

# Bitpetro PoC

*Script to price hydrocarbon in BTC*

## Prep Notes

- Create the following conda environment: `conda create --name bitpetro_poc python=3.6 pandas numpy jupyterlab`
- Within the Jupyter Notebook, run the command `pip install bitcoinaverage` once

---

## Part 1

*Retrieve BTC price in USD*

The platform [BitcoinAverage](#) has been providing API service for almost a decade, includes ask and bid values, and provides an "average" of prices from multiple exchanges. Specifically:

Our Cryptocurrency price data is sourced and aggregated from exchanges all over the world, a selection of these can be found below:

Coinbase Pro, Bitstamp, Bitfinex, Kraken, Binance, Poloniex, Bittrex, Huobi, LocalBitcoins, ItBit, HitBTC, Coinfloor, LakeBTC, Gemini, Exmo, > Korbit, QuadrigaCX, Bitsquare, Quoine, Luno, Independant Reserve, Coinmate, Bitbay, Paymium, Rocktrading, CampBX, Loyalbit, Spacebtc, Mexbt, > Okcoin, Btcc, Btc38, Bitflyer.

Our bitcoin exchange data comes from over 50 of the world's leading exchanges around the world. It is provided in real-time bitcoin price information (when available), and users can filter by order book symbol.

This collection also provides an exchange liquidity index, where exchanges are scored based on their order book depth average from the spread midpoint, particularly in a bitcoin trading environment, or for those looking to hedge or arbitrage.

The partnerships we've formed with world's largest exchanges enable us to provide the most up to date data possible, in some cases even faster than the exchanges themselves provide to their own customers through their UI.

[Click here for further details.](#)

I've created an account with *BitcoinAverage* using my TCU email address and am on the Developer Plan which provides a free two week trial for API use. The Developer Plan is \$12/month. This plan can provide minute updates with about 80(?) calls a month. I can track my API activity at [ProBitcoin Dashboard](#).

Bitcoin Average provides API's following both REST and Websocket standards. REST (stateless) is (probably) sufficient for our needs; Websocket supports many requests per second and supports

duplex communication. [Click here for further API Documentation](#) and [here](#), [here](#), and [here](#) for additional programming notes.

I've been issued a public key: `NWJmYTI2OWYyZGNkNDIzMTkwNDY5ZjI5MTcwYmEzNGU` *We should hide this in an environmental variable if we end up using this account*

```
In [82]: # Libraries
import hashlib
import hmac
import requests
import time
import json
from bitcoinaverage import RestfulClient
#import pandas as pd
```

```
In [83]: restful_client = RestfulClient(public_key)
bitcoinaverage_btc_data = restful_client.ticker_global_per_symbol('BTCUSD') # real time
```

```
In [84]: bitcoinaverage_btc_data
```

```
Out[84]: {'ask': 23105.08,
'bid': 23099.66,
'last': 23103.6,
'high': 23426.48,
'low': 22687.07,
'volume': 36553.56755738,
'open': {'hour': 23094.62,
'day': 23280.32,
'week': 21133.76,
'month': 19231.32,
'month_3': 38688.26,
'month_6': 38060.16,
'year': 39616.64},
'averages': {'day': 22987.7, 'week': 23256.39, 'month': 21769.36},
'changes': {'price': {'hour': 8.98,
'day': -176.72,
'week': 1969.84,
'month': 3872.28,
'month_3': -15584.66,
'month_6': -14956.56,
'year': -16513.04},
'percent': {'hour': 0.04,
'day': -0.76,
'week': 9.32,
'month': 20.14,
'month_3': -40.28,
'month_6': -39.3,
'year': -41.68}},
'volume_percent': 73.99,
'timestamp': 1659478505,
'display_timestamp': '2022-08-02 22:15:05',
'display_symbol': 'BTC-USD'}
```

```
In [85]: bitcoinaverage_btc_data["ask"]
```

```
Out[85]: 23105.08
```

```
In [100... bitcoinaverage_btc_price = bitcoinaverage_btc_data["ask"]
```

## Part 2

*Retrieve hydrocarbon gas liquid prices in USD*

The platform [Commodity Prices](#) provides API calls for a wide range of commodities and currencies, including:

- BTC: Bitcoin
- WTIOIL: WTI Crude Oil
- BRENT OIL: Brent Crude Oil
- NG: Natural Gas

[see here for a complete list](#)

I've created an account with *Commodities-API* using my TCU email address and am on a free plan that supports 100 calls a month. Prices are updated hourly (or each minute under more expensive plans).

Click [here](#) for programming notes.

I've been issued a public key: 43cc211nx2oabxtnh1av77vuwh21rdei0zeqcvnm5xn17klpevlnwsg2kx37  
*We should hide this in an environmental variable if we end up using this account*

```
In [86]: API_KEY = '43cc211nx2oabxtnh1av77vuwh21rdei0zeqcvnm5xn17klpevlnwsg2kx37'
```

```
In [88]: # import requests
base_currency = 'USD'
symbol = 'NG' # Note that Commodities-API provides BTC prices as well
endpoint = 'latest'
access_key = '43cc211nx2oabxtnh1av77vuwh21rdei0zeqcvnm5xn17klpevlnwsg2kx37'
resp_ng = requests.get('https://commodities-api.com/api/'+endpoint+'?access_key='+access_key)
```

```
In [92]: resp_ng.json()
```

```
Out[92]: {'data': {'success': True,
  'timestamp': 1659479220,
  'date': '2022-08-02',
  'base': 'USD',
  'rates': {'NG': 0.13046314416177, 'USD': 1},
  'unit': 'per MMBtu'}}
```

```
In [93]: json_dict= json.loads(resp_ng.text)
json_dict
```

```
Out[93]: {'data': {'success': True,
  'timestamp': 1659479220,
```

```
'date': '2022-08-02',  
'base': 'USD',  
'rates': {'NG': 0.13046314416177, 'USD': 1},  
'unit': 'per MMBtu'}}
```

```
In [97]: json_dict['data']['rates']['NG']
```

```
Out[97]: 0.13046314416177
```

```
In [98]: commodities_ng_price = json_dict['data']['rates']['NG']
```

---

## Part 3

### Price NG in BTC

```
In [101... ng_to_btc = commodities_ng_price / bitcoinaverage_btc_price
```

```
In [102... ng_to_btc
```

```
Out[102... 5.646513414442624e-06
```

---

## More Notes

### One call for NG in BTC

```
In [104... # set endpoint and your API key  
endpoint = 'convert'  
# define from currency, to currency, and amount  
from_symbol = 'NG'  
to_symbol = 'BTC'  
amount = '1'
```

```
In [105... resp = requests.get('https://commodities-api.com/api/'+endpoint+'?access_key='+access_k
```

```
In [106... resp.json()
```

```
Out[106... {'data': {'success': True,  
  'query': {'from': 'NG', 'to': 'BTC', 'amount': '1'},  
  'info': {'timestamp': 1659481080, 'rate': 0.0003329213676799282},  
  'historical': False,  
  'date': '2022-08-02',  
  'result': 0.0003329213676799282,  
  'unit': 'per barrel for Oil, per ounce for Metals. Per 10 metric tons for Crude Palm Oil, Per MMBtu for Natural gas, Per Gallon for Ethanol. Per metric ton, per lb or per bushel for Agriculture'}}
```

```
In [107... json_dict= json.loads(resp_ng.text)
```

```
json_dict
```

```
Out[107... {'data': {'success': True,
          'timestamp': 1659479220,
          'date': '2022-08-02',
          'base': 'USD',
          'rates': {'NG': 0.13046314416177, 'USD': 1},
          'unit': 'per MMBtu'}}
```

```
In [108... json_dict['data']['rates']['NG']
```

```
Out[108... 0.13046314416177
```

---

## More Notes

*A second programming approach for BTC via BitcoinAverage*

```
In [110... public_key = 'NWJmYTI20WYyZGNkNDIzMTkwNDY5ZjI5MTcwYmEzNGU' # hide this in an environmen
url = 'https://apiv2.bitcoinaverage.com/indices/global/ticker/BTCUSD'
headers = {'x-ba-key': 'NzI0MjM4Njc1NGQ3NDU4Mzg1NWU3YYmU4MTdiMaA'}

# see all websocket example at https://github.com/bitcoinaverage/api-integration-exampl
```

```
In [111... result = requests.get(url=url, headers=headers)
```

```
In [112... json_dict= json.loads(result.text)
json_dict["ask"]
```

```
Out[112... 23123.51
```

```
In [113... json_dict["ask"]
```

```
Out[113... 23123.51
```

---