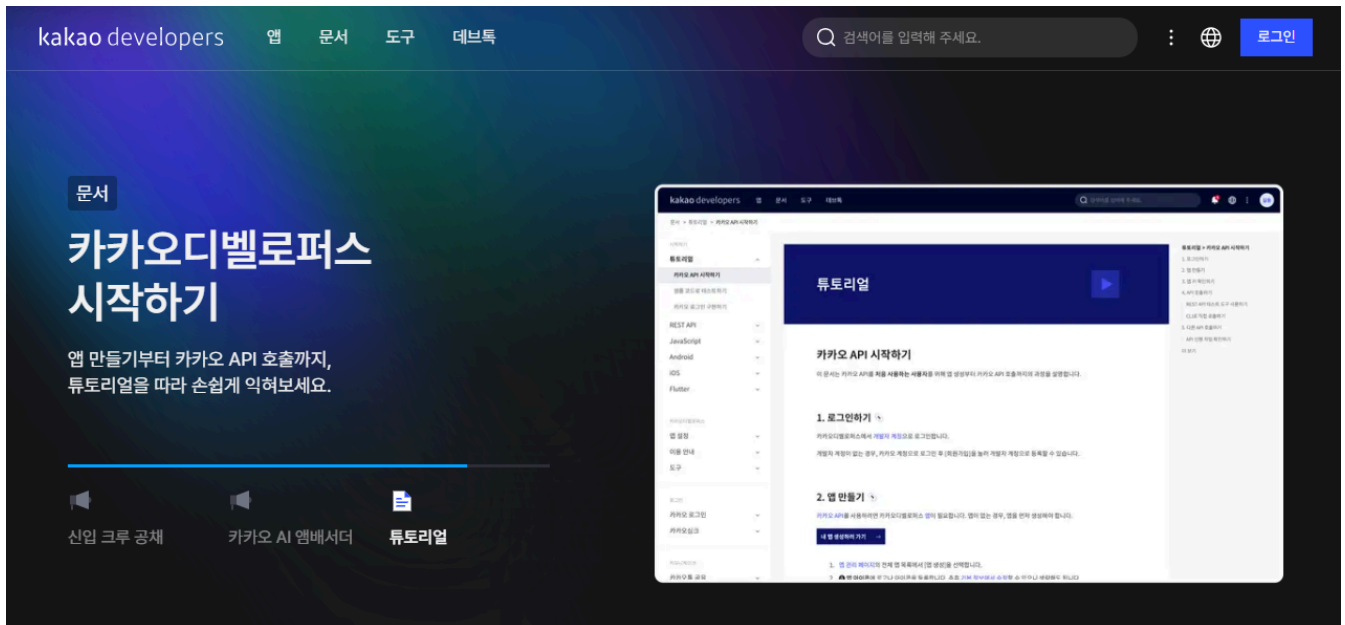


## ✓ 2. 카카오 오픈 API

### ✓ 2.1 카카오 오픈 API 사용 준비

- 카카오 개발자 사이트 가입
- OPEN API 사용을 위한 Auth Key 발급

<https://developers.kakao.com/>



카카오 계정이 있는 경우 로그인하자. 계정이 없는 경우 회원 가입 진행

카카오 계정이 있는 경우 로그인, 없는 경우 회원 가입 진행

서비스 이용 약관 동의 및 개발자 이름 입력후 회원 가입

Auth Key 생성을 위한 Application 생성

output폴더 생성한후 파일쓰기

```
#1
data = 'hello'
with open('./output/hello.txt', 'w') as fp :
    fp.write(data)
```

파일읽기

```
#2
data = 'hello'
with open('./output/hello.txt', 'r') as fp :
    print(fp.read())
```

hello

```
#3
data = 'hello'
with open('./output/hello.txt', 'r') as fp :
    print(fp.read())
```

```
import requests
```

```
def get_image(url, FileName) :
    img_response = requests.get(url)

    if img_response.status_code == 200 :
        #print(img_response.content)
        print("=" * 30 + "[이미지 저장]" + "=" * 30)
        file = "./output/" + FileName
        with open(file, "wb") as fp :
            fp.write(img_response.content)

url = "https://search1.kakaocdn.net/argon/600x0_65_wr/1mZk3b2X1w8"
get_image(url, "팬수1.jpg")
url = "https://img.hankyung.com/photo/201912/01.21302999.1.jpg"
get_image(url, "팬수2.jpg")
```

```
hello
=====
=====
```

rest key 미입력시 오류 발생

```
#4
import requests
import json
#이 주소는 카카오 이미지 검색 API의 엔드포인트(endpoint)입니다.
url = 'https://dapi.kakao.com/v2/search/image'
headers = {
    "Authorization" : "KakaoAK 7f49b07aefae8d8888ce512839ac1f0"
}
data = {
    "query" : "팬수",
    "size" : 20 # 검색 결과 사이즈를 20으로 제한
}
response = requests.post(url, headers=headers, data=data)
if response.status_code != 200 :
    print("error !!! because ", response.json())
else :
    count = 0
    for image_info in response.json()['documents'] :
        if count >= 20:
            break
        print(f"[{count}th] image_url =", image_info['image_url'])
        count = count + 1
```

[숨겨진 출력 표시](#)

```
#5
import requests
import json

def save_image(url, FileName) :
    img_response = requests.get(url)

    if img_response.status_code == 200 :
        #print(img_response.content)
        print("=" * 30 + "[이미지 저장]" + "=" * 30)
        file = "./output/" + FileName
        with open(file, "wb") as fp :
            fp.write(img_response.content)

url = 'https://dapi.kakao.com/v2/search/image'
headers = {
    "Authorization" : "KakaoAK 7f49b07aefae8d8888ce512839ac1f0"
}
data = {
    "query" : "팬수",
    "size" : 20 # 검색 결과 사이즈를 20으로 제한
}
```

```

response = requests.post(url, headers=headers, data=data)
if response.status_code != 200 :
    print("error !!! because ", response.json())
else :
    count = 0
    for image_info in response.json()['documents'] :
        if count >= 20: # 저장하는 이미지 개수를 20개로 제한
            break
        print(f"[{count}th] image_url =", image_info['image_url'])
        count = count + 1
        file_name = "test_%d.jpg" %(count)
        save_image(image_info['image_url'], file_name)

```

[숨겨진 출력 표시](#)

## ▽ IPython.display

- IPython.display: Interactive Python 모듈은 IPython 환경 (Colab과 같은 노트북 환경에서 주로 사용됩니다)에서 다양한 형식의 데이터를 풍부하게 표시하기 위한 도구들을 제공합니다. 일반적인 Python의 print() 함수보다 더 다양한 형태로 데이터를 보여줄 수 있게 해줍니다.
- Image: 이 클래스는 이미지 데이터를 다루기 위해 사용됩니다. 파일 경로를 지정하거나 이미지 데이터를 직접 전달하여 Image 객체를 생성할 수 있습니다.

```

#6
from IPython.display import Image, display

image_path = "/content/output/test_7.jpg"
display(Image(filename=image_path))

```

[숨겨진 출력 표시](#)

output폴더 이미지 모두 출력하기

```

import os
from IPython.display import Image, display

image_folder = "/content/output/"
image_files = [f for f in os.listdir(image_folder) if f.endswith(('.jpg', '.png', '.jpeg'))]

for image_file in image_files:
    image_path = os.path.join(image_folder, image_file)
    print(f"Displaying: {image_file}")
    display(Image(filename=image_path))

```

[숨겨진 출력 표시](#)

```

import os
from IPython.display import Image, display

# 이미지가 들어 있는 폴더 경로
folder = 'output'

# 폴더 내의 이미지 파일들을 하나씩 읽어들이어 작은 크기로 출력
for filename in sorted(os.listdir(folder)):
    if filename.lower().endswith(('.jpg', '.jpeg', '.png')):
        img_path = os.path.join(folder, filename)
        display(Image(filename=img_path, width=150))

```

[숨겨진 출력 표시](#)

## ▽ output폴더의 파일 삭제하기

```

import os
#output 폴더 삭제하기
folder = 'output'

# output 폴더에 있는 파일 목록을 가져옵니다.

```

```
file_list = os.listdir(folder)

print(f"Deleting files in {folder}...")

for filename in file_list:
    file_path = os.path.join(folder, filename)
    # 파일인지 확인 (하위 폴더 제외)
    if os.path.isfile(file_path):
        try:
            os.remove(file_path)
            print(f"Deleted: {filename}")
        except Exception as e:
            print(f"Error deleting {filename}: {e}")

print("Deletion complete.")
```

[숨겨진 출력 표시](#)