**Tài liệu hướng dẫn sử dụng chương trình Quản lý các API logger**

1. **Giới thiệu**

Chương trình được phát triển bằng NestJS, cung cấp các chức năng để lưu và tìm kiếm các API request và response.

**Yêu cầu Hệ thống**

* Node.js phiên bản 14 trở lên
* NestJS
* MongooseDB
* TypeORM và kết nối cơ sở dữ liệu (PostgreSQL, MySQL, v.v.)
* Cơ sở dữ liệu chứa bảng logs với các cột: \_id, level (phân loại logger ‘info’ hay ‘error’), method, originalUrl, statusCode, statusMessage, duration (thời gian gửi request và nhận response, đơn vị ms), createdAt (thời điểm logger được tạo).

**Cài đặt**

* **Clone mã nguồn**: Tải mã nguồn từ kho chứa hoặc từ file nén.
* **Cài đặt các gói cần thiết**: npm install
* **Cấu hình cơ sở dữ liệu**:
  + Mở file app.module.ts, điền thông tin kết nối cơ sở dữ liệu MongooseDB và MySQL.
  + Sử dụng TypeORM để tự động tạo bảng hoặc chạy thủ công lệnh SQL để tạo bảng actors.
* **Chạy chương trình**: npm run start

1. **Các chức năng chính**
2. **Lưu logger vào file:**

* Chương trình thông qua interceptor Log để gọi chức năng này mỗi khi một API request có tồn tại của một đối tượng được gửi (ví dụ là đối tượng actors), khi nhận API response thì thực hiện ghi ra file một Log chứa thông tin API đó.
* **Chức năng file size rotation và daily rotation:** chương trình còn áp dụng DailyRotateFile của thư viện winston nhằm tách file lưu trữ các Log khi nó vượt quá 2KB (đảm bảo chỉ có khoảng 9 Log mỗi file) và các file chứa Log này sẽ tự xóa khi chạy chương trình sau 1 ngày.
* **Ví dụ cấu trúc một Log**:

{

"\_id": "672706c1edb9e0a132dff72d",

"createdAt": "2024-1103T05:14:41.865Z",

"duration": 5,

"level": "error",

"method": "POST",

"originalUrl": "/actors",

"statusCode": 400,

"statusMessage": [

"first\_name must be longer than or equal to 1 characters",

"last\_name must be longer than or equal to 1 characters"

]

}

1. **Lưu Log vào cơ sở dữ liệu:**

* Log sẽ được lưu vào MongooseDB với cấu hình như trong file app.module.ts
* **Chức năng weekly rotation:** các Log trong cơ sở dữ liệu được lưu quá 7 ngày sẽ bị xóa khi chạy chương trình.

1. **Lưu Log trên server Seq:**

* Sau khi chạy chương trình, truy cập <http://localhost:5341/> và đăng nhập trang web với username là admin và password là 123456.
* Seq cung cấp nhiều chức năng quản lý Log như lọc chúng theo thời gian (ví dụ: Last 1d), tìm kiếm Log (ví dụ: @Level='info', …) và xóa các Log bằng cách vào Settings → Retention → Manually delete events → Biểu tượng thùng rác màu đỏ → OK.

1. **Tìm kiếm Log theo level, method hoặc khoảng thời gian tạo:**

* **Phương thức**: GET
* **URL**: /logs
* **Chuỗi query (tùy chọn các thuộc tính)**: level (info, error), method (GET, POST, PUT, …), startDate=<YYYY>-<MM><DD>T<hh>:<mm>:<ss>Z, endDate= startDate=<YYYY>-<MM><DD>T<hh>:<mm>:<ss>Z
* **Phản hồi**: Trả về danh sách Log theo level, method hoặc khoảng thời gian tạo được quy định trong chuỗi query.
* **Ví dụ**:

{

"statusCode": 200,

"message": "Loggers retrieved successfully",

"data": [/\* danh sách các Log \*/]

}

1. **Các thông báo lỗi thường gặp**
2. **404 Not Found**: Khi không tìm thấy diễn viên theo ID.
3. **400 Bad Request**: Khi dữ liệu đầu vào không hợp lệ.
4. **500 Internal Server Error**: Lỗi khi truy xuất hoặc xử lý dữ liệu từ cơ sở dữ liệu.