

0. Prerequisite : Jupyter notebook

Install conda to set up the python environment. <https://www.anaconda.com/> (<https://www.anaconda.com/>)

You also need to run

```
conda install jupyter conda install requests
```

to run this jupyter notebook.

After the intallation fininshed, move to the project root folder and then run in your command line below to launch this notebook.

```
jupyter notebook ./src/TestApp.ipynng
```

1. Run your application using the command `node app.js`

You should see in your terminal a message indicating that the server is listening in port 8000:

```
Server Listening for port: 8000
```

importrs

```
In [1]: import requests
import json
```

Logging

```
In [2]: import logging

logging.basicConfig()
logger = logging.getLogger()
logger.setLevel(logging.DEBUG)
```

2. To make sure your application is working fine and it creates the Genesis Block you can use POSTMAN to request the Genesis block:

```
In [3]: # Get Genesis block
url = 'http://localhost:8000'
genesisBlock = "/block/0"
headers = {"Content-Type": "application/json"}
resp = requests.get(url+genesisBlock,
                    headers=headers)
logger.debug(resp.json())

DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): localhost:8000
DEBUG:urllib3.connectionpool:http://localhost:8000 "GET /block/0 HTTP/1.1" 200 189
DEBUG:root:{'hash': 'db7e0ce585e1297ead064779990325d857ac9f95cc245033f02f55931cc82683', 'height': 0, 'body': '7b2264617461223a2247656e6573697320426c6f636b227d', 'time': '1563174966', 'previousBlockHash': None}
```

3. Make your first request of ownership sending your wallet address:

```
In [4]: # Request Validation
requestValidation = '/requestValidation'
data = {'address': 'n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh'}
resp = requests.post(url+requestValidation,
                    headers = headers,
                    data = json.dumps(data))
logger.debug(resp.json())

DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): localhost:8000
DEBUG:urllib3.connectionpool:http://localhost:8000 "POST /requestValidation HTTP/1.1" 200 60
DEBUG:root:n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh:1563175012:starRegistry
```

4. Sign the message with your Wallet:

After version 0.16, run the command in the console with "legacy" for address_type as below

```
getnewaddress "newaddress" "legacy"
```

After version 0.16, segwit address became the default and p2sh address is generated.

Bitcoin Core 0.16.0 introduces full support for segwit in the wallet and user interfaces. A new -addresstype argument has been added, which supports legacy, p2sh-segwit (default), and bech32 addresses.

In my test case, following is signature is generated

address: n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh

message: n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh:1562997899:starRegistry

H7PVR7vP/4acMjeKqhWvvRxTbdRa6r/JKmY6m+pD7UY7Ylkse1Awk5O/wW90wUI5u42zh9wmEdpMI4y550hZkMk=

5. Submit your Star

```
In [5]: # Submit start
submitstart = '/submitstar'
data = { 'address': 'n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh',
        'signature': 'IGtS96aWDNKjI67c/fiwW9zUFwUZh5HpQHDnWJJ/QuVMAw40Uwr+8WA27/CvgX2dMCvGTZKbA2mLXkA5LMkxh4=',
        'message': 'n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh:1563054933:starRegistry',
        'star': {
            'dec': "68° 52' 56.9",
            'ra': "16h 29m 1.0s",
            'story': "First Star"
        }
    }

resp = requests.post(url+submitstart,
                    headers = headers,
                    data = json.dumps(data))
logger.debug(resp.json())
```

```
DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): localhost:8000
DEBUG:urllib3.connectionpool:http://localhost:8000 "POST /submitstar HTTP/1.1" 200 439
DEBUG:root:{'hash': '3ac29f204b9d3c017d2194af9009ad53a6931dbc9b507ee54a312998df9c1fd0', 'height': 1, 'body': '7b226f776e6572223a226e3347766157756f5472357050466e5246444a6872486f3242795152507153664568222c2273746172223a7b22646563223a223638c2b0203532272035362e39222c227261223a223136682032396d20312e3073222c2273746f7279223a2246697273742053746172227d7d', 'time': '1563175012', 'previousBlockHash': 'db7e0ce585e1297ead064779990325d857ac9f95cc245033f02f55931cc82683'}
```

6. Retrieve Stars owned by me

```
In [6]: # Retrieve Stars owned by me
blocks = '/blocks/n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh'
resp = requests.get(url+blocks,
                    headers = headers)
logger.debug(resp.json())
```

```
DEBUG:urllib3.connectionpool:Starting new HTTP connection (1): localhost:8000
DEBUG:urllib3.connectionpool:http://localhost:8000 "GET /blocks/n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh HTTP/1.1" 200 122
DEBUG:root:[{'owner': 'n3GvaWuoTr5pPFnRFDJhrHo2ByQRPqSfEh', 'star': {'dec': "68° 52' 56.9", 'ra': '16h 29m 1.0s', 'story': 'First Star'}}]
```

Now could retrieve the same block.