

컴퓨터 네트워크

5장

5.1 CRUD

5.2 Create

5.3 Read

5.4 Update

5.5 Delete

5.1 CRUD

5.1 CRUD

1. CRUD

» Create, Read, Update, Delete

» REST API(Representational State Transfer)

- 서버의 자원을 정의하고 자원에 대한 주소를 지정하는 방법
- /user이면 사용자 정보에 관한 정보를 요청하는 것
- /post면 게시글에 관련된 자원을 요청하는 것

» HTTP 요청 메서드

- GET: 서버 자원을 가져오려고 할 때 사용
- POST: 서버에 자원을 새로 등록하고자 할 때 사용
- PUT: 서버의 자원을 요청에 들어있는 자원으로 치환하고자할 때 사용
- PATCH: 서버 자원의 일부만 수정하고자 할 때 사용
- DELETE: 서버의 자원을 삭제하고자할 때 사용

Create

Read

Update

Delete

5.2 Create

5.2 Create

1. 동적인 웹 페이지

» 기본적인 웹 페이지 제작

- 메인 페이지와 하이퍼링크로 연결된 목록 생성

```
1  <!DOCTYPE html>
2  <head>
3  |   <meta charset="UTF-8">|
4  |   <title>Web-Network</title>
5  </head>
6  <body>
7  |   <h1><a href="main.html">WEB-Network</a></h1>
8  |   <ol>
9  |   |   <li><a href="Chiikawa.html">치이카와</a></li>
10 |   |   <li><a href="pokemon.html">포켓몬</a></li>
11 |   </ol>
12 </body>
13 </html>
```

5.2 Create

1. 동적인 웹 페이지

» 기본적인 웹 페이지 제작

- 메인 페이지와 하이퍼링크로 연결된 목록 생성

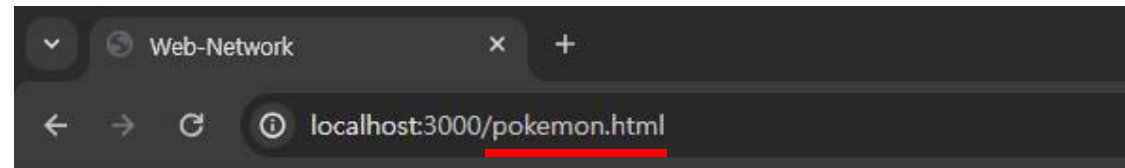
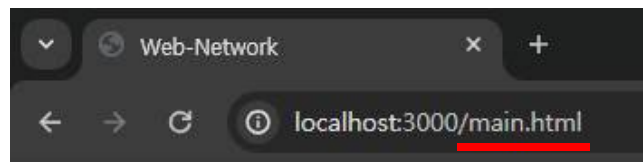
```
1  <!DOCTYPE html>
2  <head>
3      <meta charset="UTF-8">
4      <title>Web-Network</title>
5  </head>
6  <body>
7      <h1><a href="main.html">WEB-Network</a></h1>
8      <ol>
9          <li><a href="Chiikawa.html">치이카와</a></li>
10         <li><a href="pokemon.html">포켓몬</a></li>
11     </ol>
12     <p>
13         헤헤 메타몽 귀엽다
14     </p>
15     
16
17
18 </body>
19 </html>
```

5.2 Create

1. 동적인 웹 페이지

» 기본적인 웹 페이지 제작

- 메인 페이지와 하이퍼링크로 연결된 목록 생성



WEB-Network

1. [치이카와](#)
2. [포켓몬](#)

WEB-Network

1. [치이카와](#)
2. [포켓몬](#)

헤헤 메타몽 귀엽다



5.2 Create

1. 동적인 웹 페이지

» 요청 url에 따라 응답하기

- 사용자가 요청한 url이 전달되면서 디렉토리에서 파일을 불러옴

```
console.log(__dirname+url);  
res.end(fs.readFileSync(__dirname+url));
```

localhost:3000/pokemon.html

C:\Users\USER\ 개인정보 ☺ crud/main.html
C:\Users\USER\ crud/pokemon.html

5.2 Create

1. 동적인 웹 페이지

» 쿼리 스트링을 통해 본문이 변경되는 웹 앱 제작

- 3장에서 배운 쿼리 스트링 응용
- 쿼리 스트링을 통해서 동적으로 화면 구성

```
1 var http = require('http');
2 var fs = require('fs');
3 var url = require('url');
4
5 var app = http.createServer(function(req, res){
6   var _url = req.url;
7   var queryData = url.parse(_url, true).query;
8   if(_url == '/'){
9     _url = '/main.html';
10  }
11  if(_url == '/favicon.ico'){
12    return res.writeHead(404);
13  }
14  res.writeHead(200, { 'Content-Type': 'text/html; charset=utf-8' });
15
```

```
16 //추가된 부분
17 var template = `
18 <!DOCTYPE html>
19 <html>
20 <head>
21   <meta charset="UTF-8">
22   <title>Web-${queryData.id}</title>
23 </head>
24 <body>
25   <h1><a href="main.html">Web-Network</a></h1>
26   <ol>
27     <li><a href="Chiikawa.html">치이카와</a></li>
28     <li><a href="pokemon.html">포켓몬</a></li>
29   </ol>
30   <p>
31     <h2>${queryData.id}</h2>
32     해해 메타몽 귀엽다
33   </p>
34 </body>
35 </html>
36 `;
37 res.end(template);
38
39 app.listen(3000);
40 app.on('error', (error) => {
41   console.error(error);
42 });
```

- localhost:3000/?id=



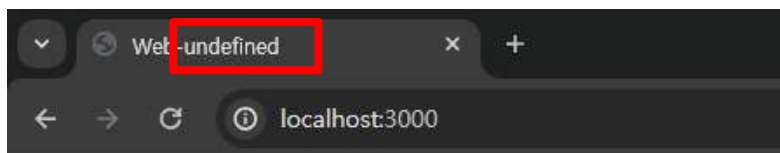
id 값에 따라 동적으로 변경

5.2 Create

1. 동적인 웹 페이지

» 쿼리 스트링을 통해 본문이 변경되는 웹 앱 제작

- localhost:3000/?id=

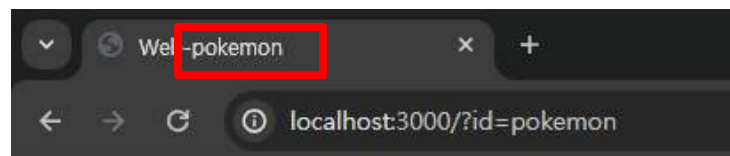


Web-Network

1. 치이카와
2. 포켓몬

undefined

헤헤 메타몽 귀엽다



Web-Network

1. 치이카와
2. 포켓몬

pokemon

헤헤 메타몽 귀엽다