# 26.Python Scrapy3

# 1. Python 고급 Crawling 기술  OPEN API

1. Naver Open API Project  개요
   - Naver Open API 이용 Crawling 하는 Program

2. **scrapy Project 생성**

1) C:₩Users₩ktg₩python-env>  # Directory 이동 만들기

2) virtScrapy₩Scripts₩activate.bat  → (virtScrapy) C:₩Users₩ktg₩python-env>  # 가상 환경 전환됨

# Scrapy1에서 만들었다면 Skip

3) (virtScrapy) C:₩Users₩ktg₩python-env>pip install scrapy  # 가상 환경 에서 scrapy Framework  설치하기

4) (virtScrapy) C:₩Users₩ktg₩python-env>cd C:₩PyCharmProject₩Sources  # 가상 환경 에서 만들 Project로 이동

5) (virtScrapy) C:₩PyCharmProject₩Sources>scrapy startproject ktgNopenApi  # 가상 환경 에서 . openApi Project 만들기

6) spider (크롤러 이름) 생성 방법

- 크롤링 프로젝트 내에, 여러 가지 크롤러(scrapy에서는 spider라고 함) 있을 수 있으므로, 각 크롤러의 이름을 지정

[1] scrapy genspider 이용

① (virtScrapy) C:₩PyCharmProject₩Sources>cd C:₩PyCharmProject₩Sources₩ktgNopenApi₩ktgNopenApi

② (virtScrapy) C:₩PyCharmProject₩Sources₩ktgNopenApi₩ktgNopenApi>

scrapy genspider naverShopping https://openapi.naver.com/v1/search/shop.json?query

[2] pycharm 으로 작성

① ktgNopenApi₩ktgNopenApi/spiders 디 렉토리에 naverShopping .py 파일(템플릿)이 생김

② 직접 scrapy genspider 명령을 사용하지 않고, 만들어도 됨

# 2. Python 고급 Crawling 기술  OPEN API Project 생성2

3. spider (크롤러) 실행

1 (virtScrapy) C:₩PyCharmProject₩Sources₩ktgNopenApi₩ktgNopenApi>scrapy crawl naverShopping

2) 다양한 데이터 포멧으로 아이템 저장하기(naverShopping 예시)

① csv, xml, json 포멧

② 터미널 환경에서, ecommerce 폴더에서 다음 명령

scrapy crawl 크롤러명 -o 저장할파일명 -t 저장포멧

(virtScrapy) C:₩PyCharmProject₩Sources₩ktgNopenApi₩ktgNopenApi>

scrapy crawl naverShopping  -o naverShopping.csv -t csv

(virtScrapy) C:₩PyCharmProject₩Sources₩ktgNopenApi₩ktgNopenApi>

scrapy crawl naverShopping -o naverShopping.xml -t xml

# json 파일을 확인하면, 한글문자가 깨져나옴

(virtScrapy) C:₩PyCharmProject₩Sources₩ktgNopenApi₩ktgNopenApi>

scrapy crawl naverShopping  -o naverShopping.json -t json

# 3. Python 고급 Crawling 기술  OPEN API 기본 구조 Setting

1. settings.py

```
# Obey robots.txt  규칙을 따르겠는가(윤리문제 ?)
# robots.txt  선 점검
ROBOTSTXT_OBEY = False
 # Field 순서 지정
FIELD_EXPORT_FIELDS=['title', 'link', 'image', 'lprice',
                     'hprice' , 'mallName', 'productId', 'productType'
                     ]

 # Configure maximum concurrent requests performed by Scrapy (default: 16)
CONCURRENT_REQUESTS = 1
 # 한글 Setting
FEED_EXPORT_ENCODING = 'utf-8'


# Log 해당
LOG_FILE="log.txt“


 # Data 후처리를 위해 PIPELINES 적용
ITEM_PIPELINES = {
   'ktgNopenApi.pipelines.KtgnopenapiPipeline': 300,
}
```

# 4. Python 고급 Crawling 기술  OPEN API 적용예시

3. NavershoppingSpider 예시

1) spider

```
import scrapy
import json
from ktgNopenApi.items import KtgNopenApiItem

class NavershoppingSpider(scrapy.Spider):
    name = 'naverShopping'

    def start_requests(self):
        start_url = 'https://openapi.naver.com/v1/search/shop.json?query='
        client_id = 'j513HictYVPoWEl27Yor'
        client_secret = 'I0m7fwb6Cn'
        header_params = {'X-Naver-Client-Id':client_id, 'X-Naver-Client-Secret':client_secret}
        query = '삼성'
        for index in range(10):
            # start 1~100 , 101~200 , ....901~1000
            yield scrapy.Request(url=start_url + query + '&display=100&start=' + str(index*100 + 1),
                                 headers=header_params)
    # callback 함수 없을 때 Default로 들어옴
    def parse(self, response):
        data = json.loads(response.body_as_unicode())
        for item in data['items']:
            nItem = KtgNopenApiItem()
            nItem['title'] = item['title']
            nItem['link'] = item['link']
            nItem['image'] = item['image']
            nItem['lprice'] = item['lprice']
            nItem['hprice'] = item['hprice']
            nItem['mallName'] = item['mallName']
            nItem['productId'] = item['productId']
            nItem['productType'] = item['productType']
            yield nItem
```

3)  items.py  예시 문장

```
import scrapy


class KtgNopenApiItem(scrapy.Item):
    # define the fields for your item here like:
    title = scrapy.Field()
    link = scrapy.Field()
    image = scrapy.Field()
    lprice = scrapy.Field()
    hprice = scrapy.Field()
    mallName = scrapy.Field()
    productId = scrapy.Field()
```