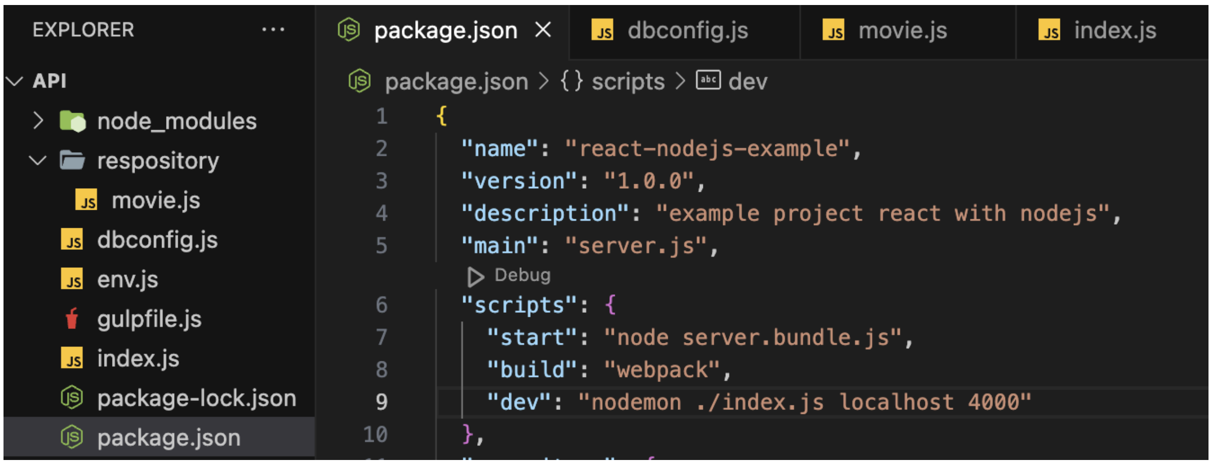
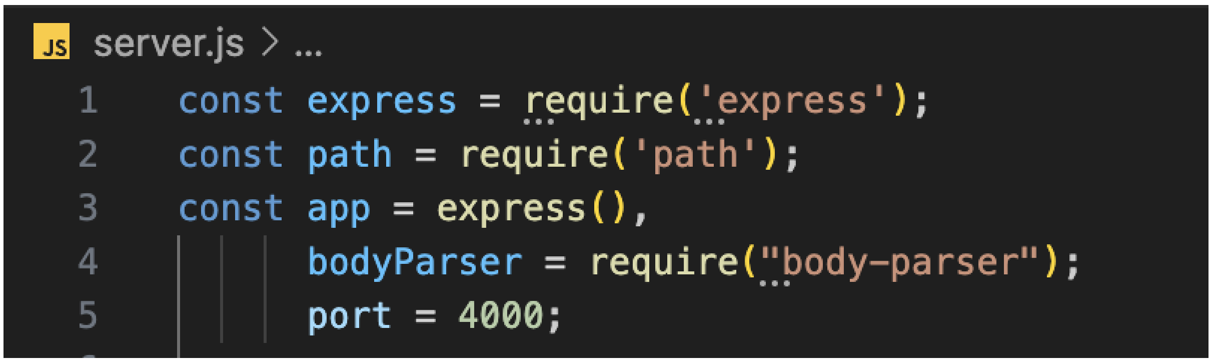
LAB4



package.json > { } scripts > dev

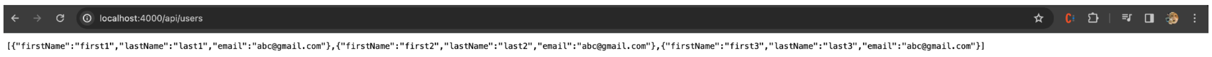
```json
{
  "name": "react-nodejs-example",
  "version": "1.0.0",
  "description": "example project react with nodejs",
  "main": "server.js",
  "scripts": {
    "start": "node server.bundle.js",
    "build": "webpack",
    "dev": "nodemon ./index.js localhost 4000"
  },
```



server.js > ...

```javascript
const express = require('express');
const path = require('path');
const app = express(),
      bodyParser = require("body-parser");
      port = 4000;
```

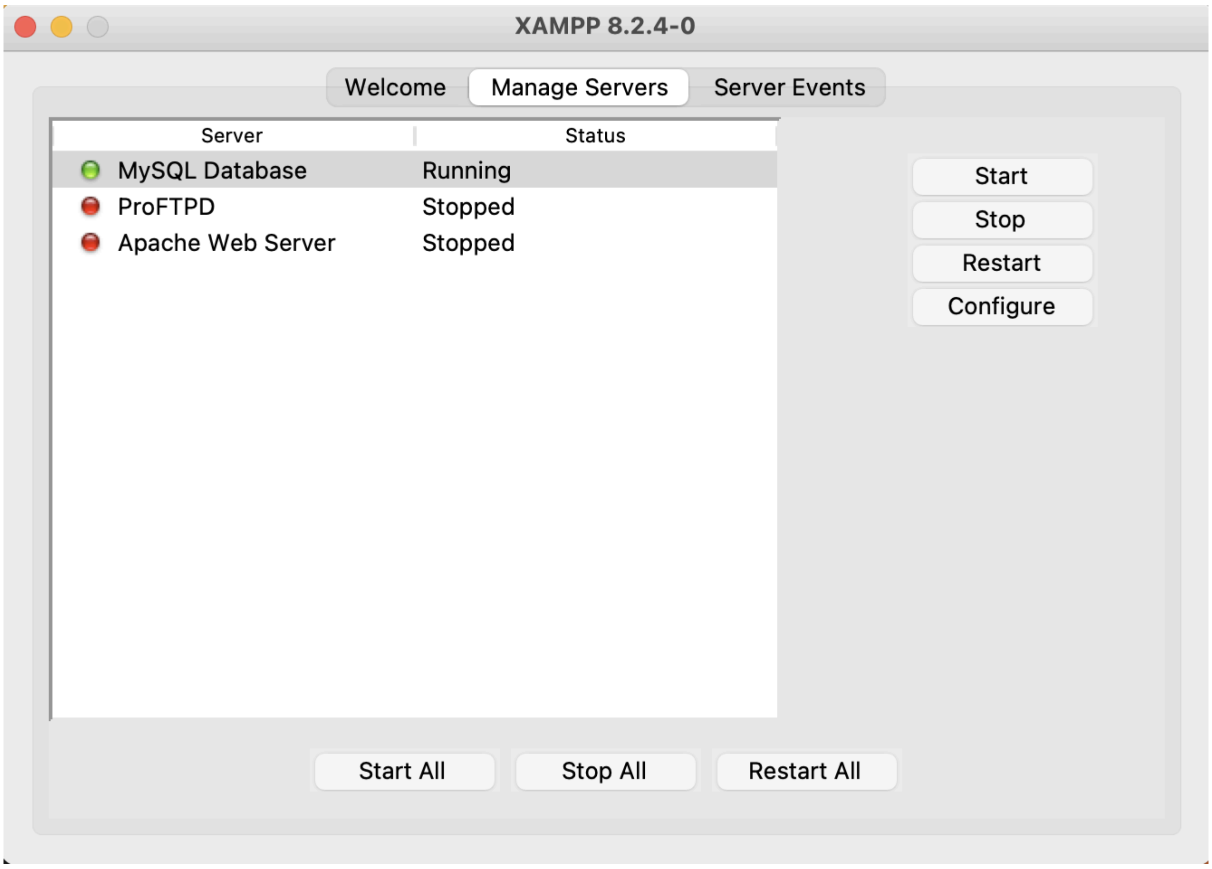
เปลี่ยน Port ของ backend เป็น 4000 หน้า package.json , server.js



localhost:4000/api/users

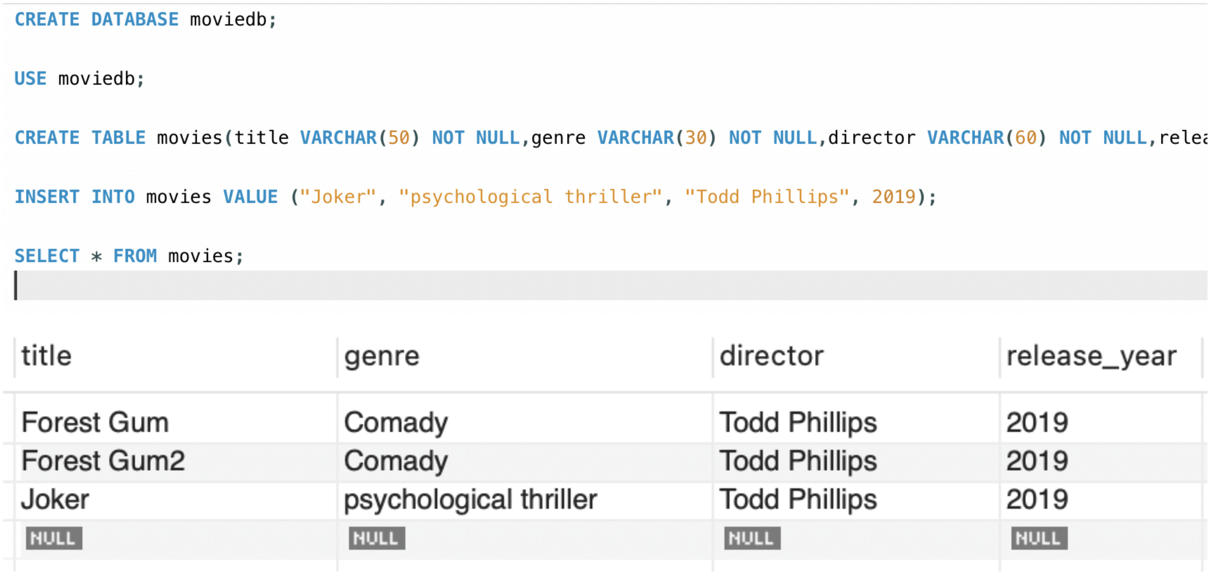
```
[{"firstName":"first1","lastName":"last1","email":"abc@gmail.com"},{"firstName":"first2","lastName":"last2","email":"abc@gmail.com"},{"firstName":"first3","lastName":"last3","email":"abc@gmail.com"}]
```

ทดสอบ



Server	Status
● MySQL Database	Running
● ProFTPD	Stopped
● Apache Web Server	Stopped

เปิด MYSQL ใน XAMPP

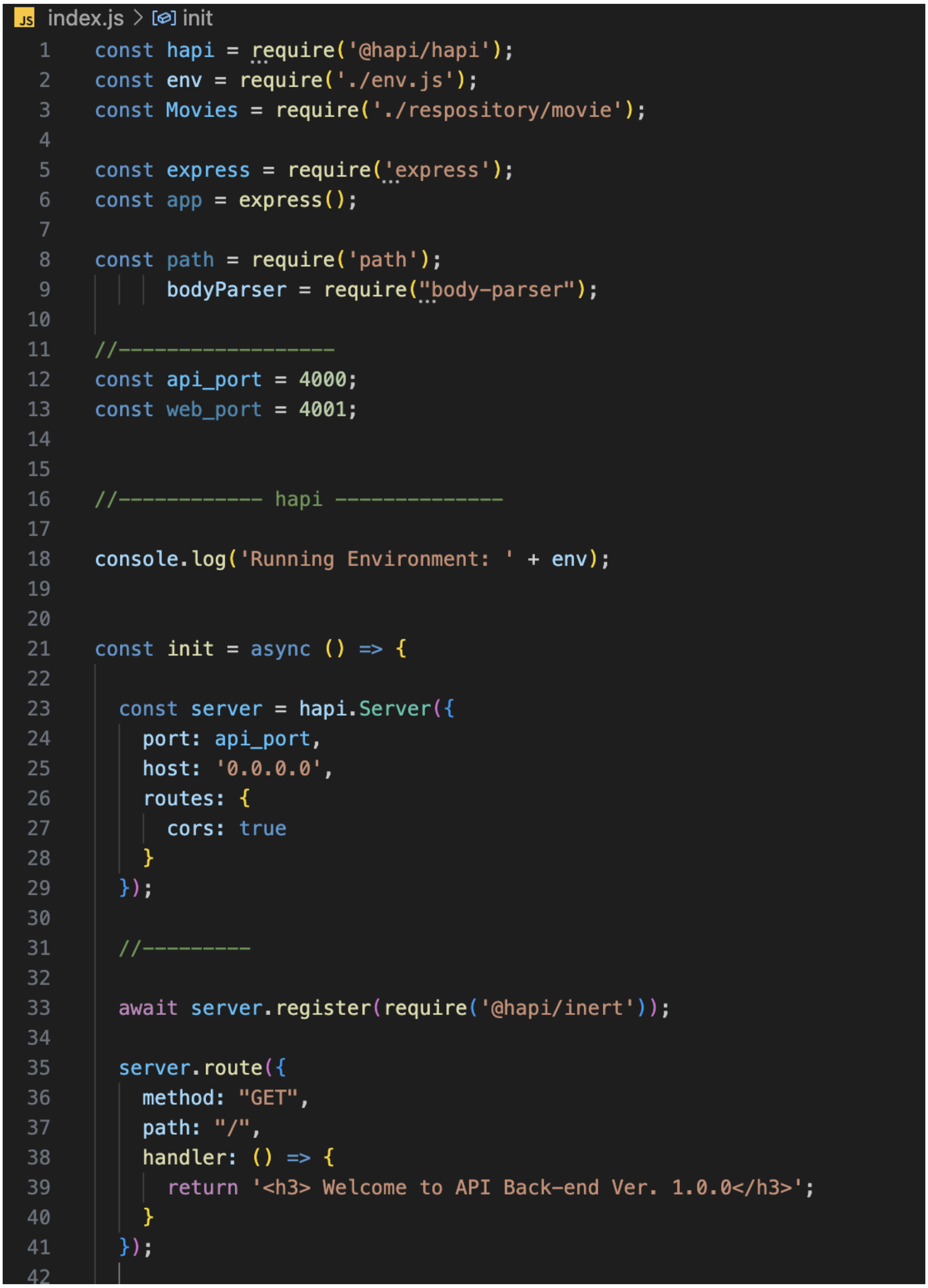


```
CREATE DATABASE moviedb;

USE moviedb;

CREATE TABLE movies(title VARCHAR(50) NOT NULL,genre VARCHAR(30) NOT NULL,director VARCHAR(60) NOT NULL,relea

INSERT INTO movies VALUE ("Joker", "psychological thriller", "Todd Phillips", 2019);

SELECT * FROM movies;
```

title	genre	director	release_year
Forest Gum	Comady	Todd Phillips	2019
Forest Gum2	Comady	Todd Phillips	2019
Joker	psychological thriller	Todd Phillips	2019
NULL	NULL	NULL	NULL

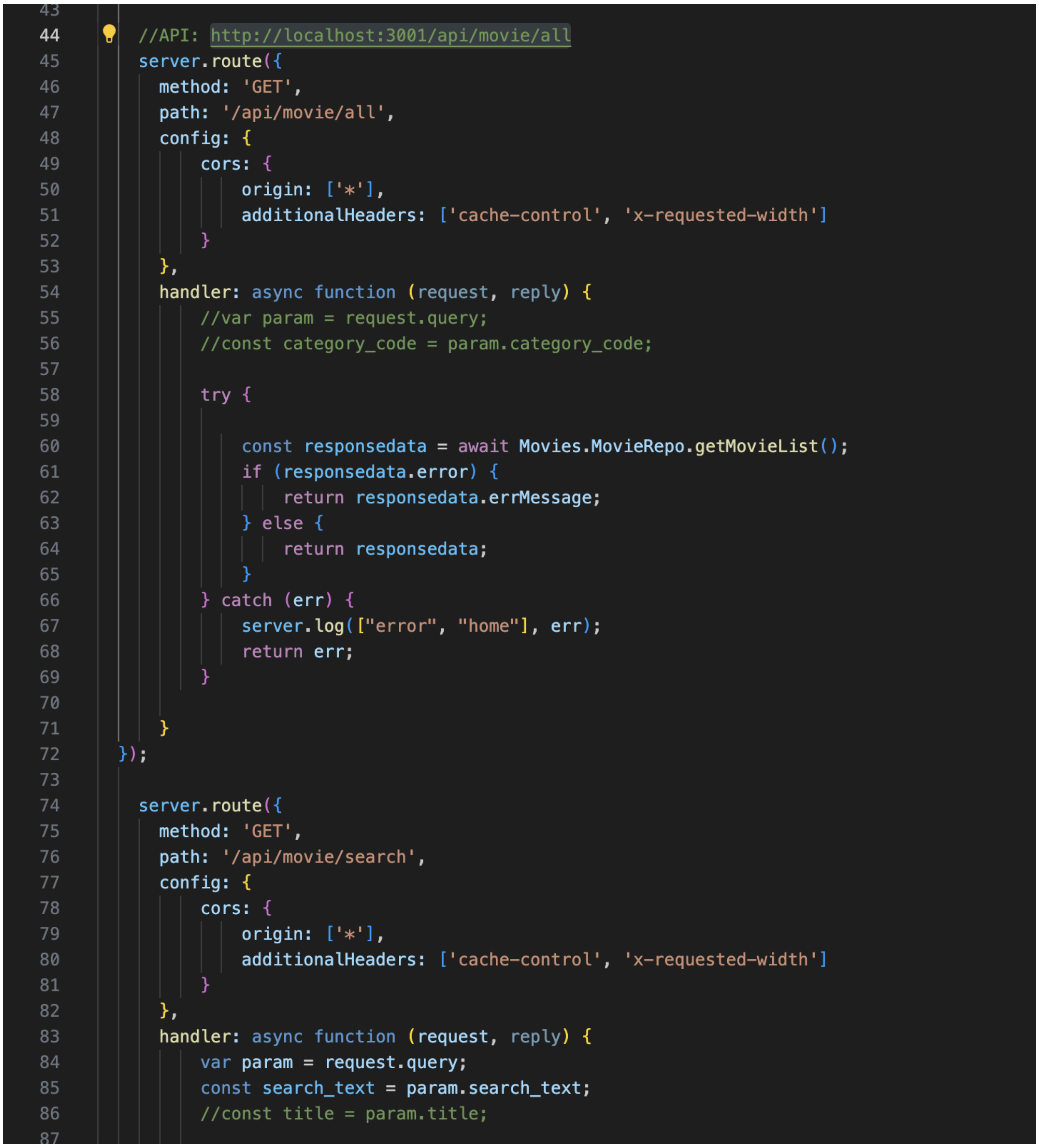
สร้าง database , เพิ่มตาราและเพิ่มข้อมูลลงในตาราง

สร้าง index.js มาใน backend และใส่โค้ด

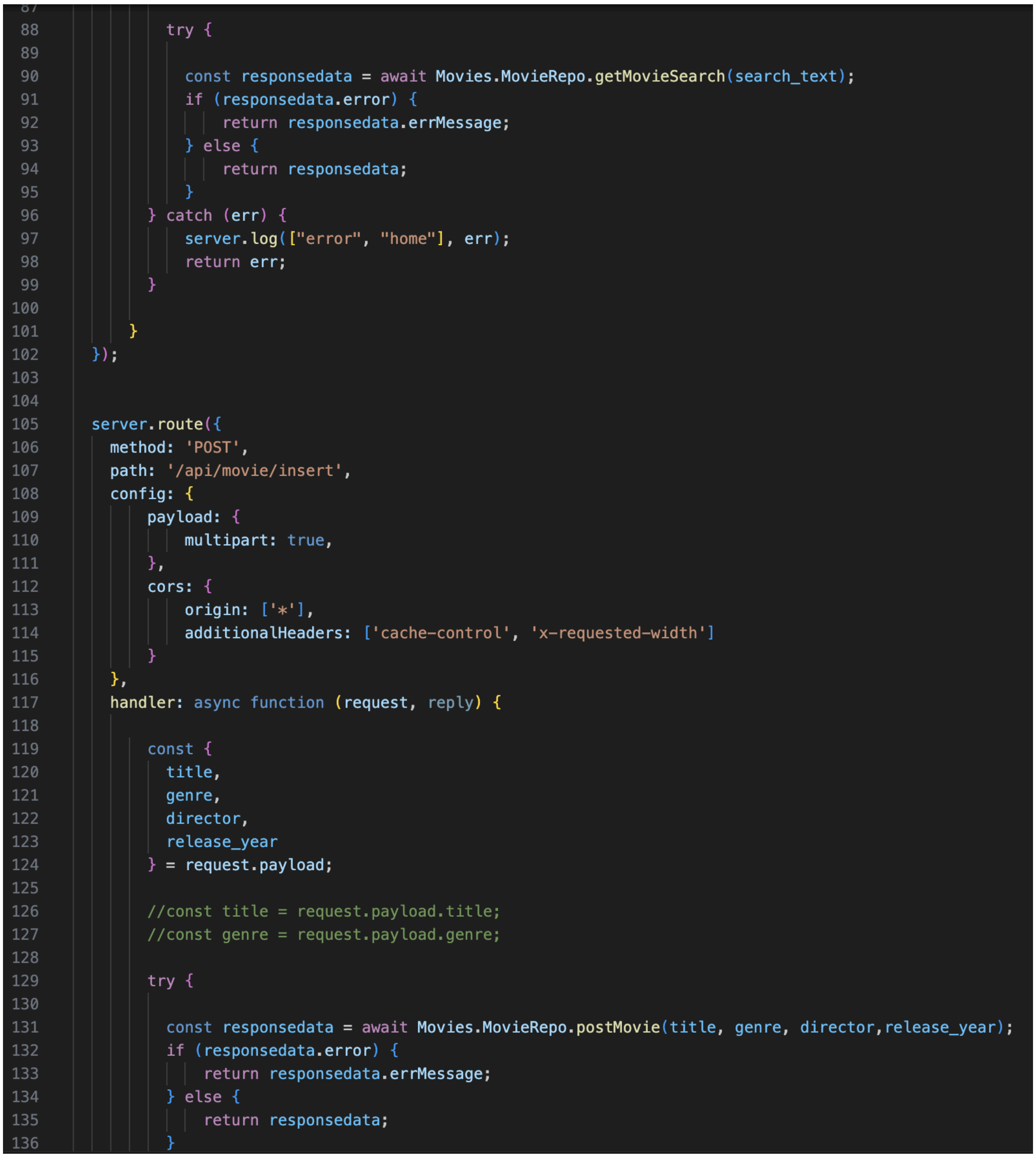


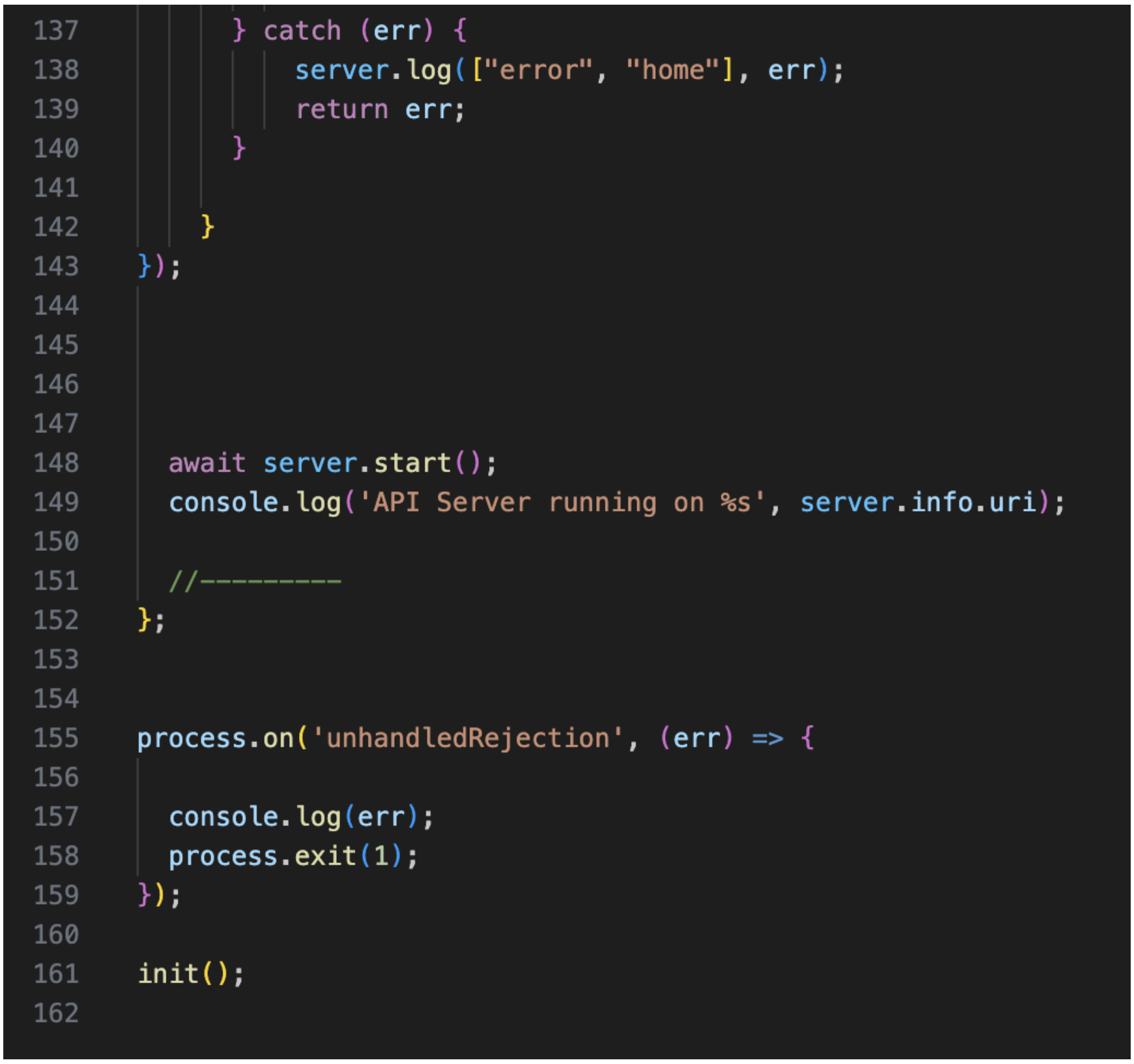
index.js > init

```javascript
const hapi = require('@hapi/hapi');
const env = require('./env.js');
const Movies = require('./respository/movie');

const express = require('express');
const app = express();

const path = require('path');
      bodyParser = require("body-parser");

//-----------------
const api_port = 4000;
const web_port = 4001;


//------------ hapi --------------

console.log('Running Environment: ' + env);


const init = async () => {

  const server = hapi.Server({
    port: api_port,
    host: '0.0.0.0',
    routes: {
      cors: true
    }
  });

  //---------

  await server.register(require('@hapi/inert'));

  server.route({
    method: "GET",
    path: "/",
    handler: () => {
      return '<h3> Welcome to API Back-end Ver. 1.0.0</h3>';
    }
  });

```



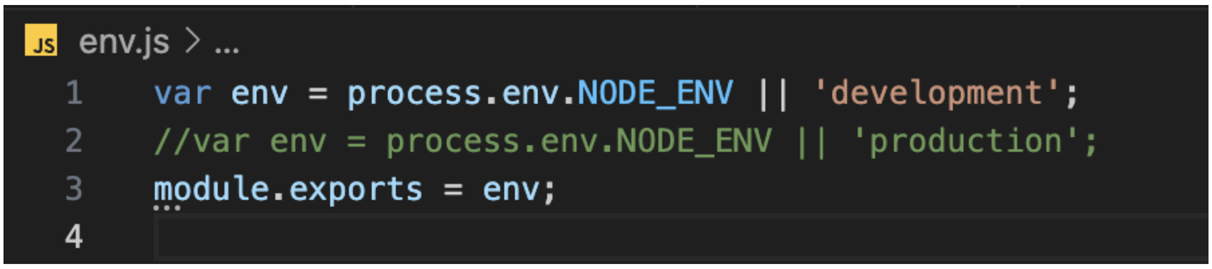
```
  //API: http://localhost:3001/api/movie/all
  server.route({
    method: 'GET',
    path: '/api/movie/all',
    config: {
        cors: {
            origin: ['*'],
            additionalHeaders: ['cache-control', 'x-requested-width']
        }
    },
    handler: async function (request, reply) {
        //var param = request.query;
        //const category_code = param.category_code;

        try {

            const responsedata = await Movies.MovieRepo.getMovieList();
            if (responsedata.error) {
                return responsedata.errMessage;
            } else {
                return responsedata;
            }
        } catch (err) {
            server.log(["error", "home"], err);
            return err;
        }

    }
});

  server.route({
    method: 'GET',
    path: '/api/movie/search',
    config: {
        cors: {
            origin: ['*'],
            additionalHeaders: ['cache-control', 'x-requested-width']
        }
    },
    handler: async function (request, reply) {
        var param = request.query;
        const search_text = param.search_text;
        //const title = param.title;
```


```
        try {

          const responsedata = await Movies.MovieRepo.getMovieSearch(search_text);
          if (responsedata.error) {
              return responsedata.errMessage;
          } else {
              return responsedata;
          }
      } catch (err) {
          server.log(["error", "home"], err);
          return err;
      }

    }
});


server.route({
  method: 'POST',
  path: '/api/movie/insert',
  config: {
      payload: {
          multipart: true,
      },
      cors: {
          origin: ['*'],
          additionalHeaders: ['cache-control', 'x-requested-width']
      }
  },
  handler: async function (request, reply) {

      const {
        title,
        genre,
        director,
        release_year
      } = request.payload;

      //const title = request.payload.title;
      //const genre = request.payload.genre;

      try {

        const responsedata = await Movies.MovieRepo.postMovie(title, genre, director,release_year);
        if (responsedata.error) {
            return responsedata.errMessage;
        } else {
            return responsedata;
        }
```



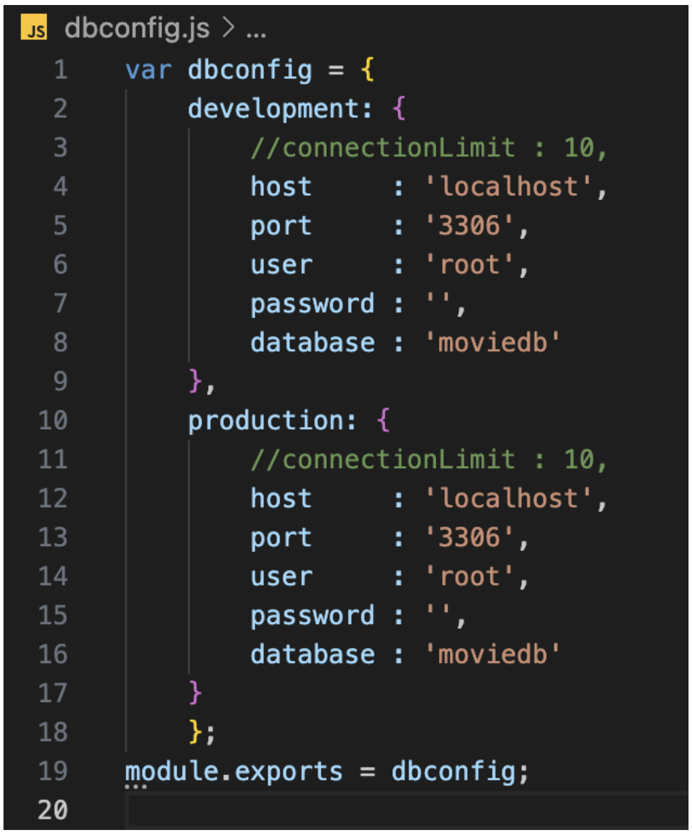
```
            } catch (err) {
                server.log(["error", "home"], err);
                return err;
            }

        }
});



  await server.start();
  console.log('API Server running on %s', server.info.uri);

  //---------
};


process.on('unhandledRejection', (err) => {

  console.log(err);
  process.exit(1);
});

init();

```

สร้าง env.js เพื่อเก็บ config ว่าทำงานอยู่ในสภาพแวดล้อมใด


env.js > ...

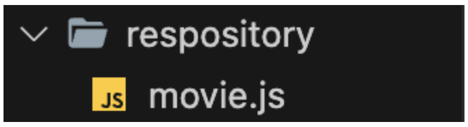
```
var env = process.env.NODE_ENV || 'development';
//var env = process.env.NODE_ENV || 'production';
module.exports = env;

```

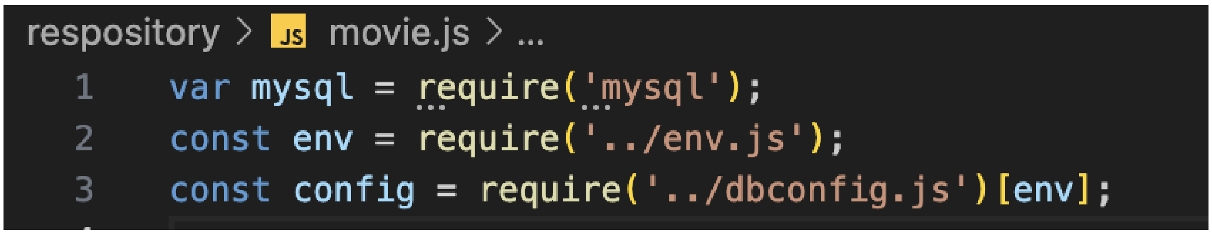
สร้าง dbconfig.js


dbconfig.js > ...

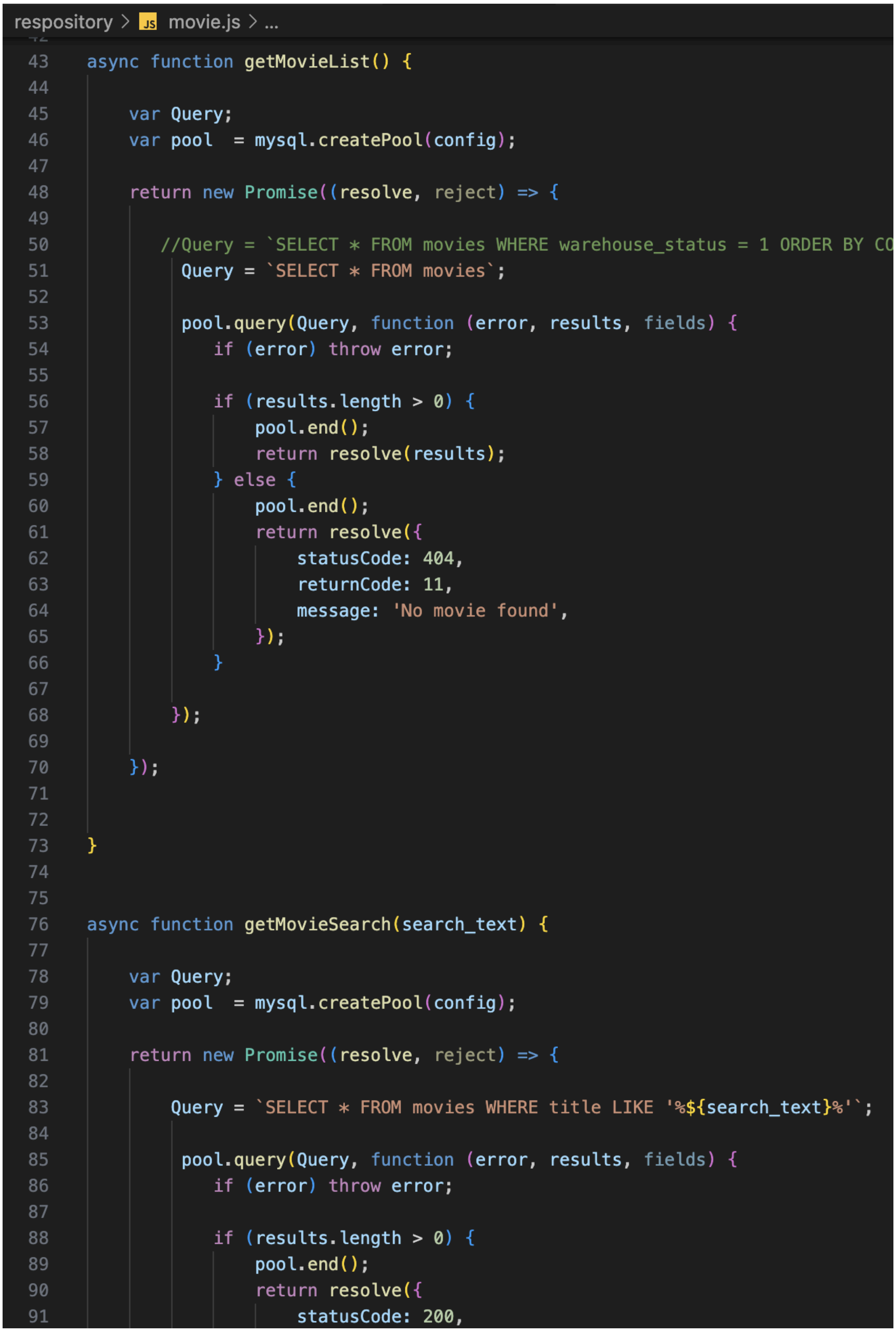
```
var dbconfig = {
    development: {
        //connectionLimit : 10,
        host     : 'localhost',
        port     : '3306',
        user     : 'root',
        password : '',
        database : 'moviedb'
    },
    production: {
        //connectionLimit : 10,
        host     : 'localhost',
        port     : '3306',
        user     : 'root',
        password : '',
        database : 'moviedb'
    }
    };
module.exports = dbconfig;

```

สร้างโฟลเดอร์ respository และสร้างไฟล์ movie.js

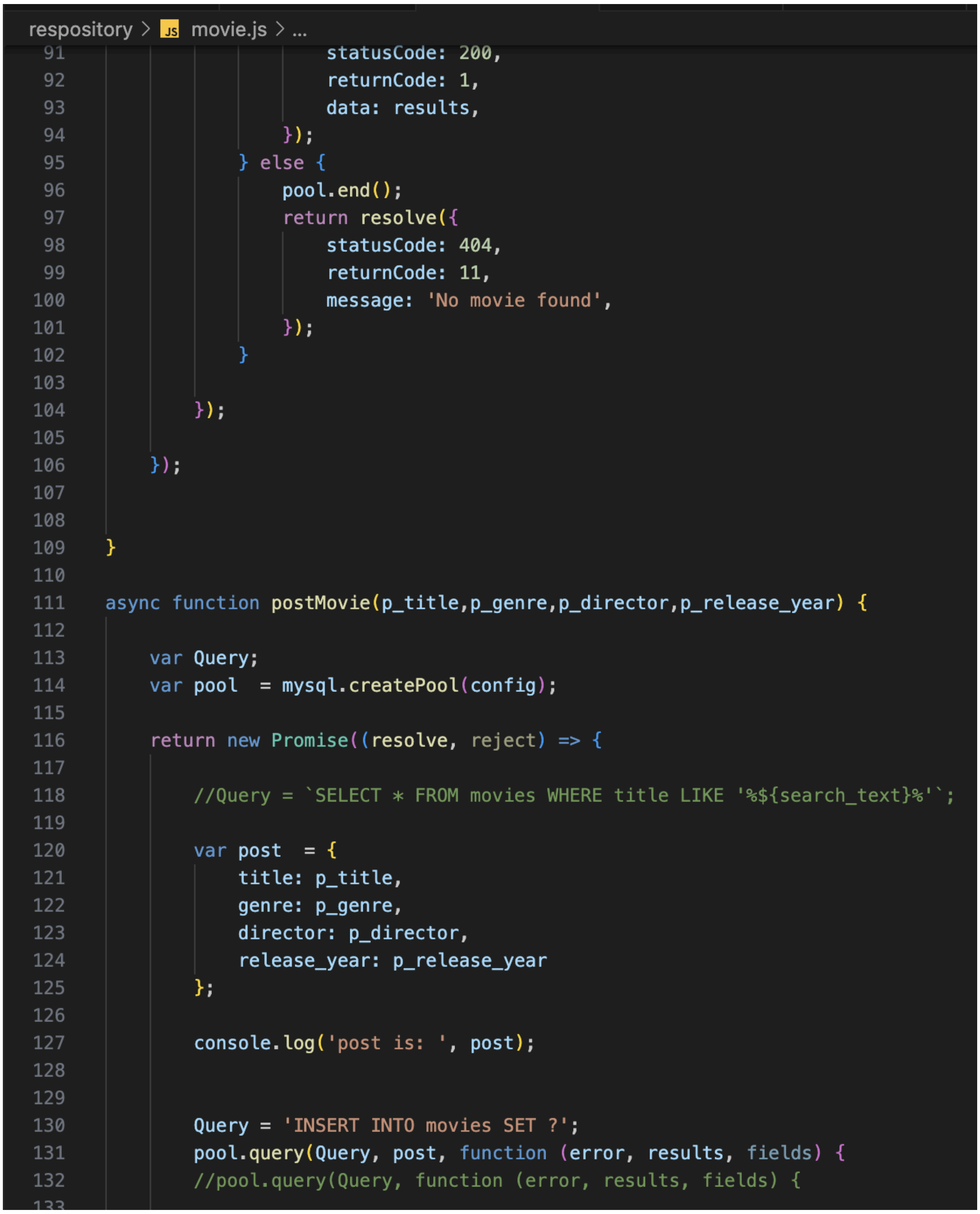



respository > movie.js > ...

```
var mysql = require('mysql');
const env = require('../env.js');
const config = require('../dbconfig.js')[env];
```


respository > movie.js > ...

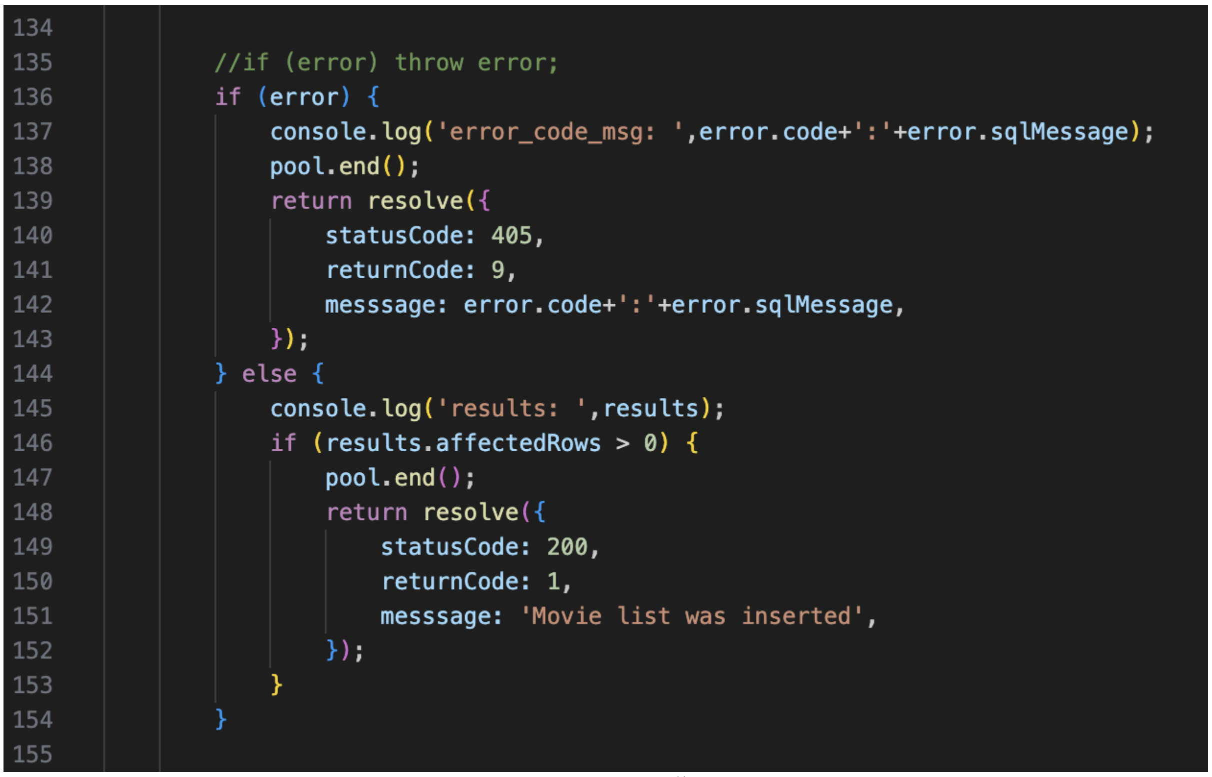
```
async function getMovieList() {

    var Query;
    var pool  = mysql.createPool(config);

    return new Promise((resolve, reject) => {

       //Query = `SELECT * FROM movies WHERE warehouse_status = 1 ORDER BY CO
         Query = `SELECT * FROM movies`;

         pool.query(Query, function (error, results, fields) {
            if (error) throw error;

            if (results.length > 0) {
                pool.end();
                return resolve(results);
            } else {
                pool.end();
                return resolve({
                    statusCode: 404,
                    returnCode: 11,
                    message: 'No movie found',
                });
            }

        });

    });


}


async function getMovieSearch(search_text) {

    var Query;
    var pool  = mysql.createPool(config);

    return new Promise((resolve, reject) => {

        Query = `SELECT * FROM movies WHERE title LIKE '%${search_text}%'`;

         pool.query(Query, function (error, results, fields) {
            if (error) throw error;

            if (results.length > 0) {
                pool.end();
                return resolve({
                    statusCode: 200,
```



respository > movie.js > ...

```
                    statusCode: 200,
                    returnCode: 1,
                    data: results,
                });
            } else {
                pool.end();
                return resolve({
                    statusCode: 404,
                    returnCode: 11,
                    message: 'No movie found',
                });
            }

        });

    });


}

async function postMovie(p_title,p_genre,p_director,p_release_year) {

    var Query;
    var pool  = mysql.createPool(config);

    return new Promise((resolve, reject) => {

        //Query = `SELECT * FROM movies WHERE title LIKE '%${search_text}%'`;

        var post  = {
            title: p_title,
            genre: p_genre,
            director: p_director,
            release_year: p_release_year
        };

        console.log('post is: ', post);


        Query = 'INSERT INTO movies SET ?';
        pool.query(Query, post, function (error, results, fields) {
        //pool.query(Query, function (error, results, fields) {
```

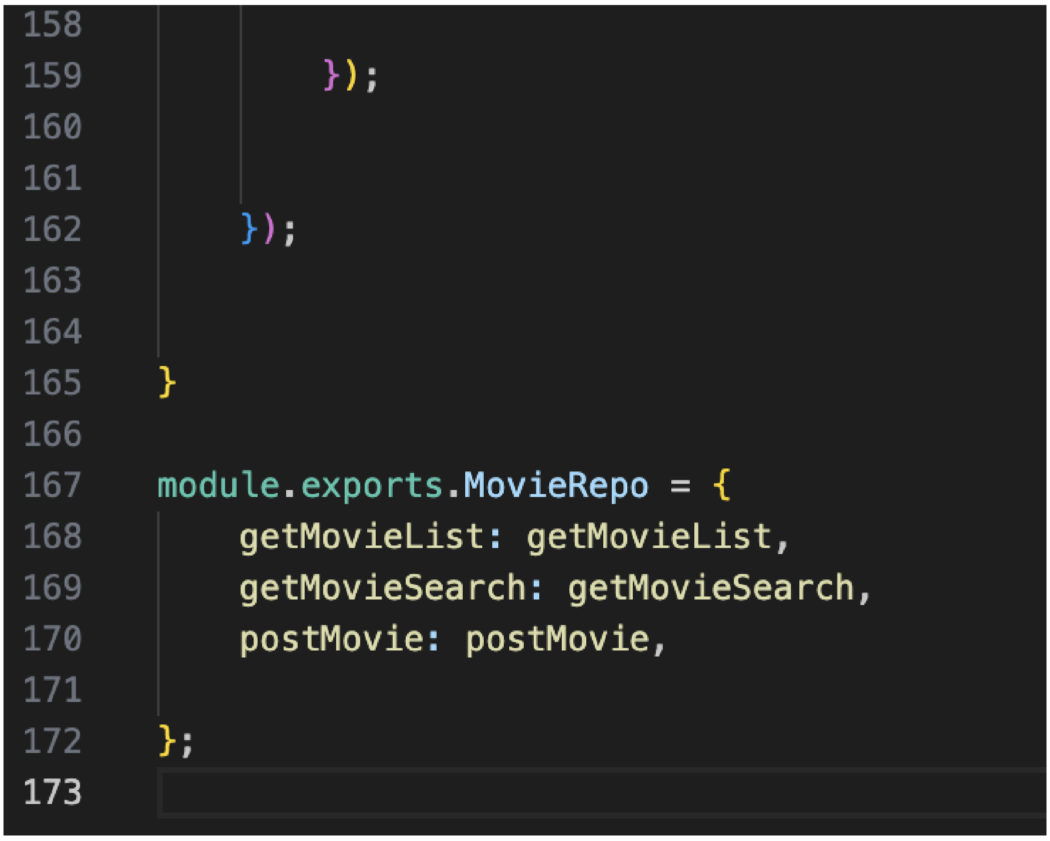
================================



```

        //if (error) throw error;
        if (error) {
            console.log('error_code_msg: ',error.code+':'+error.sqlMessage);
            pool.end();
            return resolve({
                statusCode: 405,
                returnCode: 9,
                messsage: error.code+':'+error.sqlMessage,
            });
        } else {
            console.log('results: ',results);
            if (results.affectedRows > 0) {
                pool.end();
                return resolve({
                    statusCode: 200,
                    returnCode: 1,
                    messsage: 'Movie list was inserted',
                });
            }
        }

```

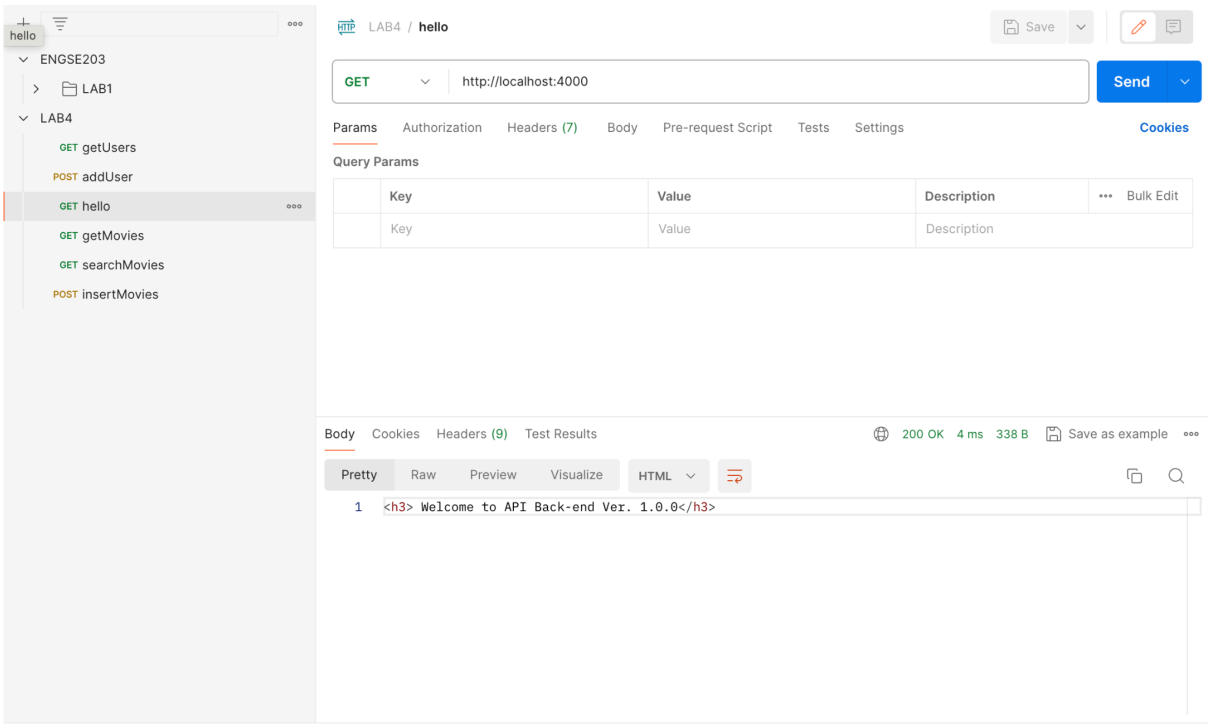
แก้โค้ดตามนี้

==



```

            });


        });


}

module.exports.MovieRepo = {
    getMovieList: getMovieList,
    getMovieSearch: getMovieSearch,
    postMovie: postMovie,

};

```

ทดสอบการทำงานใน Postman

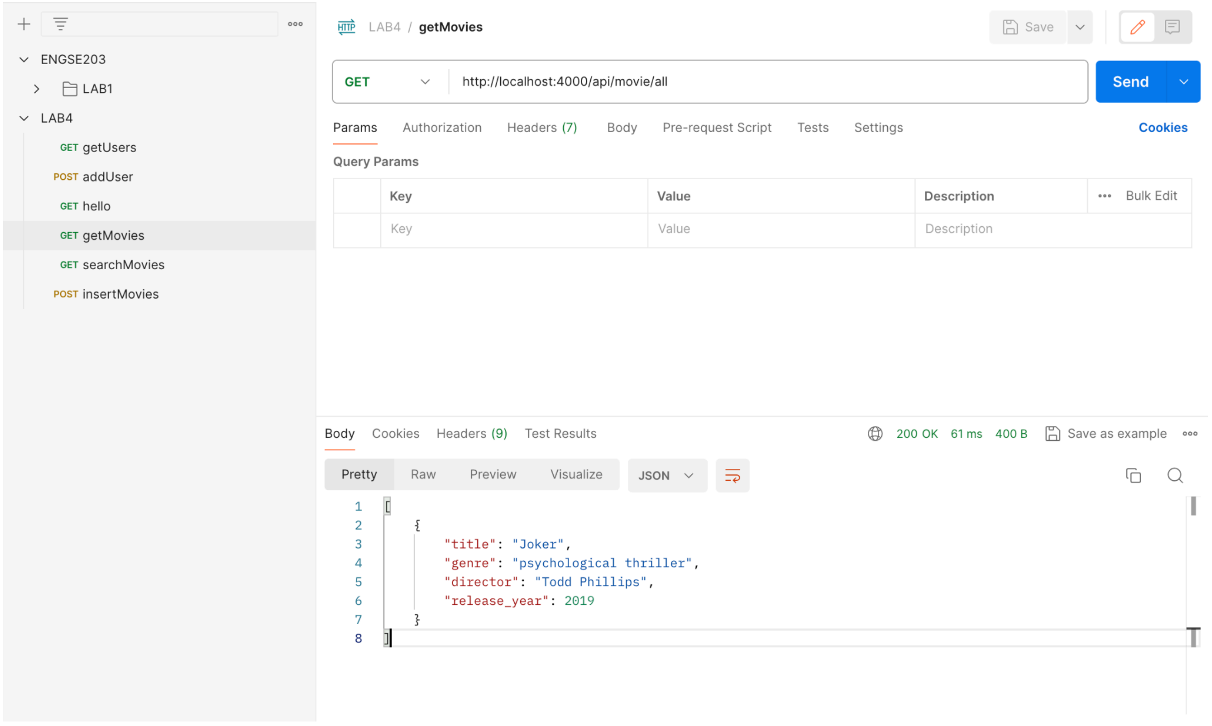


GET http://localhost:4000

200 OK 4 ms 338 B

```
<h3> Welcome to API Back-end Ver. 1.0.0</h3>
```

http://localhost:4000

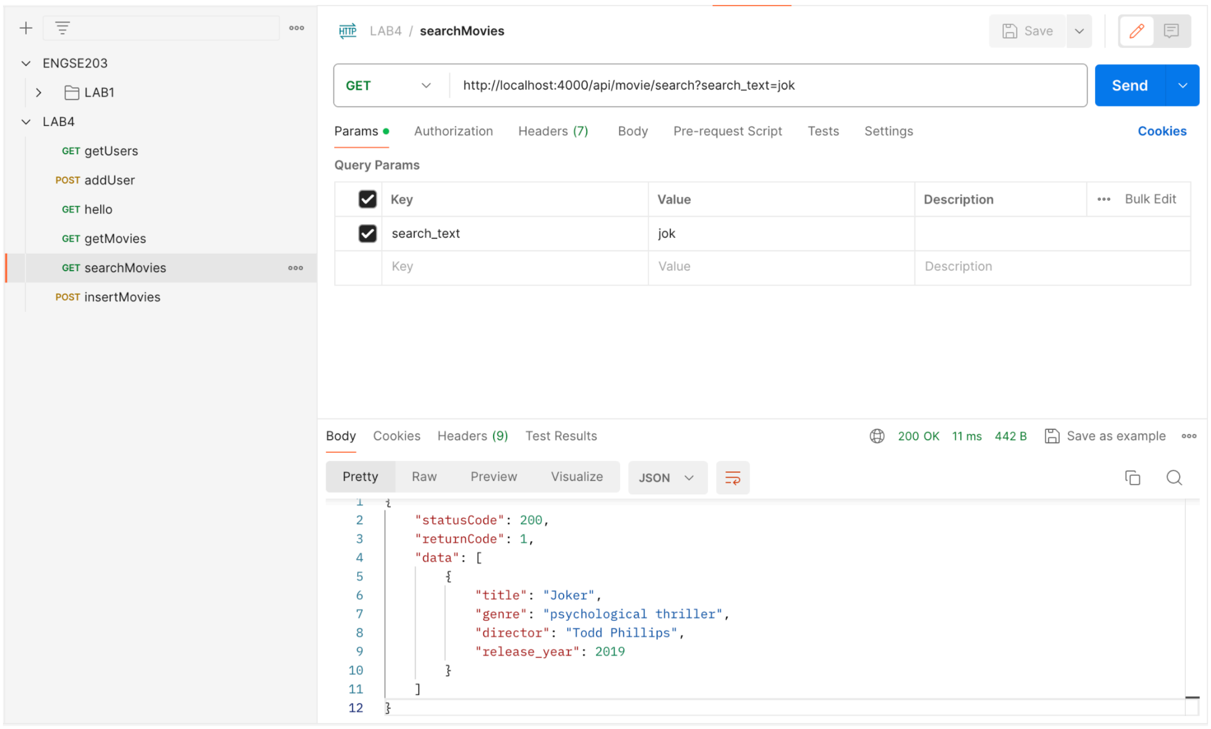


GET http://localhost:4000/api/movie/all

200 OK 61 ms 400 B

```
[
    {
        "title": "Joker",
        "genre": "psychological thriller",
        "director": "Todd Phillips",
        "release_year": 2019
    }
]
```

http://localhost:4000/api/movie/all



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

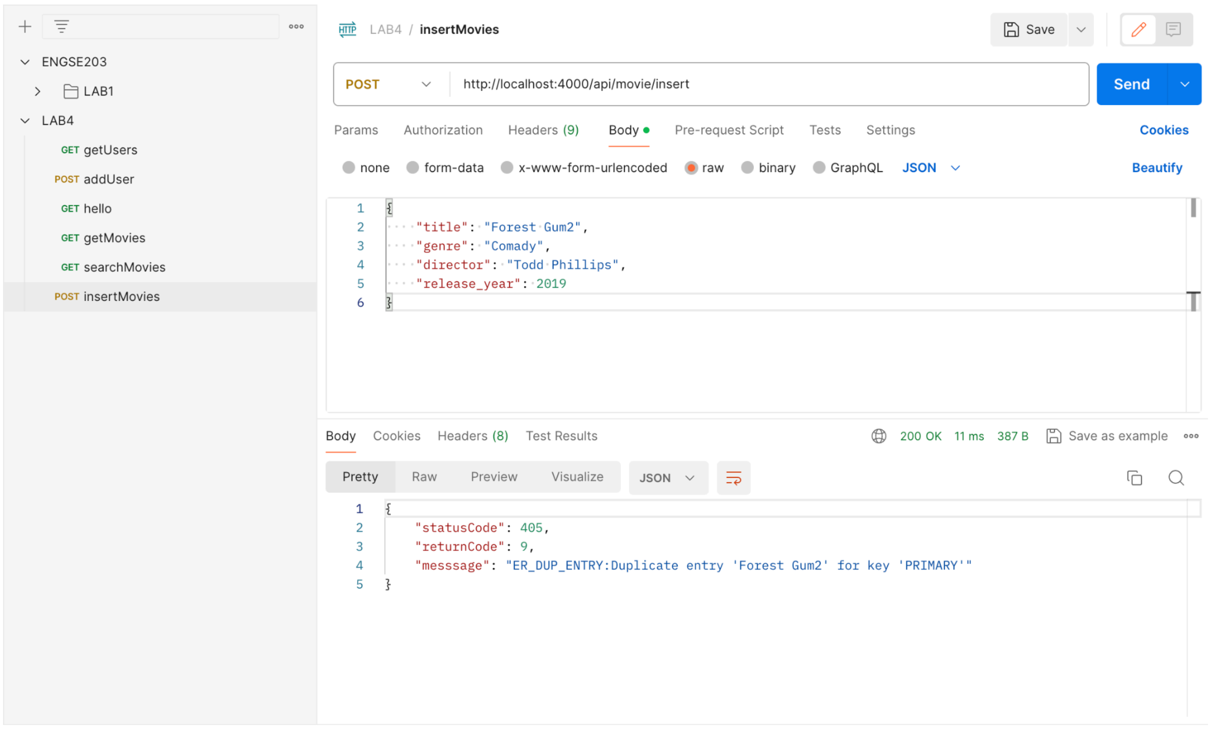
Query Params

Key	Value	Description
search_text	jok	

200 OK 11 ms 442 B

```
{
    "statusCode": 200,
    "returnCode": 1,
    "data": [
        {
            "title": "Joker",
            "genre": "psychological thriller",
            "director": "Todd Phillips",
            "release_year": 2019
        }
    ]
}
```

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



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

```
{
    "title": "Forest Gum2",
    "genre": "Comady",
    "director": "Todd Phillips",
    "release_year": 2019
}
```

200 OK 11 ms 387 B

```
{
    "statusCode": 405,
    "returnCode": 9,
    "messsage": "ER_DUP_ENTRY:Duplicate entry 'Forest Gum2' for key 'PRIMARY'"
}
```

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