

소프트웨어공학

Node.js Tutorial

Reference

<http://devlecture.tistory.com/>

Kwangwoon Univ.
Dept. of Computer Engineering
Ki-Hoon Lee



Contents

1. Node.js 설치 및 예제 실행
2. Express 설치 및 예제 실행
3. 회원가입 폼 만들기 및 nodemon 사용
4. Database(MySQL) 연동
5. 게시판 구축
6. MySQL 기본



1. node.js 설치 및 예제 실행

- nodejs.org 접속 후 node.js 설치



Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#). Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, [npm](#), is the largest ecosystem of open source libraries in the world.

Important [security releases](#), please update now!

Download for Windows (x64)

LTS 버전 설치

v4.4.0 LTS

Recommended For Most Users

v5.9.0 Stable

Latest Features

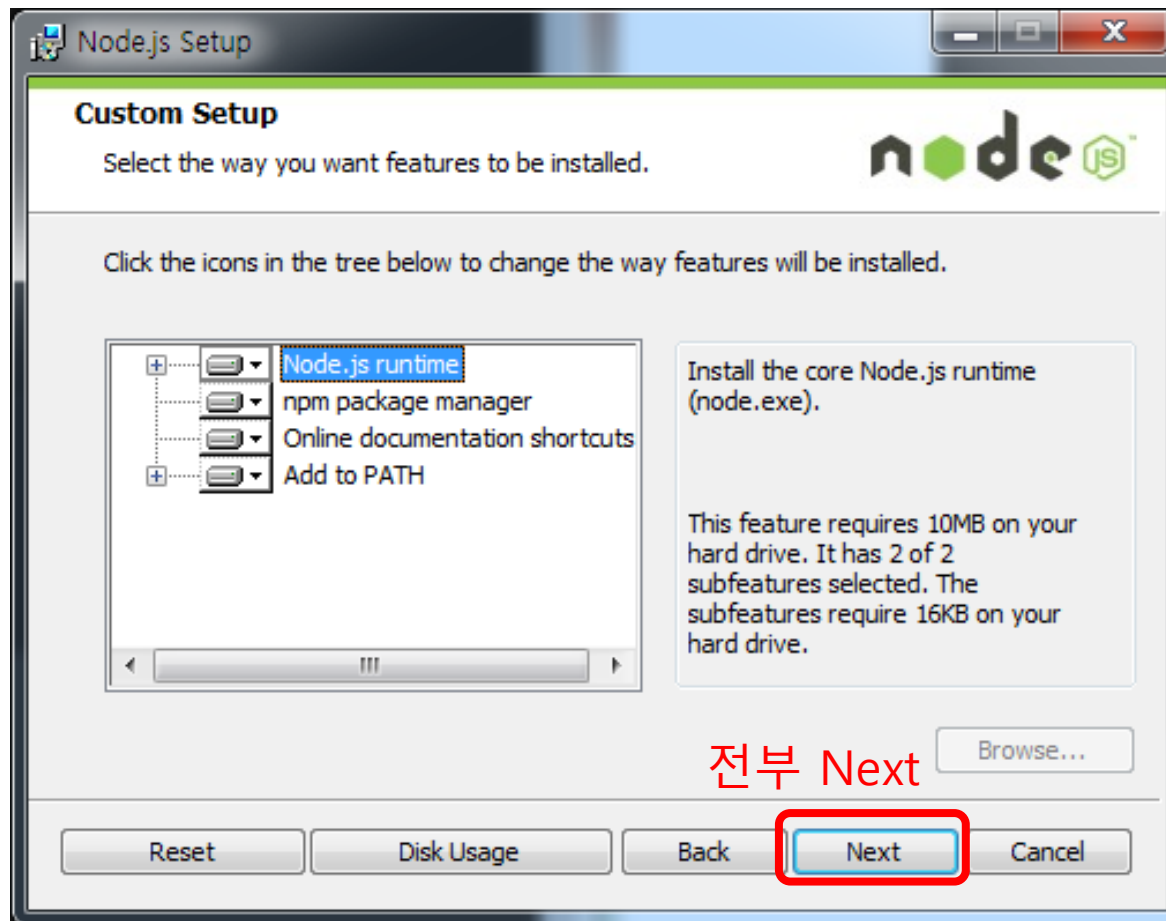
[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [LTS schedule](#).

1. node.js 설치 및 예제 실행

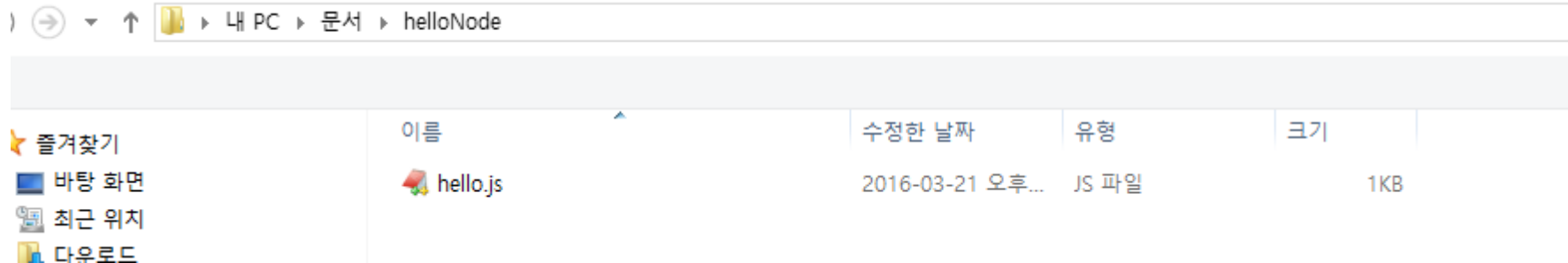
■ 설치파일 실행





1. node.js 설치 및 예제 실행

- helloNode 폴더 생성 후 폴더 안에 hello.js 파일 생성



- 예제소스 작성

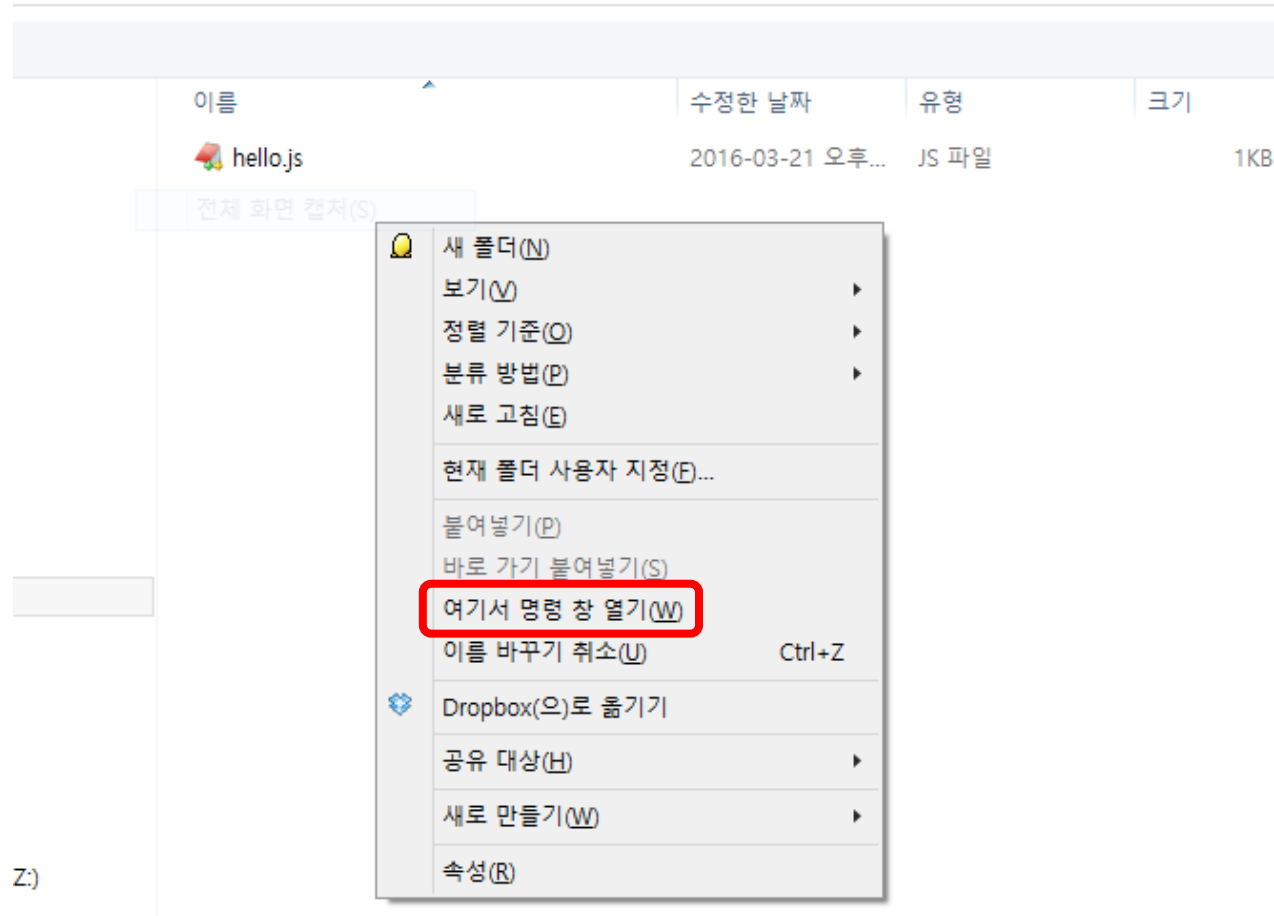
```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```



1. node.js 설치 및 예제 실행

- 폴더에서 Shift + 마우스 우클릭 -> 여기서 명령 창 열기

PC > 문서 > helloNode



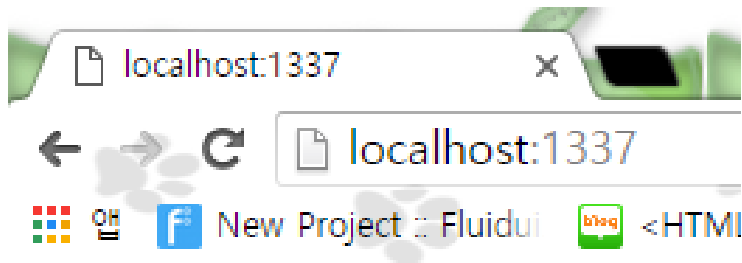
1. node.js 설치 및 예제 실행

- >node hello.js 입력



```
node hello.js  
  
D:\Users\User\Documents\helloNode>node hello.js  
Server running at http://127.0.0.1:1337/
```

- 웹 브라우저를 통해 확인



Hello World



1. node.js 설치 및 예제 실행

- Ctrl + C를 통해 종료

```
C:\WINDOWS\system32  
  
D:\Users\User\Documents\helloNode>node hello.js  
Server running at http://127.0.0.1:1337/  
^C  
D:\Users\User\Documents\helloNode>
```




2. Express 설치 및 예제 실행

- 이전 예제와 같이 새로운 폴더 생성 후 명령프롬프트 실행
- >npm install -g express-generator 입력



C:\WINDOWS\system32\cmd.exe

```
D:\Users\User\Documents\myNode>npm install -g express-generator
C:\Users\User\AppData\Roaming\npm\express -> C:\Users\User\AppData\Local\
node_modules\express-generator\bin\express
express-generator@4.13.1 C:\Users\User\AppData\Roaming\npm\node_mod
generator
├── sorted-object@1.0.0
├── commander@2.7.1 (graceful-readlink@1.0.1)
└── mkdirp@0.5.1 (minimist@0.0.8)

D:\Users\User\Documents\myNode>
```



2. Express 설치 및 예제 실행

- > express example1 입력

```
D:\Users\User\Documents\myNode>express example1
```

```
create : example1
create : example1/package.json
create : example1/app.js
create : example1/public
create : example1/public/stylesheets
create : example1/public/stylesheets/style.css
create : example1/public/images
create : example1/public/javascripts
create : example1/routes
create : example1/routes/index.js
create : example1/routes/users.js
create : example1/views
create : example1/views/index.jade
create : example1/views/layout.jade
create : example1/views/error.jade
create : example1/bin
create : example1/bin/www
```

```
install dependencies:
```

```
> cd example1 && npm install
```

```
run the app:
```

```
> SET DEBUG=example1:* & npm start
```



2. Express 설치 및 예제 실행

- >cd example1 # example1로 이동
- >npm install # 모듈 설치

```
C:\WINDOWS\system32\cmd
D:\Users\User\Documents\myNode>cd example1

D:\Users\User\Documents\myNode\example1>npm install
debug@2.2.0 node_modules\debug
├── ms@0.7.1

cookie-parser@1.3.5 node_modules\cookie-parser
├── cookie@0.1.3
└── cookie-signature@1.0.6

serve-favicon@2.3.0 node_modules\serve-favicon
├── etag@1.7.0
├── fresh@0.3.0
├── parseurl@1.3.1
└── ms@0.7.1

morgan@1.6.1 node_modules\morgan
├── on-headers@1.0.1
├── basic-auth@1.0.3
├── depd@1.0.1
└── on-finished@2.3.0 (ee-first@1.1.1)

express@4.13.4 node_modules\express
└── utils-merge@1.0.0
```



2. Express 설치 및 예제 실행

- >npm start 혹은 >node .\bin\www 실행



npm

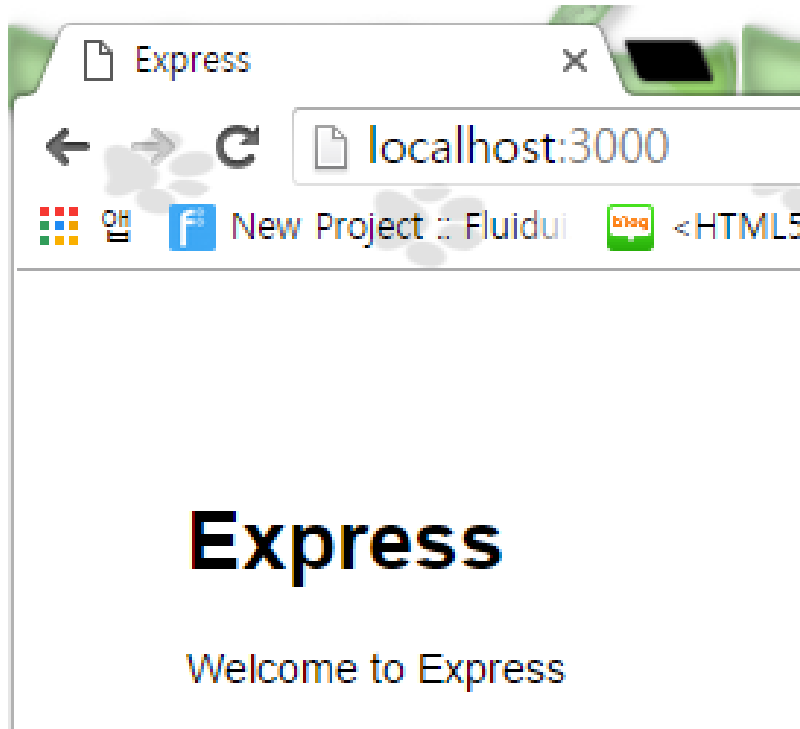
```
D:\Users\User\Documents\myNode\example1>npm start
```

```
> example1@0.0.0 start D:\Users\User\Documents\myNode\example1
```

```
> node ./bin/www
```

2. Express 설치 및 예제 실행

- 웹 브라우저를 통해 확인





2. Express 설치 및 예제 실행

- app.js에서 url을 매핑하는 부분 확인



```
8 var routes = require('./routes/index');  
9 var users = require('./routes/users');  
10  
25 app.use('/', routes);  
26 app.use('/users', users);  
--
```



3. 회원가입 폼 만들기 및 nodemon 사용

- 아이디, 비밀번호, 이름, 이메일, 전화번호, 주소, 직업, 성별, 생일을 입력받음
 - 모든 값에 공백 검사
 - 가입 버튼을 클릭 시 입력한 내용을 json 객체 형태로 표시
- 순서
 - 1) express 프로젝트 생성(ejs 템플릿엔진으로 생성)
 - 2) routes 폴더에 로직 담당하는 js 파일 생성
 - 3) app.js에 소스를 적어 2에서 생성한 js파일 연결
 - 4) views 폴더에 ejs 파일 생성
 - 5) 2에서 만든 js파일에 소스를 적어 4번에서 만든 ejs 파일 연결



3. 회원가입 폼 만들기 및 nodemon 사용

- > express --ejs joinForm # 프로젝트 생성

```
C:\> C:\WINDOWS\system32\cmd.exe

D:\Users\User\Documents\MyNode>express --ejs joinForm

create : joinForm
create : joinForm/package.json
create : joinForm/app.js
create : joinForm/public
create : joinForm/public/javascripts
create : joinForm/public/images
create : joinForm/public/stylesheets
create : joinForm/public/stylesheets/style.css
create : joinForm/routes
create : joinForm/routes/index.js
create : joinForm/routes/users.js
create : joinForm/views
create : joinForm/views/index.ejs
create : joinForm/views/error.ejs
create : joinForm/bin
create : joinForm/bin/www

install dependencies:
> cd joinForm && npm install

run the app:
> SET DEBUG=joinForm:* & npm start
```




3. 회원가입 폼 만들기 및 nodemon 사용

- >cd joinForm # joinForm 폴더로 이동
- >npm install # 모듈 설치

```
C:\WINDOWS\system32\cmd
D:\Users\User\Documents\myNode>cd joinForm

D:\Users\User\Documents\myNode\joinForm>npm install
cookie-parser@1.3.5 node_modules\cookie-parser
├── cookie@0.1.3
├── cookie-signature@1.0.6

serve-favicon@2.3.0 node_modules\serve-favicon
├── etag@1.7.0
├── fresh@0.3.0
├── parseurl@1.3.1
└── ms@0.7.1

debug@2.2.0 node_modules\debug
└── ms@0.7.1

morgan@1.6.1 node_modules\morgan
├── on-headers@1.0.1
├── basic-auth@1.0.3
├── depd@1.0.1
└── on-finished@2.3.0 (ee-first@1.1.1)

e.is@2.3.4 node_modules\e.is
```



3. 회원가입 폼 만들기 및 nodemon 사용

- routes 폴더에 joinForm.js 파일 생성
- 같은 폴더에 생성된 users.js에 res.send 부분만 변경

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET users listing. */
5 router.get('/', function(req, res, next) {
6   res.send('joinForm.js check');
7 });
8
9 module.exports = router;
```



3. 회원가입 폼 만들기 및 nodemon 사용

- app.js에 소스를 추가하여 joinForm.js 파일 연결

```
8  var routes = require('./routes/index');
9  var users = require('./routes/users');
10 var join = require('./routes/joinForm');

26 app.use('/', routes);
27 app.use('/users', users);
28 app.use('/join', join);
```



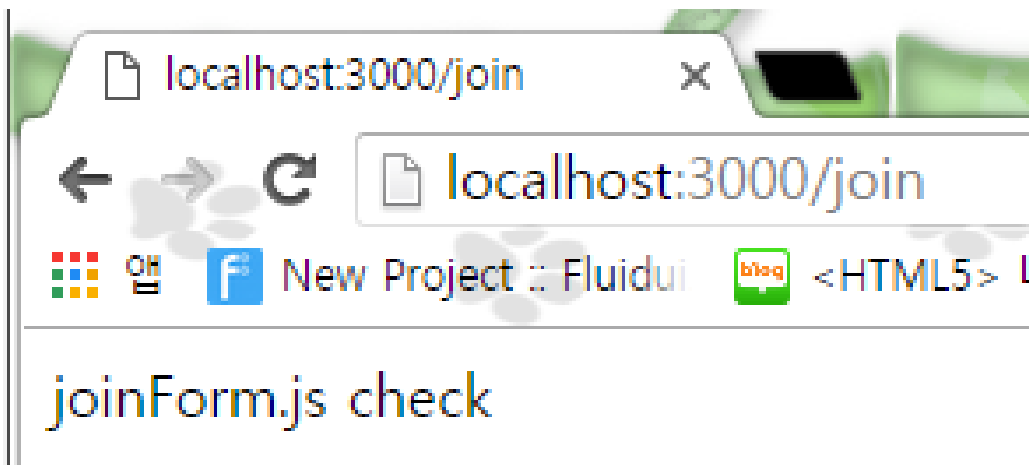
3. 회원가입 폼 만들기 및 nodemon 사용

- 서버를 시작하여 웹 브라우저를 통해 동작 확인

npm

```
D:\Users\User\Documents\myNode\joinForm>npm start
```

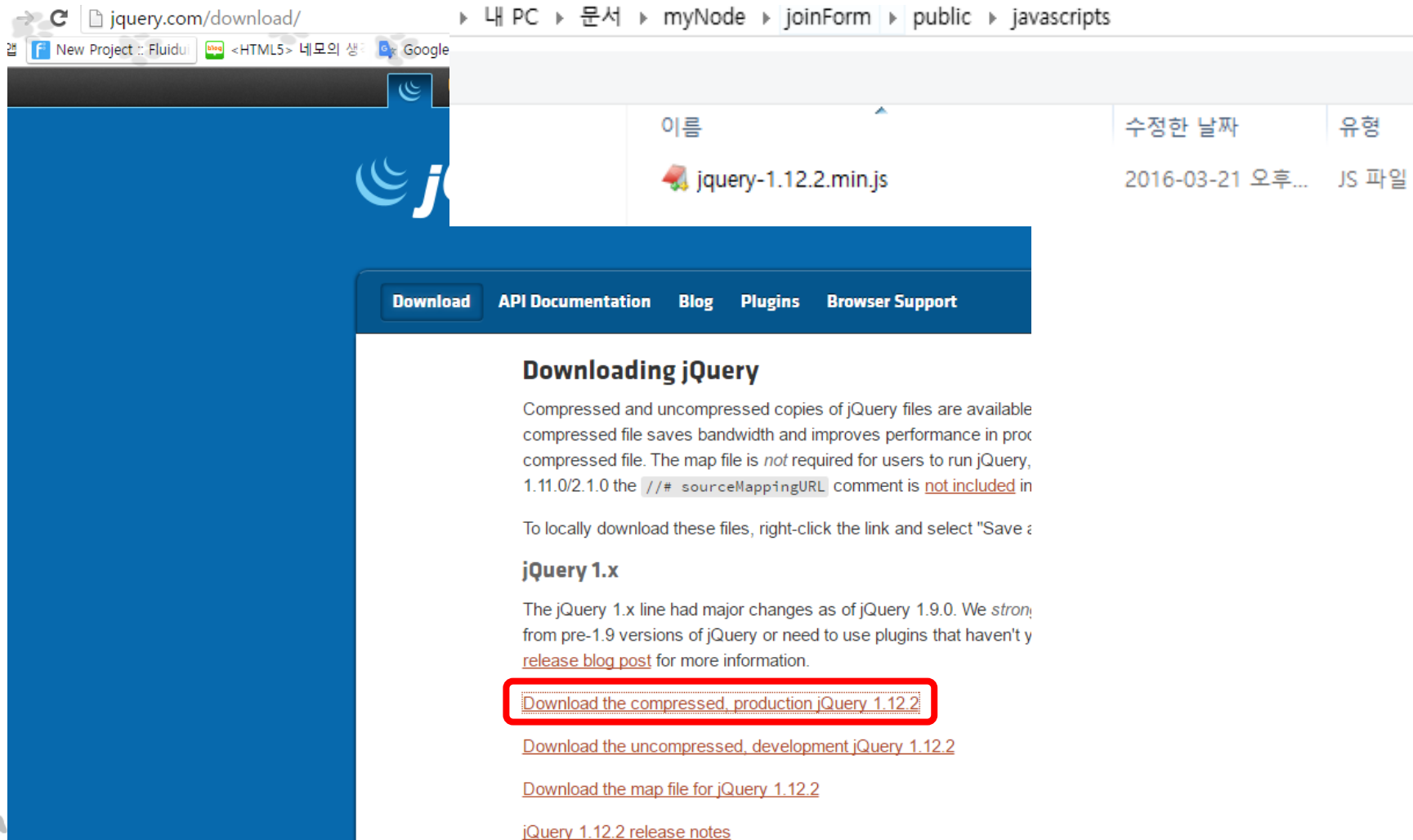
```
> joinForm@0.0.0 start D:\Users\User\Documents\myNode\joinForm
> node ./bin/www
```





3. 회원가입 폼 만들기 및 nodemon 사용

- jQuery 다운로드 후 public\javascripts 폴더에 저장



The screenshot shows a web browser window with the URL `jquery.com/download/`. The browser's address bar shows the path `내 PC > 문서 > myNode > joinForm > public > javascripts`. The main content area displays the jQuery logo and a table with the following information:

이름	수정한 날짜	유형
jquery-1.12.2.min.js	2016-03-21 오후...	JS 파일

Below the table, there is a navigation bar with links: [Download](#), [API Documentation](#), [Blog](#), [Plugins](#), and [Browser Support](#). The main content area is titled "Downloading jQuery" and contains the following text:

Compressed and uncompressed copies of jQuery files are available. The compressed file saves bandwidth and improves performance in production environments. The map file is *not* required for users to run jQuery, 1.11.0/2.1.0 the `//# sourceMappingURL` comment is *not included* in the compressed file.

To locally download these files, right-click the link and select "Save as..."

jQuery 1.x

The jQuery 1.x line had major changes as of jQuery 1.9.0. We strongly recommend upgrading from pre-1.9 versions of jQuery or need to use plugins that haven't yet released a [release blog post](#) for more information.

[Download the compressed, production jQuery 1.12.2](#)

[Download the uncompressed, development jQuery 1.12.2](#)

[Download the map file for jQuery 1.12.2](#)

[jQuery 1.12.2 release notes](#)

3. 회원가입 폼 만들기 및 nodemon 사용

- 다음 소스코드를 따라서 작성하여 views 폴더에 joinForm.ejs 템플릿 생성

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title><%= title %></title>
5     <meta charset='utf-8'>
6     <link rel='stylesheet' href='/stylesheets/style.css' />
7     <script src="/javascripts/jquery-1.12.2.min.js"></script>
8     <script>
9     $(function(){
10
11         $("#btnJoin").click(
12             function(){
13
14                 if($("#id").val() == "")
15                 {
16                     alert("아이디를 꼭 입력 하세요");
17                     $("#id").focus();
18                     return ;
19                 }
20                 if($("#passwd").val() == "")
21                 {
22                     alert("비밀번호를 꼭 입력 하세요");
23                     $("#passwd").focus();
24                     return ;
25                 }

```



3. 회원가입 폼 만들기 및 nodemon 사용

```
26         if($("#name").val() == "")
27         {
28             alert("이름을 꼭 입력 하세요");
29             $("#name").focus();
30             return ;
31         }
32         if($("#email").val() == "")
33         {
34             alert("이메일을 꼭 입력 하세요");
35             $("#email").focus();
36             return ;
37         }
38         if($("#tel2").val() == "")
39         {
40             alert("전화번호 두번째 자리를 확인해 주세요");
41             $("#tel2").focus();
42             return ;
43         }
44         if($("#tel3").val() == "")
45         {
46             alert("전화번호 세번째 자리를 확인해 주세요");
47             $("#tel3").focus();
48             return ;
49         }
50
51         var tel= ($("#tel1").val() + '-' + ($("#tel2").val() + '-' + ($("#tel3").val() ;
52         $("#tel").val(tel);
53
```

< 소스코드 2/6 >



3. 회원가입 폼 만들기 및 nodemon 사용

```
54         if($("#address").val() == "")
55         {
56             alert("주소를 꼭 입력해 주세요");
57             $("#address").focus();
58             return ;
59         }
60
61         if($("#birth").val() == "")
62         {
63             alert("생일을 꼭 입력해 주세요");
64             $("#birth").focus();
65             return ;
66         }
67         if($("#birth").val().length != 8)
68         {
69             alert("생일은 8글자로 입력해 주세요");
70             $("#birth").focus();
71             return ;
72         }
73         if(isNaN($("#birth").val()))
74         {
75             alert("생일은 숫자만 입력해 주세요");
76             $("#birth").focus();
77             return ;
78         }
79         $("#joinForm").submit();
80
81     }
82 );
83
84 });
85 </script>
86 </head>
```




3. 회원가입 폼 만들기 및 nodemon 사용

```
87 <body>
88   <h1><%= title %></h1>
89   <form id="joinForm" action="/join" method="post">
90     <table width="400" border="1">
91       <tr>
92         <th>아이디</th>
93         <td><input type="text" name="id" id="id" size="12" maxLength="12"></td>
94       </tr>
95       <tr>
96         <th>비밀번호</th>
97         <td><input type="password" name="passwd" id="passwd" size="12" maxLength="12"></td>
98       </tr>
99       <tr>
100        <th>이름</th>
101        <td><input type="text" name="name" id="name" size="10" maxLength="10"></td>
102      </tr>
103      <tr>
104        <th>이메일</th>
105        <td><input type="email" name="email" id="email" size="30" maxLength="50"></td>
106      </tr>
```

< 소스코드 4/6 >



3. 회원가입 폼 만들기 및 nodemon 사용

```
107     <tr>
108         <th>전화번호</th>
109         <td>
110             <input type="hidden" name="tel" id="tel">
111             <select name="tel1" id="tel1">
112                 <option value="010">010</option>
113                 <option value="011">011</option>
114                 <option value="016">016</option>
115                 <option value="017">017</option>
116                 <option value="018">018</option>
117                 <option value="019">019</option>
118             </select>
119             -
120             <input type="text" name="tel2" id="tel2" size="4" maxLength="4">
121             -
122             <input type="text" name="tel3" id="tel3" size="4" maxLength="4">
123         </td>
124     </tr>
125     <tr>
126         <th>주소</th>
127         <td><input type="text" id="address" size="30" maxLength="50"></td>
128     </tr>
129     <tr>
130         <th>직업</th>
131         <td><select name="job" id="job">
132             <option value="학생">학생</option>
133             <option value="직장인">직장인</option>
134             <option value="주부">주부</option>
135         </select></td>
136     </tr>
```



3. 회원가입 폼 만들기 및 nodemon 사용

```
137         <tr>
138             <th>성별</th>
139             <td>
140                 <input type="radio" name="gender" value="남" checked="checked">남
141                 <input type="radio" name="gender" value="여">여
142             </td>
143         </tr>
144         <tr>
145             <th>생일</th>
146             <td><input type="text" name="birth" id="birth" size="8" maxLength="8">*YYYYMMDD</td>
147         </tr>
148         <tr>
149             <td colspan="2">
150                 <input type="button" id="btnJoin" name="join" value="가입">
151                 <input type="reset" name="cancel" value="취소">
152             </td>
153         </tr>
154     </table>
155 </form>
156
157
158 </body>
159 </html>
```

< 소스코드 6/6 >



3. 회원가입 폼 만들기 및 nodemon 사용

- joinForm.js의 res.send 부분을 다음과 같이 수정

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET users listing. */
5 router.get('/', function(req, res, next) {
6   res.render('joinForm', { title: 'Join Form!' });
7 });
8
9 module.exports = router;
10
```



3. 회원가입 폼 만들기 및 nodemon 사용

- 서버 재시작 후 웹 브라우저를 통해 결과 확인

Join Form!

localhost:3000/join

Join Form!

아이디	<input type="text"/>
비밀번호	<input type="password"/>
이름	<input type="text"/>
이메일	<input type="text"/>
전화번호	010 ▼ - <input type="text"/> - <input type="text"/>
주소	<input type="text"/>
직업	학생 ▼
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여
생일	<input type="text"/> *YYYYMMDD
<input type="button" value="가입"/> <input type="button" value="취소"/>	



3. 회원가입 폼 만들기 및 nodemon 사용

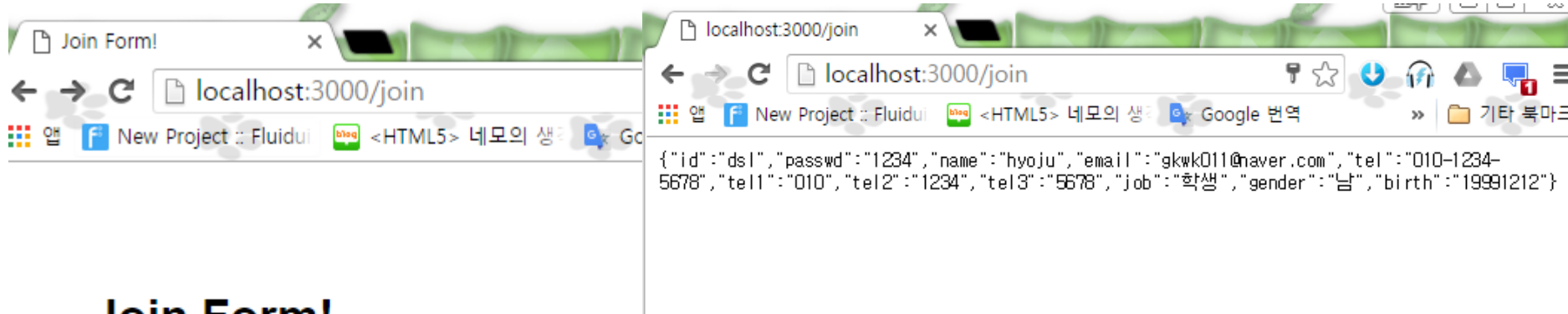
- 사용자 입력값을 JSON으로 출력하기 위해 joinForm.js에 다음 소스코드 추가

```
1 var express = require('express');
2 var router = express.Router();
3
4 /* GET users listing. */
5 router.get('/', function(req, res, next) {
6   res.render('joinForm', { title: 'Join Form!' });
7 });
8
9 router.post('/', function(req, res, next){
10   console.log('req.body : '+req.body);
11   res.json(req.body);
12 });
13
14 module.exports = router;
```



3. 회원가입 폼 만들기 및 nodemon 사용

- 서버 재시작 후 웹 브라우저를 통해 결과 확인



Join Form!

아이디	<input type="text" value="dsl"/>
비밀번호	<input type="password" value="...."/>
이름	<input type="text" value="hyoju"/>
이메일	<input type="text" value="gkwk011@naver.com"/>
전화번호	010 ▼ - 1234 - 5678
주소	<input type="text" value="광운대학교"/>
직업	학생 ▼
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여
생일	19991212 *YYYYMDD
<input type="button" value="가입"/> <input type="button" value="취소"/>	



3. 회원가입 폼 만들기 및 nodemon 사용

■ nodemon

- 소스코드를 수정할 때 마다 서버를 재시작하지 않아도 됨
- 수정된 소스가 콘솔창에 나타남

■ > npm install -g nodemon

```
C:\WINDOWS\system32\cmd.exe

D:\Users\User\Documents\MyNode\joinForm>npm install -g nodemon
npm WARN optional dep failed, continuing fsevents@1.0.8
C:\Users\User\AppData\Roaming\npm\nodemon -> C:\Users\User\AppData\Roaming\node_modules\nodemon\bin\nodemon.js
nodemon@1.9.1 C:\Users\User\AppData\Roaming\npm\node_modules\nodemon
├── ignore-by-default@1.0.1
├── undefsafe@0.0.3
├── es6-promise@3.1.2
├── debug@2.2.0 <ms@0.7.1>
├── minimatch@3.0.0 <brace-expansion@1.1.3>
├── touch@1.0.0 <nopt@1.0.10>
├── lodash.defaults@3.1.2 <lodash.restparam@3.6.1, lodash.assign@3.2.0>
├── ps-tree@1.0.1 <event-stream@3.3.2>
├── chokidar@1.4.3 <path-is-absolute@1.0.0, inherits@2.0.1, glob-parent@2.0.0, async-each@1.0.0, is-glob@2.0.1, is-binary-path@1.0.1, readdirp@2.0.0, anymatch@1.3.0>
├── update-notifier@0.5.0 <is-npm@1.0.0, semver-diff@2.1.0, string-length@1.0.1, repeating@1.1.3, chalk@1.1.1, configstore@1.4.0, latest-version@1.0.1>
```




3. 회원가입 폼 만들기 및 nodemon 사용

- package.json 수정

```
{  
  "name": "joinForm",  
  "version": "0.0.0",  
  "private": true,  
  "scripts": {  
    "start": "nodemon ./bin/www"  
  },  
  "dependencies": {  
    "body-parser": "~1.13.2",  
    "cookie-parser": "~1.3.5",  
    "debug": "~2.2.0",  
    "ejs": "~2.3.3",  
    "express": "~4.13.1",  
    "morgan": "~1.6.1",  
    "serve-favicon": "~2.3.0"  
  }  
}
```



4. Database(MySQL) 연동

- MySQL 설치
- <http://dev.mysql.com/downloads/mysql/>
 - 본인 컴퓨터 환경에 맞는 mysql 설치
- Mysql 5.7 (windows 32bit MSI)

MySQL Community Server 5.7.11

Select Platform:
Microsoft Windows ▼

Recommended Download:

MySQL Installer 5.7
for Windows

**All MySQL Products. For All Windows Platforms.
In One Package.**

Starting with MySQL 5.6 the MySQL Installer package replaces the server-only MSI packages.

Windows (x86, 32-bit), MySQL Installer MSI

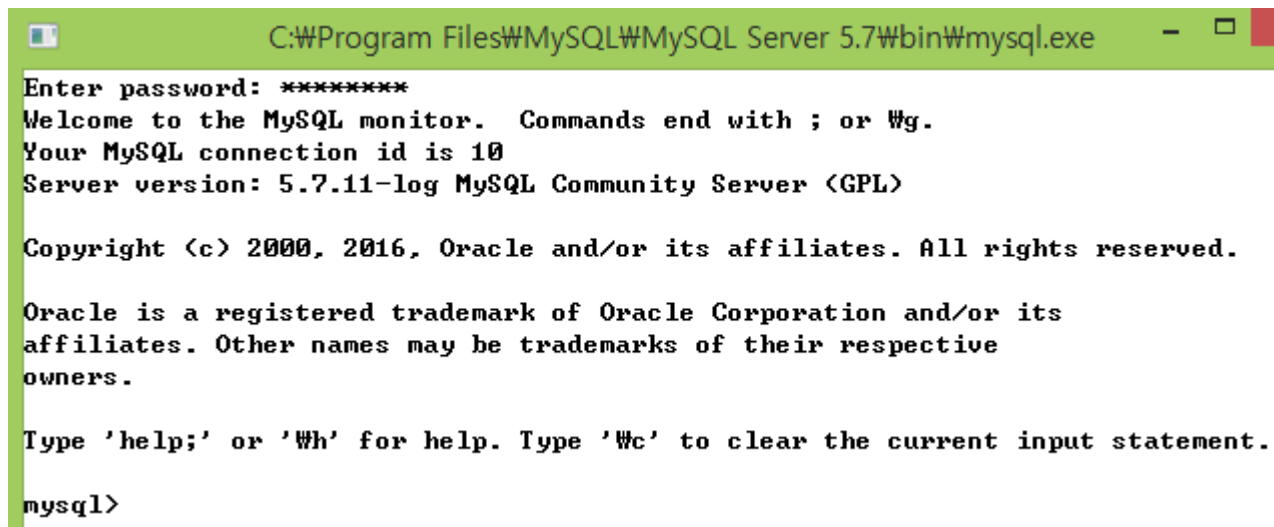
Download





4. Database(MySQL) 연동

- 설치된 MySQL 5.7 Command Line Client 실행
 - 비밀번호를 입력하고 접속되는 것을 확인



```
C:\Program Files\MySQL\MySQL Server 5.7\bin\mysql.exe
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.11-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

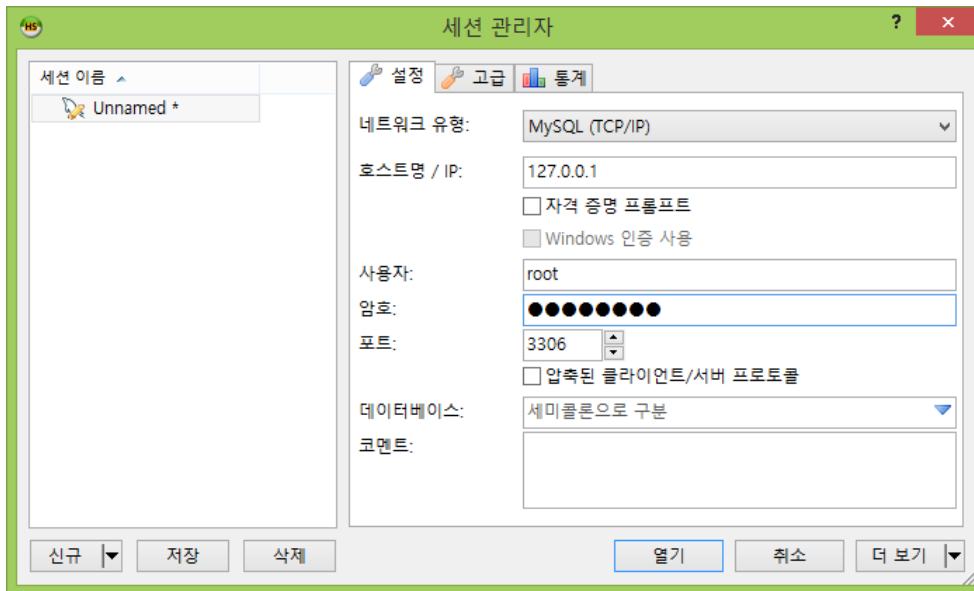
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



4. Database(MySQL) 연동

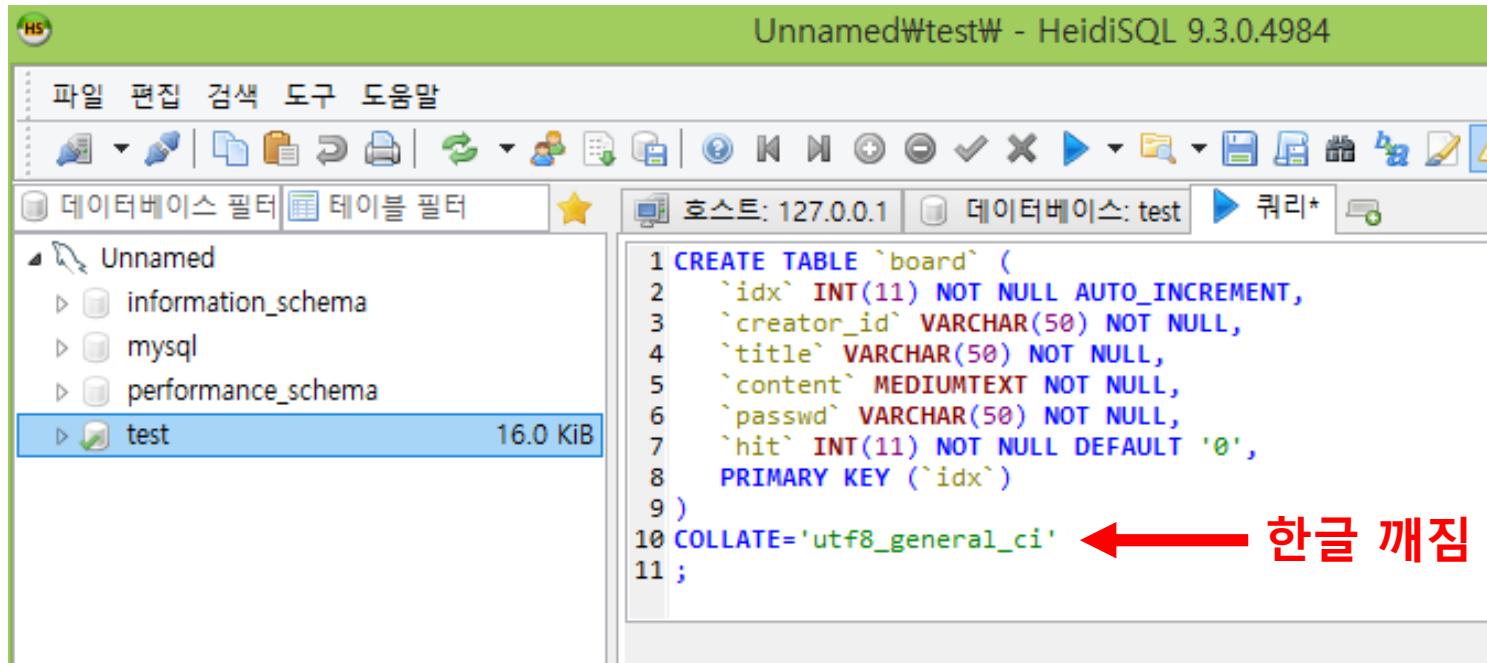
- HeidiSQL 설치
 - <http://www.heidisql.com/download.php>
- 신규 등록





4. Database(MySQL) 연동

- test database 선택 후 쿼리문 실행



UnnamedWtestW - HeidiSQL 9.3.0.4984

파일 편집 검색 도구 도움말

데이터베이스 필터 테이블 필터

호스트: 127.0.0.1 데이터베이스: test 쿼리*

Unnamed

- information_schema
- mysql
- performance_schema
- test 16.0 KiB

```
1 CREATE TABLE `board` (  
2   `idx` INT(11) NOT NULL AUTO_INCREMENT,  
3   `creator_id` VARCHAR(50) NOT NULL,  
4   `title` VARCHAR(50) NOT NULL,  
5   `content` MEDIUMTEXT NOT NULL,  
6   `passwd` VARCHAR(50) NOT NULL,  
7   `hit` INT(11) NOT NULL DEFAULT '0',  
8   PRIMARY KEY (`idx`)  
9 )  
10 COLLATE='utf8_general_ci'  
11 ;
```

← 한글 깨짐 방지

4. Database(MySQL) 연동

- 새로고침 후 board table이 생긴 것을 확인

UnnamedWtestWboardW - HeidiSQL 9.3.0.4984

파일 편집 검색 도구 도움말 Become a donor of the HeidiSQL project Update available

데이터베이스 필터 테이블 필터

호스트: 127.0.0.1 데이터베이스: test 테이블: board 데이터 쿼리*

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

이름: board

코멘트:

test 16.0 KiB

board 16.0 KiB

추가 제거 위로 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으...	기본값
1	idx	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCRE
2	creator_id	VARCHAR	50	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	기본값 없음
3	title	VARCHAR	50	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	기본값 없음
4	content	MEDIUMTEXT		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	기본값 없음
5	passwd	VARCHAR	50	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	기본값 없음
6	hit	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0

4. Database(MySQL) 연동

■ 테스트 데이터 입력

UnnamedWtestWboardW - HeidiSQL 9.3.0.4984

파일 편집 검색 도구 도움말

데이터베이스 필 테이블 필터

호스트: 127.0.0.1 데이터베이스: test 테이블: board 데이터 쿼리+

test: test

```

1 INSERT INTO board (`idx`, `creator_id`, `title`, `content`, `passwd`, `hit`) VALUES
2 (1, 'user1', 'title1', 'hello', '1234', 0),
3 (2, 'user2', 'title2', 'bye', '1234', 0);
4
5

```

test.board: 2 행 (총) (대략적)

idx	creator_id	title	content	passwd	hit
1	user1	title1	hello	1234	0
2	user2	title2	bye	1234	0



4. Database(MySQL) 연동

- >npm install mysql

```
C:\WINDOWS\system32\cmd.exe
D:\Users\User\Documents\myNode\joinForm>npm install mysql
mysql@2.10.2 node_modules\mysql
├── bignumber.js@2.1.4
├── readable-stream@1.1.13 (inherits@2.0.1, isarray@0.0.1,
0.31, core-util-is@1.0.2)
```




4. Database(MySQL) 연동

■ routesWindex.js 수정

```
1 var express = require('express');
2 var router = express.Router();
3 var mysql = require('mysql');
4 var pool = mysql.createPool({
5   connectionLimit: 5,
6   host: 'localhost',
7   user: 'root',
8   database: 'test',
9   password: '비밀번호'
10 });
11
12 /* GET home page. */
13 router.get('/', function(req, res, next) {
14   pool.getConnection(function (err, connection) {
15     // Use the connection
16     connection.query('SELECT * FROM board', function (err, rows) {
17       if (err) console.error("err : " + err);
18       console.log("rows : " + JSON.stringify(rows));
19
20       res.render('index', {title: 'test', rows: rows});
21       connection.release();
22
23       // Don't use the connection here, it has been returned to the pool.
24     });
25   });
26 });
27
28 module.exports = router;
```



4. Database(MySQL) 연동

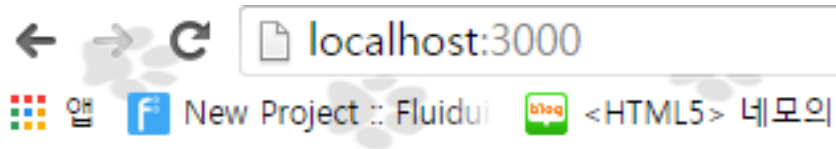
■ views#index.ejs 수정

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title><%= title %></title>
5   <link rel='stylesheet' href='/stylesheets/style.css' />
6 </head>
7 <body>
8 <h1><%= title %></h1>
9
10 <p>Welcome to <%= title %></p>
11
12 <p><%= rows[0].idx %></p>
13
14 <p><%= rows[0].creator_id %></p>
15
16 <p><%= rows[0].title %></p>
17
18 <p><%= rows[0].content %></p>
19
20 <p><%= rows[0].regdate %></p>
21 </body>
22 </html>
23
```



4. Database(MySQL) 연동

- 출력 결과를 분석 및 확인



test

Welcome to test

1

user1

title1

hello



5. 게시판 구축

- URL 설계(GET Method)
 - 게시판 글 목록 표시 화면
 - <http://localhost/board/list/?page=1>
 - 게시판 글 조회 화면
 - <http://localhost/board/read/?idx=1>
 - 게시판 글 쓰기 화면
 - <http://localhost/board/write>
 - 게시판 글 수정 화면
 - <http://localhost/board/modify>
- URL 설계(POST Method)
 - 게시판 글 쓰기 액션
 - <http://localhost/board/write>
 - 게시판 글 수정 액션
 - <http://localhost/board/modify>



5. 게시판 구축

- app.js 소스 추가

```
8 var routes = require('./routes/index');  
9 var users = require('./routes/users');  
10 var join = require('./routes/joinForm');  
11 var board = require('./routes/board');  
  
27 app.use('/', routes);  
28 app.use('/users', users);  
29 app.use('/join', join);  
30 app.use('/board', board);
```



5. 게시판 구축

- routes#board.js 생성
- users.js, index.js를
이용해 작성

```
1 var express = require('express');
2 var router = express.Router();
3
4 // MySQL 로드
5 var mysql = require('mysql');
6 var pool = mysql.createPool({
7   connectionLimit: 5,
8   host: 'localhost',
9   user: 'root',
10  database: 'test',
11  password: '비밀번호'
12 });
13
14 /* GET users listing. */
15 router.get('/', function(req, res, next) {
16   // 그냥 board/ 로 접속할 경우 전체 목록 표시로 리다이렉팅
17   res.redirect('/board/list/1');
18 });
19
20 router.get('/list/:page', function(req,res,next){
21
22   pool.getConnection(function (err, connection) {
23     // Use the connection
24     var sqlForSelectList = "SELECT idx, creator_id, title, hit FROM board";
25     connection.query(sqlForSelectList, function (err, rows) {
26       if (err) console.error("err : " + err);
27       console.log("rows : " + JSON.stringify(rows));
28
29       res.render('list', {title: ' 게시판 전체 글 조회', rows: rows});
30       connection.release();
31
32       // Don't use the connection here, it has been returned to the pool.
33     });
34   });
35 });
36
37 module.exports = router;
```



5. 게시판 구축

■ views\list.ejs 생성

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title><%= title %></title>
5   <link rel='stylesheet' href='/stylesheets/style.css' />
6 </head>
7 <body>
8 <h1><%= title %></h1>
9
10 <a href="/board/write">글쓰기로 이동</a>
11 <br>
12 <br>
13
14 <table border="1">
15   <tr>
16     <td>번호</td>
17     <td>작성자</td>
18     <td>제목</td>
19     <td>조회수</td>
20   </tr>
21
```

< 소스코드 1/2 >



5. 게시판 구축

■ views#wlist.ejs 생성

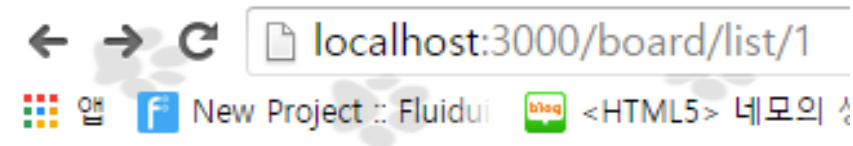
```
22 <%  
23     for(var i=0; i<rows.length; i++)  
24     {  
25         var oneItem = rows[i];  
26     %>  
27     <tr>  
28         <td><%=oneItem.idx%></td>  
29         <td><%=oneItem.creator_id%></td>  
30         <td><%=oneItem.title%></td>  
31         <td><%=oneItem.hit%></td>  
32     </tr>  
33  
34 <%  
35     }  
36 %>  
37 </table>  
38  
39 </body>  
40 </html>
```

< 소스코드 2/2 >



5. 게시판 구축

■ 결과 확인



게시판 전체 글 조회

[글쓰기로 이동](#)

번호	작성자	제목	조회수
1	user1	title1	0
2	user2	title2	0



5. 게시판 구축

- 글쓰기
- views#write.ejs

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/javascripts/jquery-1.12.2.min.js"></script>
5   <title><%= title %></title>
6   <link rel='stylesheet' href='/stylesheets/style.css' />
7 </head>
8 <body>
9 <h1><%= title %></h1>
10
11 <form action="/board/write" method="post" onsubmit="return onWriteSubmit()">
12   <table border="1">
13     <tr>
14       <td>작성자</td>
15       <td><input type="text" name="creator_id" id="creator_id" required/></td>
16     </tr>
17     <tr>
18       <td>제목</td>
19       <td><input type="text" name="title" id="title" required/></td>
20     </tr>
21     <tr>
22       <td>내용</td>
23       <td><textarea name="content" id="content" cols="30" rows="10" required></textarea></td>
24     </tr>
25     <tr>
26       <td>패스워드</td>
27       <td><input type="password" name="passwd" id="passwd" required/></td>
28     </tr>
29     <tr>
30       <td colspan="2">
31         <button type="submit">글쓰기</button>
32       </td>
33     </tr>
34   </table>
35 </form>
36
```

< 소스코드 1/3 >



5. 게시판 구축

■ 글쓰기

```
37 <script>
38     function onWriteSubmit()
39     {
40         if ( $("#creator_id").val().trim() == "" )
41         {
42             var message = "아이디를 입력해 주세요";
43             $("#creator_id").val("");
44             $("#creator_id").focus();
45             alert(message);
46             return false;
47         }
48
49         if ( $("#title").val().trim() == "" )
50         {
51             var message = "제목을 입력해 주세요";
52             $("#title").val("");
53             $("#title").focus();
54             alert(message);
55             return false;
56         }
57
58         if ( $("#content").val().trim() == "" )
59         {
60             var message = "본문 내용을 입력해 주세요";
61             $("#content").val("");
62             $("#content").focus();
63             alert(message);
64             return false;
65         }
66     }
```

< 소스코드 2/3 >



5. 게시판 구축

■ 글쓰기

```
66
67     if ( $("#passwd").val().trim() == "" )
68     {
69         var message = "패스워드를 입력해 주세요";
70         $("#passwd").val("");
71         $("#passwd").focus();
72         alert(message);
73         return false;
74     }
75 }
76 </script>
77
78 </body>
79 </html>
```

< 소스코드 3/3 >

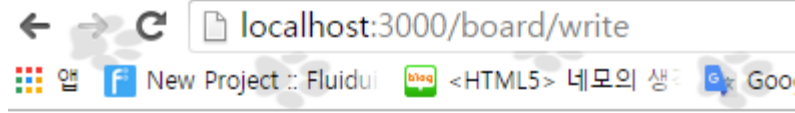
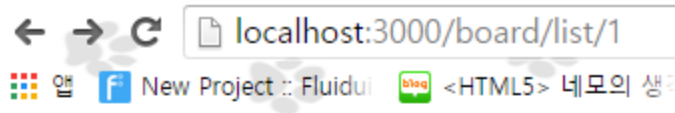
■ board.js에 소스 추가

```
37 // 글쓰기 화면 표시 GET
38 router.get('/write', function(req,res,next){
39     res.render('write',{title : "게시판 글 쓰기"});
40 });
```



5. 게시판 구축

글쓰기 결과 확인



게시판 전체 글 조회

[글쓰기로 이동](#)

번호	작성자	제목	조회수
1	user1	title1	0
2	user2	title2	0

게시판 글 쓰기

작성자	<input type="text"/>
제목	<input type="text"/>
내용	<div><div></div></div>
패스워드	<input type="password"/>
<input type="button" value="글쓰기"/>	



5. 게시판 구축

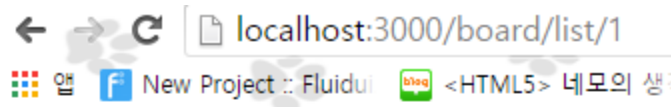
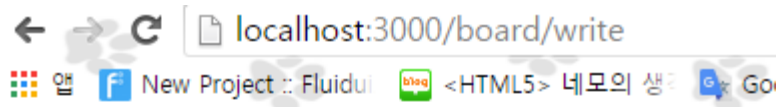
■ board.js에 POST 추가

```
42 // 글쓰기 로직 처리 POST
43 router.post('/write', function(req,res,next){
44
45     var creator_id = req.body.creator_id;
46     var title = req.body.title;
47     var content = req.body.content;
48     var passwd = req.body.passwd;
49     var datas = [creator_id,title,content,passwd];
50
51     pool.getConnection(function (err, connection)
52     {
53         // Use the connection
54         var sqlForInsertBoard = "insert into board(creator_id, title, content, passwd) values(?,?,?,?)";
55         connection.query(sqlForInsertBoard,datas, function (err, rows) {
56             if (err) console.error("err : " + err);
57             console.log("rows : " + JSON.stringify(rows));
58
59             res.redirect('/board');
60             connection.release();
61
62             // Don't use the connection here, it has been returned to the pool.
63         });
64     });
65
66 });
67
68 module.exports = router;
```



5. 게시판 구축

■ 결과 확인



게시판 글 쓰기

작성자	user3
제목	글쓰기테스트
내용	테스트 테스트
패스워드
글쓰기	

게시판 전체 글 조회

[글쓰기로 이동](#)

번호	작성자	제목	조회수
1	user1	title1	0
2	user2	title2	0
3	user3	글쓰기테스트	0



5. 게시판 구축

- 조회
- views/wread.ejs 생성

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/javascripts/jquery-1.12.2.min.js"></script>
5   <title><%= title %></title>
6   <link rel='stylesheet' href='/stylesheets/style.css' />
7 </head>
8 <body>
9 <h1><%= title %></h1>
10
11 <form action="/board/update" method="get">
12   <table border="1">
13     <input type="hidden" name="idx" value="<%=row.idx%>" />
14     <tr>
15       <td>작성자</td>
16       <td><%=row.creator_id%></td>
17     </tr>
18     <tr>
19       <td>제목</td>
20       <td><%=row.title%></td>
21     </tr>
```

< 소스코드 1/2 >



5. 게시판 구축

■ 조회

```
22         <tr>
23             <td>내용</td>
24             <td><%=row.content%></td>
25         </tr>
26         <tr>
27             <td>조회수</td>
28             <td><%=row.hit%></td>
29         </tr>
30         <tr>
31             <td colspan="2">
32                 <button type="submit">글 수정</button>
33                 <a href="/board">리스트로 돌아가기</a>
34             </td>
35         </tr>
36     </table>
37 </form>
38
39 </body>
40 </html>
```

< 소스코드 2/2 >



5. 게시판 구축

■ 조회

- board.js 파일에 아래 소스 추가

```
68 // 글조회 로직 처리 GET
69 router.get('/read/:idx',function(req,res,next)
70 {
71     var idx = req.params.idx;
72
73     pool.getConnection(function(err,connection)
74     {
75         var sql = "select idx, creator_id, title, content, hit from board where idx=?";
76         connection.query(sql,[idx], function(err,row)
77         {
78             if(err) console.error(err);
79             console.log("17# 글 조회 결과 확인 : ",row);
80             res.render('read', {title:"글 조회", row:row[0]});
81             connection.release();
82         });
83     });
84 });
85 });
```



5. 게시판 구축

■ 조회

- list.ejs 파일에 글 조회 링크 추가

```
22 <%
23     for(var i=0; i<rows.length; i++)
24     {
25         var oneItem = rows[i];
26     %>
27     <tr>
28         <td><%=oneItem.idx%></td>
29         <td><%=oneItem.creator id%></td>
30         <td><a href="/board/read/<%=oneItem.idx%>"><%=oneItem.title%></a></td>
31         <td><%=oneItem.hit%></td>
32     </tr>
33
34 <%
35     }
36 %>
37 </table>
38
39 </body>
40 </html>
```



5. 게시판 구축

■ 업데이트 조회

- viewsWupdate.ejs 생성

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="/javascripts/jquery-1.12.2.min.js"></script>
5   <title><%= title %></title>
6   <link rel='stylesheet' href='/stylesheets/style.css' />
7 </head>
8 <body>
9 <h1><%= title %></h1>
10
11 <form action="/board/update" method="post" onsubmit="return onWriteSubmit()">
12   <table border="1">
13     <input type="hidden" name="idx" value="<%=row.idx%>" />
14     <tr>
15       <td>작성자</td>
16       <td><input type="text" name="creator_id" id="creator_id" value="<%=row.creator_id%>" required/></td>
17     </tr>
18     <tr>
19       <td>제목</td>
20       <td><input type="text" name="title" id="title" value="<%=row.title%>" required/></td>
21     </tr>
22     <tr>
23       <td>내용</td>
24       <td><textarea name="content" id="content" cols="30" rows="10" required><%=row.content%></textarea></td>
25     </tr>
26     <tr>
27       <td>패스워드</td>
28       <td><input type="password" name="passwd" id="passwd" required/></td>
29     </tr>
```



5. 게시판 구축

■ 업데이트 조회

```

30         <tr>
31             <td colspan="2">
32                 <button type="submit">글쓰기</button>
33             </td>
34         </tr>
35     </table>
36 </form>
37
38 <script>
39     function onWriteSubmit()
40     {
41         if ( $("#creator_id").val().trim() == "" )
42         {
43             var message = "아이디를 입력해 주세요";
44             $("#creator_id").val("");
45             $("#creator_id").focus();
46             alert(message);
47             return false;
48         }
49
50         if ( $("#title").val().trim() == "" )
51         {
52             var message = "제목을 입력해 주세요";
53             $("#title").val("");
54             $("#title").focus();
55             alert(message);
56             return false;
57         }
58     }

```



5. 게시판 구축

■ 업데이트 조회

```
59         if ( $("#content").val().trim() == "" )
60         {
61             var message = "본문 내용을 입력해 주세요";
62             $("#content").val("");
63             $("#content").focus();
64             alert(message);
65             return false;
66         }
67
68         if ( $("#passwd").val().trim() == "" )
69         {
70             var message = "패스워드를 입력해 주세요";
71             $("#passwd").val("");
72             $("#passwd").focus();
73             alert(message);
74             return false;
75         }
76     }
77 </script>
78
79 </body>
80 </html>
```

< 소스코드 3/3 >



5. 게시판 구축

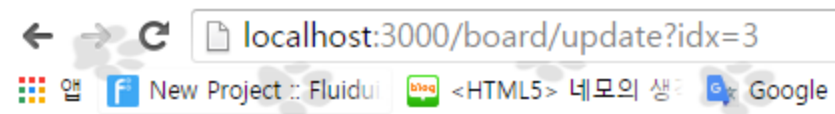
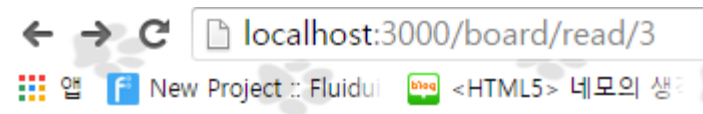
- 업데이트 조회
 - board.js 파일에 아래 소스 추가

```
87 // 글수정 화면 표시 GET
88 router.get('/update',function(req,res,next)
89 {
90     var idx = req.query.idx;
91
92     pool.getConnection(function(err,connection)
93     {
94         if(err) console.error("커넥션 객체 얻어오기 에러 : ",err);
95
96         var sql = "select idx, creator_id, title, content, hit from board where idx=?";
97         connection.query(sql, [idx], function(err,rows)
98         {
99             if(err) console.error(err);
100             console.log("update에서 1개 글 조회 결과 확인 : ",rows);
101             res.render('update', {title:"글 수정", row:rows[0]});
102             connection.release();
103         });
104     });
105
106 });
```



5. 게시판 구축

- 업데이트 조회
 - 수정 화면 확인



글 조회

작성자	user3
제목	글쓰기테스트
내용	테스트 테스트
조회수	0
글 수정	리스트로 돌아가기

글 수정

작성자	<input type="text" value="user3"/>
제목	<input type="text" value="글쓰기테스트"/>
내용	<div>테스트 테스트</div>
패스워드	<input type="password"/>
글쓰기	<input type="button" value="글쓰기"/>



5. 게시판 구축

■ 업데이트 쓰기

- board.js 파일에 아래 소스 추가

```
108 // 글수정 로직 처리 POST
109 router.post('/update',function(req,res,next)
110 {
111     var idx = req.body.idx;
112     var creator_id = req.body.creator_id;
113     var title = req.body.title;
114     var content = req.body.content;
115     var passwd = req.body.passwd;
116     var datas = [creator_id,title,content,passwd];
117
118     pool.getConnection(function(err,connection)
119     {
120         var sql = "update board set creator_id=?, title=?, content=? where idx=? and passwd=?";
121         connection.query(sql,[creator_id,title,content,idx,passwd],function(err,result)
122         {
123             console.log(result);
124             if(err) console.error("글 수정 중 에러 발생 err : ",err);
125
126             if(result.affectedRows == 0)
127             {
128                 res.send("<script>alert('패스워드가 일치하지 않거나, 잘못된 요청으로 인해 값이 변경되지 않았습니다.');
```



5. 게시판 구축

- 업데이트 쓰기
 - 업데이트 결과 확인

← → ↺ localhost:3000/board/read/3

앱 New Project :: FluidUI <HTML5> 네모의 생

글 조회

작성자	user3
제목	글쓰기테스트
내용	테스트 테스트
조회수	0
글 수정	리스트로 돌아가기

← → ↺ localhost:3000/board/update?idx=3

앱 New Project :: FluidUI <HTML5> 네모의 생 Goo

글 수정

작성자	user3
제목	글쓰기테스트
내용	수정 수정됨
패스워드
글 쓰기	

← → ↺ localhost:3000/board/read/3

앱 New Project :: FluidUI <HTML5> 네모의 생

글 조회

작성자	user3
제목	글쓰기테스트
내용	수정수정됨
조회수	0
글 수정	리스트로 돌아가기



6. MySQL 기본

- show databases
 - 데이터베이스 리스트 확인

```
C:\Program Files\MySQL\MySQL Server 5.5\bin> mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
6 rows in set (0.00 sec)

mysql>
```



6. MySQL 기본

- create database [DB명]
 - 데이터베이스를 선택

```
C:\Program Files\MySQL\bin> mysql> create database basic;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| basic |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
7 rows in set (0.00 sec)
```



6. MySQL 기본

- use [DB명]
 - 데이터베이스를 선택



C:\WP

```
mysql> use basic;  
Database changed  
mysql>
```



6. MySQL 기본

- create table
 - 테이블 생성




C:\Program Files\MySQL\MySQL

```
mysql> create table sample(  
  -> item varchar(20),  
  -> price int);  
Query OK, 0 rows affected (0.04 sec)  
  
mysql>
```



6. MySQL 기본

- show tables
 - 테이블 리스트 출력

 C:\WProgram

```
mysql> show tables;  
+-----+  
| Tables_in_basic |  
+-----+  
| sample          |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```



6. MySQL 기본

- desc [table명]
 - 테이블 속성 확인

```
C:\Program Files\MySQL\MySQL Server 5.7\bin
mysql> desc sample;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| item  | varchar(20)   | YES  |     | NULL    |       |
| price | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```




6. MySQL 기본

- insert
 - 데이터 삽입



C:\Program Files\MySQL\MySQL Server 5.7\bin\mysql>

```
mysql> insert into sample(item, price) value('사과', 2000);  
Query OK, 1 row affected (0.00 sec)
```



6. MySQL 기본

- select
 - 데이터 출력

C:\Program Files\MySQL

```
mysql> select * from sample;
```

```
+-----+-----+  
| item  | price |  
+-----+-----+  
| 사과  | 2000  |  
+-----+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select item from sample;
```

```
+-----+  
| item  |  
+-----+  
| 사과  |  
+-----+
```

```
1 row in set (0.00 sec)
```



6. MySQL 기본

- select ... where
 - 데이터 조건 출력

```
C:\Program Files\MySQL\MySQL Ser
mysql> select * from sample where item='사과';
+-----+-----+
| item  | price |
+-----+-----+
| 사과  | 2000  |
+-----+-----+
1 row in set (0.00 sec)

mysql> select * from sample where item='포도';
Empty set (0.00 sec)
```



6. MySQL 기본

- update
 - 데이터 수정

```
C:\Program Files\MySQL\MySQL Server 5.7\bin
mysql> select * from sample where item='사과';
+-----+-----+
| item  | price |
+-----+-----+
| 사과  | 2000  |
+-----+-----+
1 row in set (0.00 sec)

mysql> update sample set price=3000 where item='사과';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from sample where item='사과';
+-----+-----+
| item  | price |
+-----+-----+
| 사과  | 3000  |
+-----+-----+
1 row in set (0.00 sec)
```



6. MySQL 기본

- delete
 - 데이터 삭제

```
C:\Program Files\MySQL\MySQL
mysql> select * from sample;
+-----+-----+
| item  | price |
+-----+-----+
| 사과  | 3000  |
| 포도  | 1500  |
+-----+-----+
2 rows in set (0.00 sec)

mysql> delete from sample where item='사과';
Query OK, 1 row affected (0.01 sec)

mysql> select * from sample;
+-----+-----+
| item  | price |
+-----+-----+
| 포도  | 1500  |
+-----+-----+
1 row in set (0.00 sec)
```