

Định tuyến(Router) trong LangGraph

Router trong LangGraph là gì?

Trong LangGraph, **Router** là một **Node đặc biệt** (tức một bước - **step**) cho phép bạn **tạo ra các nhánh (branches)** dựa trên dữ liệu hoặc trạng thái hiện tại.

Nói đơn giản:

“**Router giống như ngã ba đường** — dựa trên state hiện tại, bạn chọn hướng đi tiếp theo (chuyển tới node A, B hoặc C).”

Khi nào bạn cần dùng Router?

Khi bạn có logic như:

- Nếu người dùng chọn "Coffee" → đi tới node xử lý coffee.
- Nếu người dùng chọn "Tea" → đi tới node xử lý trà.
- Nếu không rõ → hỏi lại hoặc kết thúc.

Bạn không muốn xử lý tất cả trong một node, mà **tách nhánh theo logic**, thì **Router** là giải pháp.

Cách dùng Router trong LangGraph

```
def my_router(state):  
    if state["preference"] == "coffee":  
        return "coffee_node"  
    elif state["preference"] == "tea":  
        return "tea_node"  
    else:  
        return "unknown_preference"
```

Đây là một **Router node**, tức là nó nhận một **state** (trạng thái hiện tại), và **trả về tên node kế tiếp** dựa trên logic.

`state["preference"]` là giá trị đầu vào mà bạn đưa vào để quyết định nhánh.

- Nếu thích **coffee**, thì đi tới node `coffee_node`.
- Nếu thích **tea**, thì đi tới node `tea_node`.
- Nếu không rõ, thì về `unknown_preference` (mặc dù node này **chưa được thêm** trong ví dụ – mình sẽ nói thêm ở dưới).

Đăng ký Router node trong graph

Ví dụ sau, Router không thực thi hành động chính mà chọn node tiếp theo dựa trên logic phân nhánh.

```
builder = StateGraph(StateSchema)
builder.add_node("start", start_node)
builder.add_node("router", my_router)
builder.add_node("coffee_node", coffee_node)
builder.add_node("tea_node", tea_node)

# Router branching
builder.add_edge("start", "router")
builder.add_conditional_edges(
    "router",
    {
        "coffee_node": "coffee_node",
        "tea_node": "tea_node"
    }
)
```

- Node router sử dụng hàm `my_router`, hàm này sẽ trả về các giá trị là `tea_node`, `coffee_node` hoặc `unknown_preference`, LangGraph dùng kết quả đó để **tra trong từ điển** :

```
{
    "coffee_node": "coffee_node",
    "tea_node": "tea_node"
}
```

- Nếu kết quả là `"coffee_node"` → đi đến node `"coffee_node"`
- Nếu kết quả là `"tea_node"` → đi đến node `"tea_node"`

Lợi ích của Router

- Tách logic phức tạp thành các bước nhỏ.
- Tạo flow rõ ràng và dễ debug.
- Xây dựng app agent có khả năng phân nhánh tùy theo trạng thái.
- Dễ áp dụng trong các hệ thống có điều kiện (if/else).

Gợi ý dùng thực tế

Use case	Router logic
Trợ lý cá nhân	Phân nhánh theo yêu cầu người dùng: tra cứu lịch, gửi mail, hoặc dịch văn bản
Ứng dụng học tập	Phân hướng học viên đến bài kiểm tra phù hợp trình độ
Ứng dụng đặt hàng	Nếu sản phẩm còn → đặt hàng; nếu không → đề xuất khác

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

Phiên bản #3
Được tạo 18 tháng 4 2025 02:06:56 bởi Đỗ Ngọc Tú
Được cập nhật 21 tháng 4 2025 05:23:34 bởi Đỗ Ngọc Tú