

# LangGraph Studio, giới thiệu, cài đặt và cách dùng

Giới thiệu LangGraph Studio - Trình quan sát & phát triển Agentic Graphs

## I. LangGraph Studio là gì?

LangGraph Studio là một công cụ **giao diện đồ họa (GUI)** giúp bạn:

- Quan sát trực quan đồ thị agent (agent graph)
- Gỡ lỗi (debug) các bước thực thi của LLM agent
- Theo dõi trạng thái, dữ liệu, và hướng đi qua từng node
- Phát triển và kiểm thử agent nhanh chóng

Nó giống như một **"developer console"** dành riêng cho LangGraph, giúp bạn hiểu và kiểm soát quá trình suy luận và hành động của mô hình AI một cách rõ ràng, trực quan.

## II. Tại sao nên dùng LangGraph Studio?

Vấn đề khi không có Studio	LangGraph Studio giúp
<input type="checkbox"/> Debug khó vì agent ẩn trong LLM	Hiển thị chi tiết từng bước, trạng thái
<input type="checkbox"/> Khó hiểu LLM đang đi hướng nào	Hiển thị flow trực quan
<input type="checkbox"/> Phát triển agent lặp đi lặp lại	Test từng phần trực tiếp
<input type="checkbox"/> Không biết LLM nghĩ gì	Hiển thị prompt, output, action, tool dùng

## III. Cài đặt LangGraph Studio

LangGraph Studio được tích hợp sẵn trong thư viện `langgraph`. Để cài đặt và sử dụng, bạn chỉ cần:

### 1. Cài đặt thư viện:

```
pip install langgraph[dev]
```

Hoặc nếu bạn đã có `langgraph`:

```
pip install -U "langgraph[dev]"
```

`dev` extras sẽ cài thêm các gói phục vụ cho Studio như `fastapi`, `uvicorn`, `jupyter`, v.v.

## IV. Cách dùng LangGraph Studio

LangGraph Studio hiện hỗ trợ **hai cách chính để sử dụng**:

### 1. Chạy dưới dạng local app (FastAPI server)

Bạn có thể thêm dòng sau vào project của bạn để chạy Studio như một app mini:

```
from langgraph.studio import start_trace_server

start_trace_server()
```

Mặc định server chạy tại `http://localhost:4000`

Khi bạn chạy graph của mình, Studio sẽ tự động ghi lại luồng xử lý, trạng thái, inputs/outputs, và hiển thị trên giao diện web.

### 2 . Tích hợp trong Jupyter Notebook

Nếu bạn đang làm việc trong Jupyter (hoặc Google Colab), bạn có thể dùng:

```
from langgraph.studio.jupyter import trace_graph

with trace_graph(graph) as session:
    result = session.invoke(input_data)
```

Tính năng này giúp bạn:

- Thấy flow đồ thị ngay trong notebook
- Click để xem chi tiết từng node, prompt, tool call, response
- So sánh nhiều lần chạy khác nhau (useful for debugging)

```
from langgraph.graph import StateGraph
from langgraph.studio import start_trace_server

# Define some simple node functions
def say_hello(state):
    print("Hello!")
    return state

def ask_name(state):
    state["name"] = "Alice"
    return state

# Build graph
builder = StateGraph(dict)
builder.add_node("hello", say_hello)
builder.add_node("ask", ask_name)
builder.set_entry_point("hello")
builder.add_edge("hello", "ask")

graph = builder.compile()

# Start studio server
start_trace_server()

# Run
graph.invoke({})
```

Khi chạy đoạn code này, bạn có thể truy cập `http://localhost:4000` để thấy luồng `hello → ask` cùng với trạng thái và log chi tiết.

## VI. Các tính năng nổi bật

Tính năng	Mô tả
Flow Graph View	Hiển thị đồ thị agent và luồng thực tế đã đi qua
Prompt Viewer	Hiển thị đầy đủ prompt mà LLM nhận được
Input/Output Log	Xem tất cả input và output của từng bước

Tính năng	Mô tả
Multi-run Comparison	So sánh nhiều phiên bản chạy graph khác nhau
Token usage & Cost (tương lai)	Theo dõi chi phí sử dụng OpenAI, Anthropic,...
Node debug	Hiển thị error và traceback nếu có lỗi ở một node

## VII. Khi nào nên dùng LangGraph Studio?

- Khi bạn xây dựng agent có nhiều bước
- Khi bạn xử lý luồng có điều kiện, vòng lặp, branch
- Khi cần kiểm soát tool call / prompt / phản hồi
- Khi muốn debug graph state dễ dàng

## VIII. Tổng kết

Câu hỏi	Trả lời
LangGraph Studio là gì?	Là công cụ GUI để quan sát và phát triển LangGraph Agents một cách trực quan và có thể debug.
Làm gì được với nó?	Theo dõi prompt, response, trạng thái, tool call... từng bước trong graph
Cài đặt ra sao?	<code>pip install langgraph[dev]</code>
Chạy ở đâu?	Trong local (FastAPI) hoặc trong Jupyter Notebook

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

Phiên bản #2  
Được tạo 14 tháng 4 2025 03:43:13 bởi Đỗ Ngọc Tú  
Được cập nhật 14 tháng 4 2025 03:51:31 bởi Đỗ Ngọc Tú