

LAB 4 ENGSE203

localhost:3001

React With NodeJS

Create User

First Name

First Name

Last Name

Last Name

Email

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

Font end Source code

You, 7 days ago | 2 authors (You and others)

```
1 import React, { useState, useEffect } from "react";
2 import "bootstrap/dist/css/bootstrap.min.css";
3 import "./App.css";
4 import { Header } from "./components/Header";
5 import { Users } from "./components/Users";
6 import { DisplayBoard } from "./components/DisplayBoard";
7 import CreateUser from "./components/CreateUser";
8 import { getAllUsers, createUser } from "./services/UserService";
9 //-----
10 import { Movies } from "./components/Movies";
11 import { getAllMovies, createMovie } from "./services/MovieService";
12
13 function App() {
14   const [user, setUser] = useState({});
15   const [users, setUsers] = useState([]);
16   const [numberOfUsers, setNumberOfUsers] = useState(0);
17
18   //-----
19
20   const [movies, setMovies] = useState([]);
21
22   const userCreate = (e) => {
23     createUser(user).then((response) => {
24       console.log(response);
25       setNumberOfUsers(numberOfUsers + 1);
26     });
27   };
28
29   const fetchAllUsers = () => {
30     getAllUsers().then((users) => {
31       console.log(users);
32       setUsers(users);
33       setNumberOfUsers(users.length);
34     });
35   };
36
37   useEffect(() => {
38     getAllUsers().then((users) => {
39       console.log(users);
40       setUsers(users);
41       setNumberOfUsers(users.length);
42     });
43
44     getAllMovies().then((movies) => {
45       console.log(movies);
46       setMovies(movies);
47       //setNumberOfUsers(users.length)
48     });
49   }, []);
50
51   const onChangeForm = (e) => {
52     if (e.target.name === "firstname") {
```

```

53     user.firstName = e.target.value;
54   } else if (e.target.name === "lastname") {
55     user.lastName = e.target.value;
56   } else if (e.target.name === "email") {
57     user.email = e.target.value;
58   }
59   setUser(user);
60 };
61
62 return (
63   <div className="App">
64     <Header></Header>
65     <div className="container mrgnbtm"> You, 7 days ago • Uncomm
66       <div className="row">
67         <div className="col-md-8">
68           <CreateUser
69             user={user}
70             onChangeForm={onChangeForm}
71             createUser={userCreate}
72           ></CreateUser>
73         </div>
74         <div className="col-md-4">
75           <DisplayBoard
76             numberOfUsers={numberOfUsers}
77             getAllUsers={fetchAllUsers}
78           ></DisplayBoard>
79         </div>
80       </div>
81     </div>
82     <div className="row mrgnbtm">
83       <Users users={users}></Users>
84     </div>
85     <div className="row mrgnbtm">
86       <Movies movies={movies}></Movies>
87     </div>
88   </div>
89 );
90 }
91
92 export default App;
93

```

Server Source code

```
You, last week | 2 authors (bbachi and others)
1  const express = require("express");
2  const path = require("path");
3  const app = express(),
4    bodyParser = require("body-parser");
5  port = 4000;
6
7  // place holder for the data
8  const users = [
9    {
10     firstName: "first1",
11     lastName: "last1",
12     email: "abc@gmail.com",
13   },
14   {
15     firstName: "first2",
16     lastName: "last2",
17     email: "abc@gmail.com",
18   },
19   {
20     firstName: "first3",
21     lastName: "last3",
22     email: "abc@gmail.com",
23   },
24 ];
25
26 app.use(bodyParser.json());
27 app.use(express.static(path.join(__dirname, "../my-app/build")));
28
29 app.get("/api/users", (req, res) => {
30   console.log("api/users called!");
31   res.json(users);
32 });
33
34 app.post("/api/user", (req, res) => {
35   const user = req.body.user;
36   console.log("Adding user:::::", user);
37   users.push(user);
38   res.json("user addedd");
39 });
40
41 app.get("/", (req, res) => {
42   res.sendFile(path.join(__dirname, "../my-app/build/index.html"));
43 });
44
45 app.listen(port, () => {
46   console.log(`Server listening on the port:${port}`);
47 });
48
```

ทดสอบยิง API

GET

The screenshot shows the Postman interface for a GET request to `http://localhost:4000/api/users`. The request is saved in the "getUsers" collection. The response is a 200 OK status with a 7ms response time and 499 B of data. The response body is displayed in JSON format, showing an array of four 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",
14    "lastName": "last3",
15    "email": "abc@gmail.com"
16  },
17  {
18    "firstName": "Fuwa",
19    "lastName": "waa",
20    "email": "abcc@gmail.com"
21  }
22 }
```

```
> react-nodejs-example@1.0.0 dev
> nodemon ./server.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 ./server.js localhost 4000`
Server listening on the port::4000
api/users called!
```

POST

The screenshot shows a REST client interface with a POST request configured. The URL is `http://localhost:4000/api/user`. The request body is a JSON object: `{ "user": { "firstName": "Fuwa", "lastName": "waa", "email": "abcc@gmail.com" } }`. The response status is `200 OK` with a response time of `25 ms` and a size of `247 B`. The response body is `"user addedd"`.

```
1 {
2   ... "user": {
3     ... "firstName": "Fuwa",
4     ... "lastName": "waa",
5     ... "email": "abcc@gmail.com"
6   }
7 }
```

```
1 "user addedd"
```

```
Server listening on the port::4000
api/users called!
Adding user:::: { firstName: 'Fuwa', lastName: 'waa', email: 'abcc@gmail.com' }
█
```


ตรวจสอบ User ที่ add เข้ามา

localhost:3001

React With NodeJS

Create User

First Name

Last Name

First Name

Last Name

Email

Email

Create

Users Created

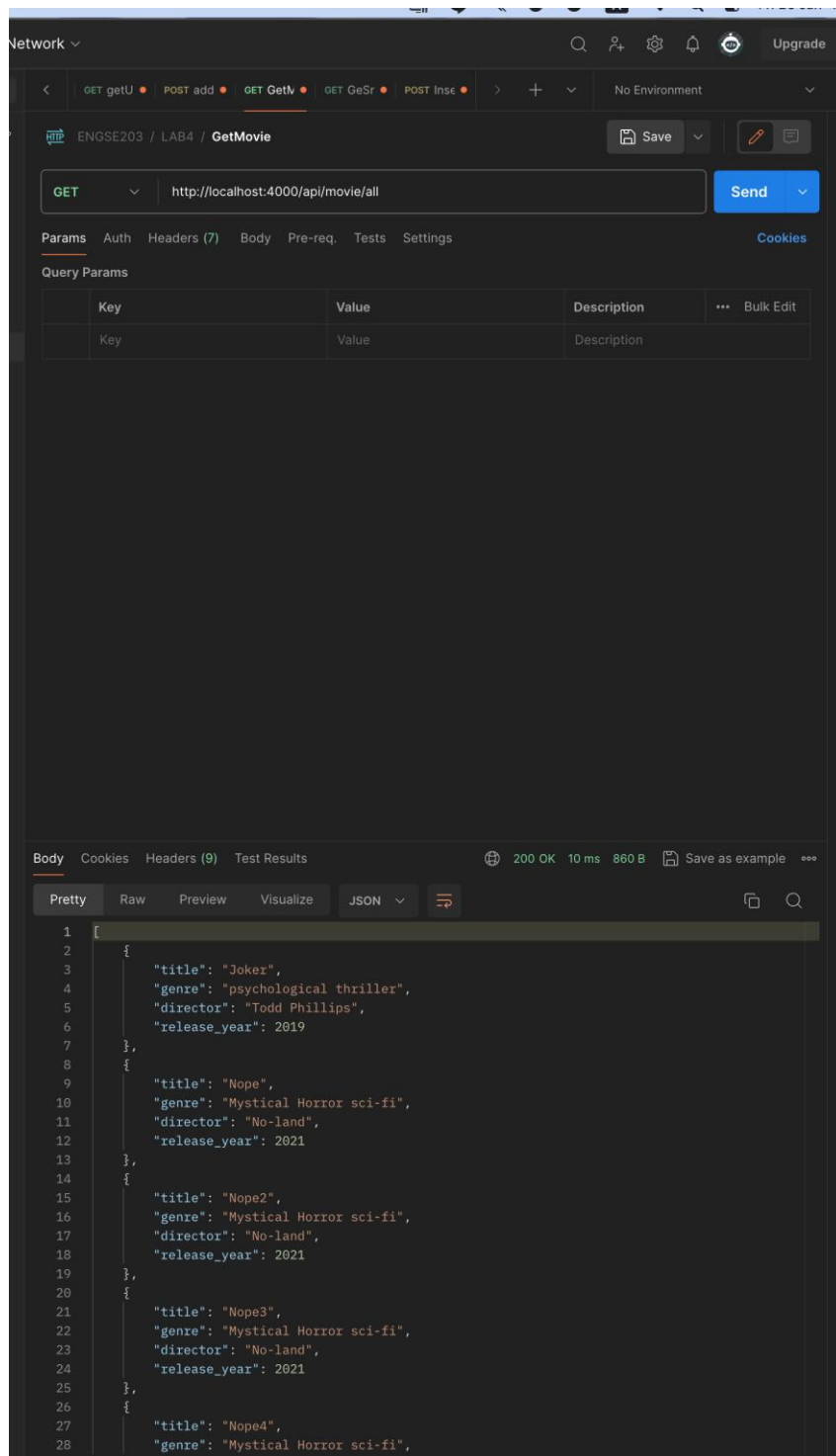
4

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
4	Fuwa	waa	abcc@gmail.com

ทดสอบยิง Api ของ Movies



ทดสอบยิง Api move Search



ENGSE203 / LAB4 / GeSrcMovie



Save



GET



http://localhost:4000/api/movie/search?search_text=jok

Send



Params • Auth Headers (7) Body Pre-req. Tests Settings

Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	⋮ Bulk Edit
<input checked="" type="checkbox"/>	search_text	jok		
	Key	Value	Description	

Body Cookies Headers (9) Test Results



200 OK

7 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 }
```

GET getU

POST add

GET GetV

GET GeSr

POST Inse

>

+

▼

No Environment

▼

🔧

ENGSE203 / LAB4 / GeSrcMovie

Save

🔗

💬

📄

GET

http://localhost:4000/api/movie/search?search_text=No

Send

💬

Params

Auth

Headers (7)

Body

Pre-req.

Tests

Settings

Cookies

</>

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	⋮ Bulk Edit
<input checked="" type="checkbox"/>	search_text	No		
	Key	Value	Description	

Body

Cookies

Headers (9)

Test Results

🌐 200 OK 6 ms 803 B

Save as example

⋮

Pretty

Raw

Preview

Visualize

JSON

🔍

📄

🔍

```
1 {
2   "statusCode": 200,
3   "returnCode": 1,
4   "data": [
5     {
6       "title": "Nope",
7       "genre": "Mystical Horror sci-fi",
8       "director": "No-land",
9       "release_year": 2021
10    },
11    {
12      "title": "Nope2",
13      "genre": "Mystical Horror sci-fi",
14      "director": "No-land",
15      "release_year": 2021
16    },
17    {
18      "title": "Nope3",
19      "genre": "Mystical Horror sci-fi",
20      "director": "No-land",
21      "release_year": 2021
22    },
23    {
24      "title": "Nope4",
25      "genre": "Mystical Horror sci-fi",
26      "director": "No-land",
27      "release_year": 2021
28    },
29  ]
30 }
```

ทดสอบ Api Insert

The screenshot displays a REST client interface with a dark theme. At the top, a tab bar shows several requests, with the 'POST Inse' tab selected. Below the tab bar, the URL bar shows 'http://localhost:4000/api/movie/insert' and the method 'POST'. The 'Body' tab is active, showing a JSON payload:

```
{  "title": "MyHeart",  "genre": "Mystical·Horror·sci-fi·Romance",  "director": "someone",  "release_year": 2021}
```

. The bottom section shows the response, which is a 200 OK status with a 30 ms response time and 349 B of data. The response body is displayed in a 'Pretty' JSON format:

```
{  "statusCode": 200,  "returnCode": 1,  "message": "Movie list was inserted"}
```

GET getU • POST add • GET GetM • GET GeSr • **POST Inse •** > + ▾ No Environment ▾

HTTP ENGSE203 / LAB4 / **InsertMovie** Save ▾

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

Params Auth Headers (9) **Body •** Pre-req. Tests Settings Cookies Beautify

raw ▾ **JSON** ▾

```
1 {
2   ..... "title": "MyHeart",
3   ..... "genre": "Mystical·Horror·sci-fi·Romance",
4   ..... "director": "someone",
5   ..... "release_year": 2021
6 }
7
```

Body Cookies Headers (8) Test Results 200 OK 30 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 }
```

ทดสอบ ใส่ข้อมูลซ้ำเพื่อไม่ให้ Api พัง

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:4000/api/movie/insert`
- Method:** `POST`
- Body (JSON):**

```
{
  "title": "MyHeart",
  "genre": "Mystical Horror sci-fi Romance",
  "director": "someone",
  "release_year": 2021
}
```
- Status:** `200 OK`, `10 ms`, `385 B`
- Response Body (JSON):**

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

The response indicates a database error (ER_DUP_ENTRY) because the title 'MyHeart' already exists in the database.