

일일 과제 명세서 - 동작코드

파이선 API 데이터 크롤러 구현해보기

목차

1. 과제 개요.....	3
2. 과제 목표.....	4
3. 동작 코드	5

1. 과제 개요

본 과제는 Python 의 fastAPI 와 MongoDB 를 활용한 간단한 Python API 서버 및 , Web API data Crawler 를 만들어보는 실습과제 입니다. Python 의 API 개발 기초와 NoSQL 사용 기초, Command 사용 기초를 전달하는 것이 주 목적입니다.

2. 과제 목표

- 1) Python 을 설치한다.
- 2) Python FastAPI 를 세팅한다.
- 3) MongoDB 를 설치한다.
- 4) MongoDB 에 조회할 기본 데이터 (상품명) 를 입력한다.
- 5) 네이버 쇼핑 API 에 개발자 등록을 하고 사용키를 취득한다.
- 6) Python FastAPI 에서 MongoDB 의 기본데이터 (상품명) 를 조회한다.
- 7) 조회한 데이터로 네이버 쇼핑 API 를 호출하여 네이버 스토어에서 판매중인 정보를 조회한다.
- 8) 조회한 데이터를 본인이 설치한 MongoDB 에 저장한다.

3. 동작 코드

1. 과제 관련 코드

2) Python FastAPI framework 구성 및 RESTful API 개발, CMD 동작 개발

Req. 2-1	FastAPI framework 구성
기능 상세	<p>FastAPI 로 RESTful(main.py 파일) 코드를 구현한다. (아래 샘플 코드 참고)</p> <pre> from fastapi import FastAPI app = FastAPI() @app.get("/") async def root(): return {"message": "Hello World"} </pre>
Req. 2-2	FastAPI command 구성
기능 상세	<p>main.py 파일에 FastAPI 활용하여 RESTful 호출시 Command 입력되는 코드를 구현한다. (아래 샘플 코드 참고)</p> <p>os.popen 을 사용하는 방법 과 subprocess.run 을 사용하는 방법이 있다.</p> <pre> import os import subprocess from fastapi import FastAPI app = FastAPI() @app.get("/") async def root(): return {"message": "Hello World"} @app.get("/test/checkSystemPythonVersion/os") def read_version(): # version = os.system('python --version') stream = os.popen('python --version') output = stream.read() print(output) return output @app.get("/test/checkSystemPythonVersion/subprocess") def read_version(): </pre>

	<pre> result = subprocess.run(["python", "--version"], stdout=subprocess.PIPE, text=True) print(result.stdout) return result.stdout </pre>
--	--

6) Python FastAPI 와 MongdoDB 연동

Req. 3-1	FastAPI 와 MongoDB 연동
기능 상세	<p>FastAPI 와 pymongo 라이브러리를 활용하여 RESTFul(main.py 파일) 코드를 구현한다. (아래 샘플 코드 참고) XXXXXX.. 와 YYYYYY.. 는 신청한 정상적인 키로 변경해야 한다.</p> <pre> import pymongo import os import subprocess from fastapi import FastAPI from pymongo import MongoClient app = FastAPI() #naver config dbaddrs = 'localhost' dbport = 27017 headers = {"X-Naver-Client-Id": "XXXXXXXXXXXXXXXXXXXXXXXXXXXX", "X-Naver-Client-Secret": "YYYYYYYYYYYYYYYYYYYY"} searchTargetCount = 100 naverUrl = 'https://openapi.naver.com/v1/search/shop.json?display='+str(sea rchTargetCount)+'&query=' @app.get("/") async def read_root(): return {"message": "Hello World"} @app.get("/test/{keyword}") def read_keyword(keyword: str): return {"keyword": keyword} @app.get("/test/checkSystemPythonVersion/os") def read_version(): # version = os.system('python --version') stream = os.popen('python --version') </pre>

```

        output = stream.read()
        print(output)
        return output

@app.get("/test/checkSystemPythonVersion/subprocess")
def read_version():
    result = subprocess.run(["python", "--version"],
stdout=subprocess.PIPE, text=True)
    print(result.stdout)
    return result.stdout

@app.get("/test/dbtest")
def process():
    dt_now = datetime.datetime.now()
    print(dt_now)
    # 2022-12-13 15:06:22.574657

    conn = MongoClient(dbaddrs,dbport)
    #print(conn)
    #print(conn.list_database_names())

    #db = conn.local
    #col = db.startup_log
    database = conn.prd
    collection = database.prod
    scol = database.prodDateSummary

    # cursur
    #seldata = col.find();

    # convert list Type
    dataList = list(collection.find())

    ## for 문을 사용하여 추가하기
    for data in dataList:
        searchKey = data['productObj']['prd_id']
        print('=====')
        print(searchKey)

        # GET
        res = requests.get(naverUrl + searchKey, headers =
headers)
        #print(str(res.status_code) + " | " + res.text)

        #print(type(res.text))

```

```

resJson = json.loads(res.text)
#print(type(resJson))

#print(resJson['lastBuildDate'])
#print(resJson['total'])
#print(resJson['start'])
#print(resJson['display'])
#print(resJson['items'])

searchResultList = resJson['items']

for sr in searchResultList:
    #print(sr['mallName'] + ' // ' + sr['brand'] + ' :: ' + sr['lprice'])

    serchRes = {
        "date":dt_now,
        "name":searchKey,
        "title":sr['title'],
        "link":sr['link'],
        "image":sr['image'],
        "lprice":sr['lprice'],
        "hprice":sr['hprice'],
        "mallName":sr['mallName'],
        "productId":sr['productId'],
        "productType":sr['productType'],
        "brand":sr['brand'],
        "maker":sr['maker'],
        "category1":sr['category1'],
        "category2":sr['category2'],
        "category3":sr['category3'],
        "category4":sr['category4']
    }

    scol.insert_one(serchRes)

print()

return {"result": "success"}

```