

So sánh Router và Node trong LangGraph

Dưới đây là **bài so sánh chi tiết giữa Router và Node** trong LangGraph, giúp bạn dễ phân biệt và áp dụng chính xác trong từng tình huống.

Tiêu chí	Node (Nút xử lý)	Router (Bộ định tuyến)
Chức năng chính	Xử lý logic cụ thể và trả về dữ liệu mới	Định tuyến (chọn nhánh tiếp theo dựa trên trạng thái)
Trả về	Một dictionary chứa các giá trị mới cho state	Một chuỗi (string) tên của node tiếp theo
Điều hướng tiếp theo	Sử dụng <code>add_edge()</code> hoặc <code>add_conditional_edges()</code>	Sử dụng <code>add_conditional_edges()</code> để ánh xạ chuỗi → node
Tính logic	Chứa logic chính như gọi LLM, xử lý dữ liệu...	Chứa logic quyết định hướng đi tiếp theo
Dữ liệu trả về	Ví dụ: <code>{ "graph_state": "Tôi chọn trà" }</code>	Ví dụ: <code>"coffee_node"</code> hoặc <code>"tea_node"</code>
Dùng khi nào?	Khi cần thực hiện một hành động/ghi nhớ dữ liệu	Khi cần rẽ nhánh theo điều kiện hoặc phân luồng logic

Ví dụ minh họa

Node bình thường trả về giá trị mới cho state:

```
def coffee_node(state):
    print("☕ Tôi chọn coffee")
    return {"result": "Bạn sẽ có một ly coffee"}
```

Router trả về tên của node tiếp theo:

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

Khi nào dùng Router thay vì Node?

- Khi bạn cần **rẽ nhánh logic** theo điều kiện → dùng `router`.
- Khi bạn cần **thực hiện hành động cụ thể** (gọi LLM, tính toán, ghi log,...) → dùng `node`.

Tác giả: Đỗ Ngọc Tú

Công Ty Phần Mềm VHTSoft

Phiên bản #1

Được tạo 21 tháng 4 2025 05:25:34 bởi Đỗ Ngọc Tú

Được cập nhật 21 tháng 4 2025 05:30:52 bởi Đỗ Ngọc Tú