

K-Digital Training 웹 풀스택 과정

# MVC MySQL

# 프로젝트에 MySQL 연결하기

# Node.js – MySQL 연결

- mysql2 패키지를 설치

```
$ npm install mysql2
```

- package.json 에서 설치 확인

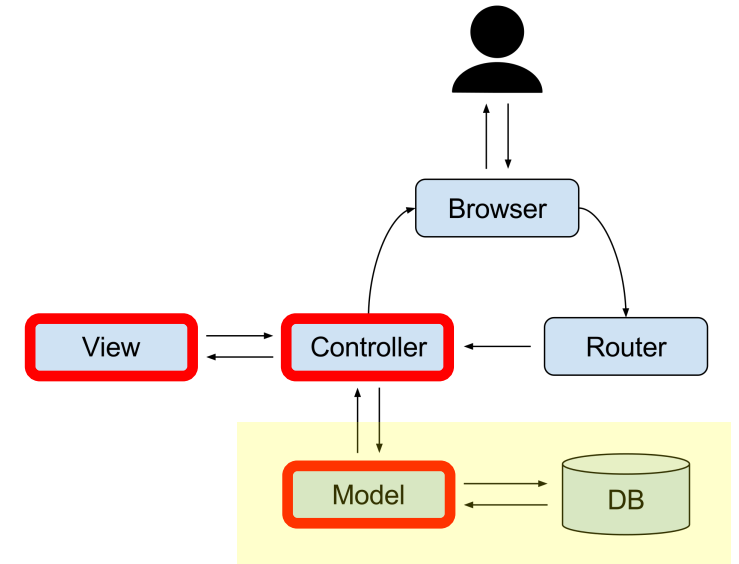
```
"dependencies": {  
  "ejs": "^3.1.8",  
  "express": "^4.18.2",  
  "mysql2": "^3.6.5"  
}
```

# model/Visitor.js

- mysql 패키지를 설치 후 이용

```
const mysql = require('mysql2/promise');  
//mysql연결  
const getConn = async () => {  
  return await mysql.createConnection({  
    host: 'localhost',  
    user: 'kdt',  
    password: '1234',  
    database: 'kdt',  
    port: 3306,  
  });  
};
```

model/Visitor.js



# Todo List

1. 작성 후 “등록” 을 누르면 **DB** 에 저장된다. (**C**reate)
2. “등록” 된 내용이 아래 Table 에 바로 보인다. (**R**ead)
3. “수정” 을 누르면 **방명록** 수정이 가능하다. (**U**ppdate)
4. “삭제” 를 누르면 **방명록**이 삭제된다. (**D**elete)

# 데이터베이스 테이블 생성

# visitor 테이블 생성하기

```
mysql> DESC visitor;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(10)	NO		NULL	
comment	mediumtext	YES		NULL	

3 rows in set (0.00 sec)

insert문 이용하여 visitor 테이블에 저장해보자!!

```
mysql> SELECT * FROM visitor;
```

id	name	comment
1	홍길동	내가 왔다.
2	이찬혁	으라차차

2 rows in set (0.00 sec)

# model/Visitor.js

```
const allVisitor = async () => {  
  const conn = await getConn();  
  const query = `SELECT * FROM visitor`;   
  const [rows] = await conn.query(query);  
  await conn.end();  
  return rows;  
};
```

model/Visitor.js – **allVisitor()**: 전체 방문록 목록을 가져옴



# 주의) Node.js mysql 연결

```
Error: Access denied for user 'user'@'localhost' (using password: YES)
```

- 외부에서 최상위 **root** 계정으로의 비밀번호 접근을 허용하지 않는다!
- 즉, 새로운 사용자(**user** 계정)를 만들고 그 사용자로 접근을 해야 한다.

# controller/Cvisitor.js

```
const getVisitor = async (id) => {  
  const conn = await getConn();  
  //const query = `SELECT * FROM visitor WHERE id=${id}`;  
  const query = 'SELECT * FROM visitor WHERE id=?'; //문자열 보간법  
  const [rows] = await conn.query(query, [id]);  
  await conn.end();  
  return rows;  
};
```

controller/Cvisitor.js

model/Visitor.js

```
//하나 방명록 조회  
const getVisitor = async (req, res) => {  
  console.log(req.params.id);  
  const data = await Visitor.getVisitor(req.params.id);  
  console.log(data);  
  res.json({ result: data });  
};
```

# 라우터 정리

1. 작성 후 “등록” 을 누르면 **DB** 에 저장된다. (**C**reate)
2. “등록” 된 내용이 아래 Table 에 바로 보인다. (**R**ead)
3. “수정” 을 누르면 방명록 수정이 가능하다. (**U**ppdate)
4. “삭제” 를 누르면 방명록이 삭제된다. (**D**elete)

GET `/` : 메인 페이지 보이기 (index.ejs)

GET `/visitor` : 방명록 전체 보이기 (visitor.ejs)

GET `/visitor/get` : 방명록 하나 조회

POST `/visitor/write` : 방명록 하나 추가

PATCH `/visitor/edit` : 방명록 하나 수정

DELETE `/visitor/delete` : 방명록 하나 삭제

# 실습. Node.js – MySQL

- 회원가입, 로그인, 회원정보 수정, 회원 탈퇴를 DB와 연동하여 구현
- 이 때, CRUD 모두 사용
  - CREATE (INSERT) : 회원 가입
  - READ (SELECT) : 로그인 (회원 검색)
  - UPDATE (UPDATE) : 회원 정보 수정
  - DELETE (DELETE) : 회원 탈퇴 (삭제)

# 실습. Node.js – MySQL

- user 테이블 구조

```
mysql> DESC user;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra           |
+-----+-----+-----+-----+-----+-----+
| id    | int           | NO   | PRI | NULL    | auto_increment |
| userid | varchar(20)   | NO   |     | NULL    |                 |
| name  | varchar(10)   | NO   |     | NULL    |                 |
| pw    | varchar(20)   | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```