2024.05.10

8-1 , 8-2, 8-3 오전

에어플로우 키기

workspace에 py만들기

import logging,subprocess

subprocess

시스템 명령어 가져온다

out = subprocess.run("echo 'hi' > /root/a.txt", shell=True)

logger.info(f'{out}')

실행하는 법

venvjotaesik@Playdata:~/workspace$ python logging_test.py

import datetime,time

print(datetime.datetime.utcfromtimestamp(time.time()).strftime('%Y-%m-%d %H:%M:%S'))

datetime.datetime.utcnow()

datetime.datetime.utcnow()datetime.timedelta(hours=9)
datetime.datetime.utcnow()+datetime.timedelta(hours=9)

import logging,subprocess, argparse 추가

argparse argument를 전달할때 파싱을 잘해주는

parser = argparse.ArgumentParser()

parser.add_argument('-d', '--day',type=str,nargs="+",metavar="N")

args=parser.parse_args()

logger.info(f'사용자가 입력한 날짜 -- {args}')

2024-05-10 14:44:18,025 - root - INFO - 사용자가 입력한 날짜 -- Namespace(day=['2024-05-10'])

python manage.py runserver

런서버를 args로 받아서 manage.py를 돌린다.

네이버주식 차트 가져오기

import logging, subprocess, argparse

import requests

로그 생성

logger = logging.getLogger()

logger.setLevel(logging.INFO)

formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')

console

s_handler = logging.StreamHandler()

s_handler.setFormatter(formatter)

logger.addHandler(s_handler)

file

f_handler = logging.FileHandler("./system.log")

f_handler.setFormatter(formatter)

logger.addHandler(f_handler)

out = subprocess.run("mysql -uroot customer > ./backup.sql", shell=True)

logger.info(f'{out}')

for i in range(10):

logger.info(f"순서 {i} ")

parser = argparse.ArgumentParser()

parser.add_argument('-s', '--start', type=str, nargs="+", metavar='yyyy-mm-dd', help="날짜선택")

parser.add_argument('-e', '--end', type=str, nargs="+", metavar='yyyy-mm-dd', help="날짜선택")

url = "https://m.stock.naver.com/front-api/external/chart/domestic/info?symbol=015760&requestType=1&startTime={}&endTime={}&timeframe=day"

args = parser.parse_args()

start = args.start

end = args.end

data = eval(requests.get(url).text.strip())

logger.info(f'사용자가 입력한 날짜 - {args}')

logger.info(f'사용자가 입력한 날짜 - {start} - {end}')

data = eval(requests.get(url.format(start[0], end[0])).text.strip())

logger.info(f'받아온 데이터 개수 - {len(data)}')

```
venvjotaesik@Playdata:~/workspace$ python logging_test.py -s 20240101 -e 20240509
2024-05-10 15:15:42,742 - root - INFO - 사용자가 입력한 날짜 - Namespace(start=['20240101'], end=['20240509'])
2024-05-10 15:15:42,742 - root - INFO - 사용자가 입력한 날짜 - ['20240101'] - ['20240509']
2024-05-10 15:15:42,826 - root - INFO - 받아온 데이터 개수 - 88
```

한국거래소 주식종목과 네이버 주식 연결

python logging_test.py -s 20240501 -e 20240509

```
import logging, subprocess, argparse

import requests, csv, os

from multiprocessing import Pool

from itertools import repeat

import time

url = "https://m.stock.naver.com/front-api/external/chart/domestic/info?symbol={}&requestType=1&startTime={}&endTime={}&timeframe=day"

logger = logging.getLogger()

def get*stock(code*, start*, end*):

    code = *code*.replace("", "")

    logger.info(f"{code*} - {start*} - {end_}")

    #logger .info(f"{url.format(code*, start*, end_)}")

    if os.path.isdir("./stock") == False:

        os.mkdir("./stock")

    data = eval(requests.get(url.format(code*, start*, end_)).text.strip())

    with open(f"./stock/{code_}.csv", "w", encoding='utf-8') as f:

        writer = csv.writer(f)

        for row in data:

            writer.writerow(row)

if **name** == "**main**":

    # 로그 생성

    logger.setLevel(logging.INFO)

    formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
```

```
    # console
    s_handler = logging.StreamHandler()
    s_handler.setFormatter(formatter)
    logger.addHandler(s_handler)
    # file
    f_handler = logging.FileHandler("./system.log")
    f_handler.setFormatter(formatter)
    logger.addHandler(f_handler)
    parser = argparse.ArgumentParser()
    parser.add_argument('-s', '--start',  type=str, nargs="+", metavar='yyyymmdd', help="날짜선택")
    parser.add_argument('-e', '--end',  type=str, nargs="+", metavar='yyyymmdd', help="날짜선택")
    # 종목코드 읽기
    with open('./data_2805_20240510.csv', 'r',
              encoding='cp949') as f:
        code = [(x.split(',')[1]) for x in f.readlines()]
    args = parser.parse_args()
    start = args.start
    end = args.end
    logger.info(f'사용자가 입력한 날짜 - {start} - {end}')
    start_time = time.time()
    with Pool(processes=6) as pool:
        #병렬처리해줘 파라미터는 3개야
        pool.starmap(get_stock,zip(code[1:], repeat(start[0]), repeat(end[0])) )
```

logger.info(f"{time.time()- start_time}")

파이썬

which python

**

pwd

vim batch.sh

#!/bin/bash

/home/gen/workspace/django/venv/bin/python /home/gen/workspace/logging_test.py -s 20240501 -e 20240509

**

/home/jotaesik/workspace/django/venv/bin/python /home/jotaesik/workspace/logging_test.py -s 20240501 -e 20240509

크론탭

venvjotaesik@Playdata:~/workspace$ crontab -e

실행권한걸기

venvjotaesik@Playdata:~/workspace$ chmod +x ./batch.sh

실행

. ./batch.sh

crontab -e

맨마지막줄에

```
30 16 * * * /home/jotaesik/workspace/batch.sh
```

크론상태확인

venvjotaesik@Playdata:~/workspace$ sudo service cron status
venvjotaesik@Playdata:~/workspace$ sudo service cron start

    with open('/home/jotaesik/workspace/data_2805_20240510.csv', 'r',
절대경로로
**

watch -n 1 "ps -ef | grep python"

watch -n 1 date

**
jotaesik@Playdata:~/workspace$ rm -rf stock
jotaesik@Playdata:~/workspace$ tail -f system.log

최종코드

import logging, subprocess, argparse

import requests, csv, os

from multiprocessing import Pool

from itertools import repeat

import time

url = "https://m.stock.naver.com/front-api/external/chart/domestic/info?symbol={}&requestType=1&startTime={}&endTime={}&timeframe=day"

logger = logging.getLogger()

stock_url=""

def get*stock(code*, start*, end*):

    code = *code*.replace("", "")

    logger.info(f"{code*}* - *{start*}* - {end_}")

    #logger .info(f"{url.format(code*, start*, end_)}")

    if os.path.isdir("/home/jotaesik/workspace/stock") == False:

        os.mkdir("/home/jotaesik/workspace/stock")

    data = eval(requests.get(url.format(code*, start*, end_)).text.strip())

```
    with open(f"/home/jotaesik/workspace/stock/{code_}.csv", "w", encoding='utf-8') as f:
        writer = csv.writer(f)
        for row in data:
            writer.writerow(row)
if name == "main":
    # 로그 생성
    logger.setLevel(logging.INFO)
    formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
    # console
    s_handler = logging.StreamHandler()
    s_handler.setFormatter(formatter)
    logger.addHandler(s_handler)
    # file
    f_handler = logging.FileHandler("./system.log")
    f_handler.setFormatter(formatter)
    logger.addHandler(f_handler)
    parser = argparse.ArgumentParser()
    parser.add_argument('-s', '--start',  type=str, nargs="+", metavar='yyyymmdd', help="날짜선택")
    parser.add_argument('-e', '--end',  type=str, nargs="+", metavar='yyyymmdd', help="날짜선택")
    # 종목코드 읽기
    with open('/home/jotaesik/workspace/data_2805_20240510.csv', 'r',
              encoding='cp949') as f:
        code = [(x.split(',')[1]) for x in f.readlines()]
```

```
    args = parser.parse_args()
    start = args.start
    end = args.end
    logger.info(f'사용자가 입력한 날짜 - {start} - {end}')
    start_time = time.time()
    with Pool(processes=6) as pool:
        #병렬처리해줘 파라미터는 3개야
        pool.starmap(get_stock,zip(code[1:], repeat(start[0]), repeat(end[0])) )
    logger.info(f"{time.time()- start_time}")
```

최종 코드 안될시

```
import logging, subprocess, argparse
import requests, csv, os
from multiprocessing import Pool
from itertools import repeat
import time
url = "https://m.stock.naver.com/front-api/external/chart/domestic/info?symbol={}&requestType=1&startTime={}&endTime={}&timeframe=day"
logger = logging.getLogger()
stock_url=""
def getstock(code, start, end):
    code = code.replace("", "")
    logger.info(f"{code} - {start} - {end_}")
    #logger.info(f"{url.format(code, start, end_)}")
    if os.path.isdir("/home/jotaesik/workspace/stock") == False:
        os.mkdir("/home/jotaesik/workspace/stock")
```

```
    data = eval(requests.get(url.format(code, start, end_)).text.strip())
    with open(f"/home/jotaesik/workspace/stock/{code_}.csv", "w", encoding='utf-8') as f:
        writer = csv.writer(f)
        for row in data:
            writer.writerow(row)
if name == "main":
    # 로그 생성
    logger.setLevel(logging.INFO)
    formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
    # console
    s_handler = logging.StreamHandler()
    s_handler.setFormatter(formatter)
    logger.addHandler(s_handler)
    # file
    f_handler = logging.FileHandler("./system.log")
    f_handler.setFormatter(formatter)
    logger.addHandler(f_handler)
    parser = argparse.ArgumentParser()
    parser.add_argument('-s', '--start',  type=str, nargs="+", metavar='yyyymmdd', help="날짜선택")
    parser.add_argument('-e', '--end',  type=str, nargs="+", metavar='yyyymmdd', help="날짜선택")
    # 종목코드 읽기
    with open('/home/jotaesik/workspace/data_2805_20240510.csv', 'r',
              encoding='cp949') as f:
```

```
    code = [(x.split(',')[1]) for x in f.readlines()]
args = parser.parse_args()
start = args.start
end = args.end
logger.info(f'사용자가 입력한 날짜 - {start} - {end}')
start_time = time.time()
with Pool(processes=6) as pool:
    #병렬처리해줘 파라미터는 3개야
    pool.starmap(get_stock,zip(code[1:], repeat(start[0]), repeat(end[0])) )
logger.info(f"{time.time()- start_time}")
```