

# 오픈API 연동 (3) - 카카오 OpenAPI 연동

카카오개발자사이트에 로그인을 하고 난 후 발급받는 REST API Key를 사용하면 카카오 Open API를 통해 다음에서 제공되는 검색 결과를 수집할 수 있습니다.

## #01. 연동 준비하기

### 1) 연동키 발급받기

- 아래 주소에 접속 후 로그인  
<https://developers.kakao.com/>
- 로그인 후 아래 페이지에서 새 앱 생성  
<https://developers.kakao.com/apps/new>
- 생성된 key 중에서 REST API Key 를 보관
- 개발가이드 > REST API 개발가이드 > 검색 메뉴로 이동하여 연동 스펙 확인

### 2) 필요한 패키지 로드하기

```
REPO_URL <- "https://cran.seoul.go.kr/"
if (!require(httr)) install.packages("httr", repos=REPO_URL)
if (!require(rjson)) install.packages("rjson", repos=REPO_URL)

library(httr)      # 온라인상의 데이터를 가져오기 위한 통신 기능 패키지
library(rjson)     # JSON 처리 패키지
```

## #02. API 접속하기

### 1) 연동에 필요한 조건값들 준비하기

```
# 카카오에서 발급받은 키
kakao_api_key <- "발급받은APIKey"

# 검색조건
keyword = "빅데이터"
page = 1
size = 80
```

### 2) 연동을 위한 인증정보와 URL 구성

카카오에서 요구하는 인증키

```
auth_key = sprintf('KakaoAK %s', kakao_api_key)
auth_key
```

▶ 출력결과

```
'KakaoAK 발급받은APIKey'
```

연동 URL

한글 데이터는 URL에 포함시킬 수 없기 때문에 URLEncoding 이라는 과정을 거쳐야 한다.

```
# 검색어에 대한 URL Encoding
query = URLEncode(iconv(keyword, to="UTF-8"))

# 카카오 API 연동 주소
kakao_api_url <- "https://dapi.kakao.com/v2/search/image?query=%s&page=%d&size=%d"
api_url = sprintf(kakao_api_url, query, page, size)
api_url
```

▶ 출력결과

```
'https://dapi.kakao.com/v2/search/image?query=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&page=1&size=80'
```

### 3) API로부터 검색결과 받아오기

`add_headers()` 함수를 사용하여 HEADER 데이터를 이름=값 형식으로 추가할 수 있다.

```
# API 접속
resp <- GET(api_url, add_headers('Authorization'=auth_key))
resp
```

▶ 출력결과

```
Response [https://dapi.kakao.com/v2/search/image?query=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&page=1&size=80]
Date: 2020-01-18 06:59
Status: 200
Content-Type: application/json; charset=UTF-8
Size: 37.2 kB
{"documents":[{"collection":"blog","datetime":"2019-07-15T21:12:00.000+09:00"...}
```

## #03. 검색결과에서 이미지 수집하기

### 1) 검색 결과 부분만 추출

```
# 검색결과를 리스트로 변환
resp_list <- content(resp, as="parse", encoding="utf-8")
# 리스트에서 검색 결과 부분만 추출
img_url_list <- resp_list$documents
img_url_list
```

▶ 출력결과

```
1. $collection
'blog'

$datetime
'2019-07-15T21:12:00.000+09:00'

$display_sitename
'네이버블로그'

$doc_url
'http://blog.naver.com/rldlsrms17/221586442688'

$height
500
```

**\$image\_url**

'http://postfiles11.naver.net/MjAxOTA3MTVfMTQx/MDAxNTYzMTkyNDc3NTY2.ppaqUuX8R9UFI8Ye8eU4vDGhAfNfH3UbIHSYu6SQ4O4g.3dZfQF0sl90cBQZAXrP1prleMS34-6imrmsudXYijlg.JPEG.rladlsrms17/%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0.JPG?type=w966'

**\$thumbnail\_url**

'https://search3.kakaocdn.net/argon/130x130\_85\_c/DtuD1F87X4E'

**\$width**

729

## 2. \$collection

'blog'

**\$datetime**

'2015-12-01T19:59:00.000+09:00'

**\$display\_sitename**

'네이버블로그'

**\$doc\_url**

'http://blog.naver.com/sbwomen11/220555788773'

**\$height**

266

**\$image\_url**

'http://postfiles12.naver.net/20151201\_155/sbwomen11\_144896725518538wpY\_JPEG/%BA%F2%B5%A5%C0%C%C5%CD6.jpg?type=w2'

**\$thumbnail\_url**

'https://search4.kakaocdn.net/argon/130x130\_85\_c/EKbS2PGTA0n'

**\$width**

550

## 3. ...생략...

## 2) 결과 수 만큼 반복하면서 이미지 다운받기

```
# 몇 번째 처리인지 판단하기 위한 변수
count <- 0

# 검색 결과 수 만큼 반복하면서 매 반복마다 검색결과값 하나를 item으로 가져옴
for (item in img_url_list) {
  # 반복하는 동안 1씩 증가함
  count <- count + 1
  # "검색어_0001.jpg" 형식으로 저장될 파일이름 구성
  fname <- sprintf("%s_%04d.jpg", keyword, count)

  # try({ 명령어 }, silent=FALSE) 안에 포함되는 명령어는 수행도중 에러가 나도 무시하고 다음을 계속 진행한다.
  try({
    # 다운받기
    download.file(url=item$image_url, destfile=fname, mode="wb")

    # 저장된 파일이름과 이미지 주소 확인
    result <- sprintf("%s 저장됨 <- %s", fname, item$image_url)
    print(result)
  }, silent=FALSE)
}
```

### ▶ 출력결과

```
[1] "빅데이터_0001.jpg 저장됨 <- http://postfiles11.naver.net/MjAxOTA3MTVfMTQx/MDAxNTYzMTkyNDc3NTY2.ppaqUuX8R9UFI8Ye8eU4vDGhAfNfH3UbIHSYu6SQ4O4g.3dZfQF0sl90cBQZAXrP1prleMS34-6imrmsudXYijlg.JPEG.rladlsrms17/%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0.JPG?type=w966"
[1] "빅데이터_0002.jpg 저장됨 <- http://postfiles12.naver.net/20151201_155/sbwomen11_144896725518538wpY_JPEG/%BA%F2%B5%A5%C0%C%C5%CD6.jpg?type=w2"
[1] "빅데이터_0003.jpg 저장됨 <- http://t1.daumcdn.net/news/201604/18/akn/20160418130006404rkse.jpg"
```

```
[1] "빅데이터_0004.jpg 저장됨 <- http://postfiles13.naver.net/MjAx0TA5MDZfMjc4/MDAxNTY3NzU00TM3NDUz.DRgrfT6XTZv
[1] "빅데이터_0005.jpg 저장됨 <- http://postfiles15.naver.net/20160316_142/destrot87_1458110069873LdFz1_JPEG/%f
... 생략 ...
```