

## LAB 4

หลังจาก clone react-nodejs-example

At api

```
{ } package.json JS server.js X
backend > react-nodejs-example > api > JS server.js > ...
1  const express = require('express');
2  const path = require('path');
3  const app = express(),
4    bodyParser = require("body-parser");
5    port = 4000;
6
```

เข้าไปแก้ไข server.js โดย เปลี่ยน port เป็น 4000

```
{ } package.json X JS server.js
backend > react-nodejs-example > api > { } package.json > { } scripts :
1  {
2    "name": "react-nodejs-example",
3    "version": "1.0.0",
4    "description": "example project react with no
5    "main": "server.js",
6    "scripts": {
7      "start": "node server.js",
```

เข้าไปแก้ไข package.js โดย เปลี่ยน start เป็น node server.js

ลอง npm start

```

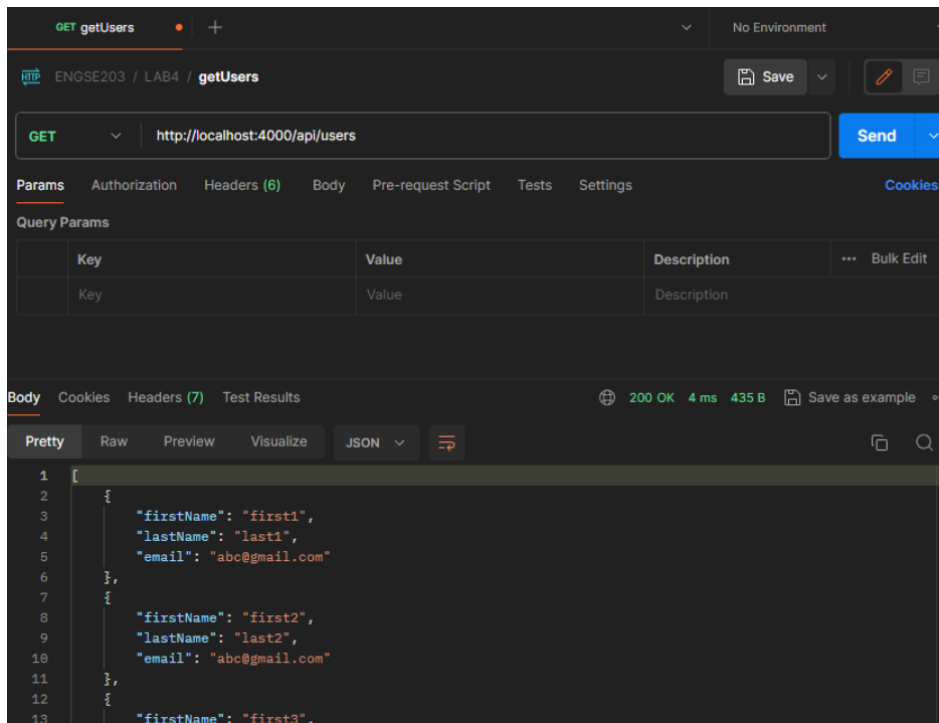
Store66@iMac-66-GOD api % npm run dev

> react-nodejs-example@1.0.0 dev
> nodemon ./index_movie.js localhost 4000

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node ./index_movie.js localhost 4000`
Running Environment: development
API Server running on http://0.0.0.0:4000

```

ทดลองรับ api จาก postman



At My-app

เข้าไปแก้ไข package.js โดย เปลี่ยน `start "set PORT=3001 && set NODE_OPTIONS=--openssl-legacy-provider && react-scripts start"` และ `"proxy": http://localhost:4000`

```

{} package.json X
backend > react-nodejs-example > my-app > {} package.json > ...
1  {
2    "name": "my-app",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^4.2.4",
7      "@testing-library/react": "^9.5.0",
8      "@testing-library/user-event": "^7.2.1",
9      "bootstrap": "^4.5.0",
10     "mysql": "^2.18.1",
11     "react": "^16.13.1",
12     "react-bootstrap": "^1.0.1",
13     "react-dom": "^16.13.1",
14     "react-scripts": "3.4.1"
15   },
16   "scripts": {
17     "start": "set PORT=3001 && set NODE_OPTIONS=--openssl-legacy-provider && react-scripts start",
18     "build": "react-scripts build",
19     "test": "react-scripts test",
20     "eject": "react-scripts eject"
21   },
22   "proxy": "http://localhost:4000",

```

ลอง run my app ด้วย npm start

```

Store66@iMac-66-GOD my-app % npm start

> my-app@0.1.0 start
> PORT=3001 react-scripts start

```

ผลลัพธ์

React With NodeJS

### Create User

First Name

Last Name

Email

Create

Users Created

3

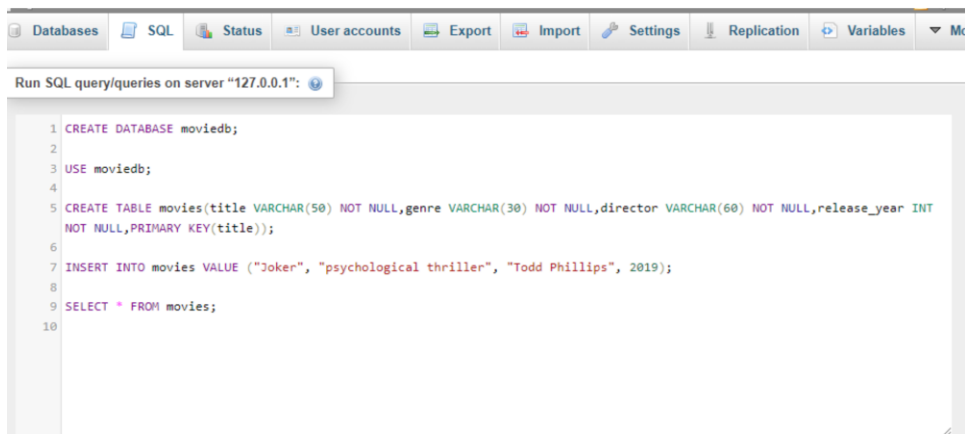
Get all Users

### Users

| User Id | Firstname | Lastname | Email         |
|---------|-----------|----------|---------------|
| 1       | first1    | last1    | abc@gmail.com |
| 2       | first2    | last2    | abc@gmail.com |
| 3       | first3    | last3    | abc@gmail.com |

เปิด xampp

## ทำการสร้าง database

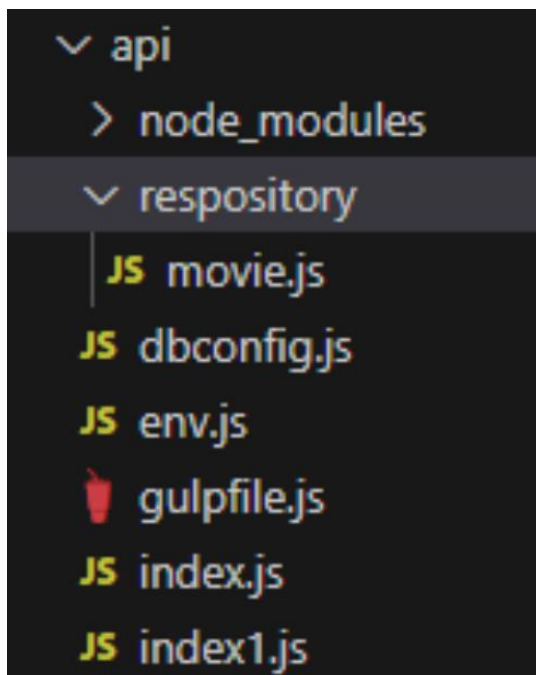


The screenshot shows a SQL client window with a toolbar at the top containing icons for Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables, and a dropdown menu. Below the toolbar, a text box indicates the connection to server "127.0.0.1". The main area displays a SQL script with line numbers 1 through 10. The script creates a database named 'moviedb', uses it, creates a table named 'movies' with columns 'title', 'genre', 'director', and 'release\_year', and inserts a record for the movie 'Joker'.

```
1 CREATE DATABASE moviedb;
2
3 USE moviedb;
4
5 CREATE TABLE movies(title VARCHAR(50) NOT NULL,genre VARCHAR(30) NOT NULL,director VARCHAR(60) NOT NULL,release_year INT
6 NOT NULL,PRIMARY KEY(title));
7
8 INSERT INTO movies VALUE ("Joker", "psychological thriller", "Todd Phillips", 2019);
9
10 SELECT * FROM movies;
```

## At api

สร้าง index1.js ที่ api และ repository/movie.js ตามในใบงาน



ทำการแก้ไข insert เป็นตามในรูป

```

130     Query = 'INSERT INTO movies SET ?';
131     pool.query(Query, post, function (error, results, fields) {
132     //pool.query(Query, function (error, results, fields) {
133         //if (error) throw error;
134
135         if (error) {
136             console.log('error_code_msg: ', error.code+':'+error.sqlMessage);
137             pool.end();
138             return resolve({
139                 statusCode: 405,
140                 returnCode: 9,
141                 message: error.code+':'+error.sqlMessage
142             });
143         }
144         else{
145             console.log('results: ',results);
146             if (results.affectedRows > 0) {
147                 pool.end();
148                 return resolve({
149                     statusCode: 200,
150                     returnCode: 1,
151                     message: 'Movie list was inserted',
152                 });
153             }

```

เข้าไปแก้ไข package.js โดย เปลี่ยนตรง dev เป็น "nodemon .index1.js localhost 4000"

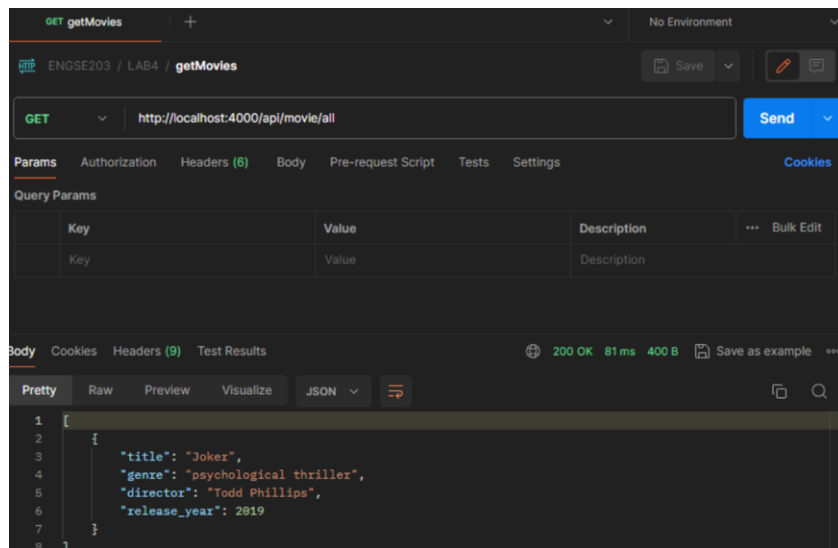
```

"scripts": {
  "start": "node server.js",
  "build": "webpack",
  "dev": "nodemon .index1.js localhost 4000"
}

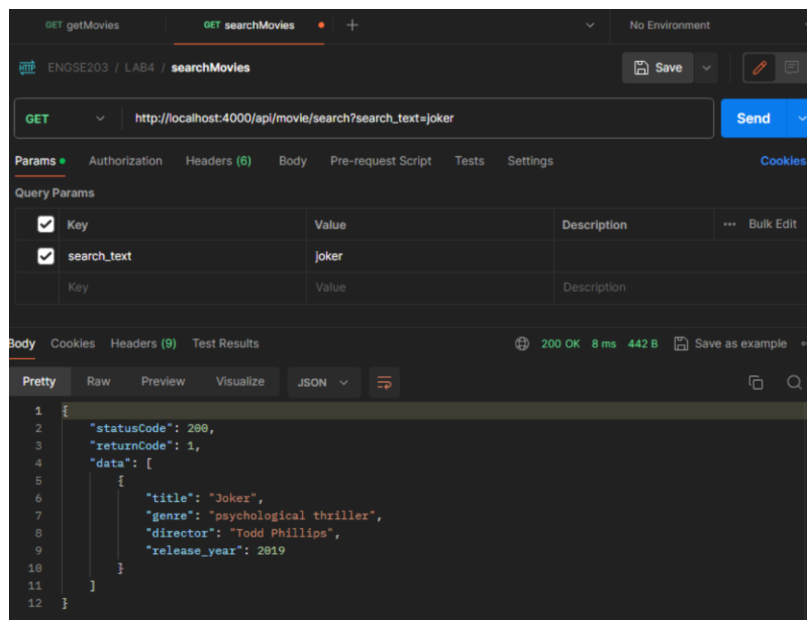
```

ทดลองรับ api จาก postman

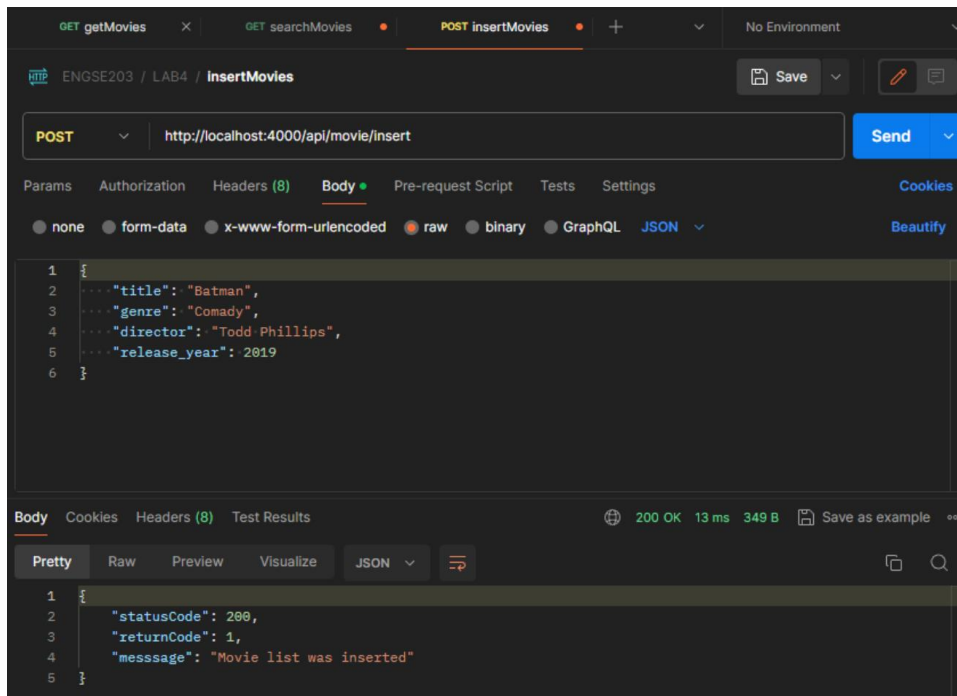
getMovies



## searchMovies



## insertMovies



insertMovies 실패

