

스마트한 재테크를 위한 어플리케이션

멋사 파이썬 5기 홍태준

목차

01 프로그램 구성

02 소스코드

03 콘솔 출력

04 개선 사항

01

프로그램 구성

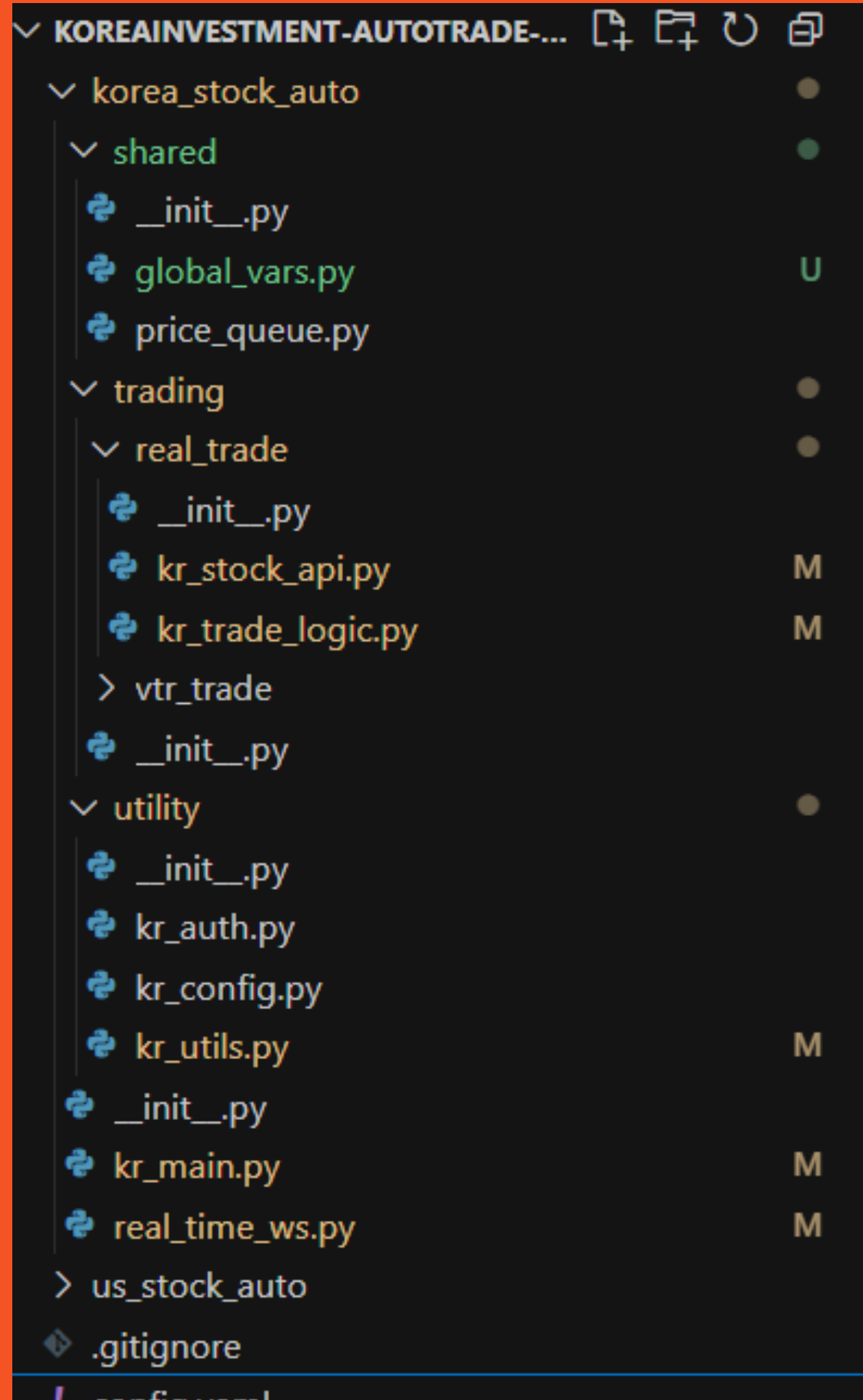
● 국내/해외 주식 통합 자동매매



02

소스코드

● 파일 경로: .utility



```
def get_access_token():
    global ACCESS_TOKEN, TOKEN_TIMESTAMP

    # 1분 이내에 요청된 경우 기존 토큰 사용
    if ACCESS_TOKEN and (time.time() - TOKEN_TIMESTAMP) < 60:
        return ACCESS_TOKEN

    headers = {'content-type': 'application/json'}
    body = {'grant_type': 'client_credentials', 'appkey': APP_KEY, 'appsecret': APP_SECRET}
    URL = f'{URL_BASE}/oauth2/tokenP'

    res = requests.post(URL, headers=headers, json=body)

    if res.status_code == 200:
        access_token = res.json().get("access_token")
        TOKEN_TIMESTAMP = time.time() # 토큰 발급 후 타임스탬프 업데이트
        ACCESS_TOKEN = access_token # 전역 변수에 저장
        print(f"✅ 액세스 토큰 발급 성공 ({'실전' if USE_REALTIME_API else '모의투자'}): {access_token}")
        return access_token
    else:
        print(f"❌ 액세스 토큰 발급 실패 ({'실전' if USE_REALTIME_API else '모의투자'}): {res.text}")
        return None
```

```
# API 설정(모의)
APP_KEY_VTS = _cfg["APP_KEY_VTS"]
APP_SECRET_VTS = _cfg["APP_SECRET_VTS"]
URL_BASE_VTS = _cfg["URL_BASE_VTS"]

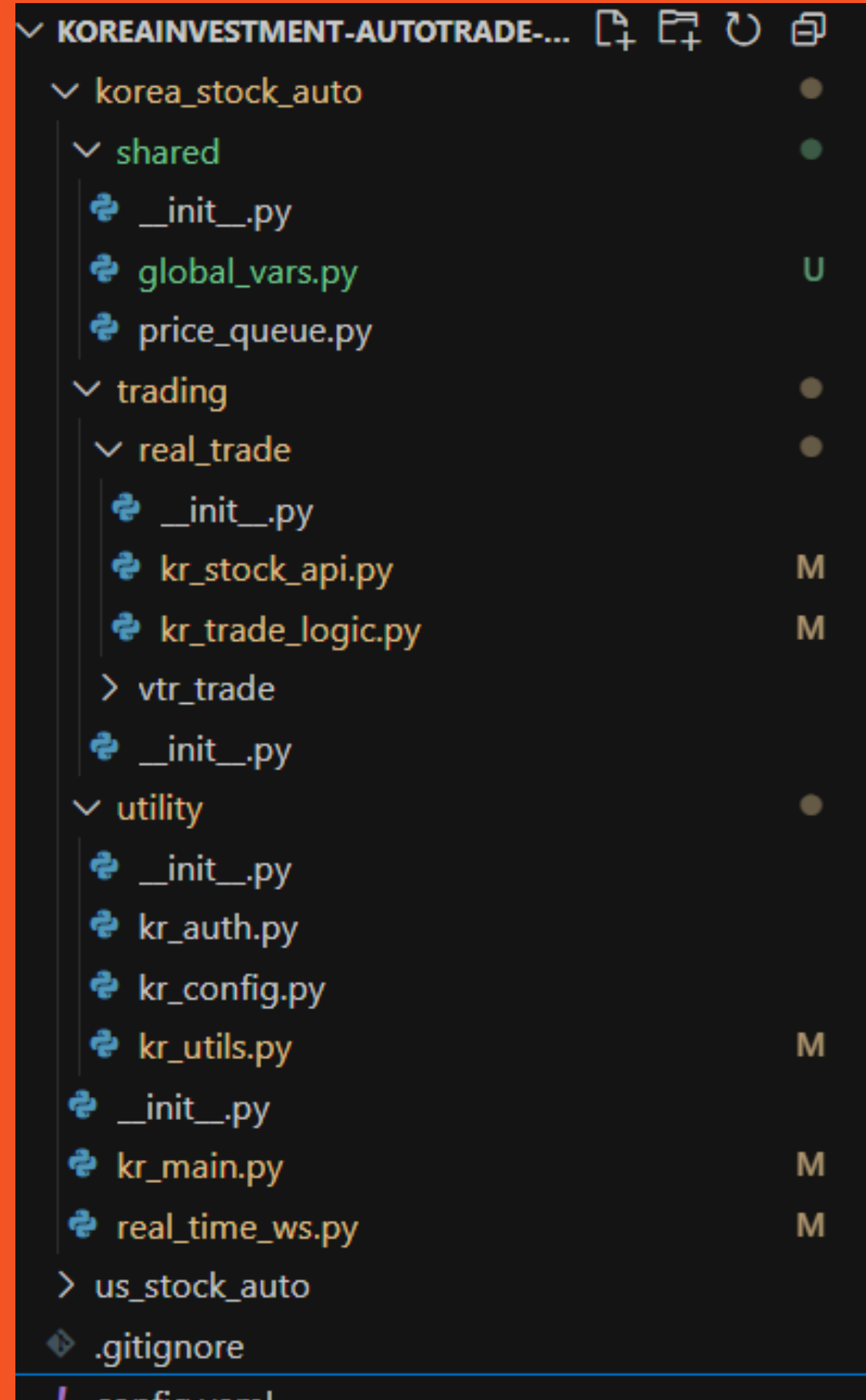
# 계좌 설정
CANO_REAL = _cfg["CANO_REAL"]
CANO_VTS = _cfg["CANO_VTS"]
ACNT_PRDT_CD = _cfg["ACNT_PRDT_CD"]

# 디스코드 웹훅
DISCORD_WEBHOOK_URL = _cfg["DISCORD_WEBHOOK_URL"]

# 웹소켓 실전/모의 분리
USE_REALTIME_API = True # 모의투자 기본값, 실전투자는 True로 변경

APP_KEY = APP_KEY_REAL if USE_REALTIME_API else APP_KEY_VTS
APP_SECRET = APP_SECRET_REAL if USE_REALTIME_API else APP_SECRET_VTS
URL_BASE = URL_BASE_REAL if USE_REALTIME_API else URL_BASE_VTS
```

• 파일 경로: real_time_ws



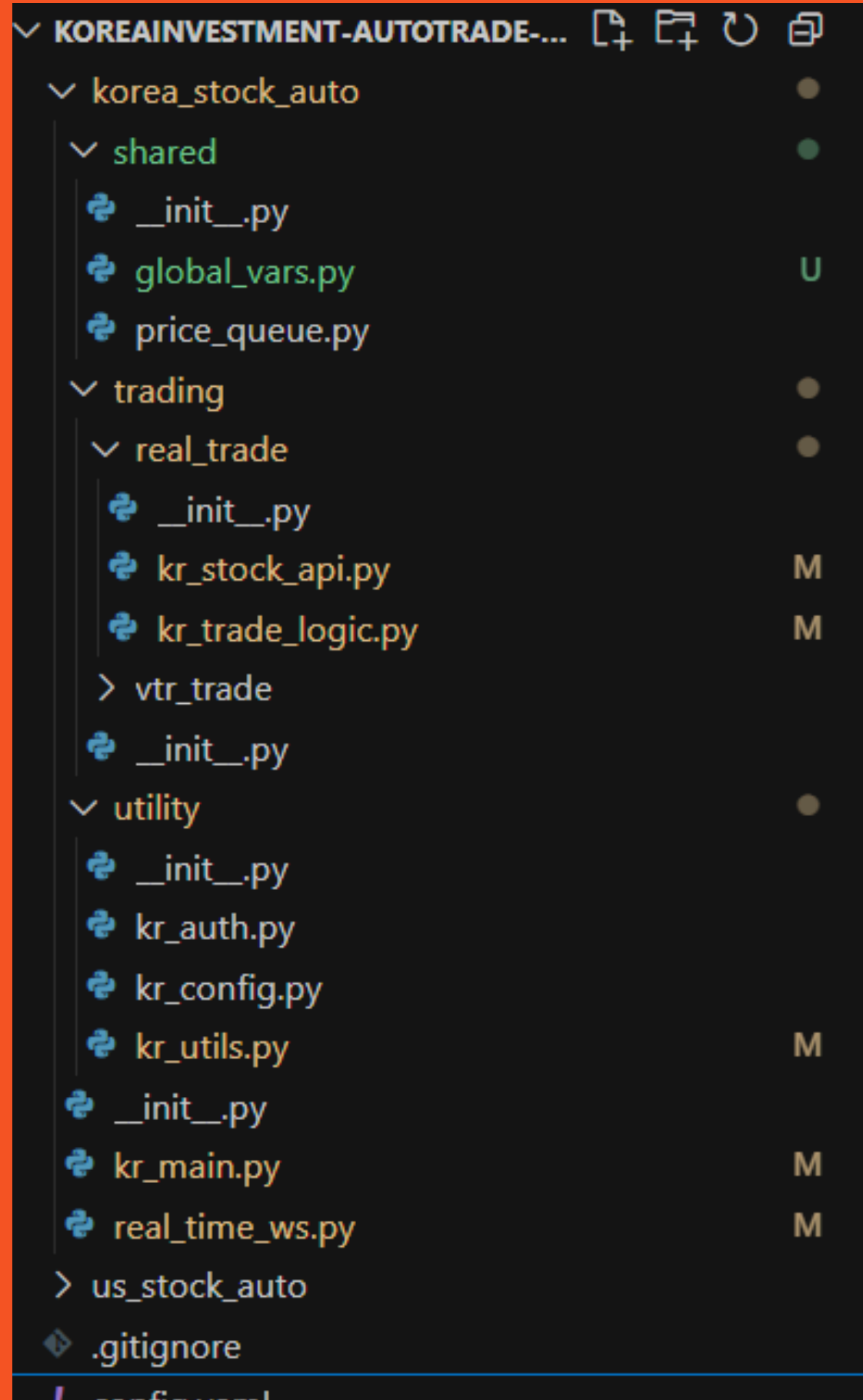
```
# WS_URL 수정: 실전 환경은 ops 도메인 사용
if USE_REALTIME_API:
    WS_URL = "wss://ops.koreainvestment.com:21000/tryitout/H0STCNT0"
else:
    WS_URL = "wss://ops.koreainvestment.com:31000/tryitout/H0STCNT0"

def request_ws_connection_key():
    if USE_REALTIME_API:
        approval_url = "https://openapi.koreainvestment.com:9443/oauth2/Approval"
        app_key = APP_KEY_REAL
        app_secret = APP_SECRET_REAL
    else:
        approval_url = "https://openapivts.koreainvestment.com:29443/oauth2/Approval"
        app_key = APP_KEY_VTS
        app_secret = APP_SECRET_VTS

    headers = {"Content-Type": "application/json"}
    body = {"grant_type": "client_credentials", "appkey": app_key, "secretkey": app_secret}
    send_message(f"WebSocket 접속키 요청 본문: {body}")
    res = requests.post(approval_url, headers=headers, json=body)
    if res.status_code == 200:
        ws_conn_key = res.json().get("approval_key")
        send_message(f"✅ WebSocket 접속키 발급 성공: {ws_conn_key}")
        return ws_conn_key
    else:
        send_message(f"❌ WebSocket 접속키 발급 실패 (HTTP {res.status_code}): {res.text}")
        return None

ACCESS_TOKEN = get_access_token()
WS_CONN_KEY = request_ws_connection_key()
```


• 파일 경로: .real_trade.kr_stock_api



```
# 국내 주식 매수 가능 조회
def get_stock_balance():
    URL = f"{URL_BASE_REAL}/uapi/domestic-stock/v1/trading/inquire-balance"
    headers = {
        "Content-Type": "application/json; charset=utf-8",
        "authorization": f"Bearer {ACCESS_TOKEN}",
        "appKey": APP_KEY_REAL,
        "appSecret": APP_SECRET_REAL,
        "tr_id": "TTTC8434R",
        "custtype": "P",
    }
    params = {
        'CANO': CANO, # 실전/모의 계좌번호 자동 변경
        'ACNT_PRDT_CD': ACNT_PRDT_CD,
        'PDNO': '', # 종목번호 (공란 입력 가능)
        'ORD_UNPR': '', # 주문 단가 (공란 가능)
        'ORD_DVSN': '01', # 주문 구분 (00: 지정가, 01: 시장가 등)
        'CMA_EVLU_AMT_ICLD_YN': 'N',
        'OVRN_ICLD_YN': 'N', # 해외 포함 여부
    }

    try:
        res = requests.get(URL, headers=headers, params=params)
        res_json = res.json()
    except Exception as e:
        send_message(f"❌ 주식 보유 잔고 API JSON 파싱 오류: {str(e)}")
        return {}

    stock_list = res_json.get('output1', [])
    evaluation = res_json.get('output2', [])
```

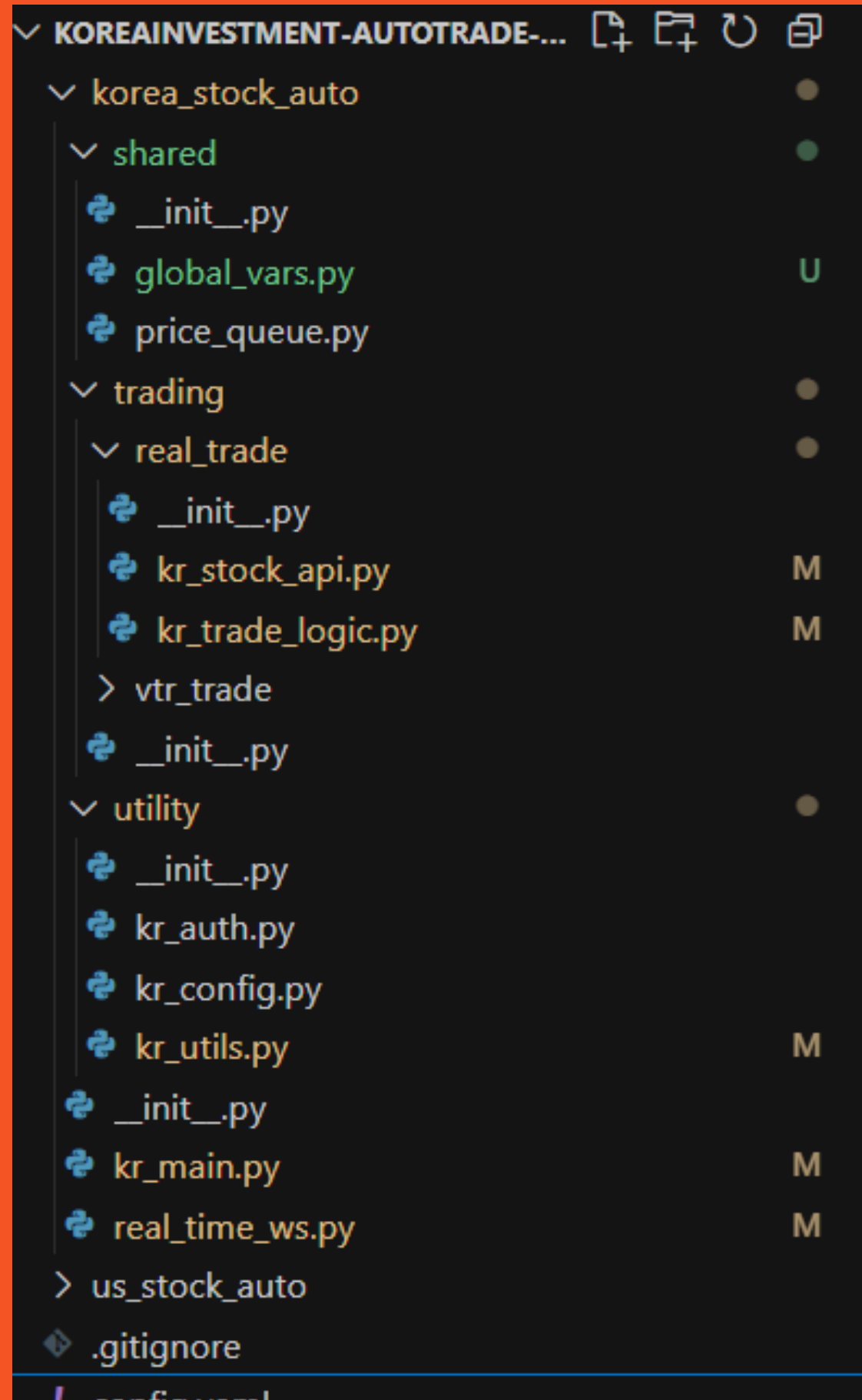
```
# 국내 주식 실시간 시세
def get_current_price(code):
    """현재가 조회"""
    while True:
        try:
            # 큐에서 시세 데이터 가져오기 (전용 모듈에서 가져온 price_queue 사용)
            received_code, price = price_queue.get(timeout=10) # 10초 대기
            if received_code == code:
                return price
        except queue.Empty:
            send_message("실시간 시세 데이터를 수신하지 못했습니다.")
            return None
```

```
# 종목 매수 주문
def buy_stock(code="005930", qty="1"):
    """주식 시장가 매수"""
    URL = f"{URL_BASE_REAL}/uapi/domestic-stock/v1/trading/order-cash"
    data = {
        "CANO": CANO_REAL,
        "ACNT_PRDT_CD": ACNT_PRDT_CD,
        "PDNO": code,
        "ORD_DVSN": "01",
        "ORD_QTY": str(int(qty)),
        "ORD_UNPR": "0",
    }
    headers = {
        "Content-Type": "application/json",
        "authorization": f"Bearer {ACCESS_TOKEN}",
        "appKey": APP_KEY_REAL,
        "appSecret": APP_SECRET_REAL,
        "tr_id": "TTTC0802U",
        "custtype": "P",
        "hashkey": hashkey(data),
    }

    try:
        res = requests.post(URL, headers=headers, data=json.dumps(data))
        res_json = res.json()
    except Exception as e:
        send_message(f"[매수 실패] JSON 파싱 오류: {str(e)}")
        return False

    if res_json.get("rt_cd") == "0":
        send_message(f"[매수 성공] {res_json}")
        return True
    else:
        send_message(f"[매수 실패] {res_json}")
        return False
```


• 파일 경로: .real_trade.kr_trade_logic.py



```
def select_interest_stocks():
    send_message("🇰🇷 관심 종목 선정 시작")
    top_stocks = get_top_traded_stocks()

    if not top_stocks:
        send_message("⚠️ 관심 종목 없음 (API 응답 실패 또는 데이터 없음)")
        return []

    # top_stocks에서 종목 코드와 이름의 매핑을 생성
    top_stock_names = { stock.get("mksc_shrn_iscd"): stock.get("hts_kor_isnm", "N/A") for stock in top_stocks }

    candidates = []
    SCORE_THRESHOLD = 2 # 최소 점수 기준

    for stock in top_stocks:
        code = stock.get("mksc_shrn_iscd")
        if not code:
            continue

        # 주식 기본 정보 조회 (종목 코드는 그대로 사용)
        info_data = get_stock_info(code)
        if not info_data:
            send_message(f"❌ {code} 상세 정보 없음")
            continue

        if isinstance(info_data, list):
            info = info_data[0]
        elif isinstance(info_data, dict):
            info = info_data
        else:
            send_message(f"❌ {code} 상세 정보 형식 오류")
            continue
```

03

콘솔 출력

● 콘솔 출력

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  TERMINAL  PORTS  COMMENTS
python + - [icon] [icon] ... ^ X

PS C:\Users\piome\Desktop\백엔드 부트캠프\koreainvestment-autotrader-main> python -m korea_stock_auto.kr_main
[icon] 액세스 토큰 발급 성공 (실전): eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0b2tlbiIsImF1ZCI6IjZmNzRmYmM0LWYzMTItNDQwOS1iOWJjLWRjNjA3MjA5OGYxMSIsInByZHRfY2QiOiIiLCJpc3MiOiJ1bm9ndyIsImV4CI6MTczOTIzMjM1MiwiYWFiOiJoxNzM5MTQ1OTUyLCJqdGkiOiJQU3lVWm9pd05yQnp3MXBnVkpBalVwVWVhGV2Q4dGIzOHVWN2QifQ.Bm4mYbS0frMuseuPA
CQOTbsmggcJnTj5fNywXTOhEtQ-k1JpbV827F6oe0fd-6sRr9ZHNBju7_2nHgd-j66B2Q
{'content': "[2025-02-10 11:40:44] WebSocket 접속키 요청 본문: {'grant_type': 'client_credentials', 'appkey': 'PSyUZoiwNrBzw1pgVJAjUpTXFwd8tb38uV7d', 'secretkey': 'vj4CzwC9QoRJSdzQ1vtxpFAYjvVIwk3xYyhvOm9TTC3ucj52D7wcQwvqPfhsisQo4WwKEB9s/Mk7fzuWb2z2dKNCFGtZuBBOS78ntZyZFeo/kiKX7v+hJWMRSwMmga+XwV/T7BijlR3QmQinzrWug1voGCUsfk2k7Rnixl4Vq4jeIhpGYpE='}"}
{'content': '[2025-02-10 11:40:45] [icon] WebSocket 접속키 발급 성공: 5a615077-3d17-4841-9cc9-abce8eb5e354'}
{'content': '[2025-02-10 11:40:46] 프로그램 시작'}
{'content': '[2025-02-10 11:40:48] [icon] 관심 종목 선정 시작'}
{'content': '[2025-02-10 11:40:49] [icon] API 요청 URL: https://openapi.koreainvestment.com:9443/uapi/domestic-stock/v1/quotations/volume-rank'}
{'content': '[2025-02-10 11:40:50] 관심종목 후보: KODEX 코스닥150선물인버스 (251340)'}
{'content': '[2025-02-10 11:40:52] 관심종목 후보: 피아이이 (452450)'}
{'content': '[2025-02-10 11:40:53] 관심종목 후보: 클로봇 (466100)'}
{'content': '[2025-02-10 11:40:54] 관심종목 후보: KODEX 코스닥150레버리지 (233740)'}
{'content': '[2025-02-10 11:40:56] 관심종목 후보: 디아이씨 (092200)'}
{'content': '[2025-02-10 11:40:57] 관심종목 후보: 한빛레이저 (452190)'}
{'content': '[2025-02-10 11:40:58] 관심종목 후보: 유진로봇 (056080)'}
{'content': '[2025-02-10 11:41:00] 관심종목 후보: 온코크로스 (382150)'}
{'content': '[2025-02-10 11:41:01] 관심종목 후보: 아스테라시스 (450950)'}
{'content': '[2025-02-10 11:41:02] 관심종목 후보: KODEX 인버스 (114800)'}
{'content': '[2025-02-10 11:41:04] 관심종목 후보: 피제이메탈 (128660)'}
{'content': '[2025-02-10 11:41:05] 관심종목 후보: KODEX 레버리지 (122630)'}

```

● 콘솔 출력

```
{ 'content': '[2025-02-10 11:41:30] 관심종목 후보: 삼성 인버스 2X 코스닥150 선물 ETN (Q530107)' }
{ 'content': '[2025-02-10 11:41:33] i 후보 탈락: KODEX 코스닥150선물인버스 (251340) - 점수: 1' }
{ 'content': '[2025-02-10 11:41:37] ✓ 후보 추가: 피아이이 (452450) - 점수: 2' }
{ 'content': '[2025-02-10 11:41:40] i 후보 탈락: 클로봇 (466100) - 점수: 1' }
{ 'content': '[2025-02-10 11:41:44] i 후보 탈락: KODEX 코스닥150레버리지 (233740) - 점수: 1' }
{ 'content': '[2025-02-10 11:41:47] ✓ 후보 추가: 디아이씨 (092200) - 점수: 3' }
{ 'content': '[2025-02-10 11:41:50] ✓ 후보 추가: 한빛레이저 (452190) - 점수: 2' }
{ 'content': '[2025-02-10 11:41:54] ✓ 후보 추가: 유진로봇 (056080) - 점수: 3' }
{ 'content': '[2025-02-10 11:41:57] i 후보 탈락: 온코크로스 (382150) - 점수: 1' }
{ 'content': '[2025-02-10 11:42:01] ✓ 후보 추가: 아스테라시스 (450950) - 점수: 2' }
{ 'content': '[2025-02-10 11:42:04] i 후보 탈락: KODEX 인버스 (114800) - 점수: 1' }
{ 'content': '[2025-02-10 11:42:08] ✓ 후보 추가: 피제이메탈 (128660) - 점수: 3' }
{ 'content': '[2025-02-10 11:42:11] i 후보 탈락: KODEX 레버리지 (122630) - 점수: 1' }
{ 'content': '[2025-02-10 11:42:15] i 후보 탈락: 삼성전자 (005930) - 점수: 1' }
{ 'content': '[2025-02-10 11:42:18] ✓ 후보 추가: 어보브반도체 (102120) - 점수: 2' }
{ 'content': '[2025-02-10 11:42:21] ✓ 후보 추가: 로보로보 (215100) - 점수: 3' }
```

04

개선 사항

• 에러코드 발생 없이 무한 루프 발생

```
✓ 관심 종목: ['092200', '056080', '128660', '215100', '006140', '264660', '452450', '452190', '450950', '102120', '316140', '005690']
{'content': "[2025-02-10 11:43:14] WebSocket 접속키 요청 본문: {'grant_type': 'client_credentials', 'appkey': 'PSyUZoiwNrBzw1pgVJAjUpTXFwd8tb38uV7d', 'secretkey': 'vj4CzwC9QoRJSdZQ1vtxpFAYjvVIWk3xYyhvOm9TTc3ucj52D7wcQwvqPfhsiSQo4wVKEB9s/Mk7fzuWb2z2dKNCFGtZuBBOS78ntZyZFeo/kiKX7v+hJWMRSwMnga+XwV/T7BijlR3QmQinzrWug1voGCUsfk2k7Rnixl4Vq4jeIhpGYpE='}"}
{'content': '[2025-02-10 11:43:16] ✓ WebSocket 접속키 발급 성공: cdf50018-f447-4f9c-9342-b0109ea2792d'}
{'content': '[2025-02-10 11:43:17] 🚀 자동 매매 시작'}
{'content': '[2025-02-10 11:43:17] ⚠️ 관심 종목 없음! 웹소켓 구독 요청 진행 중단.'}
{'content': '[2025-02-10 11:43:28] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:43:40] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:43:51] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:44:02] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:44:14] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:44:25] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:44:36] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:44:48] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:44:59] 실시간 시세 데이터를 수신하지 못했습니다.'}
{'content': '[2025-02-10 11:45:10] 실시간 시세 데이터를 수신하지 못했습니다.'}
```

- 관심 종목 선정 이후 자동 매매 로직에 문제가 있는 상황
- 로직 수정 후 실시간 매매 진행 예정
- 모의 투자의 경우 API를 지원하지 않는 경우가 많아 네이버 및 investing.com 에서 정보를 크롤링해 매매를 진행해야 하는 이슈가 있음
- 실시간 정보가 반영되지 않으므로 추가적인 로직 수정이 필요함

감사합니다