

API를 활용한

서비스 개발

날씨 및 급식 알리미

NULL4U 현욱, 박승빈 합동 프로젝트

API를 활용한 서비스 개발

Q 목차

Q API란?

Q REST API란?

Q 개발 목적

Q 개발 과정



01. WHAT IS API?

API 이란?

- API는 소프트웨어 응용 프로그램 간에 효과적으로 통신하기 위한 인터페이스를 제공합니다.
- 다른 프로그램이나 서비스에서 제공하는 기능을 활용하거나 자원에 접근할 수 있도록 돕습니다.
- 정의된 규약에 따라 데이터나 서비스를 요청하고 응답하는 방식으로 동작합니다.





02. WHAT IS REST API

REST의 정의

1. 웹의 장점을 최대한 활용할 수 있는 아키텍처 스타일이다.
2. 네트워크 상에서 Client 와 Server 사이의 통신 방식 중 하나이다.

REST의 장점

1. REST API 사용을 위한 별도의 인프라를 구축할 필요가 없다.
2. HTTP 프로토콜의 표준을 최대한 활용하여 여러 추가적인 장점을 함께 가져갈 수 있게 해준다.

REST의 구성 요소

자원(Resource): URI
모든 자원에 고유한 ID가 존재하고, 이 자원은 Server에 존재한다.

행위(Verb): HTTP Method
HTTP 프로토콜의 Method를 사용한다.

REST API: REST 기반으로 서비스 API를 구현한 것



02. WHAT IS REST API

REST의 특징

Statelessness (무상태성)

Client의 context를
Server에 저장하지 않는다

즉, 세션과 쿠키와 같은 context
정보를 신경쓰지 않아도 되므로
구현이 단순해진다

Uniform Interface (일관된 인터페이스)

URI로 지정한 Resource에 대
한 조작을 통일되고 한정적인
인터페이스로 수행한다.

HTTP 표준 프로토콜에 따르는 모
든 플랫폼에서 사용이 가능하다.

Cacheable (캐시 처리 가능)

웹에서 사용하는 기존의 인프라를
그대로 활용할 수 있다.

즉, HTTP가 가진 가장 강력
한 특징 중 하나인 캐싱 기능
을 적용할 수 있다.

웹 캐싱:정적 콘텐츠(이미지, JS, CSS 등)를 특정 위치(client, network 등)에 저장하여,사이트 응답시간을 줄이는 기능



03. 개발 목적

'학교 홈페이지에 설치되어있는
TV가 실용적이지 않아서'

85%

'세미나'

10%





04. 개발 과정

02 AWS 구현

Lambda에 적용 하
려고 하였으나 실패

03 급식 구현

나이스 API를 활용하
여 구현함

01 날씨 구현

Openweatherapi
대충 꺼드럭

04 작업 스케줄러

lambda로 하려고 하였
으나 막혀서 작업 스케
줄러로 바꿔서 적용

'핵심 키워드'
꺼드럭 꺼드럭



05. 코드 사진

```
weatherapp.py / message_send
import requests
import json
import pandas as pd
from datetime import datetime, timedelta

def weathers(lat, lon):
    url = 'https://api.openweathermap.org/data/2.5/weather?lat={}&lon={}&appid=6fc3bac96f01ebc8e27a717aad48218a&units=metric'.format(lat, lon)

    res = requests.get(url)
    data = res.json()

    temp = data['main']['temp']
    max_temp = data['main']['temp_max']
    min_temp = data['main']['temp_min']
```

대충 꺼드럭 꺼드럭



05. 코드 사진

```
def school_menu(current_date_time):
    next_day = current_date_time + timedelta(days=1)
    formatted_date = next_day.strftime("%Y%m%d")
    fields = ("MMEAL_SC_NM", "MLSV_YMD", "DDISH_NM", "CAL_INFO")
    service_key = ("15f44e0238a34188b681b4c7225143fd", "")
    url = "https://open.neis.go.kr/hub/mealServiceDietInfo"
    params = {
        "KEY": service_key[0],
        "Type": "json",
        "pIndex": "1",
        "pSize": "100",
        "SD_SCHUL_CODE": "7010537",
        "ATPT_OFCDC_SC_CODE": "B10",
        "MLSV_YMD": formatted_date
    }

    response = requests.get(url, params=params)
    # Parse the JSON response
    data = response.json()

    # Now you can extract the necessary information as shown in the previous example
    schul_info = data["mealServiceDietInfo"][1]["row"][0]
    school_name = schul_info["SCHUL_NM"]
    meal_date = schul_info["MLSV_YMD"]
    meal_menu = schul_info["DDISH_NM"].replace("<br/>", "\n")
    calories = schul_info["CAL_INFO"]
    nutrient_info = schul_info["NTR_INFO"].replace("<br/>", "\n")

    return f'오늘 급식은: {school_name} , {meal_date} , {meal_menu} , {calories} '
```

대충 꺼드럭 꺼드럭



05. 코드 사진

```
def message_send(message):
    # Your Kakao API credentials and URL
    manual_code = "WUBg0pkPR0u9Cya57FiWAUJhz655tvTde1sVEy_WAV_rLJkrhWfKECzneggKKcjZAAABjMQkykhDz1szkZmFRA"
    token_url = "https://kauth.kakao.com/oauth/token"
    send_url = "https://kapi.kakao.com/v2/api/talk/memo/default/send"

    # Request a token using the authorization code
    data = {
        "grant_type": "authorization_code",
        "client_id": "92c44268fe79aba5f10c8d4ed21c22a2",
        "redirect_url": "https://localhost:3000",
        "code": manual_code
    }
    response = requests.post(token_url, data=data)
    tokens = response.json()

    # Save the refresh token for future use
    if "refresh_token" in tokens:
        with open("./kakao_code.json", "w") as fp:
            json.dump(tokens, fp)
```

```
# Use the refresh token to get an access token
with open("./kakao_code.json", "r") as fp:
    tokens = json.load(fp)

refresh_token = tokens['refresh_token']
data = {
    "grant_type": "refresh_token",
    "client_id": "92c44268fe79aba5f10c8d4ed21c22a2",
    "refresh_token": refresh_token
}
response = requests.post(token_url, data=data)
tokens = response.json()

if 'access_token' in tokens:
    access_token = tokens['access_token']

    # Send the message using the obtained access token
    headers = {"Authorization": "Bearer " + access_token}
    data = {
        "template_object": json.dumps({
            "object_type": "text",
            "text": message,
            "link": {"web_url": "www.google.co.kr"}
        })
    }
    response = requests.post(send_url, headers=headers, data=data)
    print(response.json())
else:
    print("Error: Unable to obtain access token")
```

대충 꺼드럭 꺼드럭



05. 작동 사진

오늘 날씨는 4.29입니다. 최고기온은 4.51입니다. 최저기온은 3.41입니다.

자세히 보기

weather >

오늘 급식은: 세명컴퓨터고등학교, 20240102, 혼합잡곡밥 (5)
들깨미역국 (5.6.16)
매운돼지갈비찜 (1.5.6.10.13.18)
배추김치 (9)
생일케이크 (1.2.5.6)
도토리묵&양념장 (5.6.13), 898.9 Kcal
, 탄수화물(g) : 131.7
단백질(g) : 37.2
지방(g) : 23.0
비타민A(R.E) : 194.1
티아민(mg) : 1.1
리보플라빈(mg) : 0.9
비타민C(mg) : 8.2
칼슘(mg) : 1298.3
철분(mg) : 3.9



감사합니다.

