

01 PJT

관통 프로젝트 소개

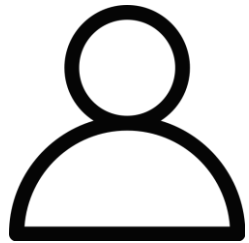
| 관통 프로젝트란?

- 강의 시간에 배운 내용을 모두 포함(관통)
- 추가적인 내용 학습
- 프로젝트 도전!

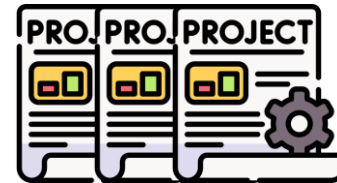
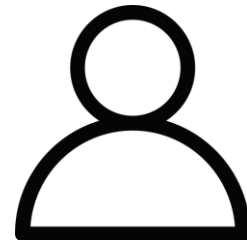


| 관통 프로젝트를 열심히 해야 하는 이유

- 어느 지원자가 더 설득력 있나요 ?



저는
이것도 할 수 있고
저것도 할 수 있고...



제가 진행한
프로젝트입니다.
저는 어떤 역할을 맡
았으며...

| 진행 방식

1. 프로젝트 목표 소개
2. 이론 설명
3. 강사님들과 함께 실습 진행
4. 도전 과제 진행

| 도전 과제는 두 개의 버전이 존재

- 버전 1: 금융 상품 비교 앱
 - PJT01 --> PJT02 --> PJT03 --> PJT04 --> PJT05 --> ... > PJT09 --> PJT10
- 버전 2: 영화 추천 서비스
 - PJT01 --> PJT02 --> PJT03 --> PJT04 --> PJT05 → ... > PJT09 --> PJT10
- * 여러분들은 선택할 수 기회가 있는 행운의 기수 *

버전1(금융 상품 비교 앱) 최종 프로젝트 소개

- SSAFY 10기부터 적용되는 신규 PJT

마이뱅크

보험환전환율비교예금비교

금융파트너

정기예금

검색조건을 입력해보세요

세율1

일반과세(15.4%)

이자지급방식

전체

가입방식

전체

예치금액21,000만원

10,000,000원

정기예금 | 정기적금 | 자유적금

전체

신협

MG 새마을금고

농/축협

저축은행

은행

순위	6개월	12개월	세전이자	세후이자	24개월	36개월	금융기관	상품	UPDATE
1	-	5.02	502,000원	424,692원	-	-	MG 연회	MG더벙킹정기예금	07.13
2	-	5.00	500,000원	423,000원	-	-	MG 안양만안	정기예탁금	07.10
3	-	5.00	500,000원	423,000원	-	-	MG 강북	MG더벙킹정기예금	07.12
4	3.50	4.90	490,000원	414,540원	-	-	MG 복현	정기예탁금	07.10
5	-	4.90	490,000원	414,540원	-	-	MG 금호	MG더벙킹정기예금	07.13

마이뱅크

test | 로그아웃

Bankbook

예금비교 환율계산기 은행 지도 게시판

test 님의 프로필 페이지

기본 정보 수정 | 포트폴리오 수정 | 상품 추천 받기

상품 추천 받기

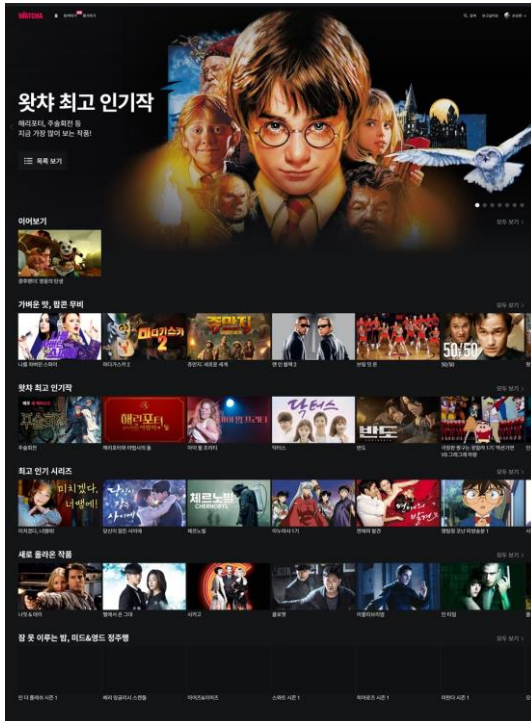
나와 나이, 현재 가진 금액, 연봉이 비슷한 사람들이 가입한 상품 리스트

- 1 : (정기예금)제주은행 - 제주Dream 정기예금 (개인/만기 지급식)
- 2 : (정기예금)광주은행 - 영문백스예금
- 3 : (정기예금)중소기업은행 - i-ONE놀이터예금
- 4 : (정기예금)부산은행 - 더(The) 특한 정기예금
- 5 : (정기예금)대구은행 - DGB행복파트너예금(일반형)
- 6 : (정기예금)수협은행 - Sh행생주거채우대예금 (만기일시지급식)
- 7 : (정기예금)제주은행 - J정기예금 (만기지급식)
- 8 : (정기예금)수협은행 - 헤이(Hey)정기예금
- 9 : (정기예금)국민은행 - KB Star 정기예금
- 10 : (정기예금)중소기업은행 - IBK행생한가축통장(실세금리정기예금)
- 11 : (정기예금)수협은행 - Sh첫만남우대예금
- 12 : (정기예금)우리은행 - WON플러스예금
- 13 : (정기예금)주식회사 카카오뱅크 - 카카오뱅크 정기예금
- 14 : (정기예금)주식회사 케이뱅크 - 코드K 정기예금
- 15 : (정기예금)경남은행 - BNK주거채우대정기예금

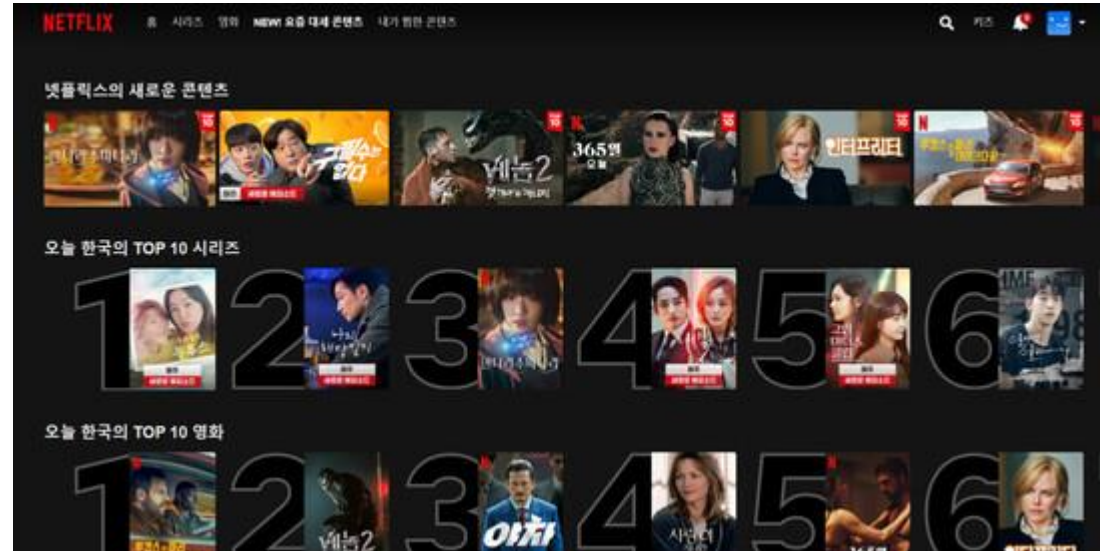
Bankbook

버전2(영화 추천 서비스) 최종 프로젝트 소개

- SSAFY 트랙 초기부터 진행하고 있는 유래 깊은 PJT



왓차



넷플릭스

| 버전 선택

- 버전 1: 금융 상품 비교 앱
 - PJT01 --> PJT02 --> PJT03 --> PJT04 --> **PJT05 --> ... > PJT09 --> PJT10**
- 버전 2: 영화 추천 서비스
 - PJT01 --> PJT02 --> PJT03 --> PJT04 --> **PJT05 --> ... > PJT09 --> PJT10**

(필수) 두 가지 과제 중 하나를 선택하여 진행해야 함

선택은 04 PJT 까지 자유롭게 변경 가능

그 후로는 최종 프로젝트를 진행하기 위한 내용이 포함 -> 변경 불가

PJT 01

이론 및 실습

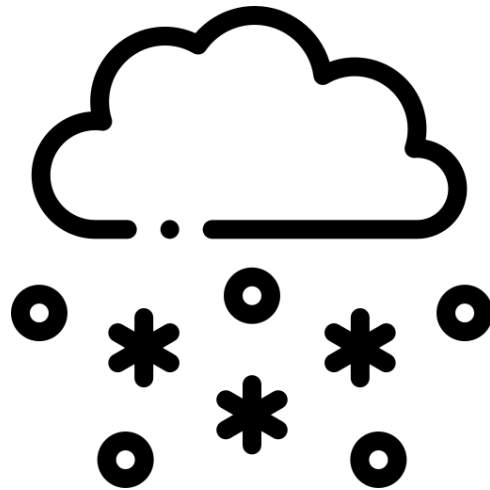
챕터의 포인트

- 목표
- API 이해하기
- 날씨 데이터 수집

실습 목표

| 실습 목표

- 파이썬으로, 인터넷에 있는 날씨 정보를 가져와, 내가 원하는 정보만 출력



| 날씨 정보

- 실습 프로젝트를 진행하기 위해선 날씨 데이터가 있어야 합니다.
- 그러나... 직접 데이터를 모으기엔 너무 어렵습니다!
- 간단하게, 인터넷에 있는 데이터를 가져오면 됩니다.
 - 데이터를 가져오는 방법을 이해하기 위해서 반드시 알아야 할 전문용어들이 있습니다.

| 전문용어 이해하기

- 먼저, 서버와 클라이언트 입니다.
- 서버: 부탁을 받으면 처리해주거나, 부탁대로 원하는 값을 돌려주는 역할을 합니다.
- 클라이언트: 부탁하는 역할입니다.

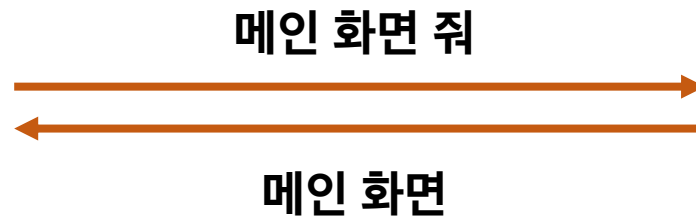


| 전문용어 이해하기

- 우리가 네이버 홈페이지에 접속하는 건 다음과 같이 표현할 수 있습니다.
 - 네이버 주소를 입력하면 익히 알고 있는 네이버 메인 화면을 달라고 요청합니다.
 - 서버는 클라이언트가 요청한 네이버 메인 화면을 돌려줍니다.



클라이언트



NAVER

네이버 서버

| 전문용어 이해하기

- 이번 프로젝트에서는 날씨 정보가 필요합니다.
- 날씨 정보를 가지고 있는 서버가 있습니다.
- 해당 서버에 날씨 정보를 달라고 요청하면 됩니다.



클라이언트



날씨 정보를
가지고 있는 서버

| 전문용어 이해하기

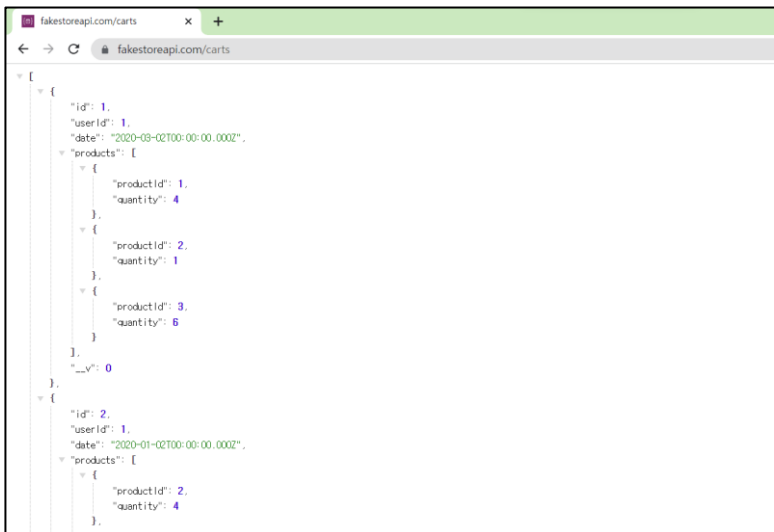
- 정리하면 다음과 같습니다.
 - 클라이언트가 정보를 달라고 요청한다.
 - 서버는 클라이언트의 요청에 따라 원하는 정보를 돌려준다.
- 클라이언트는 어떻게 요청을 보낼 수 있을까요 ?

| 클라이언트가 서버에 요청하는 두 가지 방법

1. 웹 브라우저(크롬)을 켜서 주소창에 주소(URL)를 입력한다.
2. 서버에 정보를 요청하는 파이썬 코드를 작성한다.

1. 웹 브라우저(크롬)을 켜서 URL 에 접속

- 크롬을 켜서 주소창에 아래 URL을 입력해보세요.
- <https://fakestoreapi.com/carts>
- 예시



| 2. 서버에 정보를 달라는 파이썬 코드를 작성

- **vscode를 켜고, 터미널 창을 열어주세요**
 - 터미널 창 여는 방법
 - 상단 Terminal 메뉴 + New Terminal 클릭
 - 단축키: Ctrl + ` (백틱)
- **아래 명령어를 실행하여 필요한 도구를 설치합니다.**
 - requests: 파이썬에서 서버에 요청을 보낼 수 있는 도구

```
$ pip install requests
```

| 2. 서버에 정보를 달라는 파이썬 코드를 작성

- test.py 파일을 만들고, 아래처럼 파이썬 코드를 작성하고 실행해 보세요.

```
import requests

url = 'https://fakestoreapi.com/carts'
data = requests.get(url).json()
print(data)
```


- 결과 예시

```
[{'id': 1, 'userId': 1, 'date': '2020-03-02T00:00:00.000Z', 'products': [{'productId': 1, 'quantity': 4}, {'productId': 2, 'quantity': 1}, {'productId': 3, 'quantity': 6}], '__v': 0}, {'id': 2, 'userId': 1, 'date': '2020-01-02T00:00:00.000Z', 'products': [{'productId': 2, 'quantity': 4}, {'productId': 1, 'quantity': 10}, {'productId': 5, 'quantity': 2}], '__v': 0}, {'id': 3, 'userId': 2, 'date': '2020-03-01T00:00:00.000Z', 'products': [{'productId': 1, 'quantity': 2}, {'productId': 9, 'quantity': 1}], '__v': 0}, {'id': 4, 'userId': 3, 'date': '2020-01-01T00:00:00.000Z', 'products': [{'productId': 1, 'quantity': 4}], '__v': 0}, {'id': 5, 'userId': 3, 'date': '2020-03-01T00:00:00.000Z', 'products': [{'productId': 7, 'quantity': 1}, {'productId': 8, 'quantity': 1}], '__v': 0}, {'id': 6, 'userId': 4, 'date': '2020-03-01T00:00:00.000Z', 'products': [{'productId': 10, 'quantity': 2}, {'productId': 12, 'quantity': 3}], '__v': 0}, {'id': 7, 'userId': 8, 'date': '2020-03-01T00:00:00.000Z', 'products': [{'productId': 18, 'quantity': 1}], '__v': 0}]
```

| 파이썬 코드 이해하기

- url
 - 요청을 보내는 서버의 주소
- requests.get(url)
 - 해당 서버(url)에 데이터를 달라고 요청을 보내는 함수
- .json()
 - 내부의 데이터를 JSON(파이썬의 딕셔너리와 비슷함) 형태로 변환해주는 함수
 - JSON 에 대한 자세한 설명은 뒷부분에서 다룰 예정입니다.

```
import requests  
  
url = 'https://fakestoreapi.com/carts'  
data = requests.get(url).json()  
print(data)
```

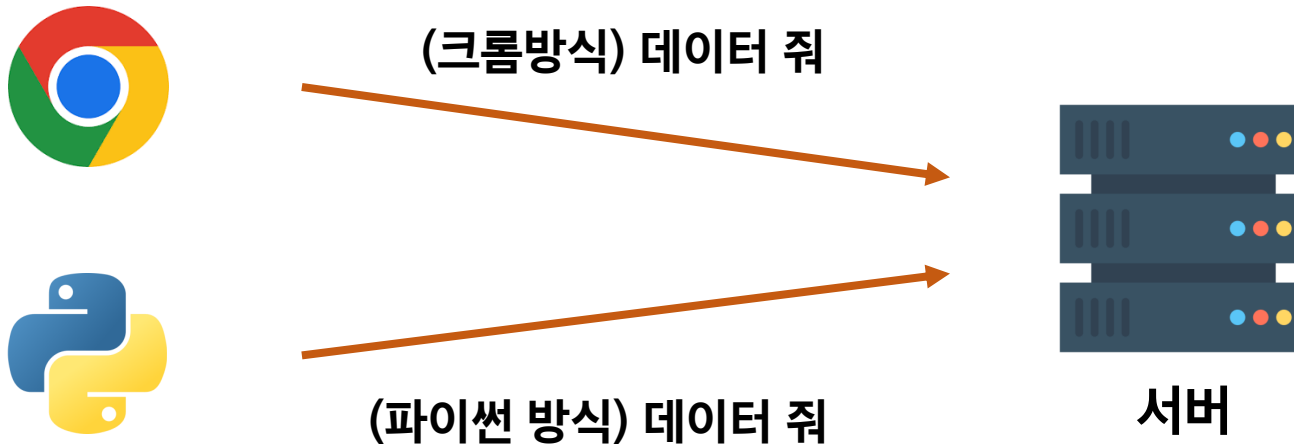


| 서버는 어떻게 요청을 해석할까

- 웹 브라우저(크롬)와 파이썬을 통해 서버에 데이터를 요청하는 방법을 알았습니다.
- 그렇다면, 서버는 어떻게 요청을 이해하고 데이터를 반환할 수 있을까요 ?

| 서버는 어떻게 요청을 해석할까

- 서버에 요청을 보내는 클라이언트는 매우 다양합니다.
- 클라이언트들은 각자 다른 방식으로 요청을 보낼 것입니다.
- 서버가 어떻게 모두 해석할 수 있을까?



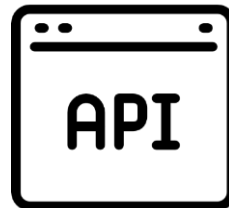
| API

- 클라이언트가 원하는 기능을 수행하기 위해서 서버 측에 만들어 놓은 프로그램
 - 기능 예시: 데이터 저장, 조회, 수정, 삭제 등등
- 서버 측에 특정 주소로 요청이 오면 정해진 기능을 수행하는 API 를 미리 만들어 둡니다.
- 클라이언트는 서버가 미리 만들어 놓은 주소로 요청을 보냅니다.



내 코드

요청
→
←
데이터 전달



데이터 검색
→
←
데이터 조회



서버

| 날씨 정보를 제공해주는 API

- 날씨 정보를 수집하기 위해서는 두 가지를 찾아야 합니다.
 - 날씨 정보를 가지고 있는 서버
 - 해당 서버가 제공하는 API

| 오픈 API

- 외부에서 사용할 수 있도록 무료로 개방(OPEN) 된 API
- 사용법은 공식 문서(Docs)에 명시되어 있습니다.
- 프로젝트에서 사용되는 API
 - [OpenWeatherMap API](#): 기상 데이터 및 날씨 정보를 제공하는 오픈 API
 - [금융상품통합비교공시 API](#): 금융감독원에서 제공하는 금융 상품 정보를 제공하는 오픈 API

| 오픈 API 특징 및 주의사항 (1/2)

- 악성 사용자가 100만 개의 계정을 생성해 API 에 요청을 보내는 상황을 생각해 봅시다.
 - 너무 많은 계정에서 동시에 요청을 보내면 서버가 견디지 못합니다!
- 이러한 문제점을 해결하기 위해 오픈 API 는 **API KEY** 를 활용하여 사용자를 확인합니다.
 - 사용자 인증 혹은 회원가입을 하면 서버에서 API KEY 를 발급해 줍니다.
 - 서버에 요청할 때 마다 해당 API KEY 를 함께 보내 정상적인 사용자인 것을 확인 받습니다.

| 오픈 API 특징 및 주의사항 (2/2)

- API KEY 를 가지고 있는 악성 사용자가 1초에 100만 개의 요청을 보내는 상황을 생각해 봅니다.
 - 서버가 견디지 못하여 정상적인 서비스가 불가능합니다.
- 일부 오픈 API 는 **사용량이 제한**되어 있습니다.
 - 공식 문서의 일일 및 월간 사용량 제한을 반드시 확인하여야 합니다.
 - **[주의]** 사용량이 초과될 경우 요금이 청구될 수 있습니다.

| Quiz.

- 무료로 사용 가능한 API 를 오픈 API 라고 합니다.
- 인증된 사용자만 제공된 API 를 사용할 수 있습니다.
- 클라이언트는 원하는 데이터를 요청 할 때

서버에게 무엇을 함께 보내어 본인이 인증된 사용자인 것을 확인 받을 수 있나요?

| 프로젝트 미션 정리

- 파이썬으로

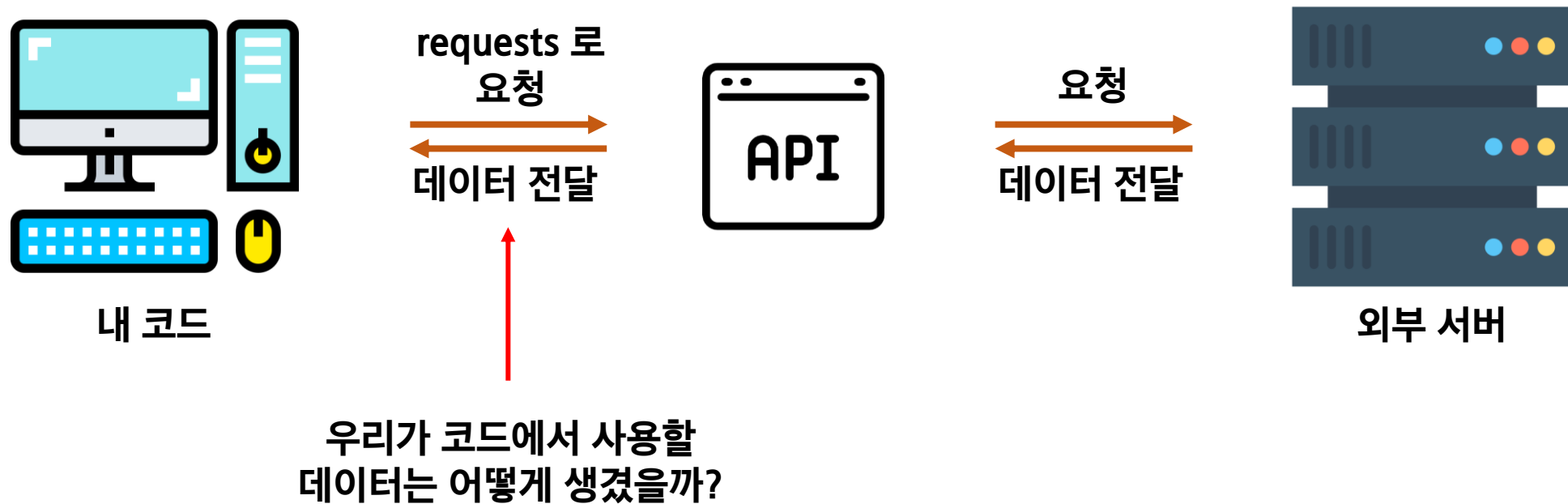
날씨 정보를 제공해주는 서버의 URL 을 이용하여

날씨 정보를 가져옵니다.

날씨 데이터 수집

| JSON

- API 가 반환하는 데이터는 어떻게 생겼을까?



| API 가 사용하는 데이터 형식 - JSON

- JavaScript Object Notation 의 약자. 직역하면 '자바스크립트 객체 표기법'
- 데이터를 저장하거나 전송할 때 많이 사용되는 **경량의 텍스트 기반의 데이터 형식**
- 통신 방법이나 프로그래밍 문법이 아니라 단순히 데이터를 표현하는 표현 방법 중 하나
- 특징
 - 데이터는 중괄호({}) 로 둘러싸인 키-값 쌍의 집합으로 표현됨
 - 키 = 문자열 / 값 = 다양한 데이터 유형을 가질 수 있다
 - 값은 쉼표(,)로 구분됨

- 예시

```
{  
  "name": "김싸피",  
  "age": 28,  
  "city": "서울 캠퍼스",  
  "hobbies": [  
    "공부하기",  
    "복습하기"  
  ],  
  "isStudent": true  
}
```

| JSON - python 예제

- 파이썬은 JSON 을 활용하는 기능을 가지고 있습니다.
- 참고
 - 파싱(Parsing): 데이터를 의미 있는 구조로 분석하고 해석하는 과정
 - `json.loads()`: JSON 형식의 문자열을 파싱하여 python Dictionary 로 변환

```
import json

# JSON 데이터
json_data = '''
{
    "name": "김싸피",
    "age": 28,
    "city": "서울 캠퍼스",
    "hobbies": [
        "공부하기",
        "복습하기"
    ],
    "isStudent": true
}
'''

# JSON 데이터 파싱하기
data = json.loads(json_data)

# JSON 데이터에서 정보 읽기
name = data["name"]
age = data["age"]
city = data["city"]
hobbies = data["hobbies"]
is_student = data["isStudent"]

print(name) # 김싸피
print(age) # 28
print(city) # 서울 캠퍼스
print(hobbies) # ["공부하기", "복습하기"]
print(is_student) # True
```

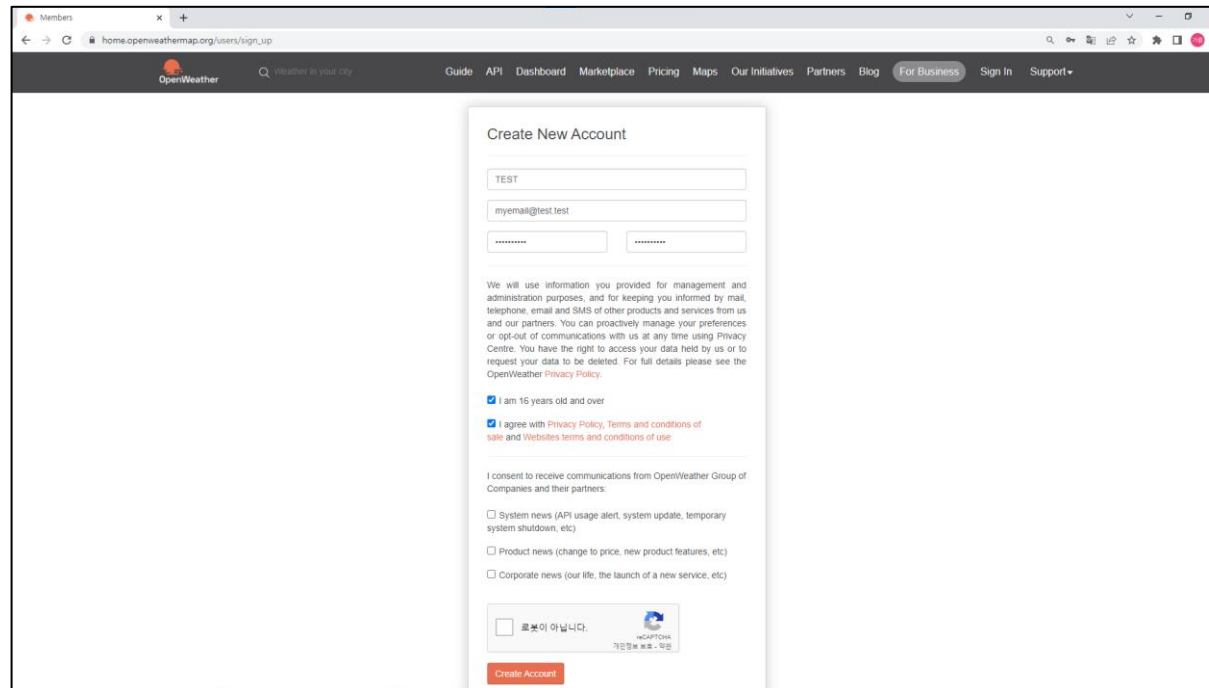
| Openweathermap API

- 기상 데이터 및 날씨 정보를 제공하는 오픈 API
- 전세계 날씨 데이터를 가져와 날씨, 일일 및 시간 별 예보 등 다양한 정보를 얻을 수 있습니다.
- API 사용량 제한
 - 60 calls/minute
 - 1,000,000 calls/month



| Openweathermap API - KEY 발급(1/3)

- 사이트 접속 및 회원가입 진행

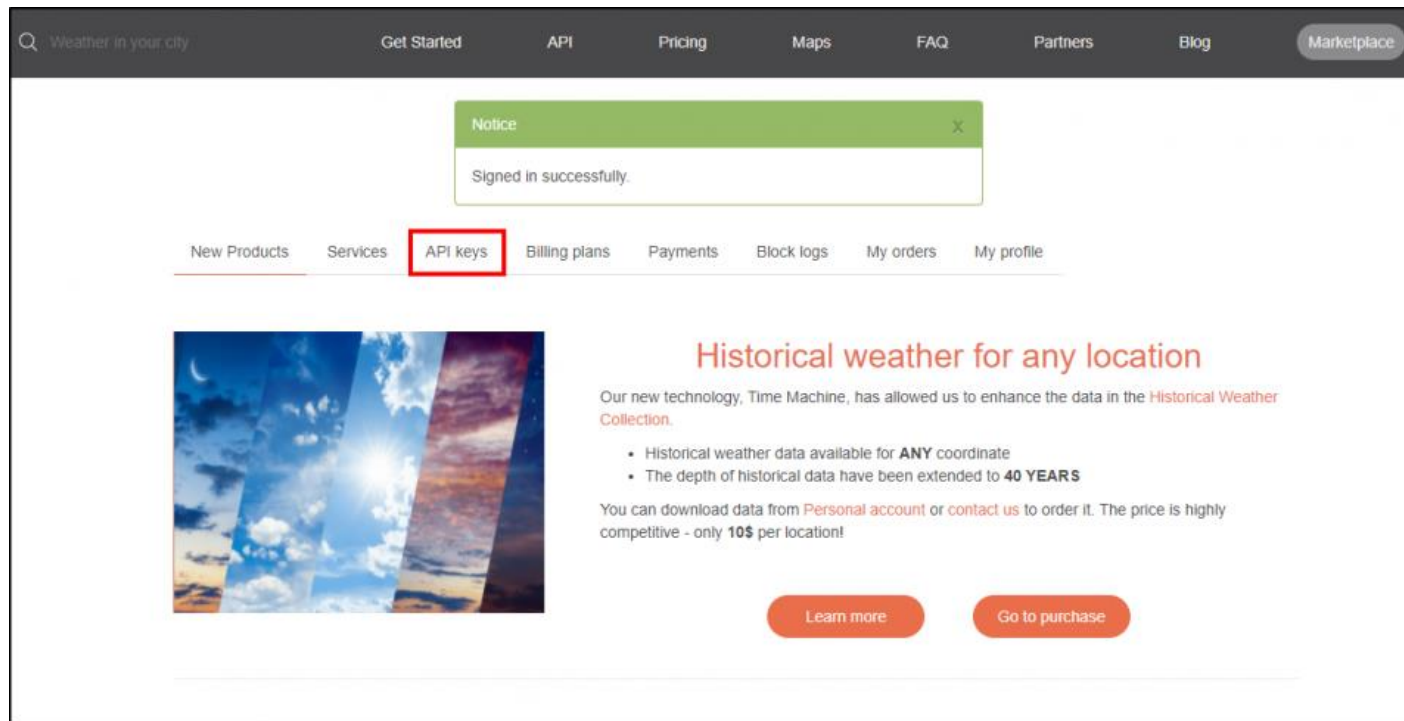


The screenshot shows the 'Create New Account' page on the OpenWeathermap website. The browser address bar shows 'home.openweathermap.org/users/sign_up'. The page has a dark navigation bar with links like Guide, API, Dashboard, Marketplace, Pricing, Maps, Our Initiatives, Partners, Blog, For Business, Sign In, and Support. The main content area is white and contains the following elements:

- Create New Account** title.
- Input fields for Username (containing 'TEST'), Email (containing 'myemail@test.test'), and Password (two fields with masked characters).
- A paragraph of text explaining data usage and privacy policy.
- Two checked checkboxes: 'I am 16 years old and over' and 'I agree with Privacy Policy, Terms and conditions of sale and Websites terms and conditions of use'.
- A section for consent to receive communications from OpenWeather Group of Companies and their partners, with three unchecked checkboxes: 'System news (API usage alert, system update, temporary system shutdown, etc)', 'Product news (change to price, new product features, etc)', and 'Corporate news (our life, the launch of a new service, etc)'.
- A checkbox for '로봇이 아닙니다.' (I am not a robot) with a reCAPTCHA logo.
- A red 'Create Account' button at the bottom.

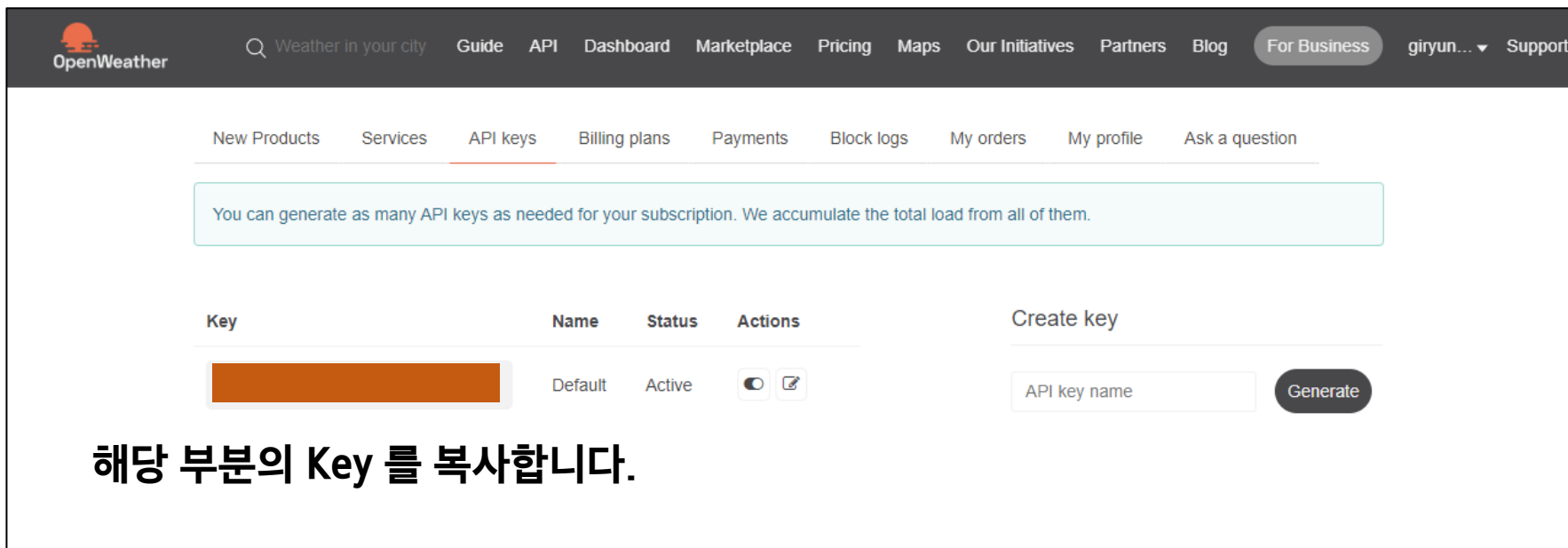
| Openweathermap API - KEY 발급(2/3)

- API Keys 탭으로 이동



| Openweathermap API - KEY 발급(3/3)

- API Key 복사
- API Key 를 복사하여 코드에서 활용합니다.



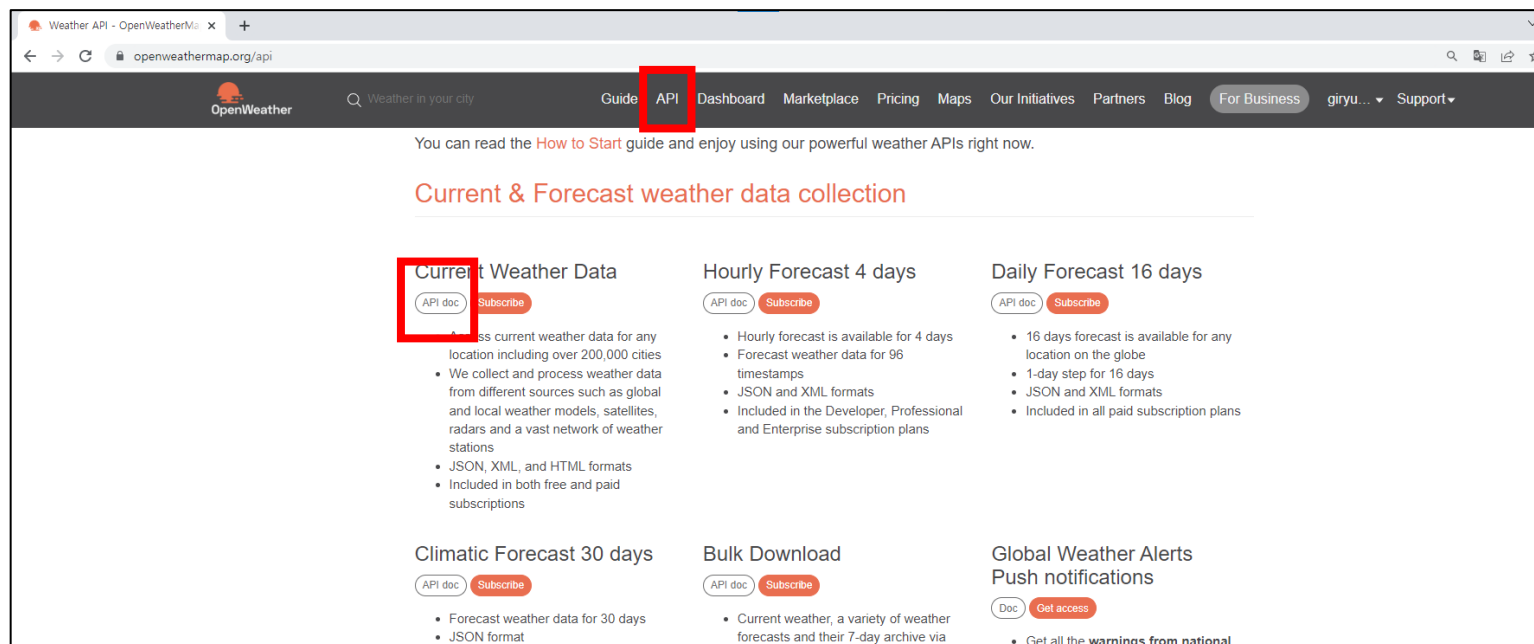
The screenshot shows the OpenWeather API dashboard. The 'API keys' tab is selected. A table displays the following information:

Key	Name	Status	Actions
[Redacted Key]	Default	Active	[Toggle] [Copy]

Below the table, the text '해당 부분의 Key 를 복사합니다.' (Copy the key from this part) is displayed. To the right, the 'Create key' section includes a text input for 'API key name' and a 'Generate' button.

Openweathermap API - 사용법 확인

- 상단 API 탭을 클릭
- 현재 날씨에 대한 데이터를 사용하기 위해 Current Weather Data 의 API doc 클릭



| Openweathermap API - 실습(1/4)

- 특정 지역의 현재 날씨에 대한 모든 정보 출력하기
 - 공식 문서 참고

Call current weather data

How to make an API call

API call

```
https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}
```

- [참고] 서울의 위도: 37.56 / 경도: 126.97

```
# 서울의 위도
lat = 37.56
# 서울의 경도
lon = 126.97

# API 요청 URL
url = f'https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={api_key}'
```

- 출력 결과

- 날씨에 따라 출력 결과는 다를 수 있습니다.

```
{'base': 'stations',
 'clouds': {'all': 0},
 'cod': 200,
 'coord': {'lat': 37.56, 'lon': 126.97},
 'dt': 1684293603,
 'id': 1835848,
 'main': {'feels_like': 300.17,
          'humidity': 39,
          'pressure': 1013,
          'temp': 300.43,
          'temp_max': 301.93,
          'temp_min': 297.96},
 'name': 'Seoul',
 'sys': {'country': 'KR',
         'id': 8105,
         'sunrise': 1684268515,
         'sunset': 1684319716,
         'type': 1},
 'timezone': 32400,
 'visibility': 10000,
 'weather': [{'description': 'clear sky',
               'icon': '01d',
               'id': 800,
               'main': 'Clear'}],
 'wind': {'deg': 200, 'speed': 6.69}}
```

Openweathermap API - 실습(2/4)

- 특정 도시의 현재 날씨를 도시 이름으로 요청하여 모든 정보 출력하기
- 공식 문서 참고
- 출력 결과

Built-in API request by city name
You can call by city name or city name, state code and country code. Please note that searching by states available only for the USA locations.

API call

```
https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
```

JSON

XML

List of condition codes

Min/max temperature in current weather API and forecast API

Other features

Geocoding API

Built-in geocoding

Built-in API request by city name

Built-in API request by city ID

Built-in API request by ZIP code

Format

- 날짜에 따라 출력 결과는 다를 수 있습니다.

```
{'base': 'stations',
'clouds': {'all': 0},
'cod': 200,
'coord': {'lat': 37.56, 'lon': 126.97},
'dt': 1684293603,
'id': 1835848,
'main': {'feels_like': 300.17,
'humidity': 39,
'pressure': 1013,
'temp': 300.43,
'temp_max': 301.93,
'temp_min': 297.96},
'name': 'Seoul',
'sys': {'country': 'KR',
'id': 8105,
'sunrise': 1684268515,
'sunset': 1684319716,
'type': 1},
'timezone': 32400,
'visibility': 10000,
'weather': [{'description': 'clear sky',
'icon': '01d',
'id': 800,
'main': 'Clear'}],
'wind': {'deg': 200, 'speed': 6.69}}
```

- [참고] 도시 이름으로 요청 URL 만들기

```
# 검색 조건
city = "Seoul,KR"

# 다른 지역
# city = "Tokyo,JP"
# city = "New York,US"

# API 요청 URL
url = f'http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}'
```

| Openweathermap API - 실습(3/4)

- 서울의 현재 날씨 중 온도만 출력하기
 - 기본적으로 켈빈(K) 온도를 반환합니다.
 - 섭씨 온도 = (켈빈 온도 - 273.15) 로 계산할 수 있습니다.
 - 출력 결과

```
켈빈 온도: 300.71K  
섭씨 온도: 27.56°C
```

| Openweathermap API - 실습(4/4)

- 서울의 현재 날씨에 대한 설명(description) 데이터만 출력하기

- 출력 결과

```
'날씨 설명: clear sky'
```

- Json 형태의 데이터를 분석하여 원하는 부분만 가져오도록 구성합니다.

```
json_response = requests.get(url).json()
description = json_response['weather'][0]['description']
return f'날씨 설명: {description}'
```

도전 과제

금융 상품 비교 앱 PJT 01

| 버전1 - PJT01 도전 과제

- 프로젝트명: 파이썬을 활용한 API 데이터 수집
- 목표
 - 파이썬으로 정기 예금 데이터 수집 및 미션 수행
- 특징
 - 외부 서버를 활용한 데이터 수집
 - 요구사항에 맞게 JSON 형태 데이터 가공

영화 추천 서비스 PJT 01

| 버전2 - PJT01 도전 과제

- 프로젝트명: 파이썬을 활용한 데이터 수집1
- 목표
 - 파이썬으로 도서 및 아티스트 데이터 가공 및 미션 수행
- 특징
 - 샘플 데이터 제공
 - 요구사항에 맞게 JSON 형태 데이터 가공