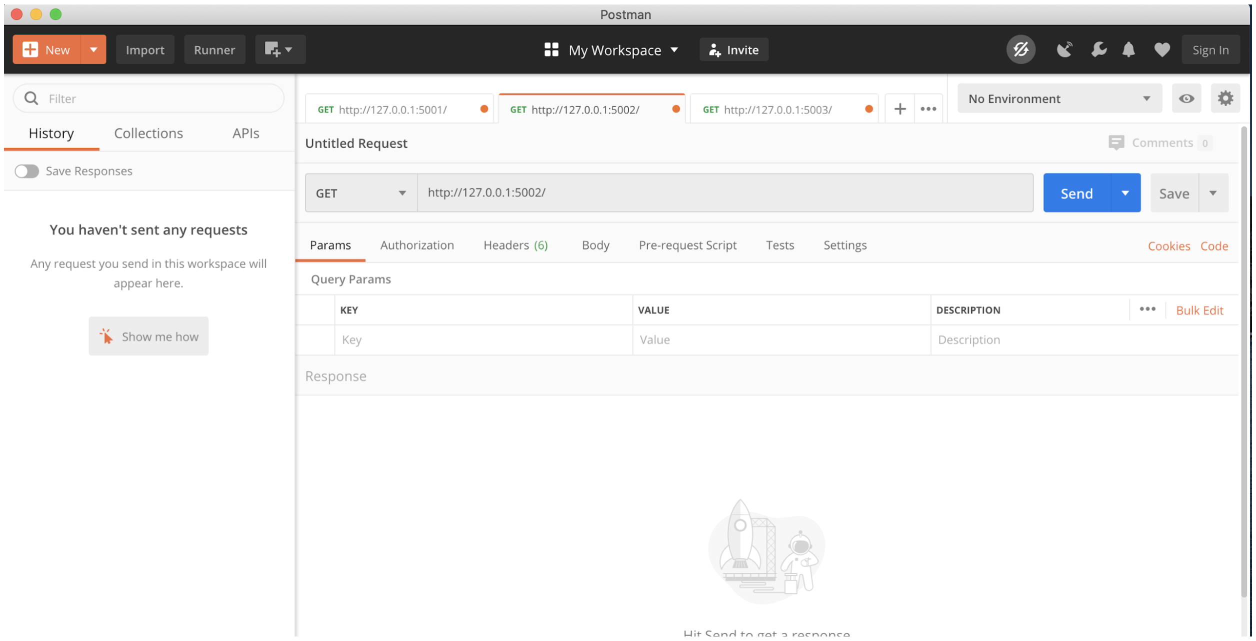
- Run node 1 on console 1, node 2 on console 2 and node 3 on console 3



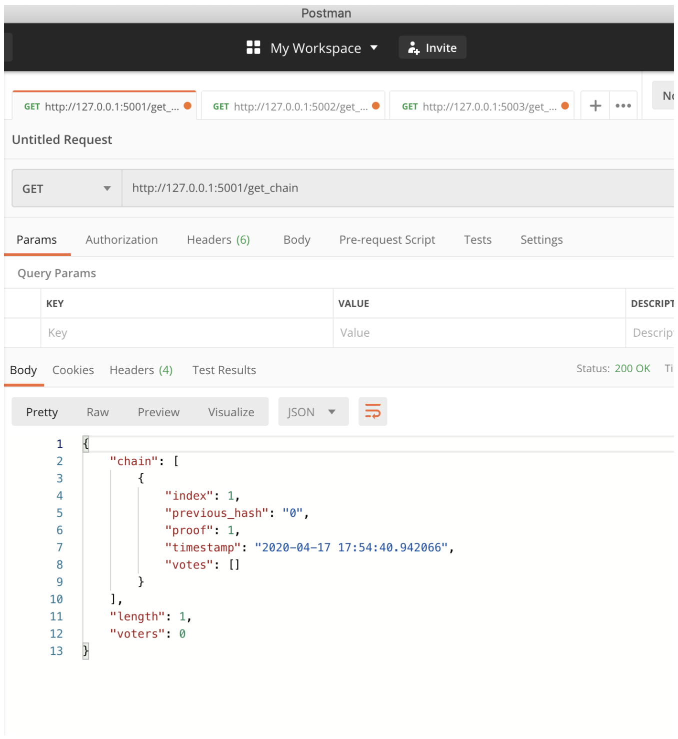
```
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/sereysathialuy/Documents/Studies/UB/
Spring2020/CPSC-570_Blockchain&Cryptocurrency/Project/Project2/
Project-Phase2/node-2.py', wdir='/Users/sereysathialuy/Documents/
Studies/UB/Spring2020/CPSC-570_Blockchain&Cryptocurrency/Project/
Project2/Project-Phase2')
 * Serving Flask app "node-2" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a
production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5002/ (Press CTRL+C to quit)
```

2. Postman:

- Open Postman, and add 3 request tabs
- Then add the address of each node
  - http://127.0.0.1:5001/ for tab 1
  - http://127.0.0.1:5002/ for tab 2
  - http://127.0.0.1:5003/ for tab 3



- Then call **get_chain** on each tab

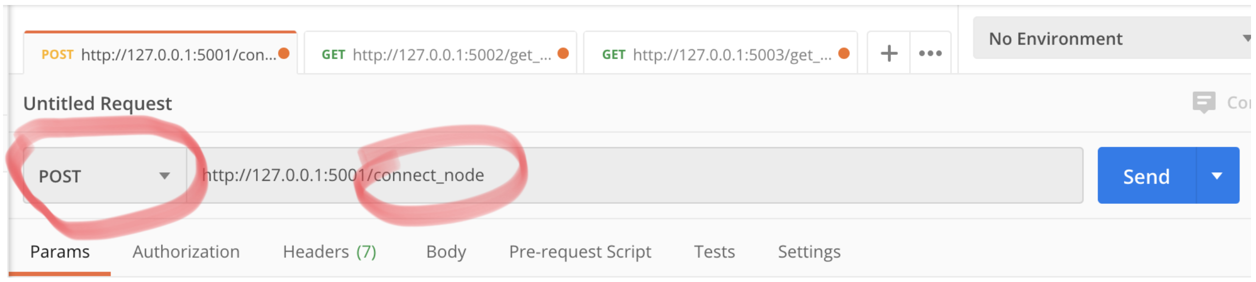


```
{
    "chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2020-04-17 17:54:40.942066",
            "votes": []
        }
    ],
    "length": 1,
    "voters": 0
}
```

- **Note**: notice that the timestamp in the genesis block of each node is a couple of seconds different
    - This is due to us calling get_chain a couple of seconds apart of each other
    - To fix this, we need to connect these 3 nodes to each other and replace the chain on each node with the longest chain in the network

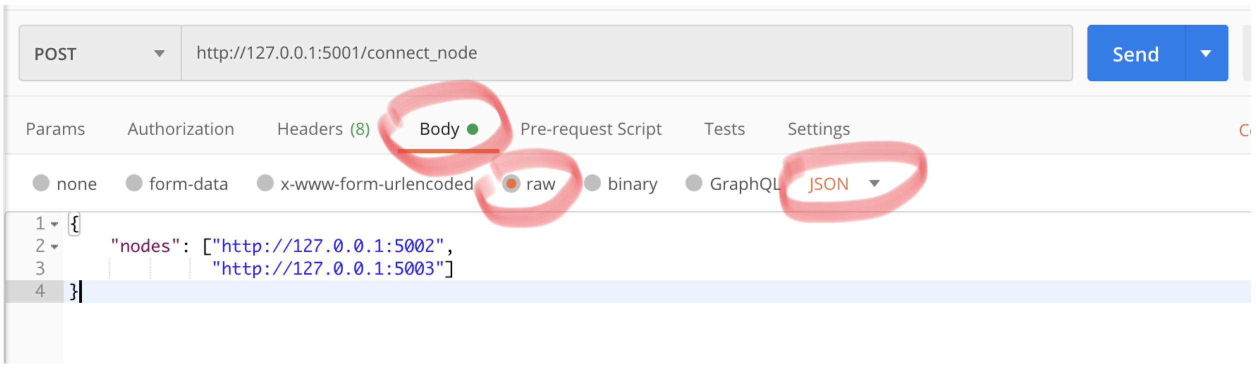
3. Connecting nodes:

- Change **GET** to **POST** and type the **connect_node**


POST http://127.0.0.1:5001/con...
GET http://127.0.0.1:5002/get_...
GET http://127.0.0.1:5003/get_...
No Environment
Untitled Request
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings

```
POST    http://127.0.0.1:5001/connect_node
```

- Then click on **Body**, then **raw** and then **JSON**, and type in the 2 other nodes' address in JSON format (check nodes.json file)

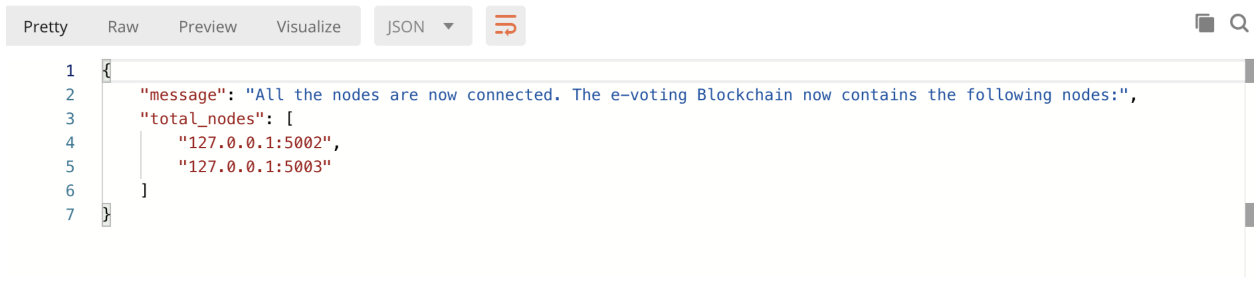


```
POST    http://127.0.0.1:5001/connect_node
```

```
{
    "nodes": ["http://127.0.0.1:5002",
              "http://127.0.0.1:5003"]
}
```

- Click send

Result:



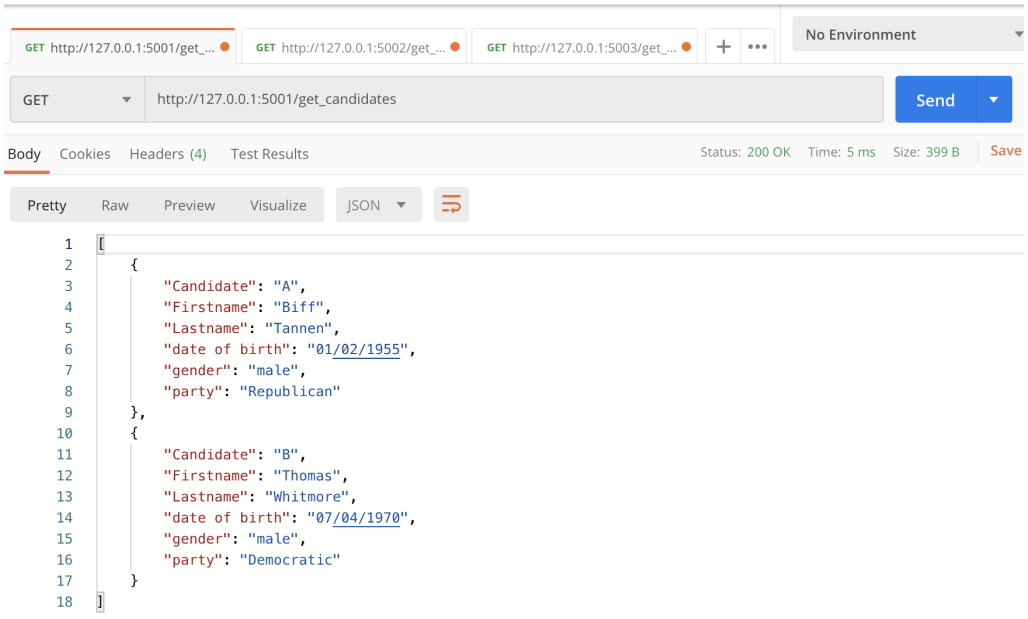
```
{
    "message": "All the nodes are now connected. The e-voting Blockchain now contains the following nodes:",
    "total_nodes": [
        "127.0.0.1:5002",
        "127.0.0.1:5003"
    ]
}
```

- Do the same to the other 2 nodes

Note: now all nodes are connected

4. Get candidates' information, register a voter and voter's details:

- Change **POST** back to **GET** and Call **get_candidates** to show all the information about the 2 primary candidate



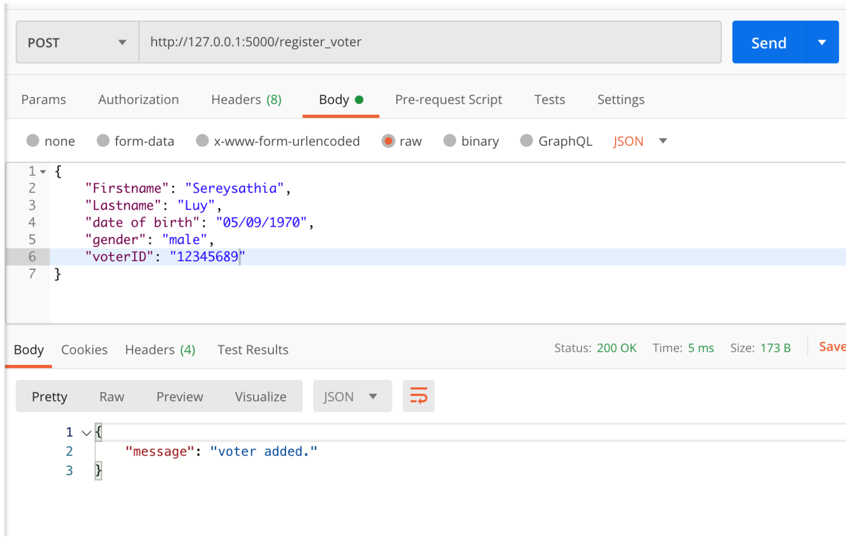
GET http://127.0.0.1:5001/get_...
GET http://127.0.0.1:5002/get_...
GET http://127.0.0.1:5003/get_...
No Environment
GET
Send
Body
Cookies
Headers (4)
Test Results
Save
Pretty
Raw
Preview
Visualize
JSON

http://127.0.0.1:5001/get_candidates

Status: 200 OK Time: 5 ms Size: 399 B

```
[
    {
        "Candidate": "A",
        "Firstname": "Biff",
        "Lastname": "Tannen",
        "date of birth": "01/02/1955",
        "gender": "male",
        "party": "Republican"
    },
    {
        "Candidate": "B",
        "Firstname": "Thomas",
        "Lastname": "Whitmore",
        "date of birth": "07/04/1970",
        "gender": "male",
        "party": "Democratic"
    }
]
```

- To add or register a voter to the system:
    - To add vote: Change **GET** to **POST,** type **register_voter**
    - Then click on **Body**, then **raw** and then **JSON**, and type in voter's information (check vote.json file)



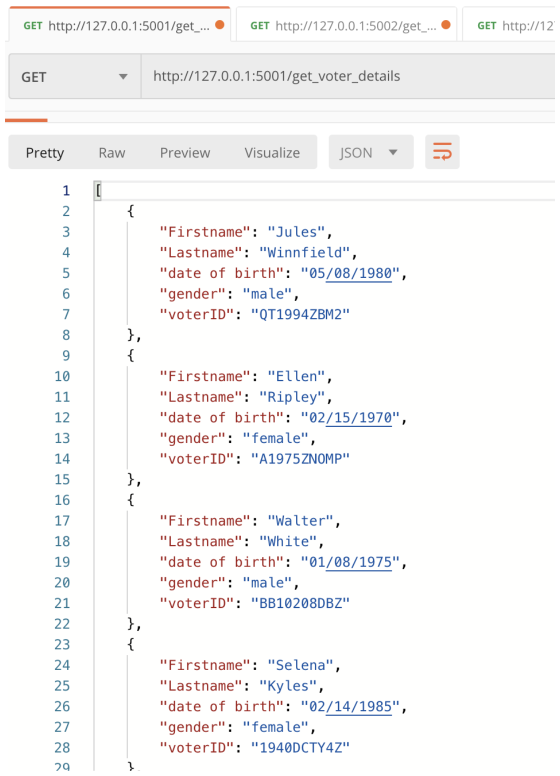
http://127.0.0.1:5000/register_voter

```
{
    "Firstname": "Sereysathia",
    "Lastname": "Luy",
    "date of birth": "05/09/1970",
    "gender": "male",
    "voterID": "12345689"
}
```

Status: 200 OK Time: 5 ms Size: 173 B

```
{
    "message": "voter added."
}
```

- Call **get_voter_details** to show all the voters' details registered in the system



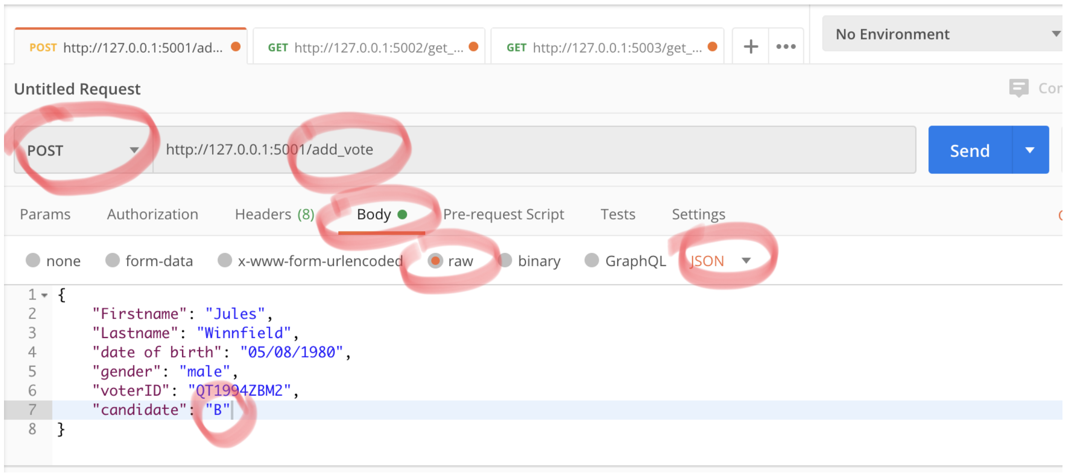
GET http://127.0.0.1:5001/get_voter_details

```
[
    {
        "Firstname": "Jules",
        "Lastname": "Winnfield",
        "date of birth": "05/08/1980",
        "gender": "male",
        "voterID": "QT1994ZBM2"
    },
    {
        "Firstname": "Ellen",
        "Lastname": "Ripley",
        "date of birth": "02/15/1970",
        "gender": "female",
        "voterID": "A1975ZNOMP"
    },
    {
        "Firstname": "Walter",
        "Lastname": "White",
        "date of birth": "01/08/1975",
        "gender": "male",
        "voterID": "BB10208DBZ"
    },
    {
        "Firstname": "Selena",
        "Lastname": "Kyles",
        "date of birth": "02/14/1985",
        "gender": "female",
        "voterID": "1940DCTY4Z"
```

5. Add vote to the blockchain, mine it, and replace chain in other nodes :

1. Add vote:

- To add vote: Change **GET** to **POST,** type **add_vote**
- Then click on **Body**, then **raw** and then **JSON**, and type in voter's information and the candidate to vote (A or B) in JSON format (check nodes.json file)

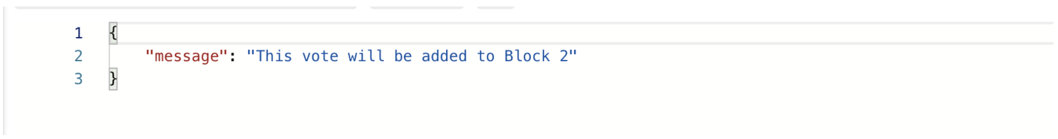


POST http://127.0.0.1:5001/add_vote

```
{
    "Firstname": "Jules",
    "Lastname": "Winnfield",
    "date of birth": "05/08/1980",
    "gender": "male",
    "voterID": "QT1994ZBM2",
    "candidate": "B"
}
```

- Click send

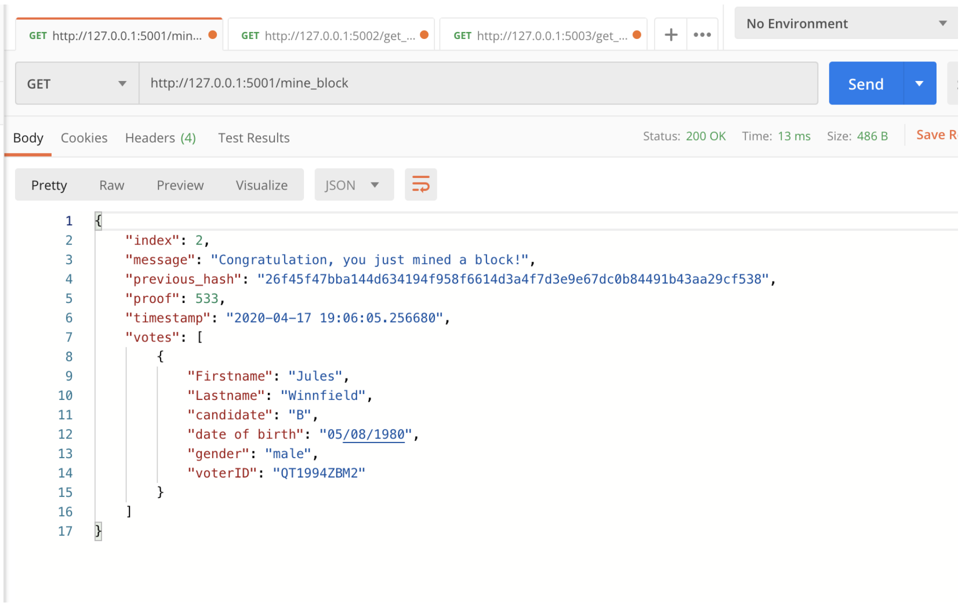
Result:



```
{
    "message": "This vote will be added to Block 2"
}
```

2. **Mine the block:**

○ Change **POST** to **GET,** call **mine_block**


GET http://127.0.0.1:5001/min...
GET http://127.0.0.1:5002/get_...
GET http://127.0.0.1:5003/get_...
No Environment
Send
Body
Cookies
Headers (4)
Test Results
Save R
Pretty
Raw
Preview
Visualize
JSON

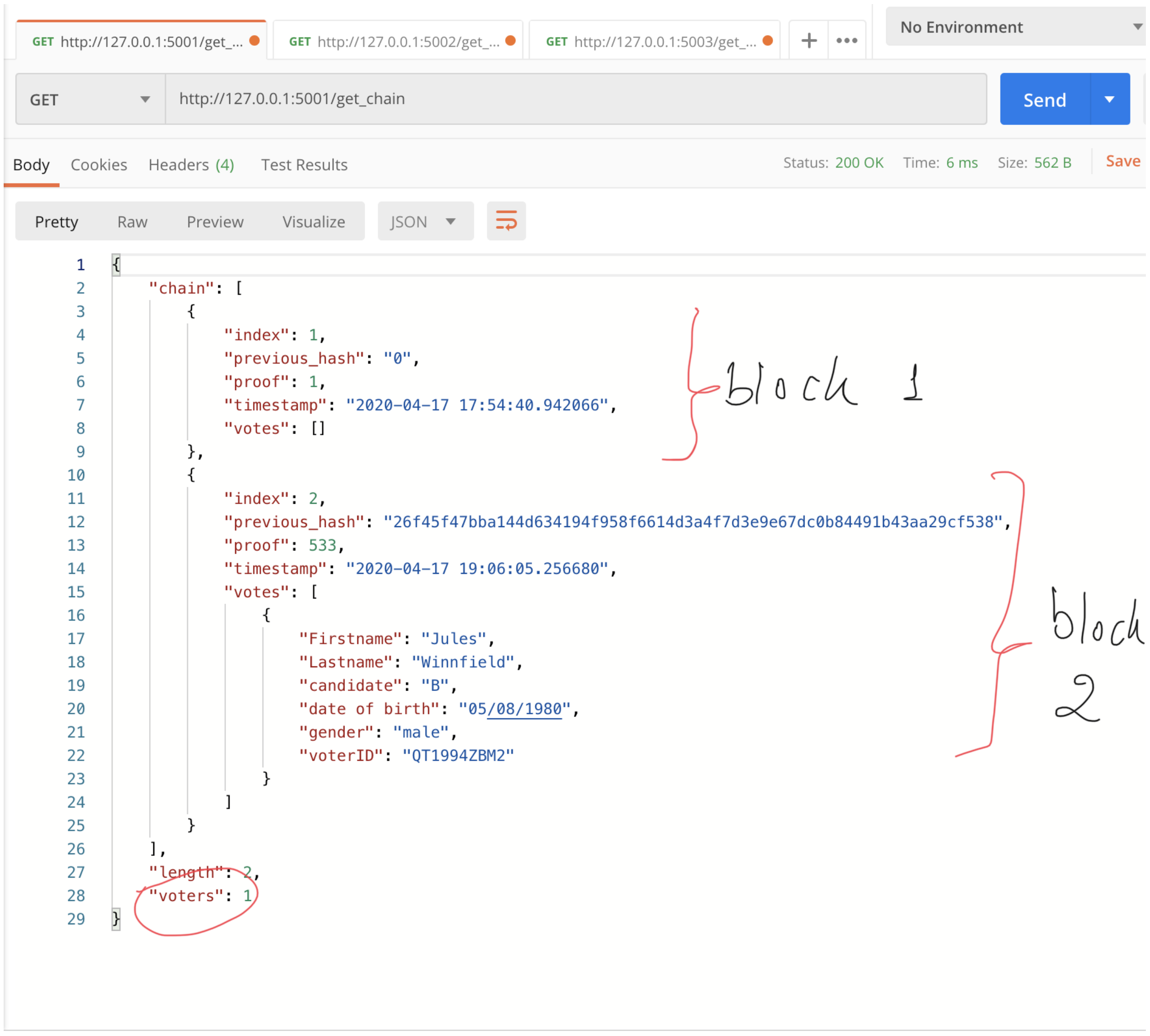
GET http://127.0.0.1:5001/mine_block

Status: 200 OK Time: 13 ms Size: 486 B

```
{
    "index": 2,
    "message": "Congratulation, you just mined a block!",
    "previous_hash": "26f45f47bba144d634194f958f6614d3a4f7d3e9e67dc0b84491b43aa29cf538",
    "proof": 533,
    "timestamp": "2020-04-17 19:06:05.256680",
    "votes": [
        {
            "Firstname": "Jules",
            "Lastname": "Winnfield",
            "candidate": "B",
            "date of birth": "05/08/1980",
            "gender": "male",
            "voterID": "QT1994ZBM2"
        }
    ]
}
```

3. **Get chain:**

- To check the blockchain, call **get_chain**


GET http://127.0.0.1:5001/get_...
GET http://127.0.0.1:5002/get_...
GET http://127.0.0.1:5003/get_...
No Environment
Send
Save
Body
Cookies
Headers (4)
Test Results
Pretty
Raw
Preview
Visualize
JSON
block 1
block 2

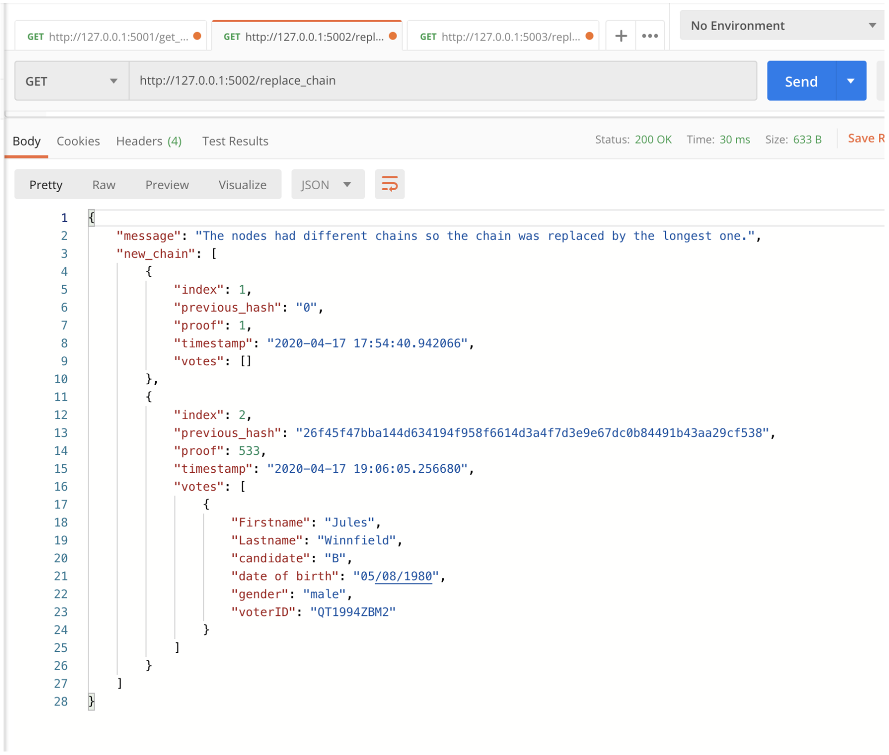
GET http://127.0.0.1:5001/get_chain

Status: 200 OK Time: 6 ms Size: 562 B

```
{
    "chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2020-04-17 17:54:40.942066",
            "votes": []
        },
        {
            "index": 2,
            "previous_hash": "26f45f47bba144d634194f958f6614d3a4f7d3e9e67dc0b84491b43aa29cf538",
            "proof": 533,
            "timestamp": "2020-04-17 19:06:05.256680",
            "votes": [
                {
                    "Firstname": "Jules",
                    "Lastname": "Winnfield",
                    "candidate": "B",
                    "date of birth": "05/08/1980",
                    "gender": "male",
                    "voterID": "QT1994ZBM2"
                }
            ]
        }
    ],
    "length": 2,
    "voters": 1
}
```

4. **Replace chain:**

- In order to replace this chain to the chain at the other 2 nodes, type replace_chain at the other 2 nodes:


GET http://127.0.0.1:5001/get_...
GET http://127.0.0.1:5002/repl...
GET http://127.0.0.1:5003/repl...
No Environment
Send
Body
Cookies
Headers (4)
Test Results
Pretty
Raw
Preview
Visualize
JSON

GET http://127.0.0.1:5002/replace_chain

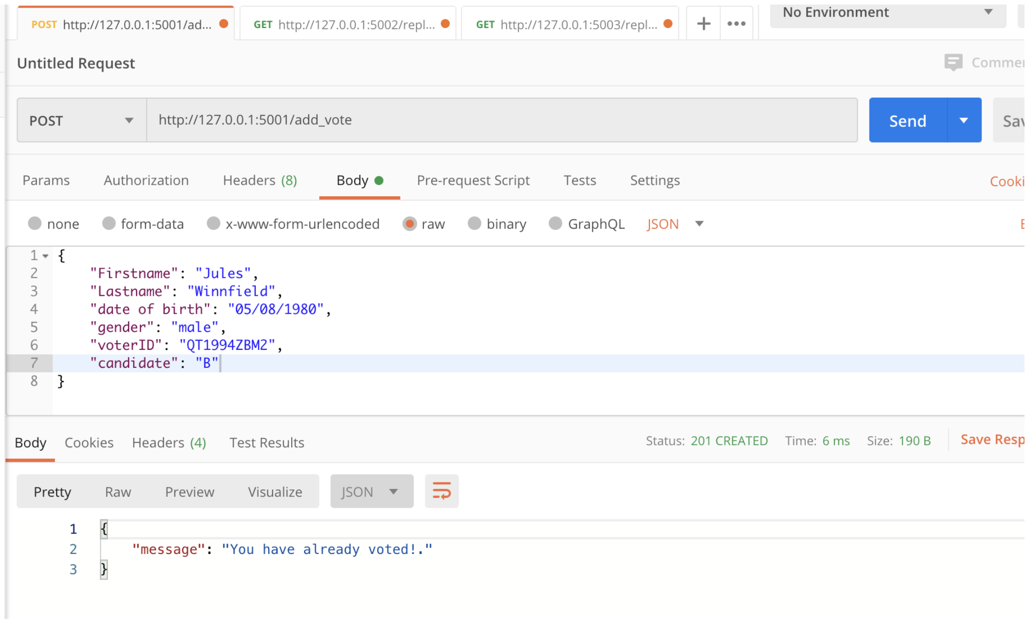
Status: 200 OK Time: 30 ms Size: 633 B

```
{
    "message": "The nodes had different chains so the chain was replaced by the longest one.",
    "new_chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2020-04-17 17:54:40.942066",
            "votes": []
        },
        {
            "index": 2,
            "previous_hash": "26f45f47bba144d634194f958f6614d3a4f7d3e9e67dc0b84491b43aa29cf538",
            "proof": 533,
            "timestamp": "2020-04-17 19:06:05.256680",
            "votes": [
                {
                    "Firstname": "Jules",
                    "Lastname": "Winnfield",
                    "candidate": "B",
                    "date of birth": "05/08/1980",
                    "gender": "male",
                    "voterID": "QT1994ZBM2"
                }
            ]
        }
    ]
}
```

Note: now all the other nodes have the same chain, basically we decentralized our blockchain

5. **If the same voter attempted to vote again:**

- To ensure a **fair election**, one voter can only vote once. If they attempted to vote again, this is the result:



POST http://127.0.0.1:5001/add_vote

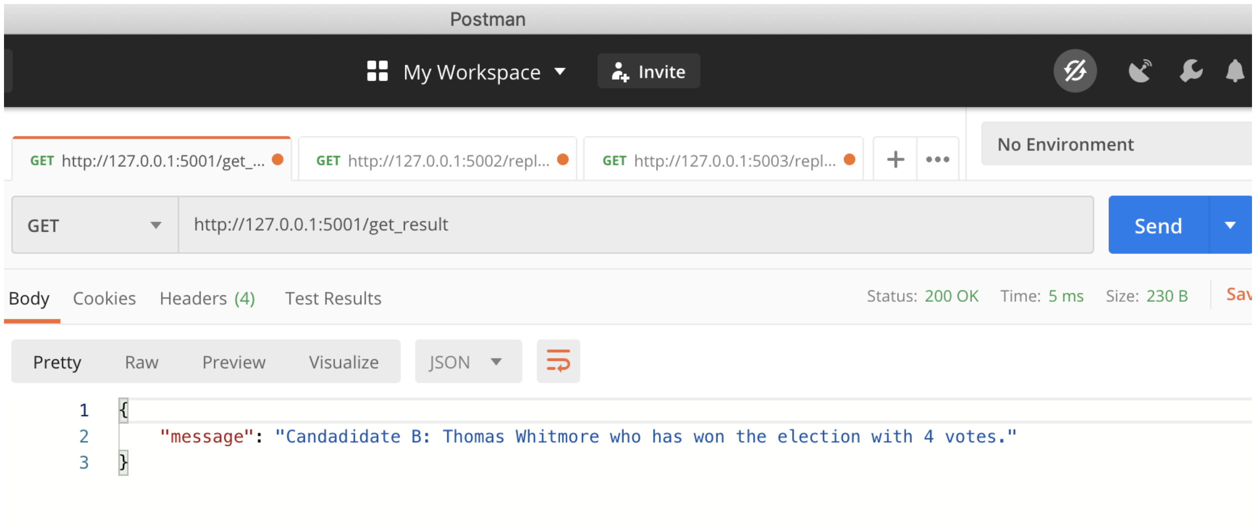
```
{
    "Firstname": "Jules",
    "Lastname": "Winnfield",
    "date of birth": "05/08/1980",
    "gender": "male",
    "voterID": "QT1994ZBM2",
    "candidate": "B"
}
```

Status: 201 CREATED Time: 6 ms Size: 190 B

```
{
    "message": "You have already voted!."
}
```

6. Get result

- Try adding more votes and mining them
  - Note: don't forget to call replace_chain on the other nodes (decentralize)
- Then after that to get the result of the election, call the **get_result**



```
http://127.0.0.1:5001/get_result
```

```
{
    "message": "Candadidate B: Thomas Whitmore who has won the election with 4 votes."
}
```