

네이버 날씨 크롤링을 이용한 RestFul API 제작

2019.05.28

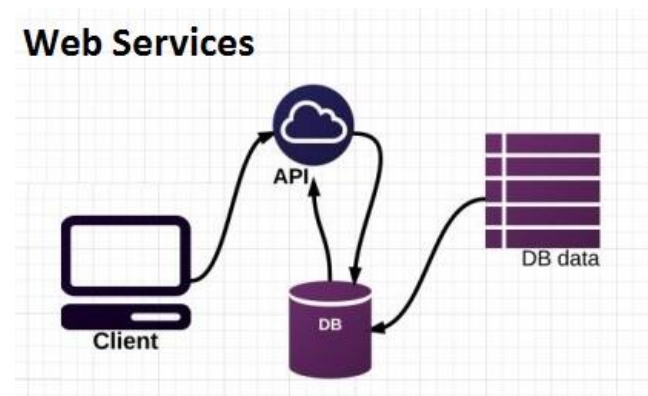
**지구과학 심화탐구 발표
신재욱**

RESTful API란?

- REpresentational State Transfer
- HTTP 프로토콜을 통해 웹으로 전달되는 API

- URL
- 인자
- 표현

의 요소가 반드시 들어있어야 함.



RestFul API 구성요소 - URL

- 서버가 있는 주소의 URL
- 예를 들어 오랑봇의 RESTful API의 URL은 free.jaeuk.xyz:7002

RestFul API 구성요소 - 인자

- HTTP 프로토콜은 인자를 다양한 형식으로 전달
- 대표적인 형식으로는 GET과 POST가 있음.
- GET은 인자의 정보를 <http://url.com?param=aaa> 와 같이 URL에 포함하여 전달
- POST는 URL에 인자의 정보를 드러내지 않음

RestFul API 구성요소 - 표현

- RESTful API는 요청이 들어올 때 그 때의 정보를 실시간으로 전달하여야 함.
- 보통 json이나 xml와 같은 표준화된 형식으로 데이터를 전달

우리가 할 것

- 네이버 날씨의 날씨 정보를 크롤링하여 Restful API 형태로 제작
- GET형식으로 지역을 인자로 요청할 경우 해당 지역의 날씨를 json형식으로 전달

날씨정보



우리가 사용할 환경

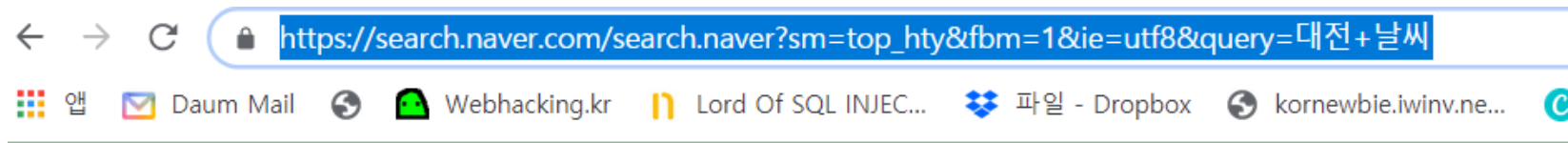
- Python 3.6.5
- IDE: Pycharm Professional
- 개발환경: Windows 10 Pro
- 서버환경: Ubuntu16.04 LTS



RestFul API 설계정보

- url: 미리 가지고 있는 서버와 도메인이용
(free.jaeuk.xyz:7003)
- 인자: GET 형식으로 region인자를 받음 (예:
free.jaeuk.xyz:7003?region=대전)
- 표현: json을 이용해
temp(기온), weather(날씨) 값 전송

분석 - 소스웹 키희리 주소



- `https://search.naver.com/search.naver?sm=top_hy&fbm=1&ie=utf8&query=대전+날씨`
- 위의 부분에서 query 인자에 의해 검색이 이루어짐

분석 - html 태그와 클래스

```
▼<div class="info_data">  
  ▼<p class="info_temperature">  
    <span class="todaytemp">17</span> == $0  
    ▶<span class="tempmark">...</span>
```

- 기온은 p태그 “todaytemp” 밑 span태그 “todaytemp”에 있음

```
  </p>  
▼<ul class="info_list">  
  ▼<li>  
    <p class="cast_txt">흐림, 어제보다 9° 낮아요</p> =  
  </li>
```

- 날씨정보는 ul태그 “info_list” 밑 p태그 “cast_txt”에 있음

사용할 모듈

- urllib : 웹을 크롤링해서 html 형식으로 가져오는 모듈
- bs4: beautifulsoup모듈로, 파싱된 html에서 일부 태그를 뽑아 사람이 보기 편한 형식으로 변환해주는 모듈
- Flask: 웹 구현
- Urllib를 제외하곤 외장함수이므로 pip(pypi)를 이용해 미리 다운로드 받아놓자

```
C:\Users\Yelli\Laptop>pip install bs4
Collecting bs4
  Downloading https://files.pythonhosted.org/packages/10/ed/7e8b97591f6f456174139ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz
Collecting beautifulsoup4 (from bs4)
  Downloading https://files.pythonhosted.org/packages/1d/5d/3260694a59df0ec52f8b4883f5d23b130bc237602a1411fa670eae12351e/beautifulsoup4-4.7.1-py3-none-any.whl (94kB)
100% |#####| 102kB 123kB/s
Collecting soupsieve>=1.2 (from beautifulsoup4->bs4)
  Downloading https://files.pythonhosted.org/packages/b9/a5/7ea40d0f867bde6e464a6435a48bc5db09b1a8f4f06d41dd997b8f3c616/soupsieve-1.9.1-py2.py3-none-any.whl
Installing collected packages: soupsieve, beautifulsoup4, bs4
  Running setup.py install for bs4 ... done
Successfully installed beautifulsoup4-4.7.1 bs4-0.0.1 soupsieve-1.9.1
You are using pip version 10.0.1, however version 19.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

전체 코드

```
# -*- coding: utf-8 -*-
from urllib.request import urlopen
import urllib
import bs4
from flask import *
from json import *

app=Flask(__name__)

@app.route('/', methods= ['get'])
def api():
    try:
        location=request.args.get('region')
        enc_location = urllib.parse.quote(location + '+날씨')
        url = 'https://search.naver.com/search.naver?ie=utf8&query='+ enc_location
        req = urllib.request.Request(url)
        page = urlopen(req)
        html = page.read()
        soup = bs4.BeautifulSoup(html,'html.parser')
        temp = str(soup.find('p', class_='info_temperature').find('span', class_='todaytemp').text)
        weather= str(soup.find('ul', class_='info_list').find('p', class_='cast_txt').text)

        jsondict={
            'temp':temp,
            'weather':weather
        }
        jsonstr=json.dumps(jsondict, ensure_ascii=False)
        res=make_response(jsonstr)
        res.headers['Content-Type']='application/json'
        return res

    except:
        return "ERROR"

if __name__=='__main__':
    app.run(host='0.0.0.0',port=7003,debug=False)
```

핵심 코드 - 인자 전달

```
@app.route( '/', methods= [ 'get' ] )  
def api():  
    try:  
        location=request.args.get( 'region' )
```

- Get으로 region인자를 받아서 location 변수에 저장

핵심 코드 - 크롤링/파싱

```
enc_location = urllib.parse.quote(location + '+날씨')
url = 'https://search.naver.com/search.naver?ie=utf8&query='+ enc_location
req = urllib.request.Request(url)
page = urlopen(req)
html = page.read()
soup = bs4.BeautifulSoup(html, 'html.parser')
temp = str(soup.find('p', class_='info_temperature').find('span', class_='todaytemp').text)
weather= str(soup.find('ul', class_='info_list').find('p', class_='cast_txt').text)
```

- 네이버 날씨 정보 크롤링 및 파싱하여 temp와 weather 변수에 기온과 날씨정보 저장

핵심 코드 - json 처리 및 웹에 출력

```
jsondict={
    'temp':temp,
    'weather':weather
}
jsonstr=json.dumps(jsondict, ensure_ascii=False)
res=make_response(jsonstr)
res.headers['Content-Type']='application/json'
return res
```

- Json으로 처리하여 웹에 출력합니다.

핵심 코드 - 보안 처리

```
except:
    return "ERROR"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=7003, debug=False)
```

- 에러가 발생할 경우 예외처리를 통해 Error라고 출력합니다.
- 보안을 위해 debug 모드를 False로 처리합니다.

서버 업로드

```
luke7864@free-tier:~/weather_api$ ls
api.py
luke7864@free-tier:~/weather_api$ python3 api.py
* Serving Flask app "api" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:7003/ (Press CTRL+C to quit)
```

- 서버에 업로드하여 실행시켜줍니다.

결과

← → ↻ ⓘ 주의 요함 | free.jaeuk.xyz:7003/?region=대전

앱 Daum Mail Webhacking.kr Lord Of SQL INJEC...

```
{"temp": "16", "weather": "흐림, 어제보다 9° 낮아요"}
```

← → ↻ ⓘ 주의 요함 | free.jaeuk.xyz:7003/?region=서울

앱 Daum Mail Webhacking.kr Lord Of SQL INJEC...

```
{"temp": "16", "weather": "구름많음, 어제보다 7° 낮아요"}
```

- 성공적으로 json 형식으로 결과값을 전송함을 알 수 있습니다.

감사합니다.

해당 PPT에서 사용된 소스코드는
제 깃허브에서 모두 확인하실 수 있습니다.

https://github.com/Luke7864/weather_api