

Giới Thiệu

Tiếp tục hành trình - Bước vào một chủ đề cực kỳ quan trọng: Memory (Bộ nhớ) trong LangGraph

Trong phần này, chúng ta sẽ đào sâu vào một chủ đề đóng vai trò sống còn khi xây dựng agent bằng LangGraph – đó chính là **memory (bộ nhớ)**. Và không chỉ nói đến **trí nhớ ngắn hạn(short term memory)**, mà cả **trí nhớ dài hạn(long term memory)** nữa!

Ngay từ đầu, bạn sẽ nhận ra: đây chính là **một trong những yếu tố quan trọng nhất để tạo nên sự khác biệt** giữa một agent “bình thường” và một agent “đỉnh cao” sẵn sàng chạy trên môi trường thực tế (production).

Nói đơn giản:

Ứng dụng của bạn có thật sự thông minh và đáng nhớ hay chỉ dừng lại ở mức “cho vui” – tất cả nằm ở cách bạn xử lý memory.

Vậy chúng ta sẽ học gì?

Chúng ta sẽ bắt đầu với phần dễ trước – **trí nhớ ngắn hạn(short term memory)**. Đây là loại bộ nhớ gắn liền với **cuộc trò chuyện hiện tại** giữa người dùng và agent.

Nếu bạn muốn agent của mình **hiểu được mạch hội thoại** (biết bạn vừa nói gì, nhớ câu hỏi trước, không bị “mất trí nhớ ngắn hạn”), thì bạn cần phải **kích hoạt trí nhớ dài hạn(long term memory)** cho nó.

Và bạn sẽ thấy: Việc này **cực kỳ đơn giản** với LangGraph.

Có những thứ bạn cần nắm trước

Trong tài liệu chính thức của LangGraph, khi họ bắt đầu nói về memory, họ thường đưa ra rất nhiều khái niệm như:

- cách định dạng schema của state,
- dùng reducer tùy chỉnh (custom reducers),
- hay sử dụng nhiều schema trạng thái trong một ứng dụng.

Thành thật mà nói, **cách LangGraph viết tài liệu không phải lúc nào cũng dễ hiểu**, thậm chí có thể gây rối nếu bạn mới bắt đầu.

Vì vậy, chúng ta sẽ làm rõ ba khái niệm đó trước – theo cách **đơn giản, trực quan và đúng bản chất**, giúp bạn hiểu sâu mà **không bị rối bởi quá nhiều dòng code**.

Mục tiêu của phần này là gì?

- Giúp bạn hiểu rõ **trí nhớ ngắn hạn(short term memory)** hoạt động như thế nào trong LangGraph.
 - Làm quen với các khái niệm nâng cao: `state schema`, `custom reducer`, và `multiple schemas`.
 - Tập trung vào phần **tư duy và logic**, không sa đà vào chi tiết lập trình.
- Hãy sẵn sàng! Trong những bài tiếp theo, chúng ta sẽ cùng nhau khám phá kỹ hơn về 4 khái niệm trên, với những ví dụ thực tế và trực quan nhất.
- Bạn chỉ cần nhớ: **đừng để tài liệu khó hiểu làm bạn nản lòng - chúng ta sẽ đi từng bước, và bạn chắc chắn sẽ làm được.**

Tác giả: **Đỗ Ngọc Tú**
Công Ty Phần Mềm **VHTSoft**

Phiên bản #1
Được tạo 20 tháng 4 2025 04:19:54 bởi Đỗ Ngọc Tú
Được cập nhật 21 tháng 4 2025 05:30:52 bởi Đỗ Ngọc Tú