

# 9주차 실습

## 1 개요

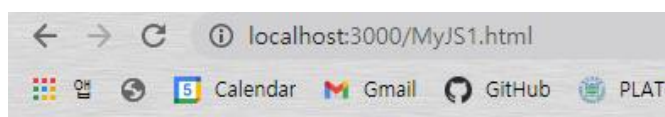
Static files가 잘 서비스되는지 확인하고 node.js를 이용하여 open API를 호출해보는 과제입니다.

## 2 Static files

### 2.1 Ass9\_01.js (코드 모두 & 중요 코드 표시)

```
JS ass9_01.js > app.get('/Path1') callback
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  app.use(express.static('public')) // 'public'
6
7  app.get('/', (req, res) => res.send('Hello World!'))
8
9  app.get('/Path1', function (req, res) { // ../Path1 현재 시각 표시
10     res.send("GET Path1");
11 })
12 app.get('/Path2', function(req,res) {
13     res.send("GET Path2 : " + Date());
14 })
15 app.put('/Path1', function (req, res) {
16     res.send("PUT Path1");
17 })
18
19
20 app.listen(port, () => console.log(`Example app listening at http://localhost:${port}`))
```

### 2.2 서비스 확인 (주소 포함)



Click the button to display the date.

The time is?

Wed May 05 2021 19:22:58 GMT+0900 (대한민국 표준시)

### 2.3 질문 : MyJS1.html이 실행 되는 곳은 ? (Node JS 서버 vs Browser)

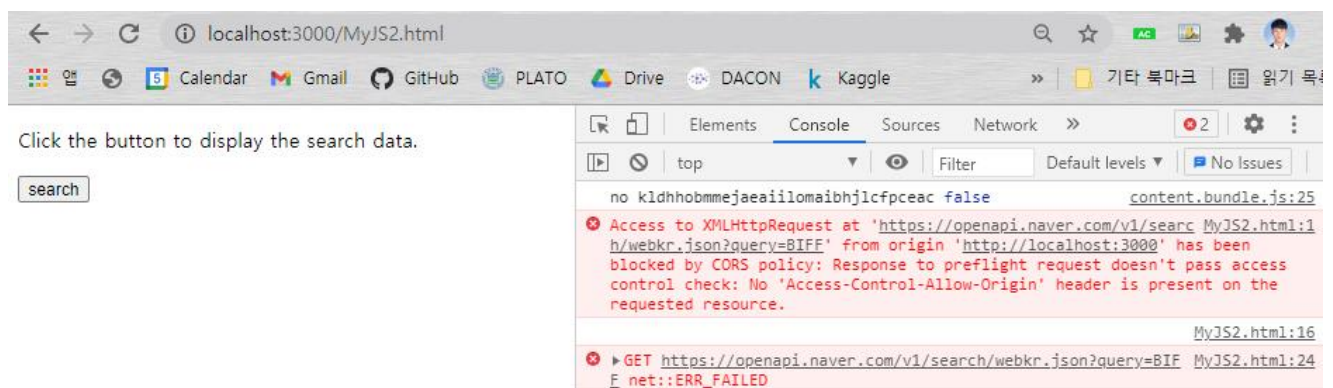
Node JS 서버

### 3 Open API 호출 1

#### 3.1 MyJS2.html (코드 모두 & 중요 코드 표시)

```
public > MyJS2.html > html > body > script > displayResponse
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <p>Click the button to display the search data.</p>
6
7  <button onclick="displayResponse()">search</button> <!-- button 클릭 시 displayResponse() 수행-->
8  <!-- Id, Secret 이용하여 URL Open and Send-->
9  <script>
10     function displayResponse(){
11         var xhr = new XMLHttpRequest();
12         xhr.withCredentials = true;
13
14         xhr.addEventListener("readystatechange", function() {
15             if(this.readyState === 4) {
16                 console.log(this.responseText);
17             }
18         });
19
20         xhr.open("GET", "https://openapi.naver.com/v1/search/webkr.json?query=BIFF");
21         xhr.setRequestHeader("X-Naver-Client-Id", "XXXXXXXXXXXX");
22         xhr.setRequestHeader("X-Naver-Client-Secret", "XXXXXXXXXXXX");
23
24         xhr.send();
25     }
26
27 </script>
28
29 <p id="demo"></p>
30
31 </body>
32 </html>
```

#### 3.2 서비스 확인 ((주소 포함) & f12 console 값 포함)



### 4 Open API 호출 2

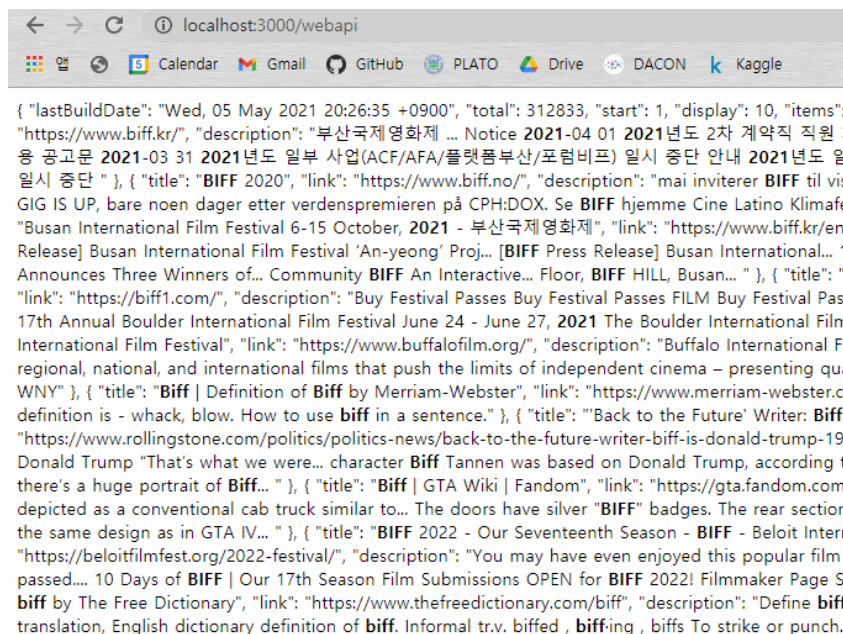
#### 4.1 Ass09\_02.js 코드 (코드 모두 & 중요 코드 표시)

```

JS ass9_02.js > app.get('/webapi') callback > options > 'headers'
1  const express = require('express')
2  const app = express()
3  const port = 3000
4
5  app.use(express.static('public')) // 'public'
6  app.get('/', (req, res) => res.send('Hello World!'))
7
8  app.get('/webapi', function (req, res) { // .../webapi 으로 들어가게 되면
9      var request = require('request');
10     var options = {
11         'method': 'GET',
12         'url': 'https://openapi.naver.com/v1/search/webkr.json?query=BIFF',
13         'headers': {
14             'X-Naver-Client-Id': '6uQiYgDu74bXPedIRSa3',
15             'X-Naver-Client-Secret': 'ZtQVNMT7U6'
16         }
17     };
18     request(options, function (error, response) { // error가 발생하지 않은 경우 수행
19         if (error) throw new Error(error);
20         console.log(response.body);
21         res.send(response.body); // response.body에 해당하는 것을 response 응답으로 전달.
22     });
23
24 })
25
26 app.get('/Path2', function(req,res) {
27     res.send("GET Path2 : " + Date());
28 })
29 app.put('/Path1', function (req, res) {
30     res.send("PUT Path1");
31 })
32
33
34 app.listen(port, () => console.log(`Example app listening at http://localhost:${port}`))

```

## 4.2 서비스 확인 (주소 포함)



## 5 Open API 호출 3

### 5.1 MyJS3.html 코드 (코드 모두 & 중요 코드 표시)

```

public > MyJS3.html > html > body > script > xhr.addEventListener("readystatechange") callback
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <p>Click the button to display the search data.</p>
6
7 <button onclick="displayResponse()">search</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
19     function displayDate(){
20         document.getElementById("demo").innerHTML = Date();
21     }
22     function displayResponse(){
23         xhr.open("GET", "http://localhost:3000/webapi"); // http://localhost:3000/webapi 에서 가져옴.
24         xhr.send();
25     }
26 </script>
27
28 <p id="demo"></p>
29
30 </body>
31 </html>
32







```

## 5.2 서비스 확인 (주소 포함)

The screenshot shows a web browser at `localhost:3000/MyJS3.html`. The page displays the text "Click the button to display the search data." and a "search" button. Below the button, there is a large block of JSON data representing search results. The console on the right shows the raw JSON response, which includes details about a BIFF (Back to the Future International Film Festival) event, such as dates, locations, and descriptions.

## 6 Open API 호출 비교

아래 표의 아이콘을 사용하여 각 open api 호출의 flow chart를 그리세요.

					
브라우저	Node JS 서버	Open API 서버	폴더	JS 파일	Html 파일

## 6.1 Open api 호출 1

JS 파일 구동 -> Browser에 URL 입력 -> Node JS 서버 -> 폴더 -> HTML 파일

## 6.2 Open api 호출 2

JS 파일 구동 -> Browser에 URL 입력 -> Node JS 서버 -> Open API 서버 -> 폴더  
-> HTML 파일

## 6.3 Open api 호출 3

JS 파일 구동 -> Browser에 URL 입력 -> Node JS 서버 -> Open API 서버 -> 폴더  
-> HTML 파일