

Open API

Application Programming Interface

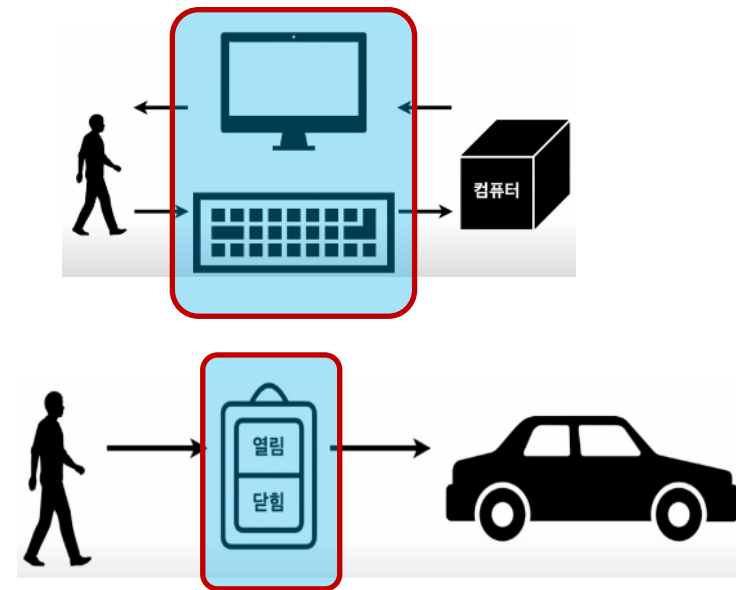


API 개요

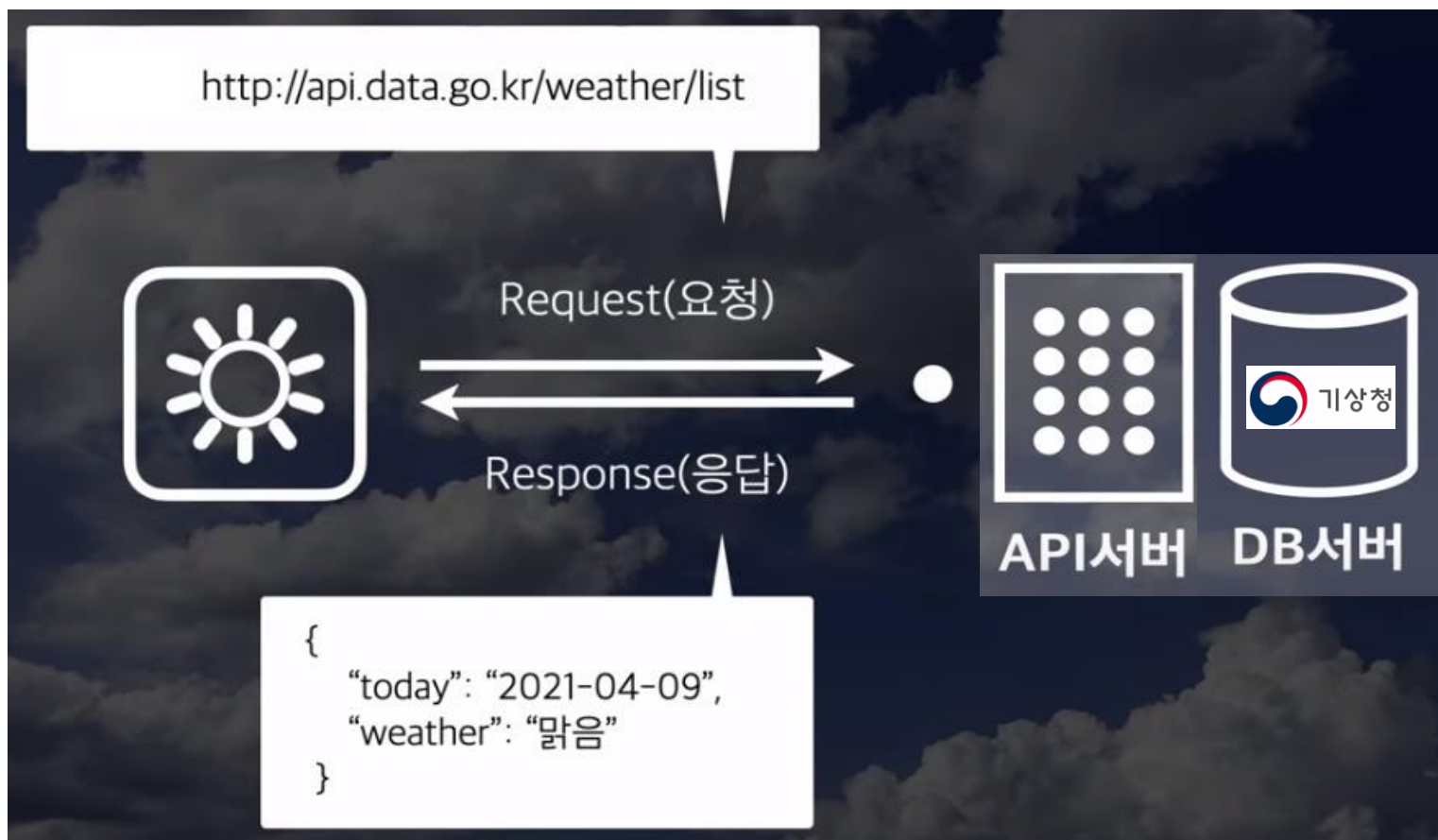
API(Application Programming Interface)는 두개의 소프트웨어가 서로 정보를 주고 받을 수 있는 인터페이스.

Interface

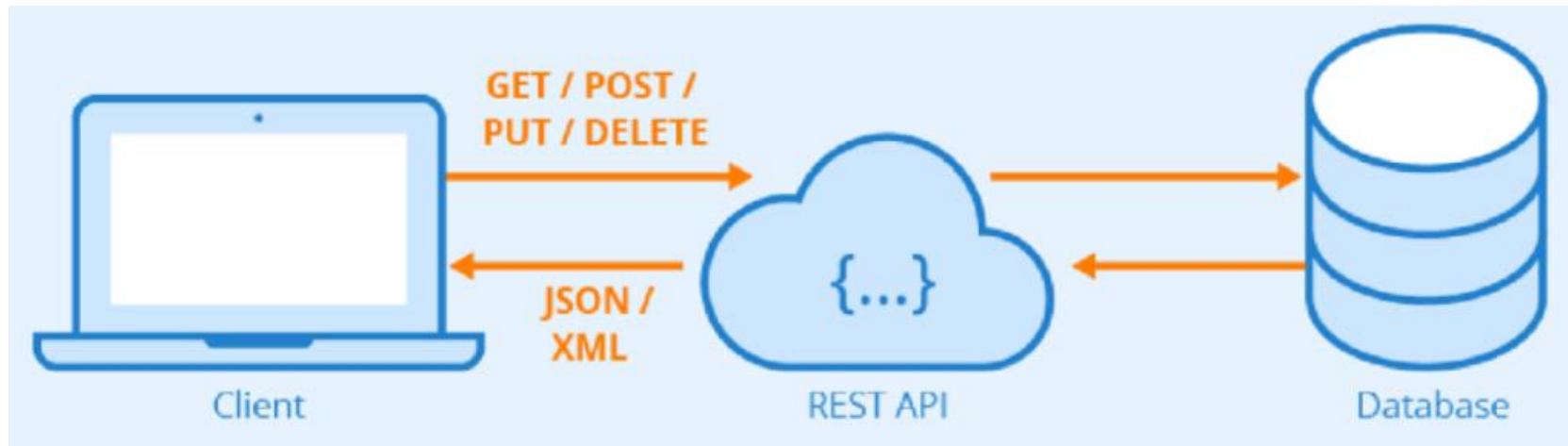
키보드나 마우스 같은 사람과 컴퓨터를 연결하는 장치를 의미하는데,
"Programming Interface"는 그것을 프로그램이라는 것을 통해 두개의 소프트웨어를 연결해 주는 것.



예를 들어 날씨 정보를 표시하는 앱을 개발하려고 한다. 그렇다면 날씨 정보를 내가 직접 온도계를 설치해서 작업할 필요 없이 기상청에서 데이터를 받아서 만들면 된다.



즉, "어떠한 방식으로 정보를 요청해야 하는지, 그리고 그러한 요청을 보냈을 때 어떠한 형식으로 무슨 데이터를 전달받을 수 있는지"에 대해 정리한 일종의 규격이라고 볼 수 있습니다.



API의 유형

<https://blog.naver.com/rankine951/222911260539>

REST API (Representational State Transfer API):

- REST는 "Representational State Transfer"의 약자로, 웹 서비스를 구현하는 데 사용되는 소프트웨어 아키텍처 스타일입니다.
- REST API는 HTTP 프로토콜을 사용하여 자원을 식별하고 상태를 전달하는 데 사용됩니다. 각 자원은 고유한 URI(Uniform Resource Identifier)로 식별되며, HTTP 메서드(GET, POST, PUT, DELETE 등)를 사용하여 해당 자원에 대한 작업을 수행합니다.
- REST API는 간단하고 유연한 구조를 가지며, 대부분의 웹 서비스에서 널리 사용됩니다. JSON 또는 XML과 같은 표준 데이터 형식을 사용하여 데이터를 전송합니다.

SOAP API (Simple Object Access Protocol API):

- SOAP는 "Simple Object Access Protocol"의 약자로, 웹 서비스를 통합하는 데 사용되는 프로토콜입니다.
- SOAP API는 XML 기반의 메시지를 사용하여 서버와 클라이언트 간에 통신합니다. SOAP API는 WSDL(Web Services Description Language)라는 문서를 사용하여 서비스의 기능 및 요구 사항을 설명합니다.
- SOAP API는 보안 및 트랜잭션 관리와 같은 고급 기능을 제공할 수 있으며, 기존의 기업 시스템과 통합하는 데 많이 사용됩니다.

REST API는 무슨 뜻인가요?

클라이언트와 서버는 서로 URL을 통해 응답한다.

REST는 웹서비스 상에 존재하는 다양한 리소스들에 고유주소를 지정하고 그 주소로 요청하면 응답하는 방식이다. 즉, 웹 사이트의 이미지, 텍스트, DB 내용 등의 모든 자원에 고유한 ID인 HTTP URL를 부여한다

JSON 혹은 XML를 통해 데이터를 주고 받는 것이 일반적이다.

참고 하세요 > <https://hanamon.kr/rest-api/>

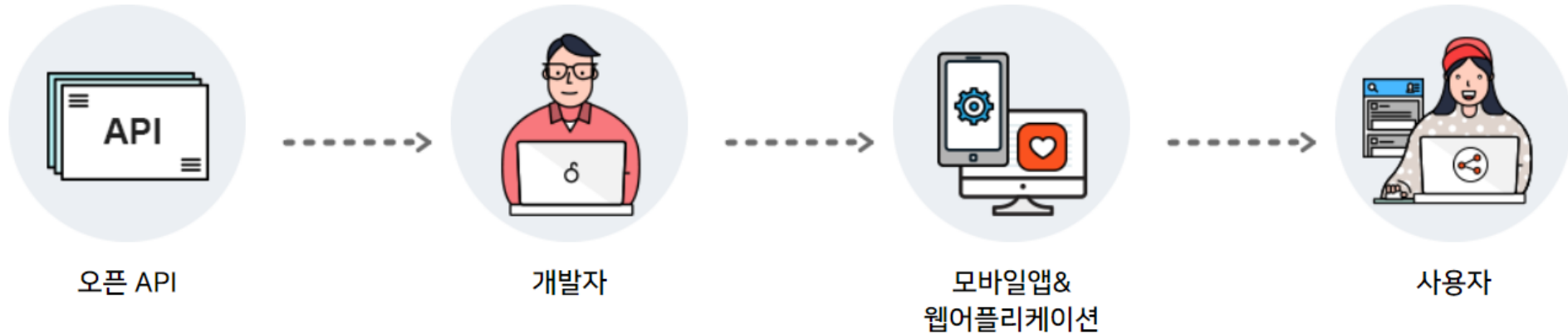
> 오픈API란?

오픈API란 누구나 사용할 수 있도록 공개된 API를 말합니다. 데이터를 표준화하고 프로그래밍해 외부 소프트웨어 개발자나 사용자가 바로 개발(어플리케이션)에 활용할 수 있는 형태의 개방 형식입니다. 개방된 오픈API를 이용해 다양하고 재미있는 서비스나 애플리케이션, 다양한 형태의 플랫폼을 개발할 수 있습니다.

(*API란? Application Programming Interface)

> 오픈 API 이럴때 유용해요!

- 업데이트가 빈번하고 활용도가 높은 대용량의 데이터를 연계 개발할 때
- 날씨나 교통 정보 등 실시간 업데이트되는 데이터를 연계가 필요할 때



인터넷으로 컴퓨터끼리 데이터를 주고 받기

전달할 데이터

학생들 운동 선호도 조사		
	학생수	구성비
축구	20	28.6%
농구	12	17.1%
야구	12	17.1%
탁구	14	20.0%
피구	12	17.1%
합계	70	100.0%

방법1 - 문자열

"축구 20 28.6% 농구 12 17.1% 야구 12 17.1%"

방법2 - XML

```
<축구>
  <학생수>20</학생수>
  <구성비>28.6%</구성비>
</축구>
```

방법3 - CSV

종목, 학생수, 구성비
축구, 20, 28.6%

방법4 - JSON

```
{"종목":"축구","학생수":20,"구성비":28.6%}
```

방법5 - YAML

```
종목:
  종목명: "축구"
  학생수: 20
  구성비: 26.8%
```

XML(Extensible Markup Language)이란?

- 데이터를 저장하고 전송하기 위한 마크업 언어이다.
- HTML과 마찬가지로 사람과 기계가 읽을 수 있는 텍스트 형식이다. 하지만, XML은 HTML과 다르게 데이터를 보여주는 목적이 아닌, 데이터를 저장하고 전달할 목적으로만 만들어졌다
- 시작 태그와 종료 태그를 사용해서 표현한다. 각 요소는 속성과 값을 가질 수 있다.

```
<users>
  <user>
    <id >1</id>
    <name>John Doe</name>
    <email>john.doe@example.com</email>
  </user>
  <user>
    <id>2</id>
    <name>Alice Smith</name>
    <email>alice.smith@example.com</email>
  </user>
</users>
```

- * users: 최상위 요소이고 하위에 user가 있다.
- * id, name, email : user의 하위 요소이다.

이처럼 xml은 데이터를 구조화 시켜서 표현하는 언어이다.

XML은 헤더와 태그 등 여러 요소들을 표현하다보니 **가독성이 떨어지고 용량도 효율적이지 않다**는 평가를 받는다.

```
<?xml version="1.0" encoding="utf-8"?>
<data>
  <people>
    <person name="Rory" surname="Becker" age="30" >
      <comment>A programmer whose interested in all sorts of misc
stuff. His Blog can be found at http://rorybecker.blogspot.com and he's on
twitter as @RoryBecker</comment>
    </person>
    <person name="Travis" surname="Illig" age="32" >
      <comment>A cool guy for who has helped me out with all sorts of
SVn information</comment>
    </person>
    <person name="Scott" surname="Hanselman" age="34" >
      <comment>Scott works for MS and has a great podcast available
at http://www.hanselminutes.com </comment>
    </person>
  </people>
</data>
```

JSON(JavaScript Object Notation)이란?

- 키(Key)와 값(Value)으로 데이터를 표현한다.
- 데이터를 저장하거나 전송할 때 많이 사용되는 경량의 **DATA** 교환 형식
- JSON 표현식은 사람과 기계 모두 이해하기 쉬우며 용량이 작아서, **최근에는 JSON이 XML을 대체해서 데이터 전송 등에 많이 사용한다.**

```
{  
  "firstName": "Kwon",  
  "lastName": "YoungJae",  
  "email": "kyoje11@gmail.com"  
}
```

CSV(comma separated value)란?

- 데이터를 콤마로 구분해서 저장한 텍스트 파일이다.
- 읽기 쉽고 간단한 형식이다.

```
no, name, age, sex, height  
1234, 이준하, 27, 남, 177  
1515, 김준준, 41, 남, 188  
7419, 홍하하, 11, 여, 144
```

YAML(YAML Ain't Markup Language)란?

- key-Value 구성으로 작성된 파일
- json은 중괄호를 이용해 데이터간의 구분을 표현하지만, yaml은 띄어쓰기로 데이터 구분

XML

```
1 <Servers>
2   <Server>
3     <name>Server1</name>
4     <owner>Prajwal</owner>
5     <status>active</status>
6   </Server>
7   <Server>
8     <name>Server2</name>
9     <owner>John</owner>
10    <status>inactive</status>
11  </Server>
12 </Servers>
```

JSON

```
1 {
2   "Servers": [
3     {
4       "name": "Server1",
5       "owner": "Prajwal",
6       "status": "active"
7     },
8     {
9       "name": "Server2",
10      "owner": "John",
11      "status": "inactive"
12    }
13  ]
14 }
```

YAML

```
1 Servers:
2   - name: Server1
3     owner: Prajwal
4     status: active
5   - name: Server2
6     owner: John
7     status: inactive
```

requests 라이브러리

Python 외부 라이브러리로 파이썬에서 HTTP 요청을 쉽게 보내고 받을 수 있도록 도와주는 라이브러리 중 하나이다.

requests 라이브러리 설치

```
pip install requests
```

requests 라이브러리 import

```
import requests
```


requests로 네이버 페이지에 접속하고 페이지 받아보기

- url로 요청하기: `requests.get(url)`

```
import requests

url="https://www.naver.com"
response=requests.get(url)
print(f"응답 코드:{response.status_code}")      # 응답 코드:200
print(response.text)
```

HTTP 응답 상태 코드

•2xx (성공):

- **200 OK:** 성공적으로 처리됨.
- **201 Created:** 새로운 리소스가 성공적으로 생성됨.
- **204 No Content:** 요청이 성공했지만 응답 바디에 콘텐츠가 없음.

•3xx (리다이렉션):

- **301 Moved Permanently:** 요청한 페이지가 새 위치로 영구적으로 이동됨.
- **302 Found (또는 307 Temporary Redirect):** 요청한 페이지가 일시적으로 새 위치로 이동됨.

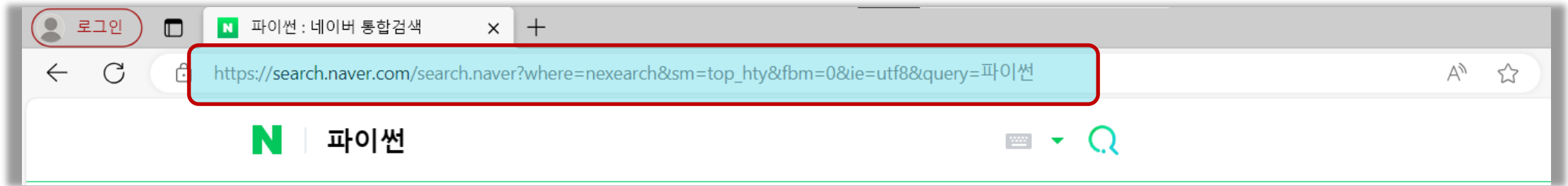
•4xx (클라이언트 오류):

- **400 Bad Request:** 잘못된 요청으로 서버가 요청을 이해할 수 없음.
- **401 Unauthorized:** 인증이 필요함.
- **403 Forbidden:** 요청이 서버에 의해 거부됨.
- **404 Not Found:** 요청한 리소스가 서버에서 찾을 수 없음.

•5xx (서버 오류):

- **500 Internal Server Error:** 서버가 요청을 처리하는 동안 오류가 발생함.
- **502 Bad Gateway:** 게이트웨이나 프록시가 업스트림 서버로부터 유효하지 않은 응답을 받음.
- **503 Service Unavailable:** 서버가 현재 사용 불가능하며 곧 복구될 것으로 예상됨.

검색어를 입력했을 때 주소창의 상태



`https://search.naver.com/search.naver?` `where=nexearch&sm=top_h ty&fbm=0&ie=utf8&` `query=파이썬`

- 물음표(?) 뒷부분이 사용자가 서버에게 요청하는 정보가 전달되는 구역이다.
- 요청 매개변수(파라미터) : where, sm, fbm, ie, query
- 요청 매개변수가 여러 개이면 &로 연결한다.

파이썬에서 requests.get()에 매개변수 지정해서 요청하기

`requests.get(url, params,...)` 함수

url: 요청을 보낼 URL을 지정.

params: URL에 추가할 쿼리 매개변수를 딕셔너리 형태로 지정. 예를 들어, `params={"key": "value"}`

```
import requests
```

```
# 방법1
```

```
url="https://search.naver.com/search.naver?query=파이썬"
```

```
response=requests.get(url)
```

```
print(response.text)
```

```
# 방법2
```

```
url="https://search.naver.com/search.naver"
```

```
param={"query": "파이썬"}
```

```
response=requests.get(url, params=param)
```

```
print(response.text)
```

pandas 라이브러리 활용

Python 외부 라이브러리로 데이터 구조화하고 분석하는 데 사용된다.

pandas 라이브러리 설치

```
pip install pandas
```

pandas 라이브러리 import

```
import pandas as pd
```

pandas로 데이터 프레임 만들고 csv 파일로 저장하기

```
import pandas as pd

name='kim lee park go han youn'.split()
addr='서울 경기 서울 제주 청주 평창'.split()
data=list(zip(name,addr))

df=pd.DataFrame(data)
print(df)

df.to_csv('test.csv',index=False,header=None)
```

pandas로 데이터 프레임 만들고 csv 파일로 저장하기

```
import pandas as pd
from faker import Faker
```

```
fake=Faker('ko-KR')
test_data=[(fake.name(),fake.address()) for i in range(30)]
# print(test_data)
```

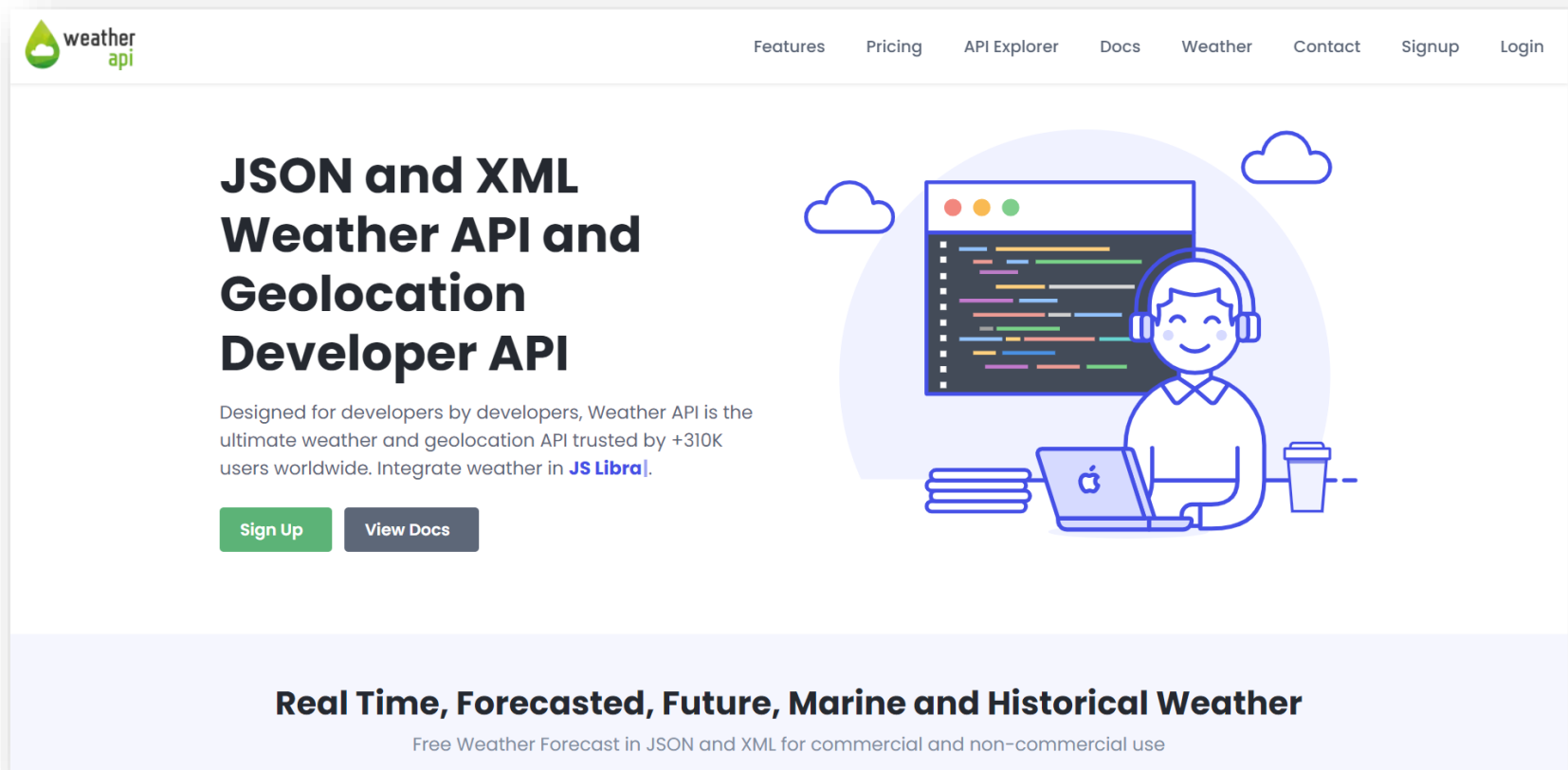
```
df=pd.DataFrame(test_data)
print(df)
df.to_csv('customers.csv',index=False,header=None)
```

이영길,전라남도 청주시 서원구 서초중앙거리 (현준김읍)
이지훈,경상북도 안산시 단원구 양재천거리
곽지민,충청남도 시흥시 가락길
최상현,인천광역시 마포구 영동대로 (수민이한마을)
최서준,경상남도 서산시 테헤란거리 (서영황이마을)
이광수,충청북도 청양군 선릉가
손준호,강원도 계룡시 개포41길
윤희남,부산광역시 중구 봉은사가

weatherapi.com API

실시간 날씨 정보 조회

1. weatherapi.com 에 회원 가입한다. 이메일로 인증을 해야 가입이 완료된다.
2. weatherapi.com 사이트 우측상단 my account 클릭하면 api key키가 보인다. 이 키를 복사한다.



- api key 아래 조금만 스크롤하면 Get Started에 "API documentation"을 클릭한다.

The screenshot shows the Weather API dashboard. On the left is a dark sidebar with navigation links: Dashboards, API Response Fields, Analytics, Accounts (expanded), Change Plan, Payment Method, Billing, Support, API Explorer, Swagger Tool, Change Password, and Log out. The main content area has a 'Welcome Back' header. Below it, the API Key is displayed as a blacked-out string with a 'Copy' button, a 'LIVE' status, and a 'TRIAL Ends on 21/Oct/2023' badge. Four colored boxes show account stats: 'Pro Plus Plan' (green), '5,000,000 Calls per Month' (orange), '0 Calls Made' (pink), and '21/Oct/23 Trial End Date' (purple). A note states: 'Note: If you are on a trial plan then after the trial plan ends your API key will be automatically moved to Free plan if you do not wish to upgrade to a paid plan.' Below this is a '☆ Get Started' section with a list of links. A red checkmark is placed over the 'API documentation' link, which is also highlighted with a red box. Other links in the list include 'Learn how to form HTTP request to get weather from API Explorer or use Swagger Tool.', 'Complete weather lookup code list.', 'Weather icons and weather lookup code list.', 'Want to choose which weather field to return in the API response? Change it from API response fields.', and 'Looking to upgrade/downgrade your API plan? Visit our Upgrade/Downgrade plan section.' At the bottom left is a 'Regenerate API Key' button. At the bottom right is a teal 'Contact us' button with an envelope icon.

weather api

⌵

Dashboards

API Response Fields

Analytics

Accounts ▾

Change Plan

Payment Method

Billing

Support

API Explorer

Swagger Tool

Change Password

Log out

Welcome Back

API Key: [REDACTED] Copy LIVE TRIAL Ends on 21/Oct/2023

Pro Plus Plan

5,000,000 Calls per Month

0 Calls Made

21/Oct/23 Trial End Date

Note: If you are on a trial plan then after the trial plan ends your API key will be automatically moved to Free plan if you do not wish to upgrade to a paid plan.

☆ Get Started

- Learn how to form HTTP request to get weather from API Explorer or use Swagger Tool.
- Complete weather [API documentation](#).
- Weather icons and weather lookup [code list](#).
- Want to choose which weather field to return in the API response? Change it from API [response fields](#).
- Looking to upgrade/downgrade your API plan? Visit our [Upgrade/Downgrade plan section](#).

⌚ Regenerate API Key

Recent Activities

Account created.

🕒 07-Oct-2023

✉ Contact us

- 화면을 스크롤해서 "Example"에 요청 url 예시가 보인다. 복사한 후 아래 코드를 작성한다.

The image shows a screenshot of the weatherapi.com website. The page is titled "Introduction" and features a sidebar with navigation links. The main content area is titled "Example" and contains a list of API request examples. A red box highlights the first example, which is a JSON request for current weather in London. A red checkmark is placed above the highlighted URL.

weather api

Features Pricing API Explorer Contact My Account

Introduction

Getting Started

Authentication

Request

Location Object

Weather Alerts **NEW**

Air Quality **NEW**

Bulk Request **NEW**

API Error Codes

APIs

Realtime API

Forecast API

History API

Marine API **NEW**

Future API **NEW**

Search/Autocomplete API

IP Lookup API

WeatherAPI.com web and mobile API

We provide

- Real-time weather data
- 14 day weather forecast
- Historical weather data
- Marine weather data
- Future weather data
- Daily and hourly weather data
- 15 min interval weather data
- Astronomical data
- Time zone data

match string Match name

Example

WeatherAPI.com API is so easy to implement. Look at following examples on how you can form a request to get data either through a web browser or in your application.

1. So to get current weather for London: JSON: `http://api.weatherapi.com/v1/current.json?key=<YOUR_API_KEY>&q=London`
- XML: `http://api.weatherapi.com/v1/current.xml?key=<YOUR_API_KEY>&q=London`
2. To get 7 day weather for US Zipcode 07112: JSON: `http://api.weatherapi.com/v1/forecast.json?key=<YOUR_API_KEY>&q=07112&days=7`
- XML: `http://api.weatherapi.com/v1/forecast.xml?key=<YOUR_API_KEY>&q=07112&days=7`
3. Search for cities starting with Lond: JSON: `http://api.weatherapi.com/v1/search.json?key=<YOUR_API_KEY>&q=lond`
- XML: `http://api.weatherapi.com/v1/search.xml?key=<YOUR_API_KEY>&q=lond`

Integrations

Please use our [API Explorer](#) to see how the request is formed and what response to expect

Contact us

방법1

```
import requests
```

```
url="http://api.weatherapi.com/v1/current.json?key=발급받은키&q=seoul"  
response=requests.get(url)  
print(response.text)
```

q=36.5684,128.7293

q=Seoul,kr # 도시명, 국가

방법2

```
import requests
```

```
url="http://api.weatherapi.com/v1/current.json"  
param={"key":"발급받은키","q":"seoul"}  
response=requests.get(url,params=param)  
print(response.text)
```

실행 결과

```
{"location":{"name":"Seoul","region":"","country":"South Korea","lat":37.57,"lon":127.0,"tz_id":"Asia/Seoul","localtime_epoch":1701747389,"localtime":"2023-12-05 12:36"},"current":{"last_updated_epoch":1701747000,"last_updated":"2023-12-05 12:30","temp_c":9.0,"temp_f":48.2,"is_day":1,"condition":{"text":"Sunny","icon":"//cdn.weatherapi.com/weather/64x64/day/113.png","code":1000},"wind_mph":5.6,"wind_kph":9.0,"wind_degree":120,"wind_dir":"ESE","pressure_mb":1018.0,"pressure_in":30.06,"precip_mm":0.0,"precip_in":0.0,"humidity":43,"cloud":0,"feelslike_c":9.1,"feelslike_f":48.5,"vis_km":10.0,"vis_miles":6.0,"uv":3.0,"gust_mph":10.1,"gust_kph":16.2}}
```

```
▼ {  
  ▼ location : {  
    name : Seoul  
    region : value  
    country : South Korea  
    lat : 37.57  
    lon : 127.0  
    tz_id : Asia/Seoul  
    localtime_epoch : 1701747799  
    localtime : 2023-12-05 12:43  
  }  
  ▼ current : {  
    last_updated_epoch : 1701747000  
    last_updated : 2023-12-05 12:30  
    temp_c : 9.0  
    temp_f : 48.2  
    is_day : 1  
    ▼ condition : {  
      text : Sunny  
      icon : //cdn.weatherapi.com/weather/64x64/day/113.png  
      code : 1000  
    }  
    wind_mph : 5.6  
    wind_kph : 9.0  
    wind_degree : 120  
  }  
}
```

```
import requests

url="http://api.weatherapi.com/v1/current.json"
param={"key": "발급받은 인증키", "q": "seoul"}
response=requests.get(url,params=param)

print(type(response.text),response.text)
result=response.json() # str -> dict 변환
# print(type(result),result)

print(f"나라명:{result['location']['country']}")
print(f"위도, 경도:{result['location']['lat']}, {result['location']['lon']}")
print(f"현재 기온(섭씨):{result['current']['temp_c']}")
print(f"날씨 상태:{result['current']['condition']['text']}")
```

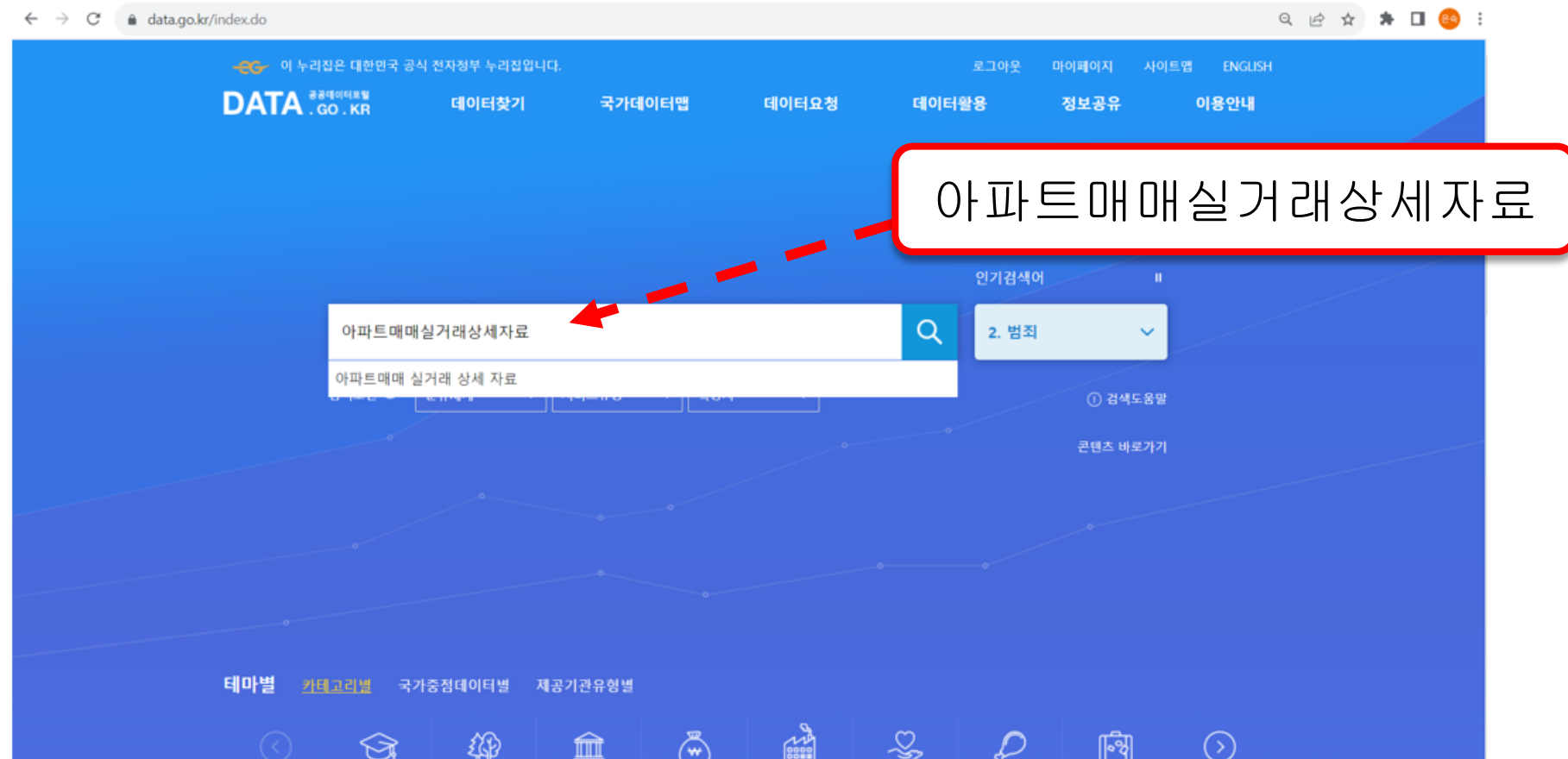
```
나라명:South Korea
위도, 경도:37.57, 127.0
현재 기온(섭씨):9.0
날씨 상태:Partly cloudy
```

data.go.kr API

아파트 매매 실거래가 조회

1. 공공데이터 포털 접속 (data.go.kr)
2. 회원가입

1. 검색창에 검색한다.



2."오픈 API(~)"를 클릭하고 맨 위에 있는 목록 오른쪽에 있는 "활용신청"을 클릭한다.

전체(11,743건) 파일데이터(10,511건) **오픈 API(1,232건)** 표준데이터셋0개(0건)

정확도순 ▾ 10개씩 ▾ 정렬

오픈 API (1,232건)

공공행정 국가행정기관 국가중점

XML 국토교통부_아파트매매 실거래 상세 자료 [미리보기](#)

부동산 거래신고에 관한 법률에 따라 신고된 주택의 실거래 자료를 제공

제공기관 국토교통부 수정일 2023-08-31 조회수 115595 활용신청 17104 키워드 주택,아파트,실거래가 [활용신청](#)

국토관리 자치행정기관

XML JSON 경기도 광명시_아파트매매 실거래 상세 자료 현황 [미리보기](#)

광명시 아파트매매 실거래 상세 자료 현황입니다

제공기관 경기도 광명시 수정일 2018-11-08 조회수 2836 활용신청 379 키워드 국가중점데이터,아파트매매,아파트,매매,매매 상세,아파트 상세,실거래 [바로가기](#)

공공행정 국가행정기관 국가중점

XML 국토교통부_아파트매매 실거래자료 [미리보기](#)

부동산 거래신고에 관한 법률에 따라 신고된 주택의 실거래 자료를 제공

제공기관 국토교통부 수정일 2023-08-31 조회수 73241 활용신청 17567 키워드 주택,아파트,실거래가 [활용신청](#)

[민원서비스 게시판](#)

3. 신청이 완료되면 “개발계정 상세보기” 화면에 보면 “서비스정보”에 인증키가 생성된 걸 확인할 수 있다.

data.go.kr/iim/api/selectAPIAccountView.do

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

마이페이지

데이터 활용 >

데이터 요청 >

나의 문의 >

회원정보 수정 >

개발계정 상세보기

기본정보

데이터명	국토교통부_아파트매매 실거래 상세 자료	상세설명	
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 활용신청	처리상태	승인
활용기간	2023-04-21 ~ 2025-04-21		

서비스정보

데이터포맷	XML
End Point	http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTradeDev?_wadl&type=xml

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)	인증키
일반 인증키 (Decoding)	

활용신청 상세기능정보

NO	상세기능	설명	일일 트래픽	미리보기
1	국토교통부_아파트매매 상세자료	지역코드와 기간을 이용하여 해당기간, 해당지역의 아파트 매매 신고자료를 제공하는 아파트 매매 신고 정보 조회	1000	확인

4. 위 화면에서 "상세설명"을 클릭한다.

5. 아래 화면이 나타난다.

data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15057511

오픈API 상세

XML 국토교통부_아파트매매 실거래 상세 자료

부동산 거래신고에 관한 법률에 따라 신고된 주택의 실거래 자료를 제공

71 7 관심

OpenAPI 정보 메타데이터 다운로드

데이터 개선요청 오류신고 및 문의

분류체계	일반공공행정 - 일반행정	제공기관	국토교통부
관리부서명	거래신고관리부	관리부서 전화번호	053-663-8642
API 유형	REST	데이터포맷	XML
활용신청	17104	키워드	주택,아파트,실거래가
등록	2017-02-06	수정	2023-08-31
비용부과유무	무료	신청가능 트랙픽	개발계정 : 1,000 / 운영계정 : 활용사례 등록시 신청하면 트랙픽 증가 가능
심의유형	개발단계 : 자동승인 / 운영단계 : 자동승인		

6. 화면을 스크롤하면 "요청변수"가 표시된다.

상세기능

활용사례

추천데이터

상세기능

목록 국토교통부_아파트매매 상세자료 조회

지역코드와 기간을 이용하여 해당기간, 해당지역의 아파트 매매 신고자료를 제공하는 아파트 매매 신고 정보 조회

활용승인 절차 개발단계 : 자동승인 / 운영단계 : 자동승인
신청가능 트래픽 개발계정 : 1,000 / 운영계정 : 활용사례 등록시 신청하면 트래픽 증가 가능
요청주소
서비스URL http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBJSvc/getRTMSDataSvcAptTradeDev

활용신청

요청변수(Request Parameter)

항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
서비스키	ServiceKey	20	필수	-	공공데이터포털에서 받은 인증키
페이지 번호	pageNo	4	옵션	1	페이지번호
한 페이지 결과 수	numOfRows	4	옵션	10	한 페이지 결과 수
지역코드	LAWD_CD	5	필수	11110	지역코드
계약월	DEAL_YMD	6	필수	201512	계약월

7. 화면을 끝까지 스크롤해서 "샘플코드"에서 Python 을 클릭한다. 소스를 복사한다.

샘플코드

Java Javascript C# PHP Curl Objective-C **Python** Nodejs R

Python3 샘플 코드

```
import requests
```

```
url = 'http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBSvc/getRTMSDataSvcAptTradeDev'  
params ={'serviceKey' : '서비스키', 'pageNo' : '1', 'numOfRows' : '10', 'LAWD_CD' : '11110', 'DEAL_YMD' : '201512' }
```

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

```
import requests
```

복사한 소스를 붙여넣기 한다. 그리고 아래처럼 url과 params를 수정한다. servicekey를 직접 url에 직접 넣어서 만든다. 안그러면 오류가 발생한다.

```
url =  
"http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBSvc/getRTMSDataSvcAptT  
radeDev?serviceKey=발급받은 인증키"  
params = {"pageNo" : "1", "numOfRows" : "10", "LAWD_CD" : "11110", "DEAL_YMD" : "201512" }
```

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

16진수로 표현된 유니코드 문자열을 UTF-8으로 디코딩하여 파이썬 문자열로 변환하려면 decode()함수 사용.

```
print(response.content.decode("utf-8"))  
print(type(response.content.decode("utf-8")))
```

실행 결과

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?> <response> <header> <resultCode>00</resultCode> <resultMsg>NORMAL
SERVICE.</resultMsg> </header> <body> <items> <item> <거래금액> 82,500</거래금액> <거래유형>
</거래유형> <건축년도>2008</건축년도> <년>2015</년> <도로명>사직로8길</도로명> <도로명건물본번호코
드>00004</도로명건물본번호코드> <도로명건물부번호코드>00000</도로명건물부번호코드> <도로명시군구코
드>11110</도로명시군구코드> <도로명일련번호코드>03</도로명일련번호코드> <도로명지상지하코드>0</도로
명지상지하코드> <도로명코드>4100135</도로명코드> <등기일자> </등기일자> <법정동>
사직동</법정동> <법정동본번코드>0009</법정동본번코드> <법정동부번코드>0000</법정동부번코드> <법정동
시군구코드>11110</법정동시군구코드> <법정동읍면동코드>11500</법정동읍면동코드> <법정동지번코드>1</
법정동지번코드> <아파트>광화문스페이스본(101동~105동)</아파트> <월>12</월> <일>10</일> <일련번호>1
1110-2203</일련번호> <전용면적>94.51</전용면적> <중개사소재지>
</중개사소재지> <지번>9</지번> <지역코드>11110</지역코드> <층>11</층> <해제사유발생일>
</해제사유발생일> <해제여부> </해제여부> </item> <item> <거래금액> 60,000</거래금액> 이하생략
```


encode(), decode() : 문자열->바이너리->문자열

encode() : 문자열을 -> 바이너리

```
binary_data = "대한민국".encode('utf-8')  
print(binary_data)
```

decode() : 바이너리 -> 문자열로

```
binary_data = b'\xeb\x8c\x80\xed\x95\x9c\xeb\xaf\xbc\xea\xb5\xad'  
decoded_string = binary_data.decode('utf-8')  
print(decoded_string)
```

xmldict 모듈 활용

```
import requests
```

```
url =  
"http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBSvc/getRTMSDataSvcAptTr  
adeDev?serviceKey=발급받은 인증키"  
params = {"pageNo" : "1", "numOfRows" : "10", "LAWD_CD" : "11110", "DEAL_YMD" : "201512"}  
response = requests.get(url, params=params)
```

응답받은 xml 형식의 데이터를 파이썬의 dict로 변경하기 위해 xmldict 설치

pip install xmldict

```
import xmldict
```

```
data=xmldict.parse(response.content)
```

거래금액, 건축년도, 도로명, 아파트, 전용면적 추출하기

```
items=data["response"]["body"]["items"]["item"]
```

```
for i,item in enumerate(items):
```

```
    print(i,":",item["거래금액"],item["건축년도"],item["도로명"],item["아파트"],item["전용면적"])
```

"202311"

※ LAWD_CD 는 행정표준코드관리시스템(www.code.go.kr)의 법정동코드 10자리 중 앞 5자리

실행 결과

- 0 : 82,500 2008 사직로8길 광화문스페이스본(101동~105동) 94.51
- 1 : 60,000 1981 세종대로23길 롯데미도파광화문빌딩 149.95
- 2 : 130,000 2004 경희궁2길 킹스매너 194.43
- 3 : 105,000 2004 사직로8길 경희궁의아침2단지 124.17
- 4 : 120,000 2003 사직로8길 경희궁파크팰리스 146.33
- 5 : 17,000 2014 대학로 이화에수풀 16.98
- 6 : 17,000 2014 대학로 이화에수풀 16.98
- 7 : 57,000 2006 혜화로3가길 렉스빌 106.98
- 8 : 44,000 1995 창경궁로 아남1 84.8
- 9 : 52,000 1995 창경궁로 아남1 84.9

pandas 라이브러리 활용

```
import requests
```

```
url =
```

```
"http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/service/rest/RTMSOBSvc/getRTMSDataSvcAp  
tTradeDev?serviceKey=발급받은 인증키"
```

```
params = {"pageNo" : "1", "numOfRows" : "10", "LAWD_CD" : "11110", "DEAL_YMD" : "201512" }
```

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

```
# 응답받은 xml 형식의 데이터를 파이썬의 dict로 변경하기
```

```
import xmltodict
```

```
data=xmltodict.parse(response.content)
```

```
# 판다스 데이터 프레임으로 처리하고 파일로 저장하기
```

```
# pip install pandas
```

```
import pandas as pd
```

```
df=pd.DataFrame(data["response"]["body"]["items"]["item"])
```

```
print(df[["년", "월", "거래유형", "거래금액", "건축년도", "전용면적", "도로명", "법정동"]])
```

```
# df.to_csv("아파트.csv", encoding="utf-8-sig") #엑셀로 열었을 때 한글 깨짐 해결
```

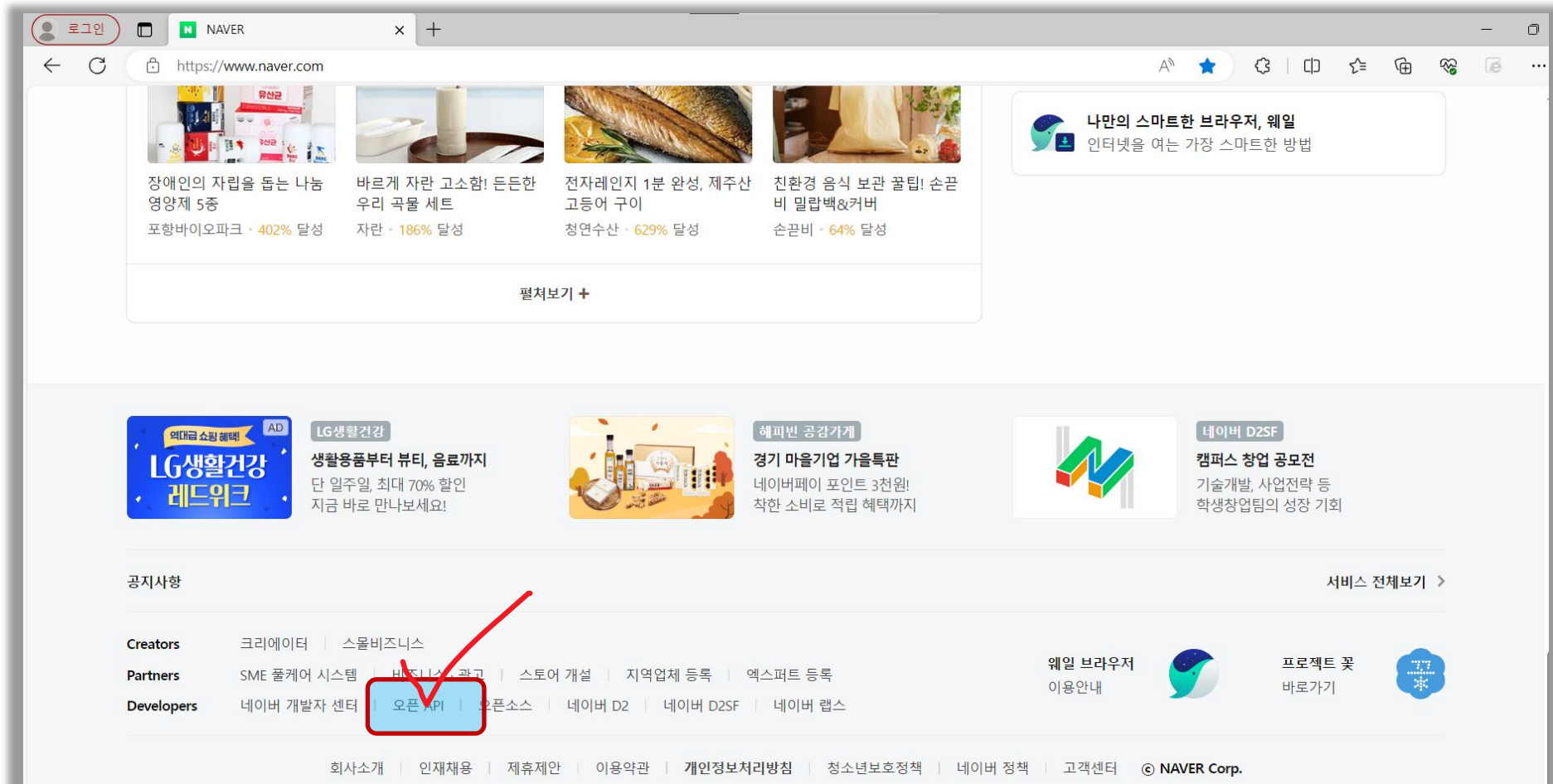
```
df.to_csv("아파트실거래가.csv")
```

실행 결과

년	월	거래유형	거래금액	건축년도	전용면적	도로명	법정동
0	2015	12	None	82,500	2008	94.51	사직로8길 사직동
1	2015	12	None	60,000	1981	149.95	세종대로23길 당주동
2	2015	12	None	130,000	2004	194.43	경희궁2길 내수동
3	2015	12	None	105,000	2004	124.17	사직로8길 내수동
4	2015	12	None	120,000	2003	146.33	사직로8길 내수동
5	2015	12	None	17,000	2014	16.98	대학로 연건동
6	2015	12	None	17,000	2014	16.98	대학로 연건동
7	2015	12	None	57,000	2006	106.98	혜화로3가길 명륜1가
8	2015	12	None	44,000	1995	84.8	창경궁로 명륜2가
9	2015	12	None	52,000	1995	84.9	창경궁로 명륜2가

네이버 오픈API 사용

- 네이버 화면 아래로 스크롤해서 "오픈 API"클릭



Documents

네이버 오픈API를 이용해 편리한 애플리케이션을 개발해보세요.

Documents > [API 공통 가이드](#)

API 공통 가이드

[네이버 오픈API 종류](#)[사전 준비 사항](#)[내 애플리케이션 관리](#)[용어 정리](#)[샘플 코드](#)[오류 코드](#)API 공통 가이드 [↗](#)

네이버 오픈API는 네이버 플랫폼의 기능을 외부 개발자가 쉽게 이용할 수 있게 웹 또는 SDK 형태로 공개한 기술들입니다. 네이버 오픈API로 활용할 수 있는 기술에는 네이버 로그인과 지도, 검색이 있으며, Clova의 음성 인식 기술과 음성 합성 기술, 얼굴 인식 기술, Papago의 기계 번역 기술 등이 있습니다.

API 공통 가이드 개요 [↗](#)

API 공통 가이드는 네이버 오픈API를 사용해 클라이언트 애플리케이션을 개발할 때 미리 알아 두어야 하는 내용을 설명합니다.

- 최종 수정일: 2021년 8월 27일

이 문서의 내용은 언제든지 변경될 수 있습니다.

Application

API 이용을 위해 애플리케이션을 등록하고 API 설정을 할 수 있습니다.

내 애플리케이션

애플리케이션 등록

API 제휴 신청

계정 설정

애플리케이션 등록 (API 이용신청)

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

애플리케이션 이름 ⇄	<input type="text" value="연습용"/> ✓
사용 API ⇄	<div>선택하세요. ▼ ⓘ 검색 네이버 로그인 네이버 인증서 네이버 전자문서 네이버페이 배송지 정보 단축 URL 데이터랩 (검색어트렌드) 데이터랩 (쇼핑인사이트) 카페 캘린더 캡차 (음성) 캡차 (이미지) Papago 번역 Papago 언어감지</div>

- [사용 API] 설정을 확인해 주세요.



내 애플리케이션

애플리케이션 등록

API 제휴 신청

계정 설정

애플리케이션 이름

연습용



- 네이버 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요.
- 40자 이내의 영문, 한글, 숫자, 공백문자, 쉼표(,), "/", "-", "_", 만 입력 가능합니다.

사용 API

선택하세요.



검색



환경 추가



WEB 설정

비로그인 오픈 API
서비스 환경

웹 서비스 URL (최대 10개)

http://naver.com



- 텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.
- http와 https는 구분하지 않습니다.
- www는 빼고 입력해 주세요. 예) http://naver.com
- 서브 도메인이 있으면 대표 도메인명만 입력해 주세요. (예: http://naver.com)
- 하이브리드 앱은 location.href 객체 출력 값을 입력하면 됩니다. (예: file://로컬 URI)

등록하기

취소

Application

API 이용을 위해 애플리케이션을 등록하고 API 설정을 할 수 있습니다.

내 애플리케이션

연습용

애플리케이션 등록

API 재휴 신청

계정 설정

연습용

개요

API 설정

멤버관리

로그인 통계

API 통계

Playground (Beta)

애플리케이션 정보

Client ID

waLhxlCLNvrlSRZxQiyg

Client Secret

.....

보기

API 호출 안내

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

비로그인 오픈 API 당일 사용량

API 호출량 / 일일 허용량



Documents

네이버 오픈 API를 이용해 창의적인 애플리케이션을 제작해 보세요.

Documents > 서비스 API > 검색

블로그

뉴스

책

성인 검색어 판별

백과사전

카페글

지식iN

지역

오타변환

웹문서

이미지

쇼핑

전문자료

검색 > 쇼핑 [↗](#)

- [쇼핑 검색 개요](#)
 - [개요](#)
 - [사전 준비 사항](#)
- [쇼핑 검색 API 레퍼런스](#)
 - [쇼핑 검색 결과 조회](#)
 - [오류 코드](#)
- [검색 API 쇼핑 검색 구현 예제](#)

쇼핑 검색 개요 [↗](#)

- [개요](#)
- [사전 준비 사항](#)

[블로그](#)[뉴스](#)[책](#)[성인 검색어 판별](#)[백과사전](#)[카페글](#)[지식iN](#)[지역](#)[오타변환](#)[웹문서](#)[이미지](#)[쇼핑](#)[전문자료](#)

쇼핑 검색 결과 조회 [↗](#)

설명 [↗](#)

네이버 검색의 쇼핑 검색 결과를 XML 형식 또는 JSON 형식으로 반환합니다.

요청 URL [↗](#)

요청 URL	반환 형식
<code>https://openapi.naver.com/v1/search/shop.xml</code>	XML
<code>https://openapi.naver.com/v1/search/shop.json</code>	JSON



프로토콜 [↗](#)

HTTPS

HTTP 메서드 [↗](#)

GET

블로그

뉴스

책

성인 검색어 판별

백과사전

카페글

지식iN

지역

오타변환

웹문서

이미지

쇼핑

전문자료

참고 사항 [↗](#)

API를 요청할 때 다음 예와 같이 HTTP 요청 헤더에 **클라이언트 아이디**와 **클라이언트 시크릿**을 추가해야 합니다.

```
> GET /v1/search/shop.xml?query=%EC%A3%BC%EC%8B%9D&display=10&start=1&sort=sim HTTP/1.1
> Host: openapi.naver.com
> User-Agent: curl/7.49.1
> Accept: */*
> X-Naver-Client-Id: {애플리케이션 등록 시 발급받은 클라이언트 아이디 값}
> X-Naver-Client-Secret: {애플리케이션 등록 시 발급받은 클라이언트 시크릿 값}
```

요청 예 [↗](#)

```
curl "https://openapi.naver.com/v1/search/shop.xml?query=%EC%A3%BC%EC%8B%9D&display=10&start=1&sort=sim" \
-H "X-Naver-Client-Id: {애플리케이션 등록 시 발급받은 클라이언트 아이디 값}" \
-H "X-Naver-Client-Secret: {애플리케이션 등록 시 발급받은 클라이언트 시크릿 값}" -v
```

응답 [↗](#)

응답에 성공하면 결과값을 XML 형식 또는 JSON 형식으로 반환합니다. XML 형식의 결과값은 다음과 같습니다.

[블로그](#)[뉴스](#)[책](#)[성인 검색어 판별](#)[백과사전](#)[카페글](#)[지식iN](#)[지역](#)[오타변환](#)[웹문서](#)[이미지](#)[쇼핑](#)[전문자료](#)

검색 API 블로그 검색 구현 예제 [↗](#)

다음은 검색 API로 블로그 검색 결과를 조회하는 구현 예제입니다. 검색 API의 다른 작업을 구현하는 방법도 이 구현 예제와 유사하기 때문에 이 구현 예제를 참고하면 검색 API를 구현할 수 있습니다.

- [Java](#)
- [PHP](#)
- [Node.js](#)
- ♥ • [Python](#)
- [C#](#)

참고

- 샘플 코드에서 `YOUR_CLIENT_ID` 또는 `YOUR-CLIENT-ID`에는 애플리케이션을 등록하고 발급받은 클라이언트 아이디 값을 입력합니다.
- 샘플 코드에서 `YOUR_CLIENT_SECRET` 또는 `YOUR-CLIENT-SECRET`에는 애플리케이션을 등록하고 발급받은 클라이언트 시크릿 값을 입력합니다.

[Java](#) [↗](#)

[블로그](#)[뉴스](#)[책](#)[성인 검색어 판별](#)[백과사전](#)[카페글](#)[지식iN](#)[지역](#)[오타변환](#)[웹문서](#)[이미지](#)[쇼핑](#)[전문자료](#)

Python

```
# 네이버 검색 API 예제 - 블로그 검색
import os
import sys
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)
```


쇼핑 검색 API (운동화 검색)

```
import requests
import pprint
```

```
client_id = "발급받은 id"
client_secret = "발급받은 secret"
search_text = "운동화"
```

```
url = "https://openapi.naver.com/v1/search/shop.json?query=" + search_text
header={"X-Naver-Client-Id":client_id, "X-Naver-Client-Secret":client_secret}
response = requests.get(url,headers=header)
data=response.json() # str --> dict
pprint.pprint(data)
```

```
for row in data['items']:
    print(row['brand'], row['lprice'])
```

```
뉴발란스 66123
나이키 77470
나이키 38900
나이키 67500
나이키 57900
나이키 127000
아디다스 29000
뉴발란스 109510
뉴발란스 126520
나이키 69000
```

쇼핑 검색 API (운동화 검색) 결과를 csv파일로 저장

```
import requests
import pandas as pd
```

```
client_id = "발급받은 id"
client_secret = "발급받은 secret"
search_text = "운동화"
```

```
url = "https://openapi.naver.com/v1/search/shop.json?query=" + search_text
header={"X-Naver-Client-Id":client_id, "X-Naver-Client-Secret":client_secret}
response = requests.get(url,headers=header)
data=response.json() # str --> dict
```

```
df=pd.DataFrame(data['items'])
df=df[['brand','title','lprice']]
print(df)
df.to_csv('쇼핑검색.csv')
```

```
,brand,title,lprice
0,뉴발란스,뉴발란스 530 MR530S6,66123
1,나이키,나이키 써밋화이트 TC7900 DD9682 100,77470
2,나이키,나이키 레볼루션 5 6 7 런닝화 남성 여성 커플 <b>운동화</b> 조깅화 발편한 헬스화 러닝화,38900
3,나이키,(국내매장판) 나이키 에어맥스 90 SC 에센셜 LTR 테라스케이프,67500
4,나이키,나이키 데이브레이크 CK2351 101,57900
5,나이키,나이키 에어맥스 97 921826-101,127000
6,뉴발란스,뉴발란스 530 MR530SH,109510
7,뉴발란스,뉴발란스 530 MR530KA,126520
8,나이키,나이키 에어맥스 AP CU4826-100,69000
9,아디다스,아디다스 런닝화 <b>운동화</b> 발볼넓은 남성 여성 발편한 헬스화 조깅화,29000
```