

[Open in app](#)

Sign in

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.



Mediu

Writing is for everyone.

[Register for Medium Day](#)



Artificial Intelligenc... · [Follow publication](#)

You're reading for free via [Sajith K's Friend Link](#). [Become a member](#) to access the best of Medium.

Member-only story

CopilotKit (AG-UI): Integrating LangGraph with Next.js and FastAPI

6 min read · Jul 11, 2025



Sajith K

[Follow](#)[Listen](#)[Share](#)

Creating an MCP Server and Integrating with LangGraph

The Model Context Protocol (MCP) is an open protocol that standardizes how applications provide context to LLMs. MCP...

[medium.com](https://medium.com/@sajith_k/copilotkit-ag-ui-integrating-langgraph-with-next-js-and-fastapi-435cac2df56b)

Introduction

Agent User Interaction Protocol (AG-UI) is a lightweight, event-based protocol that standardizes how AI agents connect to front-end applications. Built for simplicity and flexibility, it enables seamless integration between your AI agents and user interfaces.

Full story link for non members: https://medium.com/@sajith_k/copilotkit-ag-ui-integrating-langgraph-with-next-js-and-fastapi-435cac2df56b

[source=friends link&elk=58306c253c092ded2486d6654ee792c8](#)

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

- MCP gives agents tools
- A2A allows agents to communicate with other agents
- AG-UI brings agents into user-facing applications



Features

- 💬 Real-time agentic chat with streaming
- 🔄 Bi-directional state synchronization
- 🧩 Generative UI and structured messages
- 🧠 Real-time context enrichment
- ✨ Frontend tool integration
- 👤 Human-in-the-loop collaboration

CopilotKit is an open-source Agentic Application Framework and hosted service for AI-assisted applications built on top of the Agent User Interaction Protocol (AG-UI).

In this post, we'll walk through integrating a LangGraph agent deployed with FastAPI into a Next.js application using CopilotKit.

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Create LangGraph Agent

First, set up a LangGraph agent to handle the AI logic and deploy the agent using CopilotKit's FastAPI integration.

Requirements

```
requires-python = ">=3.12"
dependencies = [
    "copilotkit>=0.1.54",
    "fastapi>=0.115.14",
    "langchain>=0.3.26",
    "langchain-openai>=0.3.27",
    "langgraph>=0.4.10",
    "uvicorn>=0.35.0",
]
```

Code

Create files `agent.py` for creating the agent and `server.py` for deploying the agent.

```
my-agent
└── agent.py
└── server.py
```

1. agent.py

```
from typing import Annotated, TypedDict
from langgraph.graph.message import add_messages
from langgraph.graph.message import AnyMessage
from langgraph.graph import START, END, StateGraph
from langchain_openai import ChatOpenAI
from langgraph.checkpoint.memory import MemorySaver
from langgraph.types import Checkpointer

class State(TypedDict):
    messages: Annotated[list[AnyMessage], add_messages]
def chat_node(state: State) -> State:
    model = ChatOpenAI(model_name="gpt-4o", temperature=0, api_key="your openai key")
    state["messages"] = [AnyMessage(role="user", content="Hello")]
```

```
state["messages"] = model.invoke(state["messages"])
rei To make Medium work, we log user data. By using Medium, you agree to
graph_t our Privacy Policy, including cookie policy.
graph_builder.add_node(START, chat_node)
graph_builder.add_edge(START, "chat_node")
graph_builder.add_edge("chat_node", END)
graph = graph_builder.compile(checkpointer=MemorySaver())
```

2. server.py

```
from copilotkit.integrations.fastapi import add_fastapi_endpoint
from copilotkit import CopilotKitRemoteEndpoint, LangGraphAgent
from agent import graph
from fastapi import FastAPI

app = FastAPI()

# Initialize the CopilotKit SDK
sdk = CopilotKitRemoteEndpoint(agents=[
    LangGraphAgent(
        name="agent",
        description="An example agent to use as a starting point for your c
        graph=graph,
    )
], actions=[])

# Add the CopilotKit endpoint to your FastAPI app
add_fastapi_endpoint(app, sdk, "/copilotkit_remote", max_workers=10)

def main():
    """Run the uvicorn server."""
    import uvicorn
    uvicorn.run("server:app", host="0.0.0.0", port=8000, reload=True)

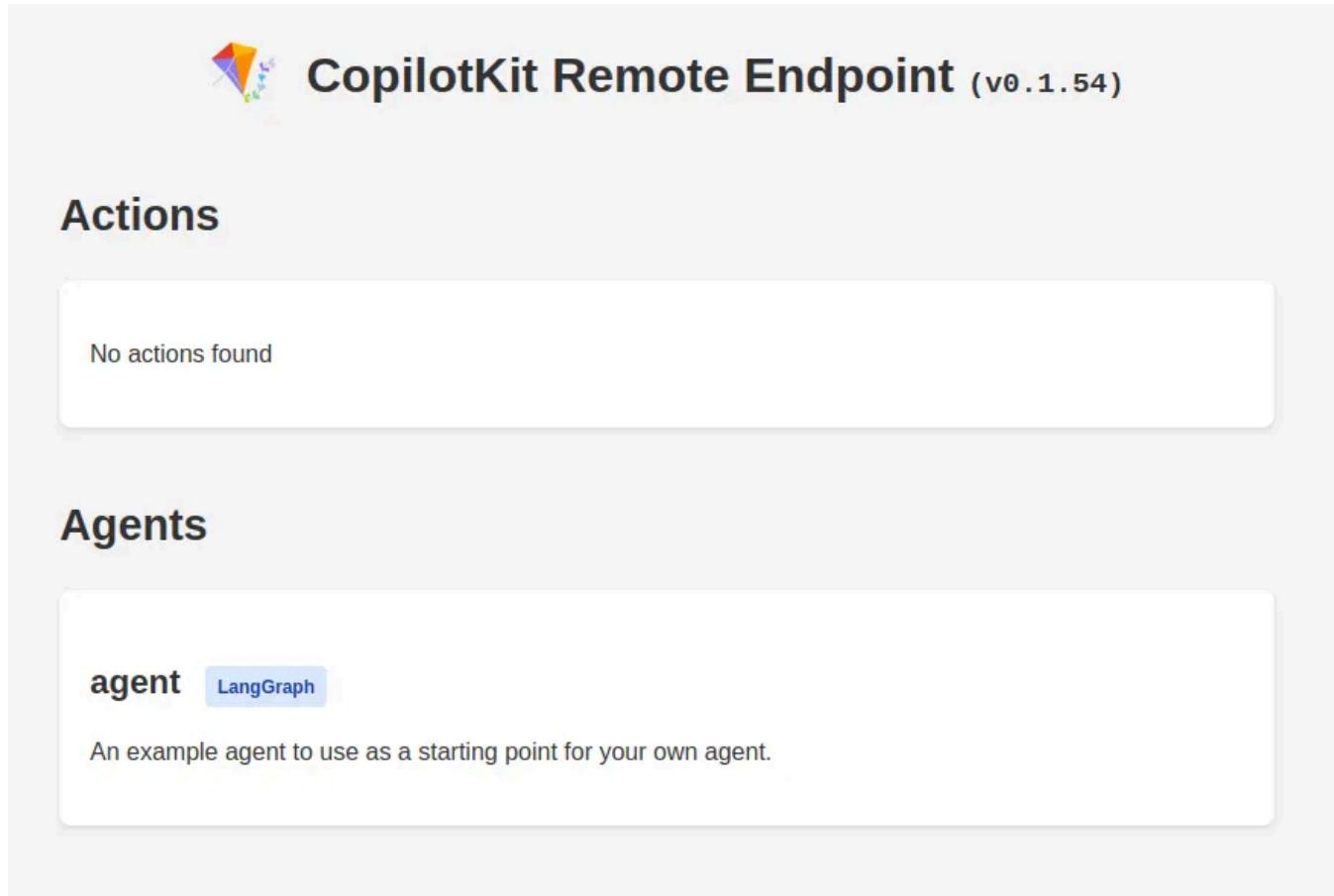
if __name__ == "__main__":
    main()
```

Execute the server

python To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Check the CopilotKit endpoint: http://localhost:8000/copilotkit_remote/

Output



The screenshot shows the CopilotKit Remote Endpoint interface. At the top, there is a logo of a colorful kite and the text "CopilotKit Remote Endpoint (v0.1.54)". Below this, the "Actions" section is shown with the message "No actions found". In the "Agents" section, there is one agent listed: "agent LangGraph", described as "An example agent to use as a starting point for your own agent."

Create the Next.js App

Create a new Next.js application.

```
npx create-next-app@latest
```

Install CopilotKit: Install the necessary CopilotKit packages for the frontend.

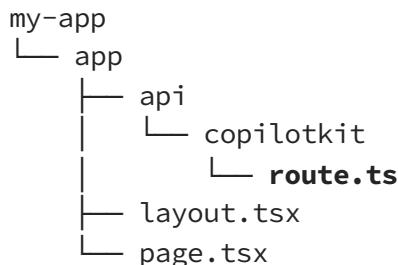
npm ins: To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Install Copilot Runtime: Copilot Runtime is a production-ready proxy for your LangGraph agents.

```
npm install @copilotkit/runtime class-validator
```

Set Up a Copilot Runtime Endpoint

Create a new route to handle the `/api/copilotkit` endpoint in your Next.js app (`app/api/copilotkit/route.ts`).



```

import {
  CopilotRuntime,
  ExperimentalEmptyAdapter,
  copilotRuntimeNextJSAppRouterEndpoint,
} from "@copilotkit/runtime";
import { NextRequest } from "next/server";

// You can use any service adapter here for multi-agent support.
const serviceAdapter = new ExperimentalEmptyAdapter();
const runtime = new CopilotRuntime({
  remoteEndpoints: [
    // Our FastAPI endpoint URL
    { url: "http://localhost:8000/copilotkit_remote" },
  ],
});
  
```

```
export
const To make Medium work, we log user data. By using Medium, you agree to
  run our Privacy Policy, including cookie policy.
  setEnvironment,
  endpoint: "/api/copilotkit",
);
return handleRequest(req);
};
```

Configure the CopilotKit Provider

Wrap your application with the `<CopilotKit>` component in `layout.tsx` to enable CopilotKit functionality.

The `<CopilotKit>` component must wrap the Copilot-aware parts of your application. For most use-cases, it's appropriate to wrap the CopilotKit provider around the entire app

```
import "./globals.css";
import { ReactNode } from "react";
import { CopilotKit } from "@copilotkit/react-core";

export default function RootLayout({ children }: { children: ReactNode }) {
  return (
    <html lang="en">
      <body>
        <CopilotKit
          runtimeUrl="/api/copilotkit"
          agent="agent" // the name of the agent you want to use
        >
          {children}
        </CopilotKit>
      </body>
    </html>
  );
}
```

Choose a Configuration

Import the Copilot UI

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

```
import "@copilotkit/react-ui/styles.css";
```

Available UIs

- CopilotChat
- CopilotSidebar
- CopilotPopup
- Headless UI

For this example, we'll use `CopilotSidebar`, which provides a collapsible and expandable sidebar chat interface.

```
import { CopilotSidebar } from "@copilotkit/react-ui";

export function YourApp() {
  return (
    <CopilotSidebar
      defaultOpen={true}
      instructions={"You are assisting the user as best as you can. Answer in text."}
      labels={{
        title: "Sidebar Assistant",
        initial: "How can I help you today?",
      }}
    >
      <YourMainContent />
    </CopilotSidebar>
  );
}
```

Updated

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

```
my-app
└── app
    ├── api
    │   └── copilotkit
    │       └── route.ts
    └── layout.tsx
        └── page.tsx
```

```
import type { Metadata } from "next";
import { Geist, Geist_Mono } from "next/font/google";
import "./globals.css";
import { CopilotKit } from "@copilotkit/react-core";
import "@copilotkit/react-ui/styles.css";
import { CopilotSidebar } from "@copilotkit/react-ui";

const geistSans = Geist({
  variable: "--font-geist-sans",
  subsets: ["latin"],
});

const geistMono = Geist_Mono({
  variable: "--font-geist-mono",
  subsets: ["latin"],
});

export const metadata: Metadata = {
  title: "Create Next App",
  description: "Generated by create next app",
};

export default function RootLayout({
  children,
}: Readonly<{
  children: React.ReactNode;
}>) {
  return (
    <html lang="en">
      <body
        className={`${geistSans.variable} ${geistMono.variable} antialiased`}
      >
        <CopilotKit
          runtimeUrl="/api/copilotkit"
          agent="agent" // the name of the agent you want to use
        >
          {children}
        </CopilotKit>
      </body>
    </html>
  );
}
```

>

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

can. Answer

```

    title: "Sidebar Assistant",
    initial: "How can I help you today?",
  }}
>
{children}
</CopilotSidebar>
</CopilotKit>
</body>
</html>
);
}

```

Execute the Next.js app

```
npm run dev
```

Check the next.js app: <http://localhost:3000>

Output



1. Get started by editing `app/page.tsx`.
2. Save and see your changes instantly.

[▲ Deploy now](#) [Read our docs](#)



To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Talk to Yo... - - - - -

Congratulations! You've successfully integrated a LangGraph agent chatbot into your Next.js application using CopilotKit. Start by asking your agent a few questions to test the integration.

Adding speech capability (optional)

TTS and STT Endpoints Using ElevenLabs

Add text-to-speech (TTS) and speech-to-text (STT) endpoints to your FastAPI server using ElevenLabs Python SDK for enhanced interaction capabilities.

Add the following code to `server.py`

```
from elevenlabs.client import ElevenLabs

# Load environment variables
load_dotenv()
ELEVENLABS_API_KEY = os.getenv("ELEVENLABS_API_KEY")

# Initialize ElevenLabs client
elevenlabs_client = ElevenLabs(api_key=ELEVENLABS_API_KEY)

@app.get("/tts")
async def text_to_speech(text: str):
    response = elevenlabs_client.text_to_speech.stream(
        text=text,
        voice_id="IKne3meq5aSn9XLyUdCD",
        model_id="eleven_multilingual_v2",
        output_format="mp3_44100_128",
    )

    audio_stream = BytesIO()

    # Write each chunk of audio data to the stream
    for chunk in response:
        if chunk:
            audio_stream.write(chunk)

    # Reset stream position to the beginning
    audio_stream.seek(0)
```

re¹

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

```
        headers={ "Content-Disposition": "attachment; filename=output.mp3" },
    )

@app.post("/stt")
async def speech_to_text(file: UploadFile = File(...)):
    # Read audio data
    audio_data = await file.read()
    if not audio_data:
        raise HTTPException(status_code=400, detail="No audio data provided.")

    transcription = elevenlabs_client.speech_to_text.convert(
        file=audio_data,
        model_id="scribe_v1", # Model to use, for now only "scribe_v1" is supported
        tag_audio_events=True, # Tag audio events like laughter, applause, etc.
        language_code="eng", # Language of the audio file. If set to None, the diarize=True, # Whether to annotate who is speaking
    )

    return transcription
```

Adding the Endpoints to CopilotKit

Update the `<CopilotKit>` component in `layout.tsx` to include the TTS and STT endpoints.

```
<CopilotKit
  runtimeUrl="/api/copilotkit"
  agent="agent"
  textToSpeechUrl="http://localhost:8000/tts"
  transcribeAudioUrl="http://localhost:8000/stt"
>
```

With added TTS and STT capabilities via ElevenLabs you can speak to your agent.

Full Code

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

GitHub - Sajith-K-Sasi/copilotkit_langgraph_chat

Copilotkit langgraph chat with fastapi

github.com

BUILDING A REAL-TIME SPEECH-ENABLED LANGGRAPH AGENT WITH LIVEKIT

This tutorial guides you through setting up a LangGraph agent integrated with LiveKit to enable real-time speech...

medium.com

A message from our Founder

Hey, Sunil here. I wanted to take a moment to thank you for reading until the end and for being a part of this community.

Did you know that our team run these publications as a volunteer effort to over 3.5m monthly readers? We don't receive any funding, we do this to support the community. ❤️

If you want to show some love, please take a moment to follow me on LinkedIn, TikTok, Instagram. You can also subscribe to our weekly newsletter.

And before you go, don't forget to clap and follow the writer!

UI

Nextjs

Langgraph

Llm

AI



To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

[Follow](#)

Published in Artificial Intelligence in Plain English

28K followers · Last published just now

New AI, ML and Data Science articles every day. Follow to join our 3.5M+ monthly readers.

[Follow](#)

Written by Sajith K

173 followers · 15 following

Responses (1)



Write a response

What are your thoughts?



Feng Yi
Jul 12

...

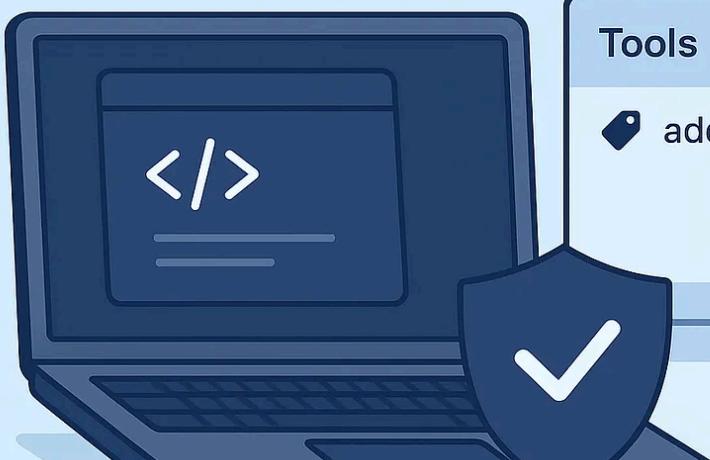
Thank you for share



[Reply](#)

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

More from Sajith K



Introducing FastMCP v2

The Pythonic Way to Build Secure MCP Servers and Clients for AI

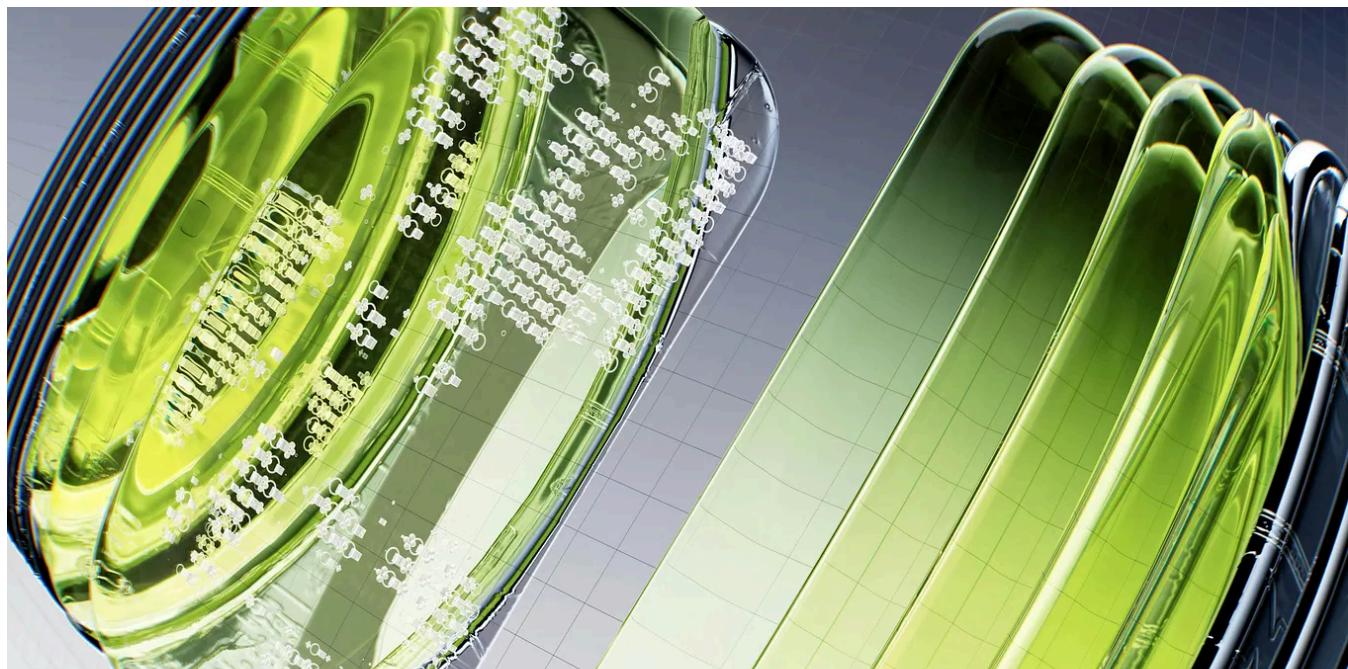
| Tools | Resources |
|---|---|
| <input checked="" type="checkbox"/> add | <input type="checkbox"/> greeting |
| | <input type="checkbox"/> greeting_by_name |
| Prompts | |
| | <input checked="" type="checkbox"/> ask_about_topic |

AI In Artificial Intelligence in Plain English by Sajith K

Introducing FastMCP v2: The Pythonic Way to Build Secure MCP Servers and Clients for AI

In the rapidly evolving world of AI, large language models (LLMs) need seamless access to external data and functionalities to go beyond...

Aug 27 125 2



AI In Artificial Intelligence

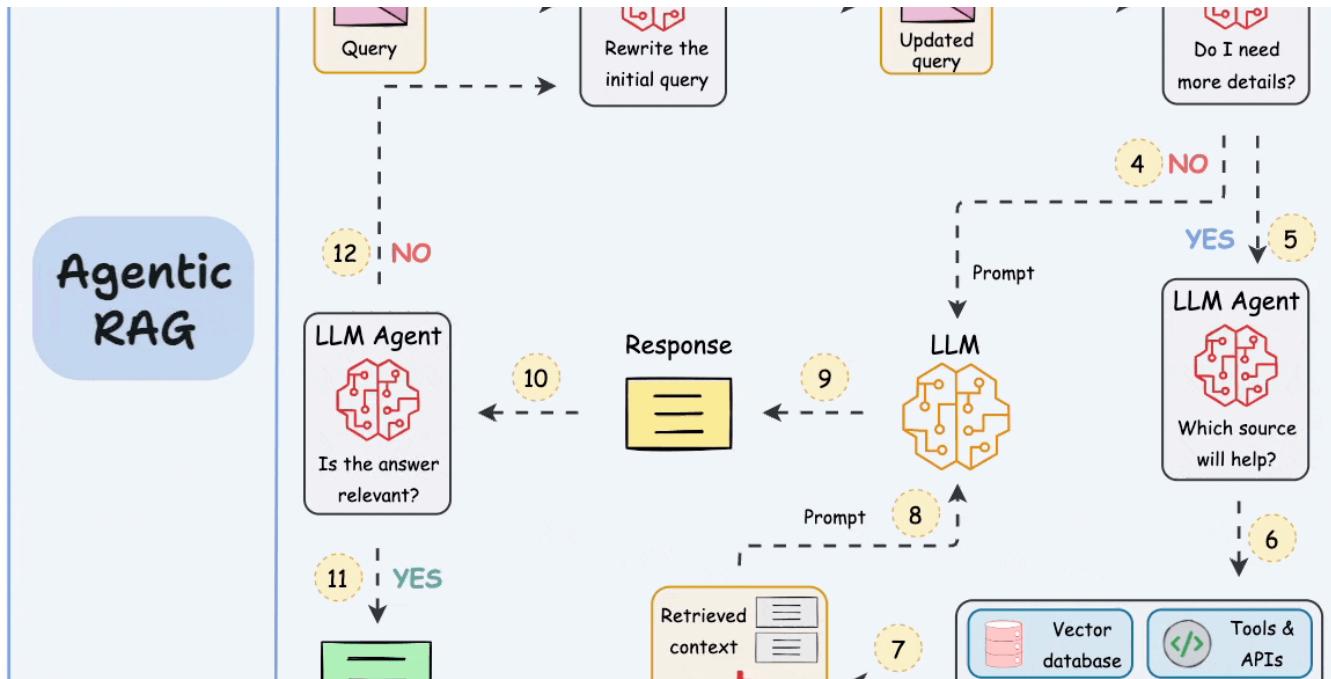
To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Here's how

I Finally Used LangGraph to Build an Agentic RAG System

It's been almost 2 years since I first encountered the "Attention is all you need" paper by Vaswani et al. (2017).

Jul 12 701 11

**AI** In Artificial Intelligence in Plain English by Piyush Agnihotri

Building Agentic RAG with LangGraph: Mastering Adaptive RAG for Production

Build intelligent RAG systems that know when to retrieve documents, search the web, or generate responses directly

Jul 20 1.9K 27





 In Artificial Intelligence in Plain English by Sajith K

Creating an MCP Server and Integrating with LangGraph

The Model Context Protocol (MCP) is an open protocol that standardizes how applications provide context to LLMs. MCP provides a...

 May 25  89  4



See all from Sajith K

See all from Artificial Intelligence in Plain English

Recommended from Medium

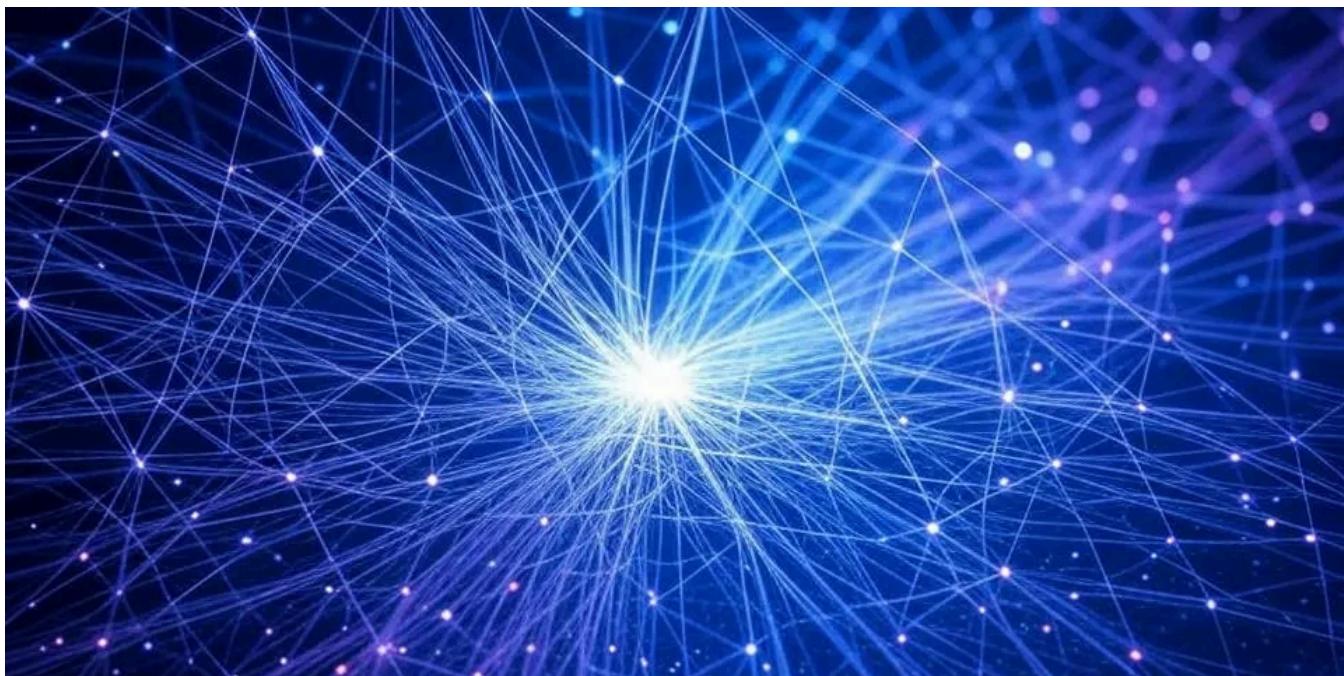


In AlgoMart by Yash Jain

Bringing AI Agents into Production: Why LangGraph Is Becoming the Framework of Choice

Prototyping an AI agent is fast. Getting one into production—where real-world performance meets unpredictable inputs and high...

Aug 21 114



In Artificial Intelligence in Plain English by Sajith K

Creating an MCP Server and Integrating with LangGraph

The Model C

provide con: To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

May 25

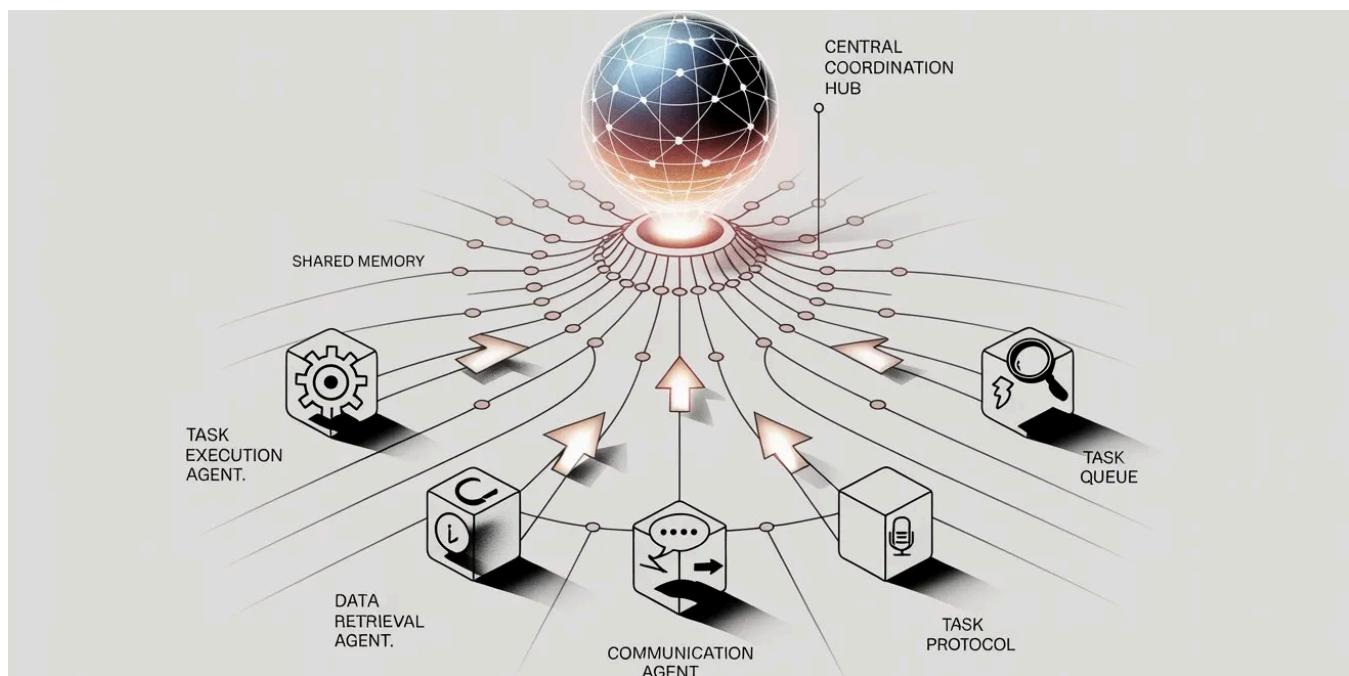


Yashwanth Sai

The DSPy Playbook: A Humble Introduction

A small part from my upcoming book on AI Agents. I'm building in public and sharing it as we go. Stay tuned!

Jul 23 258 3





Aman Raghuvanshi

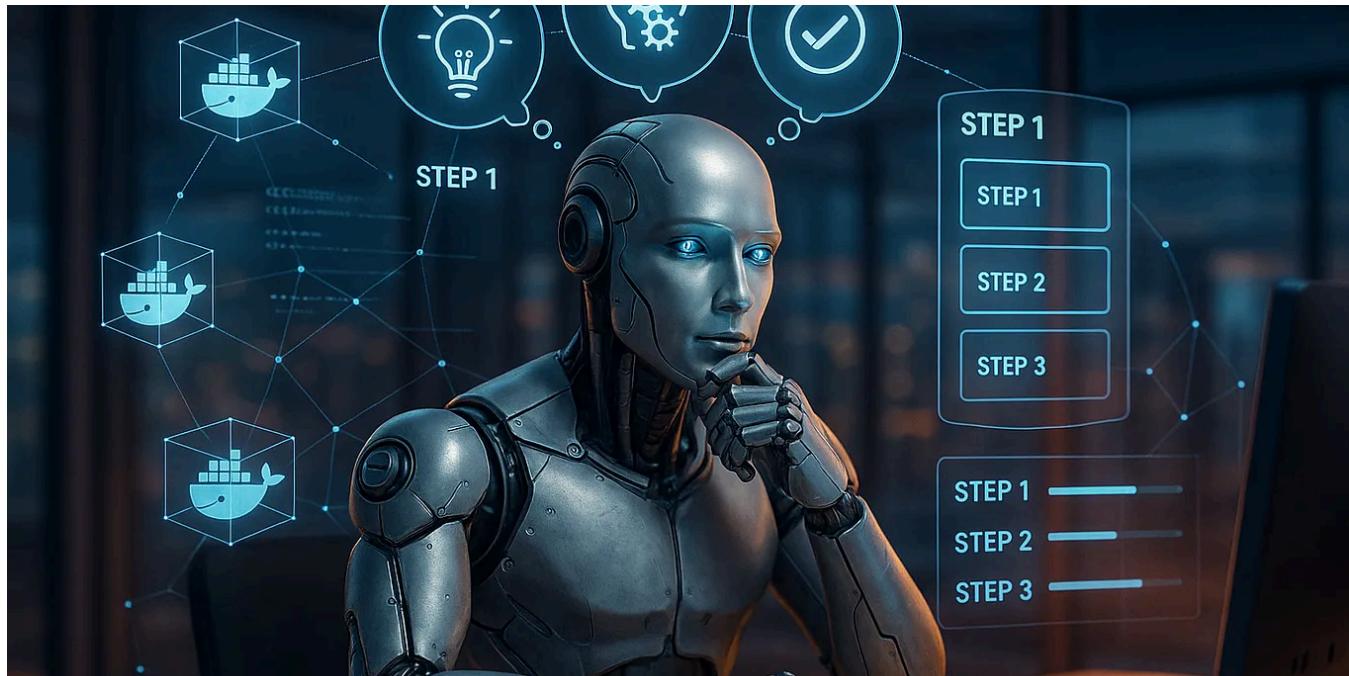
Agentic A Develop

To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.

Guide for

The concept of multi-agent systems (MAS) has been around for decades in distributed computing. Still, the advent of Large Language Models...

Aug 30 88 5



Micheal Lanham

Building AI Agents That Actually Think Before They Act: A Step-by-Step Guide to Sequential Thinking...

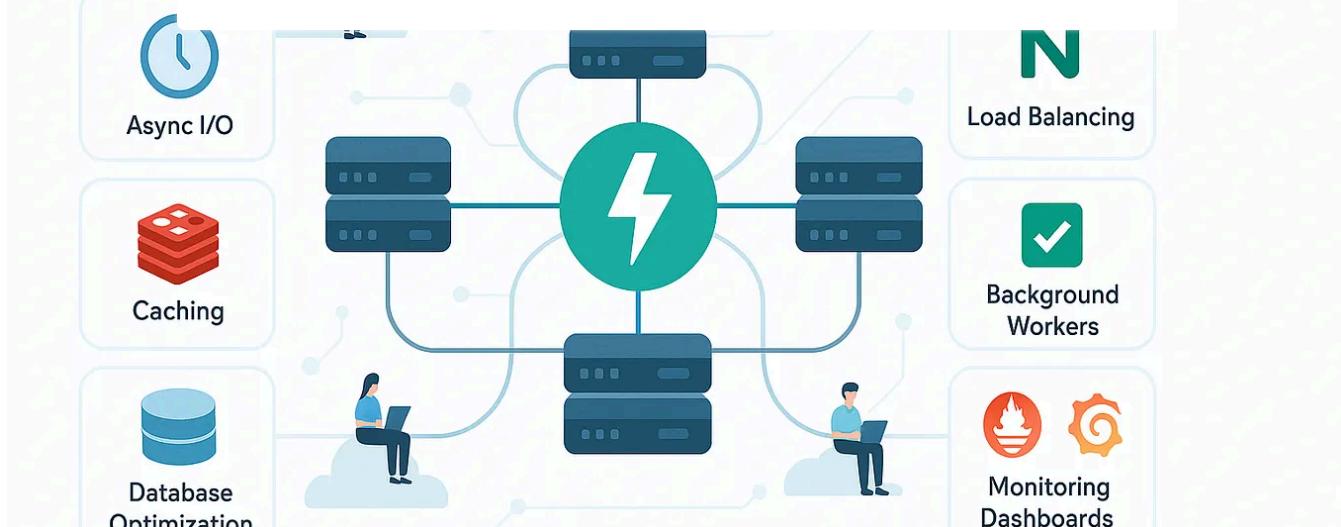
How to deploy OpenAI Agents with structured reasoning that breaks down complex problems like a human expert

Aug 21 24





To make Medium work, we log user data. By using Medium, you agree to our Privacy Policy, including cookie policy.



Hash Block

Scaling FastAPI for a Million Requests

Proven strategies to make your FastAPI app handle massive traffic without crashing.

Aug 10 216 1



See more recommendations