

API 활용 데이터 수집

API 종합 비교

	API	인증 방식	비용	데이터 유형	가이드 URL
1	공공데이터포털	파라미터: serviceKey	무료	기상,교통,부동산 등	data.go.kr
2	네이버 검색 API	헤더: Client ID/Secret	무료	뉴스,블로그,카페 등	developers.naver.com
3	네이버 데이터랩	헤더: Client ID/Secret	무료	검색어 트렌드	developers.naver.com
4	GDELT	인증 불필요	무료	글로벌 뉴스 메타데이터	gdeltproject.org
5	YouTube Data API	파라미터: key	무료	동영상,채널,댓글	developers.google.com
6	NewsAPI.org	파라미터: apiKey	무료(제한)	글로벌 뉴스 기사	newsapi.org
7	Reddit API	헤더: Bearer Token	무료	게시물,댓글	reddit.com/dev/api
8	X (Twitter) API	헤더: Bearer Token	\$200+/월	트윗,프로필	developer.x.com

공통 주의사항 — API 인증의 이해

대부분의 API는 브라우저 주소창에 URL만 입력하면 인증 에러가 발생합니다.
반드시 HTTP 헤더 또는 쿼리 파라미터로 인증 정보를 포함해야 합니다.

.env 파일로 API 키 관리

```
# .env (프로젝트 루트, .gitignore 추가)
NAVER_CLIENT_ID=발급받은_ID
NAVER_CLIENT_SECRET=발급받은_SECRET
GOOGLE_API_KEY=발급받은_KEY
YOUTUBE_API_KEY=발급받은_KEY
NEWSAPI_KEY=발급받은_KEY
DATA_GO_KR_KEY=발급받은 인증키
```

인증 방식별 코드 패턴

인증 방식	코드 패턴
헤더: ID/Secret	headers={"X-Naver-Client-Id": ...}
헤더: Bearer Token	headers={"Authorization": "Bearer ..."} ※ Bearer Token은 ID/Secret과 달리, 클라이언트 측에 저장되어 재사용되는 형태이다.
파라미터: key	params={"key": api_key}
파라미터: serviceKey	URL에 직접 삽입
인증 불필요	바로 호출 가능

1. 한국 공공데이터 API

1 공공데이터포털 (data.go.kr)

정부·공공기관 데이터 (기상, 교통, 부동산, 인구 등)

인증: serviceKey 파라미터 (URL에 직접 삽입)

비용: 무료 (개발 1,000건/일, 운영 100,000건/일)

가이드: data.go.kr/ugs/selectPublicDataUseGuideView.do

2 서울 열린데이터광장

서울시 공공데이터 (교통, 환경, 문화, 복지 등)

인증: URL 경로에 인증키 포함

비용: 무료

가이드: data.seoul.go.kr/together/guide/useGuide.do

3 인증키 주의사항

공공데이터포털 인증키에 =, +, / 등 포함

→ 이미 URL 인코딩된 상태로 발급됨

→ params 사용 시 이중 인코딩 에러 발생

→ URL 문자열에 직접 삽입해야 함

4 한국 Public API 모음 (GitHub)

한국 서비스용 Public API 큐레이션

영화, 도서, 농업, 환경, 교통 등

인증: API마다 상이

github.com/yybmion/public-apis-4Kr

1. 공공데이터포털 (data.go.kr) — 인증키 발급

1 회원가입

data.go.kr 접속

오른쪽 상단 "회원가입" 클릭

일반회원으로 가입

(네이버/카카오 소셜 로그인도 가능)

2 원하는 API 찾기

로그인 후 상단 검색창에서 검색

(예: "교통사고", "기상청", "부동산")

검색 결과에서 "오픈API" 탭 클릭

"파일데이터"가 아닌 "오픈API" 선택!

3 활용신청

API 상세 페이지에서 "활용신청" 클릭

활용목적: "학술연구" 또는 "교육용"

대부분 "자동승인" → 즉시 사용 가능

(일부 수동 심의: 1~3일 소요)

4 인증키 확인

마이페이지 → 오픈API → 개발계정

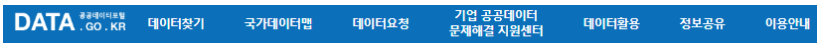
"일반 인증키 (Encoding)" 복사

(%2B, %3D 등 포함된 긴 문자열)

이 키를 .env 파일에 저장

1. 한국 공공데이터 API — 코드 예제

교통사고 데이터: <https://www.data.go.kr/data/15057467/openapi.do>



오픈API 상세



XML JSON 한국도로교통공단_지자체별 교통사고 다발지역

한국 각 지자체별 교통사고 다발지역 정보(지자체별 top3 지점)

- 대상사고 : 1년간 발생한 전체 교통사고
- 다발지역 선정조건 : 반경 150m 내, 대상사고 3건 이상인 지점 중 상위 3개소(TOP3)
- * 지자체별로 해당 조건에 해당하는 지점이 3개가 되지 않을 경우 1개 또는 2개 지점만 선정될 수 있음

- WMS 형태의 API 정보는 교통사고정보개발시스템(<https://opendata.koroad.or.kr/api/sample.do>)을 통해 확인하실 수 있습니다.

* 해당자료는 교통사고분석시스템(<https://taas.koroad.or.kr>) 데이터를 기반으로 함

4 0 다관심

OpenAPI 정보 데이터데이터 다운로드 오픈API 예제요청

데이터 개선요청 오픈API 및 문의

분류체계	공공질서및안전 - 안전관리	제공기관	한국도로교통공단
관리부서명	데이터융합처	관리부서 전화번호	033-749-5266
API 유형	REST	데이터포맷	JSON+XML
활용신청	1593	키워드	교통사고, 사고다발지역, 한국, 지자체별, 상위3개소
등록일	2019-01-11	수정일	2026-03-30
비용부과유무	무로	신청가능 트래픽	개발예정 : 10,000 / 운영예정 : 활용사례 등록시 신청하면 트래픽 증가 가능
업데이트 주기	실시간	실의유형	개발단계 : 자동승인 / 운영단계 : 자동승인
공간범위		시간범위	
이용허락범위	이윤희님님의 저작물		
참고문서	기술문서: 한국도로교통공단_지자체별_사고다발지역_정보.html		

```
import os
from urllib.parse import unquote
import requests
from dotenv import load_dotenv
```

```
load_dotenv()
service_key = unquote(os.getenv("DATA_GO_KR_KEY", ""))
```

```
# 설정: 검색연도, 시도코드, 구군코드, 응답형식(json/xml),
# 표시개수, 페이지
# siDo/guGun은 법정동 코드(다운로드:
https://www.data.go.kr/data/15063424/fileData.do)
params = {
```

```
    "serviceKey": service_key,
    "searchYearCd": "2017",
    "siDo": "11",
    "guGun": "200",
    "type": "json",
    "numOfRows": 10,
    "pageNo": 1,
```

```
}
```

```
url =
"http://apis.data.go.kr/B552061/frequentzoneLg/getRestFr
equentzoneLg"
response = requests.get(url, params=params, timeout=10)
print(response.text)
```

```
# 공공데이터 API는 serviceKey 포함해 아래 URL을 브라우저
주소창에 붙여넣어 작동 가능
prepared_url = requests.Request("GET", url,
params=params).prepare().url
print(f"\nRequest URL: {prepared_url}")
```

2&3. 네이버 오픈API — 인증키 발급 (검색 + 데이터랩)

1 네이버 개발자센터 접속

developers.naver.com 접속
네이버 계정으로 로그인
(네이버 계정이 곧 개발자 계정)

2 애플리케이션 등록

상단 메뉴 "Application" → "등록" 클릭
애플리케이션 이름: 아무 이름 (예: "수업용")
사용 API 선택: "검색" 체크 + "데이터랩(검색어트렌드)" 체크

3 환경 설정

비로그인 오픈 API 서비스 환경에서
"WEB 설정" 선택
웹 서비스 URL에 http://localhost 입력
"등록하기" 클릭

4 Client ID / Secret 확인

등록 완료 후 "내 애플리케이션" 페이지
Client ID와 Client Secret 표시됨
이 두 값을 복사하여 .env에 저장
(외부 노출 절대 금지!)

2. 네이버 검색 API

개요

용도: 뉴스, 블로그, 카페글, 지식iN, 책, 쇼핑, 웹문서, 이미지, 지역 등 분야별 수집

인증: HTTP 헤더에 X-Naver-Client-Id + X-Naver-Client-Secret 필수

비용: 무료 (최대 일 25,000건) | 키 발급: developers.naver.com/apps/#/register

가이드: developers.naver.com/docs/serviceapi/search/news/news.md

주요 엔드포인트

/v1/search/news - 뉴스

/v1/search/blog - 블로그

/v1/search/cafearticle - 카페글

/v1/search/kin - 지식iN

/v1/search/book - 책

/v1/search/shop - 쇼핑

/v1/search/local - 지역

Python 예제 — 뉴스 검색 (인증 헤더 포함)

```
import os
import requests
from dotenv import load_dotenv
load_dotenv()

client_id = os.getenv("NAVER_CLIENT_ID")
client_secret = os.getenv("NAVER_CLIENT_SECRET")

# 검색어, 표시 개수, 시작 위치, 정렬(sim: 정확도순, date: 날짜순)
params = {"query": "주식", "display": 10, "start": 1, "sort": "date"}
headers = {"X-Naver-Client-Id": client_id, "X-Naver-Client-Secret": client_secret}

url = "https://openapi.naver.com/v1/search/news.json"
response = requests.get(url, params=params, headers=headers, timeout=10)
print(response.text)
```

네이버 API는 헤더(X-Naver-Client-Id/Secret)로 인증하므로 브라우저 주소창에 URL을 붙여넣는 방식으로는 작동되지 않음.

2. 네이버 검색 API — 대규모 수집

페이지네이션 + Rate Limit

```
all_items, start = [], 1
while start <= MAX_ITEMS:
    resp = requests.get(url, params={"query": QUERY, "display":
DISPLAY, "start": start, "sort": "date"},
                        headers=headers, timeout=10)
    if resp.status_code == 429:
        time.sleep(5)
        continue
    if resp.status_code != 200:
        print(f"Error {resp.status_code}: {resp.text}")
        break

    data = resp.json()
    items = data.get("items", [])
    if not items:
        break

    all_items.extend(items)
    print(f"[{start}~{start + len(items) - 1}] 누적
{len(all_items)}건 / 전체 {data.get('total', 0)}건")

    start += DISPLAY
    if start > min(data.get("total", 0), MAX_ITEMS):
        break
    time.sleep(RATE_LIMIT_SEC)

with open("news_result.json", "w", encoding="utf-8") as f:
    json.dump(all_items, f, ensure_ascii=False, indent=2)
    print(f"\n수집 완료: {len(all_items)}건 → news_result.json")
```

```
all_items, start = [], 1
while start <= MAX_ITEMS:
    resp = requests.get(url, params={"query": QUERY, "display":
DISPLAY, "start": start, "sort": "date"},
                        headers=headers, timeout=10)
    if resp.status_code == 429:
        time.sleep(5)
        continue
    if resp.status_code != 200:
        print(f"Error {resp.status_code}: {resp.text}")
        break

    data = resp.json()
    items = data.get("items", [])
    if not items:
        break

    all_items.extend(items)
    print(f"[{start}~{start + len(items) - 1}] 누적
{len(all_items)}건 / 전체 {data.get('total', 0)}건")

    start += DISPLAY
    if start > min(data.get("total", 0), MAX_ITEMS):
        break
    time.sleep(RATE_LIMIT_SEC)

with open("news_result.json", "w", encoding="utf-8") as f:
    json.dump(all_items, f, ensure_ascii=False, indent=2)
    print(f"\n수집 완료: {len(all_items)}건 → news_result.json")
```

3. 네이버 데이터랩 API

개요

용도: 네이버 검색어 트렌드, 쇼핑인사이드

인증: 헤더 X-Naver-Client-Id/Secret 필수

비용: 무료 (최대 일 1,000건)

가이드:

developers.naver.com/docs/serviceapi/datalab/search/search.md

주요 엔드포인트 (모두 POST)

/v1/datalab/search - 검색어 트렌드

/v1/datalab/shopping/categories - 쇼핑 분야별

/v1/datalab/shopping/category/gender - 성별

/v1/datalab/shopping/category/age - 연령별

/v1/datalab/shopping/category/device - 기기별

*트렌드 값으로 검색 비율 (ratio) 사용: 기간 중
최댓값을 100으로 해서 상대적 비율로 환산

Python 예제 — 검색어 트렌드 (POST + JSON Body + 인증 헤더)

```
import json
import os
import requests
from dotenv import load_dotenv
load_dotenv()

headers = {
    "X-Naver-Client-Id": os.getenv("NAVER_CLIENT_ID"),
    "X-Naver-Client-Secret": os.getenv("NAVER_CLIENT_SECRET"),
    "Content-Type": "application/json",
}

body = {
    "startDate": "2025-01-01",
    "endDate": "2026-03-31",
    "timeUnit": "month", # date: 일별, week: 주별, month: 월별
    "keywordGroups": [{"groupName": "인공지능", "keywords": ["인공지능",
"AI"]}], # 인공지능과 "AI"를 하나로 묶어서 분석
}

url = "https://openapi.naver.com/v1/datalab/search"
resp = requests.post(url, headers=headers, data=json.dumps(body),
timeout=10)
print(resp.text)
```

4. Google 검색 API

1 Custom Search JSON API

인증: params에 key + cx 필수

비용: 100건/일 무료, 이후 \$5/1000건

주의: 2027.1 기존 고객만, 신규는 Vertex AI

가이드: developers.google.com/custom-search/v1/overview

```
url = "https://www.googleapis.com/
      customsearch/v1"
params = {
    "key": os.getenv("GOOGLE_API_KEY"),
    "cx": os.getenv("GOOGLE_SEARCH_ENGINE_ID"),
    "q": "데이터 저널리즘",
    "num": 10
}
response = requests.get(url, params=params)
```

2 Knowledge Graph Search API

인증: params에 key 필수

비용: 무료

용도: 인물, 장소, 사물 엔티티 검색

가이드: developers.google.com/knowledge-graph

```
url = "https://kgsearch.googleapis.com/
      v1/entities:search"
params = {
    "query": "경희대학교",
    "key": os.getenv("GOOGLE_API_KEY"),
    "limit": 5,
    "languages": "ko"
}
response = requests.get(url, params=params)
```

5. Google Trends API

1 공식 API (Alpha, 2025.7 출시)

인증: Google Cloud API Key (Alpha 신청 필요)

비용: Alpha 기간 무료

특징: 일관된 스케일링, 5개+ 검색어 비교

Alpha 신청: developers.google.com/search/blog/2025/07/trends-api

주의: 제한된 테스터만 접근 가능

2 pytrends (비공식 라이브러리)

인증 불필요 (웹 스크래핑 기반)

비용: 무료 | 주의: 차단 가능성 있음

```
from pytrends.request import TrendReq
pytrends = TrendReq hl="ko", tz=540
pytrends.build_payload(
    kw_list=["인공지능", "ChatGPT"],
    timeframe="2025-01-01 2026-03-31",
    geo="KR")
df = pytrends.interest_over_time()
```

6. YouTube Data API v3

용도: 동영상 검색, 채널/동영상 메타데이터, 댓글, 재생목록 수집 | 인증: params에 key 필수

비용: 무료 (일일 10,000 units) | 키 발급: console.cloud.google.com → YouTube Data API v3 활성화

가이드: developers.google.com/youtube/v3 | 쿼터 계산기: developers.google.com/youtube/v3/determine_quota_cost

동영상 검색

```
url = "https://www.googleapis.com/
      youtube/v3/search"
params = {
    "key": os.getenv("YOUTUBE_API_KEY"),
    "part": "snippet",
    "q": "데이터 저널리즘",
    "type": "video",
    "maxResults": 10,
    "order": "viewCount",
    "regionCode": "KR"
}
resp = requests.get(url, params=params)
```

동영상 통계 + 댓글 수집

```
# 통계 (조회수, 좋아요)
url = "https://www.googleapis.com/
      youtube/v3/videos"
params = {"key": API_KEY,
          "part": "snippet,statistics",
          "id": "VIDEO_ID"}

# 댓글 수집
url = "https://www.googleapis.com/
      youtube/v3/commentThreads"
params = {"key": API_KEY,
          "part": "snippet",
          "videoId": "VIDEO_ID",
          "maxResults": 20}
```

7. X (Twitter) API v2 **무료!**

개요

인증: 헤더 Authorization: Bearer TOKEN 필수

학술 연구용 무료 접근 2023년 폐지

키 발급: developer.x.com/en/portal/dashboard

가이드: developer.x.com/en/docs/x-api

요금제: docs.x.com/x-api/getting-started/pricing

요금제 (2026년 현재)

티어	가격	읽기 한도
Free	\$0	읽기 불가
Basic	\$200/월	10,000건/월
Pro	\$5,000/월	1,000,000건/월
Pay-Per-Use	종량제	크레딧 기반

Python 예제 (Basic 이상 필요)

```
url = "https://api.x.com/2/tweets/search/recent"
params = {"query": "데이터저널리즘 lang:ko", "max_results": 10,
          "tweet.fields": "created_at,public_metrics,author_id"}
headers = {"Authorization": f"Bearer {os.getenv('X_BEARER_TOKEN')}}" } # <-- 헤더 필수
response = requests.get(url, headers=headers, params=params)
if response.status_code == 403:
    print("Free 티어에서는 읽기 불가. Basic($200/월) 이상 필요")
```

8. Reddit Data API

인증: OAuth 2.0 — Bearer Token 헤더 필수 (Client ID/Secret으로 토큰 발급) | 비용: 비상업적 무료 | Rate Limit: 60req/min

키 발급: reddit.com/prefs/apps → script 타입 선택 | 가이드: support.reddithelp.com/hc/en-us/articles/16160319875092

학술 연구: reddit.com/r/reddit4researchers | Python 래퍼 (PRAW): praw.readthedocs.io

PRAW 라이브러리 (권장)

```
import praw, os
from dotenv import load_dotenv
load_dotenv()

reddit = praw.Reddit(
    client_id=os.getenv("REDDIT_CLIENT_ID"),
    client_secret=os.getenv(
        "REDDIT_CLIENT_SECRET"),
    user_agent="DataJournalism/1.0"
)

subreddit = reddit.subreddit("korea")
for post in subreddit.hot(limit=10):
    print(post.title, post.score)
```

직접 OAuth 인증 (래퍼 없이)

```
# 1단계: Access Token 발급
auth = base64.b64encode(
    f"{client_id}:{client_secret}"
    .encode()).decode()
resp = requests.post(
    "https://www.reddit.com/api/v1/
    access_token",
    headers={"Authorization":f"Basic {auth}",
            "User-Agent":"App/1.0"},
    data={"grant_type":"client_credentials"})
token = resp.json()["access_token"]

# 2단계: API 호출
headers = {"Authorization":f"Bearer {token}",
          "User-Agent":"App/1.0"}
```

9. GDELT — 글로벌 뉴스 데이터 (무료, 인증 불필요)

용도: 전 세계 뉴스 메타데이터, 이벤트 데이터, 감성 분석, 지리 정보 (65개 언어 실시간 기계번역)

인증: 불필요 — API Key 없이 바로 호출 가능 | 비용: 완전 무료

가이드: blog.gdeltdoc.org/gdelt-doc-2-0-api-debuts | Python: github.com/alex9smith/gdelt-doc-api

직접 호출 (인증 불필요)

```
import requests, pandas as pd

# 인증 없이 바로 호출 가능
url = "https://api.gdeltdoc.org/
      api/v2/doc/doc"
params = {
    "query": "South Korea",
    "mode": "artlist",
    "maxrecords": 50,
    "format": "json",
    "sort": "datedesc",
    "timespan": "1w"
}
resp = requests.get(url, params=params)
df = pd.DataFrame(resp.json()["articles"])
```

gdeltdoc 라이브러리

```
from gdeltdoc import Gdeltdoc, Filters

f = Filters(
    keyword="Korea",
    start_date="2026-03-01",
    end_date="2026-03-31",
    country="KS" # FIPS: South Korea
)
gd = Gdeltdoc()
articles = gd.article_search(f)

# 보도량 타임라인
timeline = gd.timeline_search(
    "timelinevol", f)
# 감성 타임라인
tone = gd.timeline_search("timelinetone", f)
```

9. GDELT — 고급 쿼리 & API 종류

주요 API 종류

API	설명
DOC 2.0	전문 검색 (최근 3개월)
GEO 2.0	뉴스 지리 데이터
Context 2.0	문장 단위 검색
TV 2.0	미국 TV 뉴스 자막

고급 쿼리 연산자

연산자	예시	설명
sourcecountry:	KS	발행 국가
sourcelang:	Korean	언어 필터
domainis:	nytimes.com	특정 도메인
near	near10:"climate"	근접 검색
tone< / tone>	tone<-5	감성 필터

10. 뉴스 API — NewsAPI, Mediastack, NewsAPI.ai

1 NewsAPI.org

50,000+ 소스, 50개국

인증: apiKey 파라미터

무료: 100req/일, 30일 기사

newsapi.org/docs

```
url = "https://newsapi.org/  
v2/everything"  
params = {  
    "apiKey": os.getenv(  
        "NEWSAPI_KEY"),  
    "q": "AI",  
    "language": "en",  
    "sortBy": "publishedAt",  
    "pageSize": 10  
}
```

2 Mediastack

7,500+ 소스, 50개국, 13개 언어

인증: access_key 파라미터

무료: 월 100건

mediastack.com/documentation

```
# 무료: http만 (https X)  
url = "http://api.mediastack  
.com/v1/news"  
params = {  
    "access_key": os.getenv(  
        "MEDIASTACK_KEY"),  
    "keywords": "AI",  
    "languages": "en",  
    "limit": 10  
}
```

3 NewsAPI.ai

150,000+ 소스, 50개 언어

2014년부터 아카이브

NLP 감성 분석 내장

newsapi.ai/documentation

```
from eventregistry import (  
    EventRegistry,  
    QueryArticlesIter)  
  
er = EventRegistry(  
    apiKey=os.getenv(  
        "EVENTREGISTRY_KEY"))  
q = QueryArticlesIter(  
    keywords="AI",  
    lang="eng")
```

11. 기타 유용한 API

1 Clova Face Recognition

얼굴 인식, 유명인 닥은꼴

인증: 헤더 ID/Secret 필수

[developers.naver.com/docs/
serviceapi/vision/face/face.md](https://developers.naver.com/docs/serviceapi/vision/face/face.md)

```
url = "https://openapi.naver.com/  
v1/vision/celebrity"  
headers = {  
    "X-Naver-Client-Id":  
        os.getenv("NAVER_CLIENT_ID"),  
    "X-Naver-Client-Secret":  
        os.getenv("NAVER_CLIENT_SECRET")  
}  
files = {"image":  
    open("photo.jpg", "rb")}  
resp = requests.post(  
    url, headers=headers,  
    files=files)
```

2 Google Cloud NL API

텍스트 감성 분석, 엔티티 추출

구문 분석, 콘텐츠 분류

인증: API Key 또는 서비스 계정

cloud.google.com/natural-language/docs

3 Google Cloud Vision API

이미지 라벨 감지, OCR

얼굴 감지, 랜드마크 감지

인증: API Key 또는 서비스 계정

cloud.google.com/vision/docs

Vibe Coding 활용법

1 자연어로 데이터 수집

"네이버 뉴스 검색 API로 '인공지능' 관련 기사 1000건 수집해서 pandas DataFrame으로 변환하고, 월별 기사 수 추이 그래프 그려줘.
API 키는 .env 파일에서 읽어와.
헤더에 X-Naver-Client-Id와 X-Naver-Client-Secret 넣어야 해."

2 자연어로 분석 + 시각화

"GDELT DOC API로 최근 3개월간 'Korea' 관련 글로벌 뉴스 수집하고, 국가별 보도량 비교 차트 만들어줘.
GDELT는 인증 불필요."

3 자연어로 대규모 수집

"YouTube Data API로 '데이터저널리즘' 검색 결과 동영상 500개의 제목, 조회수, 좋아요, 게시일 수집해서 CSV로 저장해줘.
API 키는 .env의 YOUTUBE_API_KEY에서 가져와. params에 key 파라미터로 넣어야 해."

4 핵심 포인트

프롬프트에 인증 방식을 명시하면 정확한 코드 생성

VSCode 터미널에서 바로 실행 가능

코드 작성보다 데이터 설계와 해석에 집중