**Task**:

**Collecting price of BTC in USDT during 48 hours**

**Summary of task:**

This task has consisted of two-part. First, the close prices of BTC in every minute candles from 24 hours ago needed to be gathered. Second, the close aforementioned part must be done for the next 24 hours in real-time.

**Bitcoin’s historical prices data in xlsx format:**

### For this part of the task, I found a third-party library named python-binance. The way this library work needed Binance API to connect to the Binance servers via Python or several other programming languages. due to that, the first step is to create an account with Binance. An API key should be created after it.

Installing library:

**pip install python-binance**

Importing library:

**from binance. client import Client**

Initialize the client by passing API key and secret:

**client = Client (api\_key, api\_secret)**

make the call for historical data by requesting historical candle data:

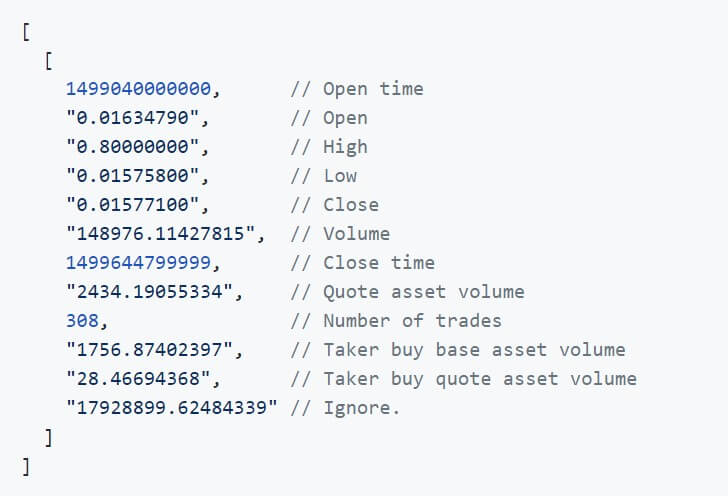
**bars = client.get\_historical\_klines ('BTCUSDT', '1m', timestamp, limit=1000)**

Candles interval

Star getting data from this Timestamp

symbol

In the background, this endpoint will continuously query the API in a loop, collecting 1000 price points at a time, until all data from the start point until today is returned.



Here is a definition of the data returned as per the Binance API documentation, and only the close and close time is needed: **bars\_arr = bars\_arr[:, [4,6]]**

After that, the data must be converted to a data frame and the time column should be transformed to date and time. At last, it would be written into excel.

**Bitcoin’s real time prices data in xlsx format:**

For this part, it is better to stream the updates. what I mean is that, The Kline/Candlestick Stream push updates to the current klines/candlestick every second and it is possible to collect the data in real-time. To stream the data, I choose the WebSocket library. Every second message would receive from the stream which is in JSON format and contain some information. Only the close price and close time of candles are important, duo to that the code would store the data from massages just when “Is this kline closed?” be True. Additionally, the massage must be converted to something that python understands.



Here is the message that we receive in the stream. the kline close time(“T”), close price(“c”) and Is this kline closed? (“x”) are needed.

Installing library:

**pip install websocket\_client**

Importing library:

**import websocket**

websocket.WebSocketApp a stream address connection needed.

* The base endpoint is: **wss://stream.binance.com:9443**
* Streams can be accessed either in a single raw stream or in a combined stream
* Raw streams are accessed at **/ws/<streamName>**
* For Kline/Candlestick Streams Name: <symbol>@kline\_<interval>

## The aforementioned lines were from [binance-spot-api](https://github.com/binance/binance-spot-api-docs) documentation. The address I will use:

**SOCKET = "wss://stream.binance.com:9443/ws/btcusdt@kline\_1m"**

## The stream must be close after 24 hours that’s why a timestamp needed to close it:

**tmiestamp\_close\_ws = datetime.now() + timedelta(hours=24)**

at the end all the collection data should be converted to data frame so they can be written on xlsx format file.