
Framework Report

for

FlexiChat

Version 1.0

Prepared by Tổng Quang Trung

29032024

1. Introduction

FlexiChat sử dụng Framework sau: React, Express.js.

React là một thư viện JavaScript linh hoạt để xây dựng giao diện người dùng tương tác, trong khi Express.js là một framework cho Node.js, thiết kế để xây dựng ứng dụng web và API nhanh chóng và dễ dàng.

2. Components

React:

- JSX (JavaScript Syntax Extension): Cho phép viết HTML trong JavaScript.
- Components: Các đơn vị độc lập tái sử dụng, có thể chứa state và logic.
- State: Quản lý trạng thái nội bộ của component.
- Props: Truyền dữ liệu giữa các components.
- Lifecycle Methods: Các phương thức quản lý vòng đời của component.
- Hooks: Cho phép sử dụng state và các tính năng React khác trong functional components.
- Context API: Cung cấp cách để truyền dữ liệu qua cây component mà không cần props.

Express.js:

- Middleware: Các hàm xử lý yêu cầu trước khi đến route handler.
- Routing: Định nghĩa các đường dẫn và phương thức HTTP cho ứng dụng.
- Error Handling: Xử lý lỗi một cách linh hoạt.
- Integration with Databases: Hỗ trợ kết nối với nhiều loại cơ sở dữ liệu.
- Templating Engines: Hỗ trợ các engine như Pug, EJS để tạo giao diện người dùng.
- Performance: Tối ưu hóa hiệu suất cho ứng dụng web và API.

3. Usage

React trong FlexiChat:

- Components: Tạo các thành phần tái sử dụng như Button, MessageList, v.v., mỗi thành phần có trạng thái và logic riêng.
- State: Quản lý trạng thái nội bộ của một component, ví dụ: tin nhắn đang nhập.

- Props: Truyền dữ liệu và sự kiện giữa các components, như việc truyền hàm gửi tin nhắn từ parent component xuống child component.
- Lifecycle Methods: Quản lý vòng đời của component, từ khi được tạo ra cho đến khi bị hủy, ví dụ: componentDidMount để thiết lập kết nối WebSocket khi component được render.
- Hooks: Cho phép sử dụng state và các tính năng React khác mà không cần viết class, ví dụ: useState để quản lý trạng thái, useEffect để xử lý side effects.

ExpressJS trong FlexiChat:

- Middleware: Các hàm xử lý yêu cầu HTTP, có thể thêm vào pipeline xử lý yêu cầu, ví dụ: xác thực người dùng, log yêu cầu.
- Routing: Định nghĩa các endpoint để xử lý các yêu cầu đến server, ví dụ: GET /messages để lấy danh sách tin nhắn.
- Socket.io: Thư viện được sử dụng với ExpressJS để xây dựng chức năng chat real-time, cho phép server gửi thông báo đến client mà không cần client yêu cầu.

Quản lý trạng thái trong FlexiChat:

- Global State Management: Sử dụng các thư viện như Redux hoặc Context API để quản lý trạng thái chung cho toàn bộ ứng dụng, ví dụ: thông tin người dùng đăng nhập, danh sách tin nhắn.
- Local State Management: Quản lý trạng thái cục bộ trong một component, ví dụ: trạng thái của input field.

Tối ưu hóa hiệu suất trong FlexiChat:

- ShouldComponentUpdate và React.memo: Giúp kiểm soát việc render lại component, tránh render không cần thiết.
- Lazy loading và Code Splitting: Chia nhỏ code và chỉ tải những phần cần thiết khi cần, giúp tăng tốc độ tải trang.

4. Benefits

React:

Hiệu suất cao: Sử dụng Virtual DOM để giảm thiểu số lần cập nhật DOM thực, tăng hiệu suất ứng dụng.

Tái sử dụng component: Khả năng tái sử dụng các component giúp tiết kiệm thời gian và công sức khi phát triển ứng dụng.

Cộng đồng lớn: Một cộng đồng lớn và sôi động hỗ trợ việc tìm kiếm giải pháp và chia sẻ kinh nghiệm.

Công cụ phát triển mạnh mẽ: React Developer Tools giúp debug và quản lý trạng thái ứng dụng dễ dàng hơn.

Express.js:

Thiết kế linh hoạt: Cho phép xây dựng ứng dụng web và API nhanh chóng với ít cấu hình.

Hỗ trợ ứng dụng thời gian thực: Tích hợp tốt với Socket.io để xây dựng ứng dụng thời gian thực như chat apps.

Dễ dàng mở rộng: Có thể mở rộng ứng dụng web thành ứng dụng di động một cách dễ dàng.

So sánh với các framework khác, React và Express.js mang lại nhiều lợi ích như hiệu suất cao, khả năng tái sử dụng code, và cộng đồng hỗ trợ mạnh mẽ, làm cho chúng trở thành lựa chọn phổ biến trong phát triển web hiện đại.

5. Limitations

Khó học.

6. Conclusion

React Framework:

- Xây dựng giao diện người dùng động và tương tác cao cho ứng dụng chat web.
- Tối ưu hóa hiệu suất với Virtual DOM, giúp cập nhật và quản lý trạng thái hiệu quả.
- Hỗ trợ xây dựng ứng dụng Single Page Application (SPA), cung cấp trải nghiệm người dùng mượt mà và liền mạch.

Express Framework:

- Cung cấp cơ sở hạ tầng server-side mạnh mẽ và linh hoạt cho ứng dụng chat.
- Quản lý và xử lý các yêu cầu HTTP, routing, và middleware một cách dễ dàng.
- Hỗ trợ tích hợp với các công cụ như Socket.io để xây dựng chức năng chat thời gian thực.