

서비스 연결

API 요청을 위해 “requests” 와 “json” 패키지가 필요합니다. “requests” 패키지는 HTTP 요청을 보낼 때, “json” 은 수신 받은 객체를 JSON 데이터로 만들어서 쓰기 위해 활용되는 패키지입니다. JSON 데이터는 pandas 데이터프레임으로 변환하기 쉬운 이점도 있습니다.

```
import requests
import json
```

requests는 크게 GET과 POST 방식이 있습니다. GET 방식은 현재가 혹은 주식 잔고 조회 같은 요청에서 쓰이고, POST 방식은 주로 주문/정정/취소 요청에서 쓰입니다. 아래 코드 셀은 GET 방식의 예제입니다. GET 방식의 주요 구성 요소는 URL, headers, 그리고 params입니다. 먼저 URL은 실전 또는 모의 투자에서 조회하고 싶은 요청의 경로 정보를 담고 있습니다. 다음으로 headers는 수신 데이터의 형태 및 요청자를 식별할 수 있는 인증 정보를 담고 있습니다. 마지막으로 params는 조회하고자 하는 시장(주식, ETF, ETN) 과 종목 코드 정보를 담고 있습니다. 각 구성 요소에서 어떤 구분자를 입력해야 하는지는 KIS Developers의 API 문서에 자세히 나와 있습니다. API 문서의 활용 방법은 주요 사용 함수를 정리하는 절에서 함께 들여다볼 계획입니다.

```
"""주식 종목 현재가 조회"""

# URL 설정
URL_BASE = "https://openapivts.koreainvestment.com:29443" # 모의 투자
PATH = "uapi/domestic-stock/v1/quotations/inquire-price" # 현재가 조회를 위한 URL 경로
URL = f"{URL_BASE}/{PATH}"
code = "005930" # 삼성전자 종목 코드

# headers 설정
headers = {
    "Content-Type": "application/json",
    "authorization": f"Bearer {ACCESS_TOKEN}", # 보안인증키
    "appKey": APP_KEY, # API 신청으로 발급 받은 Key
    "appSecret": APP_SECRET, # API 신청으로 발급 받은 Secret
    "tr_id": "FHKST01010100" # 현재가 조회를 위한 거래ID
}

# params 설정
params = {
    "fid_cond_mrkt_div_code": "J", # J: 주식
    "fid_input_iscd": code, # 조회 하고 싶은 주식 종목의 코드 ex) 삼성전자: 005930
}

# GET request 함수 호출
res = requests.get(URL, headers=headers, params=params)

print(int(res.json()['output']['stck_prpr']))
```

GET 방식과 마찬가지로 POST 방식에서도 URL에 주문하고자 하는 경로 값을 설정해 줍니다. 주문 관련 요청이 있기 때문에 headers에는 hashkey(암호화) 값을 추가하여 보안 수준을 높여 줍니다. 또한, 매수/매도를 구분하는 거래ID 값을 headers에 추가해 줍니다. 마지막으로 data에 주문 요청을 처리할 수 있는 계좌번호, 매수/매도 수량 그리고 주문가격 정보들을 담아 줍니다.

```

"""주식 시장가 매수"""

# URL 설정
URL_BASE = "https://openapivts.koreainvestment.com:29443" # 모의 투자
PATH = "uapi/domestic-stock/v1/trading/order-cash" # cash 주문
URL = f"{URL_BASE}/{PATH}"
code = "005930" # 삼성전자 종목 코드

# data 설정
data = {
    "CANO": CANO, # 계좌번호 앞자리
    "ACNT_PRDT_CD": ACNT_PRDT_CD, # 계좌번호 뒷자리
    "PDNO": code,
    "ORD_DVSN": "01", # 시장가
    "ORD_QTY": str(int(qty)), # 매수 주문 수량
    "ORD_UNPR": "0", # 시장가로 매수 시, ORD_UNPR 는 0 (지정가로 매수 시, ORD_UNPR 는 원
하는 지정가로 명시)
}

# headers 설정
headers = {"Content-Type": "application/json",
    "authorization": f"Bearer {ACCESS_TOKEN}", # 보안인증키
    "appKey": APP_KEY, # API 신청으로 발급 받은 Key
    "appSecret": APP_SECRET, # API 신청으로 발급 받은 Secret
    "tr_id": "VTTC0802U", # 매수 주문을 위한 거래ID
    "custtype": "P", # P: 개인
    "hashkey": hashkey(data) # <<<<<<<<
}

# POST request 함수 호출
res = requests.post(URL, headers=headers, data=json.dumps(data))

```

한편, 해쉬키 함수는 다음과 같이 정의 되어 있습니다.

```

def hashkey(datas):
    """암호화"""
    PATH = "uapi/hashkey"
    URL = f"{URL_BASE}/{PATH}"
    headers = {
        'content-Type': 'application/json',
        'appKey': APP_KEY,
        'appSecret': APP_SECRET,
    }
    res = requests.post(URL, headers=headers, data=json.dumps(datas))
    hashkey = res.json()["HASH"]
    return hashkey

```

위 예제 코드의 GET 과 POST 방식 모두 headers에서 ACCESS_TOKEN 을 필요로 하는데, get_access_token() 함수에서 APP_KEY 와 APP_SECRET를 이용해서 발급받을 수 있습니다.

```

def get_access_token():
    """토큰 발급"""
    headers = {"content-type": "application/json"}
    body = {"grant_type": "client_credentials",
        "appkey": APP_KEY,
        "appsecret": APP_SECRET}
    PATH = "oauth2/tokenP"
    URL = f"{URL_BASE}/{PATH}"
    res = requests.post(URL, headers=headers, data=json.dumps(body))
    ACCESS_TOKEN = res.json()["access_token"]
    return ACCESS_TOKEN

```

끝으로 headers와 data 설정에서 반복적으로 사용되는 개인 정보들은 config.yaml 파일에 일괄적으로 저장해 두면 더 간편하게 개인정보를 관리할 수 있습니다.

"""config.yaml 파일 생성"""

#홈페이지에서 API서비스 신청시 받은 Appkey, Appsecret 값 설정

```
APP_KEY: "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
```

[illegible]

#계좌번호 앞 8자리

CANO: "xxxxxxxx"

#계좌번호 뒤 2자리

ACNT_PRDT_CD: "01"

#실전투자

```
# URL_BASE: "https://openapi.koreainvestment.com:9443"
```

#모의투자

URL_BASE: "https://openapivts.koreainvestment.com:29443"

#디스코드 웹훅 URL

[illegible]