

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG KHOA AN TOÀN THÔNG TIN**

**Môn học: IOT và ứng dụng**

**Báo Cáo API**

**Họ và tên:** Trần Trọng Mạnh **Mã sinh viên:** B21DCAT127 **Nhóm môn học:** 05

**Giảng viên:** Nguyễn Quốc Uy

Hà Nội, 4/2024

MỤC LỤC

[1. Cấu trúc project 3](#_Toc180498041)

[2. API 3](#_Toc180498042)

[a. Socket.io, gửi data từ hardware lên web theo thời gian thực 4](#_Toc180498043)

[b. API để xử lý bật tắt thiết bị, cập nhật trạng thái thiết bị từ hardware lên web 5](#_Toc180498044)

[ Update status to swich 5](#_Toc180498045)

[ Xử lý bật tắt thiết bị theo topic/{device} 7](#_Toc180498046)

[c. Lưu và lấy dữ liệu cảm biến vào MongoDB 9](#_Toc180498047)

[ Lưu dữ liệu cảm biến 9](#_Toc180498048)

[ API: api/sensor-data/filter 10](#_Toc180498049)

[d. Lưu và lấy data trạng thái của thiết bị vào MongoDB 12](#_Toc180498050)

[ Thực hiện lưu data vào database mỗi khi bật tắt thiết bị 12](#_Toc180498051)

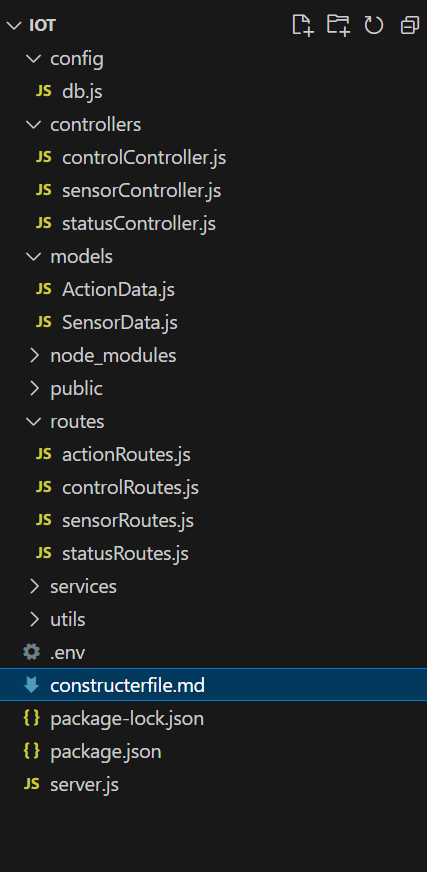
1. Cấu trúc project

Hardware: ESP8266, Arduino IDE 2.3.3

Backend: NoteJS

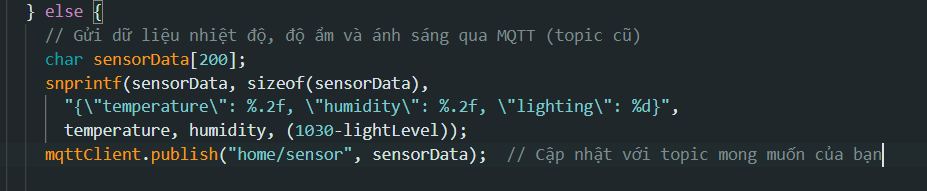
Database: MongoDB

FontEnd: Html, css, bootstrap



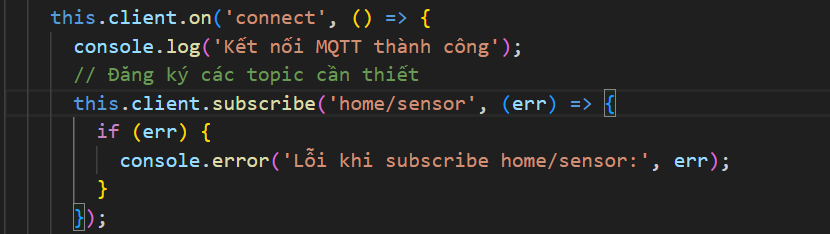
1. API
2. Socket.io, gửi data từ hardware lên web theo thời gian thực

Lấy data từ mqtt lên backend thông qua topic home/sensor



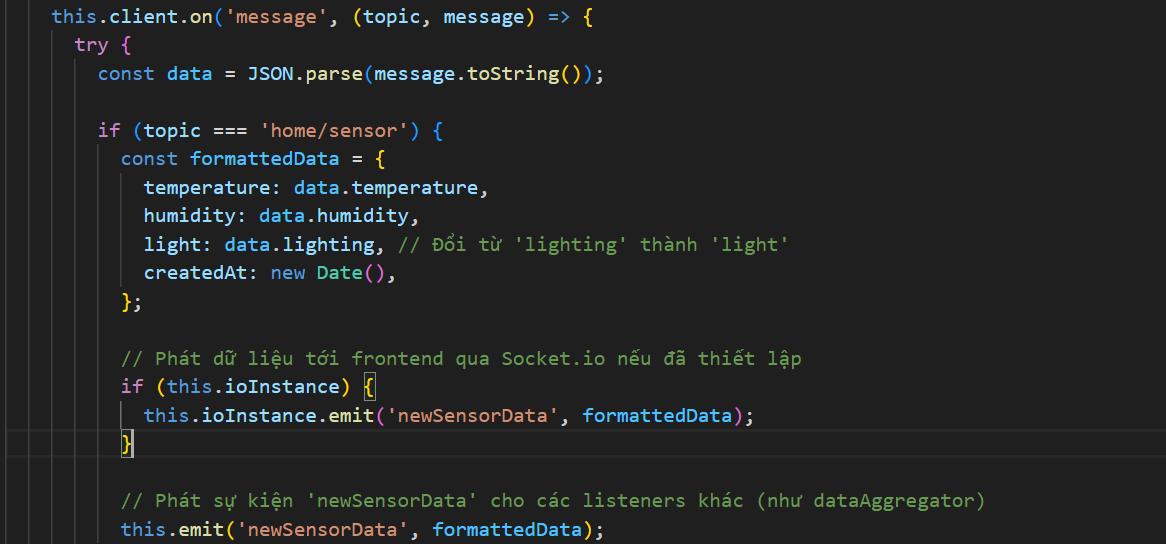
Mqtt.service.js

Thiết lập kết nối client

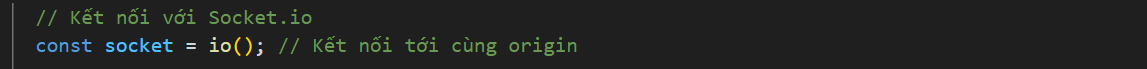


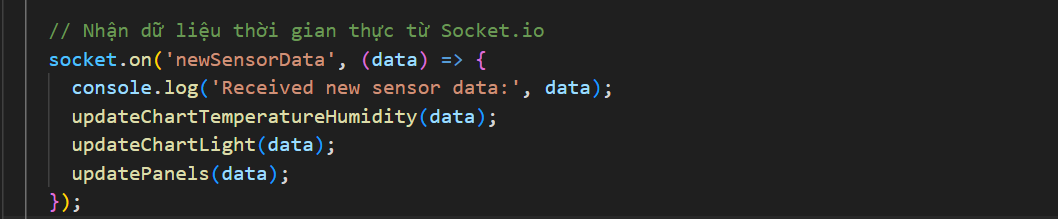


Thiết lập mqtt gửi data bằng Socket.io để gửi data theo thời gian thực từ hardware lên backend. Sau đó dụng hàm: this.emit(‘newSensorData’,formattedData);



Phía frontend: Kết nối với socket.io, nhận data từ backend

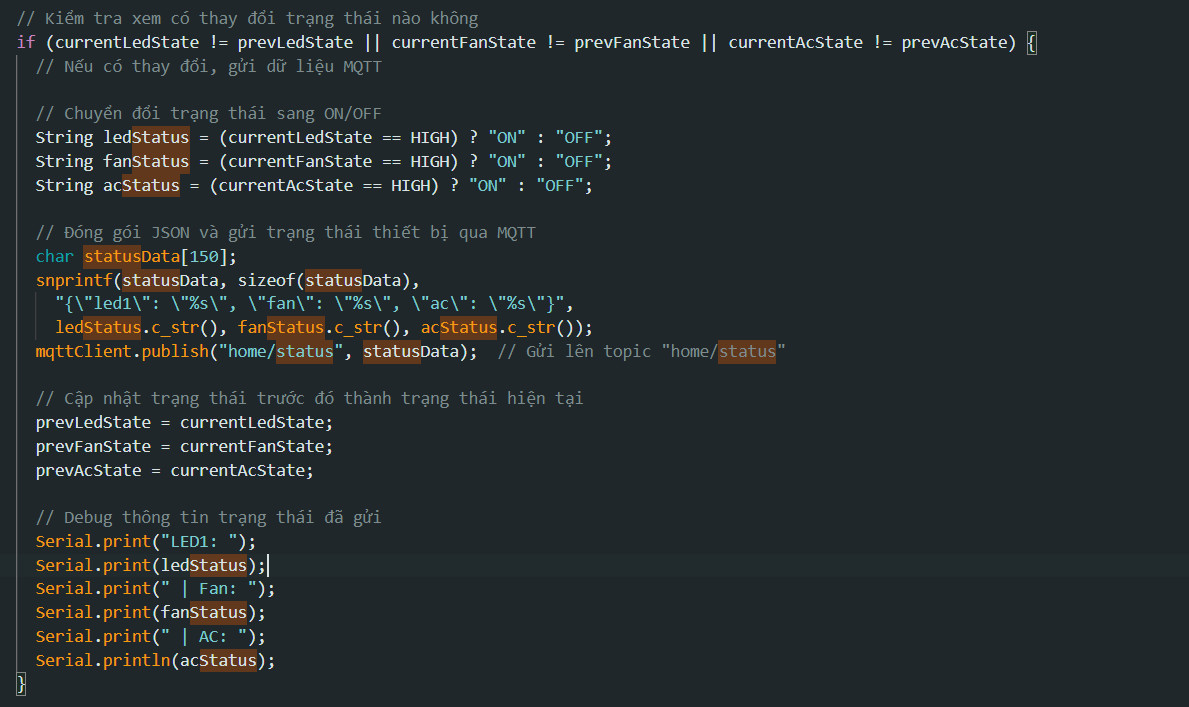


  
sau đó dùng các hàm để cập nhật data cho nhiệt độ, độ ẩm, ảnh sáng và khối thay đổi màu theo giá trị.

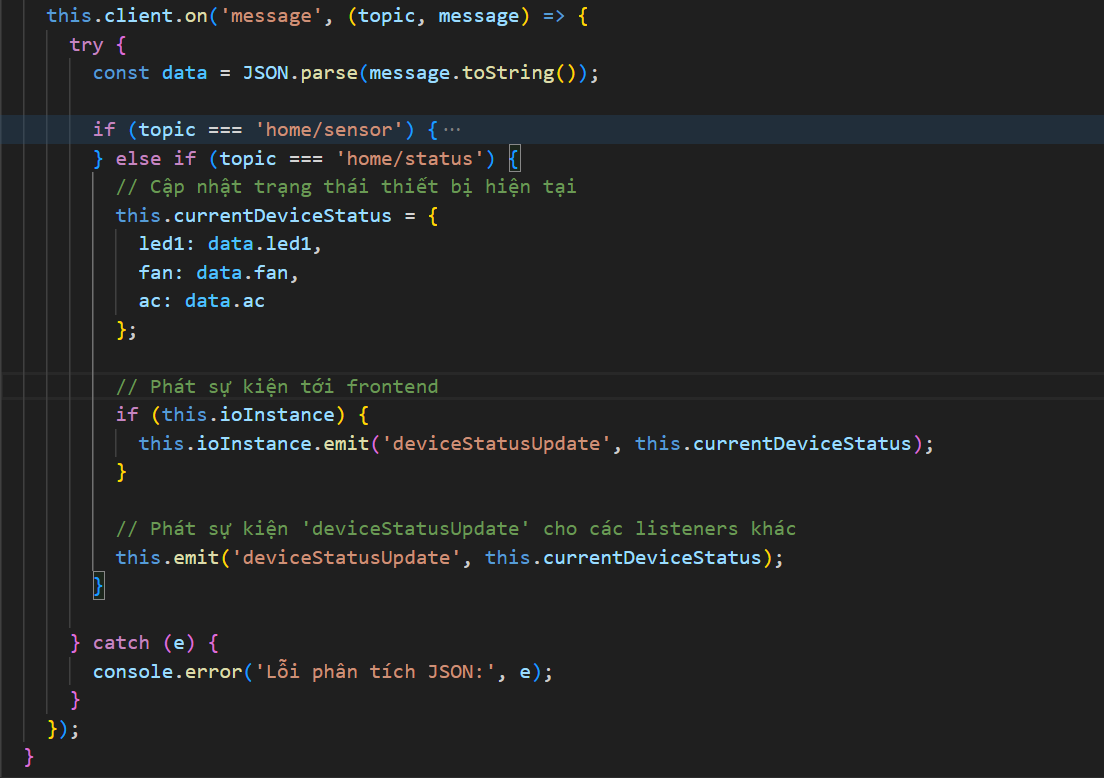
1. API để xử lý bật tắt thiết bị, cập nhật trạng thái thiết bị từ hardware lên web

* Update status to swich

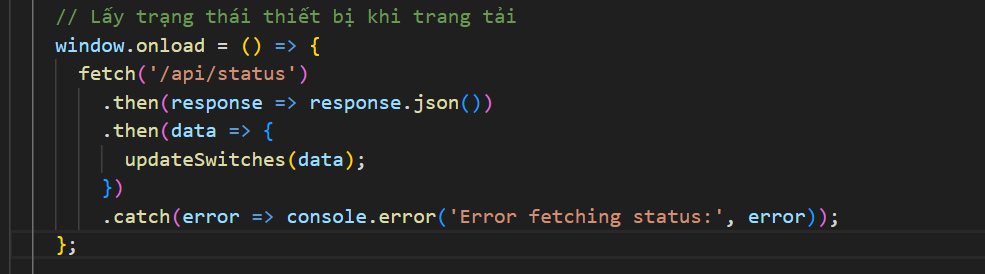
Khi load trang thì sẽ gọi api để cập nhật status cho thiết bị. Và sau đó nếu như có sự thay đổi gì từ hardware thì nó sẽ gửi lên client qua topic/status

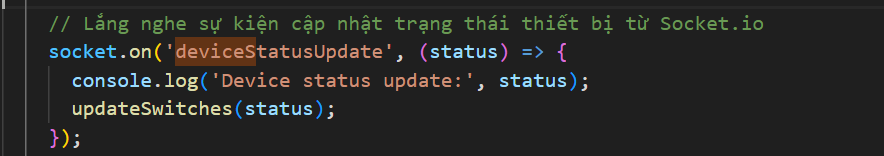


Nếu có sự kiện sẽ được gửi theo topic/status

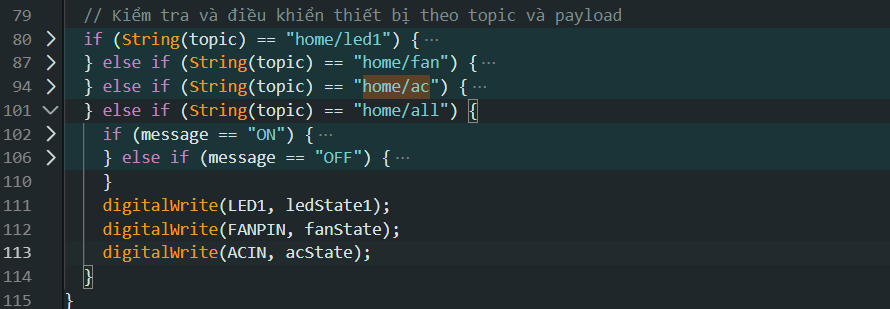


Sau đó được socket lắng nghe và cập nhật trên frontend

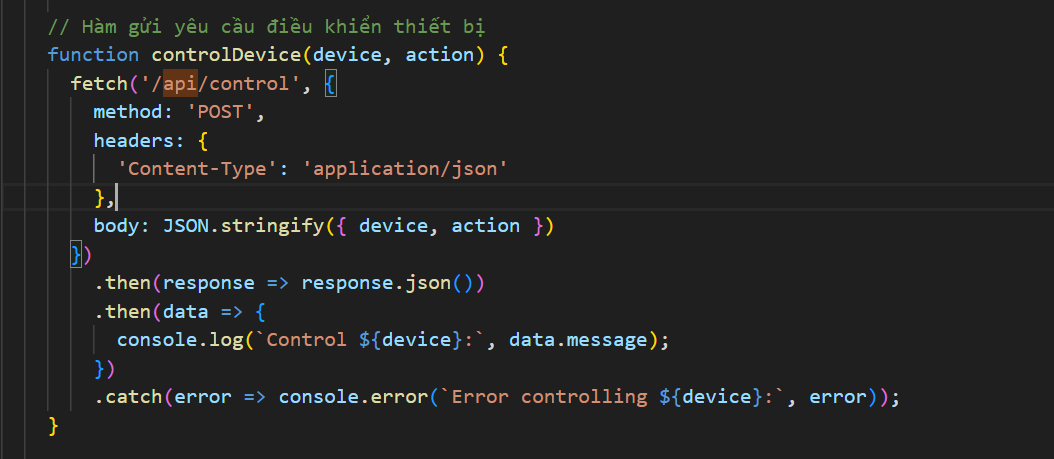




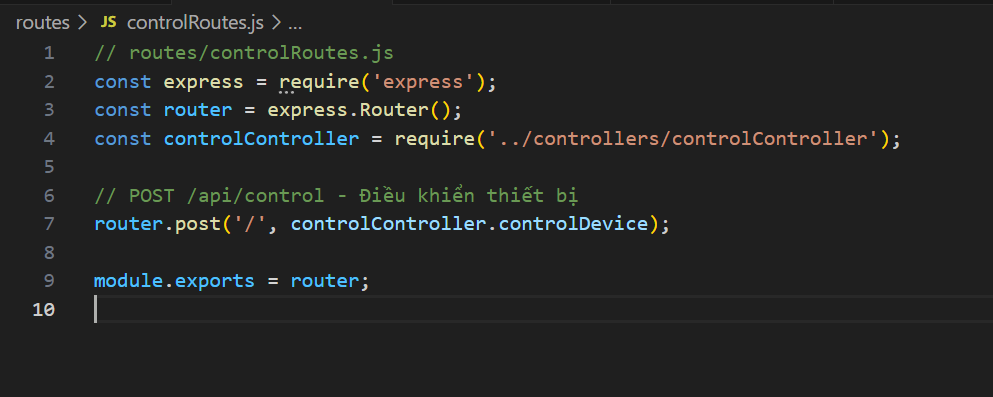
* Xử lý bật tắt thiết bị theo topic/{device}

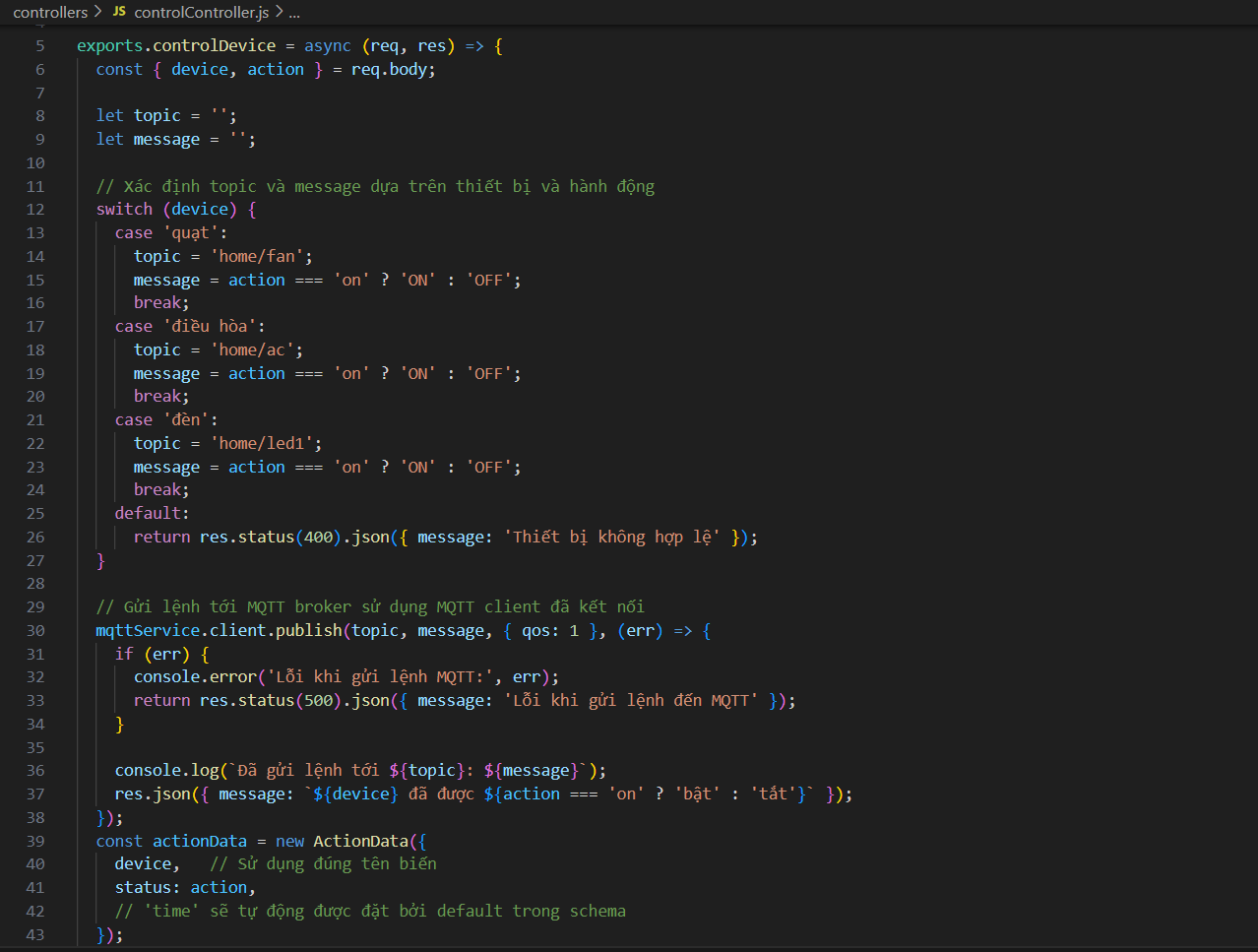


Xử lý khi frontend post API api/control



Phía backend





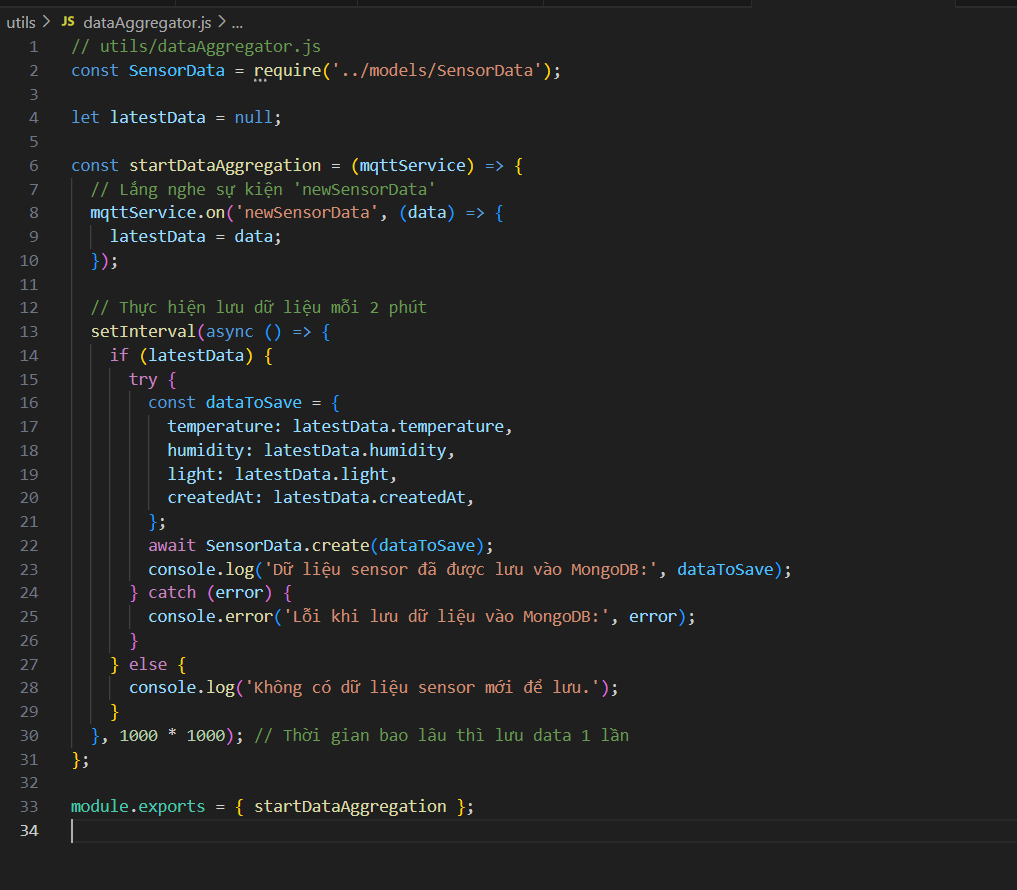
Gửi lệnh tới mqtt thông qua mqttService.client.publish

1. Lưu và lấy dữ liệu cảm biến vào MongoDB

* Lưu dữ liệu cảm biến

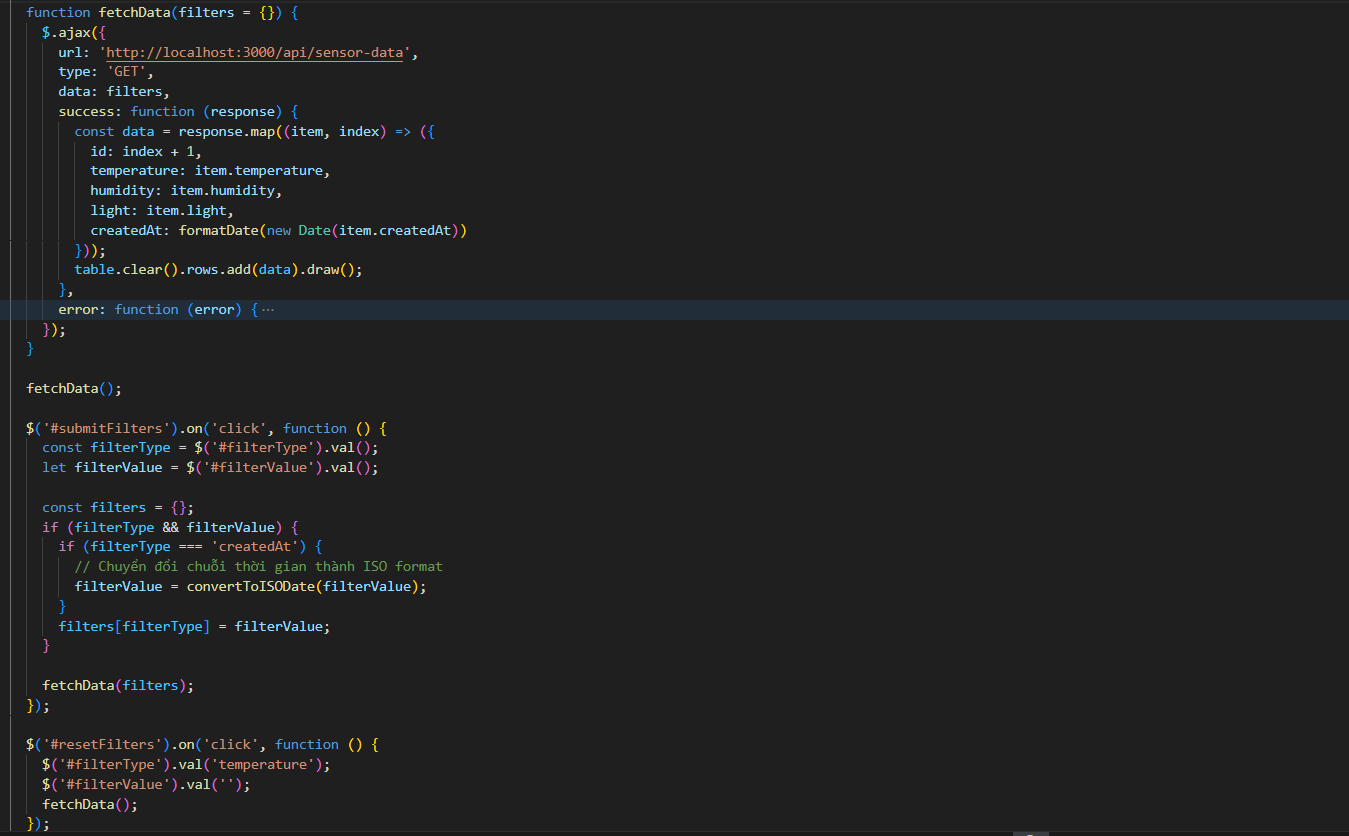
Lắng nghe sự kiện newSensorData được khai báo ở trên để gửi data lên web thông qua socket.io

Xử lý data sau đó lưu vào MongoDB

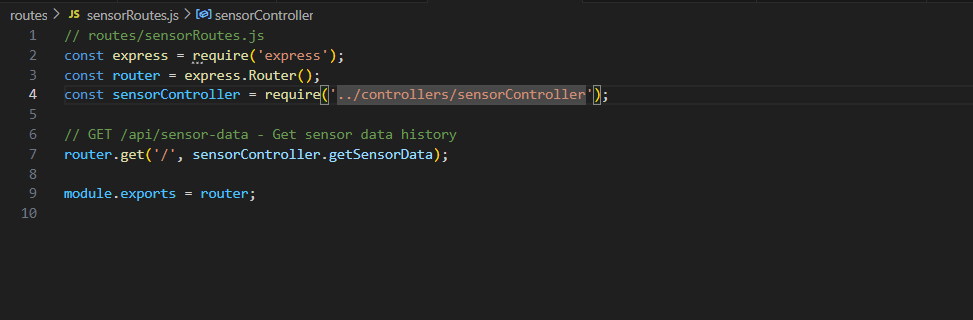


Sau đó lấy data từ database build lên trang datasensor.html

* API: api/sensor-data/filter

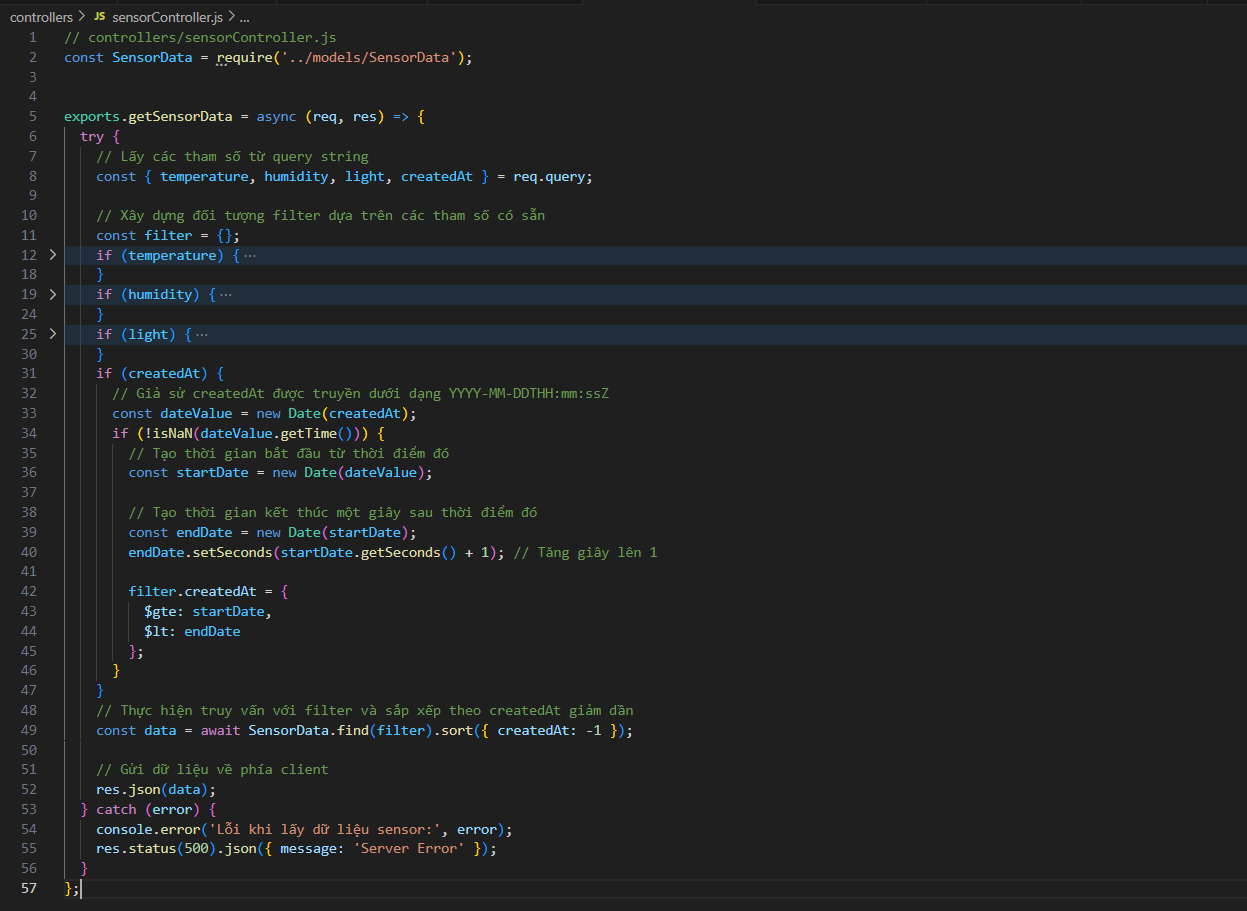


Xử lý backend để truy vấn dữ liệu vào database



Xử lý dữ liệu thời gian tìm kiếm chính xác đến giây

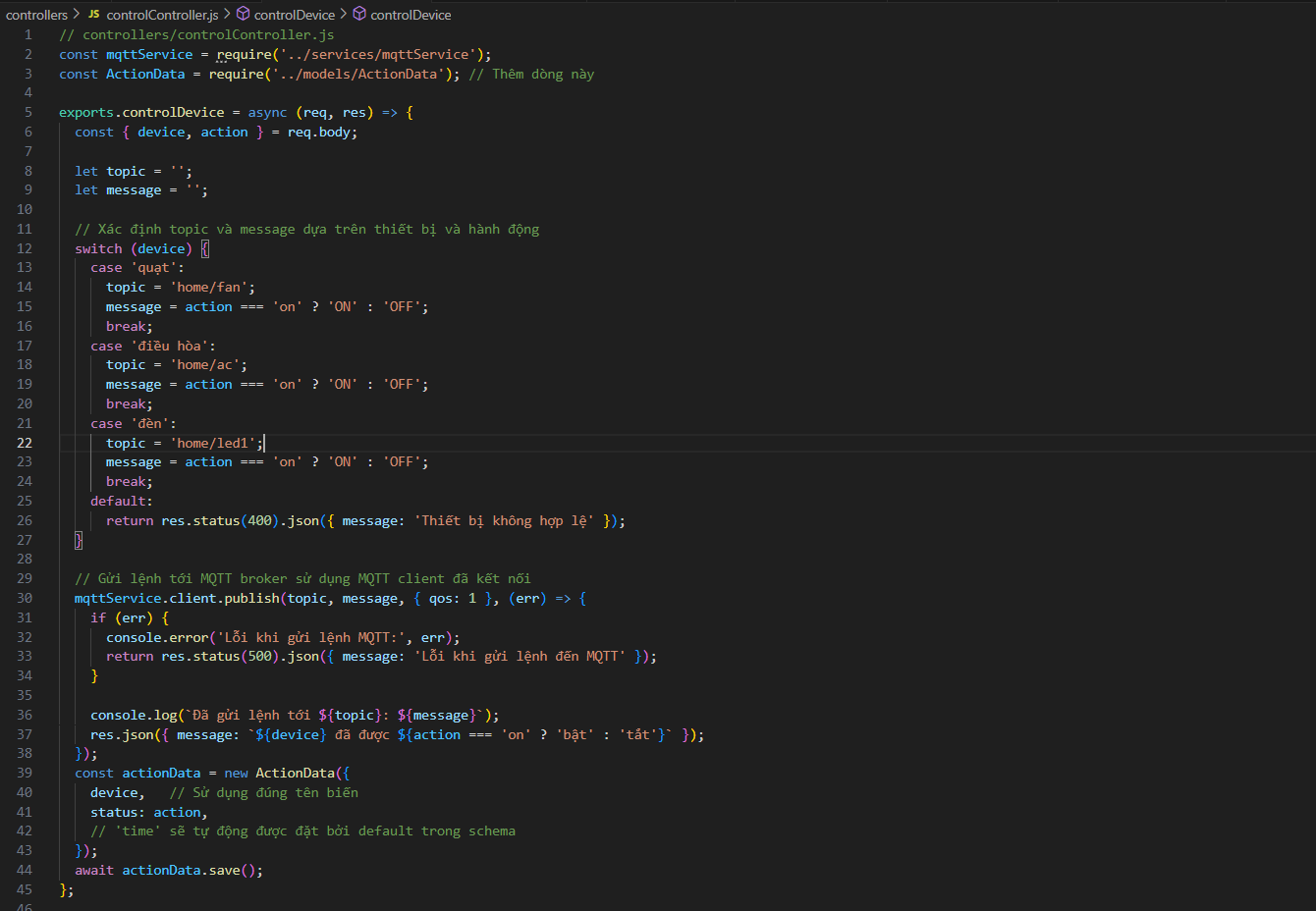
Sau đó thực hiện truy vấn



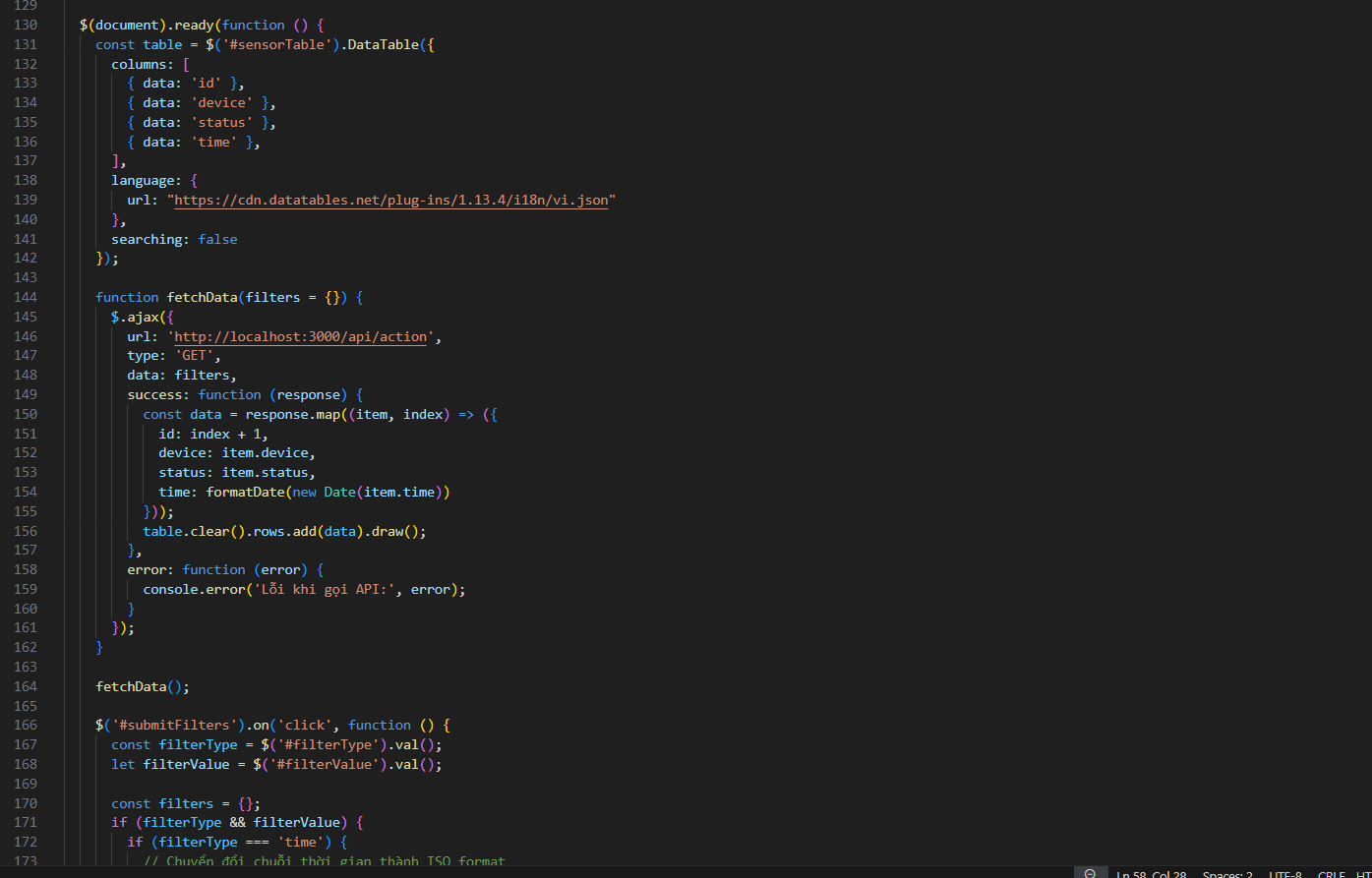
1. Lưu và lấy data trạng thái của thiết bị vào MongoDB

* Thực hiện lưu data vào database mỗi khi bật tắt thiết bị

Lấy post API api/control. Mỗi khi có 1 post API thì thực hiện lưu vào database

  
thực hiện lưu ở dòng 44

* Lấy data history-action-device  
  frontend lấy api/action



Phía backend

