

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

```
PS C:\Users\Missez\Desktop\LAB4\backend\react-nodejs-example\api> npm start

> react-nodejs-example@1.0.0 start
> node server.js

Server listening on the port::4000
```

ทดลองรับ api จาก postman

The screenshot shows the Postman interface for a GET request to `http://localhost:4000/api/users`. The request is saved and has a status of "200 OK" with a response time of "4 ms" and a body size of "435 B". The response body is displayed in the "Body" tab, showing a JSON array of three user objects.

Query Params

Key	Value	Description
Key	Value	Description

Body

```
1 [
2   {
3     "firstName": "first1",
4     "lastName": "last1",
5     "email": "abc@gmail.com"
6   },
7   {
8     "firstName": "first2",
9     "lastName": "last2",
10    "email": "abc@gmail.com"
11  },
12  {
13    "firstName": "first3",
```

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>

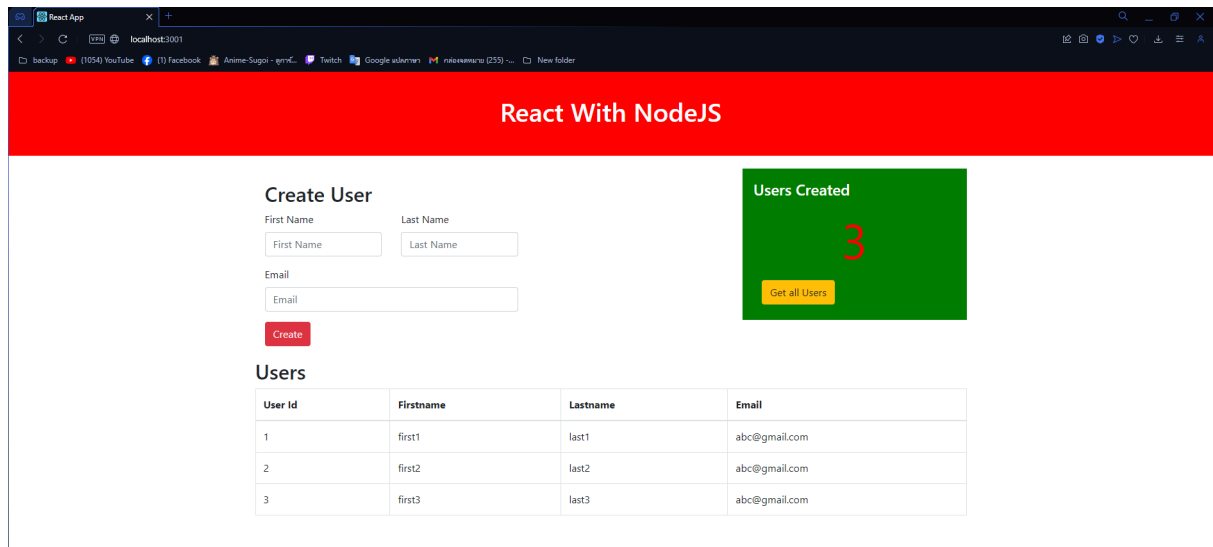
```
{ package.json X
backend > react-nodejs-example > my-app > {} package.json > ...
1  {
2    "name": "my-app",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^4.2.4",
7      "@testing-library/react": "^9.5.0",
8      "@testing-library/user-event": "^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",
23   "devDependencies": {}
24 }
```

ลอง run my app ด้วย npm start

```
PS C:\Users\Miszez\Desktop\LAB4\backend\react-nodejs-example\my-app> npm start

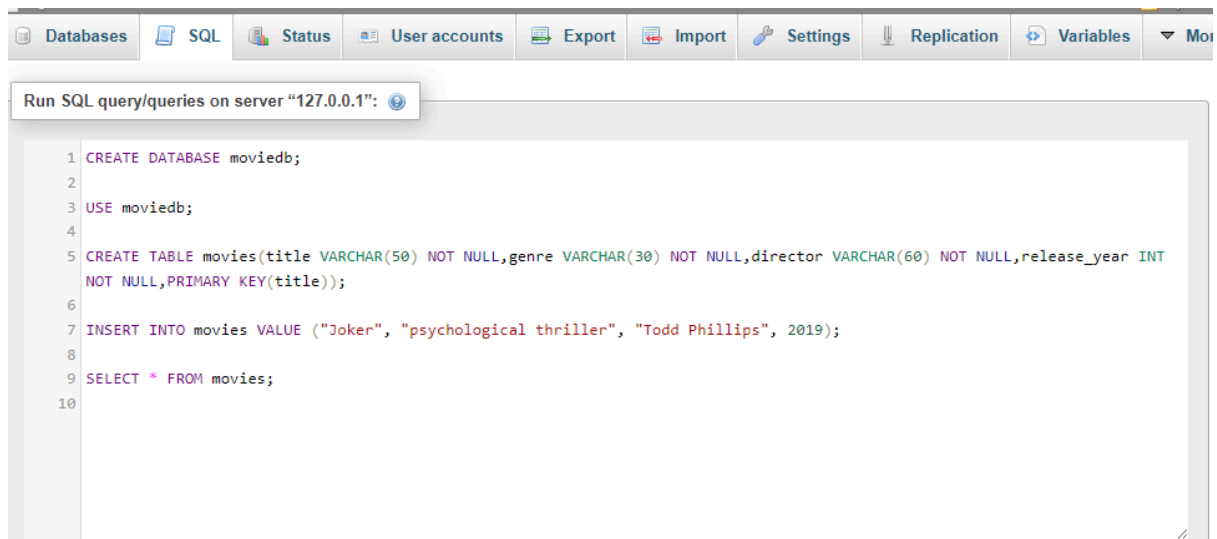
> my-app@0.1.0 start
> set PORT=3001 && set NODE_OPTIONS=--openssl-legacy-provider && react-scripts start
```

ผลลัพธ์



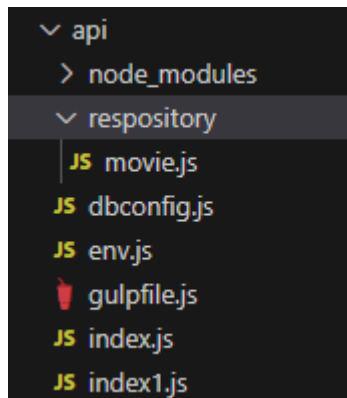
เปิด xampp

ทำการสร้าง database



At api

สร้าง index1.js ที่ api และ respository/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                 }
154             }
155         }
156     });
157 }
```

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

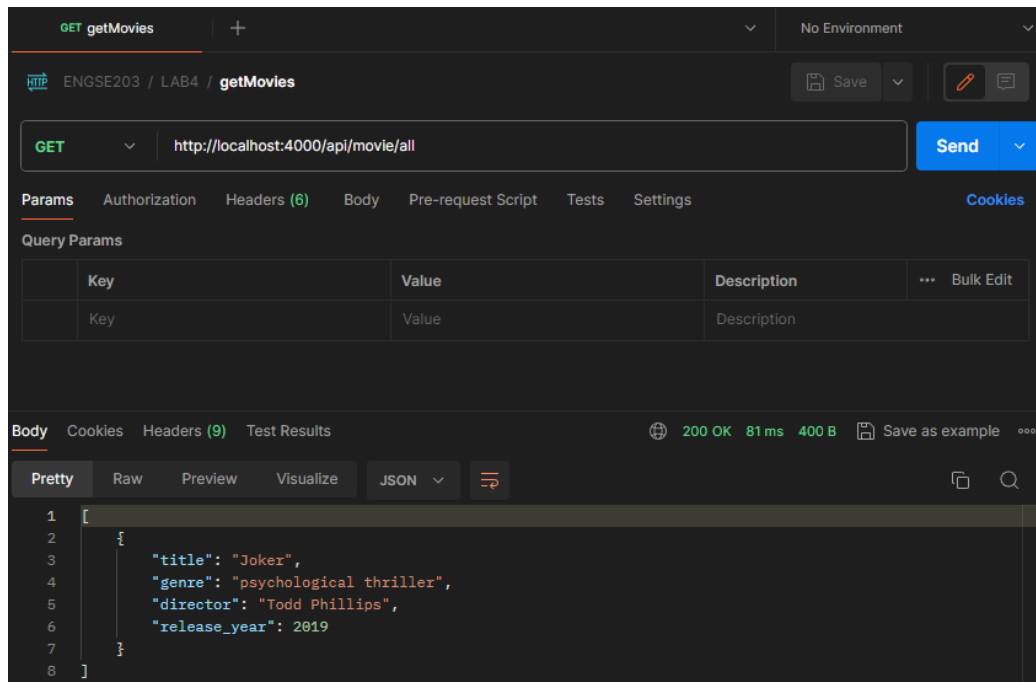
```
"scripts": {  
  "start": "node server.js",  
  "build": "webpack",  
  "dev": "nodemon .index1.js localhost 4000"  
}
```

Run npm run dev

```
C:\Users\Missez\Desktop\LAB4\backend\react-nodejs-example\api> npm run dev  
  
> react-nodejs-example@1.0.0 dev  
> nodemon .index1.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 .index1.js localhost 4000 index.js`  
Running Environment: development  
API Server running on http://0.0.0.0:4000
```

ทดลองรับ api จาก postman

getMovies



searchMovies

GET getMovies GET searchMovies + No Environment

ENGSE203 / LAB4 / searchMovies Save

GET http://localhost:4000/api/movie/search?search_text=joker Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	search_text	joker			
	Key	Value	Description		

Body Cookies Headers (9) Test Results 200 OK 8 ms 442 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "statusCode": 200,
3   "returnCode": 1,
4   "data": [
5     {
6       "title": "Joker",
7       "genre": "psychological thriller",
8       "director": "Todd Phillips",
9       "release_year": 2019
10    }
11  ]
12 }
```

insertMovies

GET getMovies GET searchMovies POST insertMovies + No Environment

ENGSE203 / LAB4 / insertMovies Save

POST http://localhost:4000/api/movie/insert Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   ...."title": "Batman",
3   ...."genre": "Comady",
4   ...."director": "Todd Phillips",
5   ...."release_year": 2019
6 }
```

Body Cookies Headers (8) Test Results 200 OK 13 ms 349 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "statusCode": 200,
3   "returnCode": 1,
4   "message": "Movie list was inserted"
5 }
```


insertMovies หน้า

The screenshot shows a REST client interface with a tab for the `POST insertMovies` endpoint. The URL is `http://localhost:4000/api/movie/insert`. The request body is a JSON object:

```
{
  "title": "Batman",
  "genre": "Comady",
  "director": "Todd Phillips",
  "release_year": 2019
}
```

The response status is `200 OK` with a response time of `10 ms` and a size of `382 B`. The response body, displayed in the 'Pretty' view, is:

```
{
  "statusCode": 405,
  "returnCode": 9,
  "message": "ER_DUP_ENTRY:Duplicate entry 'Batman' for key 'PRIMARY'"
}
```

The error message indicates a duplicate entry for the 'PRIMARY' key.