이범석 류현우

https://github.com/oisooo/ai-cryptocurrency

```
import time
import requests
import pandas as pd
import datetime
import os
import sys


while True:
    try:

        book = {}
        book_eth = {}

        response = requests.get('https://api.bithumb.com/public/orderbook/BTC_KRW/?count=5')
        response_eth = requests.get('https://api.bithumb.com/public/orderbook/ETH_KRW/?count=5')

        response.raise_for_status()  # HTTP 요청 오류를 감지하고 예외를 발생시킵니다.
        response_eth.raise_for_status()

        book = response.json()
        book_eth = response_eth.json()

        data = book['data']
        data_eth = book_eth['data']

        bids = (pd.DataFrame(data['bids'])).apply(pd.to_numeric, errors='ignore')
        bids_eth = (pd.DataFrame(data_eth['bids'])).apply(pd.to_numeric, errors='ignore')

        bids.sort_values('price', ascending=False, inplace=True)
        bids_eth.sort_values('price', ascending=False, inplace=True)
        bids = bids.reset_index();
        bids_eth = bids_eth.reset_index();

        del bids['index']
        bids['type'] = 0
        del bids_eth['index']
        bids_eth['type'] = 0

        asks = (pd.DataFrame(data['asks'])).apply(pd.to_numeric, errors='ignore')
        asks_eth = (pd.DataFrame(data_eth['asks'])).apply(pd.to_numeric, errors='ignore')
        asks.sort_values('price', ascending=True, inplace=True)
        asks_eth.sort_values('price', ascending=True, inplace=True)
        asks['type'] = 1
        asks_eth['type'] = 1

        df = pd.concat([bids, asks])
        df_eth = pd.concat([bids_eth, asks_eth])

        timestamp = datetime.datetime.now()
        req_timestamp = timestamp.strftime('%Y-%m-%d %H:%M:%S')
        current_time = timestamp.strftime('%Y-%m-%d')

        df['quantity'] = df['quantity'].round(decimals=4)
        df['timestamp'] = req_timestamp
        df_eth['quantity'] = df_eth['quantity'].round(decimals=4)
        df_eth['timestamp'] = req_timestamp

        filename = "C:₩₩Users₩₩dlqja₩₩Documents₩₩카카오톡 받은 파일₩₩aicrypto₩₩book-%s-bithumb-btc.csv" %current_time
        filename_eth = "C:₩₩Users₩₩dlqja₩₩Documents₩₩카카오톡 받은 파일₩₩aicrypto₩₩book-%s-bithumb-eth.csv" % current_time

        should_write_header = os.path.exists(filename)
        if should_write_header == False:
            df.to_csv(filename, index=False, header=True, mode = 'a')
        else:
            df.to_csv(filename, index=False, header=False, mode = 'a')

        should_write_header_eth = os.path.exists(filename_eth)
        if should_write_header_eth == False:
            df_eth.to_csv(filename_eth, index=False, header=True, mode = 'a')
        else:
            df_eth.to_csv(filename_eth, index=False, header=False, mode = 'a')

    except requests.exceptions.RequestException as e:
        print("An error occurred:", e)
        print("Retrying in 1 seconds...")
        time.sleep(1)  # 1초 후에 재시도합니다.
        continue
    except Exception as e:
        print("An unexpected error occurred:", e)
        break  # 다른 예외가 발생한 경우 프로그램을 종료합니다.

    time.sleep(1)
```