

## 05. 웹 API 서비스 활용

# contents

---

- ▶ 공공 데이터 수집
- ▶ 네이버 **API** 활용

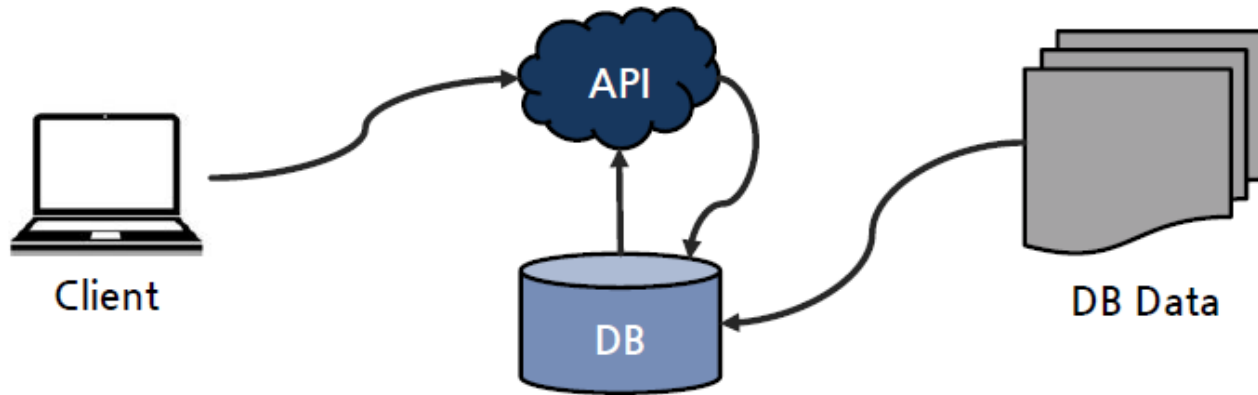


# 1. 공공 데이터 수집

## ▶ Open API vs Rest API

API

- Application Programming Interface의 약자
- 특정 프로그램을 만들기 위해 제공되는 모듈(함수 등)



# 1. 공공 데이터 수집

---

## ▶ Open API vs Rest API

### Open API

- 누구나 사용할 수 있도록 공개된 API
- 주로 Rest API 기술을 많이 사용

### Rest API

- Representational State Transfer API의 약자
- HTTP 프로토콜을 통해서 정보를 제공하는 함수
- 실질적인 API 사용은 정해진 구조의 URL 문자열 사용



일반적으로 XML, JSON의 형태로 응답 출력



# 1. 공공 데이터 수집

## ▶ 웹 API

- ▶ 웹 API는 일반적으로 HTTP 통신을 사용하는데 사용
- ▶ 지도, 검색, 주가, 환율 등 다양한 정보를 가지고 있는 웹 사이트의 기능을 외부에서 쉽게 사용할 수 있도록 사용 절차와 규약을 정의한 것

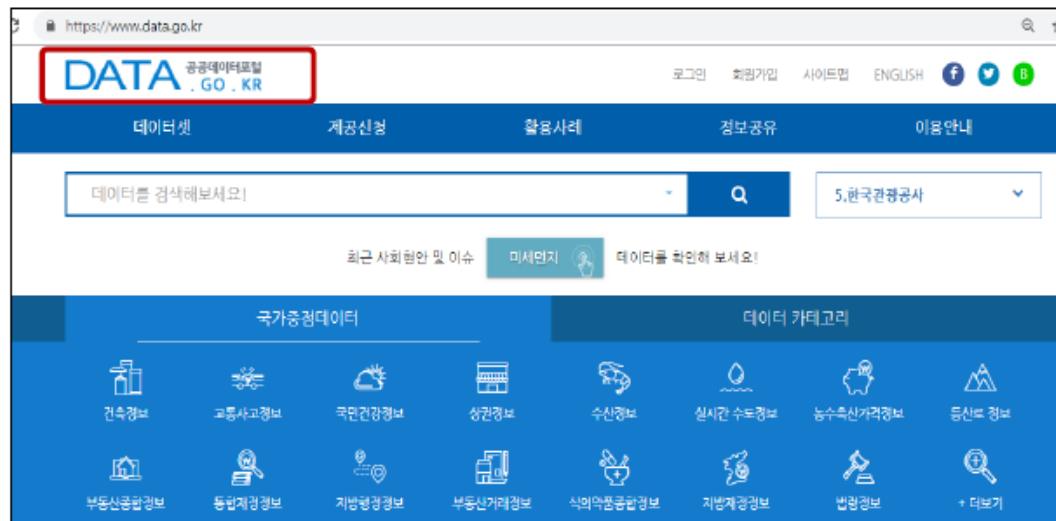


종류	주소
네이버 개발자 센터	<a href="https://developers.naver.com">https://developers.naver.com</a>
카카오 앱 개발 플랫폼 서비스	<a href="https://developers.kakao.com">https://developers.kakao.com</a>
페이스북 개발자 센터	<a href="https://developers.facebook.com">https://developers.facebook.com</a>
트위터 개발자 센터	<a href="https://developer.twitter.com">https://developer.twitter.com</a>
공공데이터포털	<a href="https://www.data.go.kr">https://www.data.go.kr</a>
세계 날씨	<a href="http://openweathermap.org">http://openweathermap.org</a>
유료/무료 API 스토어	<a href="http://mashup.or.kr">http://mashup.or.kr</a> <a href="http://www.apistore.co.kr/api/apiList.do">http://www.apistore.co.kr/api/apiList.do</a>

## 6. 공공 데이터 수집

### ▶ Open API를 활용한 공공 DB 수집

<http://www.data.go.kr/>



# 1. 공공 데이터 수집

---

## ▶ 부산광역시 버스정보시스템 조회 서비스 활용

### ▶ API 활용 신청하기 : 부산광역시 버스정보시스템 조회 서비스

### ▶ 노선정보 조회와 노선 정류소 조회를 이용하여 노선정보 및 실시간 버스 위치 정보 얻기

```
# 노선번호를 이용하여 버스 노선정보를 조회하여 파일에 저장
import urllib.request as req
from bs4 import BeautifulSoup
# API key
key='bz6p9J0zj0TIbSrGsm6BH3B4i9IEW29rnGC6LNchn750ZVUpKW3HKr211MoqBZh0y020yMRxsN60JqwPc0KpXQ=='
lineno=23 #노선 번호
url="http://apis.data.go.kr/6260000/BusanBIMS/busInfo"
params='?ServiceKey='+key+'&lineno='+str(lineno)
savepath="data/businfo.xml"
req.urlretrieve(url+params, savepath)
```



# 1. 공공 데이터 수집

---

## ▶ 부산광역시 버스정보시스템 조회 서비스 활용

```
파일에 저장된 노선 정보로부터 노선 Id(lineid) 얻기
xml=open(savepath, 'r', encoding='utf-8').read()
soup=BeautifulSoup(xml, 'xml')
lineId=None
#print(soup.find_all('item'))
for item in soup.find_all('item'):
    busno=item.find('buslinenum').text
    lineid=item.find('lineid').text
    if busno==lineno:
        break

print(busno, lineid)
```





# 1. 공공 데이터 수집

---

## ▶ 부산광역시 버스정보시스템 조회 서비스 활용

# 노선id(lineid)를 활용하여 노선 정류소 조회를 파일로 저장

```
url2='http://apis.data.go.kr/6260000/BusanBIMS/busInfoByRouteId'
```

```
params2='?ServiceKey='+key+'&lineid='+str(lineid)
```

```
savepath2='data/busposition.xml'
```

```
req.urlretrieve(url2+params2, savepath2)
```



# 1. 공공 데이터 수집

---

## ▶ 부산광역시 버스정보시스템 조회 서비스 활용

# 버스의 실시간 위치 추출

```
xml2=open(savepath2, 'r', encoding='utf-8').read()
soup=BeautifulSoup(xml2,'xml')
busPos_list=[]
for item in soup.find_all('item'):
    bstopnm=item.find('bstopnm')
    lat=item.find('lat')
    lon=item.find('lin')
    if bstopnm==None: bstopnm='-'
    else: bstopnm=bstopnm.text
    if lat==None: lat=0
    else: lat=lat.text
    if lon==None: lon=0
    else: lon=lon.text
    busPos=(bstopnm, lat, lon)
    busPos_list.append(busPos)
print(busPos_list)
```

---



## 2. 네이버 API활용 데이터 수집

### ▶ 네이버 개발자 가입

#### I. 네이버 개발자 센터 접속하기

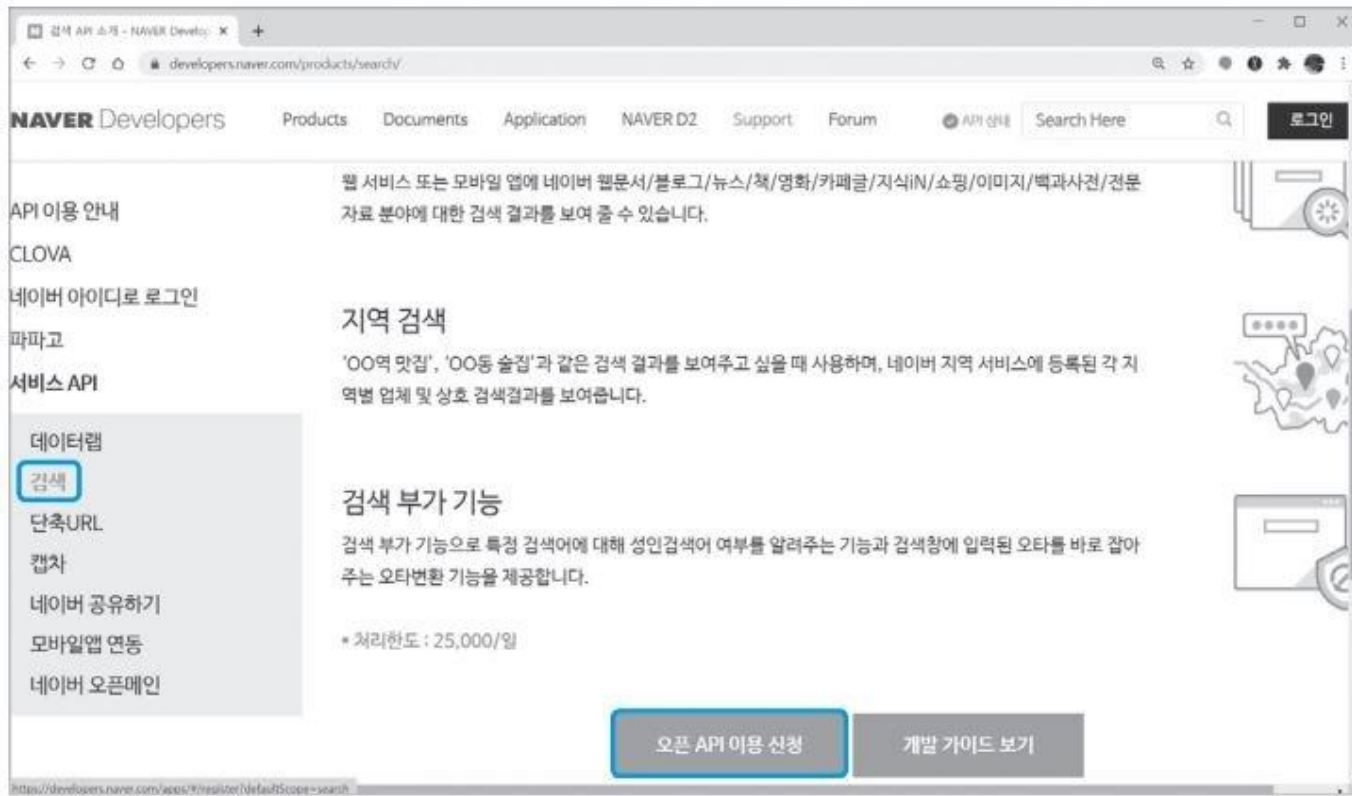
- ▶ 네이버 개발자 센터([https:// developers.naver.com](https://developers.naver.com))에 접속



## 2. 네이버 API활용 데이터 수집

### ▶ 네이버 개발자 가입

#### 2. 오픈 API 이용 신청하기



## 2. 네이버 API활용 데이터 수집

### ▶ 네이버 개발자 가입

#### 3. 애플리케이션 등록하기

내 애플리케이션

애플리케이션 등록

CLOVA Platform Console β

API 제휴 신청

계정 설정

### 애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 내 애플리케이션 메뉴의 서비스 메뉴에 등록하신 애플리케이션 이름으로 서비스 메뉴가 만들어집니다.

애플리케이션 이름	<div>myBig</div> <div>입력</div> <div><ul style="list-style-type: none"><li>• 네이버 아이디로 로그인할 때 사용자에게 표시되는 이름입니다. 가급적 10자 이내의 간결한 이름을 사용해주세요.</li><li>• 40자 이내의 영문, 한글, 숫자, 공백문자, "-", "." 만 입력 가능합니다.</li></ul></div>
사용 API	<div>선택하세요.</div> <div>검색</div>
비로그인 오픈 API 서비스 환경	<div>환경 추가</div> <div>WEB 설정</div> <div>웹 서비스 URL (최대 10개)</div> <div>http://localhost</div> <div>입력</div> <div><ul style="list-style-type: none"><li>• 텍스트 중 우측 끝의 '*' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.</li><li>• http와 https는 구분하지 않습니다.</li><li>• www는 빼고 입력해 주세요. (예: http://naver.com)</li><li>• 서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)</li><li>• 라이브러리드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file:///로컬 URL)</li></ul></div>

등록하기

취소

## 2. 네이버 API활용 데이터 수집

### ▶ 네이버 개발자 가입

#### 4. 애플리케이션 정보 확인하기



The screenshot displays the Naver Developer Console interface for an application named 'nvBig'. The left sidebar contains navigation links: '내 애플리케이션' (My Applications), '애플리케이션 등록' (Register Application), 'CLOVA Platform Console β', 'API 제휴 신청' (Apply for API Partnership), and '계정 설정' (Account Settings). The main content area has a top navigation bar with tabs: '개요' (Overview), 'API 설정' (API Settings), '멤버관리' (Member Management), '로그인 통계' (Login Statistics), 'API 통계' (API Statistics), and 'Playground(Beta)'. The 'API 설정' tab is active, showing the '애플리케이션 정보' (Application Information) section. This section contains two input fields: 'Client ID' with the value 'OLHQM4VX\_MQM5JfkXofa' and 'Client Secret' with a masked value '\*\*\*\*\*'. A '보기' (View) button is located below the 'Client Secret' field. Below the application information, there is an 'API 호출 안내' (API Call Guide) section with a note: '지도 API 인증실때나 네이버 로그인 이용 제한이 걸렸다면 [API 설정] 탭에서 URL 관련 설정을 수정하시면 정상 이용 가능합니다!!!' (If you encounter restrictions when using the Map API authentication or Naver login, you can modify the URL-related settings in the [API Settings] tab to use normally!!!).

내 애플리케이션

nvBig

nvBig

nvBig

nvBig

애플리케이션 등록

CLOVA Platform Console β

API 제휴 신청

계정 설정

nvBig

개요

API 설정

멤버관리

로그인 통계

API 통계

Playground(Beta)

애플리케이션 정보

Client ID

OLHQM4VX\_MQM5JfkXofa

Client Secret

\*\*\*\*\*

보기

API 호출 안내

지도 API 인증실때나 네이버 로그인 이용 제한이 걸렸다면 [API 설정] 탭에서 URL 관련 설정을 수정하시면 정상 이용 가능합니다!!!

## 2. 네이버 API활용 데이터 수집

---

### ▶ 뉴스 검색 샘플예제

```
# 네이버 검색 API 예제 - 블로그 검색
import urllib.request
client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET"
encText = urllib.parse.quote("검색할 단어")
url = "https://openapi.naver.com/v1/search/blog?query=" + encText # JSON 결과
# url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # XML 결과
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)
rescode = response.getcode()
if(rescode==200):
    response_body = response.read()
    print(response_body.decode('utf-8'))
else:
    print("Error Code:" + rescode)
```



## 2. 네이버 API활용 데이터 수집

### ▶ 네이버 뉴스 크롤링 : 1. 전체 작업 설계하기

작업 설계	사용할 코드
1. 검색어 지정하기	srcText = '월드컵'
2. 네이버 뉴스 검색하기	<b>getNaverSearch()</b>
2.1 url 구성하기	url = base + node + srcText
2.2 url 접속과 검색 요청하기	urllib.request.urlopen()
2.3 요청 결과를 응답 JSON으로 받기	json.load()
3. 응답 데이터를 정리하여 리스트에 저장하기	<b>getPostData()</b>
4. 리스트를 JSON 파일로 저장하기	json.dumps()
5. 데이터 프레임 작성, csv파일 저장	
6. 데이터베이스에 저장, 읽기	





## 2. 네이버 API활용 데이터 수집

### 네이버 뉴스 크롤링 : 2. 프로그램 구성 설계하기

[CODE 0]

```
def main()
```

1. 검색어 지정

2. 네이버 뉴스 검색

3. 응답 데이터 정리 후  
리스트에 저장

4. 리스트를 JSON 파일로 저장

[CODE 2]

getNaverSearch()

[CODE 1]

getRequestUrl()

json.load(responseDecode)

Response.read()

[CODE 3]

getPostData()

jsonResult



## 2. 네이버 API활용 데이터 수집

---

### ▶ 라이브러리 설정 및 client\_id, client\_secret 설정

```
import urllib.request
import datetime
import time
import json
import re
import pandas as pd

client_id = "YOUR_CLIENT_ID"
client_secret = "YOUR_CLIENT_SECRET"
```



## 2. 네이버 API활용 데이터 수집

### ▶ main() 함수

```
#[CODE 0]
def main():
    listResult, jsonResult = [], [] # 검색 결과 데이터의 tuple 리스트와 json(dictionary) 리스트
    node = 'news' #크롤링할 대상 : blog, image 등
    srcText = input('검색어를 입력하세요: ')
    cnt = 0

    jsonResponse = getNaverSearch(node, srcText, 1, 100) #[CODE 2] #검색결과 데이터 가져오기
    total = jsonResponse['total'] # 전체 검색 레코드 수

    while ((jsonResponse != None) and (jsonResponse['display'] != 0)): # 데이터가 있을 동안 반복하기
        for post in jsonResponse['items']:
            cnt += 1
            getPostData(post, jsonResult, listResult, cnt) #[CODE 3] # 결과에 필요한 데이터 추출

            start = jsonResponse['start'] + jsonResponse['display'] #start 업데이터
            jsonResponse = getNaverSearch(node, srcText, start, 100) #[CODE 2]

    print('전체 검색 : %d 건' %total)
    json_save(jsonResult, srcText, cnt, node) #json 형태로 결과 저장
    csv_save(listResult, srcText) # 데이터 프레임 작성 후 csv로 저장
    insert_nwes(listResult) # mysql DB에 저장
    select_all() # DB에 저장된 데이터 가져오기

main()
```

## 2. 네이버 API활용 데이터 수집

---

### ▶ 원격접속

```
#[CODE 1]
def getReqUrl(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
```



## 2. 네이버 API활용 데이터 수집

---

- ▶ url + parameter 결합 후 데이터 요청, 결과 json파일 변환 후 리턴

```
#[CODE 2]
def getNaverSearch(node, srcText, start, display):
    base = "https://openapi.naver.com/v1/search"
    node = "/%s.json" % node
    parameters = "?query=%s&start=%s&display=%s" %(urllib.parse.quote(srcText), start, display)

    url = base + node + parameters
    responseDecode = getReqUrl(url) #[CODE 1]

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



## 2. 네이버 API활용 데이터 수집

### ▶ 결과 데이터 에서 필요한 데이터 추출

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

    #title = post['title'].replace('<b>','').replace('</b>','').replace('&quot;','') # 불필요한 값 제거
    pattern = re.compile('[ㄱ-ㅎ가-힣]+') # 정규표현식으로 한글만 추출
    title=' '.join(pattern.findall(post['title']))

    #description = post['description'].replace('<b>','').replace('</b>','').replace('|','').replace('&quot;','')
    description = ' '.join(pattern.findall(post['description']))
    org_link = post['originallink']
    link = post['link']

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

## 2. 네이버 API활용 데이터 수집

---

### ▶ json 및 csv 저장

```
def json_save(jsonResult, srcText, cnt, node):  
  
    with open('data/%s_naver_%s.json' % (srcText, node), 'w', encoding='utf8') as outfile:  
        jsonFile = json.dumps(jsonResult, indent = 4, sort_keys = True, ensure_ascii = False)  
        outfile.write(jsonFile)  
  
    print("가져온 데이터 : %d 건" %(cnt))  
    print('%s_naver_%s.json SAVED' % (srcText, node))
```

```
def csv_save(listResult, srcText):  
  
    #print(listResult)  
    df=pd.DataFrame(listResult, columns=['title','description','original_link','pubDate'])  
    df.to_csv('data/{ }_news_result.csv'.format(srcText))  
    df.head()
```



## 2. 네이버 API활용 데이터 수집

---

### ▶ 데이터베이스에 데이터 저장

```
def insert_nwes(news_list):  
    conn = pymysql.connect( #pymysql 라이브러리 사용 DB연결  
        host='localhost',  
        user='pgm',  
        password='1234',  
        db='pydb',  
        charset='utf8')  
    cursor=conn.cursor()  
    sql="insert into news_tbl(title, description, original_link, pub_date) values(%s, %s, %s, %s)"  
    cursor.executemany(sql, news_list)  
    conn.commit()  
    conn.close()
```





## 2. 네이버 API활용 데이터 수집

---

### ▶ 데이터베이스에 저장된 데이터 읽어오기

```
def select_all():  
    conn = pymysql.connect( #pymysql 라이브러리 사용 DB연결  
        host='localhost',  
        user='pgm',  
        password='1234',  
        db='pydb',  
        charset='utf8')  
    cursor=conn.cursor()  
    sql="select * from news_tbl"  
    cursor.execute(sql)  
  
    for item in cursor:  
        print(item)  
    conn.close()
```

