

02 PJT

Python을 활용한 데이터 수집 2

INDEX

Python을 활용한 데이터 수집 2

- 사전 학습
- Aladin API
- Spotify API

사전 학습

목표

사전학습 목표

- API 이해
- API 가이드 문서 숙지 및 API 키 발급
- requests 라이브러리를 통한 데이터 요청
- 알라딘 API 요청 및 응답 데이터 JSON 형태 출력

API

API

(Application Programming Interface)

두 소프트웨어가 서로 통신할 수 있게 하는 메커니즘

Application

고유한 기능을 가진 모든 소프트웨어

Interface

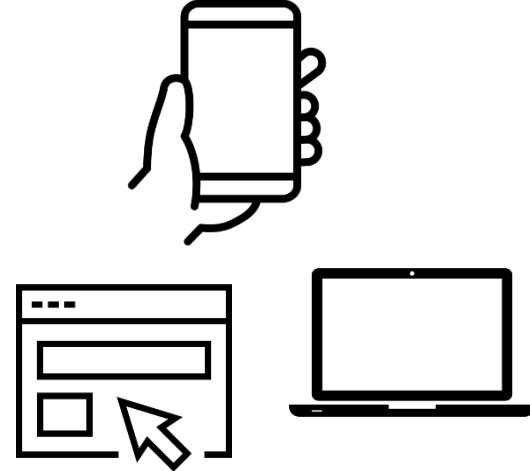
사용자가 기기를 쉽게 동작 시키는 데 도움을 주는 시스템

- 서로 다른 두 개의 시스템 사이에서 정보나 신호를 주고받는 경우의 접점

API 예시 (1/3)



기상청 시스템



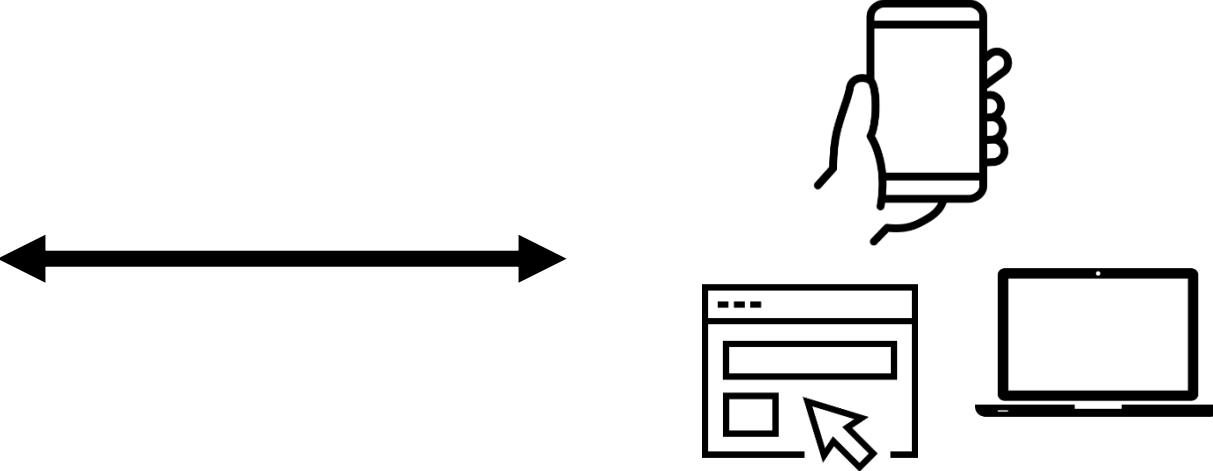
다양한 서비스

- 기상 데이터가 들어있는 기상청의 시스템
- 스마트 폰의 날씨 앱, 웹 사이트의 날씨 정보 등 다양한 서비스들이 이 기상청 시스템으로부터 데이터를 요청해서 받아감

API 예시 (2/3)



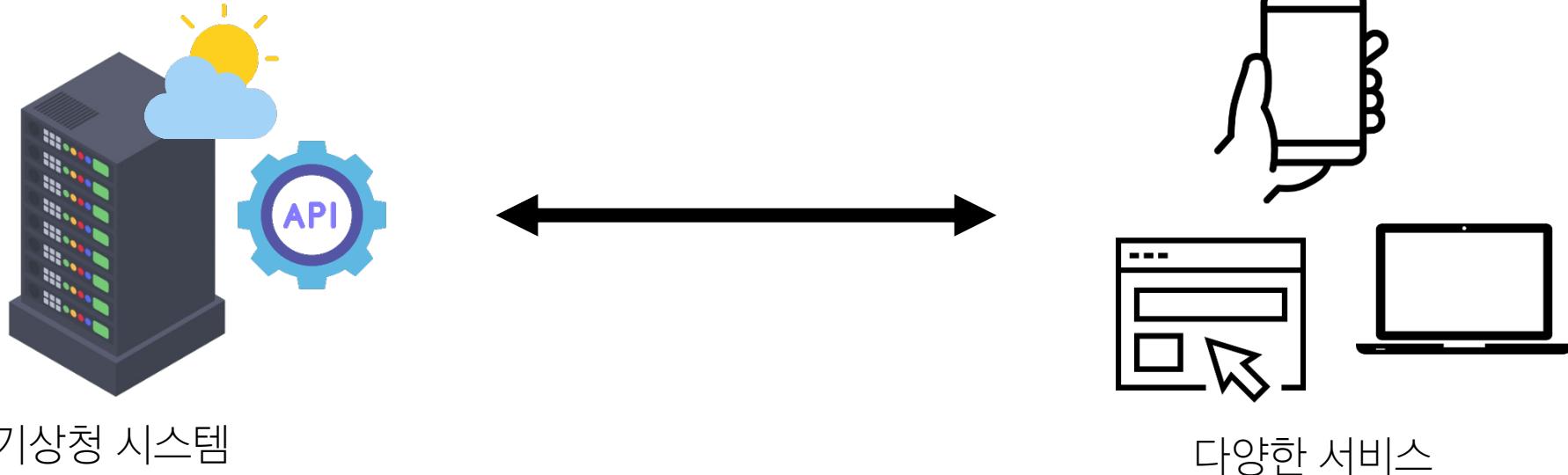
기상청 시스템



다양한 서비스

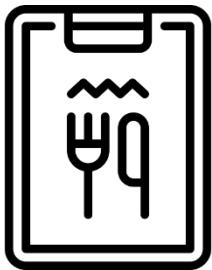
- 날씨 데이터를 얻으려면?
 - 기상청 시스템에는 정보들을 요청하는 지정된 형식이 있음
 - 지역, 날짜, 조회할 내용들(온도, 바람 등)을 제공하는 매뉴얼

API 예시 (3/3)

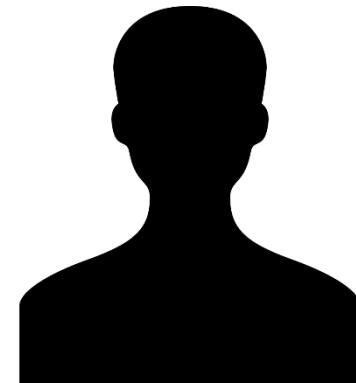
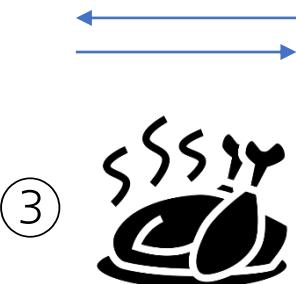


- “이렇게 요청을 보내면, 이렇게 정보를 제공 해줄 것이다”라는 매뉴얼
 - 소프트웨어와 소프트웨어 간 지정된 정의(형식)으로 소통하는 수단 → API
- 스마트폰의 날씨 앱은 기상청에서 제공하는 API를 통해 기상청 시스템과 대화하여 매일 최신 날씨 정보를 표시 할 수 있음

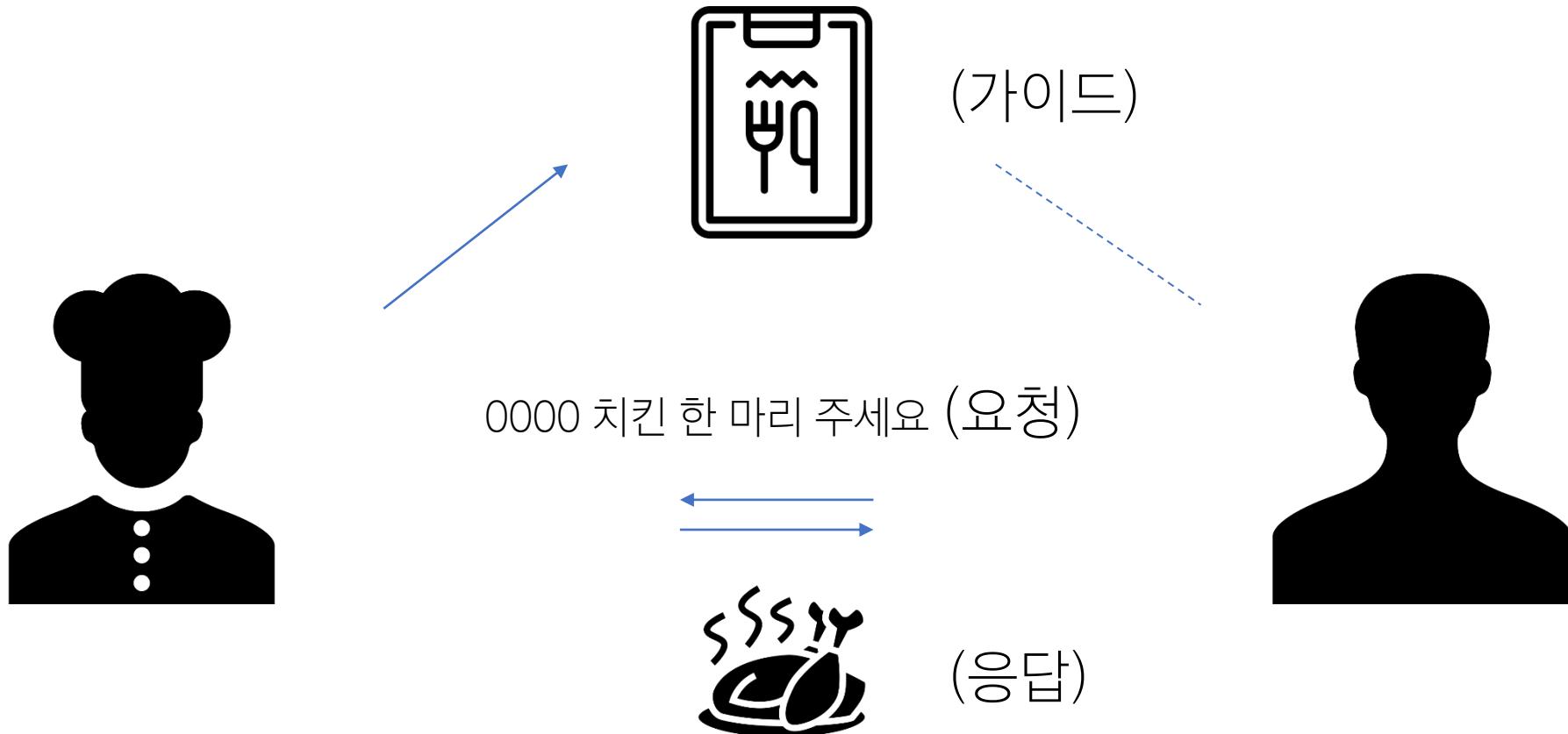
① 이런 메뉴 있어요



② 0000 치킨 한 마리 주세요



③



요리사가 만든 메뉴판(주문 가이드)에 맞춰
고객은 음식을 주문(**요청**)하고,
요리사는 주문 받은 음식을 전달(**응답**)한다.

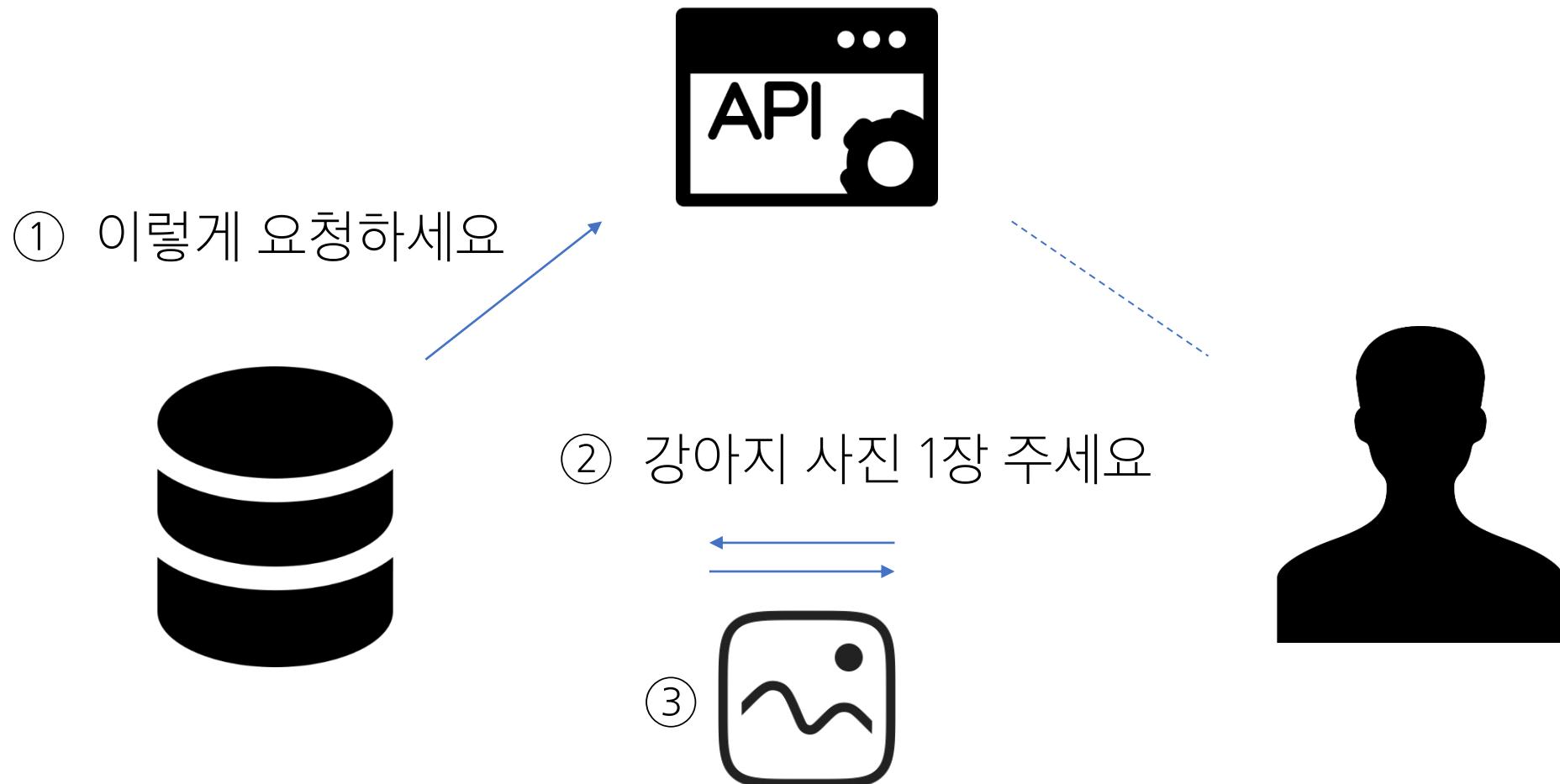
API

(Application Programming Interface)

두 소프트웨어가 서로 통신할 수 있게 하는 메커니즘

- 가이드에 맞춰 요청하고 응답하는 통신 소프트웨어

01_example.py





Dog API

Documentation
Breed list
About
Submit your dog

🐾 Dog CEO Zine
\$ Buy me a dog treat
🔗 View on GitHub
🐦 Follow on Twitter

The internet's biggest collection of **open source dog pictures.**

Read our [documentation](#) to find out more or try it out for yourself below.

Want to add your dog to the collection? Submit your photos as a [pull request here](#).

Need more dog in your life? Get issue 1 of [Dog CEO Zine](#) - a quarterly business and lifestyle magazine for the modern dog. Featuring an exclusive interview with Scottie the Monopoly dog. [Order your copy](#) from Side Orders Publishing. Ships worldwide.

<https://dog.ceo/api/breeds/image/random> Fetch!

JSON

```
{  
    "message": "https://images.dog.ceo/breeds/boxer/n02108089_35.  
    \"status\": \"success\"  
}
```

IMAGE



강아지 사진을 요청하면 응답해주는 Dog API
<https://dog.ceo/dog-api/>

무작위 강아지 사진 한 장 필요할 땐 이렇게 요청하세요.

The screenshot shows the Dog API's main interface. On the left, there's a sidebar with links like 'Documentation', 'Breeds list', 'About', 'Submit your dog', and social media links. The main area has a heading 'ENDPOINTS' with links: 'List all breeds', 'Random image', 'By breed', 'By sub-breed', and 'Browse breed list'. Below this is a section titled 'DISPLAY SINGLE RANDOM IMAGE FROM ALL DOGS COLLECTION' with a red dashed border. It contains a button labeled 'https://dog.ceo/api/breeds/image/random' and a 'Fetch!' button. To the right is a JSON response box and an 'IMAGE' box containing a small image of a dog's face. At the bottom, there's a 'JSON' box with the following JSON content:

```
{ "message": "https://images.dog.ceo/breeds/buhund-norwegian/hund-1024x1024.jpg", "status": "success" }
```

무작위 강아지 사진 여러 장 필요할 땐 이렇게 요청하세요.

The screenshot shows the Dog API's main interface. The layout is similar to the first one, with a sidebar and an 'ENDPOINTS' section. Below it is a section titled 'DISPLAY MULTIPLE RANDOM IMAGES FROM ALL DOGS COLLECTION' with a red dashed border. It contains a button labeled 'https://dog.ceo/api/breeds/image/random/3' and a 'Fetch!' button. To the right is a JSON response box and two 'IMAGE' boxes, each showing a different dog. At the bottom, there's a 'JSON' box with the following JSON content:

```
{ "message": [ "https://images.dog.ceo/breeds/terrier-sealyham/n02095881.jpg", "https://images.dog.ceo/breeds/springer-english/n02102041.jpg", "https://images.dog.ceo/breeds/hound-english/n02089973_2.jpg" ], "status": "success" }
```

적절한 요청 방법과 응답 결과를 알려주는 가이드 문서

```
# 01_example.py

import requests

URL = 'https://dog.ceo/api/breeds/image/random'

response = requests.get(URL).json()
print(response)

results = response.get('message')
print(results)
```

무작위 강아지 사진 한 장 요청

```
{'message': 'https://images.dog.ceo/breeds/spaniel-brittany/n02101388_10373.jpg',
'status': 'success'}
'https://images.dog.ceo/breeds/spaniel-brittany/n02101388_10373.jpg'
```

API는 message에 사진 경로 데이터를 담아 응답

02_example.py

알라딘 Open API 매뉴얼 ☆ ⓘ ☁

파일 수정 보기 도구 도움말

☰

1. 요청 (Request)

- 1) 상품 검색 API
- 2) 상품 리스트 API
- 3) 상품 조회 API
- 4) 중고상품 보유 매장 검색 API

2. 응답 (Response)

- 1) 상품 검색/상품 리스트/상품 조회 API
- 2) 상품 조회 API : 부기정보
- 3) 중고상품 보유 매장 검색 API

* 최종업데이트 : 2022-07-13

1. 요청 (Request)

1) 상품 검색 API

- 요청 방법

- 요청 URL : <http://www.aladin.co.kr/ttb/api/ItemSearch.aspx>
- 요청 URL샘플 :
[http://www.aladin.co.kr/ttb/api/ItemSearch.aspx?ttbkey=\[TTBKey\]&Query=aladdin&QueryType=Title&MaxResults=10&start=1&SearchTarget=Book&output=xml&Version=20131101](http://www.aladin.co.kr/ttb/api/ItemSearch.aspx?ttbkey=[TTBKey]&Query=aladdin&QueryType=Title&MaxResults=10&start=1&SearchTarget=Book&output=xml&Version=20131101)
- 결과샘플
 - XML형식 : http://www.aladin.co.kr/ttb/api/test/ItemSearch_20131101.xml
 - javascript형식 : http://www.aladin.co.kr/ttb/api/test/ItemSearch_20131101.js
- 한 페이지에 최대 50개, 총 결과는 200개까지만 검색 가능

- 상품 검색 API 요청(Request) 파라미터

구분	요청변수	변수종류	설명
필수	TTBKey	문자열	부여받은 TTBKey값
	Query	문자열	검색어
옵션 (옵션 조정 변수가 없을 경우에는 기본값으 로 검색)	QueryType	Keyword (기본값) : 제목+저자 Title : 제목검색 Author : 저자검색 Publisher : 출판사검색	검색어 종류
	SearchTarget	Book(기본값) : 도서 Foreign : 외국도서 Music : 음반 DVD : DVD Used : 중고샵(도서/음반/DVD 등) eBook : 전자책 All : 위의 모든 타겟(물)	검색 대상 Mall
	Start	1이상, 양의 정수(기본값:1)	검색결과 시작페이지
MaxResults	1이상 1000이하, 양의 정수(기본값:10)	검색결과 한 페이지당 최대 출력 개수	

도서 관련 정보를 제공하는 알라딘 API와 가이드 문서

https://docs.google.com/document/d/1mX-WxuoGs8Hy-QalhHcvuV17n50uGl2Sg_GHofqiePE/edit

알라딘을 비롯한 많은 서비스는 클라이언트에게
고유의 키를 발급해, 들어온 요청이 누구의
요청인지 확인(인증)한다.

**따라서 나를 인증 할 수 있는 (일종의 주민등록번호)
키가 외부에 유출되지 않도록 잘 관리하여야 한다.**

사전 준비

1. 공용 노션 "Aladin API Key 발급 가이드" 문서 참고

| 요청 방법 1 (1/2)

- 무슨 자료를 요청할지 (endpoint)

<http://www.aladin.co.kr/ttb/api/ItemList.aspx>
?ttbkey={key}&QueryType={query_type}

요청 방법 1 (2/2)

- 자료 요청에 필요한 정보 (query)
 - endpoint와 query는 ‘?’로 연결 및 구분
 - 각각의 query는 ‘&’으로 연결

`http://www.aladin.co.kr/ttb/api/ItemList.aspx
?ttbkey={key}&QueryType={query_type}`

요청 방법 2

```
# 02_example.py

URL = 'http://www.aladin.co.kr/ttb/api/ItemSearch.aspx'

params = {
    'ttbkey': '부여받은 TTBKey',
    'Query': '파울로 코엘료',
    'QueryType': 'Author',
    'MaxResults' : 20,
    'start' : 1,
    'SearchTarget' : 'Book',
    'output' : 'js',
    'Version' : '20131101'
}

response = requests.get(URL, params=params)
response = response.json()
```

무슨 자료를 요청할지 (endpoint)

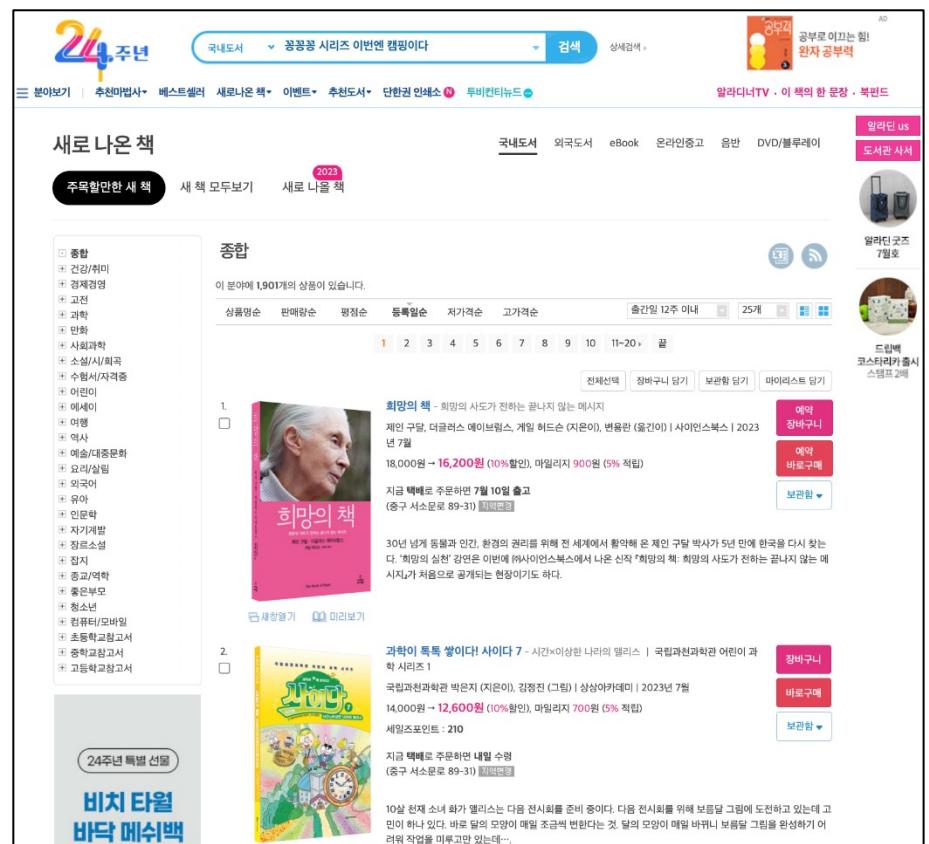
요청에 필요한 정보를 params에 담는다.

params를 포함하여 URL에 get 방식으로 요청 후 json 형태로 변환

응답 데이터

- 응답 받은 결과 중, 'item'에 도서 정보를 담은 리스트 응답
해당 정보는 알라딘 사이트에서도 확인 가능

```
{
  "version": "20131101",
  "logo": "http://image.aladin.co.kr/img/header/2011/aladin_logo_new.gif",
  "title": "알라딘 주체별 면한 신간 리스트 - 국내도서",
  "link": "https://www.aladin.co.kr/shop/common/wnew.aspx?NewType=SpecialNew&BranchType=1&partner=openAPI",
  "pubDate": "Thu 06 2023 06:49:10 GMT",
  "totalResults": 1901,
  "startIndex": 1,
  "itemsPerPage": 20,
  "query": "QueryType=ITEMNEWSPECIAL;SearchTarget=Book",
  "searchCategoryId": 0,
  "searchCategoryName": "국내도서",
  "item": [
    {
      "title": "희망의 책 - 희망의 사도가 전하는 끝나지 않는 메시지",
      "link": "http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=320162652&partner=openAPI&start=api",
      "author": "제인 구달, 더글러스 에이브럼스, 개일 하드슨 (지은이), 변용란 (옮긴이)",
      "isbn": "K662834574",
      "isbn13": "9791192908076",
      "itemId": 320162652,
      "priceSales": 16200,
      "priceStandard": 18000,
      "mallType": "BOOK",
      "stockStatus": "예약판매",
      "mileage": 980,
      "cover": "https://image.aladin.co.kr/product/32016/26/coversum/k662834574_1.jpg",
      "categoryId": 51209,
      "categoryName": "국내도서>과학>생물과학>생태학",
      "publisher": "사이언스북스",
      "salesPoint": 0,
      "adult": false,
      "fixedPrice": true,
      "customerReviewRank": 0,
      "subInfo": {}
    },
    {
      "title": "과학이 톡톡 쌉이다! 사이다 7 - 시간x이상한 나라의 앤리스",
      "link": "http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=320150463&partner=openAPI&start=api",
      "author": "국립과천과학관 박은지 (지은이), 김정진 (그림)",
      "isbn": "K992834571",
      "isbn13": "9791185402819",
      "itemId": 320150463,
      "priceSales": 14000원 ~ 12,600원 (10% 할인), 마일리지 700원 (5% 적립)
    }
  ],
  "description": "30년 넘게 동물과 인간, 환경의 권리를 위해 전 세계에서 활약해 온 제인 구달 박사가 5년 만에 한국을 다시 찾는다. '희망의 실천' 강연은 이번에 해시아인스북스에서 나온 신작 '희망의 책: 희망의 사도가 전하는 끝나지 않는 메시지'가 처음으로 출판된다. 그녀는 저마다 특별한 이야기를 전하고자 노력하는 사람들을 소개하는 책이다. 책은 제인 구달, 더글러스 에이브럼스, 개일 하드슨 (지은이), 변용란 (옮긴이) | 사이언스북스 | 2023년 7월 | 18,000원 ~ 16,200원 (10% 할인), 마일리지 900원 (5% 적립) | 24주년 특별 선물"
}
10월 현재 소녀 화가 앤리스는 다음 전시회를 준비 중이다. 다음 전시회를 위해 보름을 그림에 도전하고 있는데 고민이 하나 있다. 바로 달의 모양이 매일 조금씩 변한다는 것. 달의 모양이 매일 바뀌니 보름달 그림을 완성하기 어려워 작업을 미루고만 있는데...",
"isbn": "K992834571",
"isbn13": "9791185402819",
```



URL 파라미터 (쿼리 파라미터)

1. URL에 포함된 파라미터로 ‘?’ 이후의 문자열
2. 여기서 파라미터는 [파라미터 Key]=[파라미터 Value]이 한 세트이다.
 - 파라미터 Key는 대소문자를 구분한다.
 - API 사용 시, 공식문서에 명시된 파라미터 Key를 사용해야 한다.
3. 파라미터가 여러 개면 ‘&’로 구분한다.

예시) `https://example.com? fRuiT=orange&gAmE=tic-tac-toe`

```
URL = 'https://example.com'  
params = {  
    'fRuiT': 'orange',  
    'gAmE': 'tic-tac-toe'  
}  
response = requests.get(URL, params=params).json()  
print(response)
```

요청에 대한 응답 예시

```
{'fruit_param': 'I love orange!', 'game': 'tic-tac-toe is my favorite!'}  
{'fruit_param': 'I love orange!', 'game': 'tic-tac-toe is my favorite!'}
```

Aladin API

준비사항

개발도구 및 라이브러리

- 개발도구
 - Visual Studio Code
 - Python 3.9+
- 필수 라이브러리
 - <https://requests.readthedocs.io/en/latest/>
- API
 - [알라딘 API](#) - 도서 정보 및 API 서비스

요구사항

공통 요구사항

- 커뮤니티 서비스 개발을 위한 데이터 구성 단계로, 필요한 도서 데이터를 수집하는 과정입니다.
- 완성된 기능들은 향후 커뮤니티 서비스에서 활용할 수 있습니다.
- API 요청 시 버전은 가장 최신 버전인 20131101을 사용합니다.
- 반드시 제공된 examples/ 폴더의 예시 파일을 먼저 참고합니다.
 - 예시 파일에는 이번 프로젝트 해결을 위해 알아야 하는 혹은 직접적인 도움이 될 수 있는 코드가 작성되어 있습니다.

A. 작가의 작품 조회 (problem_a)

- 작가 ‘파울로 코엘료’를 검색해 응답 받은 작품들 20개의 제목을 리스트로 출력합니다.

1. 풀이

- requests 라이브러리를 사용하여 알라딘 open API에 작가 파울로 코엘료 데이터를 요청합니다. ([Get ItemSearch](#))
- 응답 받은 도서의 제목을 반환하는 함수 author_works를 작성합니다.

A. 작가의 작품 조회 (problem_a)

2. 결과

- problem_a.py 실행 예시

```
[['연금술사', '연금술사 (문학동네 30주년 기념 특별판)', '11분', '다섯번째 산', '브리다', '흐르는 강물처럼', '포르토벨로의 마녀', '마법의 순간 (리커버)', '오자히로', '불륜', '아처', '순례자 - 개정판', '내가 빛나는 순간', '스파이', '일러스트 연금술사', '피에트라 강가에서 나는 울었네', '아크라문서', '알레프', '승자는 혼자다 1', '악마와 미스 프랭']]
```

주의) 알라딘 Get_ItemSearch 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

B. 작가의 특정 조건에 맞는 도서 조회 1(problem_b)

- 작가 ‘파울로 코엘료’를 검색해 응답 받은 도서들 20권 중 회원 리뷰 평점 9점 이상인 도서 목록을 출력합니다 .

1. 데이터

- 알라딘 open API에 작가 파울로 코엘료 데이터를 요청합니다.
[\(Get ItemSearch\)](#)
- 응답 받은 데이터 중 회원 리뷰 평점 (customerReviewRank) 9점 이상인 도서 목록을 반환하는 함수 best_review_books를 작성합니다.

B. 작가의 특정 조건에 맞는 도서 조회 1(problem_b)

2. 결과

- problem_b.py 실행 예시

```
[{'adult': False,
  'author': '파울로 코엘료 (지은이), 최정수 (옮긴이)',
  'categoryId': 50920,
  'categoryName': '국내도서>소설/시/희곡>스페인/중남미소설',
  'cover': 'https://image.aladin.co.kr/product/30/73/coversum/8982814477_3.jpg',
  'customerReviewRank': 9,
  'description': "세상을 두루두루 여행하기 위해 양치기가 된 청년 산티아고의 '자아의 신화' 찾기 여행담. 자칫 딱딱하게 보일 "
                 '수 있는 제목과는 달리 간결하고 경쾌한 언어들로 쓰여 있어서, 물이 흘러가듯 수월하게 읽히는 작품이다.',
  'fixedPrice': True,
  'isbn': '8982814477',
  'isbn13': '9788982814471',
  'itemId': 307361,
  'link': 'http://www.aladin.co.kr/shop/wproduct.aspx?ItemId=307361&partner=openAPI&start=api',
  'mallType': 'BOOK',
  'mileage': 600,
  'priceSales': 10800,
  'priceStandard': 12000,
  'pubDate': '2001-12-01',
  'publisher': '문학동네',
  'salesPoint': 85613,
  'stockStatus': '',
  'subInfo': {},
  'title': '연금술사'},
 {'adult': False,
  'author': '파울로 코엘료 (지은이), 오진영 (옮긴이)',
  'categoryId': 50920,
  'categoryName': '국내도서>소설/시/희곡>스페인/중남미소설',
  'cover': '# 이하 생략'}
```

주의) 알라딘 Get_ItemSearch 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

C. 작가의 특정 조건에 맞는 도서 조회 2 (problem_c)

- 작가 ‘파울로 코엘료’를 검색해 응답 받은 도서 20권 중 판매지수가 높은 순으로 5개 도서의 제목 리스트를 출력합니다.

1. 풀이

- 알라딘 open API에 작가 파울로 코엘료 데이터를 요청합니다.
([Get ItemSearch](#))
- 응답 받은 데이터 중 판매지수(salespoint)를 기준으로 세일즈 포인트가 높은 순서대로 5개 도서 제목을 리스트로 반환하는 함수 `bestseller_book`을 작성합니다.
- [sort 매서드](#) 혹은 [sorted 함수](#)의 특정 파라미터를 이용합니다.

C. 작가의 특정 조건에 맞는 도서 조회 2 (problem_c)

2. 결과

- problem_c.py 실행 예시

```
[['연금술사', '11분', '다섯번째 산', '브리다', '흐르는 강물처럼']]
```

D. 작품을 쓴 작가의 다른 작품 조회(problem_d)

- 제공된 도서 제목('베니스의 상인', '개미', '*')을 검색하여
응답 받은 첫 번째 도서의 저은이가 쓴 다른 작품 5개의 도서 제목을 출력합니다.

1. 풀이

- 알라딘 open API에 제공된 도서 제목을 검색합니다. ([Get ItemSearch](#))
- 응답 받은 데이터 중 첫 번째 도서의 저은이(author)를 찾아,
해당 작가의 다른 작품 5개의 이름을 출력하는 author_other_works를 작성합니다.
- 모든 도서는 고유 값 ISBN을 통해 동일한 도서인지 확인 할 수 있습니다.

D. 작품을 쓴 작가의 다른 작품 조회 (problem_d)

2. 결과

- problem_d.py 실행 예시

```
[ '셰익스피어 5대 희극 세트 - 전 5권 ',  
  '햄릿 ',  
  '셰익스피어 4대 비극 세트 : 햄릿 .오셀로 .맥베스 .리어 왕 - 전 4권 ',  
  '맥베스 ',  
  '로미오와 줄리엣 ']  
['꿀벌의 예언 1', '꿀벌의 예언 2', '[세트] 꿀벌의 예언 1~2 세트', '베르베르 씨, 오늘은 뭘 쓰세요?', '심판']  
None
```

주의) 알라딘 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

E. 알고리즘을 사용한 데이터 출력 (problem_e)

- 제공된 도서 제목('베니스의 상인', '*')을 검색하여 응답 받은 첫 번째 도서의 E-Book 목록(ebookList) 중 종이 서적 판매가(priceSales) 보다 전자 서적의 판매가(priceSales)가 10% 이상 저렴한 전자 도서의 필요 정보를 출력합니다.
- 필요 E-Book 정보
 - itemId, isbn, priceSales, link

E. 알고리즘을 사용한 데이터 출력 (problem_e)

1. 풀이

- 알라딘 open API에 제공된 도서 제목을 검색합니다. ([Get ItemSearch](#))
- 응답 받은 데이터 중 첫 번째 도서의 전자 도서 정보(ebookList)를 찾아, 전자 서적의 판매가(priceSales)가 종이 서적의 판매가 (priceSales) 보다 10% 이상 저렴한 전자 서적의 필요 정보를 dictionary 형태로 출력하는 함수 ebook_list를 작성합니다.

E. 알고리즘을 사용한 데이터 출력 (problem_e)

2. 결과

- problem_e.py 실행 예시

```
[{'isbn': 'E893746262',
 'itemId': 24061717,
 'link': 'https://www.aladin.co.kr/shop/wproduct.aspx?ItemId=24061717&partner=openAPI',
 'priceSales': 6300}]
None
```

주의) 알라딘 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

F. 선택 과제

- 도서 데이터를 제공하는 다른 API를 사용하여 내가 원하는 데이터를 추출하고 만들어봅시다.
- API 목록
 - 국가 자료 종합 목록
(<https://www.nl.go.kr/NL/contents/N31101030400.do>)
 - 네이버 도서검색 API
(<https://developers.naver.com/docs/serviceapi/search/book/book.md>)
 - 카카오 도서검색 API
(<https://developers.kakao.com/docs/latest/ko/daum-search/dev-guide>)

Spotify API

준비사항

개발도구 및 라이브러리

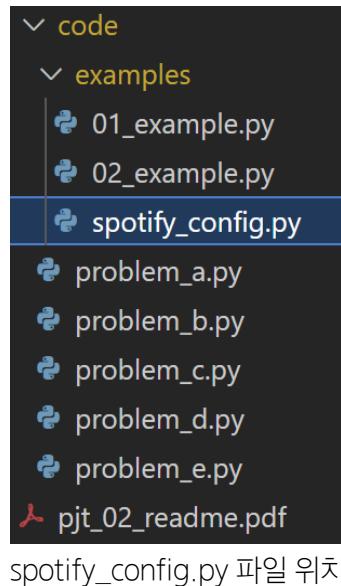
- 개발도구
 - Visual Studio Code
 - Python 3.9+
- 필수 라이브러리
 - <https://requests.readthedocs.io/en/latest/>
- API
 - [Sportify API](#) - 음악 정보 및 API 서비스

사전 준비 (1/2)

1. <https://developer.spotify.com/> 에 접속
2. 공용 노션 "Spotify API Key 발급 가이드" 문서 참고

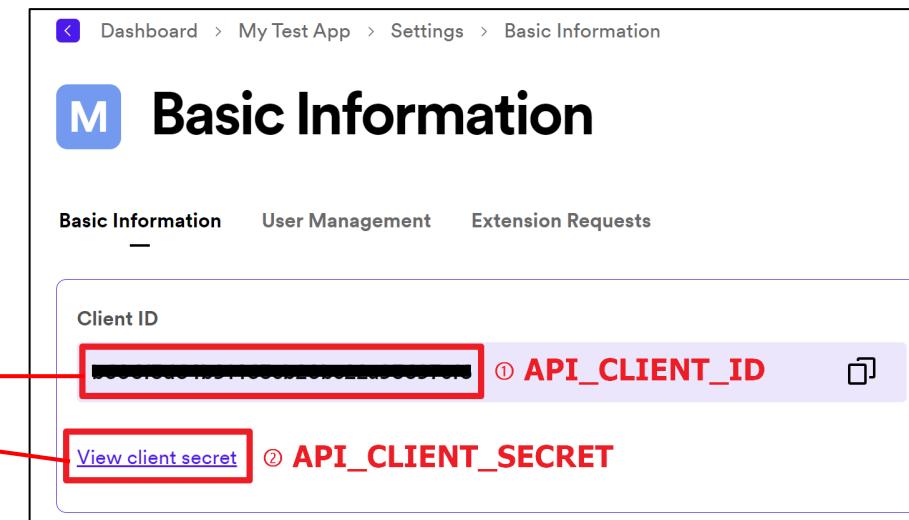
사전 준비 (2/2)

- Client ID와 Client Secret ID를 각각 복사해서 examples폴더 밑의 spotify_config.py에
발급받은 키 값을 입력합니다. 이 때, 발급받은 API Key는 외부에 유출하지 않도록 주의합니다.



```
...  
Spotify에 요청을 보내기 위한 Header파일  
...  
  
import requests  
  
# 🔔 여기에 발급받은 API_CLIENT_ID와 API_CLIENT_SECRET를 입력하세요.  
API_URL = 'https://api.spotify.com/v1'  
API_CLIENT_ID = ''  
API_CLIENT_SECRET = ''  
  
# 아래 코드는 수정하지 않습니다.
```

spotify_config.py



Spotify API 사용 주의사항

1. Spotify API를 통해 실습하기 위해서는 examples 폴더 하위에 있는 spotify_config.py를 먼저 완성해야만 실습이 가능합니다.
2. 또한, API를 요청할 때 모든 Spotify API의 Headers에 spotify_config.py의 getHeaders()가 요청 headers에 필수로 작성되어야 합니다.

```
# search API 문서: https://developer.spotify.com/documentation/web-api/reference/search
URL = 'https://api.spotify.com/v1
```

요구사항

공통 요구사항

- 커뮤니티 서비스 개발을 위한 데이터 구성 단계로, 필요한 음악 데이터를 수집하는 과정입니다.
- 완성된 기능들은 향후 커뮤니티 서비스에서 활용할 수 있습니다.
- API 요청 시 언어 및 지역 설정 데이터는 한국을 기준으로 합니다.
- 반드시 제공된 examples/ 폴더의 예시 파일을 먼저 참고합니다.
 - 예시 파일에는 이번 프로젝트 해결을 위해 알아야 하는 혹은 직접적인 도움이 될 수 있는 코드가 작성되어 있습니다.

A. 음악 정보 조회 (problem_a)

- 장르가 k-pop인 음악트랙 20개의 트랙 제목 목록을 가져옵니다.
- 풀이
 - requests 라이브러리를 사용하여 Spotify에서 장르가 k-pop인 음악 트랙 데이터([search](#)) 20개를 요청합니다.
 - 02_example 코드를 참고합니다.
 - 응답 받은 데이터의 음악 트랙 제목(name) 목록을 반환하는 함수 get_tracks를 작성합니다.

A. 음악 정보 조회 (problem_a)

2. 결과

- problem_a.py 실행 예시

```
['Cupid - Twin Ver.',  
 'Take Two',  
 'Like Crazy',  
 'MONEY',  
 'OMG',  
 'Bite Me',  
 'FLOWER',  
 'UNFORGIVEN (feat. Nile Rodgers)',  
 'Super',  
 'Hype Boy',  
 'The Planet',  
 'Dreamers [Music from the FIFA World Cup Qatar 2022 Official Soundtrack]',  
 'Like Crazy (English Version)',  
 'Cupid',  
 'Run BTS',  
 'Eve, Psyche & The Bluebeard's wife',  
 'Tally',  
 'Spicy']
```

주의) Spotify 'search' 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

B. 특정 조건에 맞는 음악 정보 조회 (problem_b)

- 장르가 k-pop인 음악트랙 20개 중 popularity가 90이상인 트랙 제목 목록을 가져옵니다.
1. 풀이
- requests 라이브러리를 사용하여 Spotify에서 장르가 k-pop인 음악 트랙 데이터([search](#)) 20개를 요청합니다.
 - 응답 받은 데이터 중 인기도(popularity)가 90 이상인 음악 트랙 제목(name) 목록을 반환하는 함수 get_popular_tracks를 작성합니다.

B. 특정 조건에 맞는 음악 정보 조회 (problem_b)

2. 결과

- problem_b.py 실행 예시

```
[ 'Cupid - Twin Ver.', 'Take Two', 'Like Crazy', 'MONEY', 'OMG', 'Like Crazy' ]
```

주의) Spotify 'search' 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

C. 특정 조건에 맞는 음악 정보 조회 2 (problem_c)

- 장르가 k-pop인 음악트랙 20개 중 앨범 발매일 최신 순으로 5개의 데이터 목록을 출력합니다.

1. 풀이

- requests 라이브러리를 사용하여 Spotify에서 장르가 k-pop인 음악 트랙 데이터([search](#)) 20개를 요청합니다.
- 응답 받은 데이터 중 앨범(album)의 발매일(release_date)를 기준으로 최신 순서로 음악 앨범 발매일 5개를 목록으로 반환하는 함수 ranking를 작성합니다.
- [sort 메서드](#) 혹은 [sorted 함수](#)의 특정 파라미터를 이용합니다.

C. 특정 조건에 맞는 음악 정보 조회 2 (problem_c)

2. 결과

- problem_c.py 실행 예시

```
[ '2023-06-09', '2023-05-22', '2023-05-12', '2023-05-08', '2023-05-01' ]
```

주의) Spotify 'search' 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

D. 연관 아티스트 목록 조회 (problem_d)

- 제공된 아티스트(NewJeans, OldShirts)를 통해 연관 아티스트 목록을 출력합니다.

1. 풀이

- 제공된 아티스트 이름으로 아티스트 정보를 검색([search](#)) 합니다.
- 응답 받은 결과 중 아티스트의 id 값을 찾아 연관 아티스트 목록 ([Get related-artists](#))을 가져옵니다.
- 연관 아티스트 목록 중 아티스트의 이름만 리스트로 담아 출력하는 함수 `get_related_artists`를 작성합니다.

D. 연관 아티스트 목록 조회 (problem_d)

2. 결과

- 검색한 아티스트 정보가 없다면 None을 반환합니다.
 - problem_d.py 실행 예시

```
[ 'STAYC',
  'NMIXX',
  'LE SSERAFIM',
  'tripleS',
  'JEON SOMI',
  '(G)I-DLE',
  'GFRIEND',
  'OH MY GIRL',
  'CHUNG HA']
None
```

주의) Spotify '[Get related-artists](#)' 응답 데이터는 시기에 따라 예시 출력과 다를 수 있습니다.

E. 추천 음악 목록 조회 (problem_e)

- 제공된 트랙 제목(HypeBoy), 아티스트(BTS), 장르(acoustic)를 통해 추천 음악 목록을 출력합니다.

1. 풀이

- 제공된 트랙 제목과 아티스트 이름으로 정보를 검색([search](#)) 합니다.
- 응답 받은 결과 중 트랙 id값과 아티스트 id 값을 찾아 주어진 장르와 함께 음악 추천 목록([Get Recommendations](#))을 가져옵니다.
- 추천 목록에서 트랙 이름을 리스트로 담아 출력하는 함수 recommendation을 작성합니다.

E. 추천 음악 목록 조회 (problem_e)

2. 결과

- problem_e.py 실행 예시

```
[ 'Best Of Me', 'A Drop in the Ocean', 'We Are', 'Dear Mr. President', 'Hurt' ]
```

주의) Spotify '[Get Recommendations](#)' 응답 데이터는 요청마다 예시 출력과 다를 수 있습니다.

F. 선택 과제

- 음악 데이터를 제공하는 다른 API를 사용하여 내가 원하는 데이터를 추출하고 만들어봅니다.
- API 목록
 - SoundCloud API
(<https://developers.soundcloud.com/docs/api/guide>)
 - Apple Music API
(<https://developer.apple.com/documentation/applemusicapi>)

제출

제출 시 주의사항

- 제출기한은 금일 18시까지입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점 등을 상세히 기록하여 제출합니다.
 - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/>에 프로젝트를 생성하고 제출합니다.
 - 프로젝트 이름은 ‘프로젝트 번호 + pjt’로 지정합니다 (ex. **02-pjt**)
 - 반드시 각 반 담당 교수님을 Maintainer로 설정해야 합니다.