

# ILE3-013 Web API

## 개요

포털사에서 제공하는 WebAPI을 사용해보자

## RPC (Remote Procedure Call)

원격 제어를 위한 코딩 없이 다른 주소 공간에서 함수나 프로시저를 실행할 수 있게하는 프로세스 간 통신 기술

- 원격 프로시저 호출을 이용하면 프로그래머는 함수가 실행 프로그램에 로컬 위치에 있던 원격 위치에 있던 동일한 코드를 이용할 수 있음

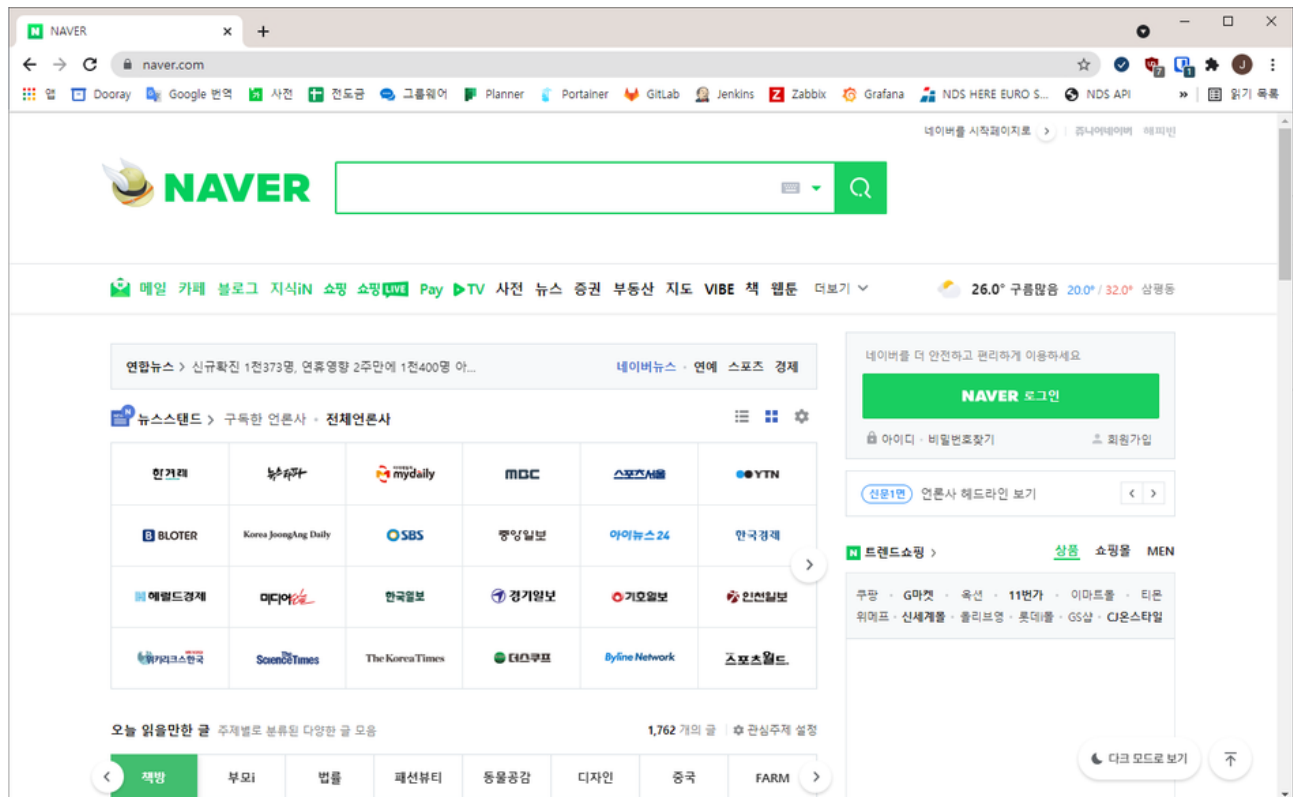
## Web API

Web을 통하여 원격지에서 다른 개발자가 구현해 둔 기능을 Web(HTTP)를 이용하여 사용하는 기술

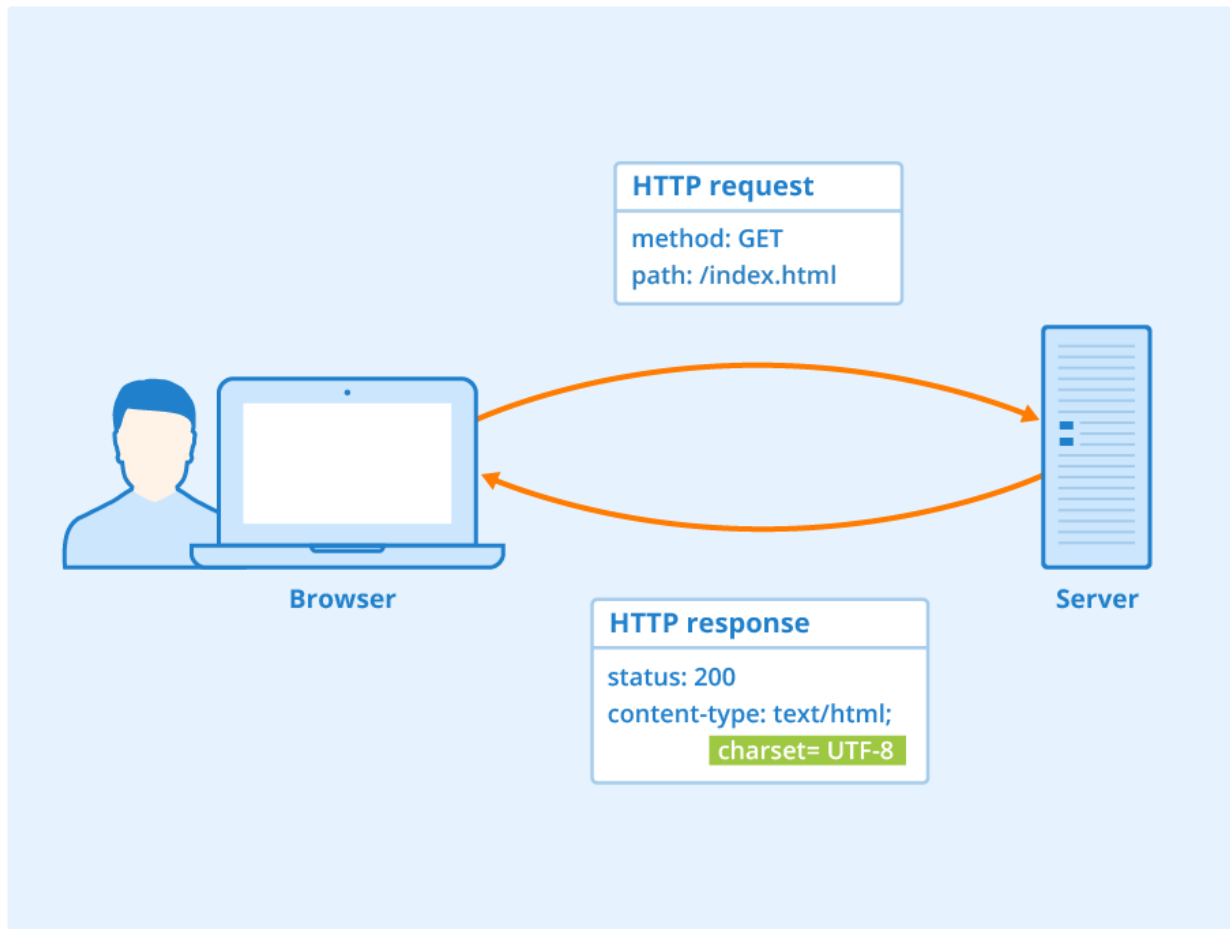
- 정부 및 대형 포털 사이트(Naver, Kakao) 등에서 제공 중

## HTTP

인터넷을 사용할 때 가장 많이 사용하는 프로토콜 -> 웹 브라우저 (Chrome 등등)



## HTTP의 구성 [🔗](#)



데이터의 요청을 요구하는 작업인 **Request** 와 Request에 대한 회신값인 **Response** 로 구성

## Request [🔗](#)

일반적으로 우리가 URL로 부르는 것 + a

```
1 GET https://www.naver.com/
```

단순하게 URL으로만 호출하는것 같지만 실제로는 더 많은 데이터가 존재

Request Headers	
GET / HTTP/1.1	
<b>Client</b>	
Accept:	text/html, application/xhtml+xml, image/jxr, */*
Accept-Encoding:	gzip, deflate
Accept-Language:	ko
User-Agent:	Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
<b>Transport</b>	
Connection:	Keep-Alive
Host:	61.33.249.242

## Response [🔗](#)

Request에 대한 회신

우리가 실제로 웹 브라우저를 이용해서 보거나, 다운로드 받은 파일이 Response에 해당 됨

```
1 <!doctype html> <html lang="ko" data-dark="false"> <head> <meta charset="utf-8"> <title>NAVER</tit
2 .....
```

Request와 같이 아래와 같이 숨겨진 데이터도 존재

Response Headers
HTTP/1.1 200 OK
<b>Cache</b>
Cache-Control: public
Date: Fri, 18 Sep 2020 00:21:43 GMT
Expires: Sat, 18 Sep 2021 00:21:43 GMT
<b>Entity</b>
Content-Encoding: gzip
Content-Type: text/css; charset=utf-8
Last-Modified: Fri, 31 Jan 2020 22:45:51 GMT
<b>Miscellaneous</b>
Referrer-Policy: strict-origin-when-cross-origin
Server: nginx
<b>Security</b>
Strict-Transport-Security: max-age=31536000
<b>Transport</b>
Connection: keep-alive
Transfer-Encoding: chunked

## Python에서의 HTTP 사용 ↗


### 표준 Module ↗

#### http Module ↗

HTTP Protocol을 구현한 Python 기본 Module →

```
1 import http
```

 [http — HTTP 모듈](#)

 Protocol 기본부터 구성되어 있어 사용하지 않음

#### http.client Module ↗

http Module 중 Client 기능에 해당하는 Python 기본 Module

```
1 import http.client
```

 [http.client — HTTP 프로토콜 클라이언트](#)

 Low Level API이기 때문에 사용하기가 어려워서 잘 안사용함

#### urllib.request Module ↗

```
1 import urllib.request
```

 [urllib.request — Extensible library for opening URLs](#)

## 기본적인 Request 전달 및 Response 수신 ↗

```

1 import urllib.request
2
3 url = 'http://www.naver.com'
4 request = urllib.request.Request(url)
5 conn = urllib.request.urlopen(request)
6 data = conn.read()
7
8 print(data.decode('utf-8'))
9 conn.close()

```

## Header를 추가하여 Request 전달 🔗

일반적으로 보내는 데이터의 Format, 압축 여부 및 인증 정보를 Header에 포함하여 전달

```

1 import urllib.request
2
3 url = 'https://www.foo.bar/api'
4 request = urllib.request.Request(url)
5 request.add_header('Header1', 'HeaderValue1')
6 request.add_header('Header2', 'HeaderValue2')
7 conn = urllib.request.urlopen(request)
8 data = conn.read()
9
10 print(data.decode('utf-8'))
11 conn.close()

```

## 매개변수를 추가하여 전달 🔗

아래와 같이 URL 뒤에 ? 를 붙여 매개변수를 전달

<https://openapi.naver.com/v1/search/local.xml?query=주식&display=10&start=1&sort=random>

## URL Encode 🔗

HTTP Request는 ASCII Format이기 때문에 한글 및 특수문자를 표현 불가능

-> ASCII 범위의 문자열로 변환하여 전송

```

1 param1=%EA%B0%80%EB%82%98%EB%8B%A4%EB%9D%BC%EB%A7%88%EB%B0%94%EC%82%AC&param2=123&param3=asdf

```

`urllib.parse.urlencode()` 를 이용하여 URL Encoding된 문자열로 변환이 가능

```

1 import urllib.request, urllib.parse
2
3 params = {}
4 params['param1'] = '가나다라마바사'
5 params['param2'] = 123
6 params['param3'] = 'asdf'
7
8
9 url = 'https://www.foo.bar/api'
10 request = urllib.request.Request(f'{url}?{urllib.parse.urlencode(params)}')
11 request.add_header('Header1', 'HeaderValue1')
12 request.add_header('Header2', 'HeaderValue2')
13
14 conn = urllib.request.urlopen(request)
15 data = conn.read()
16 print(data.decode('utf-8'))

```

```
17
18 conn.close()
```

## 과제

아래와 같이 네이버 지역 검색한 결과를 보여주는 프로그램을 Python으로 작성

### 1. Key 발급

Naver Open API에 가입하여 각자 Key 발급 (발급 받은 키는 각자 잘 관리 할 것)

### 2. 네이버 쇼핑 API를 이용해서 마음에 드는 항목에 대해서 검색 결과를 출력

## 예제

```
1  검색어 입력: 판교역
2
3  1. <b>판교역</b> 신분당선
4      link:
5      category: 교통,운수>지하철,전철
6      description:
7      telephone:
8      address: 경기도 성남시 분당구 백현동 19-2
9      roadAddress: 경기도 성남시 분당구 판교역로 160
10     mapx: 321496
11     mapy: 532902
```