

MySQL

MySQL CRUD

- ❖ MySQL에 접속해서 샘플 데이터 생성

-- 데이터베이스 확인

```
show databases;
```

-- 데이터베이스 생성

```
create database adam;
```

-- 데이터베이스 사용 설정

```
use adam;
```

-- 테이블이 존재하는 경우만 수행

```
DROP TABLE goods;
```

```
CREATE TABLE goods(
```

```
itemid int,
```

```
itemname VARCHAR(100),
```

```
price int,
```

```
description VARCHAR(200),
```

```
pictureurl VARCHAR(100),
```

```
updatedate varchar(20),
```

```
PRIMARY KEY (itemid)
```

```
)engine=InnoDB DEFAULT CHARSET=utf8;
```

MySQL CRUD

❖ MySQL에 접속해서 샘플 데이터 생성

```
insert into goods values(1, '레몬', 500,'비타민 C가 풍부한 쓴귤', 'lemon.jpg', '2020-08-01');
insert into goods values(2, '오렌지', 1500, '비타민 C가 풍부한 당귤', 'orange.jpg', '2020-08-01');
insert into goods values(3, '키위', 2000, '비타민 C가 풍부한 다래', 'kiwi.jpg', '2020-08-01');
insert into goods values(4, '포도', 1000, '항상화 성분과 당분이 높고 무기물이 많은 과일',
'grape.jpg', '2020-08-01');
insert into goods values(5, '딸기', 2000, '수분함량이 높은 과일', 'strawberry.jpg', '2020-08-01');
insert into goods values(6, '무화과', 300, '칼슘, 섬유질 및 항산화 물질을 많이 함유된 식물',
'fig.jpg', '2020-08-01');
insert into goods values(7, '레몬', 500,'비타민 C가 풍부한 쓴귤', 'lemon.jpg', '2020-08-01');
insert into goods values(8, '오렌지', 1500, '비타민 C가 풍부한 당귤', 'orange.jpg', '2020-08-01');
insert into goods values(9, '키위', 2000, '비타민 C가 풍부한 다래', 'kiwi.jpg', '2020-08-01');
insert into goods values(10, '포도', 1000, '항상화 성분과 당분이 높고 무기물이 많은 과일',
'grape.jpg', '2020-08-01');
insert into goods values(11, '딸기', 2000, '수분함량이 높은 과일', 'strawberry.jpg', '2020-08-01');
insert into goods values(12, '무화과', 300, '칼슘, 섬유질 및 항산화 물질을 많이 함유된 식물',
'fig.jpg', '2020-08-01');
insert into goods values(13, '레몬', 500,'비타민 C가 풍부한 쓴귤', 'lemon.jpg', '2020-08-01');
insert into goods values(14, '오렌지', 1500, '비타민 C가 풍부한 당귤', 'orange.jpg', '2020-08-01');
insert into goods values(15, '키위', 2000, '비타민 C가 풍부한 다래', 'kiwi.jpg', '2020-08-01');
```

MySQL CRUD

❖ MySQL에 접속해서 샘플 데이터 생성

```
insert into goods values(16, '포도', 1000, '항상화 성분과 당분이 높고 무기물이 많은 과일',
    'grape.jpg', '2020-08-01');
insert into goods values(17, '딸기', 2000, '수분함량이 높은 과일', 'strawberry.jpg', '2020-08-01');
insert into goods values(18, '무화과', 300, '칼슘, 섬유질 및 항산화 물질을 많이 함유된 식물',
    'fig.jpg', '2020-08-01');
insert into goods values(19, '딸기', 2000, '수분함량이 높은 과일', 'strawberry.jpg', '2020-08-01');
insert into goods values(20, '무화과', 300, '칼슘, 섬유질 및 항산화 물질을 많이 함유된 식물',
    'fig.jpg', '2020-08-01');
insert into goods values(21, '레몬', 500, '비타민 C가 풍부한 쓴귤', 'lemon.jpg', '2020-08-01');
insert into goods values(22, '오렌지', 1500, '비타민 C가 풍부한 당귤', 'orange.jpg', '2020-08-01');
insert into goods values(23, '키위', 2000, '비타민 C가 풍부한 다래', 'kiwi.jpg', '2020-08-01');
insert into goods values(24, '포도', 1000, '항상화 성분과 당분이 높고 무기물이 많은 과일',
    'grape.jpg', '2020-08-01');
insert into goods values(25, '딸기', 2000, '수분함량이 높은 과일', 'strawberry.jpg', '2020-08-01');
insert into goods values(26, '무화과', 300, '칼슘, 섬유질 및 항산화 물질을 많이 함유된 식물',
    'fig.jpg', '2020-08-01');
insert into goods values(27, '레몬', 500, '비타민 C가 풍부한 쓴귤', 'lemon.jpg', '2020-08-01');
insert into goods values(28, '오렌지', 1500, '비타민 C가 풍부한 당귤', 'orange.jpg', '2020-08-01');
insert into goods values(29, '키위', 2000, '비타민 C가 풍부한 다래', 'kiwi.jpg', '2020-08-01');
insert into goods values(30, '포도', 1000, '항상화 성분과 당분이 높고 무기물이 많은 과일',
    'grape.jpg', '2020-08-01');

commit;
```

MySQL CRUD

- ❖ MySQL에 접속해서 샘플 데이터 생성

```
select * from goods;
```

MySQL CRUD

- ❖ 프로젝트 생성

- ❖ 필요한 모듈 설치

```
npm install express morgan multer mysql cookie-parser express-session express-mysql-session  
dotenv compression file-stream-rotator
```

```
npm install --save-dev nodemon
```

MySQL CRUD

- ❖ 데이터베이스 접속 확인

- ✓ index.js 파일을 생성하고 작성한 후 실행

```
//mysql 모듈 가져오기
```

```
const mysql = require('mysql');
```

```
//접속 정보 생성 - 실제 연결은 하지 않음
```

```
var connection = mysql.createConnection({  
    host : '127.0.0.1',  
    port : 3306,  
    user : 'adam',  
    password : 'wnddkd',  
    database : 'adam'  
});
```

```
//데이터베이스 연결
```

```
connection.connect(function(err) {  
    if (err) {  
        console.log('mysql connection error');  
        console.log(err);  
        throw err;  
    }  
});
```

```
console.log(connection)
```

MySQL CRUD

- ❖ 데이터베이스 접속 확인
 - ✓ 접속 에러
 - ◆ caching_sha2_password 오류

sqlMessage: 'Client does not support authentication protocol requested by server; consider upgrading MySQL client',

 - 해결 – 데이터베이스의 관리자 계정에 접속

```
ALTER USER '사용자이름'@'%' IDENTIFIED WITH mysql_native_password BY  
'비밀번호';  
FLUSH privileges;
```

MySQL CRUD

- ❖ 데이터베이스 접속 확인

- ✓ 접속 에러

- ◆ host 나 port 오류

- errno: 'ETIMEDOUT',

- code: 'ETIMEDOUT',

- host ip 주소 나 port 번호 확인

- ◆ 계정 오류

- sqlMessage: "Access denied for user 'root'@'172.17.0.1' (using password: YES)",

- 계정 확인

- ◆ 데이터베이스 오류

- sqlMessage: "Unknown database 'absd'",

- 데이터베이스를 생성하거나 이름 확인

MySQL CRUD

❖ 쿼리 실행 – SELECT 이외의 구문

```
connection.query('CREATE TABLE family (id INT(3) AUTO_INCREMENT, name VARCHAR(20),  
PRIMARY KEY(id))engine=InnoDB DEFAULT CHARSET=utf8');
```

```
connection.query('INSERT INTO family (name) VALUES (?)', "을지문덕");  
connection.query('INSERT INTO family (name) VALUES (?)', "강감찬");  
connection.query('INSERT INTO family (name) VALUES (?)', "윤관");  
connection.query('INSERT INTO family (name) VALUES (?)', "권율");
```

MySQL CRUD

❖ 쿼리 실행 – SELECT

```
connection.query('SELECT * FROM family', function(err, results, fields) {  
  if (err)  
    throw err;  
  console.log(JSON.stringify(results));  
  
  for(idx = 0; idx < results.length; idx++){  
    console.log(results[idx].id, ":" , results[idx].name);  
  }  
  
});
```

```
[{"id":1,"name":"을지문덕"}, {"id":2,"name":"강감찬"}, {"id":3,"name":"윤관"}, {"id":4,"name":"권율"}]  
1 : 을지문덕  
2 : 강감찬  
3 : 윤관  
4 : 권율
```

MySQL CRUD

❖ 프로젝트 설정

- ✓ package.json 파일의 scripts 옵션을 수정

```
"scripts": {  
    "start": "nodemon app",  
    "test": "echo \"Error: no test specified\" && exit 1"  
},
```

- ✓ 프로젝트에 update.txt 파일을 추가

- ✓ 프로젝트에 .env 파일을 생성하고 작성

PORT=9000

COOKIE_SECRET=item

HOST='127.0.0.1'

MYSQLPORT=3306

USERNAME='adam'

PASSWORD='wnddkd'

DATABASE='adam'

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인

- ✓ index.js 파일을 생성하고 작성

```
const express = require('express');
const morgan = require('morgan');
const compression = require('compression')
const path = require('path');
const mysql = require('mysql');

const cookieParser = require('cookie-parser');
const session = require("express-session");

const multer = require('multer');
const dotenv = require('dotenv');

dotenv.config()

//서버 설정
const app = express();
app.set('port', process.env.PORT);
```

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인

- ✓ index.js 파일을 생성하고 작성

```
//로그 출력 설정
var FileStreamRotator = require('file-stream-rotator')
var fs = require('fs')
var logDirectory = path.join(__dirname, 'log')

// ensure log directory exists
fs.existsSync(logDirectory) || fs.mkdirSync(logDirectory)

// create a rotating write stream
var accessLogStream = FileStreamRotator.getStream({
  date_format: 'YYYYMMDD',
  filename: path.join(logDirectory, 'access-%DATE%.log'),
  frequency: 'daily',
  verbose: false
})

// setup the logger
app.use(morgan('combined', {stream: accessLogStream}))
```

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인

- ✓ index.js 파일을 생성하고 작성

```
app.use(compression());
```

```
//post 방식의 파라미터 읽기
```

```
var bodyParser = require('body-parser')
app.use(bodyParser.json()); // to support JSON-encoded bodies
app.use(bodyParser.urlencoded({ // to support URL-encoded bodies
  extended: true
}));
```

```
var options = {
  host :process.env.HOST,
  port : process.env.MYSQLPORT,
  user :process.env.USERNAME,
  password : process.env.PASSWORD,
  database : process.env.DATABASE
};
```

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인

- ✓ index.js 파일을 생성하고 작성

```
const MySQLStore = require('express-mysql-session')(session);
```

```
app.use(  
    session({  
        secret: process.env.COOKIE_SECRET,  
        resave: false,  
        saveUninitialized: true,  
        store : new MySQLStore(options)  
    })  
);
```

MySQL CRUD

❖ 프로젝트 기본 설정 및 기본 요청 확인

- ✓ index.js 파일을 생성하고 작성

```
//파일 업로드를 위한 설정
//img 디렉토리를 연결
try {
    fs.readdirSync('public/img');
} catch (error) {
    console.error('img 폴더가 없으면 img 폴더를 생성합니다.');
    fs.mkdirSync('public/img');
}
//파일이름은 기본 파일 이름에 현재 시간을 추가해서 생성
const upload = multer({
    storage: multer.diskStorage({
        destination(req, file, done) {
            done(null, 'public/img/');
        },
        filename(req, file, done) {
            const ext = path.extname(file.originalname);
            done(null, path.basename(file.originalname, ext) + Date.now() + ext);
        },
    }),
    limits: { fileSize: 10 * 1024 * 1024 },
});
```

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인

- ✓ index.js 파일을 생성하고 작성

```
app.use('/', express.static('public'))
```

```
//파일 다운로드를 위한 설정
```

```
var util = require('util')
var mime = require('mime')
```

```
//데이터베이스 연결
```

```
var connection = mysql.createConnection(options);
```

```
connection.connect(function(err) {
```

```
    if (err) {
```

```
        console.log('mysql connection error');
```

```
        console.log(err);
```

```
        throw err;
```

```
}
```

```
});
```

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인

- ✓ index.js 파일을 생성하고 작성

```
app.get('/', (req, res, next) => {
    res.sendFile(path.join(__dirname, 'index.html'))
});

//에러가 발생한 경우 처리
app.use((err, req, res, next) => {
    console.error(err);
    res.status(500).send(err.message)
});

app.listen(app.get('port'), () => {
    console.log(app.get('port'), '번 포트에서 대기 중');
});
```

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인

- ✓ 프로젝트에 public 디렉토리 생성
 - ✓ public 디렉토리에 img 디렉토리를 만들고 이미지 파일 복사
 - ✓ public 디렉토리에 index.html 파일을 생성하고 작성

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>NODE_MySQL</title>
    <link rel='stylesheet' href="/css/common.css" />
  </head>
  <body>
    <h1>MySQL</h1>
    <a href="#" id="allbtn">데이터 전체 가져오기</a>
    <div id='content'></div>
    <div id='updatearea'></div>
  </body>
</html>
```

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인
 - ✓ public 디렉토리에 css 디렉토리를 생성하고 작성

```
a{  
    text-decoration: none;  
}
```

MySQL CRUD

- ❖ 프로젝트 기본 설정 및 기본 요청 확인
 - ✓ 프로젝트를 실행하고 웹 브라우저에 기본 URL을 작성하고 확인

← → ⌂ ⓘ localhost:9000

MySQL

데이터 전체 가져오기

MySQL CRUD

- ❖ 전체 데이터 가져오기

상품 목록 화면

전체 데이터 개수:30		
상품ID	상품명	가격
30	포도	1000원
29	키위	2000원
28	오렌지	1500원
27	레몬	500원
26	무화과	300원
25	딸기	2000원
24	포도	1000원
23	키위	2000원
22	오렌지	1500원
21	레몬	500원
20	무화과	300원
19	딸기	2000원
18	무화과	300원
17	딸기	2000원
16	포도	1000원
15	키위	2000원
14	오렌지	1500원
13	레몬	500원

MySQL CRUD

❖ 전체 데이터 가져오기

- ✓ index.js 파일에 전체 데이터 가져오는 요청을 생성

```
app.get('/item/all', (req, res, next) => {
    //전체 데이터 가져오기
    var list;
    connection.query('SELECT * FROM goods order by itemid desc', function(err, results,
    fields) {
        if (err){
            throw err;
        }
        list = results;
        //전체 데이터 개수 가져오기
        connection.query('SELECT count(*) cnt FROM goods', function(err, results, fields)
{
            if (err)
                throw err;
            console.log(list)
            res.json({'count':results[0].cnt, 'list':list});

        });
    });
});
```

MySQL CRUD

❖ 전체 데이터 가져오기

- ✓ index.html 파일에 스크립트 작성

```
<script>
    var content = document.getElementById('content');
    var updatearea = document.getElementById("updatearea");

    function all(){
        var request;
        request = new XMLHttpRequest();
        request.open('GET', '/item/all');
        request.send('');

        request.addEventListener('load', function(){
            var data = JSON.parse(request.responseText);
            var display = "<div align='center' class='body'>";
            display += "<h2>상품 목록 화면</h2>";
            display += "<table border='1'>";
            display += "<tr><td colspan='3' align='right'>전체 데이터 개수:</td></tr>";
            display += "<tr class='header'>";
            display += "<th align='center' width='80'>상품ID</th>";
            display += "<th align='center' width='320'>상품명</th>";
            display += "<th align='center' width='100'>가격</th>";
            display += "</tr>"

            content.innerHTML = display;
        })
    }
    all();
```

MySQL CRUD

❖ 전체 데이터 가져오기

- ✓ index.html 파일에 스크립트 작성

```
for(var idx in data.list){  
    var item = data.list[idx];  
  
    display += "<tr class='record'>";  
    display += "<td align='center'>" + item.itemid + "</td>";  
    display += "<td align='left'>" + item.itemname + "</td>";  
    display += "<td align='right'>" + item.price + "원</td>";  
    display += "</tr>"  
}  
  
display += "</table>";  
display += "</div>"  
  
content.innerHTML = display;  
updatearea.innerHTML = "";  
});  
}
```

MySQL CRUD

- ❖ 전체 데이터 가져오기

- ✓ index.html 파일에 스크립트 작성

```
document.getElementById('allbtn').addEventListener('click', function(e){  
    e.preventDefault();  
    all();  
  
});  
  
</script>
```

MySQL CRUD

- ❖ 전체 데이터 가져오기

- ✓ common.css 파일에 스크립트 추가

```
div.body{  
    overflow-y: auto;  
    scrollbar-face-color: #C9BFED;  
    scrollbar-shadow-color: #EDEDED;  
    margin-top: 50px;  
    margin-bottom: 50px;  
}
```

```
tr.header{  
    background: #C9BFED;  
}
```

```
tr.record{  
    background: #EDEDED;  
}
```

MySQL CRUD

- ❖ 페이징 처리

MySQL

데이터 전체 가져오기 데이터 일부분 가져오기

상품 목록 화면

전체 데이터 개수:30		
상품ID	상품명	가격
30	포도	1000원
29	키위	2000원
28	오렌지	1500원
27	레몬	500원
26	무화과	300원
25	딸기	2000원
24	포도	1000원
23	키위	2000원
22	오렌지	1500원
21	레몬	500원

더보기

MySQL CRUD

❖ 페이징 처리

- ✓ 데이터 목록 처리를 위한 요청을 index.js 파일에 생성

//아이템 목록 보기 : 페이지 번호와 한 페이지 당 가져올 데이터 개수를 설정해서 데이터를 가져옴 페이지 번호와 데이터 개수를 생략하면 기본값 사용

//결과는 count에 전체 데이터 개수 그리고 list에 데이터 목록을 출력

```
app.get('/item/list', (req, res, next) => {
```

//get 방식의 파라미터 가져오기

```
const pageno = req.query.pageno;
```

```
const count = req.query.count;
```

//데이터를 가져올 시작 위치와 데이터 개수 설정

```
var start = 0
```

```
var size = 5
```

```
if(count != undefined){
```

```
    size = parseInt(count)
```

```
}
```

```
if(pageno != undefined){
```

```
    start = (parseInt(pageno) - 1) * size
```

```
}
```

MySQL CRUD

❖ 페이징 처리

- ✓ 데이터 목록 처리를 위한 요청을 index.js 파일에 생성

```
//시작 위치와 페이지 당 데이터 개수를 설정해서 가져오기
var list;
connection.query('SELECT * FROM goods order by itemid desc limit ?, ?', [start, size],
function(err, results, fields) {
    if (err){
        throw err;
    }
    list = results;
    //전체 데이터 개수 가져오기
    connection.query('SELECT count(*) cnt FROM goods', function(err, results, fields)
{
    if (err)
        throw err;
    res.json({'count':results[0].cnt, 'list':list});

});});
});});
```

MySQL CRUD

❖ 페이징 처리

- ✓ index.html 파일에 태그를 추가

```
<a href="#" id="listbtn">데이터 일부분 가져오기</a>
```

MySQL CRUD

❖ 페이징 처리

- ✓ index.html 파일에 스크립트를 추가

```
var pageno = 1;  
var count = 5;
```

```
document.getElementById('listbtn').addEventListener('click', function(e){  
    e.preventDefault();  
    pageno = 1;  
    count = 5;  
  
    var request;  
    request = new XMLHttpRequest();  
    request.open('GET', '/item/list?' + 'pageno=' + pageno + '&count=' + count);  
    request.send("");  
    updatearea.innerHTML = "";
```

MySQL CRUD

❖ 페이징 처리

- ✓ index.html 파일에 스크립트를 추가

```
request.addEventListener('load', function(){
    var data = JSON.parse(request.responseText);
    var display = "<div align='center' class='body'>";
    display += "<h2>상품 목록 화면</h2>";
    display += "<table border='1' id='tbldata'>";
    display += "<tr><td colspan='3' align='right'>전체 데이터 개수:" +
    data.count + "</td></tr>";
    display += "<tr class='header'>"
    display += "<th align='center' width='80'>상품ID</th>";
    display += "<th align='center' width='320'>상품명</th>";
    display += "<th align='center' width='100'>가격</th>";
    display += "</tr>"

    for(var idx in data.list){
        var item = data.list[idx];

        display += "<tr class='record'>";
        display += "<td align='center'>" + item.itemid + "</td>";
        display += "<td align='left'>" + item.itemname + "</td>";
        display += "<td align='right'>" + item.price + "원</td>";
        display += "</tr>";
    }
})
```

MySQL CRUD

❖ 페이지 처리

- ✓ index.html 파일에 스크립트를 추가

```
display += "</table>";
display += "</div>

content.innerHTML = display;

display = "";
if((pageno-1) * count < data.count){
    display += "<table align='center' width='500' id='tblbtn'>";
    display += "<tr id='tblbtn'>";
    display += "<td align='center' colspan='3'><span id='addbtn'>더보기
</span></td>";
    display += "</tr></table>"
}
content.innerHTML += display;
```

MySQL CRUD

❖ 페이지 처리

- ✓ index.html 파일에 스크립트를 추가

```
var addbtn = document.getElementById('addbtn');
if(addbtn != undefined){
    addbtn.addEventListener('click', (e) => {
        pageno = pageno + 1;
        var request;
        request = new XMLHttpRequest();
        request.open('GET', '/item/list?' + 'pageno=' + pageno + '&count=' +
count);
        request.send('');

        if((pageno) * count >= data.count){
            pageno = pageno -1;
            document.getElementById("tblbtn").remove();
        }
    })
}
```

MySQL CRUD

❖ 페이징 처리

- ✓ index.html 파일에 스크립트를 추가

```
request.addEventListener('load', (e) => {
    var data = JSON.parse(request.responseText);
    var display = "";

    const table = document.getElementById('tbldata');

    for(var idx in data.list){
        var item = data.list[idx];

        display += "<tr class='record'>";
        display += "<td align='center'>" + item.itemid + "</td>";
        display += "<td align='left'>" + item.itemname + "</td>";
        display += "<td align='right'>" + item.price + "원</td>";
        display += "</tr>"
    }
    table.innerHTML += display;
});
```

MySQL CRUD

❖ 상세보기

상품 상세 화면



상품명 Mandarin

가격 300원

비고 Vitamin-F

[■ 목록으로 돌아가기](#)

MySQL CRUD

❖ 상세보기

- ✓ index.js 파일에 파라미터를 이용한 상세 보기 처리 함수를 추가

```
//상세보기 - itemid를 매개변수로 받아서 하나의 데이터를 찾아서 출력해주는 처리
app.get('/item/detail', (req, res, next) => {
    const itemid = req.query.itemid;
    connection.query('SELECT * FROM goods where itemid = ?', itemid, function(err,
results, fields) {
        if (err)
            throw err;
        //데이터가 존재하지 않으면 result에 false를 출력
        if(results.length == 0){
            res.json({'result':false});
        }
        //데이터가 존재하면 result에 true를 출력하고 데이터를 item에 출력
        else{
            res.json({'result':true, 'item':results[0]});
        }
    });
});
```

MySQL CRUD

❖ 상세보기

- ✓ index.html 파일의 목록 보기 처리 코드에서 이름을 출력하는 부분을 수정

```
display += "<tr class='record'>";  
display += "<td align='center'>" + item.itemid + "</td>";  
display += "<td align='left'><a href='#' id='item" + item.itemid + "'>" + item.itemname  
+ "</a></td>";  
display += "<td align='right'>" + item.price + "원</td>";  
display += "</tr>"
```

MySQL CRUD

❖ 상세보기

- ✓ index.html 파일에 상세보기를 위한 스크립트 코드를 추가

```
document.getElementById('content').addEventListener('click', (e) => {
    if(e.target.id.startsWith("item")){
        var itemid = e.target.id.substring(4).trim();
        var request;
        request = new XMLHttpRequest();
        request.open('GET', '/item/detail?' + 'itemid=' + itemid);
        request.send('');
    }
});
```

MySQL CRUD

❖ 상세보기

- ✓ index.html 파일에 상세보기를 위한 스크립트 코드를 추가

```
request.addEventListener('load', () => {
    var data = JSON.parse(request.responseText);
    var display = "";
    display += "<div align='center' class='body'>";
    display += "<h2>상품 상세 화면</h2>";

    if(data.result === true){
        var item = data.item;
        display += "<table>";
        display += "<tr><td><img src='/img/" + item.pictureurl + "'></td>";
        display += "<td align='center'><table><tr height='50'><td";
        width='80'>상품명</td>";
        display += "<td width='160'>" + item.itemname + "</td>";
        display += "</tr><tr height='50'><td width='80'>가격</td>";
        display += "<td width='160'>" + item.price + "원</td></tr>";
        display += "<tr height='50'><td width='80'>비고</td>";
        display += "<td width='160'>" + item.description + "</td></tr>";
        display += "<tr><td colspan='2' align='center' width='240'>";
        display += "<a href='#' id='mainbtn'>■ 목록으로 돌아가기
</a></td></tr></table></td></tr>";
        display += "</table>"
    }
}
```

MySQL CRUD

❖ 상세보기

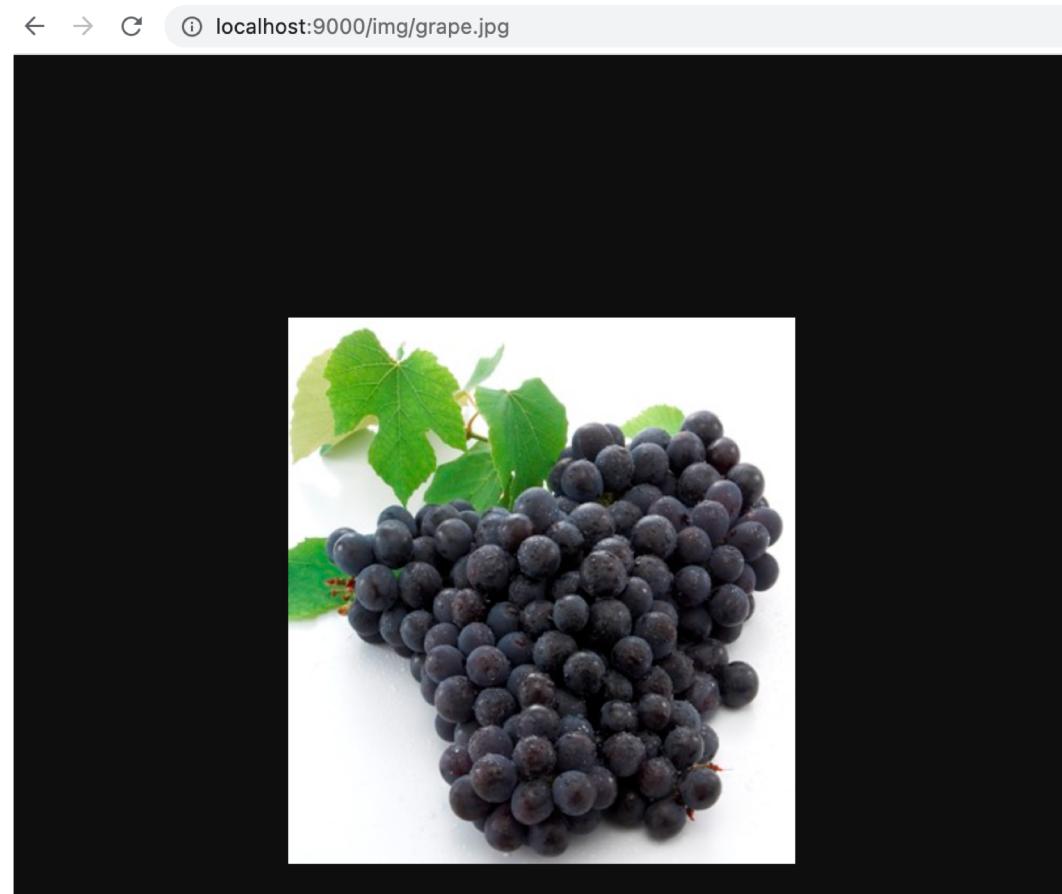
- ✓ index.html 파일에 상세보기를 위한 스크립트 코드를 추가

```
else{
    display += "<p>데이터가 존재하지 않습니다.</p>";
}

content.innerHTML = display;
updatearea.innerHTML = "";
var mainbtn = document.getElementById('mainbtn');
if(mainbtn != undefined){
    mainbtn.addEventListener('click', (e) => {
        document.getElementById('listbtn').click();
    })
}
});
```

MySQL CRUD

- ❖ 이미지 다운로드



MySQL CRUD

❖ 이미지 다운로드

- ✓ index.html 파일의 상세보기 처리 부분에서 이미지 출력 부분을 수정

```
display += "<tr><td><a href='/img/" + item.pictureurl + "'>" + "<img src='/img/" +  
item.pictureurl + "'></a></td>";
```

MySQL CRUD

❖ 이미지 다운로드

- ✓ app.js 파일에 다운로드 처리를 위한 함수를 작성 - 경로는 자신의 프로젝트 경로로 변경

```
app.get('/img/:fileid', function(req, res){  
    var fileid = req.params.fileid;  
    var file = '/Users/mac/Documents/source/node/nodemysql/img' + '/' + fileid;  
    console.log("file:" + file);  
    mimetype = mime.lookup(fileId);  
    console.log("file:" + mimetype);  
    res.setHeader('Content-disposition', 'attachment; filename=' + fileid);  
    res.setHeader('Content-type', mimetype);  
    var filestream = fs.createReadStream(file);  
    filestream.pipe(res);  
});
```

MySQL CRUD

MySQL CRUD

❖ 데이터 삽입

- ✓ public 디렉토리에 insert.html 파일을 생성하고 작성

```
<div>
<p></p>
<form method="post" id="insertform" enctype="multipart/form-data">
아이템 이름<input type="text" name="itemname" id="itemname"/><br/>
아이템 가격<input type="text" name="price" id="price" /><br/>
설명<input type="text" name="description" id="description" /><br/>
이미지<input type="file" name="pictureurl" id="pictureurl"/><br />
<input type="submit" value="삽입"/>
</form>
</div>
```

MySQL CRUD

❖ 데이터 삽입

- ✓ app.js 파일에 데이터 삽입을 위한 처리 코드 추가

//데이터 삽입:itemname, description, price, pictureurl(파일)을 받아서 처리
//itemid는 가장 큰 itemid를 찾아서 + 1

```
app.post('/item/insert', upload.single('pictureurl'), (req, res, next) => {  
    //파라미터 가져오기  
    const itemname = req.body.itemname;  
    const description = req.body.description;  
    const price = req.body.price;  
    var pictureurl;  
    if(req.file){  
        pictureurl = req.file.filename  
    }else{  
        pictureurl = "default.jpg";  
    }
```

MySQL CRUD

❖ 데이터 삽입

- ✓ app.js 파일에 데이터 삽입을 위한 처리 코드 추가

```
//가장 큰 itemid 가져오기
connection.query('select max(itemid) maxid from goods', function(err, results, fields) {
    if (err)
        throw err;
    var itemid;
    if(results.length > 0){
        itemid = results[0].maxid + 1
    }else{
        itemid = 1;
    }
})
```

MySQL CRUD

❖ 데이터 삽입

- ✓ app.js 파일에 데이터 삽입을 위한 처리 코드 추가

```
//현재 시간의 년월일 시분초 가져오기
var date = new Date()
var year = date.getFullYear();
var month = (1 + date.getMonth());
month = month >= 10 ? month : '0' + month;
var day = date.getDate();
day = day >= 10 ? day : '0' + day;

var hour = date.getHours();
hour = hour >= 10 ? hour : '0' + hour;
var minute = date.getMinutes();
minute = minute >= 10 ? minute : '0' + minute;
var second = date.getSeconds();
second = second >= 10 ? second : '0' + second;
```

MySQL CRUD

❖ 데이터 삽입

- ✓ app.js 파일에 데이터 삽입을 위한 처리 코드 추가

```
//데이터 삽입
connection.query('insert into goods(itemid, itemname, price, description, pictureurl,
updatedate) values(?,?,?,?,?,?)',
[itemid, itemname, price, description, pictureurl, year + '-' + month
+ '-' + day], function(err, results, fields) {
if (err)
    throw err;
console.log(results)
if(results.affectedRows == 1){
    const writeStream = fs.createWriteStream('./update.txt');
    writeStream.write(year + '-' + month + '-' + day + " " + hour + ":" +
minute + ":" + second);
    writeStream.end();

    res.json({'result':true});
} else{
    res.json({'result':false});
}
});
});
```

MySQL CRUD

❖ 데이터 삽입

- ✓ index.html 파일에 데이터 삽입 요청을 추가
`데이터 삽입`

MySQL CRUD

❖ 데이터 삽입

- ✓ index.html 파일에 데이터 삽입 요청을 처리하는 스크립트 추가

```
document.getElementById('insertbtn').addEventListener('click', function(e){  
    e.preventDefault();  
    e.stopPropagation()  
    request = new XMLHttpRequest();  
    request.open('GET', '/item/insert');  
    request.send();  
  
    request.addEventListener('load', () => {  
        content.innerHTML = "";  
        updatearea.innerHTML = request.responseText;  
        var f = document.getElementById("insertform");  
        if(f != undefined){  
            f.addEventListener('submit', (e) => {  
                e.preventDefault();  
                const formData = new  
Formdata(document.getElementById("insertform"));
```

MySQL CRUD

❖ 데이터 삽입

- ✓ index.html 파일에 데이터 삽입 요청을 처리하는 스크립트 추가

```
var xhr = new XMLHttpRequest();
xhr.open("POST" , "/item/insert" , true);
xhr.send(formData);

xhr.addEventListener('load', (e) => {
    var data = JSON.parse(xhr.responseText);
    if(data.result == true){
        document.getElementById("listbtn").click();
        updatearea.innerHTML = "";
    }else{
        alert("삽입에 실패")
    }
});
});
```

MySQL CRUD

❖ 데이터 삭제

MySQL

데이터 전체 가져오기 데이터 일부분 가져오기 데이터 삽입

localhost:9000 내용:
삭제 성공

확인

상품 상세 화면



상품명	포도
가격	1000원
비고	항상화 성분과 당분이 높고 무기물이 많은 과일

■ 목록으로 돌아가기
■ 데이터 삭제

MySQL CRUD

❖ 데이터 삭제

- ✓ app.js 파일에 데이터 삭제를 위한 코드 추가

```
app.post('/item/delete', (req, res, next) => {
  const itemid = req.body.itemid;
  console.log("itemid:", itemid);
  var date = new Date()
  var year = date.getFullYear();
  var month = date.getMonth() + 1;
  month = month >= 10 ? month : '0' + month;
  var day = date.getDate();
  day = day >= 10 ? day : '0' + day;

  var hour = date.getHours();
  hour = hour >= 10 ? hour : '0' + hour;
  var minute = date.getMinutes();
  minute = minute >= 10 ? minute : '0' + minute;
  var second = date.getSeconds();
  second = second >= 10 ? second : '0' + second;
```

MySQL CRUD

❖ 데이터 삭제

- ✓ app.js 파일에 데이터 삭제를 위한 코드 추가

```
connection.query('delete FROM goods where itemid = ?', itemid, function(err, results, fields) {
    if (err)
        throw err;
    if(results.affectedRows == 1){
        const writeStream = fs.createWriteStream('./update.txt');
        writeStream.write(year + '-' + month + '-' + day + " " + hour + ":" +
minute + ":" + second);
        writeStream.end();
        res.json({'result':true});
    }else{
        res.json({'result':false});
    }
});
```

MySQL CRUD

❖ 데이터 삭제

- ✓ index.html 파일의 데이터 상세보기 스크립트 코드 수정

```
document.getElementById('content').addEventListener('click', (e) => {
    if(e.target.id.startsWith("item")){
        var itemid = e.target.id.substring(4).trim();
        var request;
        request = new XMLHttpRequest();
        request.open('GET', '/item/detail?' + 'itemid=' + itemid);
        request.send('');

        request.addEventListener('load', () => {
            var data = JSON.parse(request.responseText);
            var display = "";
            display += "<div align='center' class='body'>";
            display += "<h2>상품 상세 화면</h2>";
```

MySQL CRUD

❖ 데이터 삭제

- ✓ index.html 파일의 데이터 상세보기 스크립트 코드 수정

```
if(data.result === true){  
    var item = data.item;  
    display += "<table>";  
    display += "<tr><td><a href='/img/" + item.pictureurl + "'>" + "<img  
src='/img/" + item.pictureurl + "'></a></td>";  
    display += "<td align='center'><table><tr height='50'><td  
width='80'>상품명</td>";  
    display += "<td width='160'>" + item.itemname + "</td>";  
    display += "</tr><tr height='50'><td width='80'>가격</td>";  
    display += "<td width='160'>" + item.price + "원</td></tr>";  
    display += "<tr height='50'><td width='80'>비고</td>";  
    display += "<td width='160'>" + item.description + "</td></tr>";  
    display += "<tr><td colspan='2' align='center' width='240'>";  
    display += "<a href='#' id='mainbtn'>■ 목록으로돌아가기  
</a></td></tr>";  
    display += "<tr><td colspan='2' align='center' width='240'>";  
    display += "<a href='#' id='deletebtn'>■ 데이터 삭제</a></td></tr>";  
    display += "</table></td></tr>";  
    display += "</table>"  
}else{  
    display += "<p>데이터가 존재하지 않습니다.</p>";  
}  
}
```

MySQL CRUD

❖ 데이터 삭제

- ✓ index.html 파일의 데이터 상세보기 스크립트 코드 수정

```
content.innerHTML = display;  
updatearea.innerHTML = "";  
  
var mainbtn = document.getElementById('mainbtn');  
if(mainbtn != undefined){  
    mainbtn.addEventListener('click', (e) => {  
        document.getElementById('listbtn').click();  
    })  
}
```

MySQL CRUD

❖ 데이터 삭제

- ✓ index.html 파일의 데이터 상세보기 스크립트 코드 수정

```
var deletebtn = document.getElementById('deletebtn');
if(deletebtn != undefined){
    deletebtn.addEventListener('click', (e) => {
        e.preventDefault();
        request = new XMLHttpRequest();
        request.open('POST', '/item/delete', true);
        var params = 'itemid=' + data.item.itemid;
        request.setRequestHeader('Content-type', 'application/x-www-form-
urlencoded');
        request.send(params);
        request.addEventListener('load', () => {
            var data = JSON.parse(request.responseText);
            if(data.result == true){
                document.getElementById("listbtn").click();
                updatearea.innerHTML = "";
                alert("삭제 성공");
            }else{
                alert("삭제 실패");
            }
        });
    });
}
```

MySQL CRUD

❖ 데이터 삭제

- ✓ index.html 파일의 데이터 상세보기 스크립트 코드 수정

```
        }  
    }  
});  
}  
})
```

MySQL CRUDE

- ❖ 데이터 수정

MySQL

데이터 전체 가져오기 데이터 일부분 가져오기 데이터 삽입

아이템 아이디: 26

아이템 이름: 무화과

아이템 가격: 300

설명: 칼슘, 섬유질 및 항산화 물질

새로운 이미지: 파일 선택 선택된 파일 없음



이전 이미지

수정

MySQL CRUD

❖ 데이터 수정

- ✓ 데이터 수정을 위한 update.html 파일을 public 디렉토리에 생성하고 작성

```
<form method="post" id="updateform" enctype="multipart/form-data">
아이템 아이디:<input type="text" name="itemid" id="itemid"/><br/>
아이템 이름<input type="text" name="itemname" id="itemname"/><br/>
아이템 가격<input type="text" name="price" id="price"/><br/>
설명<input type="text" name="description" id="description" /><br/>
새로운 이미지<input type="file" name="pictureurl" /><br />
이전 이미지<img width="100" height="100" id="picture"/><br/>
<input type="hidden" name="oldpictureurl" id="oldpictureurl"/><br/>
<input type="submit" value="수정">
</form>
```

MySQL CRUD

❖ 데이터 수정

- ✓ app.js 파일에 데이터 수정 처리를 위한 코드를 추가

```
app.get('/item/update', (req, res, next) => {
    fs.readFile('public/update.html', function (err, data) {
        res.end(data);
    });
});
```

MySQL CRUD

❖ 데이터 수정

- ✓ app.js 파일에 데이터 수정 처리를 위한 코드를 추가

//데이터 수정: itemid, itemname, description, price, oldpictureurl, pictureurl(파일)을 받아서 처리

```
app.post('/item/update', upload.single('pictureurl'), (req, res, next) => {
    //파라미터 가져오기
    const itemid = req.body.itemid;
    const itemname = req.body.itemname;
    const description = req.body.description;
    const price = req.body.price;
    const oldpictureurl = req.body.oldpictureurl;

    var pictureurl;
    if(req.file){
        pictureurl = req.file.filename
    }else{
        pictureurl = oldpictureurl;
    }
```

MySQL CRUD

❖ 데이터 수정

- ✓ app.js 파일에 데이터 수정 처리를 위한 코드를 추가

```
//현재 시간의 년월일 시분초 가져오기
var date = new Date()
var year = date.getFullYear();
var month = (1 + date.getMonth());
month = month >= 10 ? month : '0' + month;
var day = date.getDate();
day = day >= 10 ? day : '0' + day;

var hour = date.getHours();
hour = hour >= 10 ? hour : '0' + hour;
var minute = date.getMinutes();
minute = minute >= 10 ? minute : '0' + minute;
var second = date.getSeconds();
second = second >= 10 ? second : '0' + second;
```

MySQL CRUD

❖ 데이터 수정

- ✓ app.js 파일에 데이터 수정 처리를 위한 코드를 추가

```
//데이터 수정
connection.query('update goods set itemname=?, price=?, description=?,
pictureurl=?, updatedate=? where itemid=?',
[itemname, price, description, pictureurl, year + '-' + month + '-' + day,
itemid], function(err, results, fields) {
if (err)
    throw err;
console.log(results)
if(results.affectedRows == 1){
    const writeStream = fs.createWriteStream('./update.txt');
    writeStream.write(year + '-' + month + '-' + day + " " + hour + ":" +
minute + ":" + second);
    writeStream.end();

    res.json({'result':true});
}else{
    res.json({'result':false});
}
});
});
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
document.getElementById('content').addEventListener('click', (e) => {
    if(e.target.id.startsWith("item")){
        var itemid = e.target.id.substring(4).trim();
        var request;
        request = new XMLHttpRequest();
        request.open('GET', '/item/detail?' + 'itemid=' + itemid);
        request.send('');

        request.addEventListener('load', () => {
            var data = JSON.parse(request.responseText);
            var display = "";
            display += "<div align='center' class='body'>";
            display += "<h2>상품 상세 화면</h2>";
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
if(data.result === true){  
    var item = data.item;  
    display += "<table>";  
    display += "<tr><td><a href='/img/" + item.pictureurl + "'>" +  
    "<img src='/img/" + item.pictureurl + "'></a></td>";  
    display += "<td align='center'><table><tr height='50'><td  
width='80'>상품명</td>";  
    display += "<td width='160'>" + item.itemname + "</td>";  
    display += "</tr><tr height='50'><td width='80'>가격</td>";  
    display += "<td width='160'>" + item.price + "원</td></tr>";  
    display += "<tr height='50'><td width='80'>비고</td>";  
    display += "<td width='160'>" + item.description + "</td></tr>";  
    display += "<tr><td colspan='2' align='center' width='240'>";  
    display += "<a href='#' id='mainbtn'>■ 목록으로돌아가기  
</a></td></tr>";  
    display += "<tr><td colspan='2' align='center' width='240'>";  
    display += "<a href='#' id='deletebtn'>■ 데이터 삭제</a>";  
    display += "<a href='#' id='updatebtn'>■ 데이터 수정  
</a></td></tr>";  
    display += "</table></td></tr>";  
    display += "</table>"  
}
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
        }else{
            display += "<p>데이터가 존재하지 않습니다.</p>";
        }

        content.innerHTML = display;
        updatearea.innerHTML = "";

var mainbtn = document.getElementById('mainbtn');
if(mainbtn != undefined){
    mainbtn.addEventListener('click', (e) => {
        document.getElementById('listbtn').click();
    })
}
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
var updatebtn = document.getElementById('updatebtn');
if(updatebtn != undefined){
    updatebtn.addEventListener('click', (e) => {
        e.preventDefault();
        e.stopPropagation()
        request = new XMLHttpRequest();
        request.open('GET', '/item/update');
        request.send("");

        request.addEventListener('load', () => {
            content.innerHTML = "";
            updatearea.innerHTML = request.responseText;
            var item = data.item;

            document.getElementById("itemid").value = item.itemid;
            document.getElementById("itemid").readOnly = true;

            document.getElementById("itemname").value = item.itemname;

            document.getElementById("price").value = item.price;
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
document.getElementById("description").value = item.description;
```

```
document.getElementById("oldpictureurl").value = item.pictureurl;  
document.getElementById("picture").src = "/img/" +
```

```
item.pictureurl;
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
var f = document.getElementById("updateform");
if(f != undefined){
    f.addEventListener('submit', (e) => {
        e.preventDefault();
        const formData = new
FormData(document.getElementById("updateform"));

        var xhr = new XMLHttpRequest();
        xhr.open("POST" , "/item/update" , true);
        xhr.send(formData);
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
xhr.addEventListener('load', (e) => {
    var data = JSON.parse(xhr.responseText);
    if(data.result == true){
        document.getElementById("listbtn").click();
        updatearea.innerHTML = "";
    }else{
        alert("수정에 실패")
    }
});
}
});
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
var deletebtn = document.getElementById('deletebtn');
if(deletebtn != undefined){
    deletebtn.addEventListener('click', (e) => {
        e.preventDefault();
        request = new XMLHttpRequest();
        request.open('POST', '/item/delete', true);
        var params = 'itemid=' + data.item.itemid;
        request.setRequestHeader('Content-type', 'application/x-www-form-
urlencoded');
        request.send(params);
```

MySQL CRUD

❖ 데이터 수정

- ✓ index.html 파일의 상세보기 처리 부분의 스크립트 코드 수정

```
request.addEventListener('load', () => {
    var data = JSON.parse(request.responseText);
    if(data.result == true){
        document.getElementById("listbtn").click();
        updatearea.innerHTML = "";
        alert("삭제 성공");
    }else{
        alert("삭제 실패");
    }
});
}
});
```

MySQL CRUD

❖ 업데이트 시간 리턴



```
localhost:5000/item/date
{
  "result": "1599554695625"
}
```

A screenshot of a web browser showing a simple JSON response. The URL in the address bar is 'localhost:5000/item/date'. The response body contains a single key-value pair: 'result' with the value '1599554695625'. The browser interface includes standard navigation buttons (back, forward, refresh), a search/address bar, and a toolbar with various icons.

MySQL CRUD

- ❖ 업데이트 시간 리턴

- ✓ app.js 파일에 추가

```
app.get('/item/date', (req, res, next) => {
  fs.readFile('./update.txt', function (err, data) {
    res.json({'result':data.toString()});
  });
});
```