

LangGraph: Framework Xây Dựng Ứng Dụng LLM Dựa Trên Agent + Graph

LangGraph là một framework mã nguồn mở được xây dựng trên nền tảng của **LangChain**, cho phép bạn phát triển các ứng dụng sử dụng **LLM (Large Language Models)** theo mô hình **agentic** – nơi AI không chỉ trả lời câu hỏi, mà còn **suy luận, quyết định, gọi tool, và ghi nhớ trạng thái** trong nhiều bước.

Điểm đặc biệt của LangGraph là nó tổ chức toàn bộ logic ứng dụng dưới dạng một **đồ thị trạng thái (stateful graph)** – mỗi "node" là một hành động, một agent, hoặc một bước xử lý.

Tại sao cần LangGraph?

Các ứng dụng AI ngày nay không chỉ dừng ở hỏi-đáp đơn lẻ. Chúng cần:

- Gọi nhiều công cụ (tool use)
- Ghi nhớ trạng thái để suy luận liên tục
- Lặp lại hành động hoặc rẽ nhánh theo kết quả
- Kiểm soát logic phức tạp có điều kiện

LangGraph giúp bạn làm tất cả điều đó **một cách có cấu trúc, dễ theo dõi và mở rộng**.

LangGraph hoạt động như thế nào?

LangGraph xây dựng ứng dụng theo mô hình **Finite-State Machine (FSM)**:

- **Mỗi node** là một bước xử lý (ví dụ: gọi LLM, tìm kiếm, phân tích, v.v.)
- **State** là dữ liệu được lưu và truyền qua từng bước
- **Edge** quyết định luồng đi tiếp theo dựa trên kết quả hoặc trạng thái

Bạn có thể tạo các ứng dụng như:

- Chatbot thông minh
- Multi-tool agent (Google + Calculator + Document Retriever...)
- Workflow phân tích dữ liệu
- Trợ lý quy trình nghiệp vụ (ERP, CRM...)

Cú pháp cơ bản

```
from langgraph.graph import StateGraph

def greet(state): print("Hello!"); return state
def ask_name(state): state["name"] = "Alice"; return state

builder = StateGraph(dict)
builder.add_node("greet", greet)
builder.add_node("ask", ask_name)
builder.set_entry_point("greet")
builder.add_edge("greet", "ask")

graph = builder.compile()
graph.invoke({})
```

Phiên bản #3

Được tạo 14 tháng 4 2025 04:01:11 bởi Đỗ Ngọc Tú

Được cập nhật 29 tháng 4 2025 18:21:33 bởi Đỗ Ngọc Tú