



CÔNG NGHỆ PHẦN MỀM MỚI (MTSE431179)

# Hướng dẫn CRUD với Express.js – Sequelize – MySql

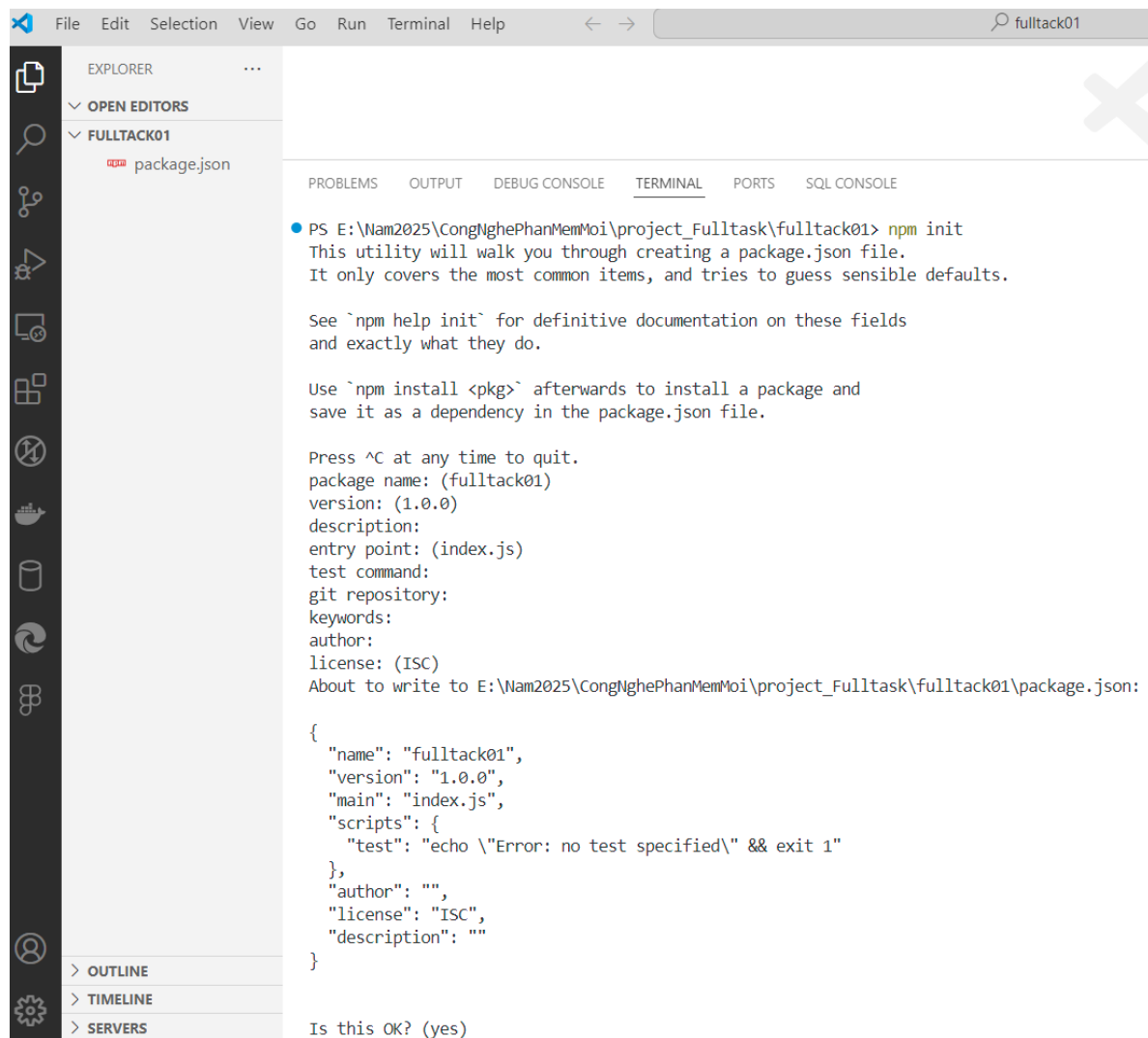
THS. NGUYỄN HỮU TRUNG

- Ths. Nguyễn Hữu Trung
- Khoa Công Nghệ Thông Tin
- Trường Đại học Sư Phạm Kỹ Thuật TP.HCM
- 090.861.7108
- [trungnh@hcmute.edu.vn](mailto:trungnh@hcmute.edu.vn)
- <https://www.youtube.com/@baigiai>



- ExpressJS là gì
- Cấu trúc của ExpressJS
- Cài đặt ExpressJS
- Request & response trong Expresss
- Router trong Express
- File tĩnh trong Express
- Pug viewEngine
- Query Parametter
- Mô hình MVC với ExpressJS

## ❑ Bước 1: Cài đặt node.js (>18.), MySQL, tạo file package.json:



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
  FULLTACK01
    package.json
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
PS E:\Nam2025\CongNghePhanMemMoi\project_Fulltask\fulltack01> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (fulltack01)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to E:\Nam2025\CongNghePhanMemMoi\project_Fulltask\fulltack01\package.json:

{
  "name": "fulltack01",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}

Is this OK? (yes)
```

**npm init**

# Hướng dẫn CRUD với Express.js – Sequelize – MySQL

5

- ❑ Bước 2: Cài đặt Express.js, Sequelize cùng các thư viện cần thiết sau trong package.json như sau:

package.json > {} dependencies

```
1  {
2    "name": "fulltack01",
3    "version": "1.0.0",
4    "main": "server.js",
5    "scripts": {
6      "test": "echo \"Error: no test specified\" && exit 1",
7      "start": "nodemon --exec babel-node src/server.js"
8    },
9    "author": "Nguyen Huu Trung",
10   "license": "ISC",
11   "description": "",
12   "dependencies": {
13     "bcryptjs": "^2.4.3",
14     "body-parser": "^1.20.2",
15     "dotenv": "^16.4.5",
16     "ejs": "^3.1.10",
17     "express": "^4.19.2",
18     "mysql2": "^3.10.2",
19     "sequelize": "^6.37.3"
20   },
21   "devDependencies": {
22     "@babel/core": "^7.24.7",
23     "@babel/node": "^7.24.7",
24     "@babel/preset-env": "^7.24.7",
25     "nodemon": "^3.1.4",
26     "sequelize-cli": "^6.6.2"
27   }
28 }
```


**npm install --save**  
express body-parser  
dotenv ejs sequelize

**npm install --save-dev**  
@babel/core @babel/node  
@babel/preset-env nodemon

# Hướng dẫn CRUD với Express.js – Sequelize - MySql

6

## □ Bước 3: Tạo một số file như sau:

 .sequelizerc > [?] <unknown>

```
1  const path = require('path');
2  module.exports = {
3    'config': path.resolve('./src/config', 'config.json'),
4    'migrations-path': path.resolve('./src', 'migrations'),
5    'models-path': path.resolve('./src', 'models'),
6    'seeders-path': path.resolve('./src', 'seeders')
7  }
```

 .gitignore


```
1  /node_modules
2  /vendor
3  /.idea
4  .idea/
5  .env
6  package-lock.json
```

 .env

```
1  PORT=8088
2  NODE_ENV=development
```

 .env.example

```
1  PORT=
```

 .babelrc > ...

```
1  {
2    "presets": [
3      "@babel/preset-env"
4    ]
5  }
```

- npm install --save mysql2
- npm install --save-dev sequelize-cli
- tự tạo file config.json : chạy lệnh  
**node\_modules/.bin/sequelize init**

# Hướng dẫn CRUD với Express.js – Sequelize - MySql

## □ Bước 4: Tạo thư mục src\config và tạo các file sau:

{ } config.json M ×

src > config > { } config.json > { } development > logging

```
1  {
2    "development": {
3      "username": "root",
4      "password": "1234567@a$",
5      "database": "node_fulltask",
6      "host": "127.0.0.1",
7      "dialect": "mysql",
8      "logging": false
9    },
10   "test": {
11     "username": "root",
12     "password": null,
13     "database": "database_test",
14     "host": "127.0.0.1",
15     "dialect": "mysql"
16   },
17   "production": {
18     "username": "root",
19     "password": null,
20     "database": "database_production",
21     "host": "127.0.0.1",
22     "dialect": "mysql"
23   }
24 }
```

JS configdb.js U ×

src > config > JS configdb.js > ...

```
1  import { Sequelize } from 'sequelize';
2  //const { Sequelize } = require('sequelize');//ES5 module
3
4  // Option 3: Passing parameters separately (other dialects)
5  const sequelize = new Sequelize('node_fulltask', 'root', '1234567@a$', {
6    host: 'localhost',
7    dialect: 'mysql',
8    logging: false
9  });
10 let connectDB = async () => {
11   try {
12     await sequelize.authenticate();
13     console.log('Connection has been established successfully.');
```

▼ config  
{ } config.json  
JS configdb.js  
JS viewEngine.js

- Bước 4: Tạo thư mục src\config và tạo các file sau:

▼ config

- { } config.json
- JS configdb.js
- JS viewEngine.js

JS viewEngine.js A X

src > config > JS viewEngine.js > ...

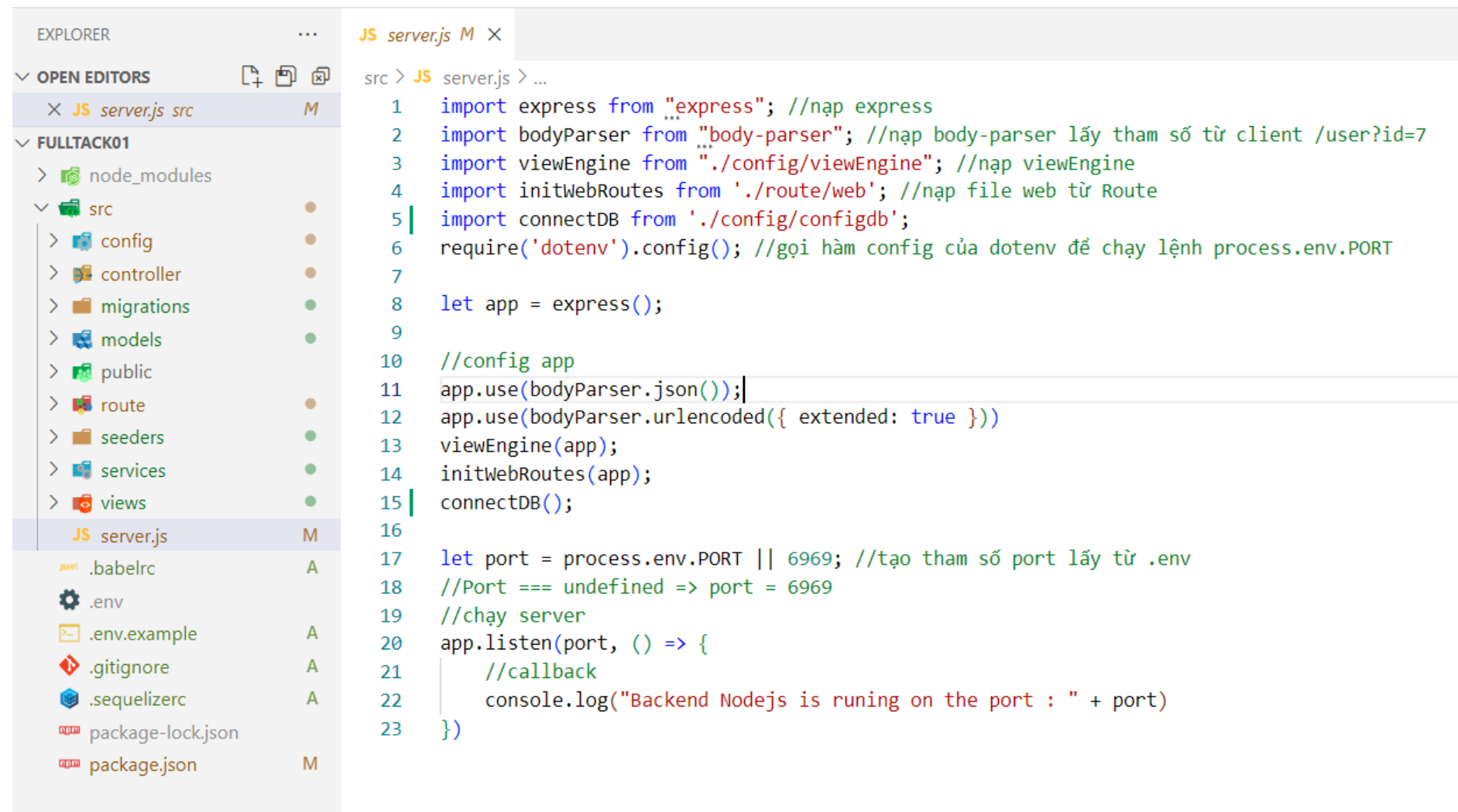
```
1 import express from "express"; // cú pháp khác tương đương: var express = require('express');
2 //javascript theo ES6
3 let configViewEngine = (app) => {
4     app.use(express.static("./src/public")); //Thiết lập thư mục tĩnh chứa images, css,..
5     app.set("view engine", "ejs"); //thiết lập viewEngine
6     app.set("views", "./src/views") //thư mục chứa views
7 }
8 module.exports = configViewEngine; //xuất hàm ra
```



# Hướng dẫn CRUD với Express.js – Sequelize - MySql

9

## □ Bước 5: Cấu hình server.js và tạo cấu trúc dự án như sau:



The image shows a VS Code editor with two panels. The left panel displays the Explorer view with the project structure, and the right panel shows the code for `server.js`.

**Project Structure (Explorer):**

- node\_modules
- src
  - config
  - controller
  - migrations
  - models
  - public
  - route
  - seeders
  - services
  - views
- JS server.js
- .babelrc
- .env
- .env.example
- .gitignore
- .sequelizerc
- package-lock.json
- package.json

**server.js Code:**

```
1 import express from "express"; // nạp express
2 import bodyParser from "body-parser"; // nạp body-parser lấy tham số từ client /user?id=7
3 import viewEngine from "../config/viewEngine"; // nạp viewEngine
4 import initWebRoutes from './route/web'; // nạp file web từ Route
5 import connectDB from './config/configdb';
6 require('dotenv').config(); // gọi hàm config của dotenv để chạy lệnh process.env.PORT
7
8 let app = express();
9
10 //config app
11 app.use(bodyParser.json());
12 app.use(bodyParser.urlencoded({ extended: true }));
13 viewEngine(app);
14 initWebRoutes(app);
15 connectDB();
16
17 let port = process.env.PORT || 6969; // tạo tham số port lấy từ .env
18 //Port === undefined => port = 6969
19 //chạy server
20 app.listen(port, () => {
21   //callback
22   console.log("Backend Nodejs is running on the port : " + port)
23 })
```

## □ Bước 6: Tạo file web.js trong thư mục src\route như sau:

JS web.js M X

src > route > JS web.js > [⌕] initWebRoutes

```
1  import express from "express"; //gọi Express
2  import homeController from "../controller/homeController"; //gọi controller
3
4  let router = express.Router(); //khởi tạo Route
5
6  let initWebRoutes = (app) => {
7      //cách 1:
8      router.get('/', (req,res) => {
9          return res.send('Nguyễn Hữu Trung');
10     });
11     //cách 2: gọi hàm trong controller
12     router.get('/home', homeController.getHomePage); //url cho trang chủ
13     router.get('/about', homeController.getAboutPage); //url cho trang about
14     router.get('/crud', homeController.getCRUD); //url get crud
15     router.post('/post-crud', homeController.postCRUD); //url post crud
16     router.get('/get-crud', homeController.getFindAllCrud); //url lấy findAll
17     router.get('/edit-crud', homeController.getEditCRUD); //url get editcrud
18     router.post('/put-crud', homeController.putCRUD); //url put crud
19     router.get('/delete-crud', homeController.deleteCRUD); //url get delete crud
20
21     return app.use("/", router); //url mặc định
22 }
23
24 module.exports = initWebRoutes;
```

## □ Bước 7: Tạo file user.js trong thư mục src\models như sau:

```
JS user.js M X
src > models > JS user.js > <unknown> > exports > User > associate
1  'use strict';
2  const {
3    Model
4  } = require('sequelize');
5  module.exports = (sequelize, DataTypes) => {
6    class User extends Model {
7      /**
8       * Helper method for defining associations.
9       * This method is not a part of Sequelize lifecycle.
10      * The `models/index` file will call this method automatically.
11      */
12      static associate(models) {
13        // định nghĩa mối quan hệ
14      }
15    }
16    User.init({
17      email: DataTypes.STRING,
18      password: DataTypes.STRING,
19      firstName: DataTypes.STRING,
20      lastName: DataTypes.STRING,
21      address: DataTypes.STRING,
22      phoneNumber: DataTypes.STRING,
23      gender: DataTypes.BOOLEAN,
24      image: DataTypes.STRING,
25      roleId: DataTypes.STRING,
26      positionId: DataTypes.STRING
27    }, {
28      sequelize,
29      modelName: 'User',
30    });
31    return User;
32  };
```

# Hướng dẫn CRUD với Express.js – Sequelize - MySql

12

- ❑ Bước 7: Tạo file migration\_create\_user.js trong thư mục src\migration để tạo bảng trong Database như sau:

JS migration-create-user.js U ×

src > migrations > JS migration-create-user.js > ...

```
1  'use strict';|
2  /** @type {import('sequelize-cli').Migration} */|
3  module.exports = {|
4    async up(queryInterface, Sequelize) {|
5      await queryInterface.createTable('users', {|
6        id: {|
7          allowNull: false,
8          autoIncrement: true,
9          primaryKey: true,
10         type: Sequelize.INTEGER
11       },|
12       email: {|
13         type: Sequelize.STRING
14       },|
15       password: {|
16         type: Sequelize.STRING
17       },|
18       firstName: {|
19         type: Sequelize.STRING
20       },|
21       lastName: {|
22         type: Sequelize.STRING
23       },|
24       address: {|
25         type: Sequelize.STRING
26       },|
27       phoneNumber: {|
28         type: Sequelize.STRING
29       },|
30       gender: {|
31         type: Sequelize.BOOLEAN
32       },|
33       image: {|
34         type: Sequelize.STRING
35       },|
36       roleId: {|
37         type: Sequelize.STRING
```

```
36     roleId: {|
37       type: Sequelize.STRING
38     },|
39     positionId: {|
40       type: Sequelize.STRING
41     },|
42     createdAt: {|
43       allowNull: false,
44       type: Sequelize.DATE
45     },|
46     updatedAt: {|
47       allowNull: false,
48       type: Sequelize.DATE
49     }
50   });|
51 },|
52 async down(queryInterface, Sequelize) {|
53   await queryInterface.dropTable('users');
54 }
55 };
```

npx sequelize-cli db:migrate

# Hướng dẫn CRUD với Express.js – Sequelize - MySql

13

- ❑ Bước 8: Tạo file CRUDService.js trong thư mục src\services như sau:

JS CRUDService.js U X

src > services > JS CRUDService.js > [⌕] createNewUser

```
1  import bcrypt from 'bcryptjs'; //import thư viện bcryptjs
2  import db from '../models/index'; //import database
3  import { where } from 'sequelize';
4  const salt = bcrypt.genSaltSync(10); // thuật toán hash password
5  let createNewUser = async (data) => { //hàm tạo user với tham số data
6      return new Promise(async (resolve, reject) => { //dùng Promise đảm bảo luôn trả kết quả, trong xử lý bất đồng bộ
7          try {
8              let hashPasswordFromBcrypt = await hashUserPassword(data.password)
9              await db.User.create({
10                  email: data.email,
11                  password: hashPasswordFromBcrypt,
12                  firstName: data.firstName,
13                  lastName: data.lastName,
14                  address: data.address,
15                  phoneNumber: data.phoneNumber,
16                  gender: data.gender === '1'? true : false,
17                  roleId: data.roleId
18              })
19              resolve('OK create a new user successfull');
20              // console.log('data from service');
21              //console.log(data) //log dữ liệu từ biến data
22              //console.log(hashPasswordFromBcrypt);
23          } catch (e) {
24              reject(e)
25          }
26      })
27  }
```

- ❑ Bước 8: Tạo file CRUDService.js trong thư mục src\services như sau:

```
28 let hashUserPassword = (password) => {
29   return new Promise(async (resolve, reject) => { //dùng Promise đảm bảo luôn trả kết quả, trong xử lý bất đồng bộ
30     try {
31       let hashPassword = await bcrypt.hashSync("B4c0/\\"", salt);
32       resolve(hashPassword);
33     } catch (e) {
34       reject(e);
35     }
36   })
37 }
38 //lấy tất cả findAll CRUD
39 let getAllUser = () => {
40   return new Promise(async (resolve, reject) => { //dùng Promise đảm bảo luôn trả kết quả, trong xử lý bất đồng bộ
41     try {
42       let users = db.User.findAll({
43         raw: true, //hiển dữ liệu gốc
44       });
45       resolve(users); //hàm trả về kết quả
46     } catch (e) {
47       reject(e)
48     }
49   })
50 }
51 }
```

- ❑ Bước 8: Tạo file CRUDService.js trong thư mục src/services như sau:

```
--  
52 //lấy findOne CRUD  
53 let getUserInfoById = (userId) => {  
54     return new Promise(async (resolve, reject) => { //dùng Promise đảm bảo luôn trả kết quả, trong xử lý bất đồng bộ  
55         try {  
56             let user = await db.User.findOne({  
57                 where: {id:userId}, //query điều kiện cho tham số  
58                 raw:true  
59             });  
60             if(user){  
61                 resolve(user); //hàm trả về kết quả  
62             }else{  
63                 resolve([]); //hàm trả về kết quả rỗng  
64             }  
65         } catch (e) {  
66             reject(e)  
67         }  
68     })  
69 }  
70 }
```

- ❑ Bước 8: Tạo file CRUDService.js trong thư mục src\services như sau:

```
71 //hàm put CRUD
72 let updateUser = (data) =>{
73     return new Promise(async (resolve, reject) => { //dùng Promise đảm bảo luôn trả kết quả, trong xử lý bất đồng bộ
74         try {
75             let user = await db.User.findOne({
76                 where: { id: data.id } //query điều kiện cho tham số
77             });
78             if(user){
79                 user.firstName = data.firstName;
80                 user.lastName = data.lastName;
81                 user.address = data.address;
82                 await user.save();
83                 //lấy danh sách user
84                 let allusers = await db.User.findAll();
85                 resolve(allusers);
86             }else{
87                 resolve(); //hàm trả về kết quả rỗng
88             }
89         } catch (e) {
90             reject(e)
91         }
92     })
93 }
```



- ❑ Bước 8: Tạo file CRUDService.js trong thư mục src/services như sau:

```
//hàm xóa user
let deleteUserById = (userId) => {
  return new Promise(async (resolve, reject) => { //dùng Promise đảm bảo luôn trả kết quả, trong xử lý bất đồng bộ
    try {
      let user = await db.User.findOne({
        where: { id : userId }
      })
      if(user){
        user.destroy();
      }
      resolve(); //là return
    } catch (e) {
      reject(e);
    }
  })
}

module.exports = { //xuất hàm ra bên ngoài
  createNewUser: createNewUser,
  getAllUser: getAllUser,
  getUserInfoById: getUserInfoById,
  updateUser: updateUser,
  deleteUserById: deleteUserById
}
```

- Bước 9: Tạo file homeController.js trong thư mục src\controllers như sau:

JS CRUService.js U

JS homeController.js M X

src &gt; controller &gt; JS homeController.js &gt; [getCRUD]

```
1 | import db from '../models/index'; //import database
2 | import CRUService from '../services/CRUService' //import service
3 | //hàm getHomePage
4 | let getHomePage = async (req, res) => {
5 |     //return res.send('Nguyễn Hữu Trung');
6 |     try {
7 |         let data = await db.User.findAll(); //lấy dữ liệu từ models/index
8 |         console.log('.....');
9 |         console.log(data);
10 |        console.log('.....');
11 |        return res.render('homepage.ejs',{
12 |            data: JSON.stringify(data) //trả dữ liệu data về view
13 |        });
14 |
15 |     } catch (e) {
16 |         console.log(e);
17 |     }
18 | }
19 | //hàm getAbout
20 | let getAboutPage = (req, res) => {
21 |     return res.render('test/about.ejs');
22 | }
```

- Bước 9: Tạo file homeController.js trong thư mục src\controllers như sau:

```
24 //hàm CRUD
25 let getCRUD = (req,res) => {
26   return res.render('crud.ejs');
27 }
28 //hàm findAll CRUD
29 let getFindAllCrud = async (req,res) => {
30   let data = await CRUDService.getAllUser();
31   // console.log('-----');
32   // console.log(data);
33   // console.log('-----');
34   //return res.send('FindAll crud to server');
35   return res.render('users/findAllUser.ejs',{
36     datalist: data
37   }); //gọi view và truyền dữ liệu ra view
38
39 }
40 //hàm post CRUD
41 let postCRUD = async (req, res) => { //dùng async để xử lý bất đồng bộ
42   let message = await CRUDService.createNewUser(req.body); //gọi service
43   //console.log(req.body); //lấy thông tin body của http request
44   console.log(message);
45   return res.send('Post crud to server');
46
47 }
```

- Bước 9: Tạo file homeController.js trong thư mục src\controllers như sau:

```
48 //hàm lấy dữ liệu để edit
49 let getEditCRUD = async (req,res) => {
50     let userId = req.query.id;
51     if(userId){ //check id
52         let userData = await CRUDService.getUserInfoById(userId);
53
54         // console.log('-----');
55         // console.log(userData);
56         // console.log('-----');
57         return res.render('users/editUser.ejs',{
58             data: userData
59         });
60     }else{
61         return res.send('không lấy được id');
62     }
63
64     // console.log(req.query.id);
65
66 }
67 let putCRUD = async (req, res) =>{
68     let data = req.body;
69     let data1 = await CRUDService.updateUser(data);//update rồi hiển thị lại danh sách user
70     //let data1 = await CRUDService.getAllUser();//hiển thị danh sách user
71     return res.render('users/findAllUser.ejs',{
72         datalist: data1
73     });
74     // return res.send('update thành công');
75 }
```

- Bước 9: Tạo file homeController.js trong thư mục src\controllers như sau:

```
76 let deleteCRUD = async (req, res) => {
77   let id = req.query.id; //vì trên view ?id=1
78   if(id){
79     await CRUDService.deleteUserById(id);
80     return res.send('Deleted!!!!!!!!!!!!');
81   }else{
82     return res.send('Not find user')
83   }
84 }
85
86 // object: {
87 //   key: '',
88 //   value: ''
89 // }
90 //export ra object
91 module.exports = {
92   getHomePage: getHomePage,
93   getAboutPage: getAboutPage,
94   getCRUD: getCRUD,
95   postCRUD: postCRUD,
96   getFindAllCrud: getFindAllCrud,
97   getEditCRUD: getEditCRUD,
98   putCRUD: putCRUD,
99   deleteCRUD: deleteCRUD
100 }
```

- ❑ Bước 10: Tạo file crud.ejs trong thư mục src\views như sau:

```
<form action="/post-crud" method="post">

  <div class="form-row">
    <div class="form-group col-md-6">
      <label for="inputEmail4">Email</label>
      <input type="email" class="form-control" id="email" name="email">
    </div>
    <div class="form-group col-md-6">
      <label for="inputPassword4">Password</label>
      <input type="password" class="form-control" id="password" name="password">
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-md-6">
      <label for="inputEmail4">First Name</label>
      <input type="text" class="form-control" id="firstName" name="firstName">
    </div>
    <div class="form-group col-md-6">
      <label for="inputPassword4">Last Name</label>
      <input type="text" class="form-control" id="lastName" name="lastName">
    </div>
  </div>
</div>
```

- Bước 10: Tạo file crud.ejs trong thư mục src\views như sau:

```
<div class="form-row">
  <div class="form-group col-md-4">
    <label for="inputCity">Phone Number</label>
    <input type="text" class="form-control" id="phoneNumber" name="phoneNumber">
  </div>
  <div class="form-group col-md-4">
    <label for="inputState">Gender</label>
    <select id="gender" name="gender" class="form-control">
      <option selected value="0">Male</option>
      <option value="1">Female</option>
    </select>
  </div>
  <div class="form-group col-md-4">
    <label for="role">Role</label>
    <select id="roleId" name="roleId" class="form-control">
      <option selected value="1">Admin</option>
      <option value="2">Doctor</option>
      <option value="3">Patient</option>
    </select>
  </div>
</div>

<button type="submit" class="btn btn-primary">Sign in</button>
</form>
```

- Bước 10: Tạo file findAllUser.ejs trong thư mục src\views\users như sau:

```
<body>
  <table style="width:100%">
    <tr>
      <th>Email</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Address</th>
      <th>Action</th>
    </tr>
    <% for(let i=0; i < datalist.length; i++) { %>
      <tr>
        <td><%= datalist[i].email %></td>
        <td><%= datalist[i].firstName %></td>
        <td><%= datalist[i].lastName %></td>
        <td><%= datalist[i].address %></td>
        <td><a href="/edit-crud?id=<%= datalist[i].id %>">Edit</a>||<a
          href="/delete-crud?id=<%= datalist[i].id %>">Delete</a></td>
      </tr>
    <% } %>
  </table>
</body>
```



- ❑ Bước 10: Tạo file updateUser.ejs trong thư mục src\views\users như sau:

```
<div class="container">
  <div class="row">
    <form action="/put-crud" method="post">
      <input type="text" class="form-control" id="id" name="id" value="<%=data.id%>">

      <div class="form-row">
        <div class="form-group col-md-6">
          <label for="inputEmail4">First Name</label>
          <input type="text" class="form-control" id="firstName" name="firstName"
            value="<%=data.firstName%>">
        </div>
        <div class="form-group col-md-6">
          <label for="inputPassword4">Last Name</label>
          <input type="text" class="form-control" id="lastName" name="lastName"
            value="<%=data.lastName%>">
        </div>
      </div>
      <div class="form-group">
        <label for="inputAddress">Address</label>
        <input type="text" class="form-control" id="address" placeholder="1234 Main St" name="address"
          value="<%=data.address%>">
      </div>

      <button type="submit" class="btn btn-primary">Update</button>
    </form>
  </div>
</div>
```

# Hướng dẫn CRUD với Express.js – Sequelize – MySQL

26

## ❑ Bước 11: Chạy dự án: npm start

```
PS E:\Nam2025\CongNghePhanMemMoi\project_Fulltask\fulltack01> npm start
```

```
> fulltack01@1.0.0 start
> nodemon --exec babel-node src/server.js
```

```
[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `babel-node src/server.js`
Backend Nodejs is runing on the port : 8088
Executing (default): SELECT 1+1 AS result
Connection has been established successfully.
```

localhost:8088/edit-crud?id=1

Android RecyclerVie... (1) #N6 Phân Tích C... NodeJS Bài 4: Query... Tim

1

First Name

Trung 11

Last Name

Nguyễn Hữu

Address

Hồ Chí Minh s

Update

localhost:8088/get-crud

Android RecyclerVie... (1) #N6 Phân Tích C... NodeJS Bài 4: Query... Tim hiểu về NodeJS

Email	First Name	Last Name	Address
trungnhspkt@gmail.com	Trung	Nguyễn Hữu	Hồ Chí Minh
trungnh@hcmute.edu.vn	Trung	Nguyễn Hữu	59/17/4/2 Đường số 8, Tổ 16, Kp5

- Thực hiện các bước tương tự như bài tập hướng dẫn nhưng áp dụng cho database MongoDB.