

Lab 07. Node.js and http open api

인터넷과웹기초



Contents

- HTTP message 형식 확인
- Node.js 및 Express.js 설치
 - Hello world 작성
 - Routing 실습 및 결과 확인
- JavaScript를 통해 Open API 호출 실습

HTTP Message 확인

The image shows a web browser window displaying the Pusan National University (PNU) homepage. The page features a blue header with a navigation bar and a main content area with a large eagle logo and the text "학문의 날개를 펼치다" (Spread the wings of learning). Below this, there are sections for "2020학년도 1학기 전체" and "2019년 1학기 9월 입학" (September 2019 admission).

On the right side of the browser window, the Network tab is open, showing a list of resources. The resource "mcon2_bg6.png" is selected, and its headers are displayed. The headers are categorized into General, Response Headers, and Request Headers.

General Headers:

- Request URL: http://www.pusan.ac.kr/_contents/kor/_Img/Main/mcon2_bg6.png
- Request Method: GET
- Status Code: 200 OK
- Remote Address: 164.125.8.119:80
- Referrer Policy: no-referrer-when-downgrade

Response Headers:

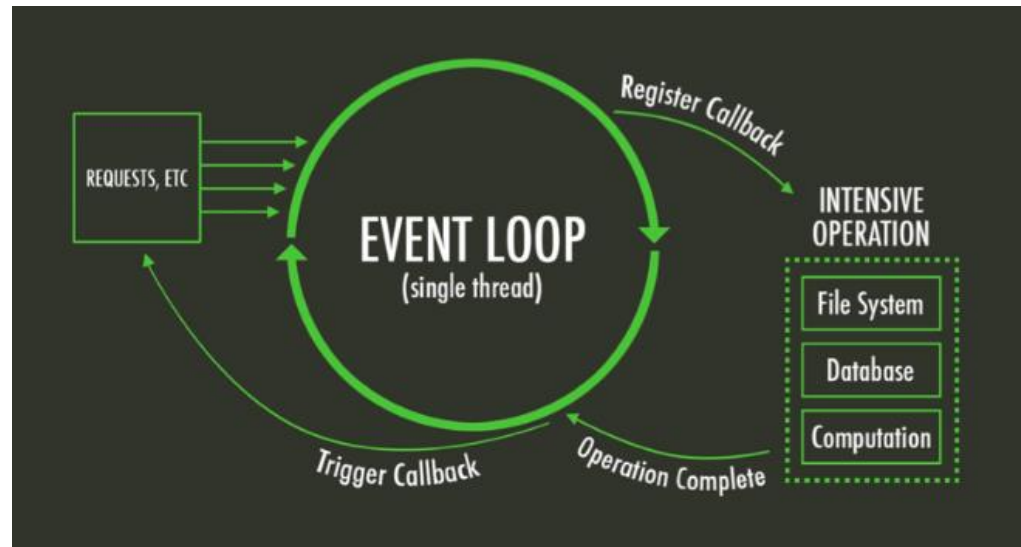
- HTTP/1.1 200 OK
- Date: Mon, 11 May 2020 06:56:15 GMT
- X-Powered-By: Servlet/3.0
- Last-Modified: Mon, 01 Jan 2018 02:06:50 GMT
- Cache-Control: no-store
- Content-Length: 7305
- Accept-Ranges: bytes
- X-Frame-Options: ALLOW-FROM http://mw.pusan.ac.kr/
- Keep-Alive: timeout=10, max=76
- Connection: Keep-Alive
- Content-Type: image/png
- Content-Language: ko-KR

Request Headers:

- GET /_contents/kor/_Img/Main/mcon2_bg6.png HTTP/1.1
- Host: www.pusan.ac.kr
- Connection: keep-alive
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36
- Accept: image/webp,image/apng,image/*,*/*;q=0.8
- Referer: http://www.pusan.ac.kr/kor/Main.do
- Accept-Encoding: gzip, deflate
- Accept-Language: ko,en-US;q=0.9,en;q=0.8,ko-KR;q=0.7
- Cookie: _ga=GA1.3.1957459281.1584168269; _gid=GA1.3.332620683.1589175251; JSESSIONID=0000V7Q4uezcov8MVXs03tfeK-1-1; _gat_gtag_UA_75737770_2=1

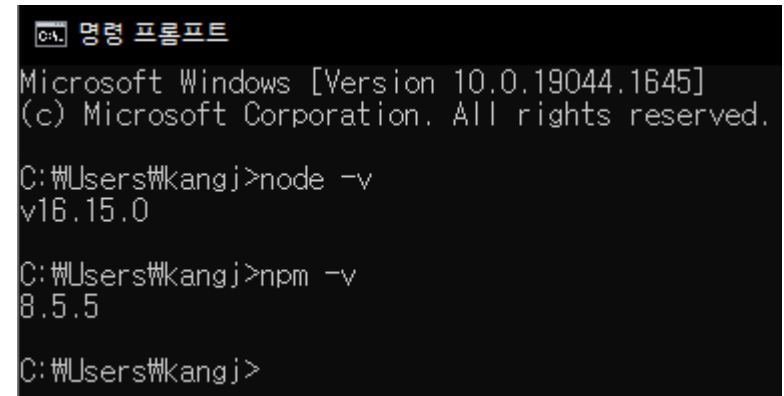
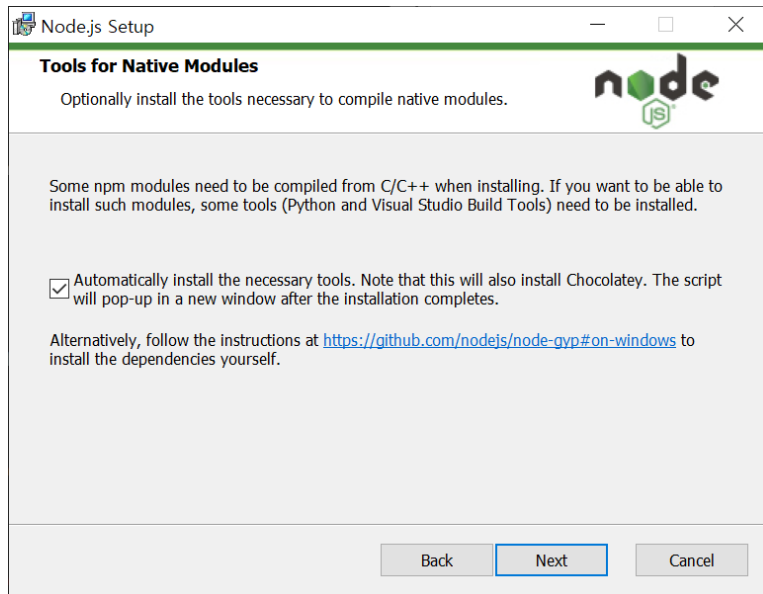
Node.js and Express.js

- Node.js: An asynchronous event-driven JavaScript runtime
 - Use v8 JavaScript engine in Google Chrome
 - Node.js runs on various platforms (Windows, Linux, Unix, Mac OS)
 - Event-driven architecture
 - Asynchronous I/O
- Express.js: fast, minimalist web framework for node.js



Node.js 설치

- Node.js 홈페이지: <https://nodejs.org/ko/download/>
- LTS (Long Term Support) Version 설치
 - Node JS를 설치하면 연관된 Package Manager 도구인 NPM(<https://www.npmjs.com/>) 이 자동 설치됨
- 설치 확인
 - CMD 창 실행 후 node 및 npm version 확인



Express.js 설치

- <http://expressjs.com>
- [Installing 페이지](#) 참고 Express 설치
 - Command Prompt 실행
 - > cd \
 - > mkdir myapp
 - Express로 구성할 myapp이라는 폴더(디렉토리) 생성
 - > cd myapp
 - 생성한 폴더로 이동
 - > npm init
 - Express Web Service의 entry-point에 해당하는 파일 지정
 - 기본값을 모두 그대로 수용
 - > npm install express --save
 - Express 설치 및 dependencies 정보 갱신

Command Prompt

```
C:\Users\admin>cd \

C:\>mkdir myapp

C:\>cd myapp

C:\myapp>npm init
This utility will walk you through creating a
package.json file.
. . .
Press ^C at any time to quit.
package name: (myapp)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\myapp\package.json:

{
  "name": "myapp",
  . . .

Is this OK? (yes) yes

C:\myapp>npm install express --save
```

Hello World – Index.js 작성

- Express를 실행하기 위한 기본 JavaScript File – index.js 작성
 - vscode를 실행하고 /myapp 폴더를 open
 - myapp 폴더에 index.js 파일을 추가
 - [Hello World Example](#) Web Page 또는 아래를 참고하여 index.js 작성
 - 또는 Download 받은 sources 내의, myapp 폴더 내의 index.js를 /myapp으로 복사

myapp



index.js



```
index.js
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => res.send('Hello World!'))

app.listen(port, () => console.log(`Example app listening at http://localhost:${port}`))
```

<http://expressjs.com/en/starter/hello-world.html> 참고

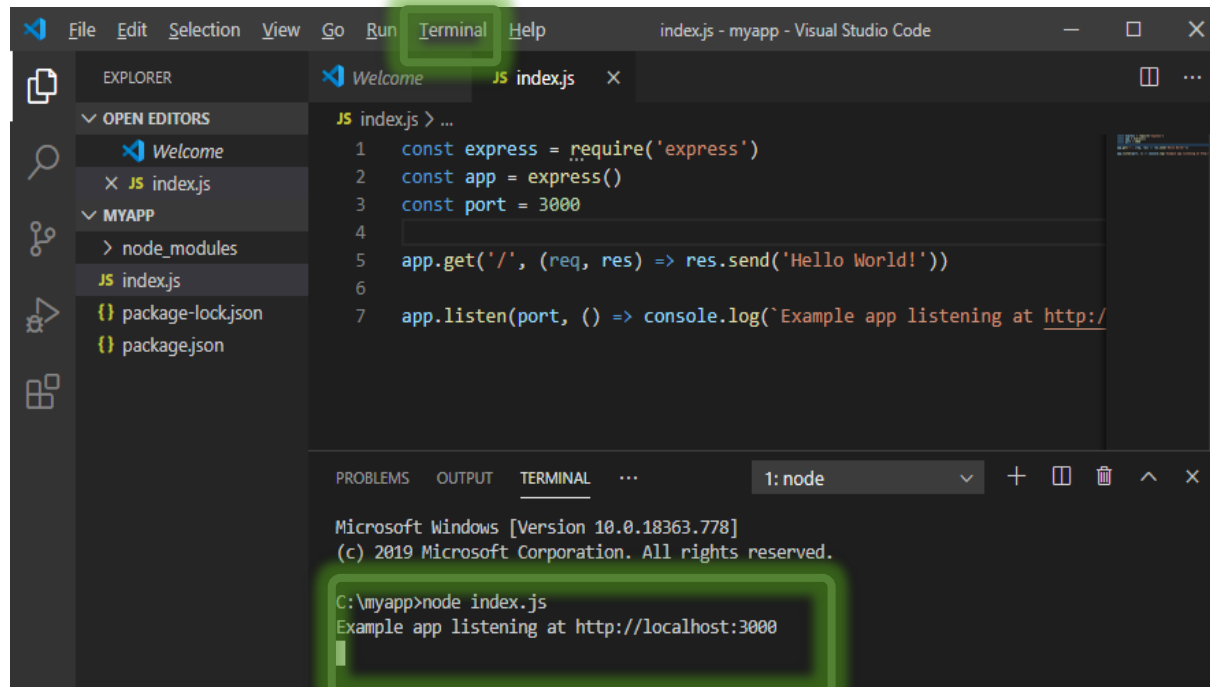
Express 실행

- Command 창에서 index.js를 인자로 node를 실행

```
C:\myapp>node index.js  
Example app listening at http://localhost:3000
```

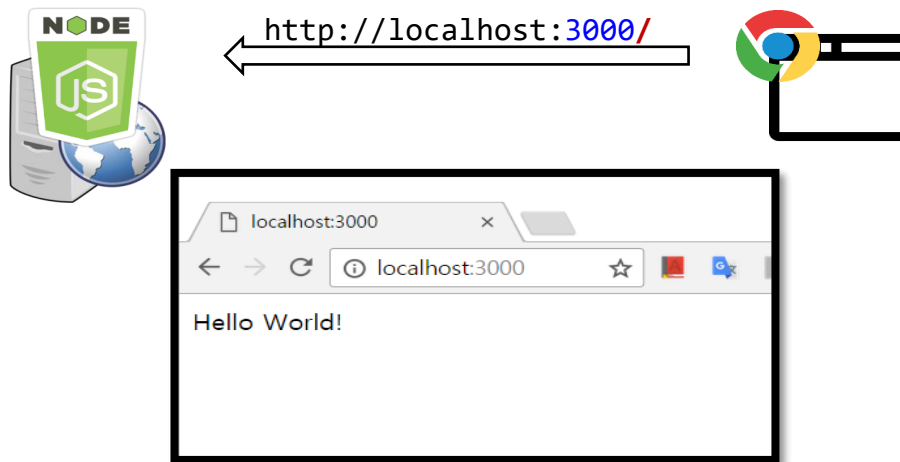
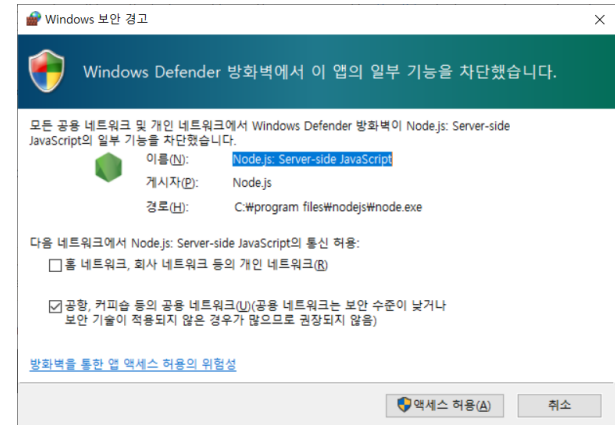
Command Prompt

- Command 창은 VS Code에서도 생성 가능



index.js 실행 확인

- Node.js 실행 시의 보안 경고 메시지
 - ➔ 액세스 허용
 - 내 Computer에서 express 라는 web 서버가 port 번호 3000에서 구동
- Browser 실행 후 내 Local Web Server에 접속
 - Hello World! 출력



index.js 수정하는 법

- 현재 구동 중인 Web Server 구동 중단
 - Node를 실행한 터미널 혹은 커맨드 창에서 Ctrl-C 입력하여 중단
- Index.js 에서 'Hello World!' 문자열 찾아서 수정

myapp



index.js



```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => res.send('Hello World!'))

app.listen(port, () => console.log(`Example app listening at http://localhost:${port}`))
```

- 수정한 index.js 파일을 재구동
- 브라우저로 접속하여 다시 확인

Node.js + Open API



Step-1: Node.js - Routing

- **Routing** refers to determining how an application responds to a client request to a particular endpoint, which is a URL (or path) and a specific HTTP request method (GET, POST, and so on)
- Index1.js라는 새로운 js 파일을 추가하라
 - Index1.js는 기존 Index.js에 아래 Code와 같이 Routing을 위한 추가 코드를 포함하고 있다.
- Index1.js 실행
 - 기존 서버 중단
 - node index1.js

myapp



index1.js



Index1.js 는 어디서 실행되나?
Node JS 서버 vs Browser

```
index1.j
...
const port = 3000

app.get('/', (req, res) => res.send('Hello World!'))

app.get('/Path1', function (req, res) {
  res.send("GET Path1");
})
app.get('/Path2', function(req,res) {
  res.send("GET Path2 : " + Date());
})
app.put('/Path1', function (req, res) {
  res.send("PUT Path1");
})

app.listen(port, () => console.log(`Example app listening at http://localhost:${port}`))
```

<http://expressjs.com/en/starter/basic-routing.html> 참고

Step-1: Node.js – Routing 수행 결과

← → ↻ 🏠 ⓘ 127.0.0.1:3000/Path1

GET Path1

← → ↻ 🏠 ⓘ 127.0.0.1:3000/Path2

GET Path2 : Tue May 12 2020 19:41:20 GMT+0900 (GMT+09:00)

PUT localhost:3000/Path1 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (6) Test Results

Status: 200 OK Time: 15 ms Size: 212 B Save

Pretty Raw Preview Visualize HTML

1 PUT Path1

PUT Method 테스트를 위해 Postman
활용

Step-2: Static Files

- (HTML, JS, CSS 등) Static Files의 추가/지원
- myapp 폴더 아래 “public” 이라는 폴더를 만들고
 - Public 폴더에 MyJS1.html
- 아래와 같이 index1.js를 수정하여 index2.js로 저장

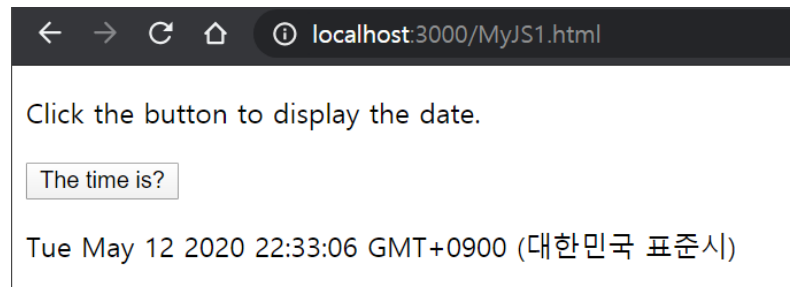
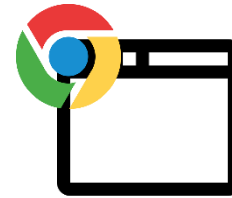


Step-2: Static Files

- index2.js를 재구동하고 static file들이 제대로 서비스되는지 확인



http://localhost:3000/
←
http://localhost:3000/MyJS1.html



MyJS1.html은 어디서 실행되나?
Node JS 서버 vs Browser

Step-3: JavaScript로 Open API 호출

- JavaScript를 이용한 공공데이터 기상청 단기예보 API 호출
 - Postman의 Code 생성 기능 활용
 - JavaScript Code 생성 (base_date에 유의)

http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst?serviceKey=2C8%2BSEhc...

GET http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst?serviceKey=2C8%2BSEhcC...

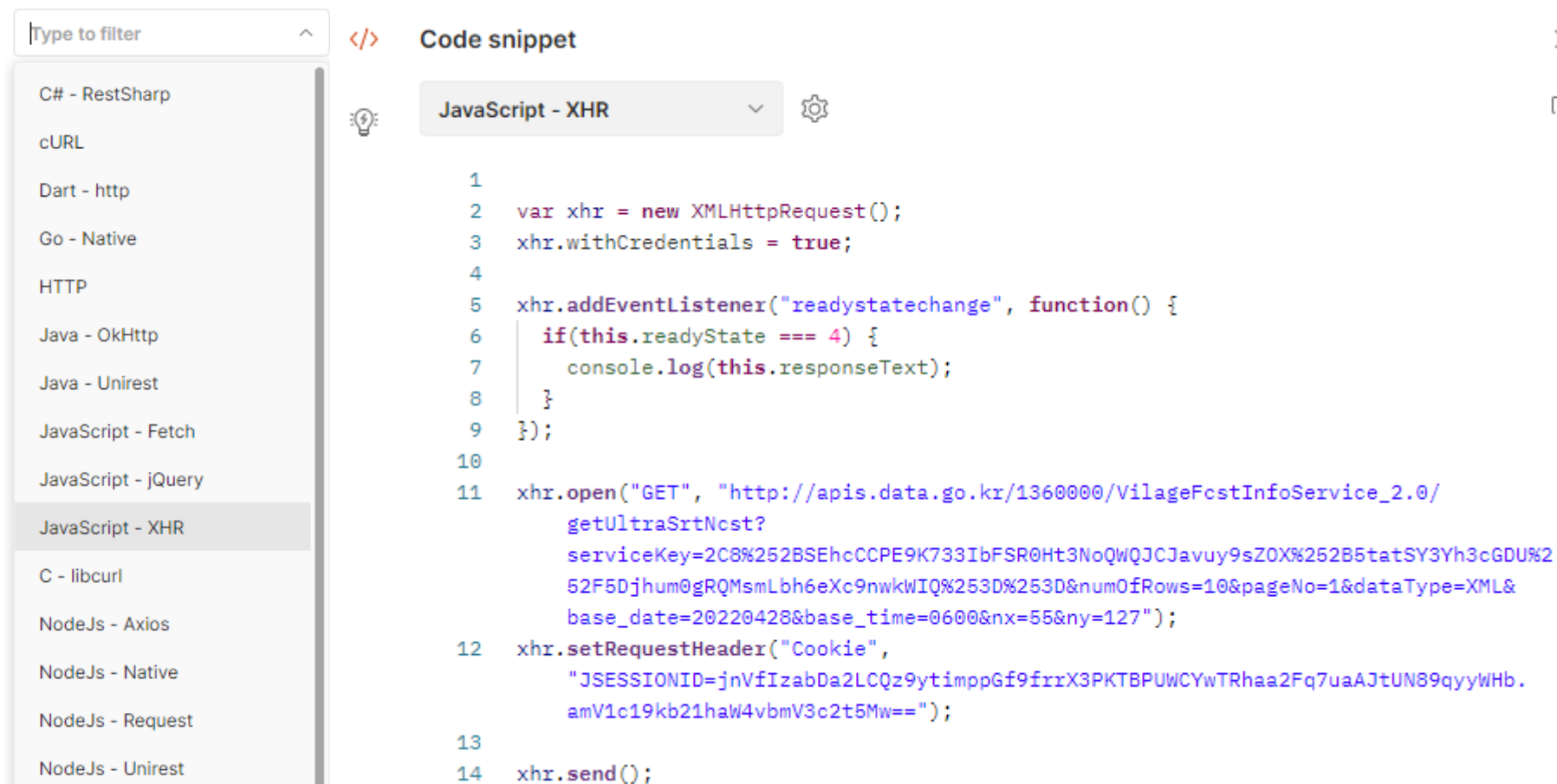
Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	serviceKey	2C8%2BSEhcCCPE9K733IbFSR0Ht3NoQ...			
<input checked="" type="checkbox"/>	numOfRows	10			
<input checked="" type="checkbox"/>	pageNo	1			
<input checked="" type="checkbox"/>	dataType	XML			
<input checked="" type="checkbox"/>	base_date	20220428			
<input checked="" type="checkbox"/>	base_time	0600			
<input checked="" type="checkbox"/>	nx	55			
<input checked="" type="checkbox"/>	ny	127			
	Key	Value	Description		

Step-3: Open API 호출

- XHR ([XMLHttpRequest](#)) 객체 활용
 - 브라우저에서 지원하는 객체로 Http Request를 쉽게 수행할 수 있음

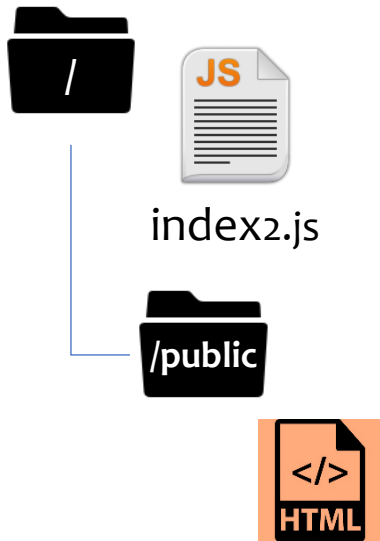


The screenshot shows a code editor interface. On the left, there is a search bar labeled "Type to filter" and a list of code snippets. The snippet "JavaScript - XHR" is selected and highlighted. On the right, the code for the XHR snippet is displayed. The code is a JavaScript snippet that creates an XMLHttpRequest object, sets its credentials to true, adds an event listener for thereadystatechange event, and then opens a GET request to a specific API endpoint. The request headers include a Cookie with a session ID and a base date. The code is as follows:

```
1
2 var xhr = new XMLHttpRequest();
3 xhr.withCredentials = true;
4
5 xhr.addEventListener("readystatechange", function() {
6     if(this.readyState === 4) {
7         console.log(this.responseText);
8     }
9 });
10
11 xhr.open("GET", "http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/
    getUltraSrtNcst?
    serviceKey=2C8%252BSEhcCCPE9K733IbFSR0Ht3NoQWQJCJavuy9sZ0X%252B5tatSY3Yh3cGDU%2
    52F5Djhum0gRQMsmLbh6eXc9nwkWIQ%253D%253D&numOfRows=10&pageNo=1&dataType=XML&
    base_date=20220428&base_time=0600&nx=55&ny=127");
12 xhr.setRequestHeader("Cookie",
    "JSESSIONID=jnVfIzabDa2LCQz9ytimppGf9frrX3PKTBPUWCYwTRhaa2Fq7uaAJtUN89qyyWHb.
    amV1c19kb21haw4vbmV3c2t5Mw==");
13
14 xhr.send();
```

Step-3: Open API 호출

- Open API를 호출하는 MyJS2.html 작성
 - MyJS1.html을 복사하고 아래와 같이 수정하여 MyJS2.html 작성



MyJS2.html 수정

```
JS index.js JS index1.js <> MyJS2.html JS index2.js
public > <> MyJS2.html > html > body
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <p>Click the button to display the date.</p>
6
7 <button onclick="displayResponse()">The time is?</button>
8
9 <script>
10 > function displayDate() { ...
13 > function displayResponse(){
14     var xhr = new XMLHttpRequest();
15     xhr.withCredentials = true;
16
17     xhr.addEventListener("readystatechange", function() { ...
23
24     xhr.open("GET", "http://apis.data.go.kr/1360000/VilageFcst
25     xhr.setRequestHeader("Cookie", "JSESSIONID=doQlQfuFJBL1BCx
26
27     xhr.send();
28 }
29 </script>
30 <p id="demo"></p>
31
32 </body>
33 </html>
```

코드 변경

Postman
생성 코드

Step-3: Open API 호출

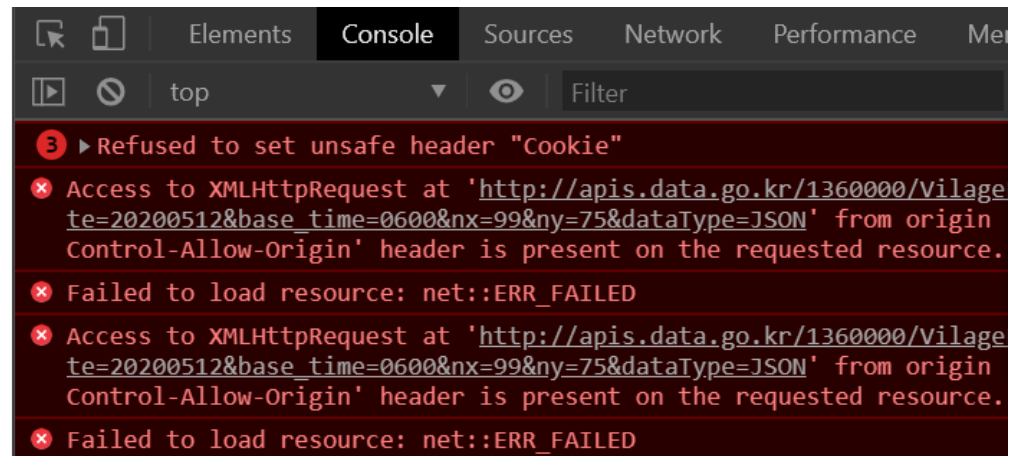
- MyJS2.html을 reload하여 결과 확인
 - 원하는 결과 값을 얻을 수 없음
 - 브라우저의 개발자 도구를 활용하여 디버깅 정보 확인 (console 확인)
 - chrome – 개발자 도구 (ctrl-shift-i)

Click the button to display the date.

The time is?

Blocked by CORS policy

<https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS> 참고

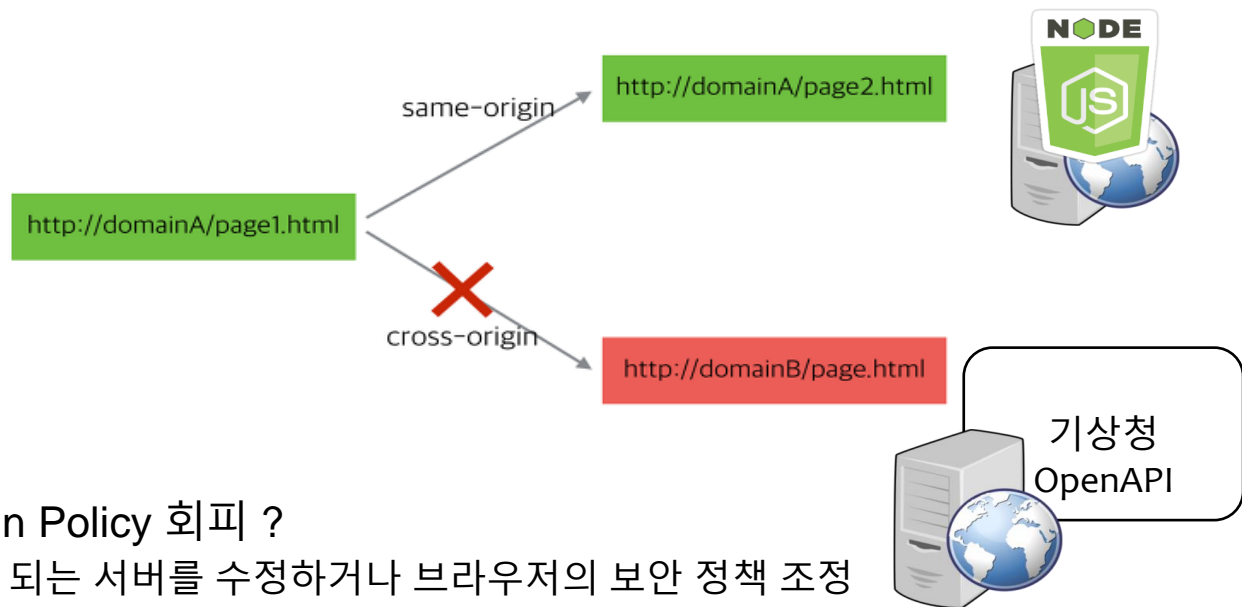


Step-3: Open API 호출

- Same Origin Policy – 웹 응용 보안 모델

- <http://adrenal.tistory.com/16> 참고

- 자바스크립트(XMLHttpRequest)로 다른 웹페이지에 접근할때는 같은 출처(same origin)의 페이지에만 접근이 가능하다. 같은 출처라는 것은 프로토콜, 호스트명, 포트가 같다는 것을 의미한다. 즉 쉽게 말하면 웹페이지의 스크립트는 그 페이지와 같은 서버에 있는 주소로만 요청을 할 수 있다는 것이다.



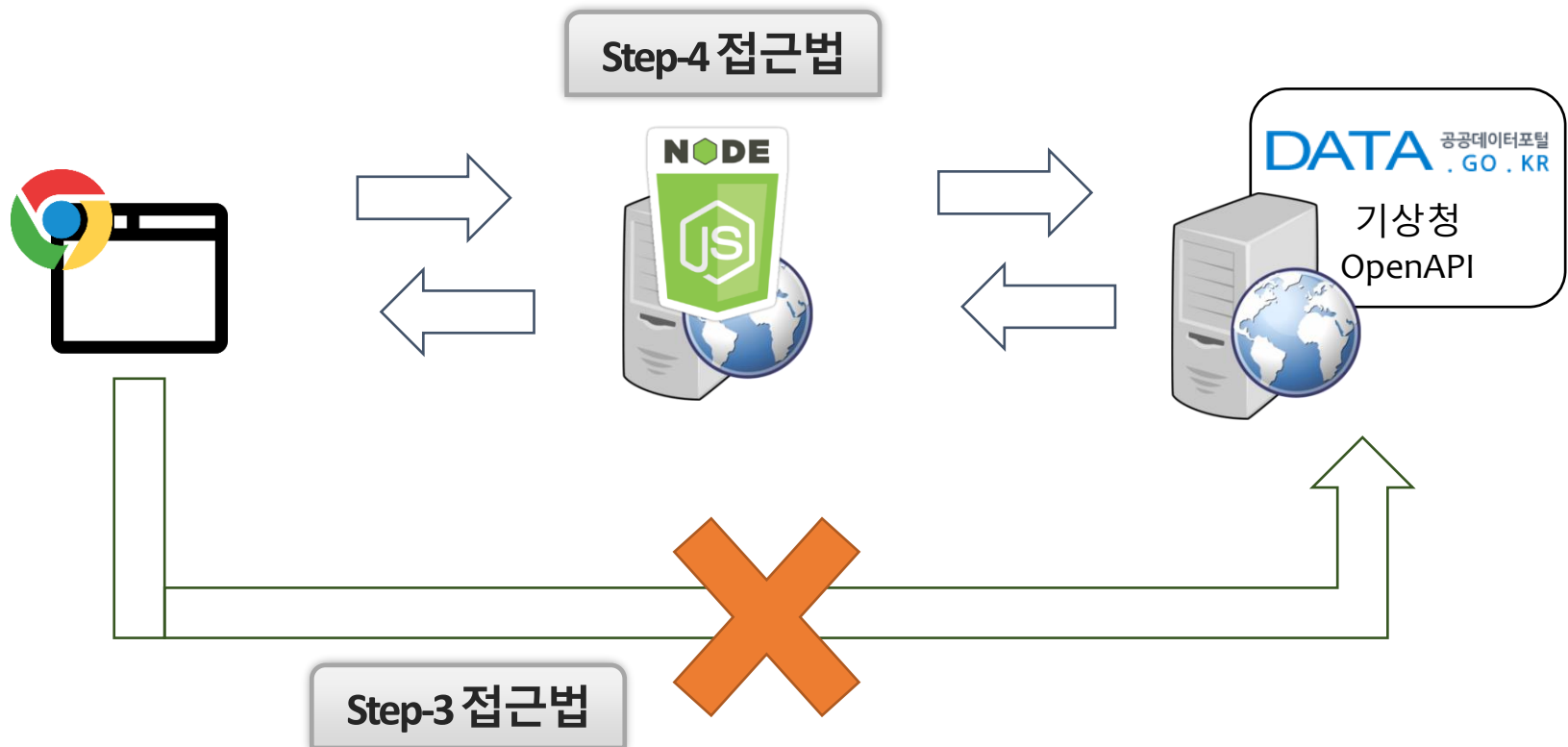
- Same Origin Policy 회피 ?

- 문제가 되는 서버를 수정하거나 브라우저의 보안 정책 조정
 - 대상 서버는 직접 수정 가능한 서버가 아님

- [Chrome에 Allow-Control-Allow-Origin Plug-In](#) 추가 : 합리적 해결책이 아님

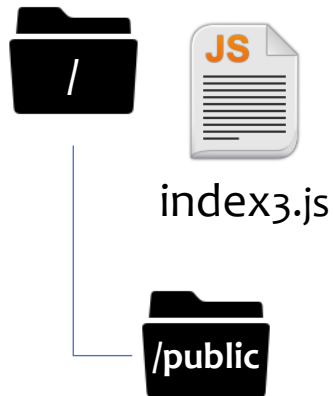
Step-4: Open API 호출 / Node.js에서 호출

- Client-Side JavaScript로 다른 도메인의 Open API 호출은 부적절함
 - Web 응용 보안 모델에 따라 제약이 발생
 - Server Side Script로 다른 도메인의 Open API 호출 ?



Step-4: Open API 호출 / Node.js

- Node JS에서 새로운 GET HTTP Request를 처리하는 index3.js 작성
 - 코드 작성을 위해 Postman의 코드 생성 기능 (NodeJS Request) 활용



```
GENERATE CODE SNIPPETS

Filter languages...

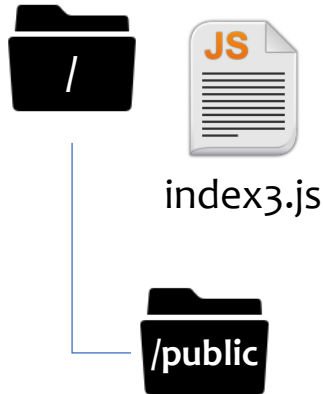
C - libcurl
C# - RestSharp
cURL
Go - Native
HTTP
Java - OkHttp
Java - Unirest
JavaScript - Fetch
JavaScript - jQuery
JavaScript - XHR
Nodejs - Native
Nodejs - Request

Generated code for Nodejs - Request

1 var request = require('request');
2 var options = {
3   'method': 'GET',
4   'url': 'http://apis.data.go.kr/1360000/VilageFcstInfoService
   /getUltraSrtFcst?serviceKey
   =jTD5BEAbCsM2ddtTbZqVrWgTXWlikqIJdhTgKJ915ZOha1%2FAjCes9bir%2FTI
   PqBiuOWraRPpgXfpVpneBjci0ww%3D%3D&base_date=20200512&base_time
   =0600&nx=99&ny=75&dataType=json',
5   'headers': {
6     'Cookie': 'JSESSIONID
   =0dqdKkUkgChEbyI1DdZU0BaeXi96u05fwxdDSow16eVpH014YJdnYLj8vWZm
   1F6.amV1c19kb21haw4vbmV3c2t5Mw=='
7   }
8 };
9 request(options, function (error, response) {
10   if (error) throw new Error(error);
11   console.log(response.body);
12 });
13
```

- Node JS에서 Request를 사용하기 위해서 **Request Library Module** 추가 필요
 - `C:\myapp> npm install --save request`

Step-4: Open API 호출 / Node.js



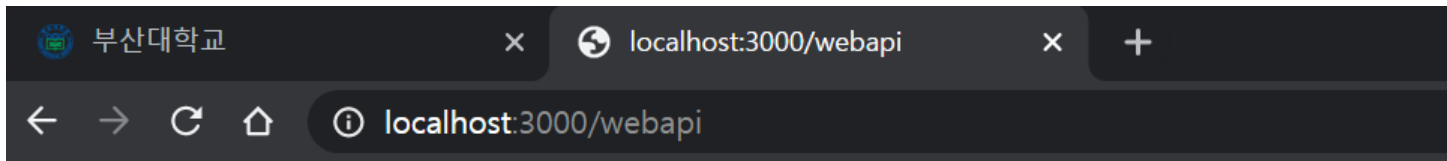
GET /webapi 에 대한 Handler 추가

Postman
생성 코드

```
3  const port = 3000
4
5  app.use(express.static('public'))
6
7  app.get('/', (req, res) => res.send('Hello World!'))
8
9  app.get('/webapi', function(req, res){ ← /webapi
10     var request = require('request');
11     var options = {
12         'method': 'GET',
13         'url': 'http://apis.data.go.kr/1360000/VilageFc
14         'headers': {
15             'Cookie': 'JSESSIONID=OdqdDKkUkgChEbyILDdZU0B
16         }
17     };
18     request(options, function (error, response) {
19         if (error) throw new Error(error);
20         console.log(response.body);
21         res.send(response.body); ← 코드추가
22     });
23
24
25 })
26
27 app.get('/Path1', function (req, res) {
```

Step-4: Open API 호출 / Node.js

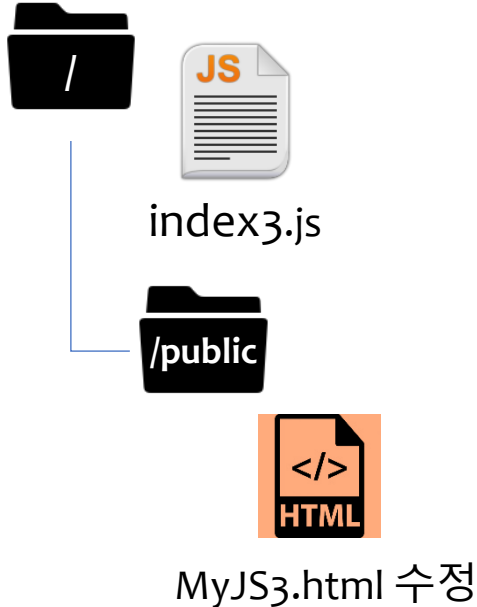
- 기존 node를 종료하고 node index3.js로 새로 실행
- 실행 화면
 - <http://localhost:3000/webapi>



```
{"response":{"header":{"resultCode":"00","resultMsg":"NORMAL_SERVICE"},"body":{"dataTy
[{"baseDate":"20200513","baseTime":"0630","category":"LGT","fcstDate":"20200513","fcstTi
{"baseDate":"20200513","baseTime":"0630","category":"LGT","fcstDate":"20200513","fcstTir
{"baseDate":"20200513","baseTime":"0630","category":"LGT","fcstDate":"20200513","fcstTir
{"baseDate":"20200513","baseTime":"0630","category":"LGT","fcstDate":"20200513","fcstTir
{"baseDate":"20200513","baseTime":"0630","category":"LGT","fcstDate":"20200513","fcstTir
{"baseDate":"20200513","baseTime":"0630","category":"LGT","fcstDate":"20200513","fcstTir
{"baseDate":"20200513","baseTime":"0630","category":"PTY","fcstDate":"20200513","fcstTir
{"baseDate":"20200513","baseTime":"0630","category":"PTY","fcstDate":"20200513","fcstTir
{"baseDate":"20200513","baseTime":"0630","category":"PTY","fcstDate":"20200513","fcstTir
{"baseDate":"20200513","baseTime":"0630","category":"PTY","fcstDate":"20200513","fcstTir
```


Step-5: Open API 호출 – Node.js

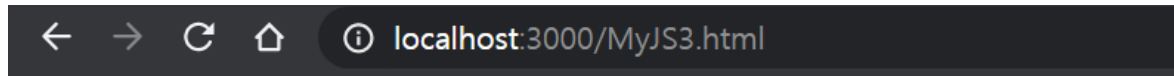
- MyJS3.html를 MyJS2.html을 복사하여 생성
- GET /webapi 를 호출하도록 수정



```
5 <p>Click the button to display the date.</p>
6
7 <button onclick="displayResponse()">The time is?</button>
8
9 <script>
10     var xhr = new XMLHttpRequest();
11     xhr.withCredentials = true;
12     xhr.addEventListener("readystatechange", function() {
13         if(this.readyState === 4) {
14             console.log(this.responseText);
15             document.getElementById("demo").innerHTML = this.responseText;
16         }
17     });
18 > function displayDate() { ...
21     function displayResponse(){
22         xhr.open("GET", "http://localhost:3000/webapi");
23         xhr.send();
24     }
25 </script>
```

Step-5: Open API 호출 – Node.js

- index3.js는 변화가 없으므로 재실행할 필요는 없음
- Step-3과 Step-5의 OPEN API 사용법 차이에 대해 생각해볼 것



Click the button to display the date.

The time is?

```
{
  "response": {
    "header": {
      "resultCode": "00",
      "resultMsg": "NORMAL_SERVICE"
    },
    "data": [
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "LGT",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "LGT",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "LGT",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "LGT",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "LGT",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "LGT",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "PTY",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "PTY",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "PTY",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      },
      {
        "baseDate": "20200512",
        "baseTime": "0630",
        "category": "PTY",
        "fcstDate": "20200512",
        "fcstTime": "0630",
        "fcstValue": "0.0",
        "unit": "mm"
      }
    ]
  }
}
```