Table of Contents

1 First attempt working with etherscan

Some links:

- address search (for attempt #1): https://etherscan.io/address/0x0008d343091ef8bd3efa730f6aae5a26a285c7a2 (https://etherscan.io/address/0x0008d343091ef8bd3efa730f6aae5a26a285c7a2)
  - shows all of the wallet's transactions
- general API website: https://etherscan.io/apis#transactions (https://etherscan.io/apis#transactions)
- bloxy site (this specific address): https://bloxy.info/address/0x0008d343091ef8bd3efa730f6aae5a26a285c7a2 (https://bloxy.info/address/0x0008d343091ef8bd3efa730f6aae5a26a285c7a2)
- GodsUnchained card holders & recent transactions: https://bloxy.info/token_holders/0x0e3a2a1f2146d86a604adc220b4967a898d7fe07 (https://bloxy.info/token_holders/0x0e3a2a1f2146d86a604adc220b4967a898d7fe07)
- **ERC721 TOKEN LIST:** https://bloxy.info/list_tokens/ERC721 (https://bloxy.info/list_tokens/ERC721)

1.1 1) Get the balance of an individual account:

https://etherscan.io/apis#accounts (https://etherscan.io/apis#accounts) - first link

- 1) open link
- 2) change URL to include address key (our example is address 0x0008d343091EF8BD3EFA730F6aAE5A26a285C7a2) and enter

In [1]:
```
from IPython.display import Image
Image(filename="search.jpg", width=1300, height=400)
```

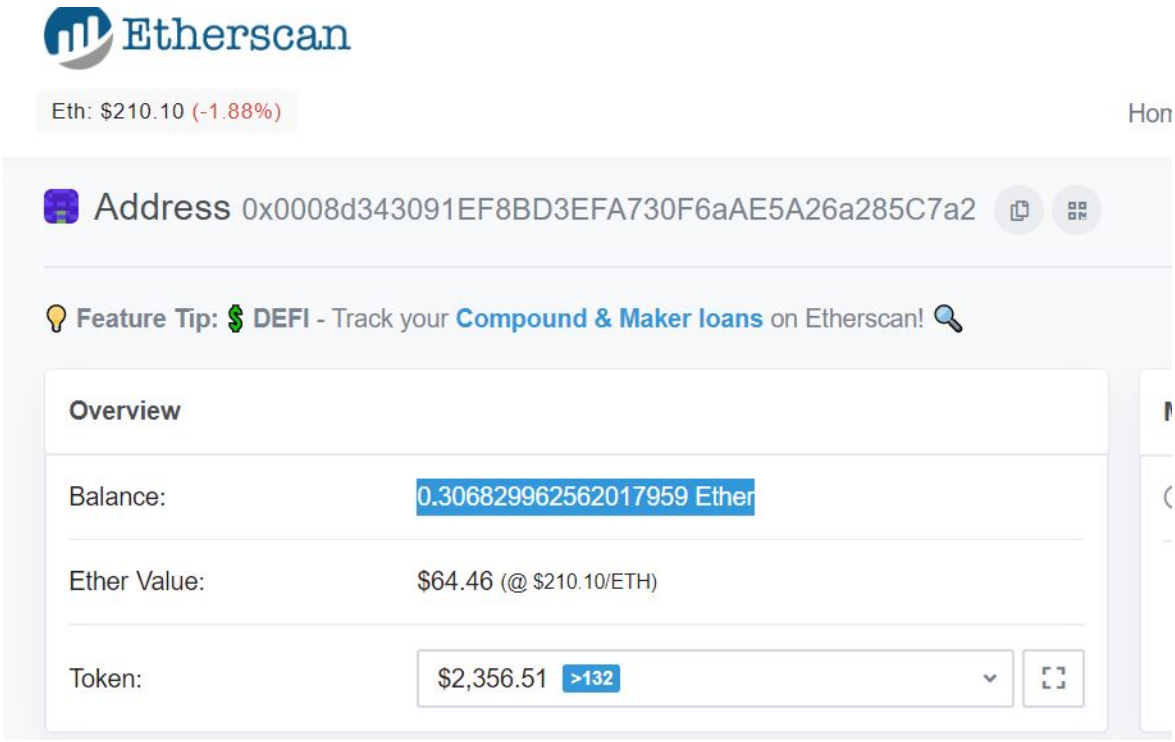
Out[1]:

As can be seen from the image above, the 'result' we got is 30682996...
Now let's look back at the Etherscan address page we saw before (the one

```
that shows all the transactions)
```

In [2]:
```
Image(filename="etherscanbal.jpg", width=800, height=300)
```

Out[2]:



In the highlighted field, we can see that the number matches what we found in our results on the other website.

1.2 2) Let's get a list of 'normal' transactions from this address

https://etherscan.io/apis#accounts (https://etherscan.io/apis#accounts) - third section

- has two links - first gives you list of last 10,000 transactions, second allows you to set a limit

```
    - (to change limit, change 'offset=10' in URL to 'offset=x' where
   x is the number of transactions you want to search for
    - sort=asc - order can also be changed
```

```
1) copy the same address, like before
2) change order to descending
2) Installed JSON formatter - made it much more readable
    download from chrome store: https://chrome.google.com/webstore/detai
l/json-formatter/bcjindcccaagfpapjjmafapmmgkkhgoa/related?hl=en
    extra help links:
        https://help.github.com/en/github/managing-files-in-a-repositor
y/adding-a-file-to-a-repository
        https://www.mattcutts.com/blog/how-to-install-a-chrome-extension
-from-github/
```

In [3]:
```
Image(filename="history.jpg", width=800, height=300)
```

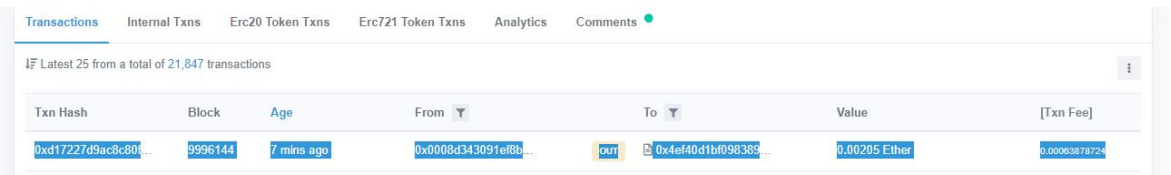
Out[3]:


```
{
    "status": "1",
    "message": "OK-Missing/Invalid API Key, rate limit of 1/3sec applied",
    "result": [
        {
            "blockNumber": "9996144",
            "timeStamp": "1588548105",
            "hash": "0xd17227d9ac8c80fa68f7aaaea26a6b7bed72cb6b2de80b3a5248f65ccf5a310c",
            "nonce": "21677",
            "blockHash": "0x23cb0ec82512e04603971cc19e562d17084d7cc7924f61bf46445c6527fe5e5e",
            "transactionIndex": "93",
            "from": "0x0008d343091ef8bd3efa730f6aae5a26a285c7a2",
            "to": "0x4ef40d1bf0983899892946830abf99eca2dbc5ce",
            "value": "2050000000000000",
            "gas": "446856",
            "gasPrice": "2210000000",
            "isError": "0",
            "txreceipt_status": "1",
            "input": "0x9e8323150000000000000000000000000000000000000000000000000000000000000000a0000000
            0000000000000000004a0000000000000000000000000000000000000000000000000000000000000004e0
            cd661d7d993c8a3b3e57b059ad8d7a4000000000000000000000000000000000000000000000000000000000000000
            0000000000000000000000000000000100000000000000000000000000000000000000000000000000000000071a
            0000000000000000000000000000005f49f0a600000000000000000000000000000000000000000000000000000000
            0000000000000000000000000000000000000002a000000000000000000000000000000000000000000000000000000
            0000000000000000000000000000000000000000000026c62b300000000000000000000000000000000000000000
            0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
            0000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000
            1bbd02aba40f1d90e009c5705bc058efef3b6ce283c560f2b15e7152323c9a7220ef020000000000000000000000
            883d200000000000000000000000000000000000000000000000000000000000000000000000000000000000000
            "contractAddress": "",
            "cumulativeGasUsed": "6574336",
            "gasUsed": "289044",
            "confirmations": "286"
        },
        { … }, // 18 items
        { … }, // 18 items
        { … }, // 18 items
        { … }, // 18 items
        { … }, // 18 items
        { … }, // 18 items
        { … }, // 18 items
        { … } // 18 items
    ]
}
```


1.2.1 Here's what the new page looks like (w/ JSON formatter chrome extension)

In [4]:
```
Image(filename="mostrecent.jpg", width=1200, height=300)
```

Out[4]:



What we're looking at is the past 10 transactions for this wallet (I als
o have the other 9 most recent, but they're condensed as can be seen by
 the arrows below)
We can see that the hash, block, value all match between the two screens
hots

Let's try to do what we just did via Python code:

In [5]:
```
import requests # the requests module will help us connect to etherscan



address = '0x0008d343091EF8BD3EFA730F6aAE5A26a285C7a2' # wallet address
key =  'MV3PBAW3Y5U9IC3WQKZGX8NTUGS9VAEZ28' # my api key
# change https to http (not encrypted)
url = 'http://api.etherscan.io/api?module=account&action=txlist&address=' + a
'&endblock=99999999&page=1&offset=10&sort=desc&apikey=' + key
# cut out default address and add the one you want to search for
"""
NOTE - in the url:
    offset changes the number of records that will be searched (in this case,
    sort can change whether you're looking at the most recent transactions or
"""



# connect to address
response = requests.get(url)
print(response)
```

```
<Response [200]>
```

Response [200] means we're entering a valid URL. If the program couldn't find it, we would get an error

In [6]:
```
# print(response.content)

"""
response.content will show all of the past transactions that we've requested.

uncomment to see how it works - I don't yet know how to truncate the results
"""
```

Out[6]:
```
"\nresponse.content will show all of the past transactions that we've reque
sted.\n\nuncomment to see how it works - I don't yet know how to truncate t
he results and they were quite long so I just commented it out after seeing
what it did\n"
```

In [7]:
```
#print(response.json()) # a cleaner way to request the content? - have not tr
```

In [8]:
```
address_content = response.json()
result = address_content.get("result") # get only the result of the transacti
#print(result)
```

Now, let's try to loop through them

In [9]:
```
# for transaction in result:
#     print(transaction)

"""
this works - but I again did not know how to truncate it so it's commented ou
"""
```

Out[9]:
```
"\nthis works - but I again did not know how to truncate it so it's comment
ed out\n"
```

In [10]:
```
for transaction in result: # this loop will request certain data from each tr
    # I believe these are pulling directly from the json text on the website
    hash_ = transaction.get("hash")
    tx_from = transaction.get("from")
    tx_to = transaction.get("to")
    value = transaction.get("value")
    confirmations = transaction.get("confirmations")

    print(f"hash: {hash_}\ntransaction from: {tx_from}\ntransaction to: {tx_t
```

```
hash: 0x9b6e1adc00089a24738f078840264497a39cc0835bb69d1eb155ef93cece8be0
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 112750000000000000
confirmations: 978
------------------------------
hash: 0x082ef1104957cdf976b9461968dc18ae01760049f1d3ab9863bf38cfd41fea8f
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2
value: 2935630000000000000
confirmations: 983
------------------------------
hash: 0x4075c05a5cd23fd35b1a30d711393d2c75ba665b911b2e5435b4c1eaad42a765
transaction from: 0x34872874b65e12408ec0265e9cf0a35fa6c8d13e
transaction to: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
value: 14768000000000000000
confirmations: 999
------------------------------
hash: 0x205b2c3a356017540f73d86b781f0e4f99e4c1d639b036a10f314bfcecb88eee
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 128125000000000000
confirmations: 7250
------------------------------
hash: 0x0a2cc10ce173476ca597098a398c1fb27a1ead32ad3ca800879811afae4d74c1
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 153750000000000000
confirmations: 20664
------------------------------
hash: 0xc84aa54858cc67bacafde2123e4d2fdce3bde07a8ead5615b549652ab8677a0e
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 153750000000000000
confirmations: 23329
------------------------------
hash: 0x81df3063e278e9a962b344d0890f7bd56f2c3ab2af32eecfd9752f32830600f3
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 451000000000000000
confirmations: 23331
------------------------------
hash: 0x891b02c691df012b9f5314c8aa49a31ea4fb13c5ed2194d4da1a4d14d1e69185
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
```

```
value: 425272500000000000
confirmations: 23719
------------------------------
hash: 0x2eef8cc199fa6c980232043bbe928f8efa78ab596e1fde4af3497eb1eed0b1cc
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 61500000000000000
confirmations: 24197
------------------------------
hash: 0x9641dd35dc590dc3e46fb76055a572e1a4206f2bc5be4810b9a420b02fcbd1f9
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 61500000000000000
confirmations: 24236
------------------------------
```

In [11]:

```
# lets try to get data from individual transactions

for n, transaction in enumerate(result): # this loop will request certain dat
    # this just adds in transaction ID (n)
    # in this case, transaction ID is relative
    hash_ = transaction.get("hash")
    tx_from = transaction.get("from")
    tx_to = transaction.get("to")
    value = transaction.get("value")
    confirmations = transaction.get("confirmations")

    print(f"transaction ID: {n}") # get ID number
    print(f"hash: {hash_}\ntransaction from: {tx_from}\ntransaction to: {tx_t
    print("\n")
```

```
transaction ID: 0
hash: 0x9b6e1adc00089a24738f078840264497a39cc0835bb69d1eb155ef93cece8be0
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 112750000000000000
confirmations: 978
------------------------------


transaction ID: 1
hash: 0x082ef1104957cdf976b9461968dc18ae01760049f1d3ab9863bf38cfd41fea8f
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2
value: 2935630000000000000
confirmations: 983
------------------------------


transaction ID: 2
hash: 0x4075c05a5cd23fd35b1a30d711393d2c75ba665b911b2e5435b4c1eaad42a765
transaction from: 0x34872874b65e12408ec0265e9cf0a35fa6c8d13e
transaction to: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
value: 14768000000000000000
confirmations: 999
------------------------------


transaction ID: 3
hash: 0x205b2c3a356017540f73d86b781f0e4f99e4c1d639b036a10f314bfcecb88eee
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 128125000000000000
confirmations: 7250
------------------------------


transaction ID: 4
hash: 0x0a2cc10ce173476ca597098a398c1fb27a1ead32ad3ca800879811afae4d74c1
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
```

```
value: 153750000000000000
confirmations: 20664
----------------------------


transaction ID: 5
hash: 0xc84aa54858cc67bacafde2123e4d2fdce3bde07a8ead5615b549652ab8677a0e
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 153750000000000000
confirmations: 23329
----------------------------


transaction ID: 6
hash: 0x81df3063e278e9a962b344d0890f7bd56f2c3ab2af32eecfd9752f32830600f3
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 451000000000000000
confirmations: 23331
----------------------------


transaction ID: 7
hash: 0x891b02c691df012b9f5314c8aa49a31ea4fb13c5ed2194d4da1a4d14d1e69185
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 425272500000000000
confirmations: 23719
----------------------------


transaction ID: 8
hash: 0x2eef8cc199fa6c980232043bbe928f8efa78ab596e1fde4af3497eb1eed0b1cc
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 61500000000000000
confirmations: 24197
----------------------------


transaction ID: 9
hash: 0x9641dd35dc590dc3e46fb76055a572e1a4206f2bc5be4810b9a420b02fcbd1f9
transaction from: 0x0008d343091ef8bd3efa730f6aae5a26a285c7a2
transaction to: 0x4ef40d1bf0983899892946830abf99eca2dbc5ce
value: 61500000000000000
confirmations: 24236
----------------------------
```

In [12]:

```
# make a python list with the different features we'll record
test = {'n': [n], 'hash': [hash_], 'from': [tx_from], 'to': [tx_to], 'value':
```

In [13]:
```
import pandas as pd

test_df = pd.DataFrame(test) # turn list into a DataFrame with one record
```

In [14]:
```
test_df
```

Out[14]:

	n	hash	
0	9	0x9641dd35dc590dc3e46fb76055a572e1a4206f2bc5be...	0x0008d343091ef8bd3efa730f6aae5a26a2

In [55]:
```
hash_[2:] # index slicing - maybe useful later on if we use this for a lot of
```

Out[55]:
```
'7173218290e224d6461b6d55ce3c8921d85b0adb5e1fc769a9676bbe6070130b'
```

Now, let's try to loop through and add each item to a DataFrame after every iteration

In [60]:
```
test_df = pd.DataFrame(columns=['ID', 'Hash', 'From', 'To', 'Value', 'Confirm
test_df # overwrite test_df and make it only contain the columns we're record
```

Out[60]:

	ID	Hash	From	To	Value	Confirmations

In [62]:
```
# lets try to get data from individual transactions
for n, transaction in enumerate(result):
    # this will loop through and get the data we want from each record and th
    hash_ = transaction.get("hash")
    tx_from = transaction.get("from")
    tx_to = transaction.get("to")
    value = transaction.get("value")
    confirmations = transaction.get("confirmations")

    #new_data = [n, hash_, tx_from, tx_to, value, confirmations]
    test_df = test_df.append({'ID': n, 'Hash': hash_, 'From': tx_from, 'To':

test_df
```

Out[62]:

	ID	Hash	
0	0	0x9b6e1adc00089a24738f078840264497a39cc0835bb6...	0x0008d343091ef8bd3efa730f6aae5a26
1	1	0x082ef1104957cdf976b9461968dc18ae01760049f1d3...	0x0008d343091ef8bd3efa730f6aae5a26
2	2	0x4075c05a5cd23fd35b1a30d711393d2c75ba665b911b...	0x34872874b65e12408ec0265e9cf0a35f
3	3	0x205b2c3a356017540f73d86b781f0e4f99e4c1d639b0...	0x0008d343091ef8bd3efa730f6aae5a26
4	4	0x0a2cc10ce173476ca597098a398c1fb27a1ead32ad3c...	0x0008d343091ef8bd3efa730f6aae5a26
5	5	0xc84aa54858cc67bacafde2123e4d2fdce3bde07a8ead...	0x0008d343091ef8bd3efa730f6aae5a26
6	6	0x81df3063e278e9a962b344d0890f7bd56f2c3ab2af32...	0x0008d343091ef8bd3efa730f6aae5a26
7	7	0x891b02c691df012b9f5314c8aa49a31ea4fb13c5ed21...	0x0008d343091ef8bd3efa730f6aae5a26
8	8	0x2eef8cc199fa6c980232043bbe928f8efa78ab596e1f...	0x0008d343091ef8bd3efa730f6aae5a26
9	9	0x9641dd35dc590dc3e46fb76055a572e1a4206f2bc5be...	0x0008d343091ef8bd3efa730f6aae5a26

1.3 The other notebook in this file will attempt to do something similar with ERC721 tokens

In [ ]:
```
```