네트워크서비스 프로토콜

12주차 1

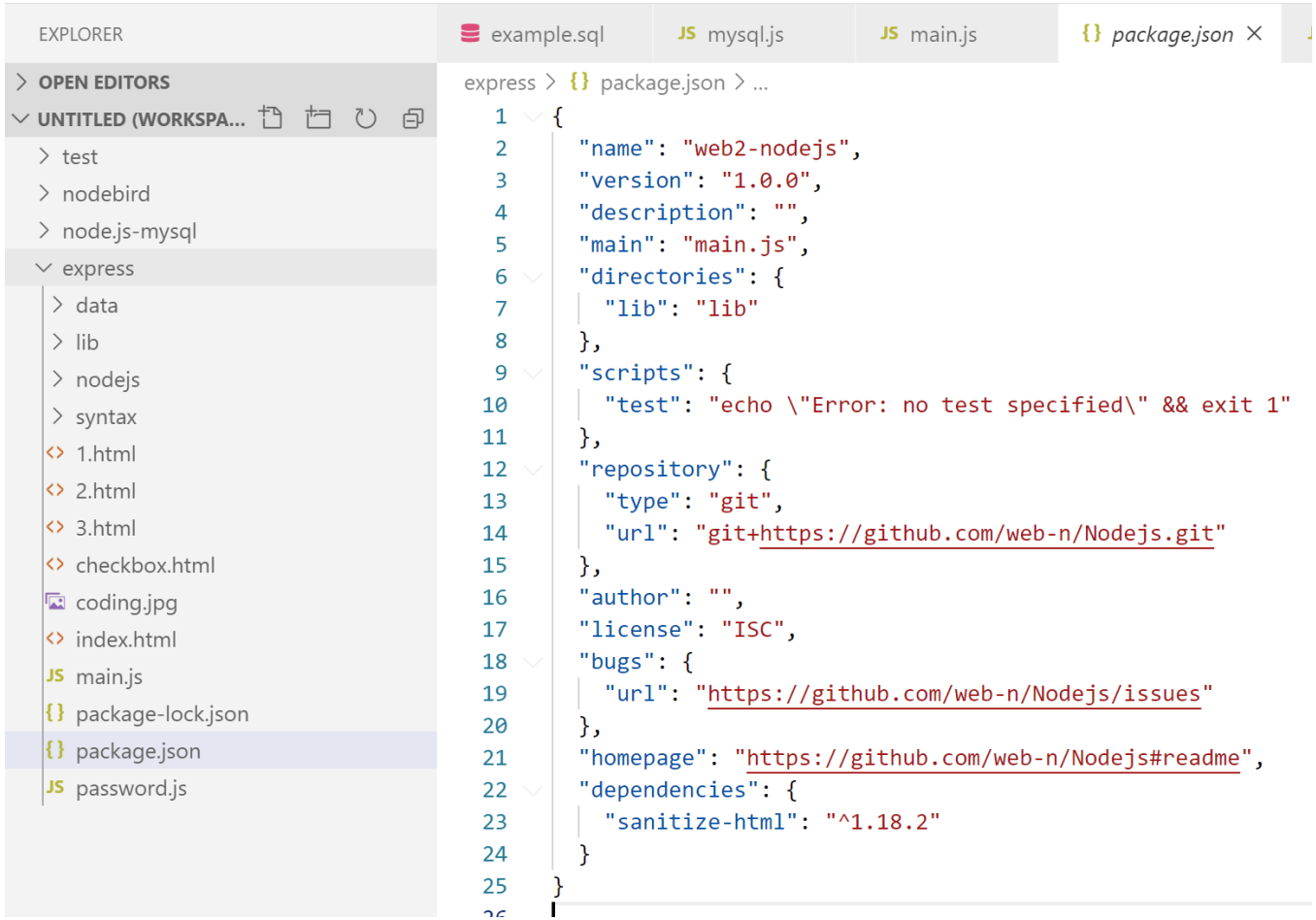
» 소프트웨어학부
» 김형균 교수

수업에 들어가며

- 지난 시간 복습
  - express 설치
- 오늘 학습할 내용
  - 라우팅 정의
  - 익스프레스 응답 메서드
  - 라우팅 실습
  - 홈페이지 Express 버전 구현
  - 상세보기 페이지 Express 버전 구현
  - 페이지 생성 기능 Express 버전 구현
  - 페이지 수정 기능 Express 버전 구현
  - Express 리다이렉션
  - 페이지 삭제 기능 Express 버전 구현
  - 삭제 기능 구현-오류해결

초기 설정

- E-캠퍼스 자료실에서 Nodejs-master.zip 다운로드 후 압축풀기
- 에디터에 프로젝트 추가하기



초기 설정

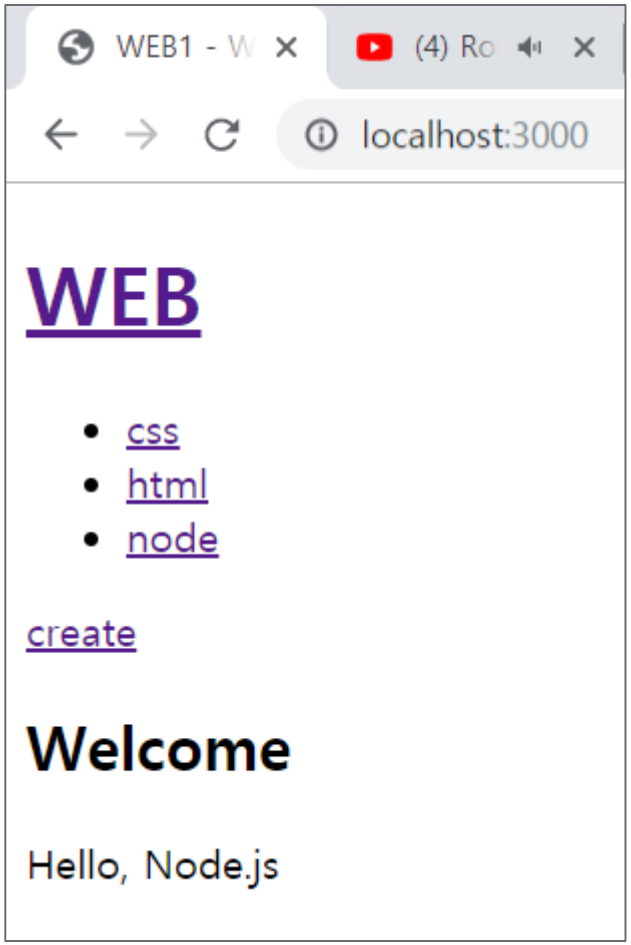
» 콘솔에서 npm install 실행

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\express\express> npm install
npm WARN web2-nodejs@1.0.0 No description

added 34 packages from 52 contributors and audited 48 packages in 0.917s
found 1 high severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details
PS C:\express\express>
```

초기 설정

» 웹서버 실행 확인



express 설치

- 프로젝트 폴더에서 설치
- 에디터 콘솔에서 다음과 같이 명령

PS C:₩express₩express> npm install express --save

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

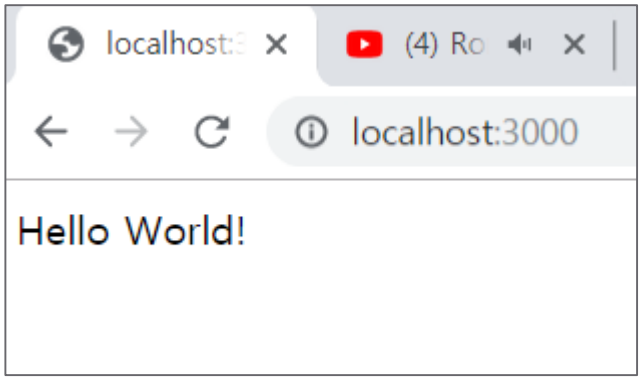
```
found 1 high severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details
PS C:\express\express> npm install express --save
npm WARN web2-nodejs@1.0.0 No description

+ express@4.17.1
added 48 packages from 36 contributors and audited 174 packages in 2.033s
found 1 high severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details
PS C:\express\express>
```

Hello World

Node.js 교과서

» main.js 기존 코드 전체 주석 처리 후 express 코드 입력

```
1  const express = require('express')
2  const app = express()
3
4  app.get('/', (req, res) => res.send('Hello World!'))
5
6  app.listen(3000, () => console.log('Example app listening on port 3000!'))
7
8  /*
9  var http = require('http');
```



라우팅 정의

- 클라이언트가 서버로 접속할때는 특정한 URL를 통해 접속한다.
- 서버에서는 이 URL에 해당하는 자원을 클라이언트로 보내준다. 혹은 POST요청일 경우는 자원을 만들기도 한다.
- 이러한 클라이언트 요청을 위한 URL 스키마를 라우트라고 한다.
- 서버에서는 라우팅 작업을 통해 클라이언트와 통신의 인터페이스를 제공해 준다.
- 익스프레스에서 중요한 것 중 하나가 이 라우팅 모듈이다.

라우팅 정의

- 기본적인 라우팅 설정 코드는 아래와 같다.

```
1  const express = require('express')
2  const app = express()
3
4  app.get('/', (req, res) => res.send('Hello World!'))
```

- 익스프레스 객체를 담고있는 app변수는 HTTP 메쏘드 명에 해당하는 함수를 가지고 있다.
  - app.post()
  - app.get()
  - app.put()
  - app.delete()
- 각 함수의 첫번째 파라매터에는 서버자원을 가리키는 URI 문자열을 지정한다.
- 두번째 파라매터는 라우팅 로직 함수를 콜백 형태로 구현한다.
- 즉 설정한 URI의 메소드로 요청이 들어오면 두번째 파라매터에 구현한 함수가 동작하는 것이다.

익스프레스 응답 메서드

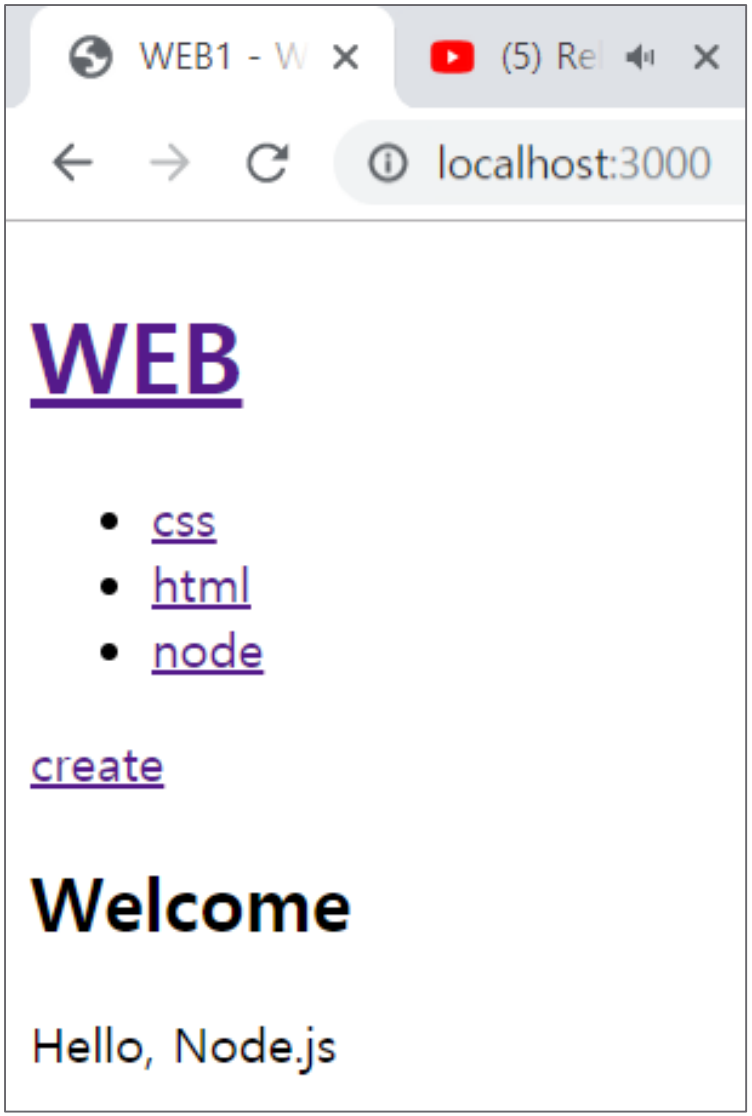
- 익스프레스가 메서드를 추가함
  - res.send(버퍼 또는 문자열 또는 HTML 또는 JSON): 기본 응답 메서드
  - res.sendFile(파일 경로): 파일 전송
  - res.json(JSON 데이터): 제이슨 데이터 응답
  - res.redirect(주소): 리다이렉트
  - res.render('템플릿 파일 경로', { 변수 }): 템플릿 엔진 렌더링

라우팅 실습

express > JS main.js > ...

```
1   var express = require('express')
2   var app = express()
3
4   app.get('/', function(req, res) {
5     return res.send('/');
6   });
7
8   app.get('/page', function(req, res) {
9     return res.send('/page');
10  });
11
12  app.listen(3000, function() {
13    console.log('Example app listening on port 3000!')
14  });
```

홈페이지 Express 버전 구현



홈페이지 Express 버전 구현

» 기존 코드 확인

```
var app = http.createServer(function(request,response){
    var _url = request.url;
    var queryData = url.parse(_url, true).query;
    var pathname = url.parse(_url, true).pathname;
    if(pathname === '/'){
      if(queryData.id === undefined){
        fs.readdir('./data', function(error, filelist){
          var title = 'Welcome';
          var description = 'Hello, Node.js';
          var list = template.list(filelist);
          var html = template.HTML(title, list,
            `<h2>${title}</h2>${description}`,
            `<a href="/create">create</a>`
          );
          response.writeHead(200);
          response.end(html);
        });
```

라우팅 처리

본문에서 사용

응답 메서드

express > JS main.js > ...

```
1   var express = require('express')
2   var app = express()
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17  });
```

상세보기 페이지 Express 버전 구현

- 상세보기 페이지를 Express 버전으로 변환
- 이 과정에서 query string을 사용하지 않는 pretty url(clean url, semantic url..)로 라우트 기능을 구현
- 경로 매개 변수(route parameters)
  - 경로 매개 변수는 URL에서 해당 위치에 지정된 값을 캡처하는 데 사용되는 이름이 지정된 URL 세그먼트
  - 캡처 된 값은 req.params경로에 지정된 라우트 매개 변수의 이름을 해당 키로 사용하여 오브젝트에 채워집니다 .

Route path: /users/:userId/books/:bookId

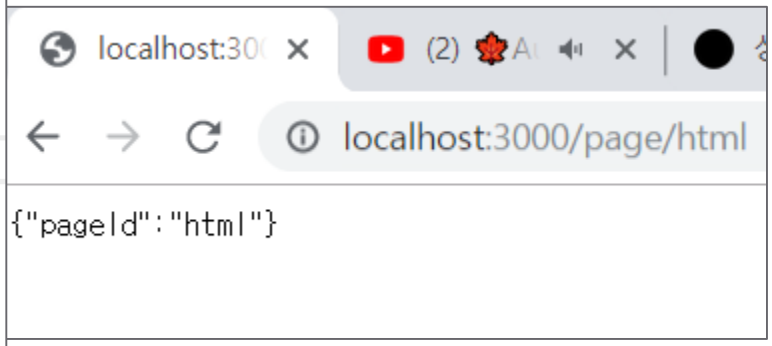
Request URL: http://localhost:3000/users/34/books/8989

req.params: { "userId": "34", "bookId": "8989" }

상세보기 페이지 Express 버전 구현

Node.js 교과서

```
app.get('/', function(request, response) {
  fs.readdir('./data', function(error, filelist){
    var title = 'Welcome';
    var description = 'Hello, Node.js';
    var list = template.list(filelist);
    var html = template.HTML(title, list,
      `<h2>${title}</h2>${description}`,
      `<a href="/create">create</a>`
    );
    response.send(html);
  });
});

app.get('/page/:pageId', function(request, response) {
  response.send(request.params);
});
```



기존 구현 방식 확인

```
47      if(pathname === '/'){
48        if(queryData.id === undefined){
49          fs.readdir('./data', function(error, filelist){…
59          });
60        } else {
61          fs.readdir('./data', function(error, filelist){
62            var filteredId = path.parse(queryData.id).base;
63            fs.readFile(`data/${filteredId}`, 'utf8', function(err, description){
64              var title = queryData.id;
65              var sanitizedTitle = sanitizeHtml(title);
66              var sanitizedDescription = sanitizeHtml(description, {
67                allowedTags:['h1']
68              });
69              var list = template.list(filelist);
70              var html = template.HTML(sanitizedTitle, list,
71                `<h2>${sanitizedTitle}</h2>${sanitizedDescription}`,
72                ` <a href="/create">create</a>
73                  <a href="/update?id=${sanitizedTitle}">update</a>
74                  <form action="delete_process" method="post">
75                    <input type="hidden" name="id" value="${sanitizedTitle}">
76                    <input type="submit" value="delete">
77                  </form>`
78              );
79              response.writeHead(200);
80              response.end(html);
81            });
82          });
83        }
```

변경 사항

```
21  app.get('/page/:pageId', function(request, response) {
22    fs.readdir('./data', function(error, filelist){
23      var filteredId = path.parse(request.params.pageId).base;
24      fs.readFile(`data/${filteredId}`, 'utf8', function(err, description){
25        var title = request.params.pageId;
26        var sanitizedTitle = sanitizeHtml(title);
27        var sanitizedDescription = sanitizeHtml(description, {
28          allowedTags:['h1']
29        });
30        var list = template.list(filelist);
31        var html = template.HTML(sanitizedTitle, list,
32          `<h2>${sanitizedTitle}</h2>${sanitizedDescription}`,
33          ` <a href="/create">create</a>
34            <a href="/update?id=${sanitizedTitle}">update</a>
35            <form action="delete_process" method="post">
36              <input type="hidden" name="id" value="${sanitizedTitle}">
37              <input type="submit" value="delete">
38            </form>`
39        );
40        response.send(html);
41      });
42    });
43  });
```

main.js | template.js

express > lib > template.js > ...

```
1   module.exports = {
2 >   HTML:function(title, list, body, control){ ···
18    },list:function(filelist){
19      var list = '<ul>';
20      var i = 0;
21      while(i < filelist.length){
22        list = list + `<li><a href="/page/${filelist[i]}">${filelist[i]}</a></li>`;
23        i = i + 1;
24      }
25      list = list+'</ul>';
26      return list;
27    }
28  }
```

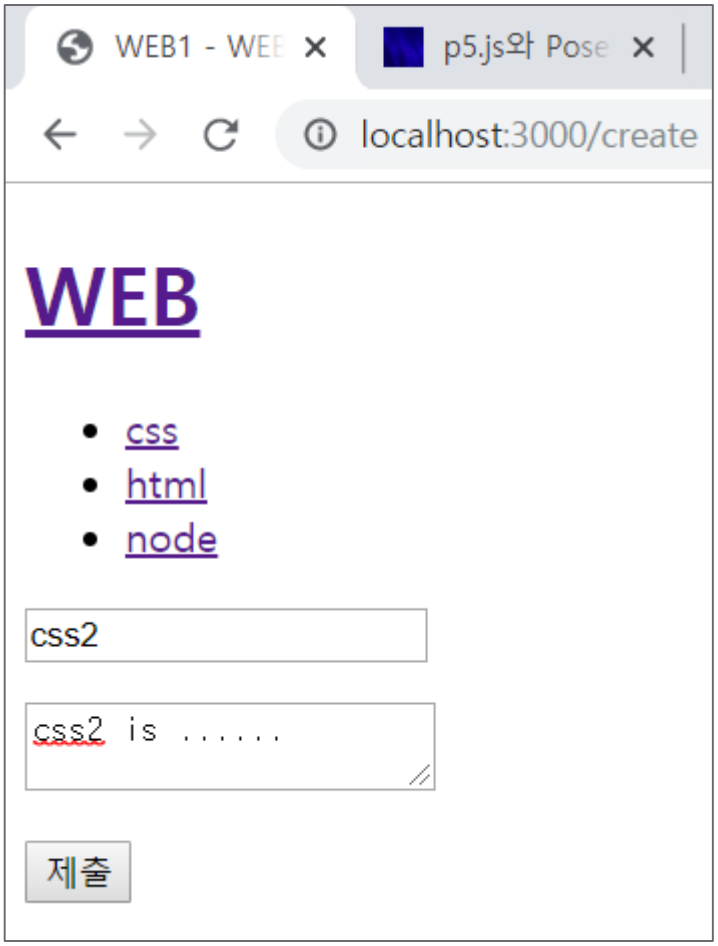
main.js | template.js ●

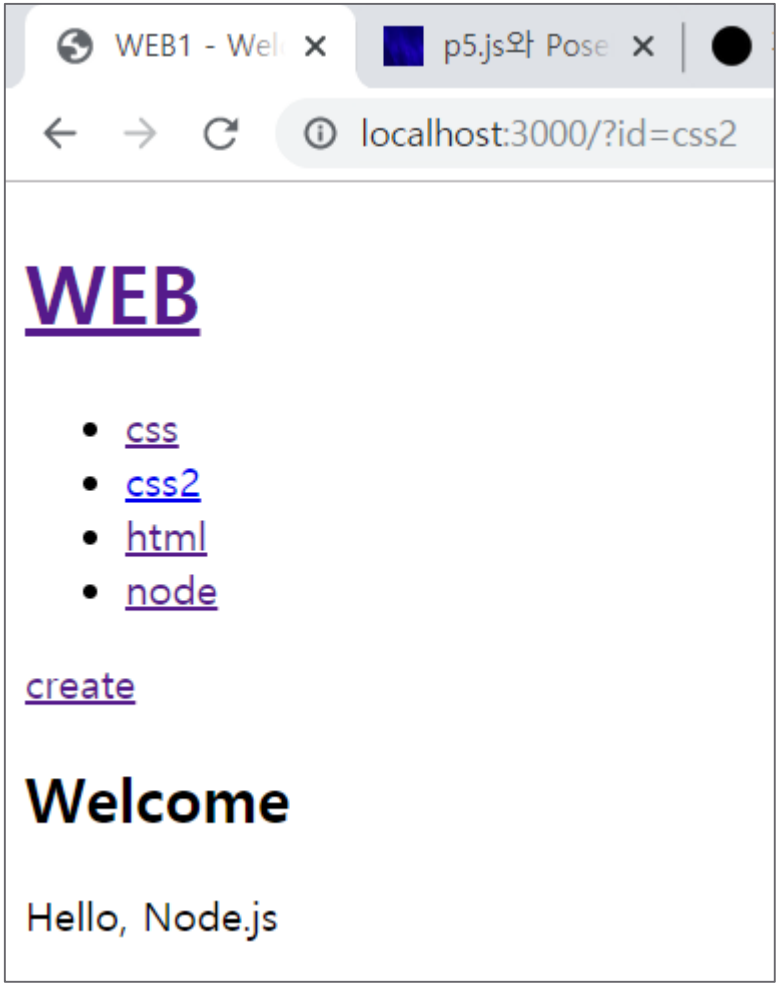
express > lib > template.js > ...

```
1   module.exports = {
2 >   HTML:function(title, list, body, control){ ···
18    },list:function(filelist){
19      var list = '<ul>';
20      var i = 0;
21      while(i < filelist.length){
22        list = list + `<li><a href="/?id=${filelist[i]}">${filelist[i]}</a></li>`;
23        i = i + 1;
24      }
25      list = list+'</ul>';
26      return list;
27    }
28  }
```

페이지 생성 기능 Express 버전 구현

» 페이지 생성 기능을 Express 버전으로 전환





페이지 생성 기능 Express 버전 구현


```
1   var express = require('express')
2   var app = express()
3   var fs = require('fs');
4   var template = require('./lib/template.js');
5   var path = require('path');
6
7   var sanitizeHtml = require('sanitize-html');
8
9   app.get('/', function(request, response) { ···
20  });
21
22  app.get('/page/:pageId', function(request, response) { ···
44  });
45
46  app.get('/create', function(request, response){···
63  });
64
65  app.post('/create_process', function(request, response){···
79  });
80
81  app.listen(3000, function() {
82    console.log('Example app listening on port 3000!')
83  });
```

페이지 생성 기능 Express 버전 구현


```
46  app.get('/create', function(request, response){
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63  });
```

페이지 생성 기능 Express 버전 구현


```
65  app.post('/create_process', function(request, response){
66
67
68
69
70
71
72
73
74
75
76
77
78
79  });
```

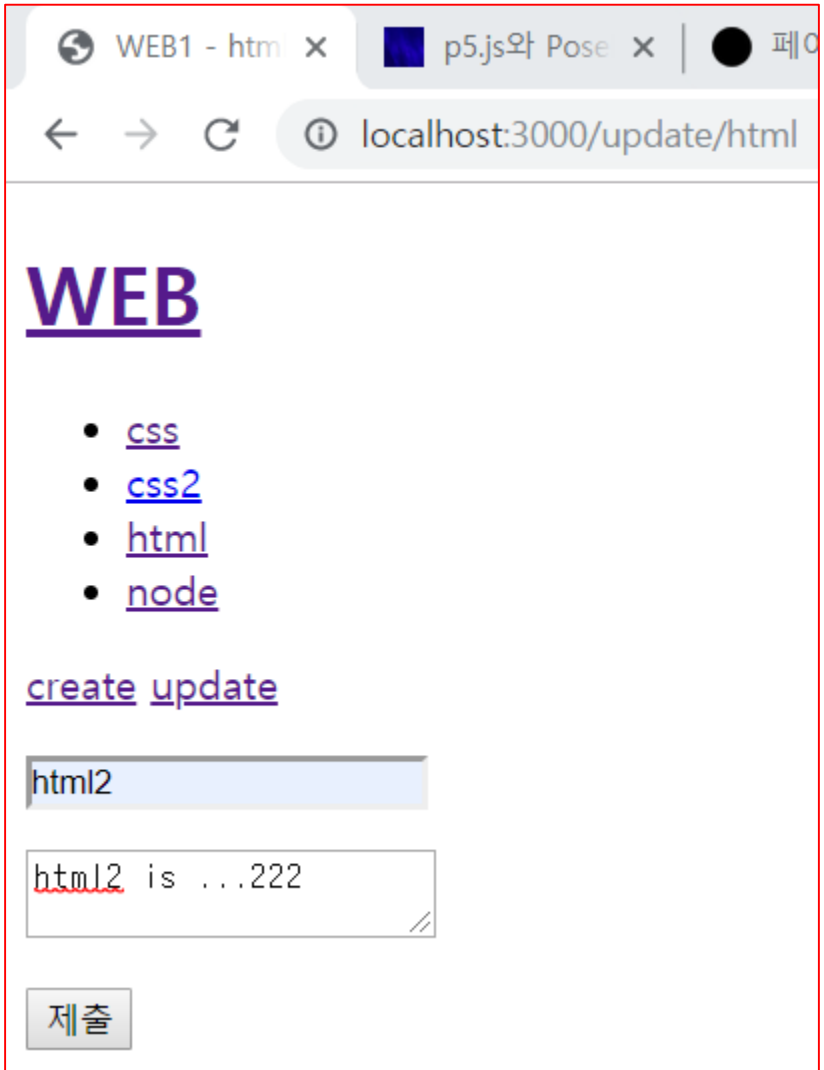
페이지 생성 기능 Express 버전 구현

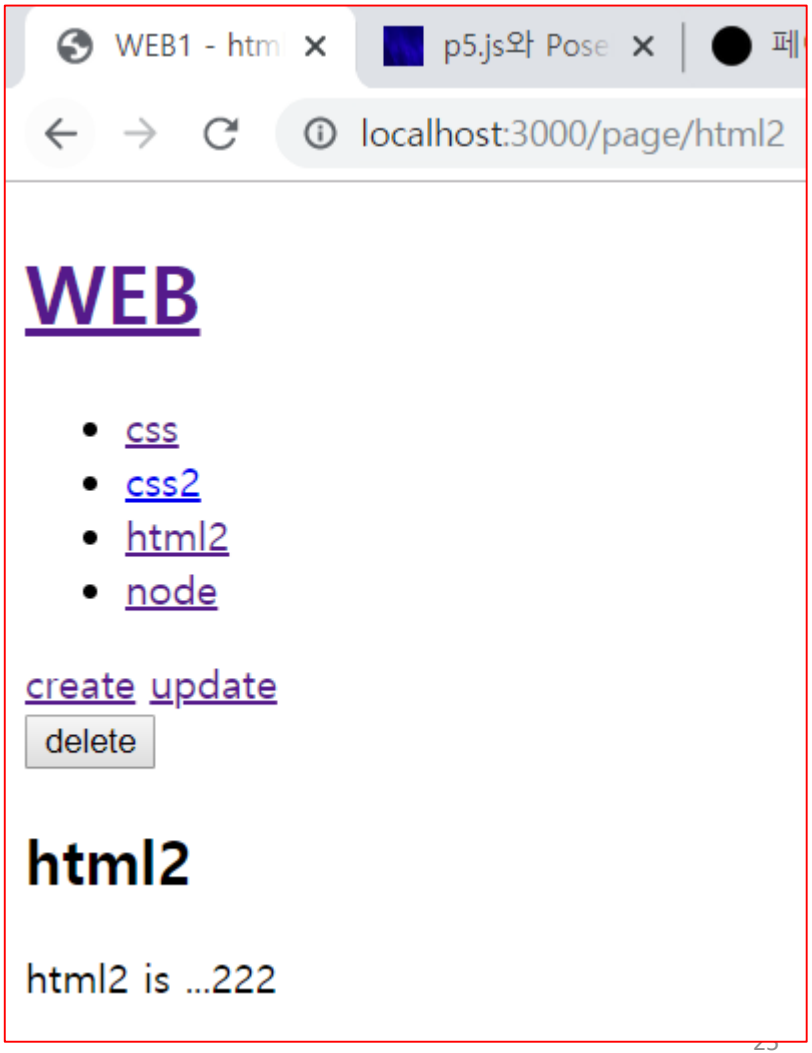

```
1   var express = require('express')
2   var app = express()
3   var fs = require('fs');
4   var template = require('./lib/template.js');
5   var path = require('path');
6
7   var sanitizeHtml = require('sanitize-html');
8
9   app.get('/', function(request, response) { ···
20  });
21
22  app.get('/page/:pageId', function(request, response) { ···
44  });
45
46  app.get('/create', function(request, response){···
63  });
64
65  app.post('/create_process', function(request, response){···
79  });
80
81  app.listen(3000, function() {
82    console.log('Example app listening on port 3000!')
83  });
```

페이지 수정 기능 Express 버전 구현

Node.js 교과서

» 페이지 수정 기능을 Express 버전으로 전환





페이지 수정 기능 Express 버전 구현


```
22  app.get('/page/:pageId', function(request, response) {
23    fs.readdir('./data', function(error, filelist){
24      var filteredId = path.parse(request.params.pageId).base;
25      fs.readFile(`data/${filteredId}`, 'utf8', function(err, description){
26        var title = request.params.pageId;
27        var sanitizedTitle = sanitizeHtml(title);
28        var sanitizedDescription = sanitizeHtml(description, {
29          allowedTags:['h1']
30        });
31        var list = template.list(filelist);
32
33
34
35
36
37
38
39
40
41
42        response.send(html);
43      });
44    });
45  });
```

페이지 수정 기능 Express 버전 구현


```
82  app.get('/update/:pageId', function(request, response){
```

페이지 수정 기능 Express 버전 구현


```
109  app.post('/update_process', function(request, response){
127    });
```

Express 리다이렉션

res.redirect ([상태,] 경로)

- HTTP 상태 코드에 해당하는 양의 정수 path로 지정된 지정된 URL에서 파생 된 URL로 리디렉션합니다 .
- res.redirect('/foo/bar')
- res.redirect('http://example.com')
- res.redirect(301, 'http://example.com')
- res.redirect('../login')

페이지 삭제 기능 Express 버전 구현

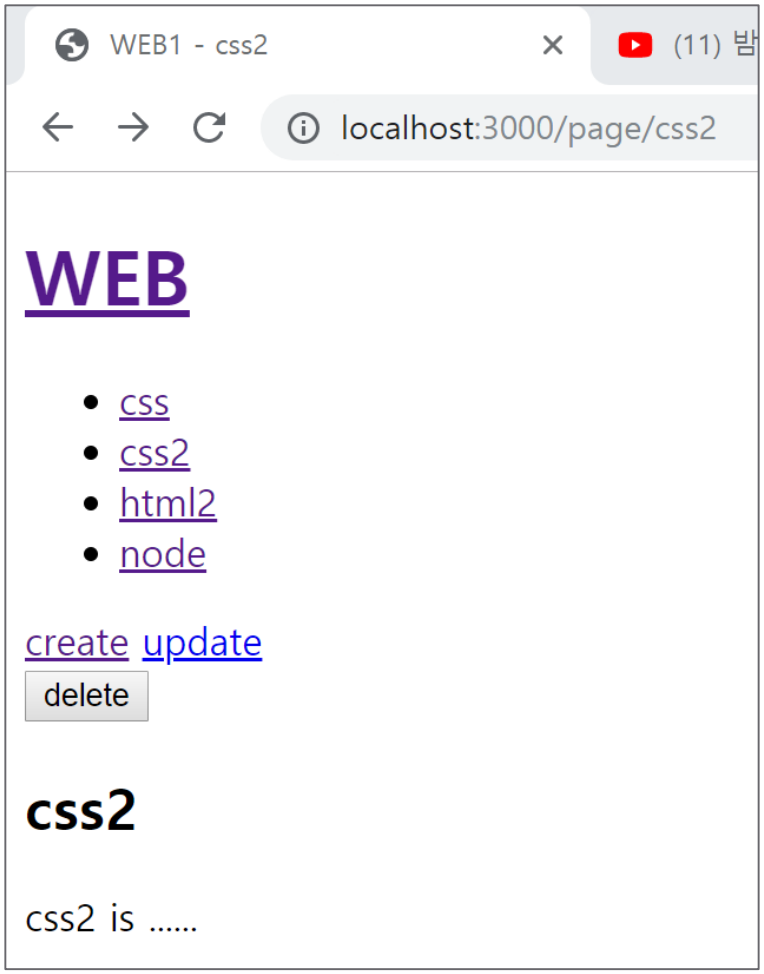
» 페이지 삭제 기능을 Express 버전으로 전환

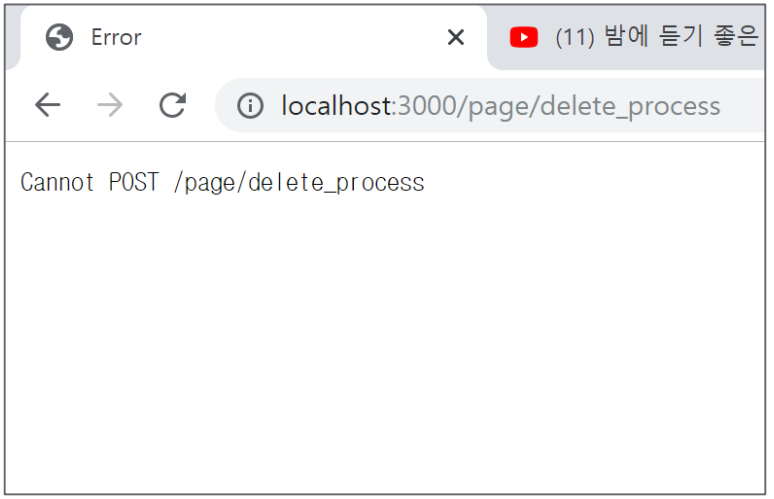
```
129 app.post('/delete_process', function(request, response){
130
131
132
133
134
135
136
137
138
139
140
141
142 });
```

페이지 삭제 기능 Express 버전 구현

Node.js 교과서

» 삭제 기능 문제점 해결-오류 찾기





삭제 기능 구현-오류해결