

네트워크서비스 프로토콜

11주차 1

» 소프트웨어학부

» 김형균 교수

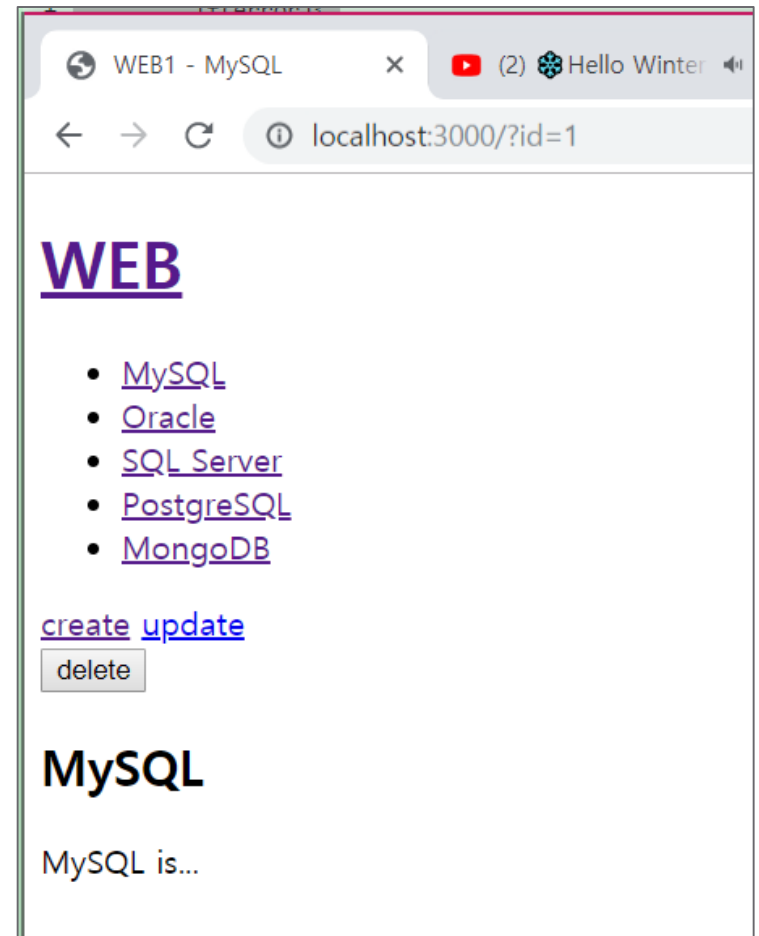
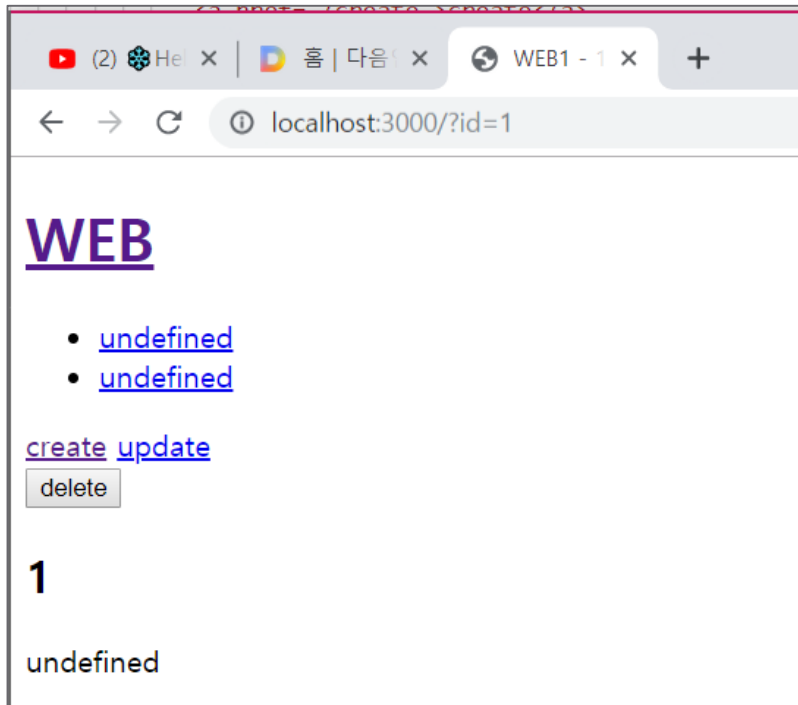


수업에 들어가며

- 지난 시간 복습
 - MySQL로 상세보기 구현
- 오늘 학습할 내용
 - MySQL로 글생성 기능 구현-1단계
 - MySQL로 글생성 기능 구현-2단계
 - MySQL로 글수정 기능 구현-1단계
 - MySQL로 글수정 기능 구현-2단계
 - MySQL로 글 삭제 기능 구현
 - SQL JOIN
 - MySQL join을 이용한 테이블 연계 기능 구현
 - MySQL join을 이용해서 글생성 구현

지난시간 진행내역

- MySQL로 상세보기 구현



```
34 } else {
35   db.query(`SELECT * FROM topic`, function(error, topics){
36     if(error){
37       throw error;
38     }
39     db.query(`SELECT * FROM topic WHERE id=?`, [queryData.id], function(error2, topic){
40       if(error2){
41         throw error2;
42       }
43       var title = topic[0].title;
44       var description = topic[0].description;
45       var list = template.list(topics);
46       var html = template.HTML(title, list,
47         `

## ${title}</h2>${description}`, 48 `


```

MySQL로 글생성 기능 구현-1단계



WEB1 - WEB - create

localhost:3000/create

WEB

- [undefined](#)
- [undefined](#)



WEB1 - Create

localhost:3000/create

WEB

- [MySQL](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [MongoDB](#)

[create](#)



localhost

localhost:3000/?id=node

An icon representing a broken or failed connection.

사이트에 연결할 수 없음

localhost에서 연결을 거부했습니다.

- <http://localhost3000.org/>을(를) 찾으셨나요?
- Google에서 localhost 3000 검색

ERR_CONNECTION_REFUSED

MySQL로 글생성 기능 구현-1단계



» 기존 수정해야할 부분 확인

```
21     if(pathname === '/'){
22 >         if(queryData.id === undefined){ ...
34 >         } else { ...
59     }
60 } else if(pathname === '/create'){
61     fs.readdir('./data', function(error, filelist){
62         var title = 'WEB - create';
63         var list = template.list(filelist);
64         var html = template.HTML(title, list, `
65             <form action="/create_process" method="post">
66                 <p><input type="text" name="title" placeholder="title"></p>
67                 <p>
68                     <textarea name="description" placeholder="description"></textarea>
69                 </p>
70                 <p>
71                     <input type="submit">
72                 </p>
73             </form>
74         `, '');
75         response.writeHead(200);
76         response.end(html);
77     });
78 > } else if(pathname === '/create_process'){ ...
```



MySQL로 글생성 기능 구현-1단계

```
60 } else if(pathname === '/create'){
61   db.query(`SELECT * FROM topic`, function(error, topics){
62     var title = 'Create';
63     var list = template.list(topics);
64     var html = template.HTML(title, list, `
65       <form action="/create_process" method="post">
66         <p><input type="text" name="title" placeholder="title"></p>
67         <p>
68           <textarea name="description" placeholder="description"></textarea>
69         </p>
70         <p>
71           <input type="submit">
72         </p>
73       </form>
74     `,
75     `<a href="/create">create</a>`
76   );
77   response.writeHead(200);
78   response.end(html);
79 });
80 } else if(pathname === '/create_process'){
```

MySQL로 글생성 기능 구현-2단계

» 테이블 레코드 추가 테스트

```
mysql> select * from topic;
```

id	title	description	created	author_id
1	MySQL	MySQL is...	2018-01-01 12:10:11	1
2	Oracle	Oracle is ...	2018-01-03 13:01:10	1
3	SQL Server	SQL Server is ...	2018-01-20 11:01:10	2
4	PostgreSQL	PostgreSQL is ...	2018-01-23 01:03:03	3
5	MongoDB	MongoDB is ...	2018-01-30 12:31:03	1

5 rows in set (0.00 sec)

```
mysql> insert into topic (title, description, created, author_id) values('Nodejs', 'Nodejs is ...', now(), 1);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from topic;
```

id	title	description	created	author_id
1	MySQL	MySQL is...	2018-01-01 12:10:11	1
2	Oracle	Oracle is ...	2018-01-03 13:01:10	1
3	SQL Server	SQL Server is ...	2018-01-20 11:01:10	2
4	PostgreSQL	PostgreSQL is ...	2018-01-23 01:03:03	3
5	MongoDB	MongoDB is ...	2018-01-30 12:31:03	1
6	Nodejs	Nodejs is ...	2019-10-27 15:13:18	1

6 rows in set (0.00 sec)

```
mysql> delete from topic where title like 'Nodejs';
Query OK, 1 row affected (0.04 sec)
```

```
mysql> select * from topic;
```

id	title	description	created	author_id
1	MySQL	MySQL is...	2018-01-01 12:10:11	1
2	Oracle	Oracle is ...	2018-01-03 13:01:10	1
3	SQL Server	SQL Server is ...	2018-01-20 11:01:10	2
4	PostgreSQL	PostgreSQL is ...	2018-01-23 01:03:03	3
5	MongoDB	MongoDB is ...	2018-01-30 12:31:03	1

MySQL로 글생성 기능 구현-2단계

» INSERT SQL 실행시

- insert할 때 생성된 PK 값은 result.insertId를 이용해서 가져올 수 있습니다.

```
dbconn.query(insertQuery, paramObj, function(err, result){  
  var clubSn = result.insertId;  
  res.redirect("/club/" + clubSn);  
});
```

MySQL로 글생성 기능 구현-2단계

» <form action="/create_process" method="post">

WEB1 - Create x (2)

localhost:3000/create

WEB

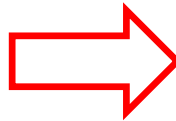
- [MySQL](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [MongoDB](#)

[create](#)

node

node is...

제출



WEB1 - node x (2)

localhost:3000/?id=7

WEB

- [MySQL](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [MongoDB](#)
- [node](#)

[create](#) [update](#)

[delete](#)

node

node is...

MySQL로 글생성 기능 구현-2단계

» 기존 수정해야할 부분 확인

```
80 } else if(pathname === '/create_process'){
81   var body = '';
82   request.on('data', function(data){
83     body = body + data;
84   });
85   request.on('end', function(){
86     var post = qs.parse(body);
87     var title = post.title;
88     var description = post.description;
89     fs.writeFile(`data/${title}`, description, 'utf8', function(err){
90       response.writeHead(302, {Location: `/?id=${title}`});
91       response.end();
92     })
93   });
94 } else if(pathname === '/update'){
```



MySQL로 글생성 기능 구현-2단계

```
80 } else if(pathname === '/create_process'){
81     var body = '';
82     request.on('data', function(data){
83         body = body + data;
84     });
85     request.on('end', function(){
86         var post = qs.parse(body);
87
88
89
90
91
92
93
94
95
96         response.end();
97     }
98 )
99 });
```



MySQL로 글수정 기능 구현-1단계

WEB1 - MySQL x (2) Relax Fall Jazz

localhost:3000/update?id=1

WEB

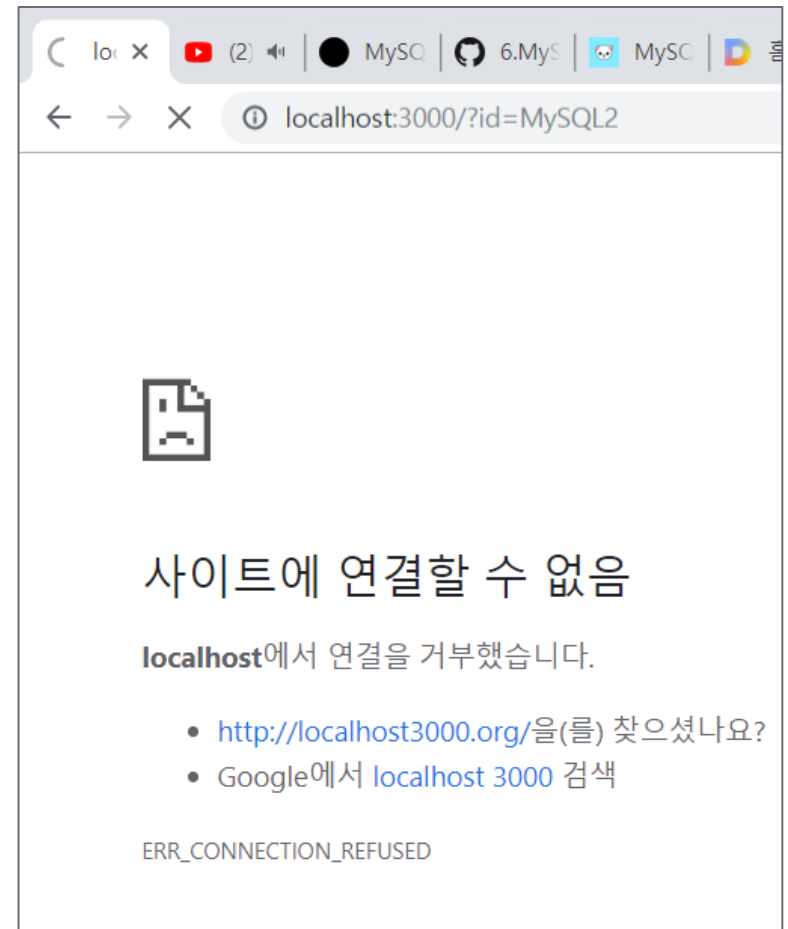
- [MySQL](#)
- [Oracle](#)
- [SQL_Server](#)
- [PostgreSQL](#)
- [MongoDB](#)
- [node](#)

[create](#) **update**

MySQL2

MySQL is...2

제출



MySQL로 글수정 기능 구현-1단계



» 기존 수정해야할 부분 확인

```
100     } else if(pathname === '/update'){
101         fs.readdir('./data', function(error, filelist){
102             var filteredId = path.parse(queryData.id).base;
103             fs.readFile(`data/${filteredId}`, 'utf8', function(err, description){
104                 var title = queryData.id;
105                 var list = template.list(filelist);
106                 var html = template.HTML(title, list,
107                     `
108                     <form action="/update_process" method="post">
109                         <input type="hidden" name="id" value="${title}">
110                         <p><input type="text" name="title" placeholder="title" value="${title}"></p>
111                         <p>
112                             <textarea name="description" placeholder="description">${description}</textarea>
113                         </p>
114                         <p>
115                             <input type="submit">
116                         </p>
117                     </form>
118                     `
119                     ,
120                     `<a href="/create">create</a> <a href="/update?id=${title}">update</a>`
121                 );
122                 response.writeHead(200);
123                 response.end(html);
124             });
125         });
126     }
```

MVC 구조를 그스저 기느 그하 1 단계

```
100 } else if(pathname === '/update'){
```

상세보기 기능과 유사

```
112 <form action="/update_process" method="post">
```

```
113   <input type="hidden" name="id" value="${topic[0].id}">
```

```
114   <p><input type="text" name="title" placeholder="title" value="${topic[0].title}"></p>
```

```
115   <p>
```

```
116     <textarea name="description" placeholder="description">${topic[0].description}</textarea>
```

```
117   </p>
```

```
118   <p>
```

```
119     <input type="submit">
```

```
120   </p>
```

```
121 </form>
```

```
122 `,
```

```
123 `<a href="/create">create</a> <a href="/update?id=${topic[0].id}">update</a>`
```

```
124 );
```

```
125 response.writeHead(200);
```

```
126 response.end(html);
```

```
127 });
```

```
128 });
```

MySQL로 글수정 기능 구현-2단계

» UPDATE : MySQL에서는 UPDATE 문을 사용하여 레코드의 내용을 수정할 수 있습니다.

» 문법

- UPDATE 테이블이름 SET 필드이름1=데이터값1, 필드이름2=데이터값2, ...
WHERE 필드이름=데이터값

» UPDATE 문은 해당 테이블에서 WHERE 절의 조건을 만족하는 레코드의 값만을 수정합니다.

» UPDATE Reservation SET RoomNum = 2002 WHERE Name = '홍길동';

실행 결과

ID	Name	ReserveDate	RoomNum
1	홍길동	2016-01-05	2002
2	임꺽정	2016-02-12	918
3	장길산	2016-01-16	1208
4	홍길동	2016-03-17	2002



MySQL로 글수정 기능 구현-2단계

WEB1 - MySQL x (2) Relax Fall Jazz

localhost:3000/update?id=1

WEB

- [MySQL](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [MongoDB](#)
- [node](#)

[create](#) [update](#)



WEB1 - MySQL2 x (2) Relax

localhost:3000/?id=1

WEB

- [MySQL2](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [MongoDB](#)
- [node](#)

[create](#) [update](#)

MySQL2

MySQL is...222

MySQL로 글수정 기능 구현-2단계

» 기존 수정해야할 부분 확인

```
129 } else if(pathname === '/update_process'){
130     var body = '';
131     request.on('data', function(data){
132         body = body + data;
133     });
134     request.on('end', function(){
135         var post = qs.parse(body);
136         var id = post.id;
137         var title = post.title;
138         var description = post.description;
139         fs.rename(`data/${id}`, `data/${title}`, function(error){
140             fs.writeFile(`data/${title}`, description, 'utf8', function(err){
141                 response.writeHead(302, {Location: `/?id=${title}`});
142                 response.end();
143             })
144         });
145     });
```



MySQL로 글수정 기능 구현-2단계

```
129 } else if(pathname === '/update_process'){
130     var body = '';
131     request.on('data', function(data){
132         body = body + data;
133     });
134     request.on('end', function(){
135         var post = qs.parse(body);
136
137
138
139
140
141     });
```

MySQL로 글 삭제 기능 구현

- » DELETE : MySQL에서는 DELETE 문을 사용하여 테이블의 레코드를 삭제
- » 문법
 - DELETE FROM 테이블이름 WHERE 필드이름=데이터값
- » DELETE 문은 해당 테이블에서 WHERE 절의 조건을 만족하는 레코드만을 삭제합니다.
- » 만약 WHERE 절을 생략하면, 해당 테이블에 저장된 모든 데이터가 삭제
- » 문법 : DELETE FROM 테이블이름;
- » 이때 테이블에 저장된 모든 데이터가 삭제되더라도 테이블은 여전히 남아있게 됩니다.
- » 해당 테이블까지 삭제하고 싶을 때는 DROP TABLE 문을 사용해야 합니다.

MySQL로 글 삭제 기능 구현

» 기존 수정해야할 부분 확인

```
142 } else if(pathname === '/delete_process'){
143     var body = '';
144     request.on('data', function(data){
145         body = body + data;
146     });
147     request.on('end', function(){
148         var post = qs.parse(body);
149         var id = post.id;
150         var filteredId = path.parse(id).base;
151         fs.unlink(`data/${filteredId}`, function(error){
152             response.writeHead(302, {Location: `/`});
153             response.end();
154         })
155     });
```



MySQL로 글 삭제 기능 구현

```
142  ✓ } else if(pathname === '/delete_process'){
143      var body = '';
144  ✓  request.on('data', function(data){
145      |      body = body + data;
146      |  });
147  ✓  request.on('end', function(){
148      |      var post = qs.parse(body);
149  ✓
150  ✓
151
152
153
154
155
156  });
```



SQL JOIN

» JOIN 절은 두 개 이상의 테이블에서 관련성이 있는 컬럼에 기초하여 행(Row)을 결합하는 데 사용됩니다.

» SQL INNER JOIN 키워드

- INNER JOIN 키워드는 양쪽 테이블 모두에서 일치하는 레코드를 반환합니다.
- SQL INNER JOIN 키워드
- `SELECT column_name(s) FROM table1
INNER JOIN table2 ON table1.column_name = table2.column_name;`

» LEFT JOIN 구문

- LEFT JOIN 키워드는 왼쪽 테이블(table1)의 모든 레코드와 오른쪽 테이블(table2)의 일치하는 레코드를 반환합니다. 오른쪽에 일치하는 레코드가 없으면 오른쪽에서는 결과가 NULL이 됩니다.
- Left Join 쿼리 예:
- `SELECT column_name(s) FROM table1
LEFT JOIN table2 ON table1.column_name = table2.column_name;`

SQL JOIN



» SQL RIGHT JOIN 키워드

- RIGHT JOIN 키워드는 오른쪽 테이블(table2)의 모든 레코드와 왼쪽 테이블(table1)의 일치하는 레코드를 반환합니다. 왼쪽에 일치하는 레코드가 없으면 오른쪽에서는 결과가 NULL이 됩니다.
- Right Join 키워드예
- ```
SELECT column_name(s) FROM table1
 RIGHT JOIN table2 ON table1.column_name = table2.column_name;
```

## » SQL FULL OUTER JOIN 키워드

- FULL OUTER JOIN 키워드는 왼쪽 테이블(table1) 또는 오른쪽 테이블(table2) 레코드에 일치하는 모든 레코드를 반환합니다.
- SQL FULL OUTER JOIN 키워드 예:
- ```
SELECT column_name(s) FROM table1  
  FULL OUTER JOIN table2 ON table1.column_name = table2.column_name;
```




MySQL join을 이용한 테이블 연계 기능 구현

```
mysql> show tables;
+-----+
| Tables_in_opentutorials |
+-----+
| author |
| topic |
+-----+
2 rows in set (0.02 sec)

mysql> select * from author;
+----+-----+-----+
| id | name | profile |
+----+-----+-----+
| 1 | kim | developer |
| 2 | lee | database administrator |
| 3 | park | data scientist, developer |
+----+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from topic;
+----+-----+-----+-----+-----+
| id | title | description | created | author_id |
+----+-----+-----+-----+-----+
| 1 | MySQL | MySQL is... | 2018-01-01 12:10:11 | 1 |
| 2 | Oracle | Oracle is ... | 2018-01-03 13:01:10 | 1 |
| 3 | SQL Server | SQL Server is ... | 2018-01-20 11:01:10 | 2 |
| 4 | PostgreSQL | PostgreSQL is ... | 2018-01-23 01:03:03 | 3 |
| 5 | MongoDB | MongoDB is ... | 2018-01-30 12:31:03 | 1 |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

WEB1 - MySQL x (3) Milky J x

localhost:3000/?id=1

WEB

- [MySQL2](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [node](#)

[create](#) [update](#)

[delete](#)

MySQL2

MySQL is...222

by kim

WEB1 - SQL Se x (3) Milky J x

localhost:3000/?id=3

WEB

- [MySQL2](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [node](#)

[create](#) [update](#)

[delete](#)

SQL Server

SQL Server is ...

by lee



MySQL join을 이용한 테이블 연계 기능 구현

```
SELECT * FROM topic LEFT JOIN author ON  
topic.author_id=author.id WHERE topic.id=?
```

```
mysql> use opentutorials;  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_opentutorials |  
+-----+  
| author  
| topic  
+-----+  
2 rows in set (0.00 sec)  
  
mysql> SELECT * FROM topic;  
+----+-----+-----+-----+-----+  
| id | title      | description      | created      | author_id |  
+----+-----+-----+-----+-----+  
| 1  | MySQL2     | MySQL is...222   | 2018-01-01 12:10:11 | 1  
| 2  | Oracle     | Oracle is ...    | 2018-01-03 13:01:10 | 1  
| 3  | SQL Server | SQL Server is ... | 2018-01-20 11:01:10 | 2  
| 4  | PostgreSQL | PostgreSQL is ... | 2018-01-23 01:03:03 | 3  
| 7  | node       | node is...       | 2019-10-27 15:30:23 | 1  
+----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)  
  
mysql> SELECT * FROM author;  
+----+-----+-----+  
| id | name      | profile  
+----+-----+-----+  
| 1  | kim       | developer  
| 2  | lee       | database administrator  
| 3  | park      | data scientist, developer  
+----+-----+-----+
```



MySQL join을 이용한 테이블 연계 기능 구현

» 기존 수정해야할 부분 확인

```
17 var app = http.createServer(function(request,response){
18   var _url = request.url;
19   var queryData = url.parse(_url, true).query;
20   var pathname = url.parse(_url, true).pathname;
21   if(pathname === '/'){
22 >     if(queryData.id === undefined){...
34     } else {
35       db.query(`SELECT * FROM topic`, function(error,topics){
36 >         if(error){...
38       }
39       db.query(`SELECT * FROM topic WHERE id=?`,[queryData.id], function(error2, topic){
40         if(error2){
41           throw error2;
42         }
43         var title = topic[0].title;
44         var description = topic[0].description;
45         var list = template.list(topics);
46         var html = template.HTML(title, list,
47           `

## ${title}</h2>${description}`, 48 `


```



```
21 if(pathname === '/'){
22 >   if(queryData.id === undefined){ ...
34   } else {
35     db.query(`SELECT * FROM topic`, function(error, topics){
36 >       if(error){ ...
38       }
39       [REDACTED]
40       [REDACTED]
41       if(error2){
42         throw error2;
43       }
44       var title = topic[0].title;
45       var description = topic[0].description;
46       var list = template.list(topics);
47       var html = template.HTML(title, list,
48         `

## ${title}</h2>${description} 49 [REDACTED] 50 <a href="/create">create</a> 51 <a href="/update?id=${queryData.id}">update</a> 52 <form action="delete_process" method="post"> 53 <input type="hidden" name="id" value="${queryData.id}"> 54 <input type="submit" value="delete"> 55 </form>` 56 );


```

MySQL join을 이용해서 글생성 구현



WEB1 - Create x (4) Rom. x

localhost:3000/create

WEB

- [MySQL2](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [node](#)

[create](#)

title

description

제출

WEB1 - Create x (4) Rom. x

localhost:3000/create

WEB

- [MySQL2](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [node](#)

[create](#)

SQL2

SQL2 IS, , ,

kim

제출

WEB1 - SQL x (4) Rom. x

localhost:3000/?id=8

WEB

- [MySQL2](#)
- [Oracle](#)
- [SQL Server](#)
- [PostgreSQL](#)
- [node](#)
- [SQL2](#)

[create](#) [update](#)

delete

SQL2

SQL2 IS, , ,

by kim

MySQL join을 이용해서 글생성 구현



» 기존 수정해야할 부분 확인

```
62 } else if(pathname === '/create'){
63   db.query(`SELECT * FROM topic`, function(error, topics){
64     var title = 'Create';
65     var list = template.list(topics);
66     var html = template.HTML(title, list, `
67       <form action="/create_process" method="post">
68         <p><input type="text" name="title" placeholder="title"></p>
69         <p>
70           <textarea name="description" placeholder="description"></textarea>
71         </p>
72         <p>
73           <input type="submit">
74         </p>
75       </form>
76     `,
77     `<a href="/create">create</a>`
78   );
79   response.writeHead(200);
80   response.end(html);
81 }));
```

```
82 } else if(pathname === '/create_process'){
83   var body = '';
84   request.on('data', function(data){
85     body = body + data;
86   });
87   request.on('end', function(){
88     var post = qs.parse(body);
89     db.query(`
90       INSERT INTO topic (title, description, created, author_id)
91       VALUES(?, ?, NOW(), ?)`,
92     [post.title, post.description, 1],
93     function(error, result){
94       if(error){
95         throw error;
96       }
97       response.writeHead(302, {Location: `/?id=${result.insertId}`});
98       response.end();
99     }
100   )
101 }));
```

```
62 } else if(pathname === '/create'){
63   db.query(`SELECT * FROM topic`, function(error, topics){
64     db.query('SELECT * FROM author', function(error2, authors){
65       var title = 'Create';
66       var list = template.list(topics);
67       var html = template.HTML(title, list,
68         `
69         <form action="/create_process" method="post">
70           <p><input type="text" name="title" placeholder="title"></p>
71           <p>
72             <textarea name="description" placeholder="description"></textarea>
73           </p>
74           <p>
75             ${template.authorSelect(authors)}
76           </p>
77           <p>
78             <input type="submit">
79           </p>
80         </form>
81         `
82         ,
83         `<a href="/create">create</a>`
84       );
85       response.writeHead(200);
86       response.end(html);
87     });
88   });
89 }
```

```
88 } else if(pathname === '/create_process'){
89     var body = '';
90     request.on('data', function(data){
91         body = body + data;
92     });
93     request.on('end', function(){
94         var post = qs.parse(body);
95         db.query(`
96             INSERT INTO topic (title, description, created, author_id)
97             VALUES(?, ?, NOW(), ?)`,
98             [post.title, post.description, post.author],
99             function(error, result){
100                 if(error){
101                     throw error;
102                 }
103                 response.writeHead(302, {Location: `/?id=${result.insertId}`});
104                 response.end();
105             }
106         )
107     });
```




lib > template.js 수정

```
1  module.exports = {  
2  >  HTML:function(title, list, body, control){ ...  
18    },list:function(topics){  
19      var list = '<ul>';  
20      var i = 0;  
21      while(i < topics.length){  
22        list = list + `<li><a href="/?id=${topics[i].id}">${topics[i].title}</a></li>`;   
23        i = i + 1;  
24      }  
25      list = list+'</ul>';  
26      return list;  
27    },authorSelect:function(authors){  
28      var tag = '';  
29        
30        
31        
32        
33        
34        
35        
36        
37        
38        
39    }  
40  }
```