4. Naver & 공공DB API

Naver API 사용

공공DB API 사용

1. Naver API

- Client ID와 Client Secret 설정
  - 비인증 API 방식은 Request 헤더에 ID와 Secret를 함께 전송하여 REST API 사용

```
#[CODE 1]
def get_request_url(url):

    req = urllib.request.Request(url)
    req.add_header("X-Naver-Client-Id", client_id)
    req.add_header("X-Naver-Client-Secret", client_secret)
    이하 생략
```

Query 매개변수

```
#[CODE 2]
def getNaverSearchResult(sNode, search_text, page_start, display):

    base = "https://openapi.naver.com/v1/search"
    node = "/%s.json" % sNode
    parameters = "?query=%s&start=%s&display=%s" %
(urllib.parse.quote(search_text), page_start, display)
    #parameters = "?query=%s" % urllib.parse.quote(search_text)
    url = base + node + parameters
```

query	검색어 파라미터(UTF-8 인코딩 형식)
start	검색의 시작점(최대값 1,000)
display	1회 검색에 가지고 올 데이터 레코드 수(최대 100)
sort	sim(유사도순: 기본값), date(날짜순)

JSON 데이터 추출 -저장

- JSON 형식 수신데이터는 ['items'] 내부에 레코드 형태로 존재, 제목(title), 내용(description), 링크(link)를 공통으로 가지며 검색 영역(뉴스, 블로그, 까페 등)에 따라 고유한 네임을 가짐

```
#[CODE 3]
def getPostData(post, jsonResult):
    title = post['title']
    description = post['description']
    org_link = post['originallink']
    link = post['link']

    #Tue, 14 Feb 2017 18:46:00 +0900
    pDate = datetime.datetime.strptime(post['pubDate'],  '%a, %d %b %Y %H:%M:%S +0900')
    pDate = pDate.strftime('%Y-%m-%d %H:%M:%S')

    jsonResult.append({'title':title, 'description': description,
                       'org_link':org_link, 'link': org_link, 'pDate':pDate})
    return
```

1. Naver API 사용(검색)

```
import urllib.request
import datetime
import time
import json

#[CODE 1]
def get_request_url(url):

    req = urllib.request.Request(url)
    req.add_header("X-Naver-Client-Id", client_id)
    req.add_header("X-Naver-Client-Secret", client_secret)
    try:
        response = urllib.request.urlopen(req)
        if response.getcode() == 200:
            print ("[%s] Url Request Success" % datetime.datetime.now())
            return response.read().decode('utf-8')
    except Exception as e:
        print(e)
        print("[%s] Error for URL : %s" % (datetime.datetime.now(), url))
        return None
```

```
#[CODE 2]
def getNaverSearchResult(sNode, search_text, page_start, display):

    base = "https://openapi.naver.com/v1/search"
    node = "/%s.json" % sNode
    parameters = "?query=%s&start=%s&display=%s"
            %(urllib.parse.quote(search_text), page_start, display)  parse.quote = 16진수로 인코딩
    #parameters = "?query=%s" % urllib.parse.quote(search_text)
    url = base + node + parameters

    retData = get_request_url(url)

    if (retData == None):
        return None
    else:
        return json.loads(retData)
```

```
#[CODE 3]
def getPostData(post, jsonResult):

    title = post['title']
    description = post['description']
    org_link = post['originallink']
    link = post['link']

    #Tue, 14 Feb 2017 18:46:00 +0900
    pDate = datetime.datetime.strptime(post['pubDate'],
                            '%a, %d %b %Y %H:%M:%S +0900')
    pDate = pDate.strftime('%Y-%m-%d %H:%M:%S')

    jsonResult.append({'title':title, 'description': description,
                       'org_link':org_link, 'link': org_link,
                       'pDate':pDate})
    return
```

```
def main():
    jsonResult = []
    # 'news', 'blog', 'cafearticle'
    sNode = 'news'
    search_text = '코로나'
    display_count = 100

    jsonSearch = getNaverSearchResult(sNode, search_text, 1, display_count)

    while ((jsonSearch != None) and (jsonSearch['display'] != 0)):
        for post in jsonSearch['items']:
            getPostData(post, jsonResult)

        nStart = jsonSearch['start'] + jsonSearch['display']
        jsonSearch = getNaverSearchResult(sNode, search_text, nStart, display_count)

    with open('%s_naver_%s.json' % (search_text, sNode), 'w', encoding='utf8') as outfile:
        retJson = json.dumps(jsonResult,
                             indent=4, sort_keys=True,
                             ensure_ascii=False)
        outfile.write(retJson)

    print ('%s_naver_%s.json SAVED' % (search_text, sNode))

if __name__ == '__main__':
    main()
```

네이버 API(파파고-번역)

- Papago NMT API
  - 네이버 Papago에 적용된 번역 REST API, 입력된 텍스트를 다른 나라 언어(영어, 중국어)로 번역한 텍스트로 출력해주는 REST API
- Papago NMT API 예제 코드
  - https://developers.naver.com/docs/nmt/examples/
- Papago NMT API Reference
  - https://developers.naver.com/docs/nmt/reference/

네이버 API(파파고-번역)

```
import urllib.request
client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET”

encText = urllib.parse.quote("번역할 문장을 입력하세요")
data = "source=ko&target=en&text=" + encText
url = https://openapi.naver.com/v1/papago/n2mt

request = urllib.request.Request(url)

request.add_header("X-Naver-Client-Id",client_id)
request.add_header("X-Naver-Client-Secret",client_secret)
response = urllib.request.urlopen(request, data=data.encode("utf-8"))
rescode = response.getcode()

if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)
```

2. 공공DB API-

```
import urllib.request
import datetime
import time
import json
import matplotlib.pyplot as plt
import matplotlib
from matplotlib import font_manager, rc

def get_request_url(url):
    req = urllib.request.Request(url)

    try:
        response = urllib.request.urlopen(req)
        if response.getcode() == 200:
            print ("[%s] Url Request Success" % datetime.datetime.now())
            return response.read().decode('utf-8')
    except Exception as e:
        print(e)
        print("[%s] Error for URL : %s" % (datetime.datetime.now(), url))
        return None
```

```
#[CODE 1]
def getNatVisitor(yyyymm, nat_cd, ed_cd):
    access_key="인증키"
    end_point =
"http://openapi.tour.go.kr/openapi/service/EdrcntTourismStatsService/getEdrcnt
TourismStatsList"

    parameters = "?_type=json&serviceKey=" + access_key
    parameters += "&YM=" + yyyymm
    parameters += "&NAT_CD=" + nat_cd
    parameters += "&ED_CD=" + ed_cd

    url = end_point + parameters

    retData = get_request_url(url)

    if (retData == None):
        return None
    else:
        return json.loads(retData)
```

```
def main():
    jsonResult = []
    #중국: 112 / 일본: 130 / 미국: 275
    national_code = "275"
    ed_cd = "E"

    nStartYear = 2011
    nEndYear = 2017

    for year in range(nStartYear, nEndYear):
        for month in range(1, 13):
            yyyymm = "{0}{1:0>2}".format(str(year), str(month))
            jsonData = getNatVisitor(yyyymm, national_code, ed_cd)

            print (json.dumps(jsonData,indent=4, sort_keys=True,ensure_ascii=False))

            if (jsonData['response']['header']['resultMsg'] == 'OK'):
                krName = jsonData['response']['body']['items']['item']["natKorNm"]
                krName = krName.replace(' ', '')
                iTotalVisit = jsonData['response']['body']['items']['item']["num"]
                print('%s_%s : %s' %(krName, yyyymm, iTotalVisit))
                jsonResult.append({'nat_name': krName, 'nat_cd': national_code,
                                   'yyyymm': yyyymm, 'visit_cnt': iTotalVisit})
```

```
cnVisit = []
VisitYM = []
index = []
i = 0
for item in jsonResult:
    index.append(i)
    cnVisit.append(item['visit_cnt'])
    VisitYM.append(item['yyyymm'])
    i = i + 1

with open('%s(%s)_해외방문객정보_%d_%d.json'
          % (krName, national_code, nStartYear, nEndYear-1), 'w', encoding='utf8') as outfile:
    retJson = json.dumps(jsonResult,indent=4, sort_keys=True, ensure_ascii=False)
    outfile.write(retJson)

#[CODE 2]
font_location = "c:/Windows/fonts/malgun.ttf"
font_name = font_manager.FontProperties(fname=font_location).get_name()
matplotlib.rc('font', family=font_name)

plt.xticks(index, VisitYM)
plt.plot(index, cnVisit)
plt.xlabel('방문월')
plt.ylabel('방문객수')
plt.grid(True)
plt.show()
```

공공DB API CSV 이용

- **데이터 다운로드 : https://goo.gl/oJydAv**
- **시각화 도구 설치**
  - Plotnine 설치 : pip install plotnine
  - missingno 설치 : pip install missingno
  - pip show plotnine
  - pip show missingno

```
import warnings
warnings.filterwarnings('ignore') #warning 데이터가 출력되지 않도록
```

```
import pandas as pd
import numpy as np
import re # 정규표현식
from plotnine import *
```

```
!pip show plotnine #모듈 설치 확인
```

```
# 현재 위치정보를 봅니다 %를 사용하면 터미널에 사용할수 있는 명령어를 사용할수 있음
%pwd
```

```
pre_sale=pd.read_csv('d:/data/go/전국_평균_분양가격_2018.7월_.csv', encoding='utf-8')
pre_sale.shape
```

```
pre_sale.head()
```

```
pre_sale.head()
```

```
pre_sale.info() #데이터 프레임에 요약 확인
```

```
pre_sale.dtypes
```

```
pre_sale.isnull().sum() # 결측치의 개수 계산
```

```
# 결측치 보기
import missingno as msno
msno.matrix(pre_sale, figsize=(18,6))
```

```
# 연도와 월은 카테고리 형태의 데이터이기 때문에 스트링 형태로 변경
pre_sale['연도']=pre_sale['연도'].astype(str)
pre_sale['월']=pre_sale['월'].astype(str)
pre_sale.dtypes
```

```
pre_sale_price=pre_sale['분양가격(㎡)']
#분양가격을 숫자로 변경
pre_sale['분양가격']=pd.to_numeric(pre_sale_price,errors='coece')
#평당 분양가격 구하기
pre_sale['평당분양가격']=pre_sale['분양가격']*3.3
```

```
pre_sale.info()
```

```
pre_sale.dtypes
```

```
pre_sale.isnull().sum()
```

```
pre_sale.describe()
```

```
pre_sale.describe(include=[np.object])
```

```
#2017년 데이터만 보기
pre_sale_2017=pre_sale.loc[pre_sale['연도']=='2017']
pre_sale_2017.shape
```

```
# 같은 값을 갖고 있는걸로 시도별로 동일하게 데이터가 들어있는 것을 확인
pre_sale['규모구분'].value_counts()
```

```
pre_sale['지역명'].value_counts()
```