**변수 List**

**DataCrawling.py**

**#코로나 백신접종 센터 데이터 크롤링 함수(공공기관/ 위탁의료기관)**

client\_id – 경로에 접근하기 위한 아이디 (naver 발급)

client\_secret – 경로에 접근하기 위한 비밀번호 (naver 발급)

response – URL을 통해 데이터를 요청해서 받은 결과

GetGoVSearchResult함수에서의 호출 예

resultData = Get\_Request\_Url(url)

client\_id = "5l5yadvenl"

client\_secret = "ygf36whrwrx4XQGFoGaepijGIrvEdHWIvgCgnmF8"

baseUrl = "https://naveropenapi.apigw.ntruss.com/map-geocode/v2/geocode"

*def* Get\_Request\_Url(*url*):

    req = urllib.request.Request(*url*) #검색 URL 경로 지정

    req.add\_header("X-NCP-APIGW-API-KEY-ID", client\_id)

req.add\_header("X-NCP-APIGW-API-KEY", client\_secret)

    try:

        response = urllib.request.urlopen(req)

if response.getcode() == 200: # 성공코드 : 200

            print("[%s] Url 요청 성공 : " % datetime.datetime.now())

            return(response.read().decode('utf-8'))

    except Exception as ex: # 예외처리

        print(ex)

        print("[%s] 오류 : %s " % datetime.datetime.now(), *url*)

        return None

**#사전 준비 작업 및 분류(크롤링을 위한 url 생성)**

resultData – Get\_Request\_Url 함수에서 크롤링해온 데이터(공공기관)

url – url 생성

Main에서의 호출 예

jsonSearchResult = GetGoVSearchResult(endPoint, pageData, perPageData, keyValue)

*def* GetGoVSearchResult(*endPoint*, *pageValue*, *perPageValue*, *keyValue*) :

    paraData = "page=" + str(*pageValue*)

    paraData += "&perPage=" + str(*perPageValue*)

    # paraData에 문자형식으로 pageValue, perPageValue 저장

    url = *endPoint* + paraData + *keyValue* # url 생성

    resultData = Get\_Request\_Url(url)

    # resultData = Get\_Request\_Url 함수에서 크롤링해온 데이터

    if(resultData == None):

        return None

    else:

        return json.loads(resultData)

    # resultData파일 json파일 형식으로 리턴

**#위도/경도 값 출력을 위한 사전 준비 작업**

resulturl – url 생성

resultData – Get\_Request\_Url 함수에서 크롤링해온 데이터(위탁기관, 위도ㆍ경도 데이터 없음)

DataCleaning함수에서의 호출 예

centerLocation = GetLngLatData(GetGeoLocationData(addrStr))

*def* GetGeoLocationData(*addr*):

        paraData = "?query=%s" % urllib.parse.quote(*addr*)

        resulturl = baseUrl + paraData

        resultData = Get\_Request\_Url(resulturl)

        if(resultData == None):

            return None

        else:

            return json.loads(resultData)

**DataCleaning.py**

**#백신접종센터 데이터 정제 함수 (주소/센터명/전화번호/위도/경도 추출)**

resultData - 백신접종센터 JSON 데이터

addr - JSON 파일의 주소 변수 명

name - JSON 파일의 센터명 변수 명

num - JSON 파일의 전화번호 변수 명

lat - JSON 파일의 위도 변수 명

lng - JSON 파일의 경도 변수 명

city - 정제할 지역명

Main에서의 함수 호출 예

GetCenterData(jsonSearchResult, "address","centerName","phoneNumber","lat","lng","서울특별시")

*def* GetCenterData(*resultData*, *addr*, *name*, *num*, *lat*, *lng*, *city*):

    # json 파일의 data 키값 데이터를 DataFrame 형태로 저장

    tempResult = pd.DataFrame(*resultData*['data'])

    # 컬럼 확인 후 입력한 지역명의 데이터만 저장

    centerResult = tempResult[tempResult[str(*addr*)].str.contains(*city*)]

    # 주소/센터명/번호/위도/경도 값만 추출하여 DataFrame 형태로 저장

    centerAddr = centerResult[str(*addr*)].values

    centerName = centerResult[str(*name*)].values

    centerNum = centerResult[str(*num*)].values

    if(*lat*!=None):

        centerlng = centerResult[*lng*].values

        centerlat = centerResult[*lat*].values

    else:

        centerlng = []

        centerlat = []

        for addrStr in centerAddr :

            centerLocation = GetLngLatData(GetGeoLocationData(addrStr))

            centerlat.append(centerLocation[0])

            centerlng.append(centerLocation[1])

        #centerlat= None

        #centerlng=None

    return [centerAddr, centerName, centerNum,centerlat,centerlng]

**#민간병원 데이터 정제 함수 (위도/경도 추출)**

LocationData - 위치 JSON 데이터

Main에서의 함수 호출 예

DataCleaning.GetLngLatData(LocationData)

*def* GetLngLatData(*LocationData*):

        # 정제 - JSON 파일 분석 후 데이터 찾기 - 위도 / 경도 값

        xdata = *LocationData*['addresses'][0]['x']

        ydata = *LocationData*['addresses'][0]['y']

        return [ydata, xdata]

**DataCleaning.py (멀티프로세싱.ver)**

**#백신접종센터 데이터 정제 함수 (주소/센터명/전화번호/위도/경도 추출)**

import pandas as pd

from DataCrawling import GetGeoLocationData

import multiprocessing

*def* GetCenterData(*resultData*, *addr*, *name*, *num*, *lat*, *lng*, *city*):

    # json 파일의 data 키값 데이터를 DataFrame 형태로 저장

    tempResult = pd.DataFrame(*resultData*['data'])

    # 컬럼 확인 후 입력한 지역명의 데이터만 저장

    centerResult = tempResult[tempResult[str(*addr*)].str.contains(*city*)]

    # 주소/센터명/번호/위도/경도 값만 추출하여 DataFrame 형태로 저장

    centerAddr = centerResult[str(*addr*)].values

    centerName = centerResult[str(*name*)].values

    centerNum = centerResult[str(*num*)].values

    if(*lat*!=None):

        centerlng = centerResult[*lng*].values

        centerlat = centerResult[*lat*].values

        return [centerAddr, centerName, centerNum,centerlat,centerlng]

    else:

        centerlng = []

        centerlat = []

        pool=multiprocessing.Pool(*processes*=multiprocessing.cpu\_count())

        centerData=pool.map(mtPr,centerAddr)

        return [centerAddr, centerName, centerNum, centerData]

*def* mtPr(*centerAddr*):

    centerData=GetGeoLocationData(*centerAddr*)

    centerLocation=GetLngLatData(centerData)

    centerlng=centerLocation[0]

    centerlat=centerLocation[1]

    return [centerlng,centerlat]

*def* GetLngLatData(*LocationData*):

        # 정제 - JSON 파일 분석 후 데이터 찾기 - 위도 / 경도 값

        xdata = *LocationData*['addresses'][0]['x']

        ydata = *LocationData*['addresses'][0]['y']

        return [ydata, xdata]

**main\_input.py**

**#메인 함수(병원명을 입력받아서 지도에 위치 표시)**

jsonSearchResult – 공공기관 예방접종센터

jsonSearchResult2 – 사설기관 예방접종센터

from DataCrawling import \*

from DataCleaning import \*

import pandas as pd

import folium

*def* Main():

    keyValue = "&serviceKey=SVAIOlt%2FwZYCMlBnNwQY0KF1QgscYyBeOzrRxwAlFGGNXwxR4I5vGO4LNfv7VvkPb%2B%2BI6q0Rk26GaRQzOI1wew%3D%3D"

    endPoint = "https://api.odcloud.kr/api/15077586/v1/centers?"

    endPoint2 = "https://api.odcloud.kr/api/apnmOrg/v1/list?"

    pageData = 0

    perPageData = 248

    perPageData2 = 10000

    jsonSearchResult = GetGoVSearchResult(endPoint, pageData, perPageData, keyValue)

    jsonSearchResult2 = GetGoVSearchResult(endPoint2, pageData, perPageData2, keyValue)

    #"서울특별시"에 있는 센터주소,센터명,센터전화번호를 가져옴

    addr, name, num, lat, lng= GetCenterData(jsonSearchResult,

"address","centerName","phoneNumber","lat","lng","서울특별시")

    addr2, name2, num2, location = GetCenterData(jsonSearchResult2,

"orgZipaddr","orgnm","orgTlno",None,None,"서울특별시")

    # 센터주소, 센터명, 센터전화번호 데이터를 이용하여 데이터프레임 생성

    jsonCleaningData = pd.DataFrame(*data*=list(zip(addr, name, num, lat, lng)),

*columns* = ['Addr', 'name', 'num','lat','lng'])

    jsonCleaningData2 = pd.DataFrame(*data*=list(zip(addr2, name2, num2, location)),

*columns* = ['Addr', 'name', 'num','location'])

    print(jsonCleaningData, jsonCleaningData2)

    # 사용자 입력 받기

    userInput= input("병원 이름을 입력하세요 : ")

    # 공공 기관인 경우

    if("코로나19" in userInput) :

        # 센터명을 포함하는 행 추출

        centerResult = jsonCleaningData[jsonCleaningData['name'].str.contains(userInput)]

        # 위도/경도 추출

        centerLat = centerResult['lat'].values

        centerLng = centerResult['lng'].values

        centerLocation = [centerLat, centerLng]

    else: # 민간 기관인 경우

        centerResult = jsonCleaningData2[jsonCleaningData2['name'].str.contains(userInput)]

        centerLocation = centerResult['location'].values

        centerLocation = centerLocation[0]

    # 좌표 찍기

    map\_data = folium.Map(*location*=centerLocation, *zoom\_start*=15)

    map\_data = folium.Marker(centerLocation,

*tooltip*=str(centerResult['name'].values)).add\_to(map\_data)

    map\_data.save(*r*'.\googlemap\_input.html')

    print("맵 표시 완료")

if \_\_name\_\_ == '\_\_main\_\_':

    Main()

**main.py (모듈프로젝트1 기존 과업)**

**#메인 함수(전국의 데이터 수집**ㆍ**정제 후 서울에 있는 백신 접종센터만 지도에 표시)**

jsonSearchResult – 공공기관 예방접종센터의 데이터

jsonSearchResult2 – 위탁기관 예방접종센터의 데이터

map\_data – 지정한 위도ㆍ경도 근처의 지도를 나타냄

*def* Main():

    keyValue = "&serviceKey=SVAIOlt%2FwZYCMlBnNwQY0KF1QgscYyBeOzrRxwAlFGGNXwxR4I5vGO4LNfv7VvkPb%2B%2BI6q0Rk26GaRQzOI1wew%3D%3D"

    endPoint = "https://api.odcloud.kr/api/15077586/v1/centers?"

    endPoint2 = "https://api.odcloud.kr/api/apnmOrg/v1/list?"

    pageData = 1

    perPageData = 284 # 공공기관

    perPageData2 = 10000 # 위탁기관

    jsonSearchResult = GetGoVSearchResult(endPoint, pageData, perPageData, keyValue)

    jsonSearchResult2 = GetGoVSearchResult(endPoint2, pageData, perPageData2, keyValue)

    #"서울특별시"에 있는 센터주소,센터명,센터전화번호를 가져옴

    addr, name, num, lat, lng= GetCenterData(jsonSearchResult,

"address","centerName","phoneNumber","lat","lng","서울특별시")

    addr2, name2, num2, lat2, lng2 = GetCenterData(jsonSearchResult2,

"orgZipaddr","orgnm","orgTlno",None,None,"서울특별시")

    # 센터주소, 센터명, 센터전화번호 데이터를 이용하여 데이터프레임 생성

    jsonCleaningData = pd.DataFrame(*data*=list(zip(addr, name, num, lat, lng)),

*columns* = ['Addr', 'name', 'num','lat','lng'])

    jsonCleaningData2 = pd.DataFrame(*data*=list(zip(addr2, name2, num2, lat2, lng2)),

*columns* = ['Addr', 'name', 'num','lat','lng'])

    map\_data = folium.Map([37.56595045963169, 126.98918361888224],*zoom\_start*=12)

    for i in range(0, len(jsonCleaningData)):

      Location=[jsonCleaningData['lat'][i],jsonCleaningData['lng'][i]]

      marker= folium.Marker(Location,*color*='blue',*popup*=jsonCleaningData['num'][i],

*tooltip*=jsonCleaningData['name'][i],*icon*=folium.Icon(*color*="green"))

      marker.add\_to(map\_data)

    for i in range(0, len(jsonCleaningData2)):

      Location=[jsonCleaningData2['lat'][i],jsonCleaningData2['lng'][i]]

      marker= folium.Marker(Location, *popup*=jsonCleaningData2['num'][i],

*tooltip*=jsonCleaningData2['name'][i],*icon*=folium.Icon(*color*="red"))

      marker.add\_to(map\_data)

    map\_data.save(*r*'c:\CovidVaccineMap.github.io\결과물\navermap4.html')

    print("[%s] 저장 성공 : " % datetime.datetime.now())

if \_\_name\_\_ == '\_\_main\_\_':

    Main()

**main.py (멀티프로세싱.ver)**

**#메인 함수(전국의 데이터 수집**ㆍ**정제 후 서울에 있는 백신 접종센터만 지도에 표시)**

from DataCrawling import \*

from DataCleaning import \*

import pandas as pd

import folium

import time

*def* Main():

    print("[%s] 저장 성공 : " % datetime.datetime.now())

    keyValue = "&serviceKey=SVAIOlt%2FwZYCMlBnNwQY0KF1QgscYyBeOzrRxwAlFGGNXwxR4I5vGO4LNfv7VvkPb%2B%2BI6q0Rk26GaRQzOI1wew%3D%3D"

    endPoint = "https://api.odcloud.kr/api/15077586/v1/centers?"

    endPoint2 = "https://api.odcloud.kr/api/apnmOrg/v1/list?"

    pageData = 1

    perPageData = 284

    perPageData2 = 10000

    jsonSearchResult = GetGoVSearchResult(endPoint, pageData, perPageData, keyValue)

    jsonSearchResult2 = GetGoVSearchResult(endPoint2, pageData, perPageData2, keyValue)

    #"서울특별시"에 있는 센터주소,센터명,센터전화번호를 가져옴

    addr, name, num, lat, lng= GetCenterData(jsonSearchResult,

"address","centerName","phoneNumber","lat","lng","서울특별시")

    addr2, name2, num2, location = GetCenterData(jsonSearchResult2,

"orgZipaddr","orgnm","orgTlno",None,None,"서울특별시")

    # 센터주소, 센터명, 센터전화번호 데이터를 이용하여 데이터프레임 생성

    jsonCleaningData = pd.DataFrame(*data*=list(zip(addr, name, num, lat, lng)),

*columns* = ['Addr', 'name', 'num','lat','lng'])

    jsonCleaningData2 = pd.DataFrame(*data*=list(zip(addr2, name2, num2, location)),

*columns* = ['Addr', 'name', 'num','location'])

    map\_data = folium.Map([37.56595045963169, 126.98918361888224],*zoom\_start*=12)

    for i in range(0, len(jsonCleaningData)):

      Location=[jsonCleaningData['lat'][i],jsonCleaningData['lng'][i]]

      marker= folium.Marker(Location,*popup*=jsonCleaningData['num'][i],

*tooltip*=jsonCleaningData['name'][i],*icon*=folium.Icon(*color*="green"))

      marker.add\_to(map\_data)

    for i in range(0, len(jsonCleaningData2)):

      Location=[jsonCleaningData2['location'][i]]

      marker= folium.Marker(Location[0],*popup*=jsonCleaningData2['num'][i],

*tooltip*=jsonCleaningData2['name'][i],*icon*=folium.Icon(*color*="red"))

      marker.add\_to(map\_data)

    map\_data.save(*r*'c:\CovidVaccineMap.github.io\결과물\navermap4.html')

    print("[%s] 저장 성공 : " % datetime.datetime.now())

if \_\_name\_\_ == '\_\_main\_\_':

    Main()