

huobi_interface.py

```
# ===== Imports =====
# Library imports
import requests
import pydoc
import pandas as pd

# Local imports
from classes.interface_classes import Interface, DataStore, SymbolsManagerBase

# HuobiSDK imports
import huobi as hb
from huobi.client.market import MarketClient
from huobi.constant import *
from huobi.exception.huobi_api_exception import HuobiApiException
from huobi.model.market import *

# ===== Classes =====

class HuobiAPI(Interface):
    def __init__(self, access_key, secret_key, host="api.huobi.pro"):
        super().__init__(access_key, secret_key, host)
        self.__access_key = access_key
        self.__secret_key = secret_key
        self.__host = host

    def __get(self, path, params=None):
        return requests.get("https://{host}{path}".format(host=self.__host, path=path),
                             params=params).json()

    def get_symbols(self):
        return self.__get("/v1/common/symbols")

    #def get_candlestick(self, symbol, interval, size):
    #    return self.__get("/market/history/kline", {"symbol": symbol, "period": interval,
    #        "size": size})

    def get_kline_history(self, symbol, interval, limit):
        return self.__get("/market/history/kline", {"symbol": symbol, "period": interval,
            "size": limit})

    def subscribe_to_candlestick(self, symbol="btcusdt", interval="1min",
        callback_func=None):
        def callback(candlestick_event: 'CandlestickEvent'):
            candlestick_event.print_object()
            print("\n")

        def error(e: 'HuobiApiException'):
            print(e.error_code + e.error_message)

        market_client = MarketClient()
        if (callback_func == None):
            callback_func = callback
        market_client.sub_candlestick(symbol, interval, callback_func, error)

    def request_trades(self, symbol="btcusdt", callback_func=None):
        def callback(trade_req: 'TradeDetailReq'):
```

```
        return trade_req

    def error(e: 'HuobiApiException'):
        print(e.error_code + e.error_message)

    market_client = MarketClient()
    if (callback_func == None):
        callback_func = callback
    market_client.req_trade_detail(symbol, callback_func, error)


class HuobiSymbolsManager(SymbolsManagerBase):
    def __init__(self, interface: Interface):
        self.interface = interface

    def convert_to_dataframe(self, symbols: list):
        df = pd.DataFrame().from_dict(symbols)
        return df

    def filter_excluded(self, symbols: pd.DataFrame, excluded_coins: list = []):
        return super().filter_excluded(symbols, excluded_coins)

    def filter_offline(self, symbols: pd.DataFrame):
        return super().filter_offline(symbols)
```