

02_API 크롤링

- Naver Rest API

- > 파파고 번역 api
- > 통합검색어 트렌드 api
- > request token 얻기
 - <https://developers.naver.com>
 - Request Token 얻기 : 애플리케이션등록 > app_key 획득
 - app_key를 이용해서 데이터 가져오기

Client ID	0keVMr8xAhCCKKHqq_SZ
Client Secret	znXZuY_dDI
	<button>재발급</button>

- Naver Rest API

- > request token 얻기

- <https://developers.naver.com>
 - Request Token 얻기 : 애플리케이션등록 > app_key 획득
 - app_key를 이용해서 데이터 가져오기

```
import requests, json
```

```
CLIENT_ID = "0keVMr8xAhCCKKHqq_SZ"
```

```
CLIENT_SECRET = "znXZuY_dDI"
```

- Naver Rest API

- > 파파고 번역 api

- <https://developers.naver.com/docs/papago>

```
url = 'https://openapi.naver.com/v1/papago/n2mt'
msg = "데이터 사이언스는 재미있습니다."
params = { "source": "ko", "target": "en", "text": msg }
headers = { "Content-Type": "application/json",
            "X-Naver-Client-Id": CLIENT_ID,
            "X-Naver-Client-Secret": CLIENT_SECRET}
response = requests.post(url, json.dumps(params),
                        headers=headers)

print(response)
print(response.json()["message"]["result"]["translatedText"])
```

Out : <Response [200]>

'Data Science is fun.'

- Naver Rest API

- > 함수로 만들기

```
def translate(msg, source = "ko", target = "en"):
    url = "https://openapi.naver.com/v1/papago/n2mt"
    params = { "source": source, "target": target, "text": msg }
    headers = {"Content-Type": "application/json",
               "X-Naver-Client-Id": CLIENT_ID,
               "X-Naver-Client-Secret": CLIENT_SECRET}
    response = requests.post(url, json.dumps(params),
                             headers=headers)

    if response.status_code == 200:
        return response.json()["message"]["result"]["translatedText"]
    else:
        return response.status_code
```

- Naver Rest API

> 파파고 번역 api

```
msg = "데이터 분석 및 크롤링"  
translated_text = translate(msg)  
translated_text
```

Out : 'Analyze and crawl data'

```
msg = "Analyze and crawl data"  
translated_text = translate(msg, source="en", target="ko")  
translated_text
```

Out : '데이터 분석 및 탐색'

- Naver Rest API

- > 통합검색어 트렌드 api

- 내 어플리케이션 > API 설정 > 데이터랩(검색어 트렌드) 추가

```
url = "https://openapi.naver.com/v1/datalab/search"
params = {
    "startDate": "2023-01-01",
    "endDate": "2023-12-31",
    "timeUnit": "month",
    "keywordGroups": [
        {"groupName": "트위터", "keywords": ["트위터", "트윗"]},
        {"groupName": "페이스북", "keywords": ["페이스북", "페북"]},
        {"groupName": "인스타그램", "keywords": ["인스타그램",
            "인스타"]}]
    }
response = requests.post(url, data = json.dumps(params), headers=headers)
```

- Naver Rest API

- > 통합검색어 트렌드 api

```
datas = response.json()['results']
```

Out :

```
[{'title': '트위터',  
  'keywords': ['트위터', '트윗'],  
  'data': [{'period': '2023-01-01', 'ratio': 94.93332},  
            {'period': '2023-02-01', 'ratio': 90.62362},  
            {'period': '2023-03-01', 'ratio': 89.60876},  
            {'period': '2023-04-01', 'ratio': 100},  
            {'period': '2023-05-01', 'ratio': 81.95453},  
            {'period': '2023-06-01', 'ratio': 85.40499},  
            {'period': '2023-07-01', 'ratio': 96.99514},  
            {'period': '2023-08-01', 'ratio': 95.33897},  
            {'period': '2023-09-01', 'ratio': 81.96185},  
            {'period': '2023-10-01', 'ratio': 85.98988},  
            {'period': '2023-11-01', 'ratio': 81.43833},  
            {'period': '2023-12-01', 'ratio': 82.00285}]},  
 {'title': '페이스북',  
  'keywords': ['페이스북', '페북'],  
  'data': [{'period': '2023-01-01', 'ratio': 53.00228},  
            {'period': '2023-02-01', 'ratio': 46.08832},  
            {'period': '2023-03-01', 'ratio': 47.57329},  
            {'period': '2023-04-01', 'ratio': 44.75489},  
            {'period': '2023-05-01', 'ratio': 45.13131},  
            {'period': '2023-06-01', 'ratio': 42.12839},  
            {'period': '2023-07-01', 'ratio': 42.25194},  
            {'period': '2023-08-01', 'ratio': 41.05227},  
            {'period': '2023-09-01', 'ratio': 37.28701},  
            {'period': '2023-10-01', 'ratio': 37.21907},  
            {'period': '2023-11-01', 'ratio': 35.51946},  
            {'period': '2023-12-01', 'ratio': 36.08099}]},
```


- Kakao Rest API

- > Kakao 지도 검색 api
- > request token 얻기
 - <https://developers.kakao.com>
 - Request Token 얻기 : 애플리케이션등 추가하기
 - rest api key를 이용해서 데이터 가져오기

앱 키

네이티브 앱 키	a3ab81f5e7fbb855928c771976c866f7
REST API 키	d44eaac9a1ae4db5ec4a3b895553fb84
JavaScript 키	eb99ed1481d326dc9d039d160f34debe
Admin 키	2501367cd946a07ec6e3cac2c524c656

- Kakao Rest API

- > Kakao 지도 검색 api

- <https://developers.kakao.com/docs/latest/ko/local/dev-guide>

```
restapi_key = "d44eaac9a1ae4db5ec4a3b895553fb84"
url = "https://dapi.kakao.com/v2/local/search/address.json"
params = { "query": '집현동로 77' }
headers = {"Authorization": "KakaoAK " + restapi_key}
response = requests.get(url, headers=headers, params=params)
response
```

Out : <Response [200]>

- Kakao Rest API

- > Kakao 지도 검색 api

```
datas = response.json()
datas['documents'][0]
```

Out : {'address': {'address_name': '세종특별자치시 집현동 1129',
'b_code': '3611011800',
'h_code': '3611055600',
'main_address_no': '1129',
'mountain_yn': 'N',
'region_1depth_name': '세종특별자치시',
'region_2depth_name': '',
'region_3depth_h_name': '반곡동',
'region_3depth_name': '집현동',
'sub_address_no': '',
'x': '127.325356593945',
'y': '36.4910924250367'},
'address_name': '세종특별자치시 집현동로 77',
'address_type': 'ROAD_ADDR',
'road_address': {'address_name': '세종특별자치시 집현동로 77',
'building_name': '',
'main_building_no': '77',
'region_1depth_name': '세종특별자치시',
'region_2depth_name': '',
'region_3depth_name': '집현동',
'road_name': '집현동로',
'sub_building_no': '',
'underground_yn': 'N',
'x': '127.325356593945',
'y': '36.4910924250367',
'zone_no': '30141'},
'x': '127.325356593945',
'y': '36.4910924250367'}

- Kakao Rest API

- > 함수로 만들기

```
def search_location(query):  
    url = "https://dapi.kakao.com/v2/local/search/address.json"  
    params = { "query": query }  
    headers = {"Authorization": "KakaoAK " + restapi_key}  
    response = requests.get(url, headers=headers, params=params)  
    if response.status_code == 200:  
        return response.json()['documents'][0]  
    else:  
        return response.status_code
```

- Kakao Rest API

> 함수로 만들기

```
address = search_location('집현동로 77')  
print(address['address_name'])  
print(address['x'])  
print(address['y'])
```

Out : 세종특별자치시 집현동로 77
127.325356593945
36.4910924250367

- Kakao Rest API

- > 위도, 경도 데이터를 사용하여 매물 데이터 수집
 - 위도 경도로 geohash 알아내기
 - geohash로 매물 정보 가져오기

```
address = search_location('집현동로 77')  
lng = float(address['x'])  
lat = float(address['y'])
```

```
!pip install geohash2
```

```
import geohash2  
geohash = geohash2.encode(lat, lng, precision=5)  
geohash
```

Out : 'wy6x9'

- Kakao Rest API

> 직방 api 사용하여 geohash로 데이터 검색

```
url = f'https://apis.zigbang.com/v2/items/oneroom?geohash={geohash}₩  
&depositMin=0&rentMin=0&salesTypes[0]=월세&salesTypes[1]=전세₩  
&domain=zigbang&checkAnyItemWithoutFilter=true'
```

```
response = requests.get(url)  
datas = response.json()['items']  
ids = [data["itemId"] for data in datas]  
len(ids), ids[:5]
```

Out : (40, [39211105, 39232370, 39338080, 39362579, 39381566])

- Kakao Rest API

> 매물 아이디로 정보 가져오기

```
url = "https://apis.zigbang.com/v2/items/list"
params = {"item_ids": ids, "domain": "zigbang"}
response = requests.post(url, params)
datas = response.json()['items']
datas[0]
```

Out : {'item_id': 39211105,
'section_type': None,
'images_thumbnail': 'https://ic.zigbang.com/ic/items/39211105/1.jpg',
'sales_type': '월세',
'sales_title': '월세',
'deposit': 300,
'rent': 60,
'size_m2': 39.79,
'공급면적': {'m2': 39.79, 'p': '12'},
'전용면적': {'m2': 26.59, 'p': '8'},
'계약면적': None,
'room_type_title': None,
'floor': '고',
'floor_string': '고',
'building_floor': '8',
'title': '세종시청 맞은편 보람도 유일의 원룸. 시청방면 개방감 좋음',

• 공공데이터포털

> request token 얻기

- <https://data.go.kr>
- Request Token 얻기 : 회원가입 > 본인인증
- 인증키를 통해 데이터 활용 신청 가능

017

오픈API 상세



XML JSON 대구광역시_민원데이터 분석 결과

대구시로 접수되는 민원데이터를 분석하여 민원 TOP 키워드 정보와 처리완료건수, 이관건수, 처리중건수 등의 월간 민원접수 및 처리 현황



활용신청

오류신고 및

담당자 문의

활용목적 선택

*표시는 필수 입력항목입니다.

*활용목적

☐ 웹 사이트 개발 ☐ 앱개발 (모바일,솔루션등) ☐ 기타 ☒ 참고자료 ☐ 연구(논문 등)

데이터 분석 학습용

10/250

- 공공데이터포털

- > request token 얻기

- <https://data.go.kr>
 - Request Token 얻기 : 회원가입 > 본인인증
 - 인증키를 통해 데이터 활용 신청 가능

서비스정보

참고문서	대구광역시 민원분석조회서비스 OpenAPI가이드 v1.0.docx
데이터포맷	JSON+XML
End Point	http://apis.data.go.kr/6270000/dgMwData
API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다. 포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다. * 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.	
일반 인증키 (Encoding)	n5rRBCVx%2B%2BFYhgyS2HeF64cUX7DNtn0XsOyzAJK%2Fon8r9mcOANXspw9fNMtz3J7XBbD6FjnFRGrX6TKWDbZxnQ%3D%3D
일반 인증키 (Decoding)	n5rRBCVx++FYhgyS2HeF64cUX7DNtn0XsOyzAJK/on8r9mcOANXspw9fNMtz3J7XBbD6FjnFRGrX6TKWDbZxnQ==

```
service_key =  
'n5rRBCVx++FYhgyS2HeF64cUX7DNtn0XsOyzAJK/on8r9mcOANXspw9fNMtz3J7XBbD6FjnFRGrX6TKWDbZxnQ=='
```

- 공공데이터 포털

- > 샘플코드 사용하기

```
url='http://apis.data.go.kr/6270000/dgMwData/getTopKeyword'
params ={'serviceKey' : service_key, 'type' : 'xml', 'gbn' : 't',
         'startDt' : '20191001', 'endDt' : '20191031',
         'gugun' : '중구', 'cni' : 'dud' }

response = requests.get(url, params=params)
print(response.text)
```

Out : <response>
<header>
<resultCode>200</resultCode>
<resultMsg>OK</resultMsg>
</header>
<body>
<count>20</count>
<items>
<item>
<no>1</no>
<word>휠 손</word>
<docCount>113</docCount>
</item>