Node.js

DB 연동 - 김근형 강사

DB 연동

- DB 연동
  - 서버 프로그램 운영 시 발생되는 데이터는 반드시 저장되어야 다음에 사용이 가능하다.
  - 서버 프로그램 개발 시 데이터를 저장할 때 데이터베이스를 많이 이용하는데 node.js 프로그램으로 데이터베이스를 사용할 수 있다.
  - DB 사용에 대한 기능은 기본 모듈로는 제공되지 않으며 외부 모듈을 사용해야 하는데 npmjs.com 에서 검색하면 모듈을 찾을 수 있다.

MySQL 연동

- MySQL 테이블 세팅

```
create database mydb default character set utf8 collate utf8_general_ci;

use mydb;

create table custommers(
    id int auto_increment primary key,
    name varchar(255),
    address varchar(255)
);
```

MySQL 연동

- node.js package.json 세팅 후 mysql 연결 확인

```
{
    "name": "database",
    "version": "0.0.0",
    "private": true,
    "dependencies": {
        "mysql2": "*"
    }
}
```

```
const mysql = require('mysql2');

let conn_info = {
    host : 'localhost',  // 호스트 정보
    port : 3306,  // 포트 번호
    user : 'root',  // 접속id
    password : '1234',  // 접속 패스워드
    database : 'mydb'  // 디비이름
};

let conn = mysql.createConnection(conn_info);

conn.connect(function(err){
    if(err){
        console.log('접속오류');
        console.log(err);
    }else{
        console.log('접속성공');
        conn.end();
    }
});
```

접속성공

MySQL insert

- insert 예제

```
접속성공
저장완료1
아이디 : 16
저장완료2
개수 : 14
```

id	name	address
16	Company Inc	Highway 37
17	John	Highway 71
18	Peter	Lowstreet 4
19	Amy	Apple st 652
20	Hannah	Mountain 21
21	Michael	Valley 345
22	Sandy	Ocean blvd 2
23	Betty	Green Grass 1
24	Richard	Sky st 331
25	Susan	One way 98
26	Vicky	Yellow Garden 2
27	Ben	Park Lane 38
28	William	Central st 954
29	Chuck	Main Road 989
30	Viola	Sideway 1633

```js
conn.connect(function(err){
    if(err){
        console.log('접속오류');
        console.log(err);
    }else{
        console.log('접속성공');

        let sql1 = "insert into custommers (name, address) values (?, ?)";
        let input_data1 = ['Company Inc', 'Highway 37'];
        conn.query(sql1, input_data1, (err, result)=>{
            console.log("저장완료1");
            console.log("아이디 : "+result.insertId);
        });

        let sql2 = "insert into custommers (name, address) values ?";
        let input_data2 = [
            ['John', 'Highway 71'],
            ['Peter', 'Lowstreet 4'],
            ['Amy', 'Apple st 652'],
            ['Hannah', 'Mountain 21'],
            ['Michael', 'Valley 345'],
            ['Sandy', 'Ocean blvd 2'],
            ['Betty', 'Green Grass 1'],
            ['Richard', 'Sky st 331'],
            ['Susan', 'One way 98'],
            ['Vicky', 'Yellow Garden 2'],
            ['Ben', 'Park Lane 38'],
            ['William', 'Central st 954'],
            ['Chuck', 'Main Road 989'],
            ['Viola', 'Sideway 1633']
        ];
        conn.query(sql2, [input_data2], (err, result)=>{
            console.log("저장완료2");
            console.log("개수 : "+result.affectedRows);
        });

        conn.end();
    }
});
```

MySQL insert

- select 예제

```
}else{
    console.log('접속성공');

    var sql1 = "SELECT * FROM custommers";
    conn.query(sql1, (err, result, fields)=>{
        console.log('컬럼 정보');
        for(let obj of fields){
            console.log(obj.name,'  ',obj.table);
        }
        console.log('데이터 정보1');
        for(let obj of result){
            console.log("id : "+obj.id+"   name : "+obj.name
            +"   address : "+obj.address);
        }
    });

    var sql2 = "SELECT name, address FROM custommers WHERE id = ?";
    let data1 = 22;
    conn.query(sql2, data1, (err, result, fields)=>{
        console.log('데이터 정보2');
        for(let obj of result){
            console.log("id : "+obj.id+"   name : "+obj.name
            +"   address : "+obj.address);
        }
    });

    conn.end();
}
```

```
접속성공
컬럼 정보
id    custommers
name    custommers
address    custommers
데이터 정보1
id : 16   name : Company Inc   address : Highway 37
id : 17   name : John   address : Highway 71
id : 18   name : Peter   address : Lowstreet 4
id : 19   name : Amy   address : Apple st 652
id : 20   name : Hannah   address : Mountain 21
id : 21   name : Michael   address : Valley 345
id : 22   name : Sandy   address : Ocean blvd 2
id : 23   name : Betty   address : Green Grass 1
id : 24   name : Richard   address : Sky st 331
id : 25   name : Susan   address : One way 98
id : 26   name : Vicky   address : Yellow Garden 2
id : 27   name : Ben   address : Park Lane 38
id : 28   name : William   address : Central st 954
id : 29   name : Chuck   address : Main Road 989
id : 30   name : Viola   address : Sideway 1633
데이터 정보2
id : undefined   name : Sandy   address : Ocean blvd 2
```

MySQL insert

- delete, update 예제

```
let sql = "DELETE FROM custommers WHERE id = ?";
let data = 17
conn.query(sql, data, function (err, result) {
    console.log("지운 row 갯수: " + result.affectedRows);
});

conn.end();
```

```
접속성공
지운 row 갯수: 1
```

```
let sql = "UPDATE custommers SET address = ? WHERE id = ?";
let data = ['Seoul 21', 18];
conn.query(sql, data, function (err, result) {
    console.log(result.affectedRows + " record(s) updated");
});
```

```
접속성공
1 record(s) updated
```

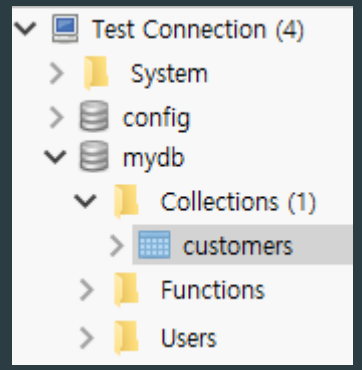
16	Company Inc	Highway 37
18	Peter	Seoul 21
19	Amy	Apple st 652
20	Hannah	Mountain 21
21	Michael	Valley 345
22	Sandy	Ocean blvd 2
23	Betty	Green Grass 1
24	Richard	Sky st 331
25	Susan	One way 98
26	Vicky	Yellow Garden 2
27	Ben	Park Lane 38
28	William	Central st 954
29	Chuck	Main Road 989
30	Viola	Sideway 1633

MongoDB 연동

- package.json 예제 및 collection 생성

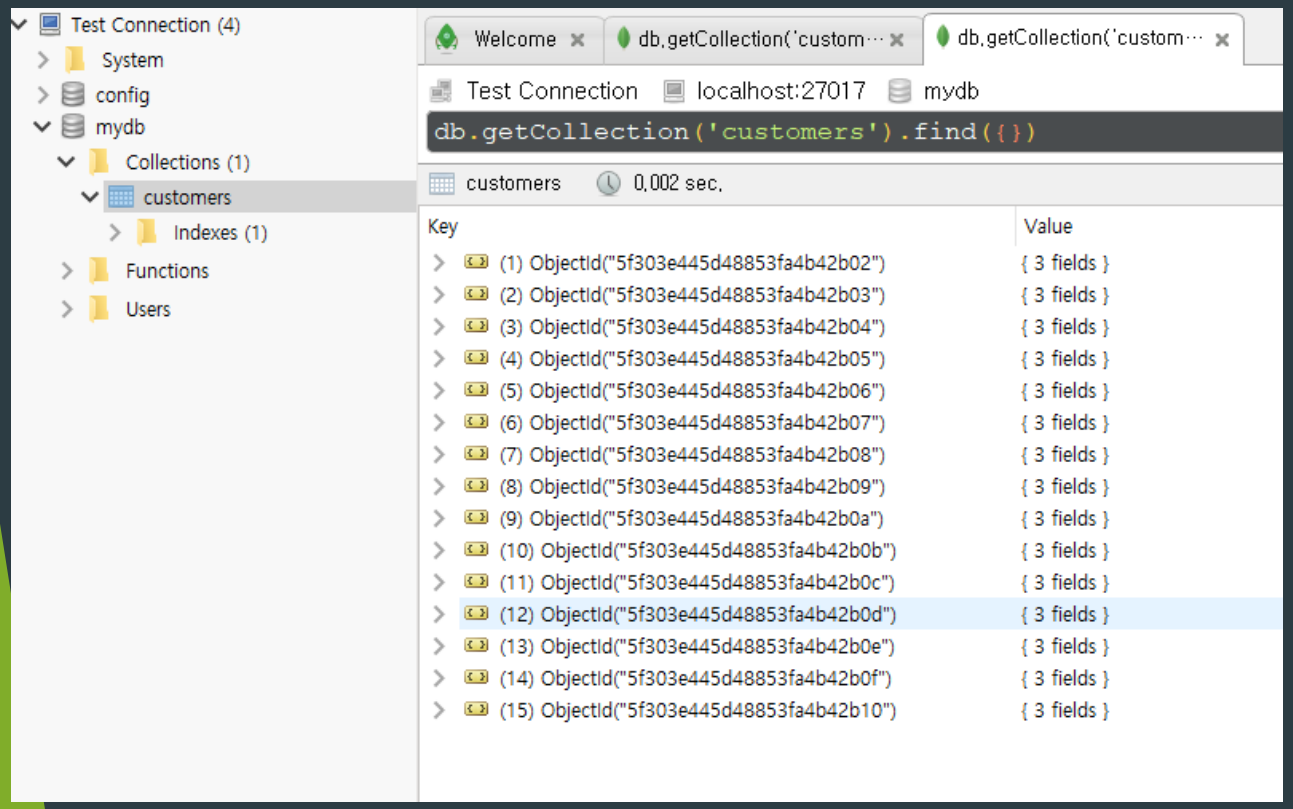
```
{
    "name": "database",
    "version": "0.0.0",
    "private": true,
    "dependencies": {
        "mysql2": "*",
        "mongodb": "*"
    }
}
```

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/mydb";

MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    dbo.createCollection("customers", function(err, res) {
        if (err) throw err;
        console.log("Collection created!");
        db.close();
    });
});
```



MongoDB 연동

- insert 예제

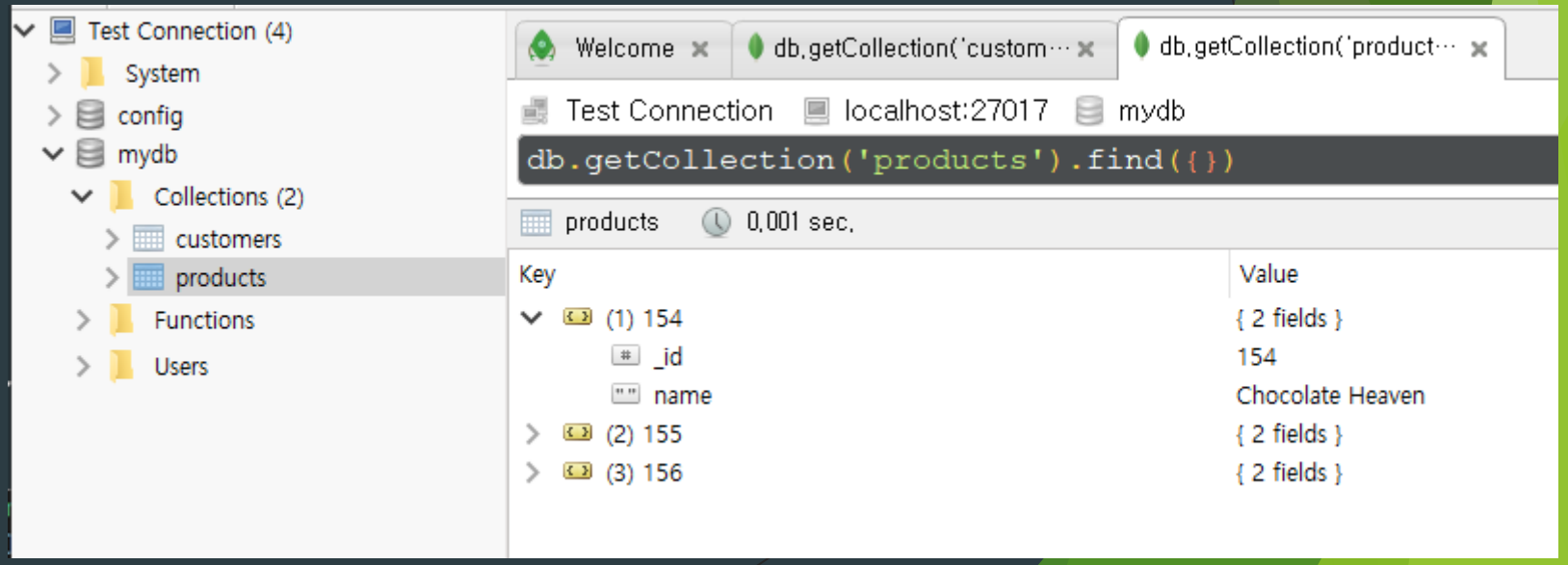



```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    var myobj = { name: "Company Inc", address: "Highway 37" };
    // 한개만 넣을 경우 사용
    dbo.collection("customers").insertOne(myobj, function(err, res) {
        if (err) throw err;
        console.log("1 document inserted");
    });

    var myobj = [
        { name: 'John', address: 'Highway 71'},
        { name: 'Peter', address: 'Lowstreet 4'},
        { name: 'Amy', address: 'Apple st 652'},
        { name: 'Hannah', address: 'Mountain 21'},
        { name: 'Michael', address: 'Valley 345'},
        { name: 'Sandy', address: 'Ocean blvd 2'},
        { name: 'Betty', address: 'Green Grass 1'},
        { name: 'Richard', address: 'Sky st 331'},
        { name: 'Susan', address: 'One way 98'},
        { name: 'Vicky', address: 'Yellow Garden 2'},
        { name: 'Ben', address: 'Park Lane 38'},
        { name: 'William', address: 'Central st 954'},
        { name: 'Chuck', address: 'Main Road 989'},
        { name: 'Viola', address: 'Sideway 1633'}
    ];
    // 여러개 열을 입력할 경우 사용
    dbo.collection("customers").insertMany(myobj, function(err, res) {
        if (err) throw err;
        console.log("Number of documents inserted: " + res.insertedCount);
    });
    db.close();
});
```

MongoDB 연동

- insert 예제 - id 값 포함

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    var myobj = [
        { _id: 154, name: 'Chocolate Heaven'},
        { _id: 155, name: 'Tasty Lemon'},
        { _id: 156, name: 'Vanilla Dream'}
    ];
    dbo.collection("products").insertMany(myobj, function(err, res) {
        if (err) throw err;
        console.log(res);
        db.close();
    });
});
```



MongoDB 연동

- find 예제 - 1

```
MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    // 한개만 출력
    dbo.collection("customers").findOne({}, function(err, result) {
        console.log(result.name);
    });
    console.log('');
    // 여러개 출력
    dbo.collection("customers").find({}).toArray(function(err, result) {
        for(var obj of result){
            console.log(obj._id,",",obj.name,",",obj.address);
        }
    });
    console.log('-----------------');
    // 특정 부분만 출력 1
    dbo.collection("customers").find({}, { projection: { _id: 0, name: 1, address: 1 } }).toArray(function(err, result) {
        for(var obj of result){
            console.log(obj.name,",",obj.address);
        }
    });
    console.log('-----------------');
    // 쿼리 검색 결과 출력
    let query = { address: "Park Lane 38" };
    dbo.collection("customers").find(query).toArray(function(err, result) {
        for(var obj of result){
            console.log("query result : ",obj._id,",",obj.name,",",obj.address);
        }
    });
    // 정규표현식 쿼리
    var query = { address: /^S/ };
    dbo.collection("customers").find(query).toArray(function(err, result) {
        for(var obj of result){
            console.log("query result : ",obj._id,",",obj.name,",",obj.address);
        }
    });
    db.close();
});
```

MongoDB 연동

- find 예제 - 2

```
MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    // name 오름차순
    var mysort = { name: 1 };
    dbo.collection("customers").find().sort(mysort).toArray(function(err, result) {
        for(var obj of result){
            console.log("query result : ",obj._id,",",obj.name,",",obj.address);
        }
    });
    // name 내림차순
    var mysort = { name: -1 };
    dbo.collection("customers").find().sort(mysort).toArray(function(err, result) {
        for(var obj of result){
            console.log("query result : ",obj._id,",",obj.name,",",obj.address);
        }
    });
    // limit 구현
    dbo.collection("customers").find().limit(5).toArray(function(err, result) {
        for(var obj of result){
            console.log("query result : ",obj._id,",",obj.name,",",obj.address);
        }
    });
    db.close();
});
```

MongoDB 연동

- delete 예제 - 1

```
MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    // 검색된 대상 중 하나만 지우기
    var myquery = { address: 'Mountain 21' };
    dbo.collection("customers").deleteOne(myquery, function(err, obj) {
        if (err) throw err;
        console.log("1 document deleted");
    });
    // 검색된 대상 모두 지우기 (정규표현식 사용)
    var myquery = { address: /^O/ };
    dbo.collection("customers").deleteMany(myquery, function(err, obj) {
        if (err) throw err;
        console.log(obj.result.n + " document(s) deleted");
    });
    db.close();
});
```

MongoDB 연동

- update 예제 - 1

```
MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    // 주어진 조건 쿼리
    var myquery = { address: "Valley 345" };
    // 업데이트 할 내용
    var newvalues = { $set: {name: "Mickey", address: "Canyon 123" } };
    // 맨 위의 최상단에 위치한 하나의 도큐먼트만 업데이트
    dbo.collection("customers").updateOne(myquery, newvalues, function(err, res) {
        if (err) throw err;
        console.log("1 document updated");
    });
    // 주어진 조건 쿼리 (정규표현식)
    var myquery = { address: /^S/ };
    // 업데이트 할 내용
    var newvalues = {$set: {name: "Minnie"} };
    // 조건 쿼리에 해당되는 모든 내용을 업데이트할 내용으로 업데이트
    dbo.collection("customers").updateMany(myquery, newvalues, function(err, res) {
        if (err) throw err;
        console.log(res.result.nModified + " document(s) updated");
    });
    db.close();
});
```