

API 활용 데이터 수집

# 1주차 API 활용

API 활용

# 강의순서

1

공공 데이터 API

2

네이버 API

공공 데이터 API

# 1차시

## 공공 데이터 API

# 공공 데이터 - 공공 데이터 포털

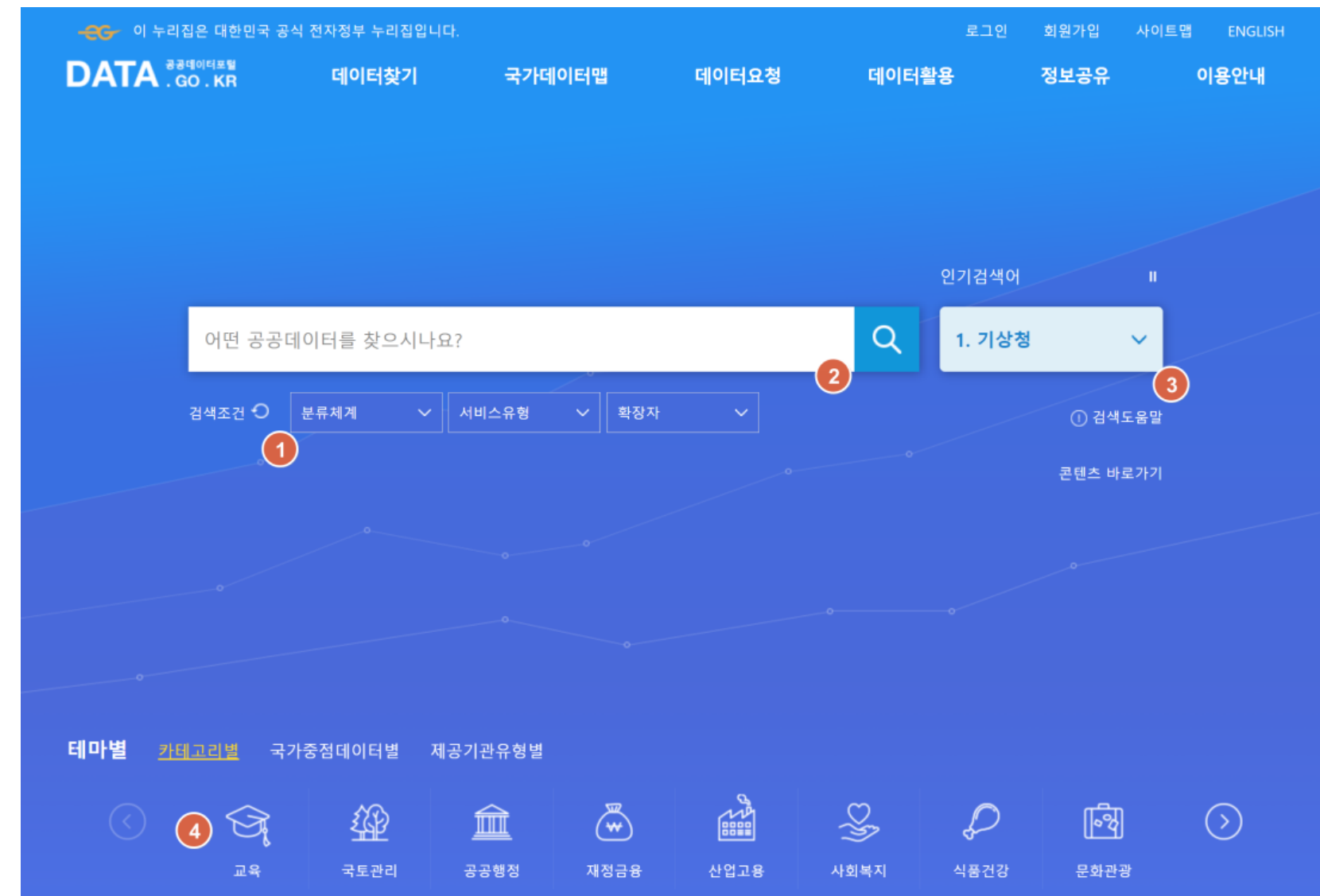
## 공공 데이터란?

- 공기관에서 만들어내는 모든 자료나 정보, 국민 모두의 소통과 협력을 이끌어내는 공적인 정보를 공공 데이터라고 합니다.
- 각 공공기관이 보유한 공공 데이터 목록과 국민에게 개방할 수 있는 공공 데이터를 포털에 등록하여 모두가 사용할 수 있도록 공유되어 있습니다.

## 공공 데이터 포털

- 공공기관이 보유하고 있는 공공 데이터를 하나로 통합 관리하는 창구

1. 구체적인 검색 조건을 설정하여 검색할 수 있습니다.
2. 검색어로 검색해볼 수 있습니다.
3. 인기 검색어 중에서 선택하여 검색할 수 있습니다.
4. 테마별로 중요한 데이터를 정리하여 찾아볼 수 있습니다.



# Open API

## API란?

- Application Programming interface 의 약자로 응용 프로그램 프로그래밍 인터페이스입니다.
- 다양한 응용 프로그램에서 사용할 수 있는 운영 체제, 혹은 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스입니다.

## Open API란?

- Open API란 누구나 사용할 수 있도록 공개된 API를 말합니다.
- 데이터를 표준화하고 프로그래밍하여 외부 소프트웨어 개발자나 사용자와 공유하는 프로그램을 말합니다.

# Open API – 공공 데이터

- 공공 데이터 중에서 필요한 오픈 API를 검색해봅니다.
- 기상청\_관광코스별 관광지 상세 날씨 조회 서비스 API를 선택해 봅니다.
- 활용 목적 선택 후 활용신청을 합니다.

전체(112건)

파일데이터(86건)

오픈 API(26건)

표준데이터셋0개(0건)

정확도순 10개씩 정렬

오픈 API (26건)

교통물류

공공기관

미리보기

XML

JSON

한국도로공사\_휴게소별 날씨 정보

고속도로 휴게소별 날씨 정보를 제공한다.

제공기관 한국도로공사 수정일 2021-01-26 조회수 54443 활용신청 3623 키워드 고속도로,휴게소,날씨

바로가기

환경기상

자치행정기관

미리보기

XML

경상남도 창원군 날씨정보

창녕군 현재 날씨정보 오픈API 제공

제공기관 경상남도 창원군 수정일 2016-05-16 조회수 51664 활용신청 1589 키워드 날씨,기후,온도

바로가기

과학기술

국가행정기관

국가중립

미리보기

XML

JSON

기상청\_관광코스별 관광지 상세 날씨 조회서비스

관광코스별 관광지의 동네예보, 기상지수예보, 시군구별 관광기후지수 정보를 조회하는 서비스

제공기관 기상청 수정일 2023-11-30 조회수 80917 활용신청 2931 키워드 관광코스별 동네예보,기상지수예보,시군구별관광기후지수

활용신청

## 활용목적 선택

\*표시는 필수 입력항목입니다.

### \*활용목적

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

데이터 분석에 활용

10/250

### 첨부파일

파일 선택

Drag & Drop으로 파일을 선택 가능합니다.

# Open API – 공공 데이터

- 신청이 완료되면 일반 인증키(Decoding)을 따로 저장해 둡니다.
- 상세설명을 누르면 API사용 방법 등에 대해서 자세하게 알 수 있는 창으로 이동합니다.

홈 > 마이페이지 > 데이터 활용 > Open API > 활용신청 현황

## 마이페이지

### 데이터 활용

#### Open API

활용신청 현황

인증키 발급현황

파일 데이터

관심 데이터

### 데이터 요청

### 나의 문의

### 회원정보 수정

## 개발계정 상세보기

### 기본정보

|       |                           |      |      |      |
|-------|---------------------------|------|------|------|
| 데이터명  | 기상청_관광코스별 관광지 상세 날씨 조회서비스 |      |      | 상세설명 |
| 서비스유형 | REST                      | 심의여부 | 자동승인 |      |
| 신청유형  | 개발계정   활용신청               | 처리상태 | 승인   |      |
| 활용기간  | 2024-02-26 ~ 2026-02-26   |      |      |      |

### 서비스정보

|   |  |
|---|--|
| 데이터포맷   | JSON+XML   |
| End Point   | http://apis.data.go.kr/1360000/TourStnInfoService1 |
| API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다.<br>포털에서 제공되는 <b>Encoding/Decoding</b> 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다.<br>* 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다. |  |
| 일반 인증키 (Encoding)   |  |
| 일반 인증키 (Decoding)   |  |

# Open API - 공공 데이터

- 상세설명 페이지에서 볼 수 있는 화면입니다.

1. BASE URL : 기본 API 주소입니다.
2. API 목록 : 요청할 수 있는 유형입니다. (여기에서는 두 가지)

## 기상청\_관광코스별 관광지 상세 날씨 조회서비스 1.0.0

[ Base URL: apis.data.go.kr/1360000/TourStnInfoService1 ]

관광코스별 관광지의 동네예보, 기상지수예보, 시군구별 관광기후지수 정보를 조회하는 서비스

### API 목록

|     |                                  |
|-----|----------------------------------|
| GET | /getTourStnVilageFcst1 동네예보조회    |
| GET | /getCityTourCimIdx1 시군구별관광기후지수조회 |

- 2번 API 선택시 기본 API 요청을 진행할 수 있고, 필수 요청 사항들이 \* required 라고 표시되어 있습니다.

## 기상청\_관광코스별 관광지 상세 날씨 조회서비스 1.0.0

[ Base URL: apis.data.go.kr/1360000/TourStnInfoService1 ]

관광코스별 관광지의 동네예보, 기상지수예보, 시군구별 관광기후지수 정보를 조회하는 서비스

### API 목록

| GET   | /getTourStnVilageFcst1 동네예보조회          |
|---|--|
| 조회조건(현재시각, 예보기간, 코스ID)으로 동네예보 데이터(코스명, 관광지명, 테마, 기온, 습도 등)를 조회하는 기능 |  |
| Parameters <span>OpenAPI 실행 준비</span>                               |  |
| Name  | Description                            |
| ServiceKey * required<br>string<br>(query)                          | 공공데이터포털에서 받은 인증키<br>ServiceKey         |
| pageNo * required<br>string<br>(query)                              | 페이지번호<br>pageNo                        |
| numOfRows * required<br>string<br>(query)                           | 한 페이지 결과 수<br>numOfRows                |
| dataType<br>string<br>(query)                                       | 요청자료형식(XML/JSON)<br>dataType           |
| CURRENT_DATE * required<br>string<br>(query)                        | 2016-12-01 01시부터 조회<br>CURRENT_DATE    |
| HOUR * required<br>string<br>(query)                                | CURRENT_DATE부터 24시간 후까지의 자료 호출<br>HOUR |
| COURSE_ID * required<br>string<br>(query)                           | 관광 코스ID<br>COURSE_ID                   |



# Open API - 공공 데이터 요청 방법

- API 요청시에는 requests 라는 외부 라이브러리를 사용합니다.
- BASE\_URL 에는 기본 BASE\_URL 을 입력합니다. 이때, http://를 포함해야 합니다.
- SERVICE\_KEY 에는 일반 인증키(Decoding)을 입력합니다.
- ENDPOINT1 에는 동네예보 조회의 주소를 선택합니다.

```
!pip install -qq requests
```

✓ 1.3s

Python

```
import requests  
import json
```

```
# API 정보
```

```
BASE_URL = "http://apis.data.go.kr/1360000/TourStnInfoService1"
```

```
SERVICE_KEY = "키값을 입력해주세요"
```

✓ 0.0s

Python

```
ENDPOINT1 = "/getTourStnVilageFcst1"
```

Python

# Open API - 공공 데이터 요청 방법

- 파라미터를 설정합니다. 필수로 표기되어 있는 것은 반드시 추가해야 하며, 형식이 맞는지 예시를 보고 설정합니다. URL에 포함될 옵션값들을 나타냅니다.
- API 요청은 requests.get(API 요청 주소, 파라미터)로 요청합니다.
- response 변수에는 API 요청에 대한 응답 정보가 담겨 있습니다. 200: 성공 400: 잘못된 요청 500: 서버 오류

```
# 파라미터 설정
params = {
    "ServiceKey": SERVICE_KEY,
    "pageNo": "1",
    "numOfRows": "10",
    "dataType": "JSON",
    "CURRENT_DATE": "2018123110",
    "HOUR": "24",
    "COURSE_ID": "52",
}
# API 요청
response = requests.get(
    f"{BASE_URL}{ENDPOINT1}",
    params=params,
    verify=False
)
```

# Open API - 공공 데이터 요청 방법

- status\_code - 200인 경우, 성공적으로 요청을 받은 상태입니다.
- response.json()을 통해 json데이터로 파싱합니다.
- response.json()["response"]["body"]["items"]["item"]에는 각 결과 정보가 담겨 있습니다.
- json 정보 쉽게 확인하기 :  
<https://jsonformatter.org/json-parser>

```
# 응답 처리
if response.status_code == 200:
    # JSON 데이터 파싱
    data = response.json()
    for content in data['response']['body']['items']['item'][0].items():
        print(content)
else:
    print(f"API 요청 실패: {response.status_code}")
```

✓ 0.6s

Python

결과 정보 중 첫 번째 결과

```
('tm', '2019-01-02 09:00')
('thema', '문화/예술')
('courseId', '52')
('courseAreaId', '1100000000')
('courseAreaName', '서울특별시')
('courseName', '자유여행코스 - 홍대')
('spotAreaId', 5202)
('spotAreaName', '서울')
('spotName', '(서울)KT&G 상상마당')
('th3', -6)
('wd', 302)
('ws', 0)
('sky', 1)
('rhbm', 55)
('pop', 0)
```

# 서울시 공공 데이터

로그인 | 회원가입 | 사이트맵

서울 열린데이터 광장

| 공공데이터                            | 통계         | 서울빅데이터       | 소식&참여       | 이용안내        |
|----------------------------------|------------|--------------|-------------|-------------|
| <a href="#">공공데이터</a>            | 통계표        | 서울 생활인구      | 데이터소식       | 열광소개        |
| <a href="#">인기그룹데이터</a>          | 통계목록       | 서울 생활이동      | 공지사항        | Open API 소개 |
| <a href="#">새로 개방한 데이터</a>       | 통계분석       | 서울 시민생활 데이터  | FAQ         | 공공데이터 제공신청  |
| <a href="#">인기데이터</a>            | 서울의하루      | 서울 실시간 도시데이터 | 문의하기        |             |
| <a href="#">Linked Open Data</a> | 서울통계간행물    | 서울 생활물류      | 활용사례(갤러리)   |             |
| <a href="#">자치구 열린데이터</a>        | 서울의 인기통계   | AI학습데이터      | 데이터 수요창구    |             |
|                                  | 서울의 100대통계 |              | 열광 뉴스레터     |             |
|                                  |            |              | 서울 데이터 펠로우십 |             |
|                                  |            |              | 경진대회        |             |

# 서울시 공공 데이터

미리보기

단함 -

SheetOpen API

필드명필터선택

검색명

내려받기(CSV)내려받기(JSON)조회

| 주차장코드   | 주차장명           | 주소               | 주차장 종류 | 주차장 종류명 | 운영구분 | 운영구분명          | 전화번호        |
|---------|----------------|------------------|--------|---------|------|----------------|-------------|
| 1010089 | 초안산근린공원주...    | 도봉구 창동 24-0      | NW     | 노외 주차장  | 1    | 시간제 주차장        |             |
| 1012254 | 마들스타디움(근린...   | 노원구 상계동 770-2    | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2289-673 |
| 1013181 | 마장동(건물) 공영...  | 성동구 마장동 463-2    | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2204-797 |
| 1025695 | 영등포여고 공영(구)    | 영등포구 신길동 184-3   | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2677-140 |
| 1025696 | 당산근린공원 공영(...  | 영등포구 당산동3가 ...   | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2677-140 |
| 1025697 | 대림운동장(구)       | 영등포구 대림동 780-0   | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2677-140 |
| 1027845 | 휘경마을 공영주차...   | 동대문구 휘경동 308...  | NW     | 노외 주차장  | 3    | 시간제 + 거주자 주... | 02-2247-966 |
| 1028036 | 중산공영주차장(구)     | 동대문구 답십리동 2...   | NW     | 노외 주차장  | 3    | 시간제 + 거주자 주... | 02-2247-966 |
| 1028245 | 방학동 도깨비시장 ...  | 도봉구 방학동 632-1    | NW     | 노외 주차장  | 3    | 시간제 + 거주자 주... | 02-901-5100 |
| 1032247 | 은평평화공원(구)      | 은평구 녹번동 153-1    | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-350-5183 |
| 1033125 | 압구정 428 공영주... | 강남구 압구정동 428-0   | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2176-097 |
| 1033590 | 신림동공영주차장(...   | 관악구 신림동 1456-3   | NW     | 노외 주차장  | 3    | 시간제 + 거주자 주... | 02-2081-267 |
| 1033754 | 포이초교 공영주차...   | 강남구 개포동 1273-0   | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2176-092 |
| 1034344 | 수유2동 공영주차...   | 강북구 수유동 660-30   | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-900-7443 |
| 1034345 | 송중동(임시)공영...   | 강북구 미아동 860-1... | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-988-1283 |
| 1037569 | 바람개비주차장(구)     | 양천구 목동 919-8     | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2644-138 |
| 1037932 | 구로디지털단지역 ...   | 구로구 구로동 810-3    | NW     | 노외 주차장  | 1    | 시간제 주차장        | 02-2290-603 |

※ sheet는 최대 1,000건까지 노출됩니다. 전체 데이터는 CSV파일을 내려받아 확인해 주시기 바랍니다.

# 서울시 공공 데이터

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import koreanize_matplotlib
4 df = pd.read_csv("서울시 공영주차장 안내 정보.csv",encoding="cp949")
```

✓ 0.0s

Python

```
1 df['주소']
```

✓ 0.0s

Python

```
0      도봉구 창동 24-0
1      노원구 상계동 770-2
2      성동구 마장동 463-2
3      영등포구 신길동 184-3
4      영등포구 당산동3가 385-0
...
18779   강북구 미아동 748-2
18780   강북구 수유동 49-7
18781   종로구 구기동 139-9
18782   구로구 구로동 1-4
18783   구로구 구로동 102-0
Name: 주소, Length: 18784, dtype: object
```

```
1 df.info()
```

✓ 0.0s

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18784 entries, 0 to 18783
Data columns (total 39 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   주차장코드                            18784 non-null  int64
1   주차장명                              18784 non-null  object
2   주소                                  18784 non-null  object
3   주차장 종류                            18784 non-null  object
4   주차장 종류명                          18784 non-null  object
5   운영구분                              18784 non-null  int64
6   운영구분명                            18784 non-null  object
7   전화번호                              9756 non-null   object
8   주차현황 정보 제공여부                18784 non-null  int64
9   주차현황 정보 제공여부명              18784 non-null  object
10  총 주차면                             18605 non-null  float64
11  유무료구분                            18784 non-null  object
12  유무료구분명                          18784 non-null  object
13  야간무료개방여부                      18784 non-null  object
14  야간무료개방여부명                    18784 non-null  object
15  평일 운영 시작시각(HHMM)              18784 non-null  int64
16  평일 운영 종료시각(HHMM)              18784 non-null  int64
17  주말 운영 시작시각(HHMM)              18784 non-null  int64
18  주말 운영 종료시각(HHMM)              18784 non-null  int64
19  공휴일 운영 시작시각(HHMM)            18784 non-null  int64
...
37  위도                                  17957 non-null  float64
38  경도                                  17957 non-null  float64
dtypes: float64(14), int64(9), object(16)
memory usage: 5.6+ MB
```

# 서울시 공공 데이터

```
1 ku_dict = {}
2
3 for temp in df['주소']:
4     ku = temp.split()[0]
5
6     if ku in ku_dict:
7         ku_dict[ku] = ku_dict[ku] + 1
8         # ku_dict[ku] += 1
9     else:
10        ku_dict[ku] = 1
11 print(ku_dict)
```

✓ 0.0s

Python

{'도봉구': 93, '노원구': 566, '성동구': 3532, '영등포구': 1453, '동대문구': 607, '은평구': 26, '강남구': 1408, '관악구': 362, '강북구': 465, '양천구': 933, '서초구': 629, '송파구': 1453}

```
1 ku_df = pd.DataFrame.from_dict([ku_dict])
2 ku_df
```

✓ 0.0s

Python

|   | 도봉구 | 노원구 | 성동구  | 영등포구 | 동대문구 | 은평구 | 강남구  | 관악구 | 강북구 | 양천구 | ... | 서초구 | 송파구  |
|---|-----|-----|------|------|------|-----|------|-----|-----|-----|-----|-----|------|
| 0 | 93  | 566 | 3532 | 1453 | 607  | 26  | 1408 | 362 | 465 | 933 | ... | 629 | 1453 |

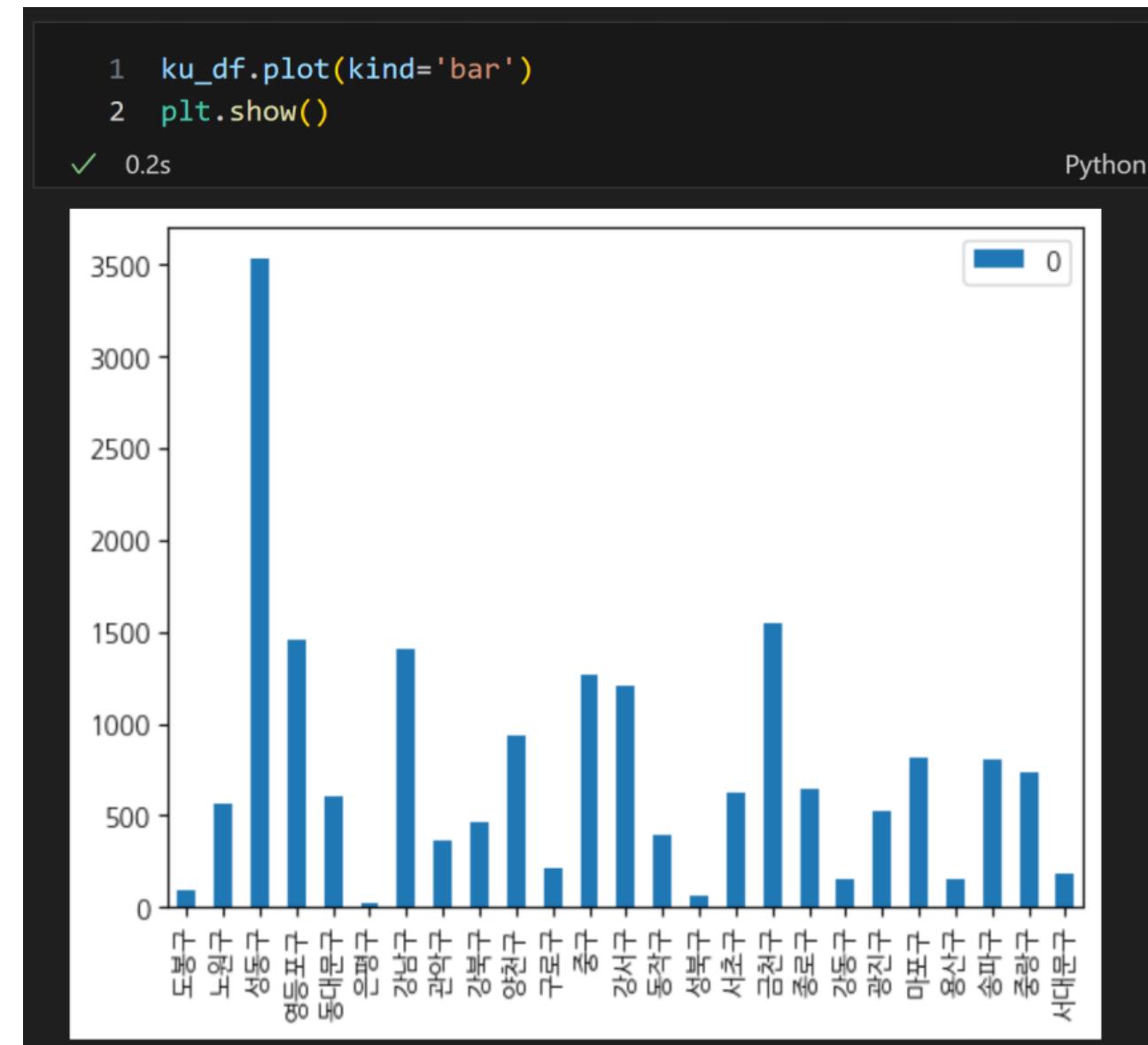
```
1 ku_df = ku_df.T
2 ku_df
```

✓ 0.0s

Python

|      | 0    |
|------|------|
| 도봉구  | 93   |
| 노원구  | 566  |
| 성동구  | 3532 |
| 영등포구 | 1453 |
| 동대문구 | 607  |
| 은평구  | 26   |
| 강남구  | 1408 |
| 관악구  | 362  |
| 강북구  | 465  |
| 양천구  | 933  |

# 서울시 공공 데이터





# 네이버 API

- Naver에서 제공하는 오픈 API, 검색 등의 기능 제공
- <https://developers.naver.com/>

## 1.회원가입 후 로그인

✓ API 상태

🔍

로그인

## 2.애플리케이션 등록

**NAVER** Developers   Products   Documents   Application   Support   Forum

내 애플리케이션   애플리케이션 등록   API 재휴 신청   계정 설정

# 네이버 API

## 3.API 이용신청

|                    |  |
|--------------------|--|
| 애플리케이션 이름 ⇄        | <div>네이버api맵 ✓</div> <ul style="list-style-type: none"><li>• 네이버 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요.</li><li>• 40자 이내의 영문, 한글, 숫자, 공백문자, 쉼표(,), "/", "-", "_", 만 입력 가능합니다.</li></ul> |
| 사용 API ⇄           | 1 <div>선택하세요. ▼ ✓</div>  |
|                    | 2 <div>검색 ✕</div>  |
| 비로그인 오픈 API 서비스 환경 | 3 <div>환경 추가 ▼</div>   |
|                    | 4 <div>WEB 설정 ✕ ^</div> <div>웹 서비스 URL (최대 10개)</div> <div><div>http://naver.com - ✓</div><div>http://도메인주소, file:///로컬URI - !</div></div>   |

## 4.등록정보 확인

### 애플리케이션 정보

|               |                                |
|---------------|--------------------------------|
| Client ID     | 5ApKdjvgYDMysclL77yF           |
| Client Secret | <div>.....</div> <div>보기</div> |

API 접속시, client ID, client Secret 필요

# 네이버 API

네이버 블로그 검색 API 파이썬 코드

- <https://developers.naver.com/docs/serviceapi/search/blog/blog.md#python>

## 검색 코드

```
1 # 네이버 검색 API 예제 - 블로그 검색
2 import os
3 import sys
4 import urllib.request
5 client_id = "qm83FBIXGeZFWkhBTu3i"
6 client_secret = "6c1vpFYusZ"
7 encText = urllib.parse.quote("커피")
8 url = "https://openapi.naver.com/v1/search/blog?query=" + encText # JSON 결과
9 # url = "https://openapi.naver.com/v1/search/blog.xml?query=" + encText # XML 결과
10 request = urllib.request.Request(url)
11 request.add_header("X-Naver-Client-Id",client_id)
12 request.add_header("X-Naver-Client-Secret",client_secret)
13 response = urllib.request.urlopen(request)
14 rescode = response.getcode()
15 if(rescode==200):
16     response_body = response.read()
17     print(response_body.decode('utf-8'))
18 else:
19     print("Error Code:" + rescode)
```

✓ 0.2s

Python

## 검색 결과

```
{
  "lastBuildDate": "Tue, 27 Feb 2024 19:14:36 +0900",
  "total": 40753857,
  "start": 1,
  "display": 10,
  "items": [
    {
      "title": "한성대 사만 <b>커피</b> 로스터스 라떼 (구 네임드 에스프레소)",
      "link": "https://blog.naver.com/nowwegom/223363998864",
      "description": "나폴레옹 과자점에서 빵을 사고 나서 좀 쉬었다 가고 싶어 들",
      "bloggername": "먹기 위해 사는 위장 환자",
      "bloggerlink": "blog.naver.com/nowwegom",
      "postdate": "20240224"
    },
    {
      "title": "성남 분당 에스프레소 <b>커피</b> 맛집 수내로",
      "link": "https://blog.naver.com/dubungstory/223363900789",
      "description": "내부공간 수내로는 테이블 수가 많지는 않은 아담한 공간의 <",
      "bloggername": "두봉어의 소소한 일상",
      "bloggerlink": "blog.naver.com/dubungstory",
      "postdate": "20240224"
    },
    {
      "title": "신방동카페 통정<b>커피</b>미식가 천안프렌치 토스트 굿",
      "link": "https://blog.naver.com/sensefull7/223365469276",
      "postdate": "20240225"
    }
  ]
}
```

# 네이버 API

검색 결과를 확인 후에, 원하는 데이터를 전처리 해야 합니다.

## 데이터 처리

```
1 import json
2 text_data = response_body.decode('utf-8')
3 json_data = json.loads(text_data)
4 print(json_data)
```

✓ 0.0s Python

{'lastBuildDate': 'Tue, 27 Feb 2024 19:14:36 +0900', 'total': 40753857, 'start': 1, 'display': 10}

## 데이터 확인

```
1 json_data["items"]
```

✓ 0.0s Python

```
[{'title': '한성대 사만 <b>커피</b> 로스터스 라떼 (구 네임드 에스프레소)',
 'link': 'https://blog.naver.com/nowwegom/223363998864',
 'description': '나폴레옹 과자점에서 빵을 사고 나서 좀 쉬었다 가고 싶어 들렀던 사만 <b>커피</b>',
 'bloggername': '먹기 위해 사는 위장 환자',
 'bloggerlink': 'blog.naver.com/nowwegom',
 'postdate': '20240224'},
 {'title': '성남 분당 에스프레소 <b>커피</b> 맛집 수내로',
 'link': 'https://blog.naver.com/dubungstory/223363900789',
 'description': '내부공간 수내로는 테이블 수가 많지는 않은 아담한 공간의 <b>커피</b>숍이에요. 인',
 'bloggername': '두봉어의 소소한 일상',
 'bloggerlink': 'blog.naver.com/dubungstory',
 'postdate': '20240224'},
 {'title': '신방동카페 통정<b>커피</b>미식가 천안프렌치 토스트 굿',
 'link': 'https://blog.naver.com/sensefull7/223365469276',
 'description': '오랫만에 동네 맛난 <b>커피</b>맛집을 찾았다. 오며가며 보긴 했는데 잘 알진 못했던',
 'bloggername': 'sensefull's travel',
 'bloggerlink': 'blog.naver.com/sensefull7',
 'postdate': '20240226'},
```

# 실습 - 3인 1조(공공데이터 or 네이버 API)

공공데이터 or 네이버 API를 활용하여 데이터 수집 진행

예시)

1. 네이버 영화 API 활용: 한국 영화의 변화 분석
  - a. 네이버 블로그 api : 영화 리뷰, 평가, 분석 등의 데이터 수집해보기
  - b. 네이버 뉴스 api : 영화 관련 뉴스 기사 수집해보기