

Node. JS and HTTP Open API

Introduction to Internet and Web



부산대학교 정보·의생명공학대학
정보컴퓨터공학부



Table of 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 URL is `pusan.ac.kr/kor/Main.do`. The page features a banner for the university's COVID-19 status and a large image of a statue with the text "학문의 날개를 펼치다" (Spread the wings of learning). Below the banner, there are sections for "Come with PNU!" and "PNU FOCUS".

The browser's 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://mmw.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=0000VTQ4Uezcov8MVXs03tfeK-1-1; _gat_gtag_UA_75737770_2=1`

NodeJS 설치

❖ NodeJS 홈 페이지 : <https://nodejs.org/en/>

❖ 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

```
C:\> 명령 프롬프트

Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\최성기>node -v
v16.15.0

C:\Users\최성기>npm -v
8.9.0

C:\Users\최성기>
```

ExpressJS 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:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\myapp\package.json:

{
  "name": "myapp",
  "version": "1.0.0",
  "main": "index.js",
  . . .

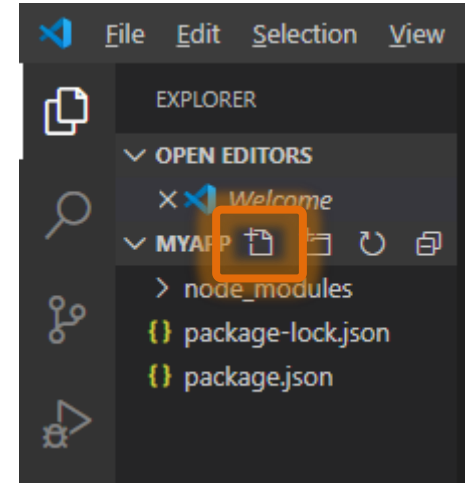
Is this OK? (yes)

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

Hello World – Index.JS 작성

❖ Express를 실행하기 위한 기본 JavaScript File – Index.JS 작성

- VS Code를 실행 시키고 /myapp 폴더를 Open하라
- MyApp 폴더에 Index.JS 파일을 추가하라
- [Hello World Example](#) Web Page 또는 아래를 참고하여 index.js 작성하라
 - 또는 Download 받은 Sources 내의 myapp 폴더 내의 index.js를 /myapp 으로 복사하라



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}`))
```

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

Express 실행

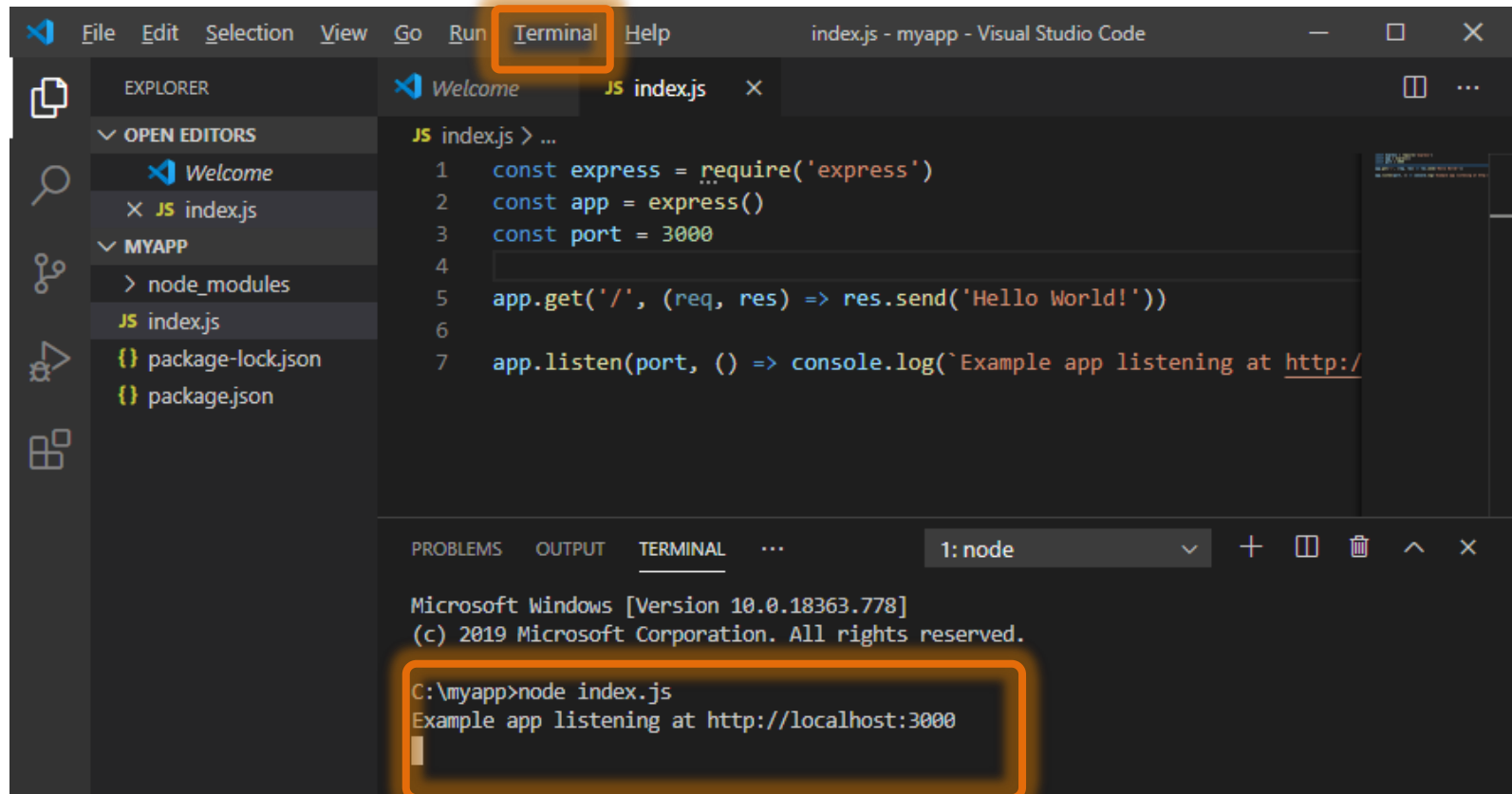
❖ Command 창에서 index.js를 인자로 node를 실행시켜라

```
C:\myapp>node index.js
```

```
Example app listening at http://localhost:3000
```

Command Prompt

- Command 창은 VS Code에서도 생성 가능



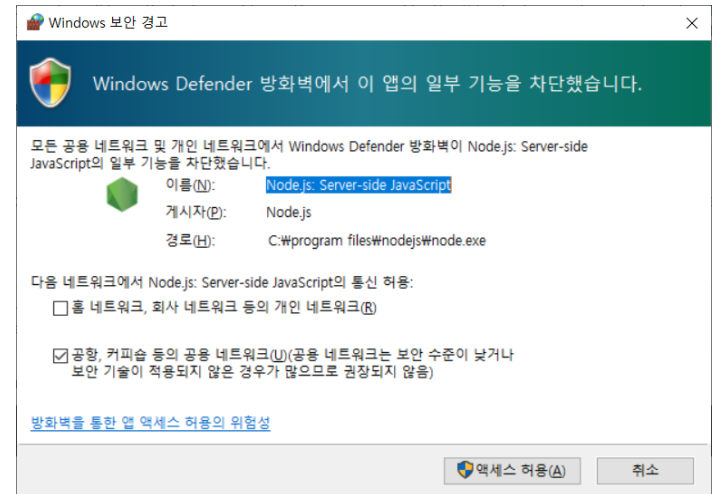
Index.JS 실행 확인

❖ Node JS 실행하면 보안 경고 메시지

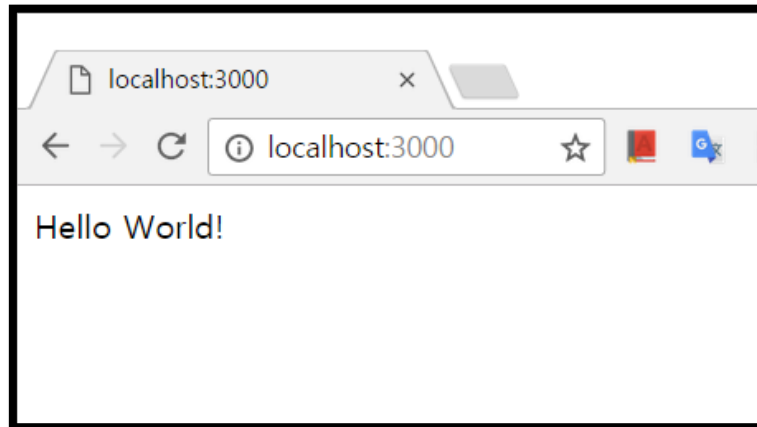
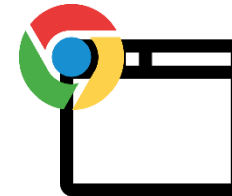
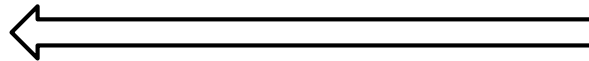
- → 액세스 허용
- 당신의 Computer에서 Express라는 Web 서버가 Port 번호 3000번에서 구동

❖ Browser 실행하고 당신의 Local Web 서버에 접속

- Hello World 출력



<http://localhost:3000/>



구동 중단 → Index.js 수정 → 재구동

❖ Index.js Code에서 'Hello World!' 문자열 찾아 수정

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}`))
```

❖ 수정한 Index.js를 반영하기 위해 현재 Web 서버 구동 중단

- Node를 실행 시킨 Terminal/Command 창에서 Ctrl-C를 입력하여 중단
 - Ctrl-C 입력하여도 바로 종료하지 않으면 또 다시 Ctrl-C를 입력
- 수정한 내용이 Index.js 저장되었는지를 확인하고 재구동하기
 - node index.js
- Browser로 접속하여 수정이 제대로 되었는지 확인하기

Step-1 : NodeJS – 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.js는 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 : NodeJS – Routing 수행 결과 Test

← → ↻ 🏠 ⓘ 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

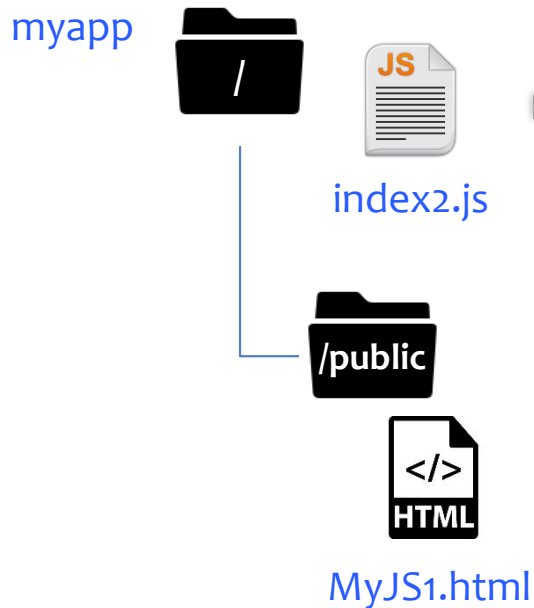
1 PUT Path1

PUT Method 테스트를 위해 Postman 활용

Step-2 : Static Files

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

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <p>Click the button to display the date.</p>
6
7 <button onclick="displayDate()">The time is?</button>
8
9 <script>
10   function displayDate() {
11     document.getElementById("demo").innerHTML = Date();
12   }
13 </script>
14
15 <p id="demo"></p>
16
17 </body>
18 </html>
```



```
index2.js
const express = require('express')
const app = express()
const port = 3000

app.use(express.static('public')) ← 코드추가

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

app.get('/Path1', function (req, res) {
  res.send("GET Path1");
})
```

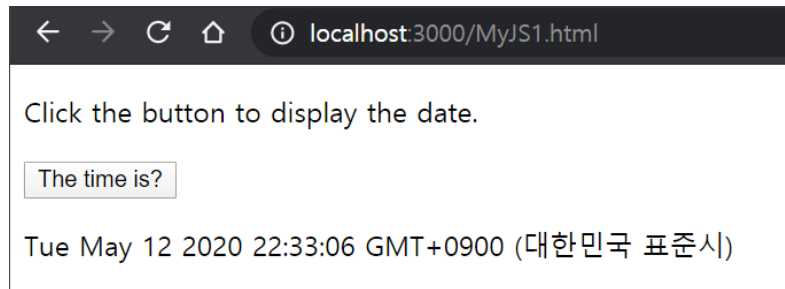
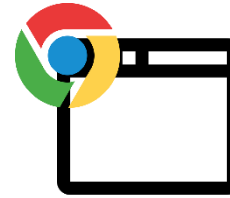
<http://expressjs.com/en/starter/static-files.html> 참고

Step-2 : Static Files

❖ Index2.js를 재구동하고 Static File 들이 제대로 서비스되는지 확인하라



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



MyJS1.html 은 어디서 실행되나?

Node JS 서버 vs Browser

Step-3 : JavaScript를 이용한 Open API 호출

❖ JavaScript를 이용한 공공데이터 기상청 초단기예보 서비스 API 호출

- Postman Code 생성 기능 활용
- JavaScript Code 생성 (실습 시기에 따라 base_date를 바꿔주어야 함.)

공공데이터-기상청-단기예보조회서비스 / 초단기예보조회

GET http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtFcst?s... Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> serviceKey	Vnrso4ayvW2S%2B26mo5JHOJpGzpXdp4H8...	공공데이터
<input checked="" type="checkbox"/> base_date	20220507	발표일자 Y
<input checked="" type="checkbox"/> base_time	1600	발표시각 H
<input checked="" type="checkbox"/> nx	98	예보지점 X
<input checked="" type="checkbox"/> ny	77	예보지점 Y
<input checked="" type="checkbox"/> dataType	JSON	요청자료형
Key	Value	Description

Body 200 OK 135 ms 660 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "response": {
```

Step-3 : Open API 호출

❖ XHR ([XMLHttpRequest](#)) 객체 활용

- 브라우저에서 지원하는 객체로 Http Request를 쉽게 수행할 수 있음

공... / 초

Save

GET http://apis.data.go.kr/1:...

Send

Params

KEY	VALUE
<input checked="" type="checkbox"/> serviceKey	Vnrso4ay
<input checked="" type="checkbox"/> base_date	2022050
<input checked="" type="checkbox"/> base_time	1600
<input checked="" type="checkbox"/> nx	98
<input checked="" type="checkbox"/> ny	77
<input checked="" type="checkbox"/> dataType	JSON
Key	Value

Body 200 OK 135 ms 660 B Save Response

Pretty JSON

Code snippet

JavaScript - XHR

```
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/
    VillageFcstInfoService_2.0/
    getUltraSrtFcst?
    serviceKey=Vnrso4ayvW2S%252B2
    6mo5JH0JpGzpXdp4H88RzhvTm0MoS
    VUKDsZT5983m0mC0gkZ3ktiBKAYUY
    mzJjFz0U%252FP%252BK6Q%253D%2
    53D&base_date=20220507&
    base_time=1600&nx=98&ny=77&
    dataTvne=JSON")
```

Code snippet

Type to filter

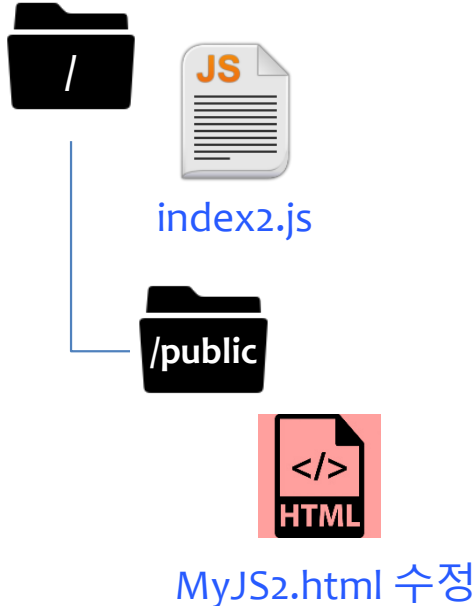
- C# - RestSharp
- cURL
- Dart - http
- Go - Native
- HTTP
- Java - OkHttp
- Java - Unirest
- JavaScript - Fetch
- JavaScript - jQuery
- JavaScript - XHR
- C - libcurl
- NodeJs - Axios
- NodeJs - Native
- NodeJs - Request
- NodeJs - Unirest
- Objective-C - NSURLSession
- OCaml - Cohttp
- PHP - cURL
- PHP - HTTP Request2

Bootcamp Runner Trash

Step-3 : Open API 호출

❖ Open API를 호출하는 MyJS2.html 작성

- MyJS1.html을 복사하고 아래와 같이 수정하여 MyJS2.html 작성



```
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() { ...
22
23     xhr.open("GET", "http://apis.data.go.kr/1360000/VilageFcst
24     xhr.setRequestHeader("Cookie", "JSESSIONID=doQlQfuFJBL1BCx
25
26     xhr.send();
27 }
28 </script>
29
30 <p id="demo"></p>
31
32 </body>
33 </html>
```

코드 변경

Postman
생성 코드

Step-3 : Open API 호출

- Index2.js는 변하지 않았으므로 중단/재구동은 필요 없음

❖ MyJS2.html을 Reload하여 결과 확인

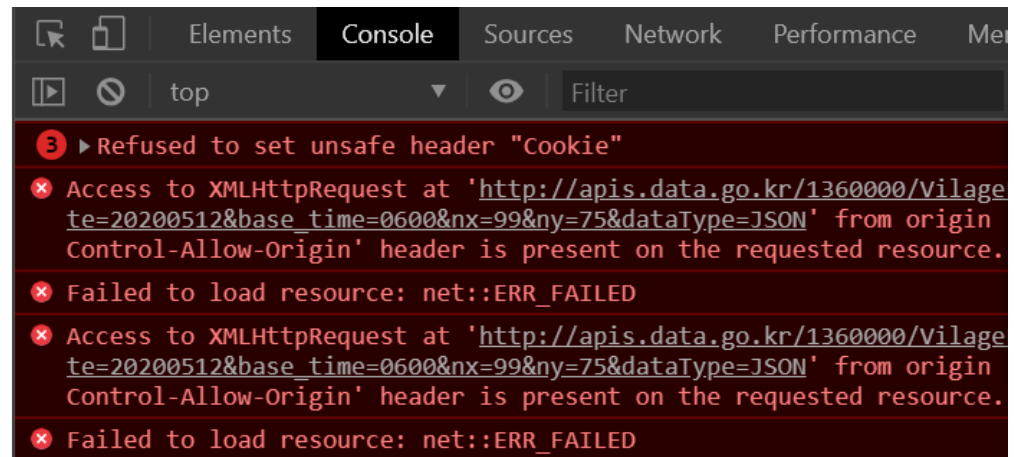
- 원하는 결과 값을 얻을 수 없음
- 브라우저의 개발자 도구를 활용하여 디버깅 정보 확인 (Console 확인)
 - Chrome - 도구 더보기 - 개발자 도구 (Ctrl - Shift - I 또는 F12)

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

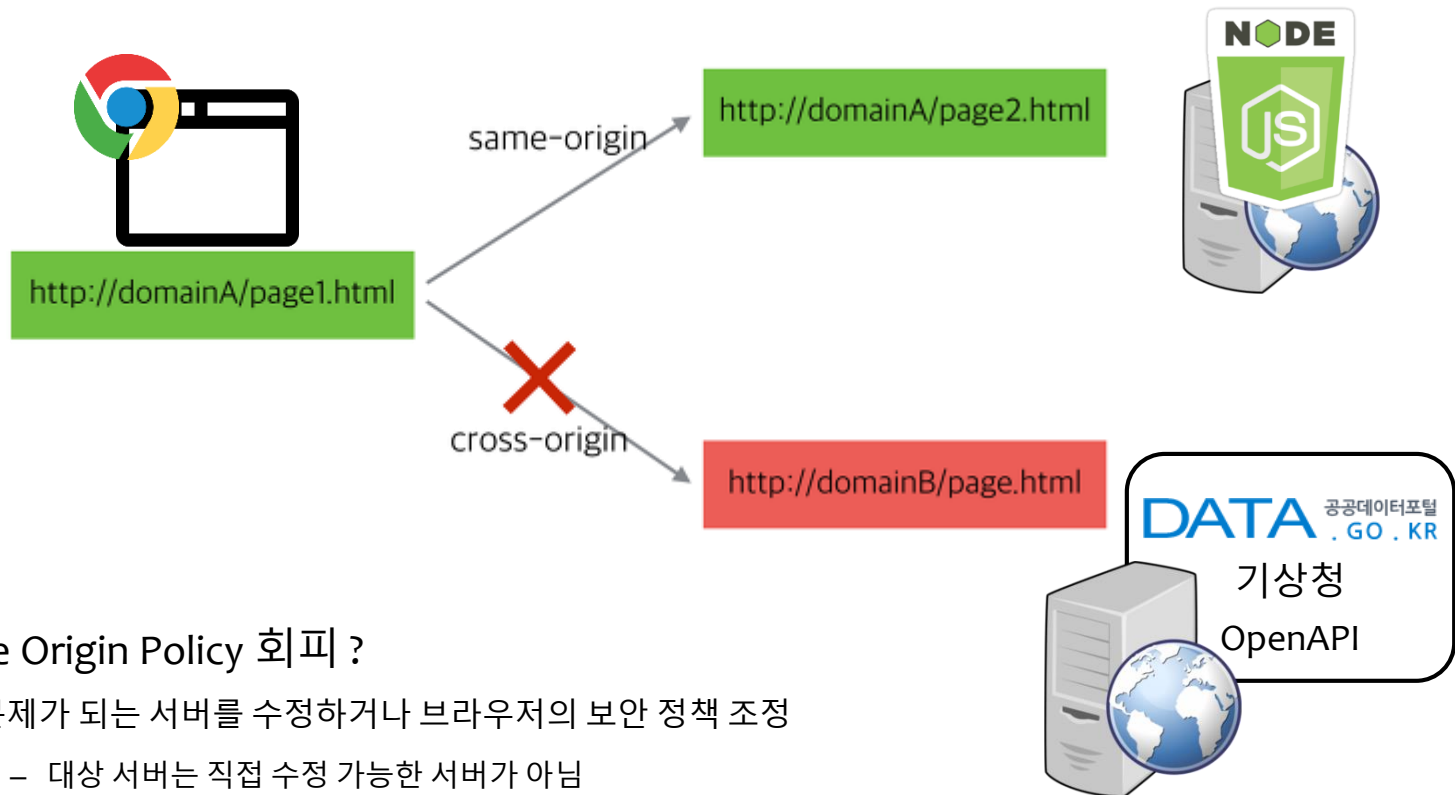


Step-3 : Open API 호출

❖ Same Origin Policy – 웹 응용 보안 모델

▪ <http://adrenal.tistory.com/16> 참고

- 자바스크립트(XMLHttpRequest)로 다른 웹페이지에 접근할 때는 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.



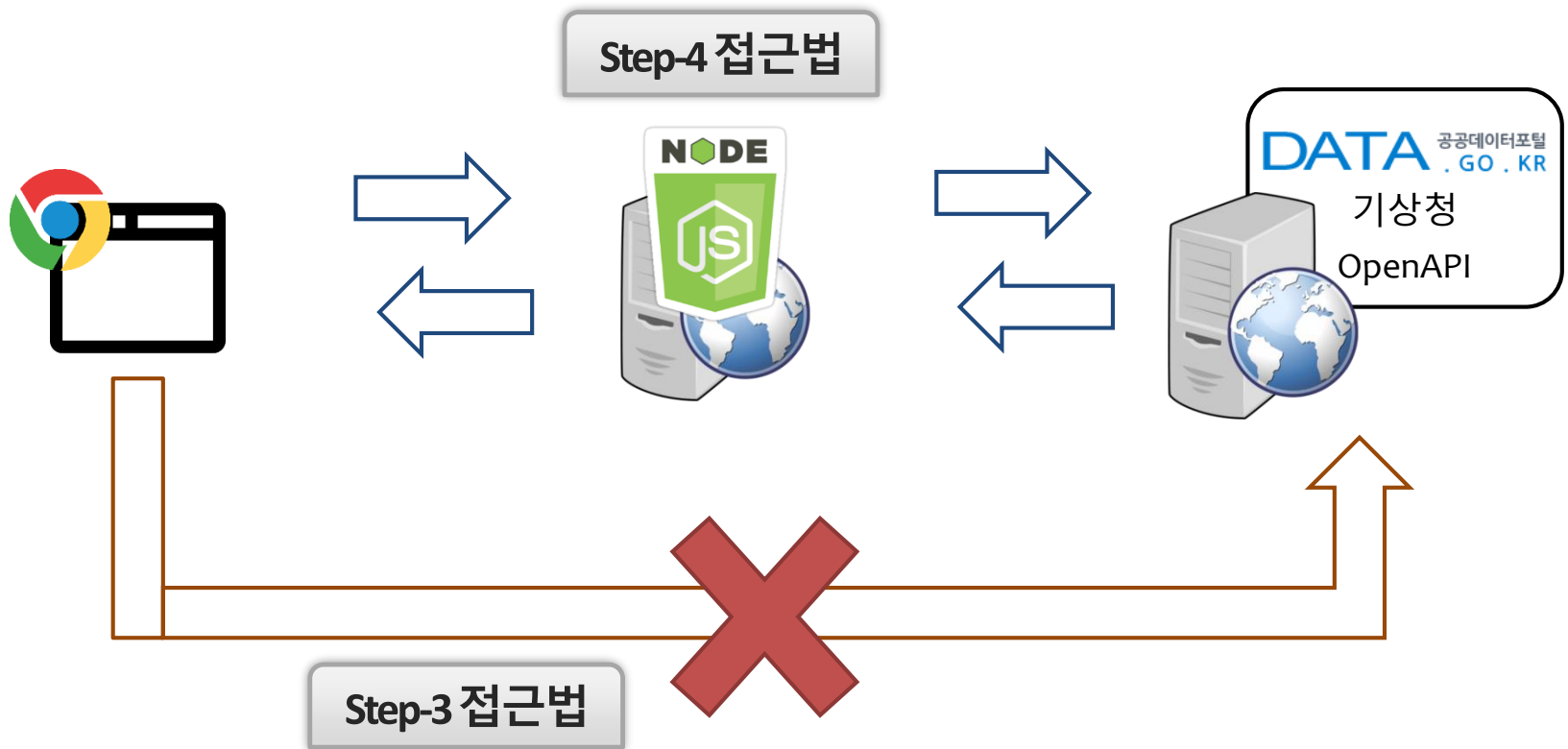
▪ Same Origin Policy 회피 ?

- 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 JS에서의 호출

❖ Client-Side JavaScript로 다른 도메인의 Open API 호출은 부적절함

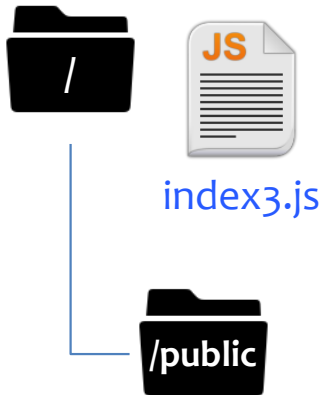
- Web 응용 보안 모델에 따라 제약이 발생
- Server Side Script로 다른 도메인의 Open API 호출 ?



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



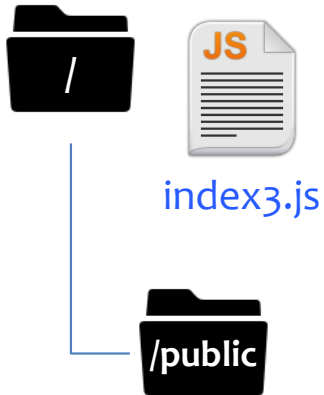
```
Code snippet
NodeJs - Request

1  var request = require('request');
2  var options = {
3    'method': 'GET',
4    'url': 'http://apis.data.go.kr/1360000/
      VillageFcstInfoService_2.0/getUltraSrtFcst?
      serviceKey=Vnrso4ayvW2S%2B26mo5JH0JpGzpXdp
      4H88RzhvTm0MoSVUKDsZT5983m0mC0gkZ3ktiBKAYU
      YmzJjfz0U%2FP%2BK6Q%3D%3D&
      base_date=20220507&base_time=1600&nx=98&
      ny=77&dataType=JSON',
5    'headers': {
6      'Cookie':
7        'JSESSIONID=3H0jHFUtpwMbhfAIwYBExJlip
          moFGvcBo25mEPyT2lpoox7ToxaYRZ0fmapNoC
          .amV1c19kb21haW4vbmV3c2t5Mw=='
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

index3.js



GET /webapi 에 대한 Handler 추가

Postman
생성 코드

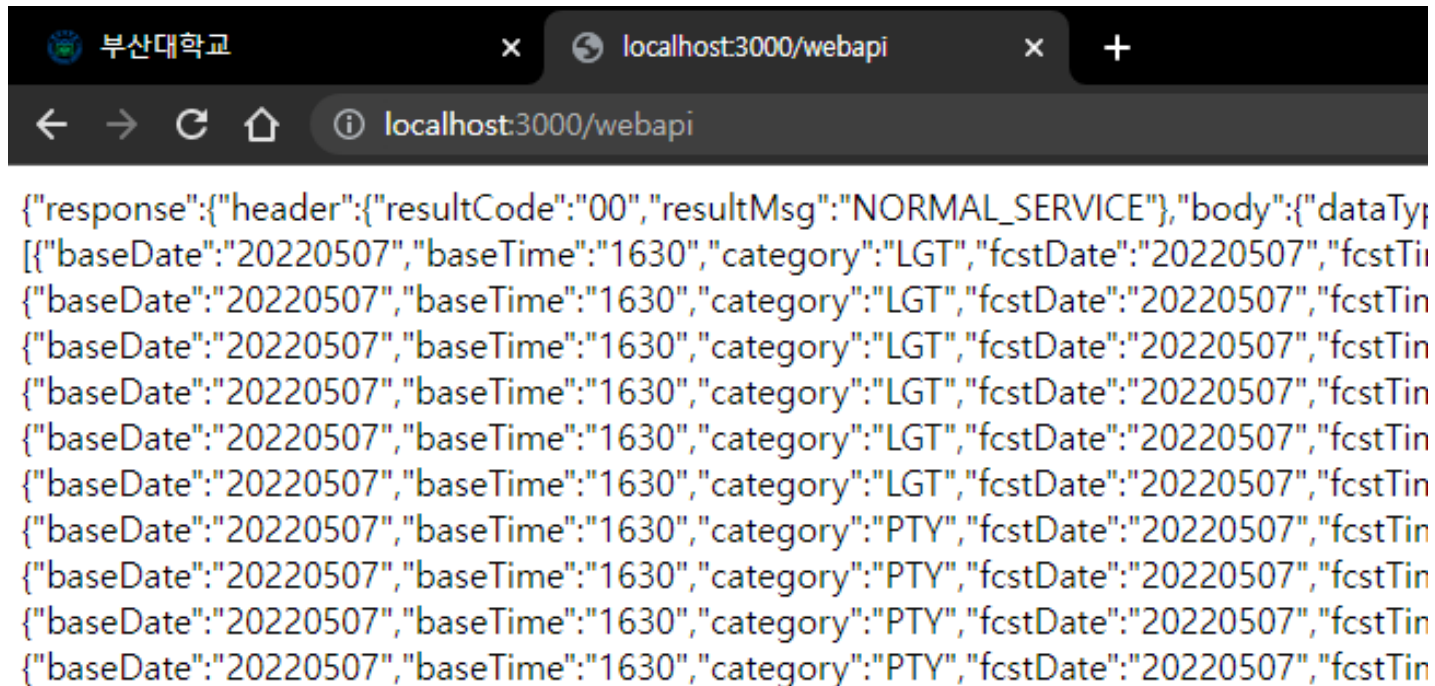
```
3  const port = 3000
4
5  app.use(express.static('public'))
6
7  app.get('/', (req, res) => res.send('Hello World!'))
8
9  app.get('/webapi', function(req, res){ ← /webapi
10     var request = require('request');
11     var options = {
12         'method': 'GET',
13         'url': 'http://apis.data.go.kr/1360000/VilageFc
14         'headers': {
15             'Cookie': 'JSESSIONID=OdqdDKkUkgChEbyI1DdZU0B
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) {
28     res.send('GET Path1');
```

Step-4 : Open API 호출/ Node JS

❖ 기존 node를 종료하고 node index3.js로 새로 실행

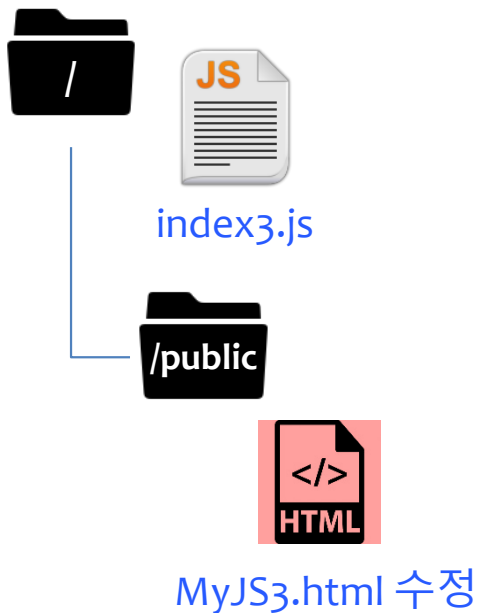
❖ 실행 화면

- <http://localhost:3000/webapi>



Step-5 : Open API 호출 - Node JS

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



```
4 <p>Click the button to display the date.</p>
5
6 <button onclick="displayResponse()">The time is?</button>
7
8 <script>
9 > function displayDate() { ...
12 function displayResponse() {
13     var xhr = new XMLHttpRequest();
14     xhr.withCredentials = true;
15     xhr.addEventListener("readystatechange", function() {
16         if(this.readyState === 4) {
17             console.log(this.responseText);
18             코드추가 → document.getElementById("demo").innerHTML = this.responseText;
19         }
20     });
21     xhr.open("GET", "http://localhost:3000/webapi"); ← 주소변경
22     xhr.send();
23 }
24 </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

