



5주차 Python 심화 (2)



카카오 메시지 API

목차

1. 인가 코드, 토큰 발급
2. 나에게 카톡 보내기

Python 심화

카카오 메시지 API



Python 심화

카카오 메시지 API



인가 코드는 [토큰 받기](#)에 필요한 값으로, 토큰에 부여될 권한 정보를 포함하고 있습니다!

어떻게 발급해주는데?

- 사용자는 동의 화면에서 서비스 이용 시 필요한 사용자 정보 및 권한 제공에 동의하고 로그인을 요청하거나 로그인을 취소할 수 있습니다.
- 사용자가 필수 동의 항목에 모두 동의한 뒤 [동의하고 계속하기] 버튼을 누르면,
- 카카오 인증 서버는 해당 사용자에 대한 인가 코드를 발급해주고
- 이를 서비스의 [redirect_uri](#)에 전달해줍니다.

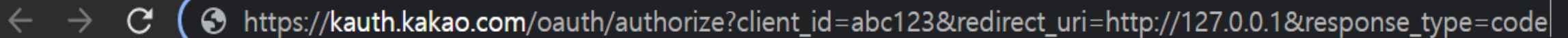
| Name | Type | Description | Required |
|---------------|--------|--|----------|
| client_id | String | 앱 REST API 키 [내 애플리케이션] > [앱 키]에서 확인 가능 | O |
| redirect_uri | String | 인가 코드가 리다이렉트될 URI | O |
| response_type | String | code로 고정 | O |


1. 카카오 로그인을 위한 인가 코드 (Authorize code) 요청을 url을 통해 보내봅시다.

```
https://kauth.kakao.com/oauth/authorize?client_id={REST_API_KEY}&
redirect_uri=http://127.0.0.1&response_type=code
```

(ex) REST_API_KEY 가 abc123 이라면,

```
https://kauth.kakao.com/oauth/authorize?client_id=abc123&redirect_uri=http://127.0.0.1&response_type=code
```



← → ↻  https://kauth.kakao.com/oauth/authorize?client_id=abc123&redirect_uri=http://127.0.0.1&response_type=code|

url을 크롬창이나 인터넷 브라우저의 url에 복사 붙여넣기 하면 요청이 가게 됩니다!

Python 심화

카카오 메시지 API



저희가 미리 사전 과제에서 설정해둔
닉네임, 프로필 사진 필수 동의 옵션 및 카카오톡 메시지 전송 선택 동의 옵션이 담긴
동의항목 창이 나타날 것입니다!

이렇게 나타난 창이 바로 카카오가 저희의 인가코드 받기 요청에 대한 응답인, 인증 및 동의를 요구하는 창입니다.

APP

서강 멋사 10기 최고♥
sogang

☒ **전체 동의하기**
전체동의를 선택목적에 대한 동의를 포함하고 있으며, 선택목적에 대한 동의를 거부해도 서비스 이용이 가능합니다.

 junior0614@naver.com

[계정 변경](#)

서강 멋사 10기 최고♥ 서비스 제공을 위해 회원번호와 함께 개인정보가 제공됩니다. 보다 자세한 개인정보 제공항목은 동의 내용에서 확인하실 수 있습니다. 정보는 서비스 탈퇴 시 지체없이 파기됩니다.

동의하고 계속하기

APP

서강 멋사 10기 최고♥
sogang

☒ **전체 동의하기**
체없이 파기됩니다.

☒ [필수] 필수 제공 항목

[보기](#)

닉네임, 프로필 사진

☐ [선택] 서비스 접근 권한

☒ 카카오톡 메시지 전송

동의하고 계속하기

Python 심화

카카오 메시지 API



Python 심화

카카오 메시지 API

이제 인증 및 동의를 요구하는 카카오에게 동의를 해준다면 로그인 완료됩니다.

이때 반드시 **전체 동의** 하기를 눌러 주신 후 **동의하고 계속하기** 를 진행해주세요!
이렇게 해주시면 로그인이 완료된 상태가 됩니다.

The diagram illustrates the Kakao login consent process in three stages:

- Initial Screen:** Shows the app '서강 멋사 10기 최고♥' (sogang) with the '전체 동의하기' (Agree to all) option selected. The '동의하고 계속하기' (Agree and continue) button is at the bottom.
- Transition:** A yellow arrow points from the initial screen to the final screen.
- Final Screen:** The '전체 동의하기' option is highlighted with a red box. The '동의하고 계속하기' button is also highlighted with a red box.

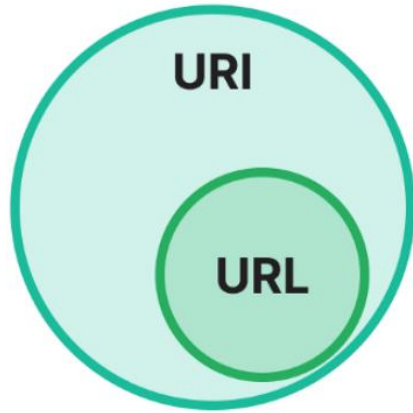
Python 심화

카카오 메시지 API



1) 서버에서 Redirect URI 서비스에서 필요한 로그인 인증 정보를 보냅니다.

2) 서비스는 **Redirect URI** 로 받은 로그인 인증 정보를 처리해 다음 단계 요청을 보냅니다.



URI

- URI는 특정 리소스를 식별하는 통합 자원 식별자(Uniform Resource Identifier)
- 웹 기술에서 사용하는 논리적 또는 물리적 리소스를 식별하는 고유한 문자열 시퀀스

URL

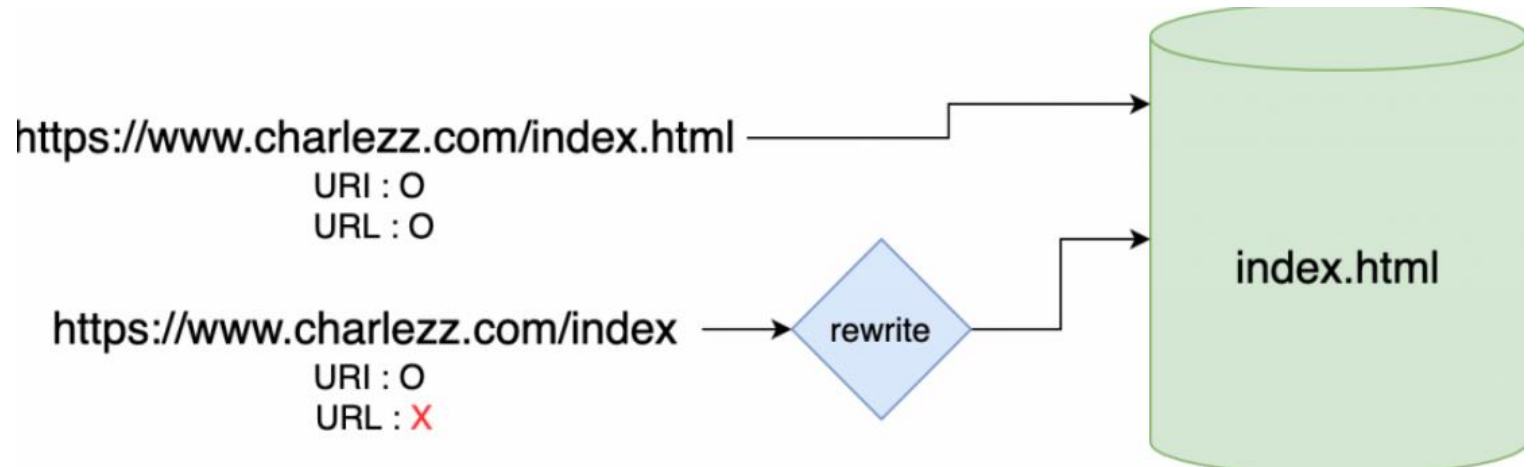
- 웹 주소
- 컴퓨터 네트워크 상에서 리소스가 어디 있는지 알려주기 위한 규약. URI의 부분집합

첫번째 주소

- 웹서버의 실제 파일 위치를 나타내는 주소
이므로 URI이면서 URL

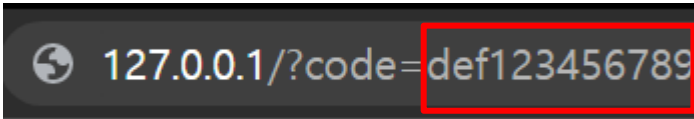
두번째 주소

- 실제로 index라는 파일이 웹서버에 존재하지 않으므로 URL X
- BUT 서버 내부에서 이를 처리하여 결국 index.html을 가리키기 때문에 URI



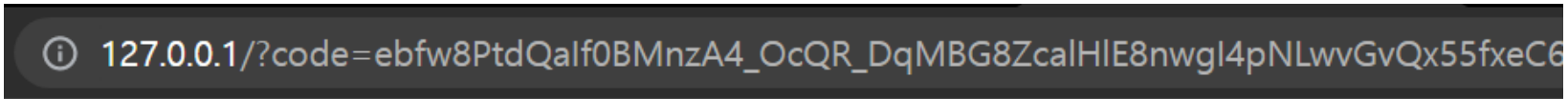
출처 : <https://www.charlezz.com/?p=44767>

2. example.com/oauth?code=에서 '=' 이후의 코드를 복사해주세요. (빨간 네모 박스 부분만 저장)
바로 이 부분이 저희가 로그인을 성공한 후 받게 되는 인가 코드입니다!



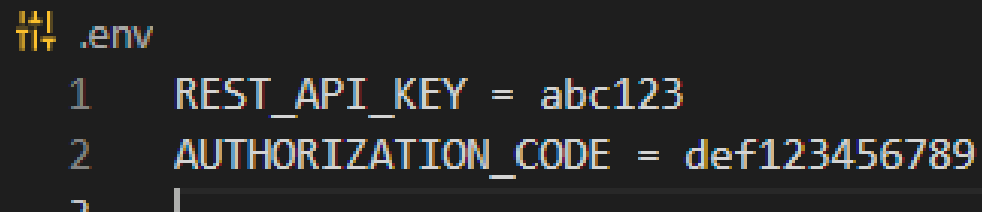
127.0.0.1/?code=def123456789

실제로는 인가코드가 아래와 같이 길게 옵니다! 저는 보안상 자른 것이니 당황하지 말아주세요!



127.0.0.1/?code=ebfw8PtdQalf0BMnzA4_OcQR_DqMBG8ZcalHIE8nwgI4pNLwvGvQx55fxeC6

.env 파일에 인가 코드 환경변수도 만들어둡시다.



```
1 REST_API_KEY = abc123
2 AUTHORIZATION_CODE = def123456789
3
```

Python 심화

카카오 메시지 API



2. 인가코드로 토큰을 발급하는 요청을 진행하겠습니다. 이를 위해 필요한 모듈들을 먼저 import 해와줍시다.

```
from dotenv import load_dotenv
import os
#환경변수를 위한 모듈 (저번 시간에 사용했었죠!?)
import requests
# 카카오에게 요청을 보내기 위한 모듈
import json
#python 객체를 JSON 데이터로 만들기 위해 쓰이는 파이썬의 내장 json 모듈
```

GET 방식: requests.get() #가져오기
POST 방식: requests.post() #보내기
PUT 방식: requests.put()
DELETE 방식: requests.delete()

(ex)
파이썬 객체 그 자체 출력 예시 :
<__main__.pineapple object at 0x7f8749ee9f28>

Python 객체를 JSON 데이터로 쓰기, 직렬화 (Serialize), 인코딩
json.dump() => 파이썬 객체 직렬화 후 디스크에 저장해주는 것

JSON 포맷 데이터를 Python 객체로 읽기, 역직렬화, 디코딩
json.loads() => 디스크에 있는 JSON 포맷 데이터를 Python 객체로 읽어오기



**이제부터 작성할 코드는 다 작성 후,
두 번 이상 run 하면 안됩니다!**



**authorization code로 토큰을 발급받는 것은
한번만 가능하기 때문입니다!**

두 번 이상 인가코드로 토큰을 발급받으려 하면
에러 response 가 반환되고,
해당 에러 response가 json 파일에 덮여 저장될 것이므로,
토큰을 다시 발급 받는 과정을 거쳐야 합니다.



두 번 이상 실행시키지 말아주세요!

(저도 알고 싶지 않았습니다 ㅎㅎ..)

만약 두 번 이상 누르셨다고 하면,
다시 인가코드 발급 후 앞과
똑같은 과정 진행해주시면 됩니다!



Python 심화

카카오 메시지 API

```
load_dotenv() #환경변수 호출
rest_api_key = os.getenv("REST_API_KEY")
# 환경변수에서 REST_API_KEY로 저장된 값 가져와서
rest_api_key로 저장

authorization_code = os.getenv("AUTHORIZATION_CODE")
# 환경변수에서 AUTHORIZATION_CODE로 저장된 값
# 가져와서 authorization_code 로 저장

url = 'https://kauth.kakao.com/oauth/token'
# 토큰 발급을 해달라고 요청할 url

redirect_uri = 'http://127.0.0.1'
# 발급된 토큰을 반환해줄 url

data = {
    'grant_type': 'authorization_code',
    'client_id': rest_api_key, # 클라이언트 인증 키
    'redirect_uri': redirect_uri, # 응답을 반환할 url
    'code': authorization_code, # 인가 코드
}

# data는 토큰을 발급할 때 카카오가 요구하는 정보들입니다.
```

```
response = requests.post(url, data=data)
# 우리가 설정한 토큰 발급을 요청할 url에 , 카카오가 토큰 발
급을 위해 필요하다고 한 정보들을 넣어
# 토큰을 발급해달라는 요청을 보내면 돌아올 응답을 response에
답습니다.

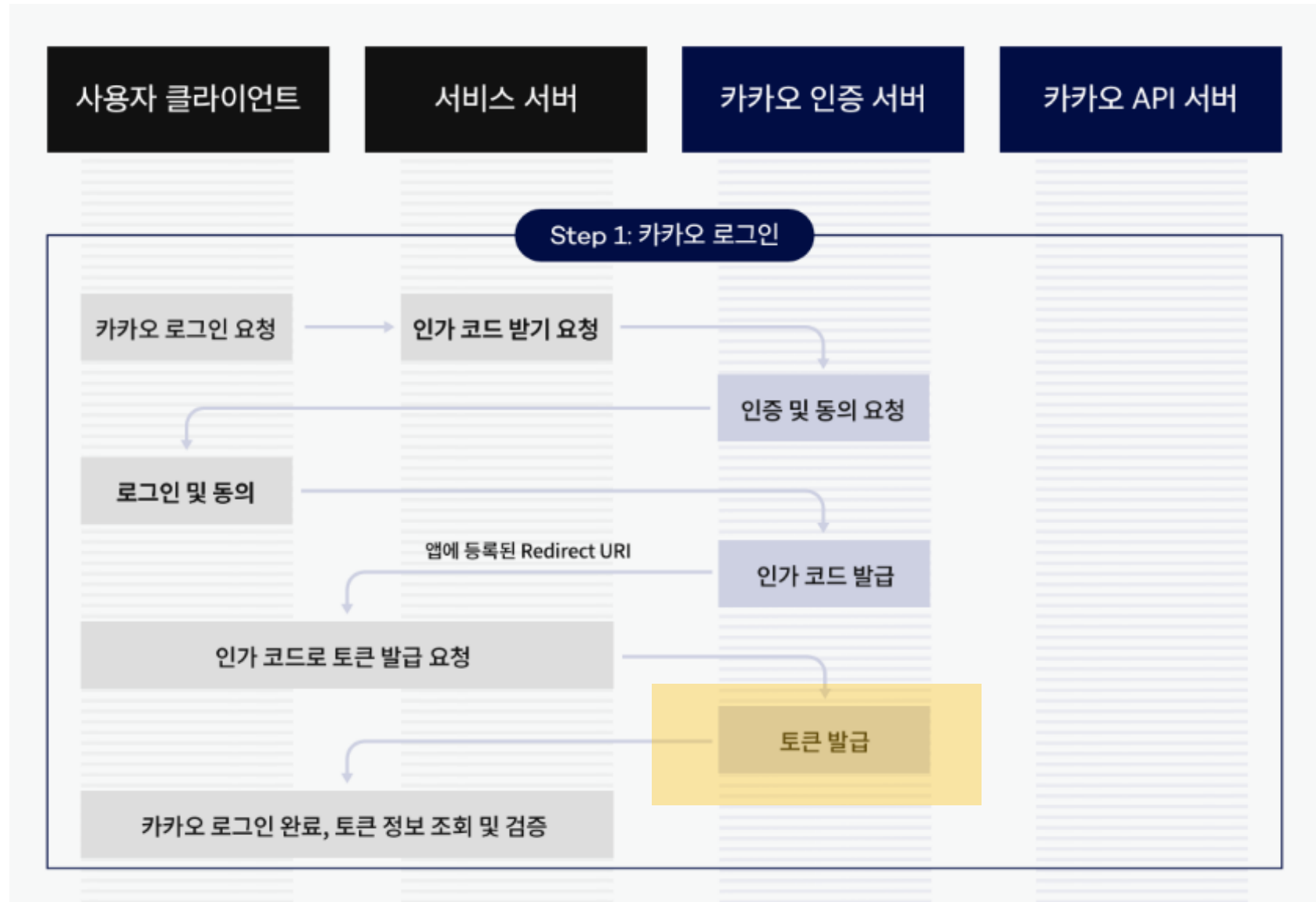
tokens = response.json()
# 토큰 발급 응답을 json 형태로 변환해줍니다.

print(tokens)
# 우리의 터미널에 한번 토큰을 출력해줍니다. (확인용)

with open("kakao_token.json", "w") as json_file:
    # "kakao_code_json" 이라는 이름의 파일을 쓰기("w") 모드로
    열어놓고,
    json.dump(tokens, json_file)
# json으로 내보내고자 하는 파이썬 객체인 tokens를
# 직렬화된 데이터가 쓰여질 파일 json_file 에 쓰기를 해주었습
니다
```

Python 심화

카카오 메시지 API



터미널에 아래와 같이 찍히게 되면 응답 성공입니다.

```
{ 'access_token': 'r96xrqGz_AvqW3NxCqK-QSmmZJPV1HLyT8QPEgo9c-wAAAGAOHAEYA', 'token_type': 'bearer',  
'refresh_token': 'aYT9j9kbq26XU-h0-_d-16LWRhsR3CUr2wKb1go9c-wAAAGAOHAEXg', 'expires_in': 2159  
9, 'refresh_token_expires_in': 5183999 }
```

access token : 사용자 인증 수단 (적절한 로그인 사용자에게 주는 서비스 이용 티켓)

refresh token : access_token을 재발급 해주기 위한 토큰

expires_in , refresh_token_expires_in : 각 access token, refresh token의 만료 시간

일반적으로 access token의 만료 시간은 짧고, refresh token은 만료 시간이 더 깁니다!

Python 심화

카카오 메시지 API

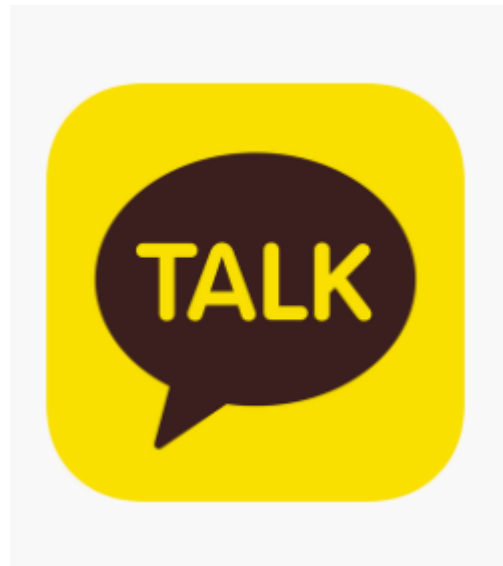
저희가 짰 코드에서 받은 응답을 json 형태로 저장도 되었을 것입니다.
확인했을 때 아래와 같이 토큰을 발급받았을 때
미리 print문으로 터미널에 찍힌 response 와 같은 형태가 json에 저장됐다면 성공입니다.

```
{-} kakao_code.json > token_type
1 [{"access_token": "t_RkC7N-NVuUWIG0Ttem21-AgR1CS_5cIGbVNAo9dGgAAAGAO3wTnw", "token_type": "bearer", "refresh_token": "RD1X0628WoRAZm2f0pxEj", "expires_in": 21599, "refresh_token_expires_in": 5183999}]
```

```
{-} kakao_code.json > ...
1 ✓ [{"access_token": "t_RkC7N-NVuUWIG0Ttem21-AgR1CS_5cIGbVNAo9dGgAAAGAO3wTnw", "token_type": "bearer", "refresh_token": "RD1X0628WoRAZm2f0pxEj", "expires_in": 21599, "refresh_token_expires_in": 5183999}]
```

Json 파일에 담긴 요소는
왼쪽과 같아야 할 것입니다.

이제 로그인 된 유저임을
인증해주는 토큰들도 발급을 성공적으로 받았으니,
메시지 보내기 요청을 보내는 코드를 작성해봅시다!



Python 심화

카카오 메시지 API (다른파일)

```
load_dotenv()
#환경변수 불러오기
rest_api_key = os.getenv("REST_API_KEY")
# 환경변수에서 REST_API_KEY로 저장된 값 가져와서 rest_api_key로 저장
with open("kakao_token.json","r") as json_file:
    tokens = json.load(json_file)
# 토큰 response를 담은 json 파일을 tokens라는 변수로 불러와 담아줍니다.

url="https://kapi.kakao.com/v2/api/talk/memo/default/send"
# 우리가 요청을 보낼 url

headers={
    "Authorization" : "Bearer " + tokens["access_token"]
}#카카오 api 서비스 사용을 위해선 내가 서비스를 이용할 수 있다는 유저임을
인증하기 위한
# authorization 토큰을 요청하는 헤더에 담아서 보내주기
# 이때 tokens에서 access token이라는 키를 가지는 값을 데려와서 값을 넣어주기

data={ #우리가 메시지로 보낼 내용
    "template_object": json.dumps({ #메모리 상에 JSON 포맷 데이터를
        "object_type":"text", # 우리가 보내는 객체의 타입은 text
        "text":"우리는 서강대 멋사 10기야~", #우리가 보내는 텍스트 내용
        "link":{ # 우리가 보낼 링크 주소 (임의로 작성해줘도 됩니다.)
            "web_url":"https://www.instagram.com/likelion_sg"
        }
    })
}
```

```
response = requests.post(url, headers=headers, data=data)
# 요청을 보낼 때 우리가 위에서 변수로 선언한
# 1) 요청 보낼 url 2) 액세스 토큰이 담긴 헤더 3) 보낼 메시지 (json data)
# 요청을 보내면 그에 대한 응답이 올텐데, 이를 response로 담아오기

print(response.status_code)
# 카카오가 보내주는 응답에는 항상 코드가 존재합니다!
# status code가 200 이라면 요청이 제대로 수행된 것입니다.

if response.json().get('result_code') == 0:
    # result code가 0이라면 성공적으로 메시지가 와있을 것입니다.
    print('메시지를 성공적으로 보냈습니다!')
else:
    print('메시지를 성공적으로 보내지 못했습니다. 오류메시지 : ' +
str(response.json()))
# result code가 0이 아니라면 제대로된 요청이 수행되지 않은 것이므로
오류 메시지를 찍어봅시다
info_url = "https://kapi.kakao.com/v2/user/scopes"
params = {"secure_resource": True}
info_res = requests.get(info_url, headers=headers, params=params)
# 동의항목이 잘 설정되어 있는지 확인해주기 위해서 수행하는 요청입니다.
# 이 요청을 수행하면 우리가 동의를 요구한 항목들의 상태가
#true , false로 나옵니다.
print(info_res.json())
```

Python 심화

카카오 메시지 API

성공 시 터미널에 찍히는 문자열

```
200
메시지를 성공적으로 보냈습니다!
```

실패 시 찍히는 문자와 동의항목 여부

1) 액세스 토큰이 유효하지 않을 때

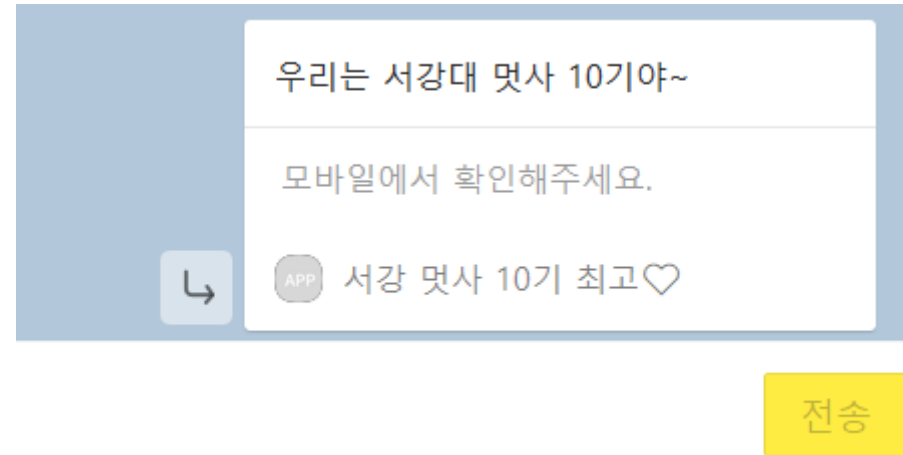
```
401
메시지를 성공적으로 보내지 못했습니다. 오류메시지 : {'msg': 'this access token does not exist', 'code': -401}
{'msg': 'this access token does not exist', 'code': -401}
```

(+) 동의항목 정상 형태 (만약 이 중에 하나라도 true 가 안되어있으면 동의가 제대로 안되어 에러가 난 것입니다.)

```
{'id': 2206968497, 'scopes': [{ 'id': 'profile_nickname', 'display_name': 'Nickname', 'type': 'PRIVACY', 'using': True, 'agreed': True, 'revocable': False}, { 'id': 'profile_image', 'display_name': 'Profile image', 'type': 'PRIVACY', 'using': True, 'agreed': True, 'revocable': False}, { 'id': 'talk_message', 'display_name': 'Send message in KakaoTalk', 'type': 'SERVICE', 'using': True, 'agreed': True, 'revocable': True}]}
```

2) 동의항목이 제대로 설정되지 않았을 시 응답 (ex : talk_message 가 agreed 되지 않음)

```
403
메시지를 성공적으로 보내지 못했습니다. 오류메시지 : {'msg': 'insufficient scopes', 'code': -402, 'api_type': 'TALK_MEMO_DEFAULT_SEND', 'required_scopes': ['talk_message'], 'allowed_scopes': []}
{'id': 2206483816, 'scopes': [{ 'id': 'talk_message', 'display_name': 'Send message in KakaoTalk', 'type': 'SERVICE', 'using': True, 'agreed': False}]}
```

나와의 카톡을 확인해보았을 때 이렇게 와있으면 성공입니다!



수고하셨습니다

