

[JSON] (JavaScript Object Notation)

-속성-값 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷.

-비동기 AJAX을 위해, 넓게는 XML(AJAX가 사용)을 대체하는 주요 데이터 포맷.

-본래는 자바스크립트 언어로부터 파생되어 자바스크립트의 구문 형식을 따르지만 언어 독립형 데이터 포맷. 즉, 프로그래밍 언어나 플랫폼에 독립적, C, C++, C#, JAVA, 자바스크립트, 펄, 파이썬 등 많은 프로그래밍 언어에서 쉽게 이용할 수 있음.1

JSON 객체 기본형태 { "속성명1": "value1" , "속성명2": "value2" }

JSON 객체 배열 기본형태

[{ "name" : "김준서", "addr" : "인천시" }, { "name" : "이승희", "addr" : "서울" }]	[{ "속성명1": "value1", "속성명2": "value2" }, { "속성명1": "value1", "속성명2": "value2" }]
---	--

JSON 예

```
{
  "response":
  {
    "header":
    { "resultCode": "0000", "resultMsg": "OK" },
    "body":
    { "items":
      { "item":
        [
          {
            "addrCd": 1111,
            "csForCnt": 39559,
            "csNatCnt": 52082,
            "gungu": "종로구",
            "resNm": "창덕궁",
            "rnum": 1,
            "sido": "서울특별시",

```

```

        "ym":201612},
        {
        "addrCd":1111,
        "csForCnt":142110,
        "csNatCnt":151317,
        "gungu":"종로구",
        "resNm":"경복궁",
        "rnum":2,
        "sido":"서울특별시",
        "ym":201612
        }
    ]
},
"numOfRows":100,
"pageNo":1,
"totalCount":12
}
}

```

딕셔너리를 JSON으로(json.dumps()) **json으로 바꾸면--> 모두 문자열로 처리

ex01_to_json.py

```

import json

inst = {1:'guitar', 2:'violin', 3:'piano'}
to_json = json.dumps(inst)
print(type(to_json), to_json)

```

실행결과 :

```
<class 'str'> {"1": "guitar", "2": "violin", "3": "piano"}
```

ex01_to_json.py 계속

```

to_json = json.dumps(inst, indent=4, ensure_ascii=False)
                        #indent=4 들여쓰기, ensure_ascii=False 한글처리
print(to_json)

```

실행결과 :

```

{
  "1": "guitar",
  "2": "violin",
  "3": "piano"
}

```

딕셔너리 리스트를 JSON으로(json.dumps())

ex01_to_json.py 계속

```
import json

inst = [ { 1:'guitar', 2:'violin', 3:'piano'},
          {1: 'seoul', 2: 'busan', 3: 'incheon'} ]

to_json = json.dumps(inst)
print(type(to_json), to_json)
```

실행결과 :

```
<class 'str'> [{"1": "guitar", "2": "violin", "3": "piano"}, {"1": "seoul", "2": "busan", "3": "incheon"}]
```

ex01_to_json.py 계속

```
to_json = json.dumps(inst, indent=4, ensure_ascii=False)
                        #indent=4 들여쓰기, ensure_ascii=False 한글처리
print(to_json)
```

실행결과 :

```
[
  {
    "1": "guitar",
    "2": "violin",
    "3": "piano"
  },
  {
    "1": "seoul",
    "2": "busan",
    "3": "incheon"
  }
]
```

JSON을 딕셔너리로(json.loads())

ex02_from_json.py

```
import json
to_json = '{"1": "guitar", "2": "violin", "3": "piano"}' #json 문자열

from_json = json.loads(to_json)
print(type(from_json), from_json)

print(from_json["3"]) # piano
```

실행결과 :

```
<class 'dict'> {'1': 'guitar', '2': 'violin', '3': 'piano'}
piano
```

JSON 배열을 딕셔너리 리스트로(json.loads())

ex02_from_json.py 계속

```
import json
to_json = '[{"1": "guitar", "2": "violin", "3": "piano"}, {"1": "seoul", "2": "busan", "3": "incheon"}]'

from_json = json.loads(to_json)
print(type(from_json), from_json)

print(from_json[0]["2"]) # violin
```

실행결과 : <class 'list'> [{'1': 'guitar', '2': 'violin', '3': 'piano'}, {'1': 'seoul', '2': 'busan', '3': 'incheon'}]

Violin

Open API 를 이용한 데이터 수집

Open API

Open API 는 말 그대로 개발형 API이다.

API가 응용 프로그램을 개발할 때 사용하는 인터페이스라는 의미이므로, Open API는 프로그래밍에서 사용할 수 있는 개방되어 있는 상태의 인터페이스를 말한다.

- Daum, Naver 등의 포털 사이트나 통계청, 기상청 등과 같은 공공서에서도 가지고 있는 데이터를 외부 응용 프로그램에서 사용할 수 있도록 Open API 를 제공하고 있다.

- Open API 와 함께 자주 거론되는 기술이 REST 이며, 대부분 Open API는 REST 방식으로 지원되고 있다.

네이버 포털 API 이용하기

1. open API 인증 키를 발급받아야 함 - (Client ID와 Client Secret 발급)

1) 네이버 로그인 후 다음 사이트 접속

2) <https://developers.naver.com/docs/search/news/>

The screenshot shows the Naver Developers website. At the top, there's a navigation bar with 'NAVER Developers', 'Products', 'Documents', 'Application', 'NAVER D2', and 'Support'. A search bar and a user profile icon are also present. Below the navigation bar, a large blue banner reads 'Documents | 네이버 오픈 API를 이용해 창의적인 애플리케이션을 제작해 보세요.' Below this, the breadcrumb path 'Documents > 서비스 API > 검색 > 뉴스' is shown. On the left sidebar, under '서비스 API', the '검색' (Search) category is selected. The main content area shows the '검색 > 뉴스' page. It includes a description of the REST API and a green button labeled '오픈 API 이용 신청 >' which is highlighted with a red box. Below this, the '1. 준비사항' (Preparation) section lists steps for registering an application and obtaining Client ID and Client Secret.

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 서버 메뉴에 등록하신 애플리케이션 이름으로 서버 메뉴가 만들어집니다.

애플리케이션 이름	test_1 ✓
사용 API	선택하세요. ✓ 검색 ✕
비로그인 오픈 API 서비스 환경	환경 추가 ▼ WEB 설정 ✕ ^ 웹 서비스 URL (최대 10개) http://localhost ✕ + ✓ <small>* 텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.</small>

등록하기 버튼 클릭

Application | API 이용을 위해 애플리케이션을 등록하고 API 설정을 할 수 있습니다.

내 애플리케이션 ^

borabora
test
test1
snsCrawler
snsCrawler1
test_1

애플리케이션 등록
Clova Platform Console β
API 제휴 신청
계정 설정

test_1

개요	API 설정	멤버관리	로그인 통계	API 통계	Playground(Beta)
----	--------	------	--------	--------	------------------

애플리케이션 정보

Client ID	1Jjc0fWhpS6Rc_7UyCj
Client Secret	●●●●●●●● 보기

config.py 에 발급받은 Client ID와 Client Secret 키 복사

config.py

```
client_id = "이곳에 발급받은 client_id 복사"
client_secret = "이곳에 발급받은 client secret 복사"
```

2. 네이버 뉴스를 검색페이지 크롤링 프로그램 작성.

네이버 홈페이지(<https://developers.naver.com/docs/search/news/>)에서 URL 및 파라미터를 확인하고, 크롤링 프로그램을 작성한다.

NAVER Developers Products Documents Application NAVER D2 Support API 상태 Search Here

API 공통 가이드
SDK & Tools
Clova
네이버 아이디로 로그인
지도
파파고
서비스 API

데이터랩
검색
- 블로그
- 뉴스
- 책
- 성인 검색어 관별
- 백과사전
- 영화
- 카툰
- 지식IN
- 지역
- 오락변환
- 웹문서
단축 URL

검색 > 뉴스

네이버 뉴스 검색 결과를 출력해주는 REST API입니다. 비로그인 오픈 API이므로 GET으로 호출할 때 HTTP Header에 애플리케이션 등록 시 발급받은 Client ID와 Client Secret 값을 같이 전송해 주시면 활용 가능합니다.

오픈 API 이용 신청

1. 준비사항

- 애플리케이션 등록: 네이버 오픈 API로 개발하시려면 먼저 'Application-애플리케이션 등록' 메뉴에서 애플리케이션을 등록하셔야 합니다. [자세한 방법 보기]
- 클라이언트 ID와 secret 확인: '내 애플리케이션'에서 등록한 애플리케이션을 선택하면 Client ID와 Client Secret 값을 확인할 수 있습니다.
- API 권한 설정: '내 애플리케이션'의 'API 권한관리' 탭에서 사용하려는 API가 체크되어 있는지 확인합니다. 체크되어 있지 않을 경우 403 에러(API 권한 없음)가 발생하니 주의하시기 바랍니다.

2. API 기본 정보

메서드	인증	요청 URL	출력 포맷
GET	-	https://openapi.naver.com/v1/search/news.xml	XML
GET	-	https://openapi.naver.com/v1/search/news.json	JSON

3. 요청 변수 (request parameter)

요청 변수	타입	필수 여부	기본값	설명
query	string	Y	-	검색을 원하는 문자열로서 UTF-8로 인코딩한다.
display	integer	N	10(기본값), 100(최대)	검색 결과 출력 건수 지정
start	integer	N	1(기본값), 1000(최대)	검색 시작 위치로 최대 1000까지 가능
sort	string	N	sim(기본값), date	정렬 옵션: sim (유사도순), date (날짜순)

<https://openapi.naver.com/v1/search/news.json?query=검색을원하는문자열&display=10&start=1>

4. 크롤링 프로그램을 작성한다.

config.py

```
client_id = "이곳에 발급받은 client_id 복사"
client_secret = "이곳에 발급받은 client secret 복사"
```

naverCrawler.py

```
import urllib.request
import config

def makeURL(sNode, search_text):
    base = "https://openapi.naver.com/v1/search"
    node = "/%s.json" % sNode
    parameters = "?query=%s" % urllib.parse.quote(search_text)
    url = base + node + parameters
    return url

def requestURL(url):
    req = urllib.request.Request(url)
    req.add_header("X-Naver-Client-Id", config.client_id) # open api 키를 header에 추가
    req.add_header("X-Naver-Client-Secret", config.client_secret) # open api 키를 header에 추가
    try:
        response = urllib.request.urlopen(req) # 크롤링
        if response.status == 200: # 200이면 정상 응답
            print("Url Request Success")
            data = response.read().decode('utf-8')

            return data # 크롤링 결과 반환
    except Exception as e: # url로 요청 중 오류가 발생시 실행
        print(e)
        print("Error for URL : %s" %url)
        return None

def main():
    sNode = 'news' # 'news', 'blog', 'cafearticle'
    search_text = input(">검색할 키워드를 입력하세요. : ")

    targetURL = makeURL(sNode, search_text) #1. URL 만들기
    result = requestURL(targetURL) #2. URL로 크롤링
    print(result) #3. 크롤링 결과 출력

main()
```

실행결과:

>검색할 키워드를 입력하세요. : 문대통령

Url Request Success

```
{
  "lastBuildDate": "Mon, 17 Dec 2018 16:06:28 +0900",
  "total": 2399311,
  "start": 1,
  "display": 10,
  "items": [
```



```
{
  "title": "<b>문</b> 의장, 17일부터 UAE·요르단·이스라엘 순방",
  "originallink": "http://www.dt.co.kr/contents.html?article_no=2018121702109957798003&ref=naver",
  "link": "https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=100&oid=029&aid=0002498323",
  "description": "첫 일정은 17일 (이하 현지시간) UAE에서 전몰장병 추념비에 헌화로, 이어 이어 자이드 UAE  
초대 <b>대통령</b> 묘소를 참배한다. 또 아말 압둘라 주르아 알-쿠바이시 UAE 연방평의회 의장과 면담한다.  
<b>문</b> 의장과 알-쿠바이시... ",
  "pubDate": "Mon, 17 Dec 2018 16:04:00 +0900"
},
{
  "title": "'정책 감수성' 언급한 文<b>대통령</b>...포용국가 유지하되 '유연성' 강조",
  "originallink": "http://www.newsis.com/view/?id=NISX20181217_0000504945&cID=10301&pID=10300",
  "link": "https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=100&oid=003&aid=0008966597",
  "description": "<b>문 대통령</b>은 17일 청와대 본관에서 주재한 2019년도 경제정책 방향을 논의하는  
확대경제장관회의 모두 발언에서 "어려움을 호소하는 현장의 목소리에 더욱 감수성 있게 대응  
바란다"고 주문했다. 이어 "최저임금 인상... ",
  "pubDate": "Mon, 17 Dec 2018 16:02:00 +0900"
},
... 중략
```

naverCrawler2.py

- 1) 크롤링 결과 전체 데이터를 파일(검색키워드_naver_news_all.json) 로 저장
- 2) 일부 정보(title, originallink, description)만 파싱하여 출력 및
파일(검색키워드_naver_news.json) 로 저장 프로그램을 작성
- 3) 다음과 같이 UI를 개선

>네이버 검색 프로그램입니다.

>검색할 키워드를 입력하세요.

--> 탄핵

>검색 카테고리를 선택하세요

1. 네이버 뉴스, 2. 네이버 블로그 3. 네이버 카페

--> 네이버 블로그로 검색합니다.

```
import urllib.request
import config

def makeURL(sNode, search_text ):
    base = "https://openapi.naver.com/v1/search"
    node = "/%s.json" % sNode
    parameters = "?query=%s" % urllib.parse.quote(search_text)
    url = base + node + parameters
    return url

def requestURL(url):
    req = urllib.request.Request(url)
    req.add_header("X-Naver-Client-Id", config.client_id)
    req.add_header("X-Naver-Client-Secret", config.client_secret)
    try:
        response = urllib.request.urlopen(req) # 크롤링
        if response.status == 200: # 200이면 정상 응답
            print("Url Request Success")
            data = response.read().decode('utf-8')
            return data # 크롤링 결과 반환
    except Exception as e: # url로 요청 중 오류가 발생시 실행
        print(e)
        print("Error for URL : %s" %url)
        return None
```

```
import json

def main():
    print("네이버 검색 프로그램입니다.")
    search_text = input(">검색할 키워드를 입력하세요.")
    #search_text = '탄핵'

    answer = input(">검색 카테고리를 선택하세요 \n 1. 네이버 뉴스,\n 2. 네이버 블로그 3. 네이버 카페")

    sNode = 'news'
    if answer=='1' :
        sNode = 'news' # 'news', 'blog', 'cafearticle'
    elif answer=='2' :
        sNode = 'blog'
    elif answer=='3' :
        sNode = 'cafearticle'
    print("*****",answer)
    targetURL = makeURL(sNode, search_text ) #1. URL 만들기
    result = requestURL(targetURL) #2. URL로 크롤링
    #print(result) #3. 크롤링 결과 출력

    #4. 크롤링 전체 결과를 파일에 저장
    filename = '%s_naver_%s_all.json' % (search_text, sNode)
    with open(filename, 'w', encoding='utf8') as outfile:
        outfile.write(result)
    print(filename + ' 저장 완료' )

    #5. 크롤링 결과 중 일부만 파싱할 때
    jsonResult = []
    if result != None:
        jsonSearch = json.loads(result) # JSON 문자열을 파이썬 딕셔너리로 L

        for post in jsonSearch['items']: # items에 있는 데이터만
            title = post['title']
            description = post['description']
            if sNode=='news' : #news만 originallink가 있음
                org_link = post['originallink']
            else :
                org_link = ""
            #link = post['link']

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

```
# 6. 파싱 결과 일부만 파일에 저장
filename = '%s_naver_%s.json' % (search_text, sNode)
with open(filename, 'w', encoding='utf8') as outfile:
    outfile.write(retJson)
print(filename + ' 저장 완료')

main()
```

실행결과

네이버 검색 프로그램입니다.

>검색할 키워드를 입력하세요. 서울

>검색 카테고리를 선택하세요

1. 네이버 뉴스, 2. 네이버 블로그 3. 네이버 카페 1

***** 1

Url Request Success

서울_naver_news_all.json 저장 완료

```
[
  {
    "title": "프로미스나인 노지선, 귀여운 볼터치 (엠카운트다운)",
    "description": "프로미스나인 노지선이 20일 오후 <b>서울</b> 마포구 상암동 CJ ENM 센터에서 열린 Mnet '엠카운트다운' 리허설 전 포토타임에 참석하고 있다. 이날 Mnet '엠카운트다운' 녹화에는 전소미, 아이즈, 버스터즈, 프로미스나인...",
    "org_link": "http://sports.hankooki.com/lpage/entv/201906/sp20190621044509136730.htm"
  },
  {
    "title": "'게임돌림픽 2019 골든카드' 17일 <b>서울</b> 상암동 에스플렉스센터 OGN...",
    "description": "사진=조현 SNS [시사매거진=주진현 기자] 17일 <b>서울</b> 상암동 에스플렉스센터 OGN e스타디움에서 열린 '게임돌림픽 2019:골든카드'에 참석한 조현은 과한 노출 의상을 입고 등장했다. 조현은 게임 '리그 오브 레전드...',
    "org_link": "http://www.sisamagazine.co.kr/news/articleView.html?idxno=207288"
  },
  {
    "title": "[기상특보]기상청, 오늘날씨 및 주간-주말날씨 예보, 천둥`번개 동반한 소나기...",
    "description": "15~18시에는 <b>서울</b>.경기남부와 강원남부, 충청북부, 경북북부로 확대되었다가 밤(24시)에 대부분... 내일(22일) <b>서울</b>.경기도와 강원도는 중국 북부지방에 위치한 고기압 가장자리에 들겠으나, 충청도와 남부지방은...",
    "org_link": "http://www.kndaily.co.kr/news/articleView.html?idxno=90953"
  },
]
```

중략..

서울_naver_news_all.json 파일 생성됨

서울_naver_news.json 파일 생성됨

공공 데이터 포털 API 이용하기

Open API

Open API 는 말 그대로 개발형 API이다.

API가 응용 프로그램을 개발할 때 사용하는 인터페이스라는 의미이므로, Open API는 프로그래밍에서 사용할 수 있는 개방되어 있는 상태의 인터페이스를 말한다.

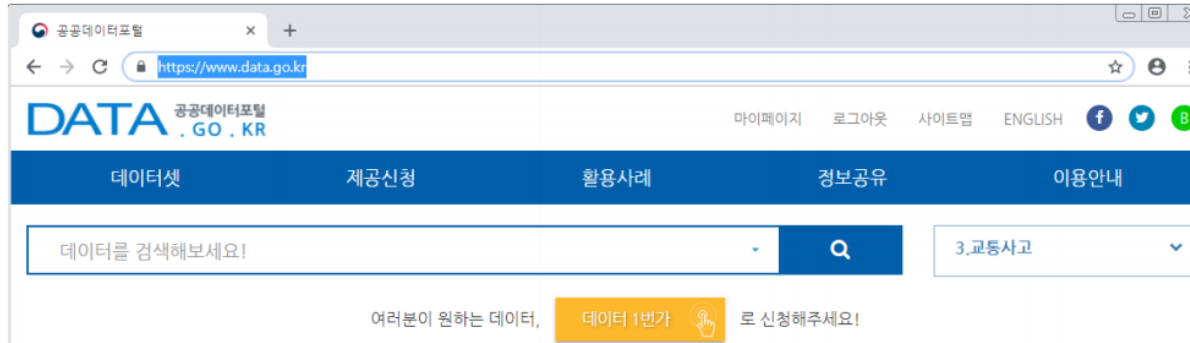
- Daum, Naver 등의 포털 사이트나 통계청, 기상청 등과 같은 관공서에서도 가지고 있는 데이터를 외부 응용 프로그램에서 사용할 수 있도록 Open API 를 제공하고 있다.

- Open API 와 함께 자주 거론되는 기술이 REST 이며, 대부분 Open API는 REST 방식으로 지원되고 있다.

1) open API 인증 키를 발급받아야 함 - 회원 가입 후, 개발자 계정 등록 후 발급.

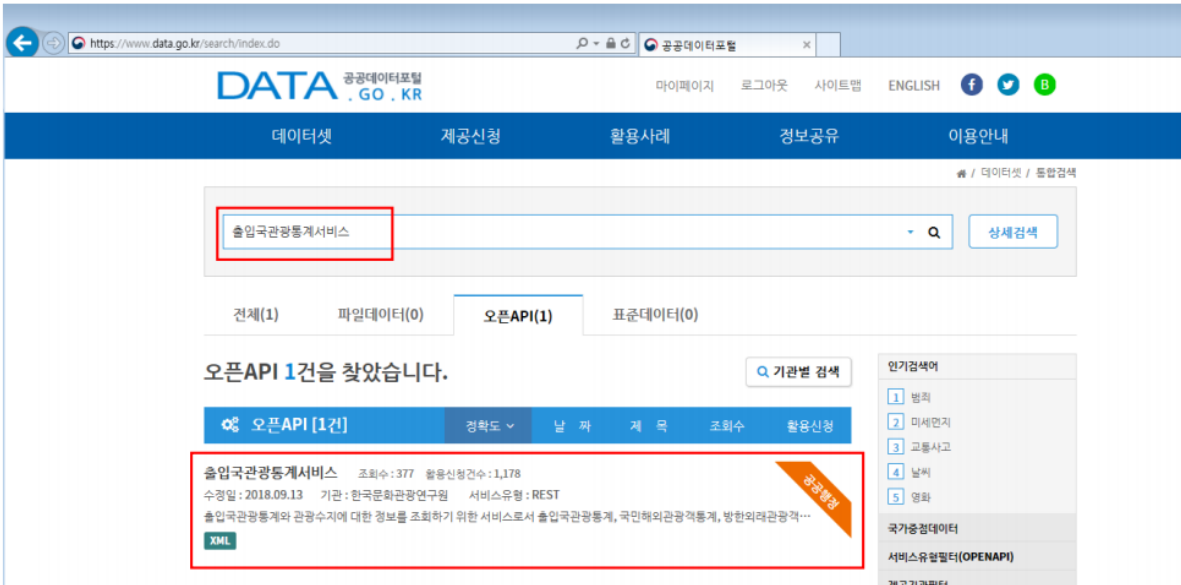
<https://www.data.go.kr/>

(주의 : 키 발급 후, 심사를 거쳐 승인을 받아야 사용 가능한 서비스들 있음. 한시간 이상 소요될 수 있음)





2) “출입국 관광 통계 서비스” 검색



활용 신청을 클릭하세요.

DATA 공공데이터포털
.GO.KR

마이페이지 로그인 사이트맵 ENGLISH

데이터셋 제공신청 활용사례 정보공유 이용안내

데이터셋 / 오픈API

오픈 API 공공행정

- 제공기관
- 한국문화관광연구원
- 관리부서명
- 통계평가센터, 통계평가센터
- 관리부서 전화번호
- 02-2669-8473
- 등록일
- 2012-12-06
- 키워드
- 보유근거
- 행정업무
- 수집방법
- 정보시스템

출입국관광통계서비스

출입국관광통계와 관광수지에 대한 정보를 조회하기 위한 서비스로서 출입국관광통계, 국민해외관광통계, 방한외래관광객통계, 관광수지 등 을 조회할 수 있다.

활용신청 (바로그) 건수: 1,176

※ 서비스 오류가 있을시 오류신고 버튼을 이용해주세요.

XML **출입국관광통계서비스** **활용신청** **닫기** **오류신고**

출입국관광통계서비스			
End Point	http://openapi.tour.go.kr/openapi/service 스크립		
데이터포맷	XML	API 유형	REST
비용부과유무	무료	활용신청 건수	1176
심의유형	개발계정: 자동승인 / 운영계정: 자동승인		
등록일	2012-12-06	수정일	2018-09-13
이용허락범위	공공저작물 _ 출처표시		
참고문서	한국문화관광연구원 출입국관광통계서비스 활용가이드_v1.1.docx 미리보기		
관리부서명	통계평가센터, 통계평가센터	관리부서 전화번호	02-2669-8473

상세기능 **활용사례**

- 출입국관광통계조회
- 방한외래관광객통계조회
- 국민해외관광통계조회
- 관광수지조회
- 방한외래관광객통계조회

요청 url 확인 참고 문서 참고.

마이페이지에서 승인된 서비스 확인 가능

DATA 공공데이터포털
.GO.KR

마이페이지 로그인 사이트맵 ENGLISH

데이터셋 제공신청 활용사례 정보공유 이용안내

마이페이지

오픈API

- 개발계정
- 활용현황
- 운영계정
- 인증키 발급현황

DATA

나의 문의

나의 관심

회원정보

마이페이지 > OPEN API > 개발계정

개발계정

<p>활용자가 개발계정을 신청중인 단계</p> <p>신청 0건 · 반려 0건</p> <p>보류 0건</p>	<p>활용자의 개발계정신청이 승인되어 활용중인 단계</p> <p>활용 2건 · 변경신청 0건</p> <p>반려 0건 · 보류 0건</p>	<p>활용자가 중지신청하여 운영이 중지된 단계</p> <p>중지 0건</p>
---	--	--

상세검색

총 2 건

[승인] 출입국관광통계서비스

서비스 유형: REST 분류: 일반공공행정 > 정부자원관리 신청일: 2018-07-10 [활용신청] 만료예정: 2020-07-10 제공기관: 한국문화관광연구원

키 확인 가능

DATA 공공데이터포털 .GO.KR

마이페이지 | 로그아웃 | 사이트맵 | ENGLISH | f | t | B

데이터셋 | 제공신청 | 활용사례 | 정보공유 | 이용약관

마이페이지

오픈API >

- 개발계정

- 활용현황

- 운영계정

- 인증키 발급현황

DATA

나의 문의

나의 관심

회원정보

공공데이터 제공신청

마이페이지 > OPEN API > 개발계정 상세보기

개발계정 상세보기

운영계정 신청 | 연장 신청 | 변경 신청 | 중지 신청 | 일반 인증키 재발급 | 목록

> 기본정보

서비스명	출입국관광통계서비스 상세설명				
서비스 유형	REST	일일트래픽	0	평균응답속도(초)	0
심의여부	자율승인	신청유형	개발계정 활용신청	처리상태	승인
활용기간	2018-07-10 ~ 2020-07-10				

> 서비스정보

일반 인증키 (UTF-8)	qr26h%2FzivU3LX0%2Ffk56SIf4S67Dus8rTNIQ2G7zn7x5DOOxsjNDt3uh44O32VtUMAJy4MfJ666sUZ9k%2FujSjqQ%3D%3D	복사
End Point	http://openapi.tour.go.kr/openapi/service	
데이터포맷	XML	
참고문서	한국문화관광연구원 출입국관광통계서비스 활용가이드_v1.1.docx	



> 상세기능정보

개발가이드

홈페이지 참고 문서에서 URL 및 파라미터를 확인

3. “관광자원통계서비스” 도 검색하여, 같은 방법으로 활용 신청한다.

DATA 공공데이터포털
 .GO.KR

마이페이지 로그인 사이트맵 ENGLISH
 



데이터셋 제공신청 활용사례 정보공유 이용약관

🔍 / 데이터셋 / 통합검색

전체(1) 파일데이터(0) 오픈API(1) 표준데이터(0)

전체 1건을 찾았습니다.

📁

파일데이터 [0건]

정확도

날 짜

제 목

조회수 ▾

다운로드

🔍

검색결과가 없습니다.

⚙️

오픈API [1건]

정확도

날 짜

제 목

조회수 ▾

활용신청

관광자원통계서비스

조회수 : 394 활용신청건수 : 871

수정일 : 2018.09.13 기관 : 한국문화관광연구원 서비스유형 : REST

전국의 주요 유로관광지 방문객수를 조회하기 위한 서비스로서 기간,지역, 관광자별 외국인 방문객수와 내국인 방문객수를 조...

XML

공공행정

⚙️

표준데이터 [0건]

정확도

날 짜

제 목

조회수 ▾

인기검색어

1

범죄

2

미세먼지

3

날씨

4

주차장

5

연속지적도

국가중점데이터

서비스유형필터

서비스유형필터(OPENAPI)

제공기관필터

분류체계필터

이용허락범위필터

확정자

🔍 / 데이터셋 / 오픈API

관광자원통계서비스

ENGLISH

전국의 주요 유로관광지 방문객수를 조회하기 위한 서비스로서 기간,지역, 관광자별 외국인 방문객수와 내국인 방문객수를 조회할 수 있다.

활용신청 (바로그기) 건수 : 869

※ 서비스 오류가 있을시 오류신고 버튼을 이용해주세요.

XML 관광자원통계서비스

API 신청 닫기 오류신고 ★

관광자원통계서비스			
End Point	http://openapi.tour.go.kr/openapi/service		스크랩
데이터포맷	XML	API 유형	REST
비용부과유무	무료	활용신청 건수	869
심의유형	개발계정: 자동승인 / 운영계정: 자동승인		
등록일	2012-12-06	수정일	2018-09-13
이용허락범위	이용허락범위 제한 없음		
참고문서	한국문화관광연구원 관광자원통계서비스 활용가이드_v1.1.docx		미리보기
관리부서명	통계평가센터,통계평가센터	관리부서 전화번호	02-2669-8473

상세기능 활용사례

· 유로관광지방문객수조회

관광 자원 통계 서비스 중 유료 관광지 방문객 수 조회 기능은 외국인, 내국인 방문객수, 관광지 주소 등의 정보를 제공한다. 지역별 관광지 입장객 수를 분석하기 위하여 다음 코드를 작성

config.py

```
access_key="키복사"
```

public_TourPointVisitor1.py

```
import urllib.request
import config

#유료 관광지 방문객 수

"""
data.go.kr 사이트의 open API는
키를
url의 파라미터(serviceKey=...)로 추가 전달. (참고 - 네이버는 header로 전달)
"""

#유료 관광지 방문객 수 조회를 위한 url 생성
def makeURL(yyyymm, sido, gungu, nPagenum, nItems):
    end_point = "http://openapi.tour.go.kr/openapi/service/TourismResourceStatsService/getPchrgTrrsrtVisitorList"
    parameters = "?_type=json&serviceKey=" + config.access_key # config.py 파일에 있는 변수. access_key="B0j5Y8uGX2Bl.....
                                                                # open api를 사용하기 위한 키. (홈페이지 개발자 등록 후 발급받는다.)

    parameters += "&YM=" + yyyymm # p121 요청 파라미터 참고
    parameters += "&SIDO=" + urllib.parse.quote(sido) # 한글은 인코딩하여 사용
    parameters += "&GUNGU=" + urllib.parse.quote(gungu)
    parameters += "&RES_NM=&pageNo=" + str(nPagenum) # 숫자는 문자로 변경 후 사용
    parameters += "&numOfRows=" + str(nItems)
    url = end_point + parameters

    return url
```

url 생성

```
def requestURL(url):
    req = urllib.request.Request(url)
    try:
        response = urllib.request.urlopen(req) # 요청
        if response.status == 200: # 200이면 정상 응답
            print("Url Request Success")
            data = response.read().decode('utf-8')
            return data # 크롤링 결과 반환
    except Exception as e: # url로 요청 중 오류가 발생시 실행
        print(e)
        print("Error for URL : %s" %url)
        return None
```

요청

```
def main():
    sido = '서울특별시'
    gungu = ''
    nItems = 100
    yyyymm = '201710' # 2017년 10월 조회
    nPagenum = 1

    targetURL = makeURL(yyyymm, sido, gungu, nPagenum, nItems) # 1. URL 만들기
    result = requestURL(targetURL) # 2. URL로 크롤링
    print(result) # 3. 크롤링 결과 출력
```

main()

2011~2017년 유료 관광지 방문객 수 추이를 분석하고자 한다면,

- * 이 사이트는 달별로만 요청 가능. 즉. 201101 ~ 201712 까지 반복적으로 요청해야 함.
- * 이 사이트는 하나의 요청 결과가 100개 이상이면 pageNo=2 로 다음 페이지 요청해야 전체 결과를 받을 수 있음. totalCount를 응답으로 제공하므로 전체 데이터가 100 이상인 경우 총 페이지 수를 계산하여 여러번 반복해야 함. 예) totalCount가 1000이면 pageNo=10까지.

```
public_TourPointVisitor2.py
import urllib.request
import config
import json
import math

#유료 관광지 방문객 수 조회를 위한 url 생성
def makeURL(yyyyymm, sido, gungu, nPagenum, nItems):
    end_point = "http://openapi.tour.go.kr/openapi/service/TourismResourceStatsService/getPchrgTrrsrtVisitorList"
    parameters = "?_type=json&serviceKey=" + config.access_key # config.py 파일에 있는 변수. access_key="B0j5Y8uG%2B1..."
    parameters += "&YM=" + yyyyymm # p121 요청 파라미터 참고
    parameters += "&SIDO=" + urllib.parse.quote(sido) # 한글은 인코딩하여 사용
    parameters += "&GUNGU=" + urllib.parse.quote(gungu)
    parameters += "&RES_NM=&pageNo=" + str(nPagenum) # 숫자는 문자로 변경 후 사용
    parameters += "&numOfRows=" + str(nItems)
    url = end_point + parameters
    return url

def requestURL(url):
    req = urllib.request.Request(url)
    try:
        response = urllib.request.urlopen(req) # 크롤링
        if response.status == 200: # 200이면 정상 응답
            print("Url Request Success")
            data = response.read().decode('utf-8')
            return data # 크롤링 결과 반환
    except Exception as e: # url로 요청 중 오류가 발생시 실행
        print(e)
        print("Error for URL : %s" %url)
        return None

def getTourPointData(item, yyyyymm, jsonResult):
    addrCd = 0
    if 'addrCd' in item:
        addrCd = item['addrCd']

    gungu = ''
    if 'gungu' in item:
        gungu = item['gungu']
```

item 딕셔너리에서 필요한 데이터만 꺼내서
jsonResult에 저장

```

sido = ''
if 'sido' in item:
    sido = item['sido']

resNm = ''
if 'resNm' in item:
    resNm = item['resNm']

rnum = ''
if 'rnum' in item:
    rnum = item['rnum']

ForNum = ''
if 'csForCnt' in item:
    ForNum = item['csForCnt']

NatNum = ''
if 'csNatCnt' in item:
    NatNum = item['csNatCnt']

jsonResult.append({'yyyyymm': yyyyymm, 'addrCd': addrCd,
                  'gungu': gungu, 'sido': sido, 'resNm': resNm,
                  'rnum': rnum, 'ForNum': ForNum, 'NatNum': NatNum})

return

```

```

def main():
    sido = '서울특별시'
    gungu = ''
    nPagenum = 1
    nItems = 100
    nStartYear = 2011
    nEndYear = 2018
    jsonResult = []

```

201101 ~ 201712 까지
반복적으로 요청

```

for year in range(nStartYear, nEndYear): #시작년도부터 종료년도(endyear-1) 까지 반복 요청
    for month in range(1, 13): #1월~12월까지 반복

```

```
# 201101, 201102, 201103, 201104 .... 을 하나씩 생성
```

```
yyyyymm =str(year)
```

```
if month < 10:
```

```
    yyyyymm = yyyyymm+'0' +str(month)
```

```
else:
```

```
    yyyyymm = yyyyymm +__str(month)
```

```
nPagenum = 1
```

전체 데이터가 100 이상인 경우 총 페이지 수를 계산하여 여러번 반복요청

```
while True:
```

```
    targetURL = makeURL(yyyyymm, sido, gungu, nPagenum, nItems) # 1. URL 만들기
```

```
    result = requestURL(targetURL) # 2. URL로 크롤링
```

```
    print(result) # 3. 크롤링 결과 출력
```

```
    if (result != None):
```

```
        jsonData = json.loads(result)
```

Dictionary 로

```
if (jsonData['response']['header']['resultMsg'] == 'OK'): # 요청에 대한 응답이 정상이면
```

```
    nTotal = jsonData['response']['body']['totalCount'] # 요청 결과 건수
```

```
    if nTotal == 0: # 해당 페이지에 데이터가 없는 경우 반복 나눔.
```

```
        break
```

```
    for item in jsonData['response']['body']['items']['item']: # 응답 결과 건수 만큼 반복 돌면서 data 저장
```

```
        getTourPointData(item, yyyyymm, jsonResult)
```

```
    nPage = math.ceil(nTotal / 100) # 이 사이트는 한번에 100개의 데이터를 요청.
```

```
    if (nPagenum == nPage): # 전체 데이터가 100 이상인 경우 총 페이지 수를 계산하여 여러번 반복해야 함
```

```
        break
```

```
    nPagenum += 1
```

```
    # nPagenum를 증가 시켜가면 계속 요청
```

```
else:
```

```
    break
```

```
fileName = '%s_관광지입장정보_%d_%d.json' % (sido, nStartYear, nEndYear - 1
```

jsonResult를 파일로 저장

```
with open(fileName, 'w',
```

```
    encoding='utf8') as outfile: # 파일을 저장하기위해 파일 오픈(파일 없는 경우 생성된다)
```

```
    retJson = json.dumps(jsonResult, # jsonResult(딕셔너리, 리스트)를 JSON 형태의 문자열로 생성
```

```
        indent=4, # indent를 주면 들여쓰기
```

```
        ensure_ascii=False) # 한글 저장 시 ensure_ascii=False
```

```
    outfile.write(retJson) # 파일에 저장
```

```
print(fileName, " 저장 성공")
```

실행결과 :

다음과 같은 "서울특별시_관광지입장정보_2011_2017.json" 파일이 생성됨

```
[
  {
    "yyyyymm": "201101",
    "addrCd": 1111,
    "gungu": "종로구",
    "sido": "서울특별시",
    "resNm": "창덕궁",
    "rnum": 1,
    "ForNum": 14137,
    "NatNum": 43677
  },
  {
    "yyyyymm": "201101",
    "addrCd": 1111,
    "gungu": "종로구",
    "sido": "서울특별시",
    "resNm": "운현궁",
    "rnum": 2,
    "ForNum": "",
    "NatNum": 6523
  },
  ... .. 중략
]
```