import time

import hmac

import hashlib

import requests

import pandas as pd

from ta.trend import EMAIndicator, ADXIndicator

# ========== CONFIG ==========

API\_KEY = 'yqa47IvNwHjzP6VDlCfuyWW5LUfKVV'

API\_SECRET = 'fqTbbylycY6AsxjYis3oSVu141zarM4M6SfWKHhoSkVFX5WYtaEHsI28U8Dw'

BASE\_URL = 'https://api.delta.exchange'

TRADE\_SYMBOLS = ['BTCUSDT', 'ETHUSDT']

CAPITAL\_PER\_TRADE = 15

CHECK\_INTERVAL = 60 \* 15 # 15 minutes

# ========== AUTH ==========

def generate\_signature(method, path, query\_string=''):

ts = str(int(time.time() \* 1000))

msg = method + path + ts + query\_string

signature = hmac.new(API\_SECRET.encode(), msg.encode(), hashlib.sha256).hexdigest()

return ts, signature

# ========== FETCH CANDLES ==========

def fetch\_ohlcv(symbol, interval='15'):

url = f"{BASE\_URL}/v2/markets/{symbol}/candles?resolution={interval}&limit=100"

res = requests.get(url)

df = pd.DataFrame(res.json()['result'])

df['timestamp'] = pd.to\_datetime(df['time'], unit='s')

df.set\_index('timestamp', inplace=True)

return df[['open', 'high', 'low', 'close', 'volume']]

# ========== STRATEGY ==========

def generate\_signal(df):

df['ema20'] = EMAIndicator(df['close'], window=20).ema\_indicator()

df['ema50'] = EMAIndicator(df['close'], window=50).ema\_indicator()

df['adx'] = ADXIndicator(df['high'], df['low'], df['close'], window=14).adx()

last = df.iloc[-1]

if last['adx'] > 25:

if last['close'] > last['ema20'] > last['ema50']:

return 'BUY'

elif last['close'] < last['ema20'] < last['ema50']:

return 'SELL'

return None

# ========== PLACE ORDER ==========

def place\_market\_order(symbol, side):

endpoint = "/v2/orders"

ts, signature = generate\_signature("POST", endpoint)

headers = {

"api-key": API\_KEY,

"timestamp": ts,

"signature": signature,

"Content-Type": "application/json"

}

# Fetch price to calculate quantity

price\_data = requests.get(f"{BASE\_URL}/v2/tickers/{symbol}").json()

mark\_price = float(price\_data['result']['mark\_price'])

qty = round(CAPITAL\_PER\_TRADE / mark\_price, 4)

data = {

"order\_type": "market\_order",

"product\_id": get\_product\_id(symbol),

"side": side.lower(),

"size": qty

}

res = requests.post(BASE\_URL + endpoint, json=data, headers=headers)

print(f"[{symbol}] Order Response: {res.status\_code}, {res.json()}")

return res.json()

# ========== GET PRODUCT ID ==========

def get\_product\_id(symbol):

products = requests.get(f"{BASE\_URL}/v2/products").json()

for prod in products['result']:

if prod['symbol'] == symbol:

return prod['id']

return None

# ========== MAIN LOOP ==========

def run\_bot():

print("🚀 Starting Delta Exchange Bot (15-min interval)")

while True:

for symbol in TRADE\_SYMBOLS:

try:

df = fetch\_ohlcv(symbol)

signal = generate\_signal(df)

if signal:

print(f"[{symbol}] Signal: {signal}")

place\_market\_order(symbol, signal)

else:

print(f"[{symbol}] No trade signal at this time.")

except Exception as e:

print(f"[{symbol}] ERROR: {str(e)}")

print("⏱ Waiting 15 minutes...\n")

time.sleep(CHECK\_INTERVAL)

# ========== RUN ==========

if \_\_name\_\_ == "\_\_main\_\_":

run\_bot()