개인 포트폴리오_김민준

빅데이터를 활용한

SNS 인기 여행지

대한민국 구석구석 SNS 인기 여행지 데이터 수집, 저장 및 RPA 활용

대한민국 구석구석
+
인스타그램
+
RPA

1. 추진 목표

SNS 인기 여행지 데이터 수집, 저장 및 RPA 활용

- 대한민국 구석구석의 인스타그램 인기 여행지 데이터 수집 및 저장
- 수집 데이터를 저장 및 RPA활용(이메일 전송, 카카오톡 메시지 전달)

2. 과제 수행 범위

대한민국 구석구석

인스타그램 인기 여행지
TOP 11 수집

여행지명, 지역, 상세정보,
해시태그, 여행지별 이미지 3장

Instagram

여행지명에 따른 게시물
개수 수집

인기 여행지일수록 게시물
개수 증가

NAVER

네이버 날씨 크롤링

오늘,내일,모레

3. 요구사항 정의

인스타그램 게시물 수집

여행지 명이 불명확할 경우 공백처리

CSV, JSON 파일 저장

업무용 혹은 웹 데이터 활용

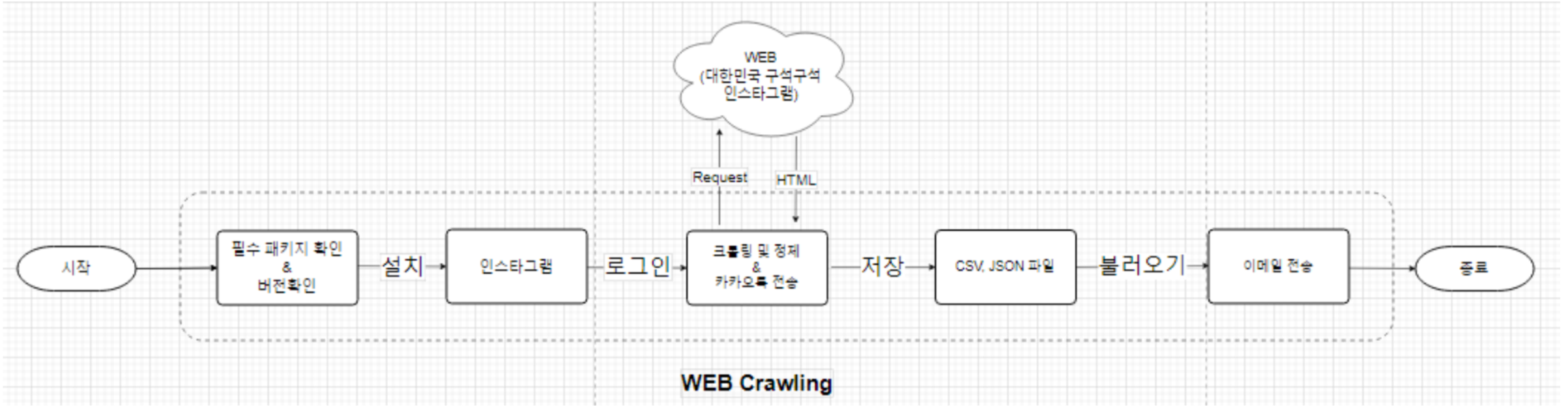
파일 이메일 전송

이메일 자동화

카카오톡 메시지 전송

카카오톡 자동화

4. 프로세스 설계



1. 필수 패키지 확인 & 버전확인

데이터 수집에 필요한 패키지 설치 및 최신버전 확인

2. 인스타그램

데이터 수집 전 인스타그램 로그인

3. 크롤링 및 정제 & 카카오톡 전송

대한민국 구석구석 수집 및 여행지 명에 맞는 인스타그램 수집, 수집 내용 카카오톡 피드 형식으로 전송

4. CSV, JSON 파일 저장

데이터 수집내용 csv 파일과 json 파일 저장

5. 이메일 전송

csv파일과 json 파일을 이메일 전송

5. 구현

5-1) 필수 패키지 확인 및 버전 확인

```python
print('='*120)
print('''
필요 모듈 설치 여부를 확인합니다.\n
체크 모듈 : requests, pandas,selenium, bs4
''')
print('='*120)

import sys # 파이썬 인터프리터의 변수와 함수를 직접 제어하기 위해 호출
import subprocess # 모듈들 확인 및 설치를 위한 쉘 명령 실행 모듈 호출

try :
    import requests # 웹페이지 url 서칭을 위한 requests 모듈 호출
    import pandas # 데이터프레임 저장을 위한 pandas 모듈 호출
    import selenium # 웹페이지 제어를 위한 selenium 모듈 호출
    import bs4 # html파싱을 위한 BeautifulSoup 모듈 호출


except :
    # pip 모듈 업그레이드
    subprocess.check_call([sys.executable, "-m", "pip", "install", "--upgrade", "pip"])
    # requests 모듈 최신버전 설치
    subprocess.check_call([sys.executable, "-m", "pip", "install", "--upgrade", "requests"])
    # pandas 모듈 최신버전 설치
    subprocess.check_call([sys.executable, "-m", "pip", "install", "--upgrade", "pandas"])
    # selenium 모듈 최신버전 설치
    subprocess.check_call([sys.executable, "-m", "pip", "install", "--upgrade", "selenium"])
    # BeautifulSoup 모듈 최신버전 설치
    subprocess.check_call([sys.executable, "-m", "pip", "install", "--upgrade", "bs4"])
    print('='*120)
    print('필요 모듈 설치가 완료되었습니다.')
    print('='*120)

else:
    print('='*120)
    print('필요 모듈 설치가 완료되어 버전을 확인합니다.')
    print('='*120)
    # BeautifulSoup 모듈 최신버전 확인
    print('requests module version check :'+ bs4.__version__)
    # pandas 모듈 최신버전 확인
    print('pandas module version check :'+ pandas.__version__)
    # selenium 모듈 최신버전 확인
    print('selenium module version check :'+ selenium.__version__)
    # requests 모듈 최신버전 확인
    print('requests module version check :'+ requests.__version__)
    print('='*120)
```

```
========================================================================================================================

필요 모듈 설치 여부를 확인합니다.

체크 모듈 : requests, pandas,selenium, bs4

========================================================================================================================
========================================================================================================================
필요 모듈 설치가 완료되어 버전을 확인합니다.
========================================================================================================================
requests module version check :4.10.0
pandas module version check :1.3.4
selenium module version check :4.1.0
requests module version check :2.26.0
========================================================================================================================
```

5. 구현

5-2) 인스타그램

```
'''
함수명 : instagram_login
기능 : 인스타그램 로그인(id,pw) 입력
매개변수 : user_id , user_pw
- user_id : (str) 인스타그램 아이디
- user_pw : (str) 인스타그램 패스워드
리턴값 : 없음
'''

# 인스타그램 로그인 함수 정의
def instagram_login(user_id, user_pw) :
    # 아이디 입력
    id_input = driver.find_element_by_xpath('//*[@id="loginForm"]/div/div[1]/div/label/input') # 인스타그램 로그인 입력창
    id_input.send_keys(user_id) # 인스타그램 아이디 입력
    time.sleep(1) # 아이디 입력 후 1초 대기

    # 비밀번호 입력
    pw_input = driver.find_element_by_xpath('//*[@id="loginForm"]/div/div[2]/div/label/input') # 인스타그램 패스워드 입력창
    pw_input.send_keys(user_pw) # 인스타그램 비밀번호 입력
    time.sleep(1) # 비밀번호 입력 후 1초 대기
    pw_input.send_keys(keys.ENTER)  # 비밀번호 입력창에 엔터 입력
    time.sleep(120)


'''
함수명 : make_cooki
기능 : 인스타그램 로그인 성공 쿠키를 만듦
매개변수 : driver , user_id
- driver : 웹 드라이버 변수 호출
- user_id : (str) 인스타그램 아이디
리턴값 : 없음
'''

def make_cooki (driver, user_id) : # 쿠키만들기 함수정의
    with open(f'./cookies/instagram_{user_id}_cookies.pkl', 'wb') as f : # cookies폴더에 인스타그램 아이디와 동일한 pkl(쿠키) 파일 만들기
        pickle.dump(driver.get_cookies(),f) # 웹드라이버로 쿠키를 가져온다.


'''
함수명 : load_cooki
기능 : 인스타그램 로그인 성공 쿠키를 불러오기
매개변수 : driver , user_id
- driver : 웹 드라이버 변수 호출
- user_id : (str) 인스타그램 아이디
리턴값 : 없음
'''

def load_cooki(driver, user_id) : # 쿠키 불러오기 함수정의
    with open(f'./cookies/instagram_{user_id}_cookies.pkl','rb') as f : # cookies폴더에 인스타그램 아이디와 동일한 pkl(쿠키) 파일 불러오기
        cookies = pickle.load(f) # cookies 변수에 쿠키파일를 불러오고, 저장

        for cookie in cookies : # cookies 리스 cookie를 반복
            driver.add_cookie(cookie) # 웹드라이버에 cookie 정보를 추가


# 쿠키방식 로그인
if os.path.isfile(f'./cookies/instagram_{user_id}_cookies.pkl') : # 경로에 쿠키파일이 있다면
    print('='*120)
    print('쿠키가 존재합니다. 인스타그램 로그인에 성공하였습니다.')
    print('='*120)
    load_cooki(driver, user_id) # 쿠키 불러오기 함수 호출
    driver.refresh() # 웹드라이버 새로고침


else : # 경로에 쿠키파일이 없다면
    print('='*120)
    print('쿠키가 존재하지 않습니다. 인스타그램 로그인을 시도합니다.')
    print('='*120)
    instagram_login(user_id, user_pw) # 인스타그램 로그인 함수 호출
    make_cooki(driver,user_id) # 쿠키 만들기 함수 호출
```

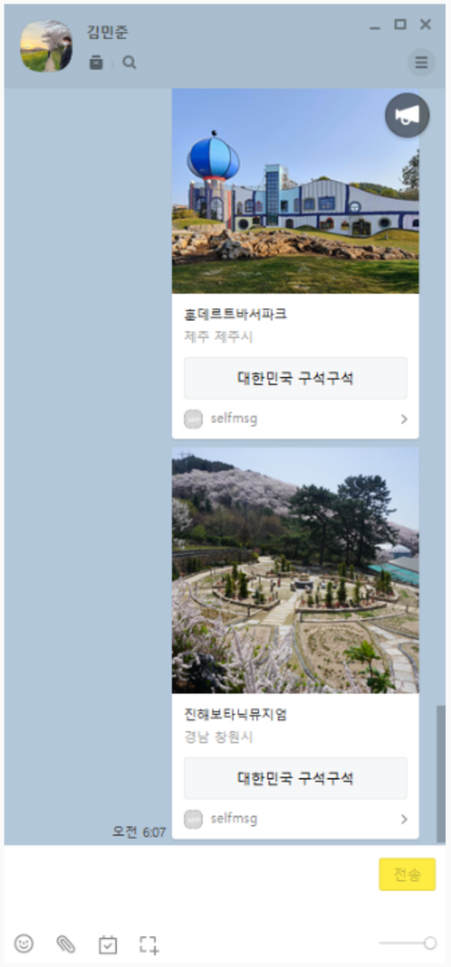
```
========================================================================================================================
쿠키가 존재합니다. 인스타그램 로그인에 성공하였습니다.
========================================================================================================================
```

5. 구현

5-3) 크롤링 및 정제 & 카카오톡 전송

```
'''
함수명 : get_korea
기능 : 대한민국 구석구석 SNS 인기여행지 TOP11 수집
매개변수 : 없음
리턴값 : 없음

'''

def get_korea() :  # 대한민국 구석구석 top 11 수집 함수 정의
    #리턴 값 고민하기
    for x in range(1,7) : # 1부터 6까지 반복
        driver.find_element_by_xpath(f'/html/body/div[3]/div[3]/div[1]/div[1]/ul[1]/li[{x}]').click()  # 대한민국 구석구석 인스타그램 추천 여행지 1번째 줄
        time.sleep(3)
        detail_page() # 상세페이지 함수 호출
        time.sleep(3) # 웹페이지 로딩을 위한 3초 대기


    # 넘어 갈 때 자바 스크립트 동작 에러 발생
    for y in range(1,6) : # 1부터 5까지 반복
        driver.find_element_by_xpath(f'/html/body/div[3]/div[3]/div[1]/div[1]/ul[2]/li[{y}]').click()  # 대한민국 구석구석 인스타그램 추천 여행지 2번째 줄
        time.sleep(3) # 웹페이지 로딩을 위한 3초 대기
        detail_page() # 상세페이지 함수 호출


    save_data.csv_save() # save_data 모듈의 csv 저장 함수호출
    time.sleep(2) # 로딩을 위한 2초 부여
    save_data.json_save()  # save_data 모듈의 json 저장 함수호출
    time.sleep(2) # 로딩을 위한 2초 부여
    send_mail.email(full_dir) # send_mail 모듈의 email 함수 호출(csv,json파일을 이메일로 보내준다.)

    driver.close() # 창닫기
```

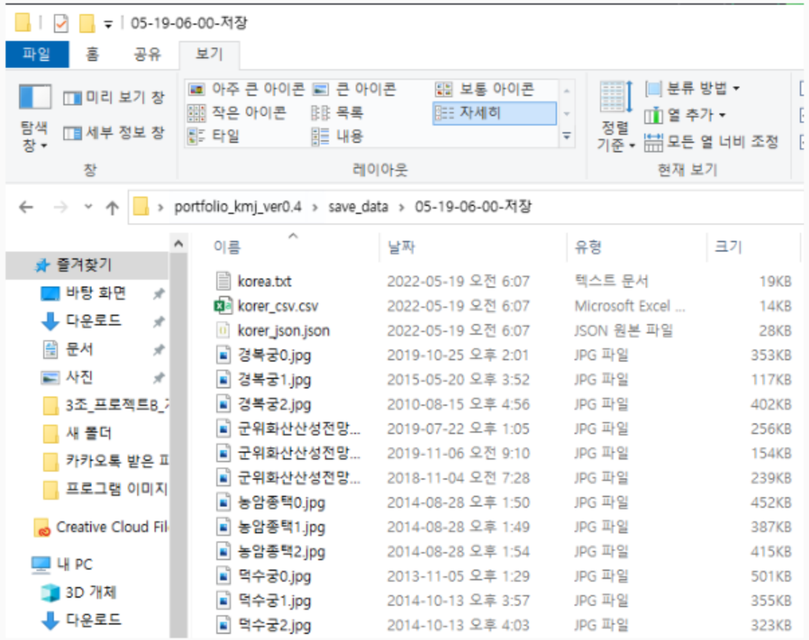
```
==========================================================================================================================
설악산국립공원 사진 저장을 완료하였습니다.
==========================================================================================================================
여행지: 설악산국립공원 / 강원 속초시
검색태그:  설악산국립공원 / 인스타그램 게시물 수:  44,097 건
상세정보: 설악산은 강원도 속초시를 포함해 고성과 인제, 양양군에 걸쳐 있는 유명한 자연 명소이다. 1970년 한려해상국립공원에 이어 다섯 번째로 국립공원 명칭을 달았으며, 1982년에는 국내 최초로 유네스코 생물권보전지역에 지정됐다. 또한 국제적으로 우수하게 관리되고 있는 보호지역 명단인 세계자연보전연맹(IUCN) 녹색목록에도 등재되어 있다. 설악산국립공원은 총면적이 약 398km²에 이를 만큼 광대한 규모를 자랑한다. 주봉인 대청봉을 기준 삼아 속초시에 접한 동쪽은 외설악, 서쪽은 내설악, 한계령과 오색 방면은 남설악으로 구분한다. 한라산과 지리산 다음으로 높은 대청봉은 1년 중 5~6개월은 눈에 덮여 있는데 '설악'이란 이름은 여기서 유래되었다. 대청봉에 오르면 수려한 자연 경관을 품은 설악산 전경과 동해 바다를 조망할 수 있으며 특히 일출과 낙조가 일품으로 꼽힌다. 대청봉 외에도 소청봉, 화채봉, 중청봉 등 30여개의 산봉우리들이 웅장한 장관을 이룬다. 이외에 비룡폭포, 울산바위, 흔들바위, 금강굴 등 숨은 비경들이 가득하다. 설악산 소공원 인근에 운영 중인 케이블카를 이용하면 노약자나 장애인도 어렵지 않게 설악산 풍경을 감상할 수 있다.[체험프로그램]등산, 트래킹
URL :  https://korean.visitkorea.or.kr/detail/ms_detail.do?cotid=4db10875-c210-476a-9873-dc307faccf47&big_category=A01&mid_category=A0101&big_area=32
해시태그:  #21-22한국관광100선#21_22한국관광100선#강원권#강원도_여행지_추천#관광지#국립공원#다양한식물#비룡폭포#설악동아영장#설악산국립공원#식물자원외_보고
==========================================================================================================================
카카오톡 메시지를 보냅니다.
==========================================================================================================================
==========================================================================================================================
카카오톡 메시지 전송 성공하였습니다.
==========================================================================================================================
```



5. 구현

5-4) CSV, JSON 저장

```python
'''
함수명 : csv_save
기능 :  txt 파일을 csv 파일로 저장하는 함수
매개변수 : 없음
리턴값 : 없음
'''
def csv_save() : # txt 파일 -> csv 파일로 저장 함수 정의

    print('='*120)
    print('CSV 파일로 저장을 시작합니다') # csv 저장 시작 출력
    print('='*120)
    try :  # 해당 경로에 txt파일이 있다면 파일 불러오기
        csv_data = pd.read_csv('./korea.txt', sep='\t',encoding='utf-8', lineterminator='\n', error_bad_lines=False)
    except : # 해당 경로에 txt파일이 없을 경우
        print('='*120)
        print('CSV 파일로 저장을 실패했습니다.(파일유무를 확인하세요.)')
        print('='*120)

    else :
        # csv 변환
        csv_data.columns = ['여행지명','위치','인스타그램 게시물 수','상세정보','해시태그']
        # csv 파일로 저장
        csv_data.to_csv('./korer_csv.csv', encoding='utf-8-sig', index=False)
        # 저장 완료 출력
        print('='*120)
        print('CSV 파일로 저장을 완료했습니다.')
        print('='*120)

#csv_save(full_dir) # 테스트 출력


'''
함수명 : json_save
기능 :  csv 파일을 json 파일로 저장하는 함수
매개변수 : 없음
리턴값 : 없음

'''
def json_save() : # csv파일 -> json 파일로 저장 함수 정의
    print('='*120)
    print('JSON 파일로 저장을 시작합니다') # json 저장 시작 출력
    print('='*120)
    try : # 해당 경로에 csv파일 있다면 파일 불러오기
        json_data = pd.read_csv('./korer_csv.csv',sep=',',encoding='utf-8')
    except : # 해당 경로에 csv파일이 없을 경우
        print('='*120)
        print('JSON 파일로 저장을 실패했습니다.(CSV 파일유무를 확인하세요.)')
        print('='*120)

    else:
        # json 파일로 저장
        json_data.to_json('./korer_json.json', orient='records')
        # 저장완료 출력
        print('='*120)
        print('JSON 파일로 저장을 완료했습니다.')
        print('='*120)
```

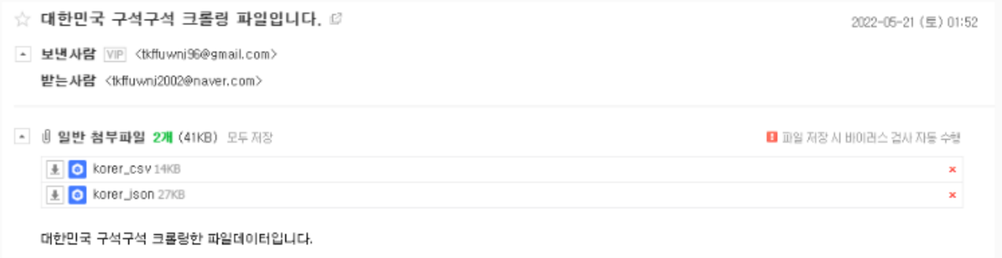



```
============================================================================================================================
CSV 파일로 저장을 시작합니다
============================================================================================================================
c:\Users\jmk\Desktop\portfolio_kmj_ver0.4_20220520\portfolio\complete_code\korea_crawling.py:180: FutureWarning: The erro
 and will be removed in a future version.


  save_data.csv_save() # save_data 모듈의 csv 저장 함수호출
============================================================================================================================
CSV 파일로 저장을 완료했습니다.
============================================================================================================================
============================================================================================================================
JSON 파일로 저장을 시작합니다
============================================================================================================================
============================================================================================================================
JSON 파일로 저장을 완료했습니다.
============================================================================================================================
```

5. 구현

5-5) 이메일 전송

```python
'''
함수명 : email
매개변수 : full_dir
- full_dir : 파일 저장된 경로가 저장된 변수
리턴값 : 없음

'''
def email(full_dir): # 이메일 보내는 함수 정의
    print('='*120)
    print('메일을 보내는 중입니다.') # 이메일 시작 메시지 출력
    print('='*120)

    # 전자메일 메시지 작성을 위한 모듈 호출
    msg = EmailMessage()
    msg["Subject"] = "대한민국 구석구석 크롤링 파일입니다." # 이메일 제목
    msg["From"] = email_address # 보내는 사람
    msg["To"] = "tkffuwnj2002@naver.com" # 받는 사람
    msg.set_content('대한민국 구석구석 크롤링한 파일데이터입니다.') # 이메일 본문 텍스트

    # 구글 -> mimetype을 검색하면 검색하는 포맷이 나온다.
    # 보내려는 파일 타입을 찾은 후 메인,서브 속성에 넣으면 된다.
    # json => minatype ='application' , sub = json
    # csv => text , csv
    # 파일이름에 문자열을 입력하면 입력값으로 파일이 들어간다.

    # csv파일 불러오기
    with open('./korer_csv.csv', 'rb') as f_csv :
        # msg 객체에 파일 첨부를 활성화(open한 파일을 읽고, 메인타입=, 서브타입=, 파일이름=)
        msg.add_attachment(f_csv.read(), maintype ='text', subtype='csv', filename = 'korer_csv')

    # json파일 불러오기
    with open('./korer_json.json','rb') as f_json :
        # msg 객체에 파일 첨부를 활성화(open한 파일을 읽고, 메인타입=, 서브타입=, 파일이름=)
        msg.add_attachment(f_json.read() ,maintype ='application', subtype='json', filename = 'korer_json')

    # 이메일 보내기
    with smtplib.SMTP("smtp.gmail.com",587) as smtp :
        smtp.ehlo() # 연결이 잘 되는지 확인
        smtp.starttls() # 메일 내용이 암호화되어 전송
        smtp.login(email_address,email_pw) # 로그인에 필요한 이메일 주소 및 비밀번호 함수 호출
        smtp.send_message(msg) # 메시지 보내기


    print('='*120)
    print('메일을 보내기를 완료했습니다.') # 이메일 완료 메시지 출력
    print('='*120)
```




```
==========================================================================================
메일을 보내는 중입니다.
==========================================================================================
==========================================================================================
메일을 보내기를 완료했습니다.
==========================================================================================
```

THANK YOU

이름 / 연락처

김민준 / 010-9608-2015

이메일

tkffuwnj2002@naver.com

GitHub

https://github.com/jmk2015
git/portfolio_koreatrip.git