**Các lệnh cơ bản:**

1. Pm2 kill
2. Pm2 stop on (dừng tất cả) hoặc pm2 stop .pm2.local.js
3. npm i express (tạo node moduless)
4. npm i --save (tạo node moduless)
5. npm i pm2
6. pm2 start . tab
7. pm2 log
8. npm start (chạy dự án mẫu do **ko có file local**)
9. npm install -g npm-check-updates
10. ncu -u (upgrade)
11. npm up ‘tên’(update)
12. npm un ‘tên’ (unstall)

**Hướng làm chức năng đăng nhập:**

1. Tạo bảng
2. Liên kết database với pm2
3. Tạo đường dẫn trang đầu tiên vào (file app.js)
4. Tạo các route để điều hướng UI
5. Post dữ liệu lên database

* file ejs (cần có form action= đường dẫn, method post); **name cho phone, pass.**
* route (đường dẫn, service.hàm post);
* service (kết hợp các biến trong ejs với database); tạo hàm create data()

Đăng nhập:

* file ejs (cần có form action= đường dẫn, method post);
* Route(đường dẫn trang chủ,service.hàm post); **có dkien**
* Service: hàm tìm user(kiểm tra phone, pass) trong data của user
* Middlewares (điều kiện trong route **permission.hàm (phải đăng nhập)**)

Kiểm tra id; kiểm tra phone; pass

**Sequelize:** gần giống mySQL mà rõ hơn.

**Route:** đường dẫn (app.js), route (hàm get,post), service(đường dẫn render UI). **VD: action= “/…../….”**

**Upload thông tin:**

* file ejs (cần có form action= đường dẫn, method post); **name cho phone, pass.**
* route (đường dẫn, service.hàm post);
* service (kết hợp các biến trong ejs với database); tạo hàm create data()

**Đăng nhập:**

* file ejs (cần có form action= đường dẫn, method post);
* Route(đường dẫn trang chủ,service.hàm post); **có dkien**
* Service: hàm tìm user(kiểm tra phone, pass)
* Middlewares (điều kiện trong route **permission.hàm (phải đăng nhập)**)

Kiểm tra id; kiểm tra phone; pass

**LogOut:**

Hàm GET Xóa id session hiện tại.

const getLogout = async(req, res) => {

    try {

        req.session = null;

        res.redirect('/auth/signin');

    } catch (ex) {

        console.log(ex);

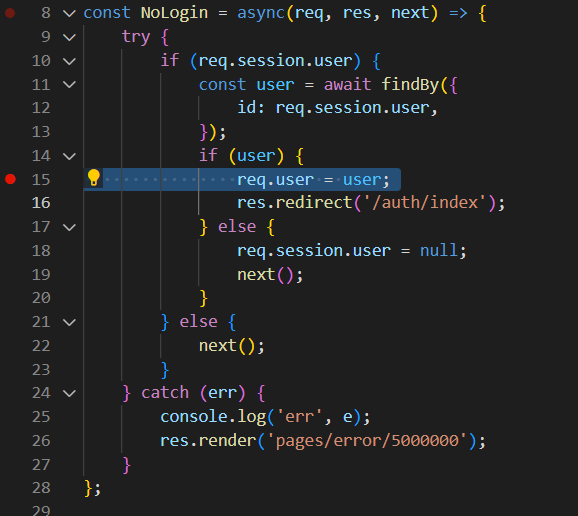
        res.render('pages/error/500');

    }

};

Check user đúng: gán id của user vào bảng database.

Lưu id của database



Định tuyến đường đi (URI) phản hồi yêu cầu của khách hàng,

Xác định định tuyến bằng các phương thức

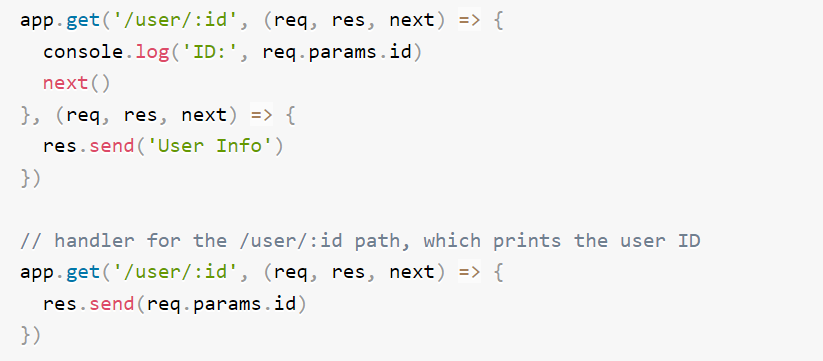
app.get(), app.post để xử lý các yêu cầu gửi nhận của người dùng.

App.all() các phương thức trên???? middlewares: bác bảo vệ.( request object (req), the response object (res))

it must call next() to pass control to the next middleware function

Các loại middleware:

1. ***Application-level middleware:*** đưa yêu cầu để xử lý



Req,res,next != next ()????

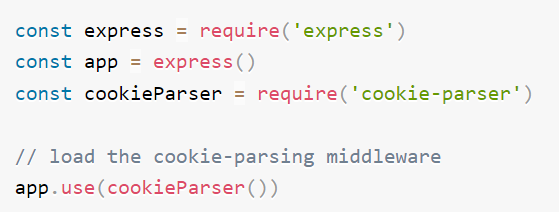
1. ***Router-level middleware:*** để sử dụng cần gọi thư viện const router = express.Router()

Để sử dụng **router.use() and router.METHOD()** functions.

1. Error-handling middleware:

Có 4 vấn đề app.use((err, req, res, next)

1. Built-in middleware
2. Third-party middleware; phân tích cookie



**Redis** :

Check tên miền, session…. trong có thể sử dụng lại

Constant.js chỉnh **role, status** trong database.

Service.js xử lý logic.

Chức năng nhỏ của project

1. Bảng database
2. 3 trang html
3. Kết nối database???
4. Login đúng / sai??