

# 이화이언

## 기술설계팀 스터디

2020.06.06

[ 생활코딩 Node.js - MySQL ]

37 기 수습운영진 김윤서

## 1. VSCode 에서 node.js - MySQL 연결하기

### 1) MySQL

sql server(mssql) 깔고, reload 까지 완료한 후

Terminal 창에서

➤ **cd C:\Bitnami\wampstack-7.4.6-1\mysql\bin**

➤ **.\mysql -uroot -p**

쳐서 연결.

➤ **CREATE DATABASE opentutorials;** // 스키마 만들고(이미 있으면 USE opentu~)

➤ **SHOW DATABASE;** // 등등 하면 Mysql 과 똑같이 나옴

### 2) Node.js

npm install 명령어 이용해서 package.json에 있는 sanitize-html 파일들을 깔면, node\_module 이라는 폴더에 파일들이 생긴다.

➤ **node main.js**

➤ **pm2 start main.js --watch** //pm2 와 연결- 바로 변경사항 확인 할 수 있도록

➤ **pm2 log**

// 등등 node.js 명령어 Terminal 창에서 모두 작동.

```

Microsoft Windows [version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\A\Desktop\node.js-mysql-1>^Z^Z
C:\Users\A\Desktop\node.js-mysql-1>npm install
npm WARN web2-nodejs@1.0.0 No description

added 34 packages from 52 contributors and audited 34 packages in 2.698s
found 1 high severity vulnerability
  run `npm audit fix` to fix them, or `npm audit` for details

C:\Users\A\Desktop\node.js-mysql-1>node main.js
^C
C:\Users\A\Desktop\node.js-mysql-1>pm2 start main.js --watch
[PM2] Spawning PM2 daemon with pm2_home=C:\Users\A\.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting C:\Users\A\Desktop\node.js-mysql-1\main.js in fork_mode
(1 instance)
[PM2] Done.

```

id	name	mode	U	status	cpu	memory
0	main	fork	0	online	0%	29.6mb

```

C:\Users\A\Desktop\node.js-mysql-1>pm2 log
[TAILING] Tailing last 15 lines for [all] processes (change the value with --lines option)
C:\Users\A\.pm2\pm2.log last 15 lines:
PM2 | 2020-06-06T21:14:54: PM2 log: Node.js version : 12.16
.3
PM2 | 2020-06-06T21:14:54: PM2 log: Current arch : x64

```

//연결 완료

## 2. Node.js 에서 mysql 제어할 수 있는 모듈 설치

<https://www.npmjs.com/package/mysql> 참고.

Terminal창에 **npm install --save mysql** 하면 package.json에 mysql 깔림.

## 3. mysql의 데이터를 Node.js 로 로드 해오는 방법

//위 페이지에 나온 코드 [mysql.js] 파일 만들어서 복사

```
example.sql  {} package.json  JS mysql.js  ●
nodejs > JS mysql.js > ...
1  var mysql      = require('mysql');
2  var connection = mysql.createConnection({
3    host        : 'localhost',
4    user        : 'root',
5    password    : 'root1',
6    database    : 'opentutorials'
7  });
8
9  connection.connect();
10
11 connection.query('SELECT 1 + 1 AS solution', function (error, results, fields) {
12   if (error) throw error;
13   console.log('The solution is: ', results[0].solution);
14 });
15
16 connection.end();
```

쿼리문을 다이나믹하게 바꿈으로써 mysql 이용해서 nodejs 제어하는 것

connection.query( 안에 db의 sql문이 들어감 ); //

[main.js] - 기본 문법에 대하여...

```
db.query(`SELECT * FROM topic`, function(error, topics){
  console.log(topics);
  response.writeHead(200);
  response.end(`Success`);
});
```

**db.query('SELECT \* FROM topic' , function(error, topics){ 응답 결과 이용하는 코드 });**

.query( )문의 **2번째 인자**로는, 이 sql문이 실행된 후에 서버가 응답하면 응답 결과를 우리가 처리

할 수 있도록 callback 함수를 정의하도록 되어있음

\* 콜백 func의 시그니처 -> (error, results)

: **error**에 오류가 났을 경우가, / 두번째 인자 results엔 sql의 결과가 담기도록 약속되어있음

#### 4. 이 데이터로 웹페이지 구현하기

lib 디렉토리의 template.js 파일 : 웹페이지 기본 틀이 약속되어있음

```
module.exports = {
  HTML:function(title, list, body, control){
    return `
    <!doctype html>
    <html>
    <head>
      <title>WEB1 - ${title}</title>
      <meta charset="utf-8">
    </head>
    <body>
      <h1><a href="/">WEB</a></h1>
      ${list}
      ${control}
      ${body}
    </body>
    </html>
    `;
  },list:function(topics){
    var list = '<ul>';
    var i = 0;
    while(i < topics.length){
      list = list + `<li><a href="/?id=${topics[i].id}">${topics[i].title}</a></li>`;
      i = i + 1;
    }
    list = list+'</ul>';
    return list;
  }
}
```

SQL 커맨트 창에서, SELECT \* FROM topic WHERE id = 3;

```
Database changed
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> SELECT * FROM topic WHERE id=3;
+----+-----+-----+-----+-----+
| id | title   | description | created      | author_id |
+----+-----+-----+-----+-----+
| 3  | SQL Server | SQL Server is ... | 2018-01-20 11:01:10 | 2         |
+----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

=> 웹페이지( nodejs로?) 에서 구현하려면?

```
db.query(`SELECT * FROM topic`, function(error, topics){
  if(error){
    throw error;
  } //에러 발생한 경우, 아래 진행하지 않고 콘솔에 에러내용 보여줌
  db.query(`SELECT * FROM topic WHERE ${queryData.id}`, function(error2, topics){
    if(error2){
      throw error2;
    }
    var title = 'Welcome';
    var description = 'Hello, Node.js';
    var list = template.list(filelist);
    var html = template.HTML(title, list,
      `

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


```