dYdX CLI Python Script

Document dated: Jul 17, 2023
Document written by: Lawrence Chiu (lawrence@dydx.exchange)
Version: 1.1

Update History

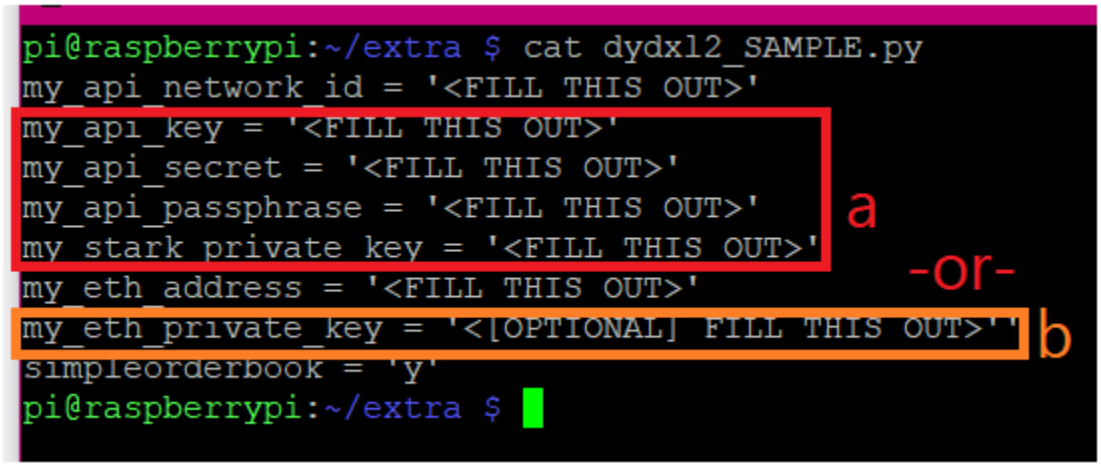
Version 1.1: Initial Jul 17, 2023. Commands supported: balance, positions, buyquantity, sellquantity, buyusdc, and sellusdc. Note that in the documentation below, the program alphatradingbotdydx.py (dev name) is same as dydxcli.py (release name on github).

Requirements:

1. Python3 Library: dydx-v3-python

Preparation:

1. First you have to create an apikeyfile. It looks like this:



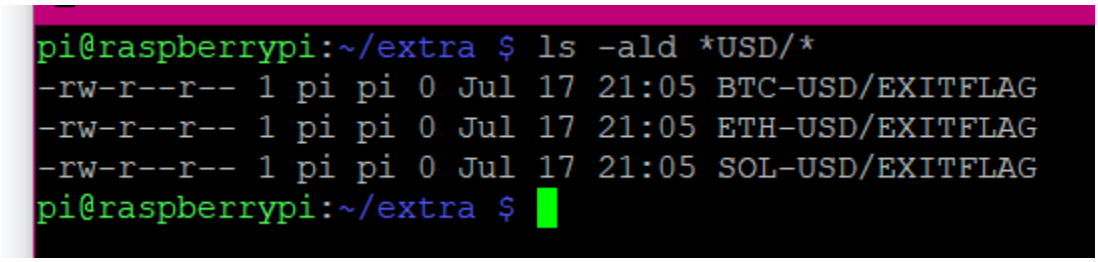
Note that you only fill out box a or box b but not both. If you fill out box a, then set the my_eth_private_key to null (two single quotes like this = ‘’). Or vice versa, if you fill out my_eth_private_key, then set the parameters in box a to null.

Next, You set my_api_network_id to either ‘5’ for testnet or ‘1’ for mainnet.

Then you set my_eth_address to your ethereum address, and

finally, you set simpleorderbook = ‘y’. If you run a local dYdX Orderbook websocket implementation on the same server, you can get a performance boost by setting this to ‘n’ and it’ll access the local files instead of calling the API. For your initial testing, please set this to ‘y’ for simplified operation.

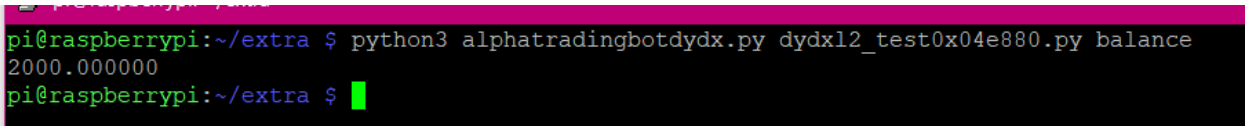
2. Create a directory for each market you are interested in. For our example, we will mkdir BTC-USD, ETH-USD, and SOL-USD.
3. In each of the above directories, create an empty file EXITFLAG.



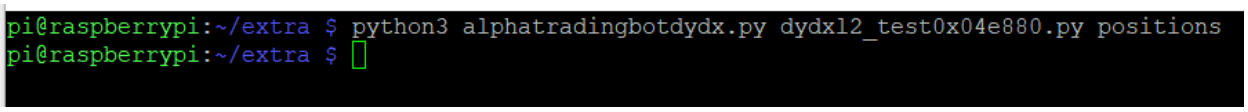
4. You will need the files alphatradingbotdydx.py, dydxob2.py (simpleorderbook), and tydxob2b.py (simpleorderbook for testnet) from the github in the working directory.

Commands:

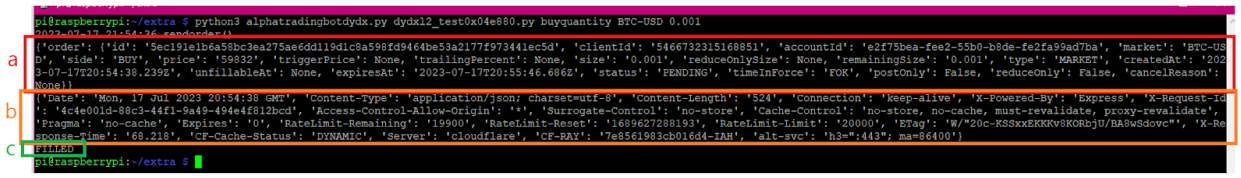
1. The CLI supports the following commands: balance, positions, buyquantity, sellquantity, buyusdc, and sellusdc. You run the python script with the desired command and any required parameters as shown below:
    a. ‘balance’: Gets your account equity.



    b. ‘positions’: Gets your account positions. In this example, we do not have any positions.

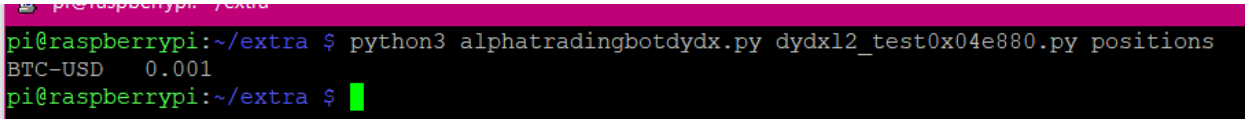


    c. ‘buyquantity’: Buys the specified quantity at market.

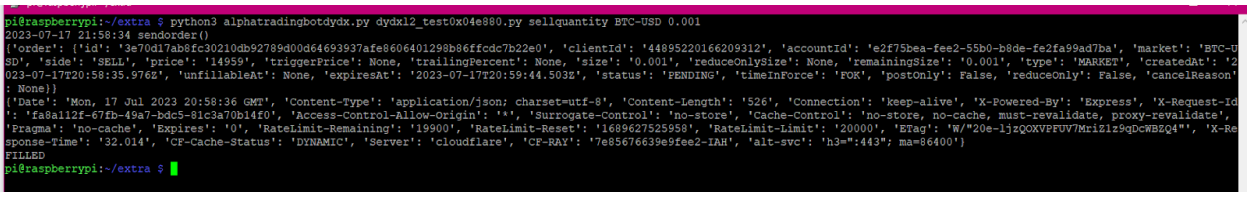


The output consists of 3 parts: a) is the api response, b) is the api response headers, and c) is the order status = FILLED in this case.

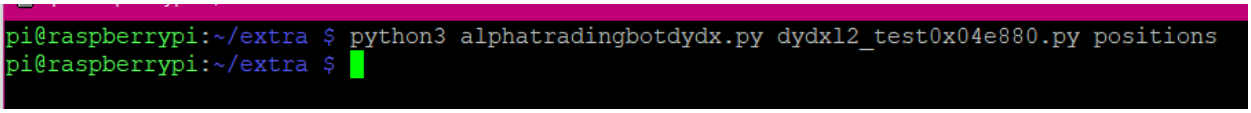
    d. Check the positions again and you will see 0.001 BTC.



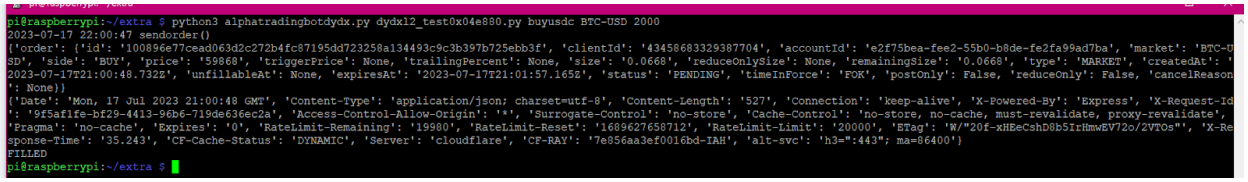
    e. ‘sellquantity’: works the same way like buyquantity.



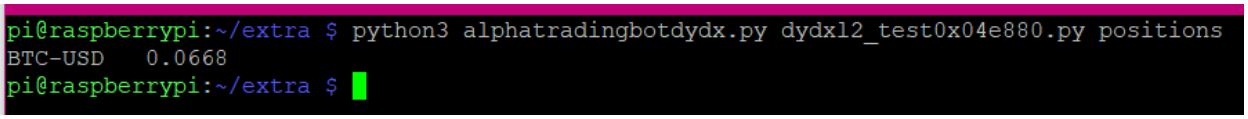
    f. Check the positions again and see you have closed the BTC position.



    g. ‘buyusdc’ and ‘sellusdc’: Buys and sells the specified USDC worth at market. These commands are generally used to open a new position because the crypto quantity depends on the price paid for each, whereas the other commands ‘buyquantity’ and ‘sellquantity’ are useful for closing an open position since you know the exact quantity.



h. Check the positions again and you will see 0.0668 BTC.



i. Let's close the position by selling 0.0668 BTC exactly. You can verify the position is closed.