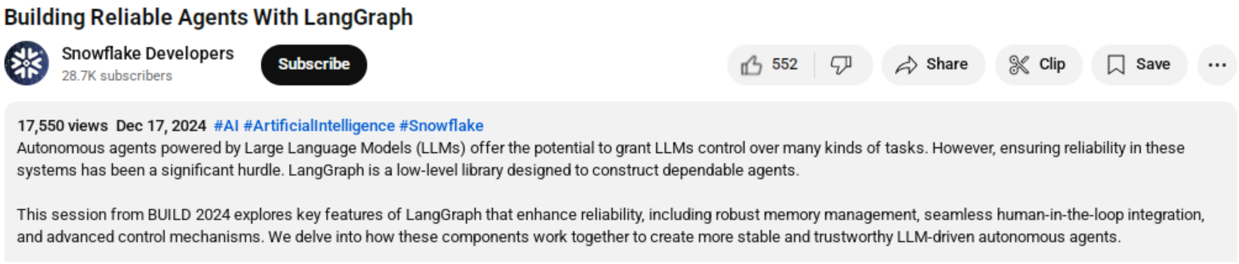


Building Reliable Agents With LangGraph

Snowflake Developers
28.7K subscribers

17,550 views Dec 17, 2024 #AI #ArtificialIntelligence #Snowflake

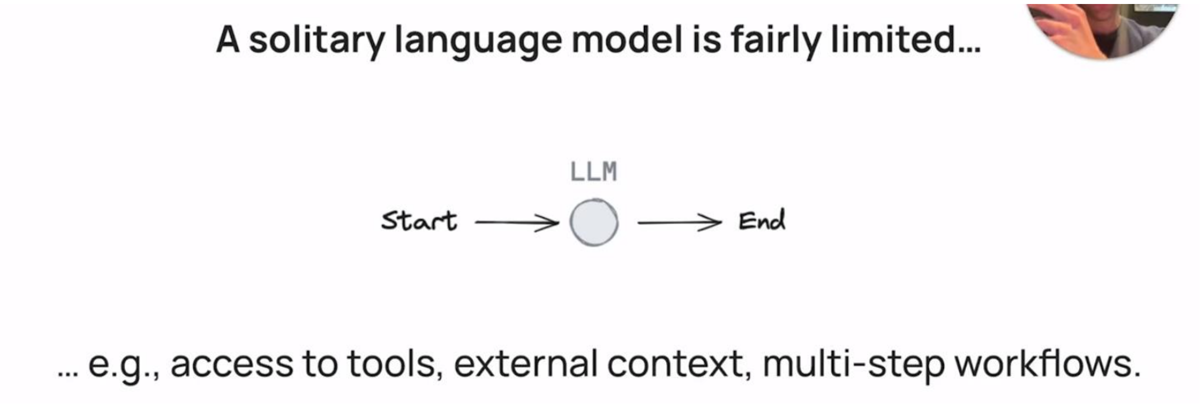
Autonomous agents powered by Large Language Models (LLMs) offer the potential to grant LLMs control over many kinds of tasks. However, ensuring reliability in these systems has been a significant hurdle. LangGraph is a low-level library designed to construct dependable agents.

This session from BUILD 2024 explores key features of LangGraph that enhance reliability, including robust memory management, seamless human-in-the-loop integration, and advanced control mechanisms. We delve into how these components work together to create more stable and trustworthy LLM-driven autonomous agents.

Building Reliable Agents Using LangGraph

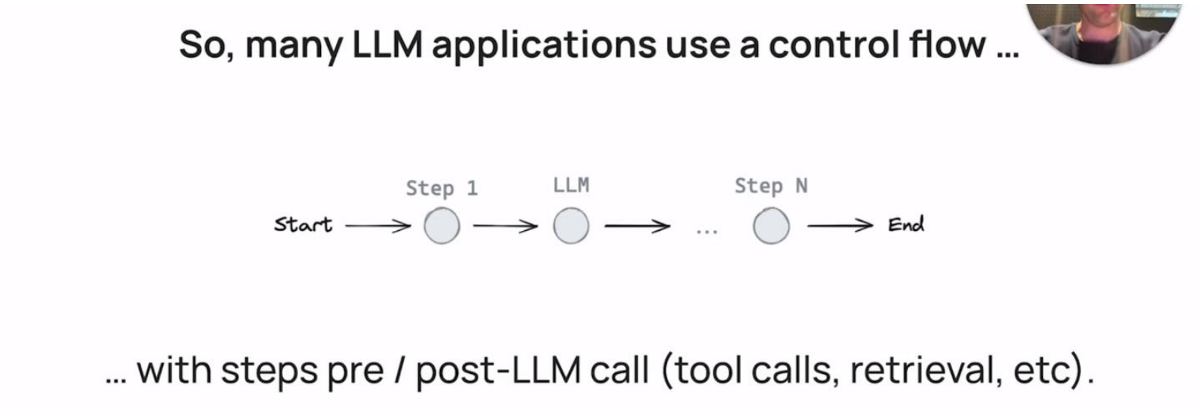
Lance Martin, Software Engineer

1 Introduction and Basics: Motivation



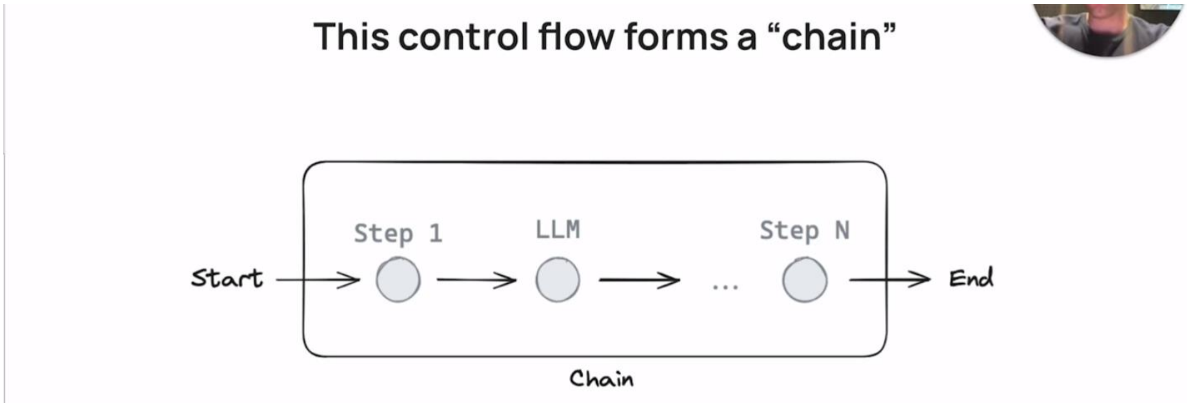
A solitary language model is fairly limited…

… e.g., access to tools, external context, multi-step workflows.

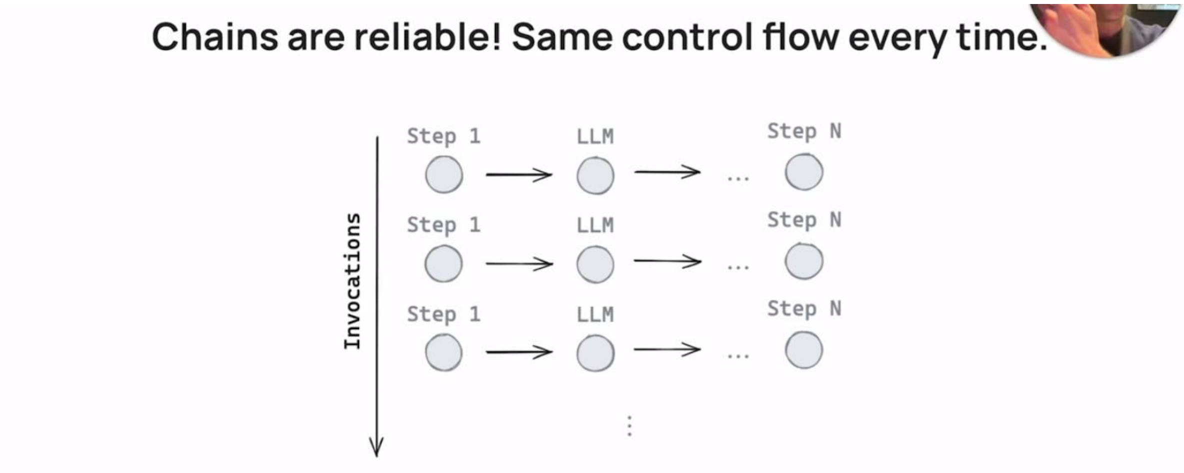


So, many LLM applications use a control flow …

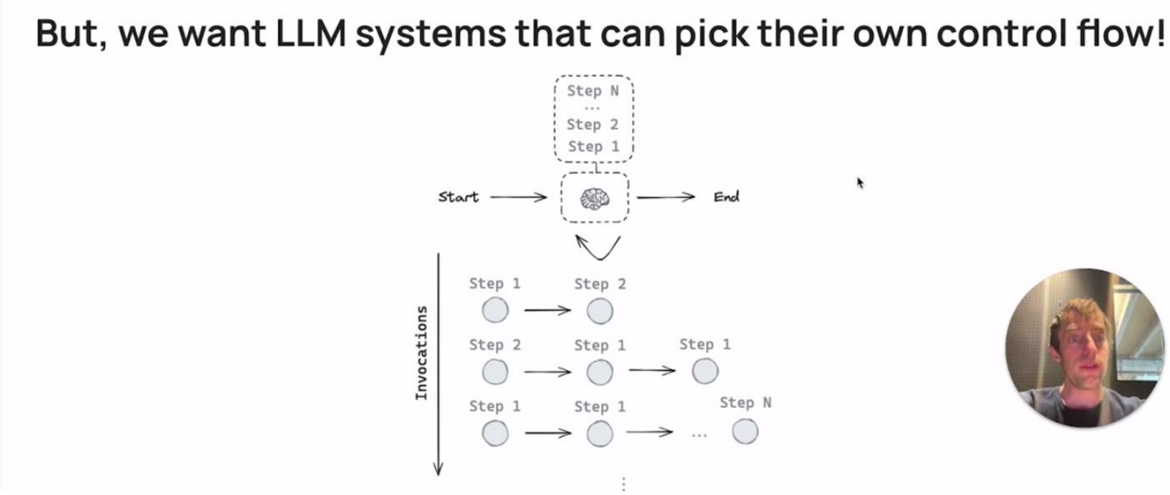
… with steps pre / post-LLM call (tool calls, retrieval, etc).



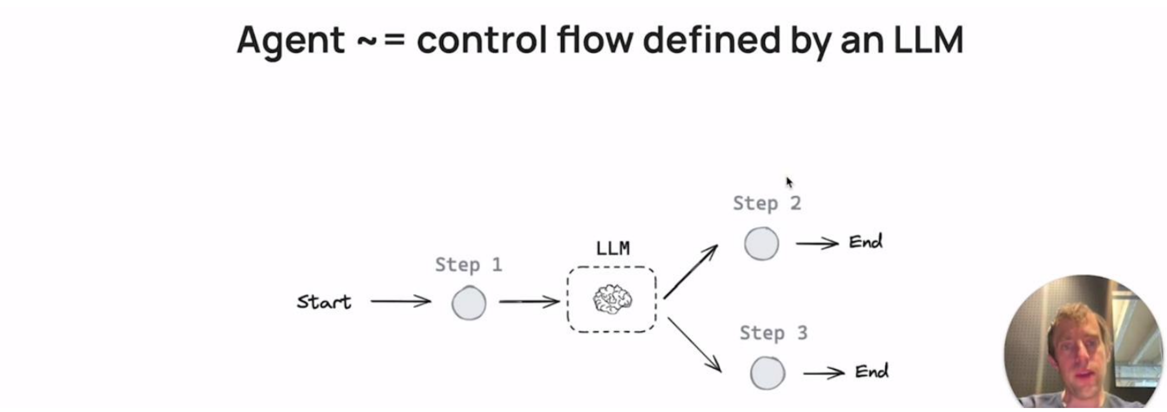
This control flow forms a "chain"



Chains are reliable! Same control flow every time.



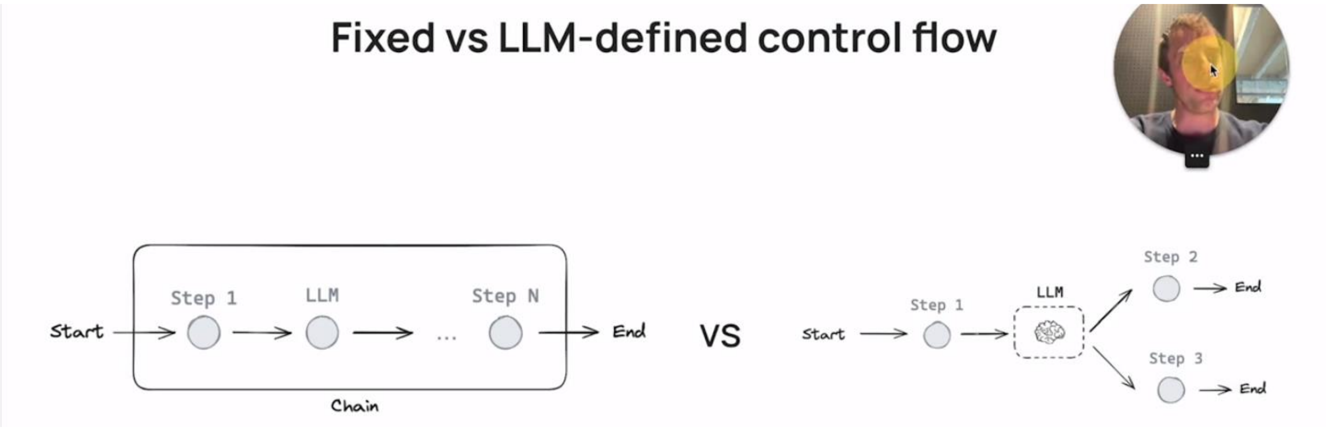
But, we want LLM systems that can pick their own control flow!



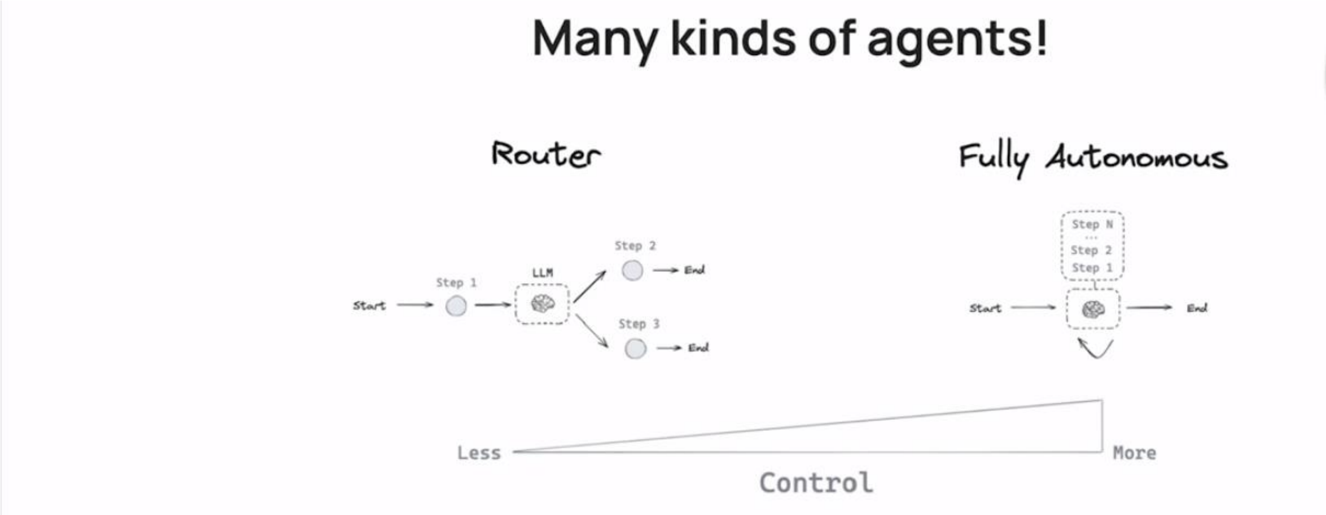
Agent ~= control flow defined by an LLM

https://blog.langchain.dev/what-is-an-agent/

7



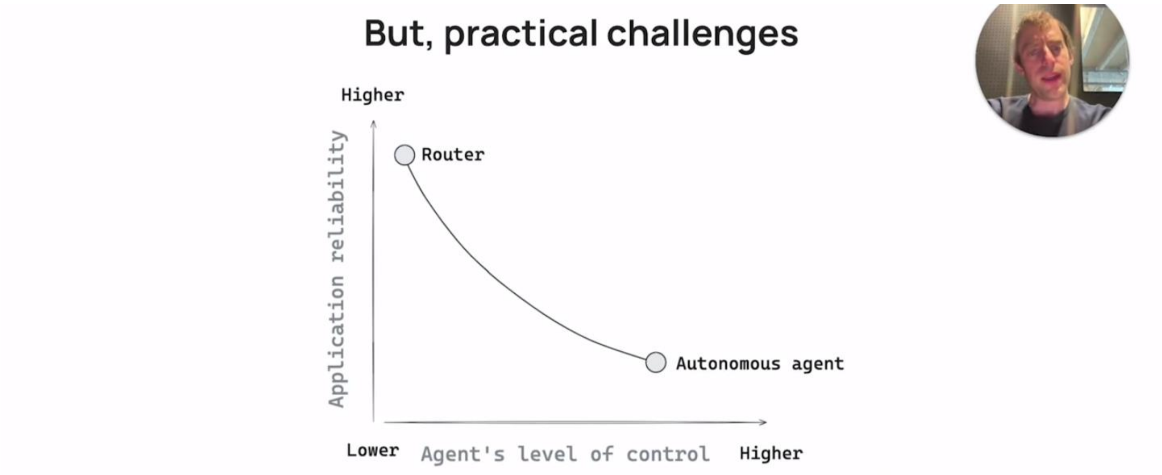
Fixed vs LLM-defined control flow



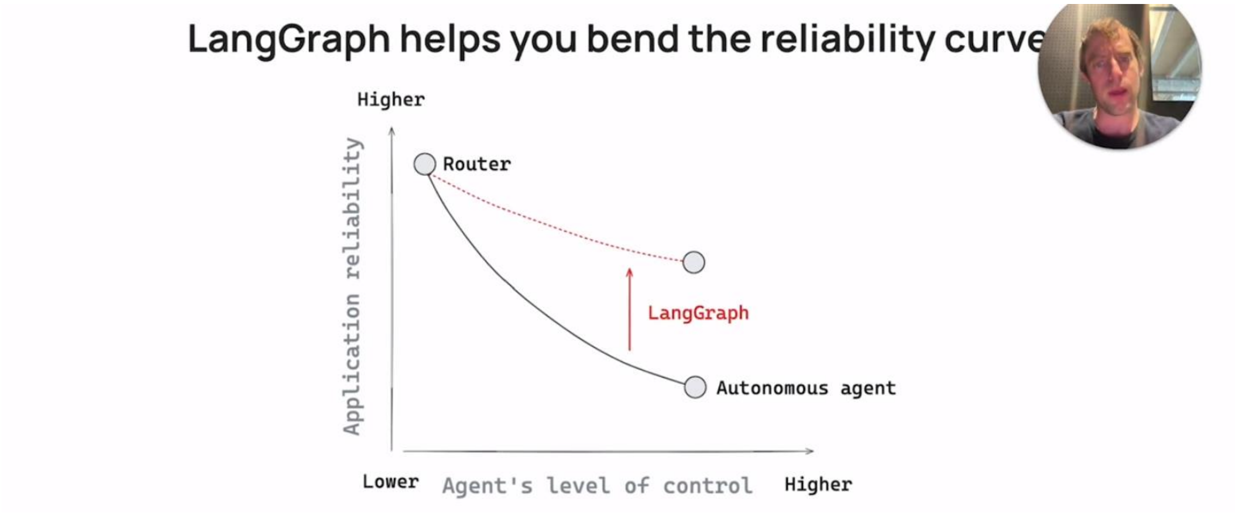
Many kinds of agents!

https://blog.langchain.dev/what-is-a-cognitive-architecture/

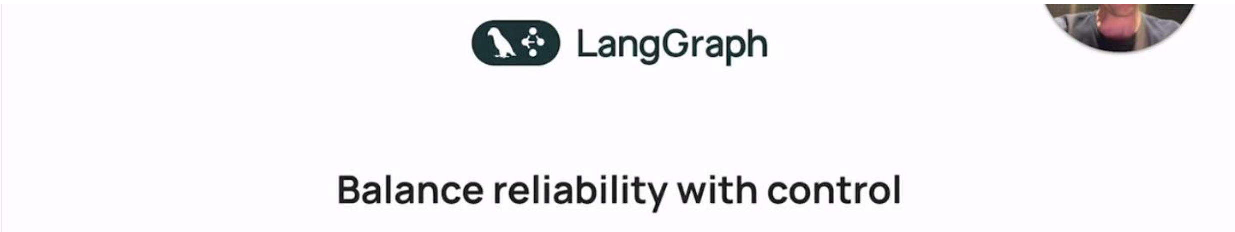
9



But, practical challenges

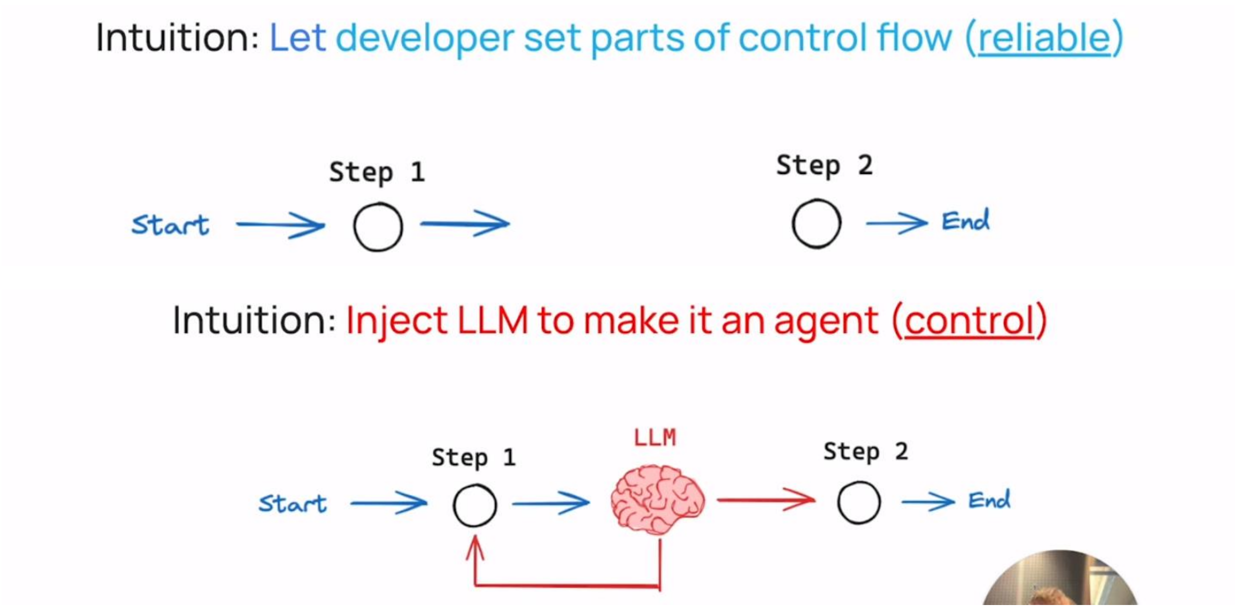


LangGraph helps you bend the reliability curve



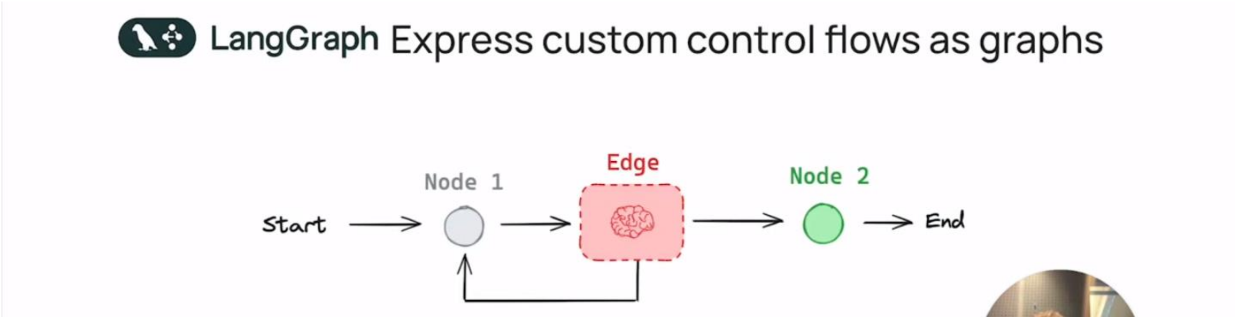
LangGraph

Balance reliability with control



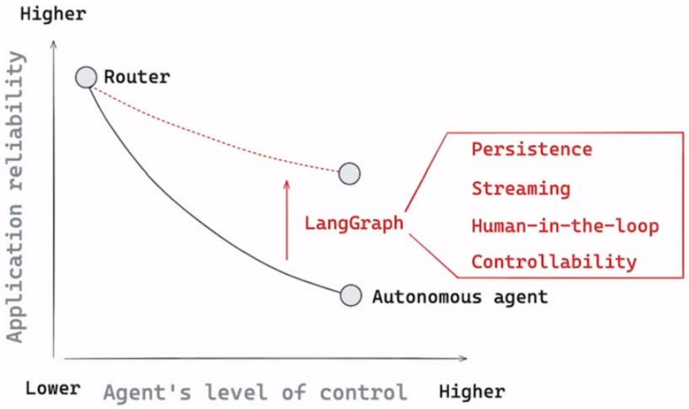
Intuition: Let developer set parts of control flow (reliable)

Intuition: Inject LLM to make it an agent (control)

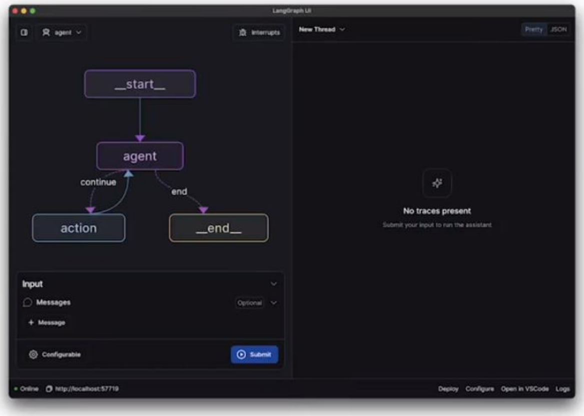


LangGraph Express custom control flows as graphs

Pillars

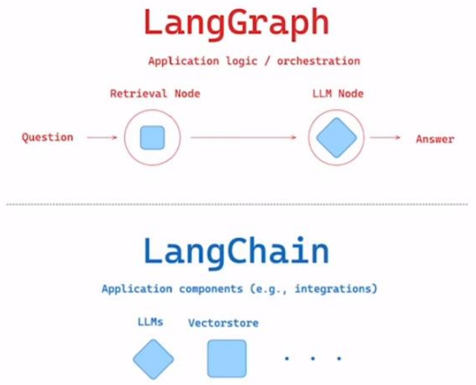


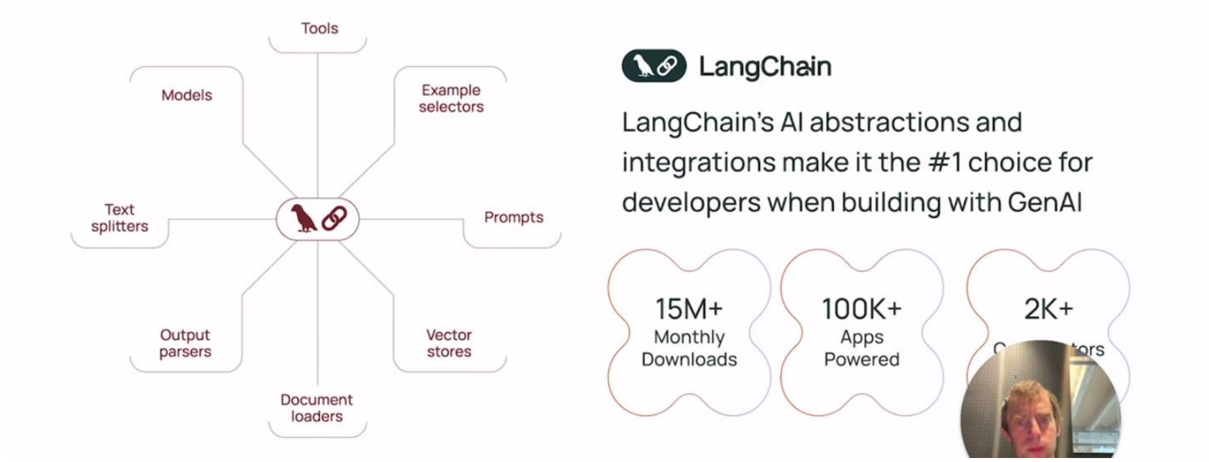
IDE: LangGraph Studio



https://github.com/langchain-ai/langgraph-studio

LangGraph 🤝 LangChain


LangGraph
Application logic / orchestration
Retrieval Node
LLM Node
Question
Answer
LangChain
Application components (e.g., integrations)
LLMs
Vectorstore
. . .



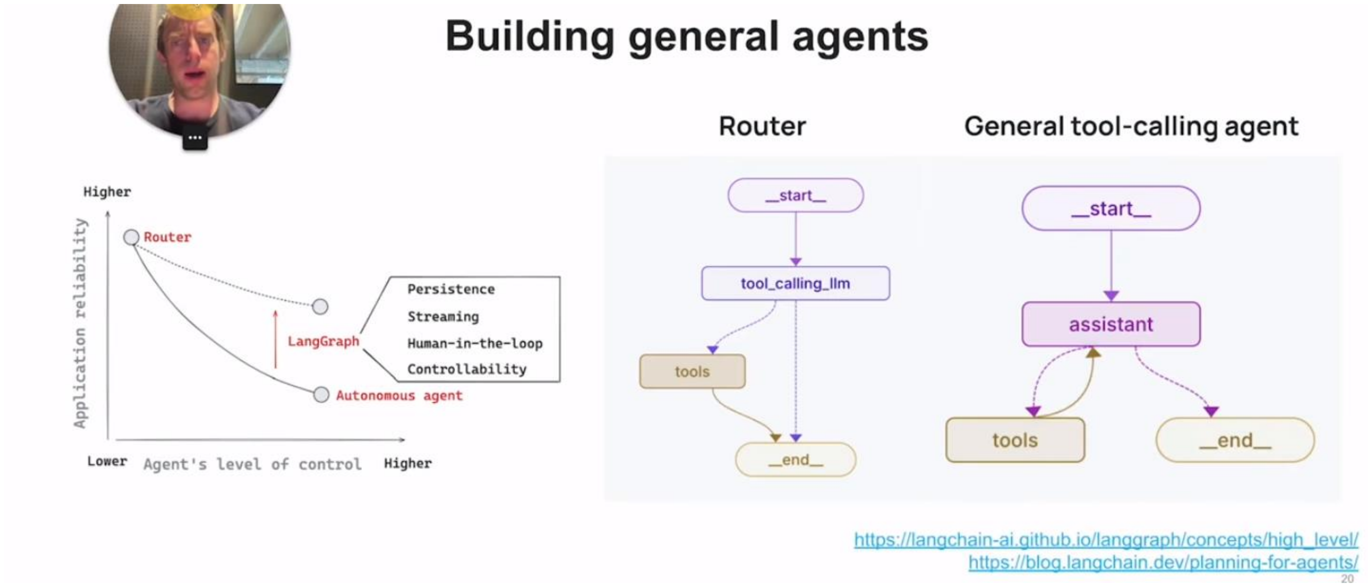
LangChain

LangChain's AI abstractions and integrations make it the #1 choice for developers when building with GenAI

15M+ Monthly Downloads

100K+ Apps Powered

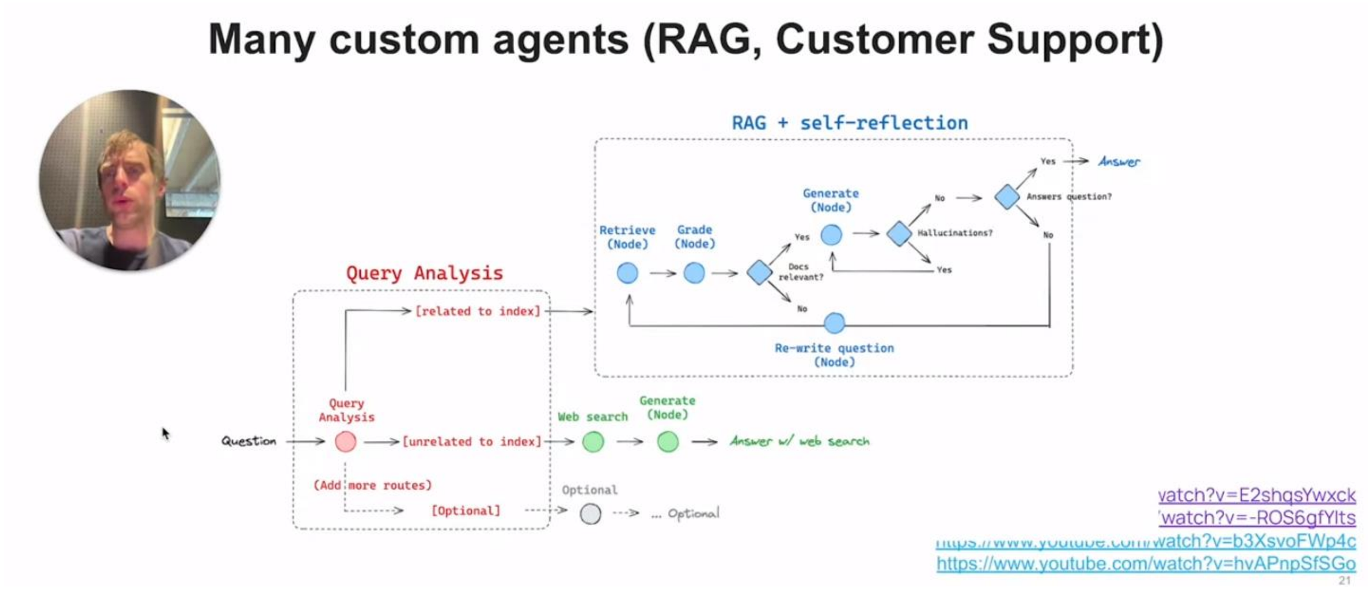
2K+ Contributors



Building general agents

https://langchain-ai.github.io/langgraph/concepts/high_level/
https://blog.langchain.dev/planning-for-agents/
20

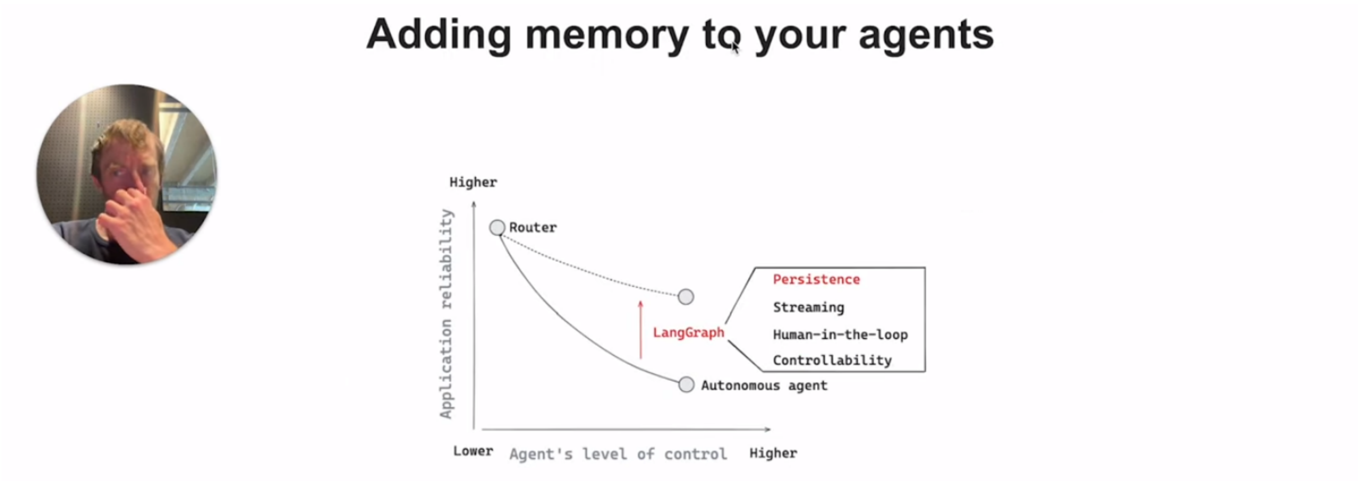
2 Custom agent architectures



Many custom agents (RAG, Customer Support)

vatch?v=E2shqsYwxck
'watch?v=-ROS6gfYlts
https://www.youtube.com/watch?v=b3XsvoFWp4c
https://www.youtube.com/watch?v=hvAPnpSfSGo
21

3 Memory



Adding memory to your agents

Higher

Application reliability

Router

Persistence

Streaming

Human-in-the-loop

Controllability

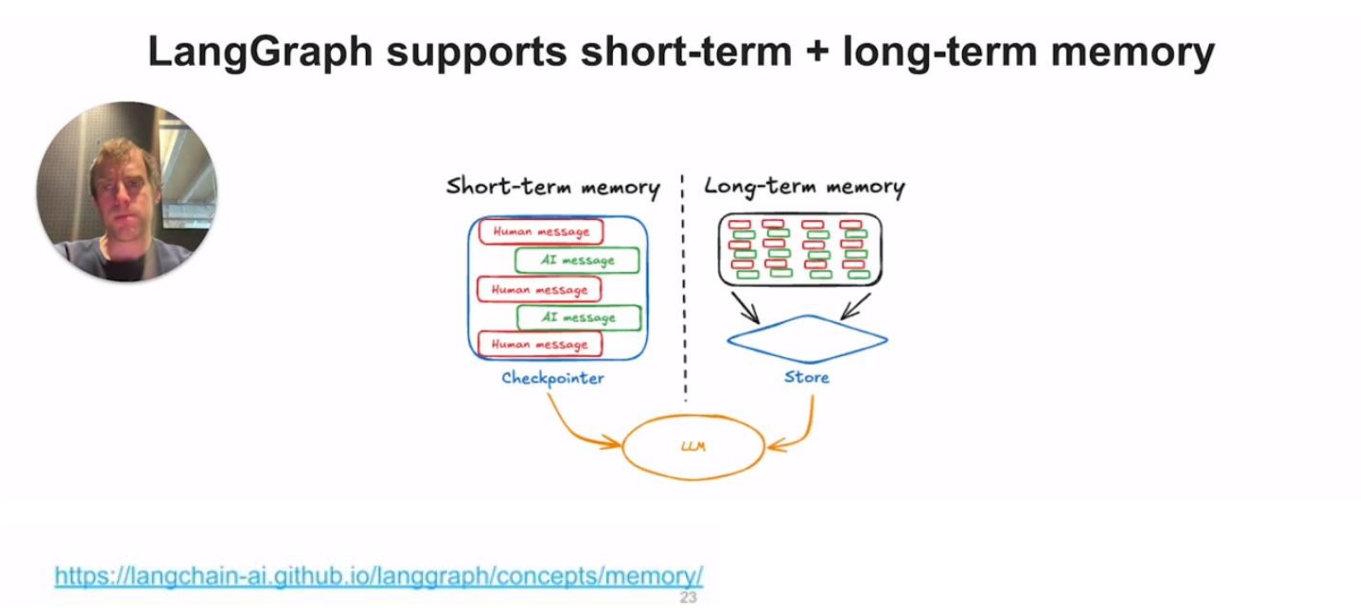
LangGraph

Autonomous agent

Lower

Agent's level of control

Higher



LangGraph supports short-term + long-term memory

Short-term memory

Long-term memory

Human message

AI message

Human message

AI message

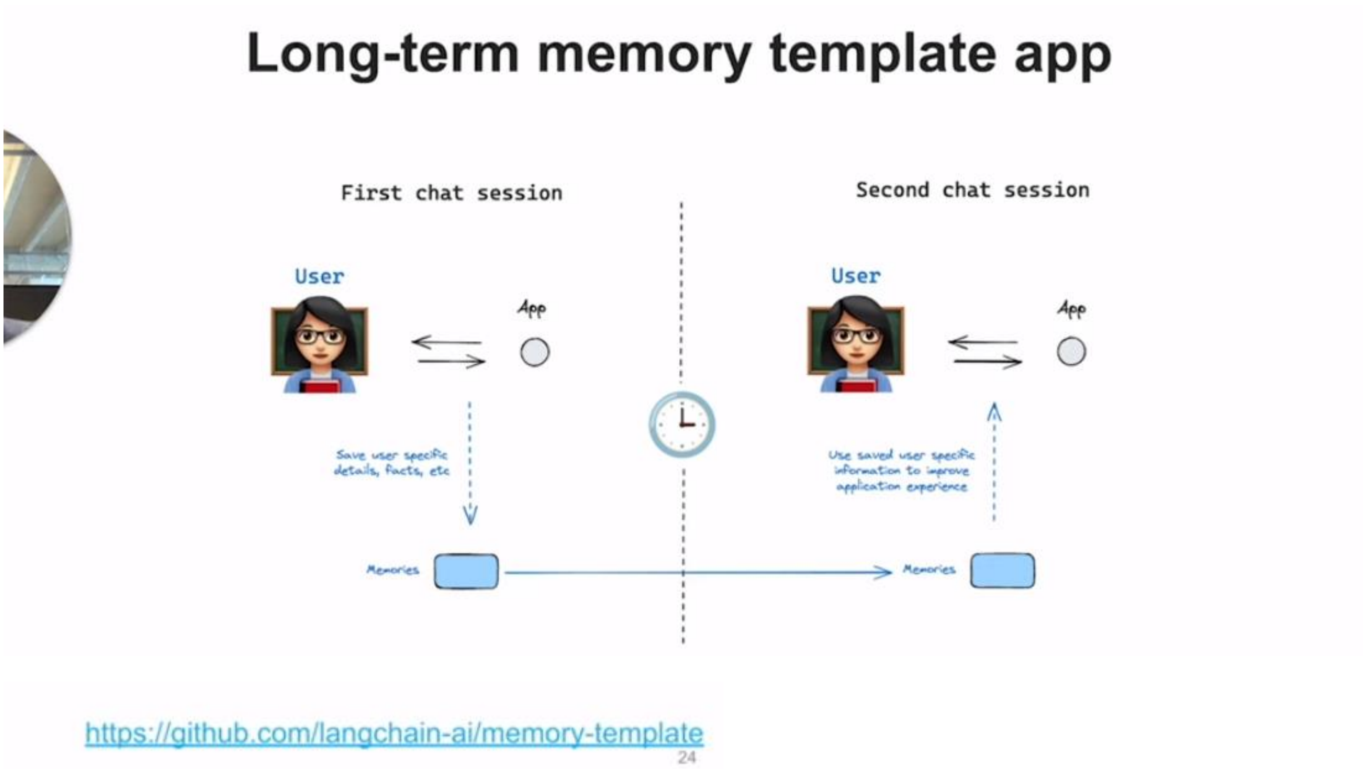
Human message

Checkpointer

Store

LLM

https://langchain-ai.github.io/langgraph/concepts/memory/



Long-term memory template app

First chat session

Second chat session

User

App

User

App

Save user specific details, facts, etc

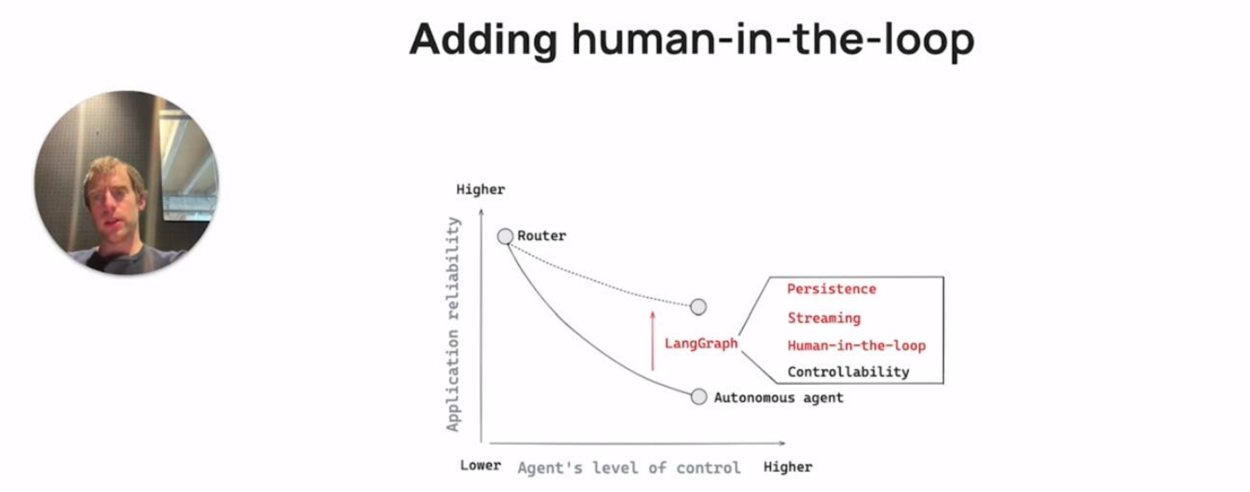
Use saved user specific information to improve application experience

Memories

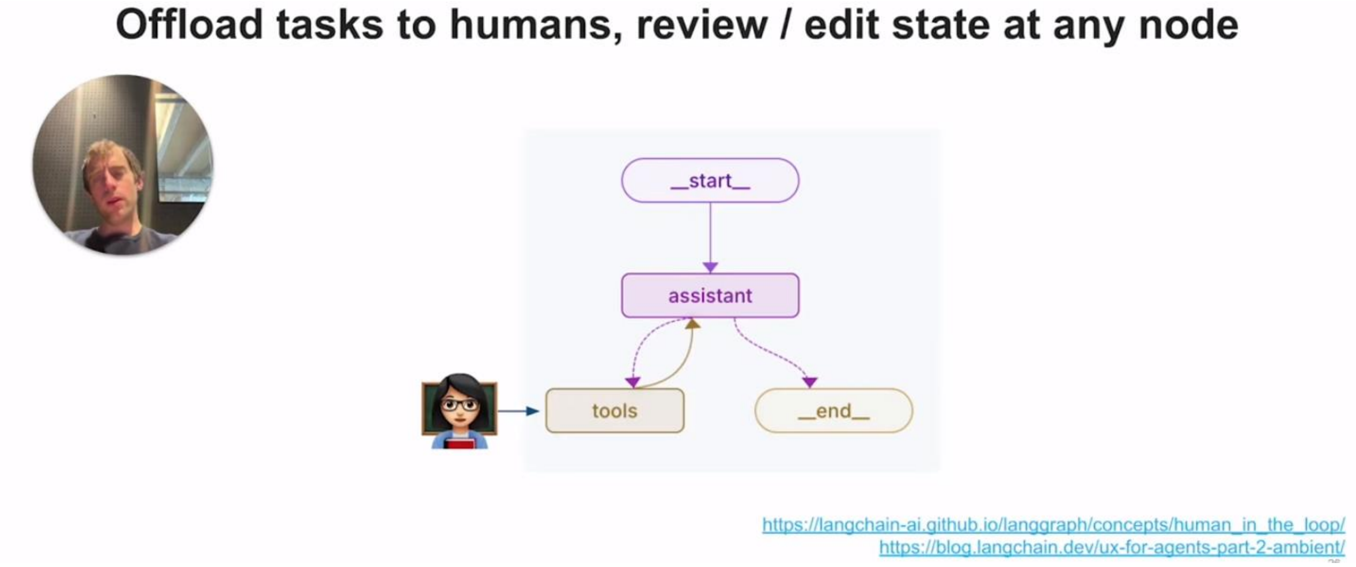
Memories

https://github.com/langchain-ai/memory-template

4 Human-in-the-loop



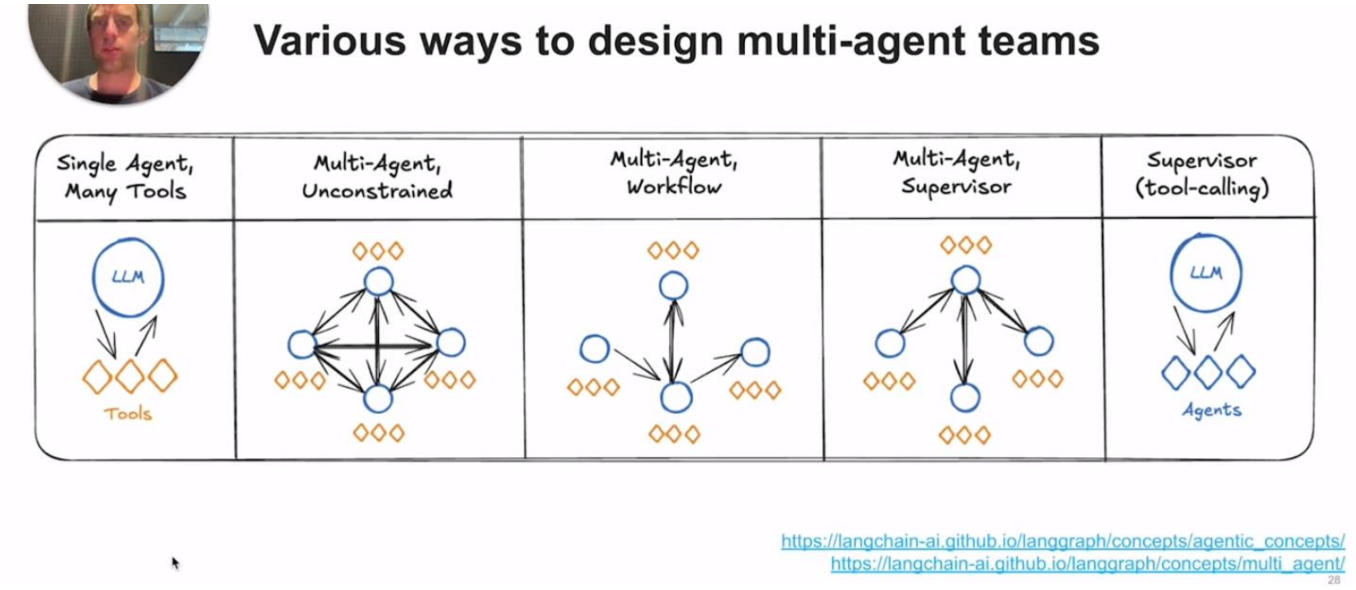
Adding human-in-the-loop



Offload tasks to humans, review / edit state at any node

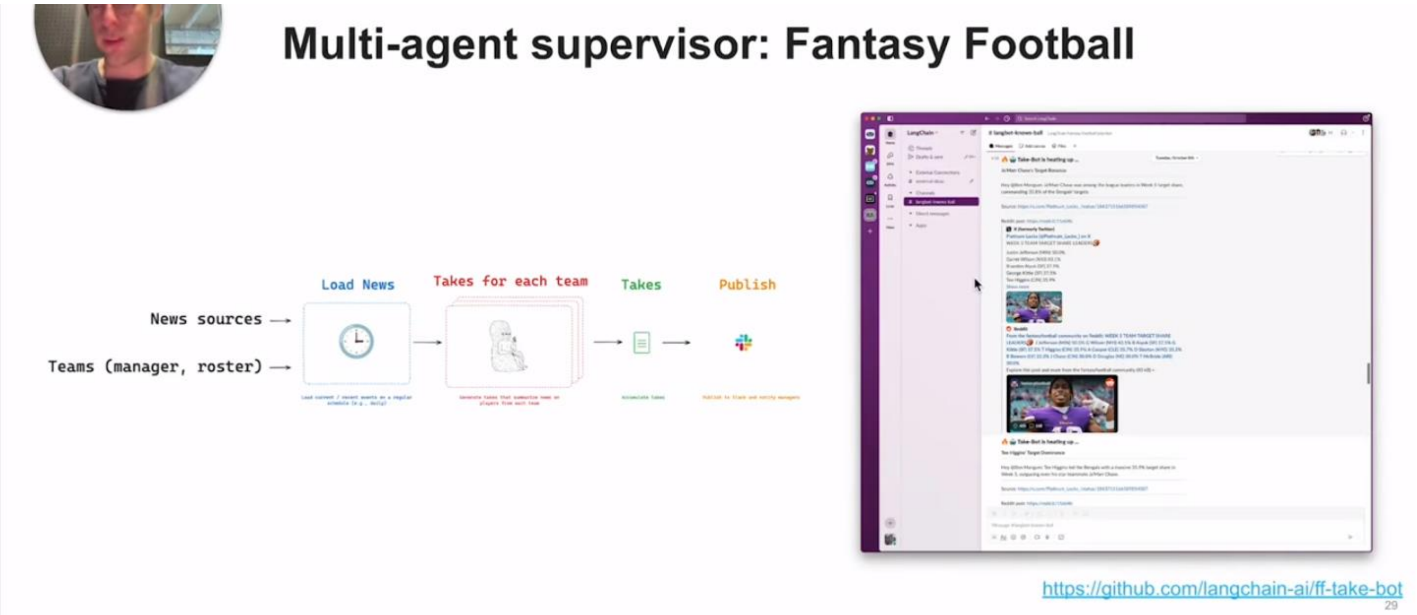
https://langchain-ai.github.io/langgraph/concepts/human_in_the_loop/
https://blog.langchain.dev/ux-for-agents-part-2-ambient/

5 Controllability



Various ways to design multi-agent teams

https://langchain-ai.github.io/langgraph/concepts/agentic_concepts/
https://langchain-ai.github.io/langgraph/concepts/multi_agent/

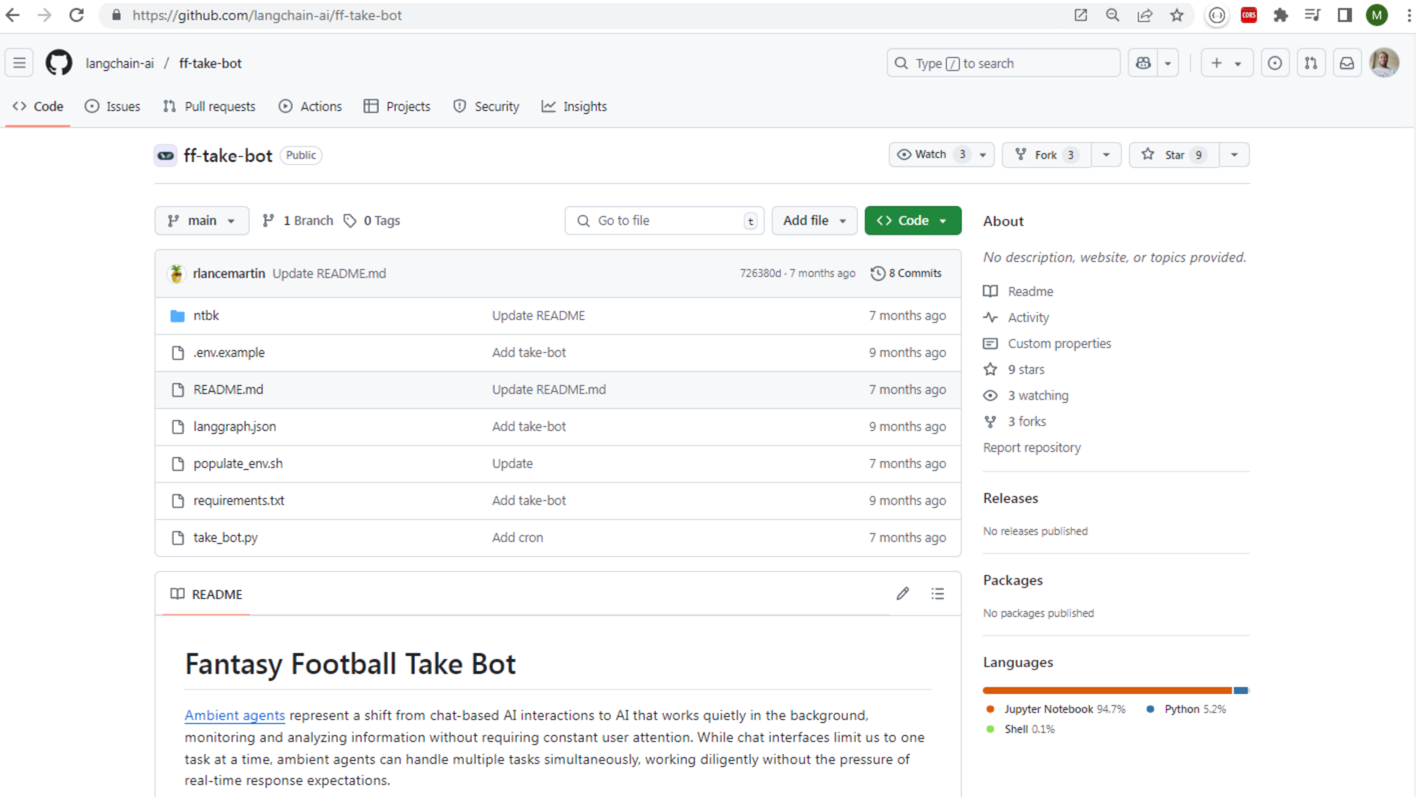


Multi-agent supervisor: Fantasy Football

News sources →
Teams (manager, roster) →
Load News
Load current / recent events on a regular schedule (e.g., daily)
Takes for each team
Generate takes that summarize news on players from each team
Takes
Accumulate takes
Publish
Publish to Slack and notify managers

https://github.com/langchain-ai/ff-take-bot

29



ff-take-bot Public

rlancemartin Update README.md 726380d · 7 months ago 8 Commits

ntbk	Update README	7 months ago
.env.example	Add take-bot	9 months ago
README.md	Update README.md	7 months ago
langgraph.json	Add take-bot	9 months ago
populate_env.sh	Update	7 months ago
requirements.txt	Add take-bot	9 months ago
take_bot.py	Add cron	7 months ago

About

No description, website, or topics provided.

Readme
Activity
Custom properties
9 stars
3 watching
3 forks
Report repository

Releases

No releases published

Packages

No packages published

Languages

Jupyter Notebook 94.7% Python 5.2%
Shell 0.1%

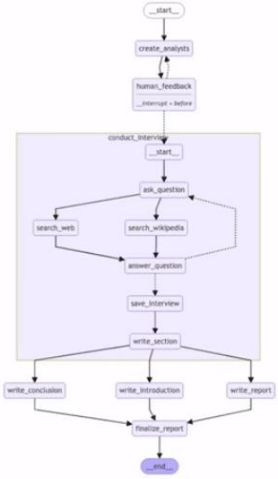
README

Fantasy Football Take Bot

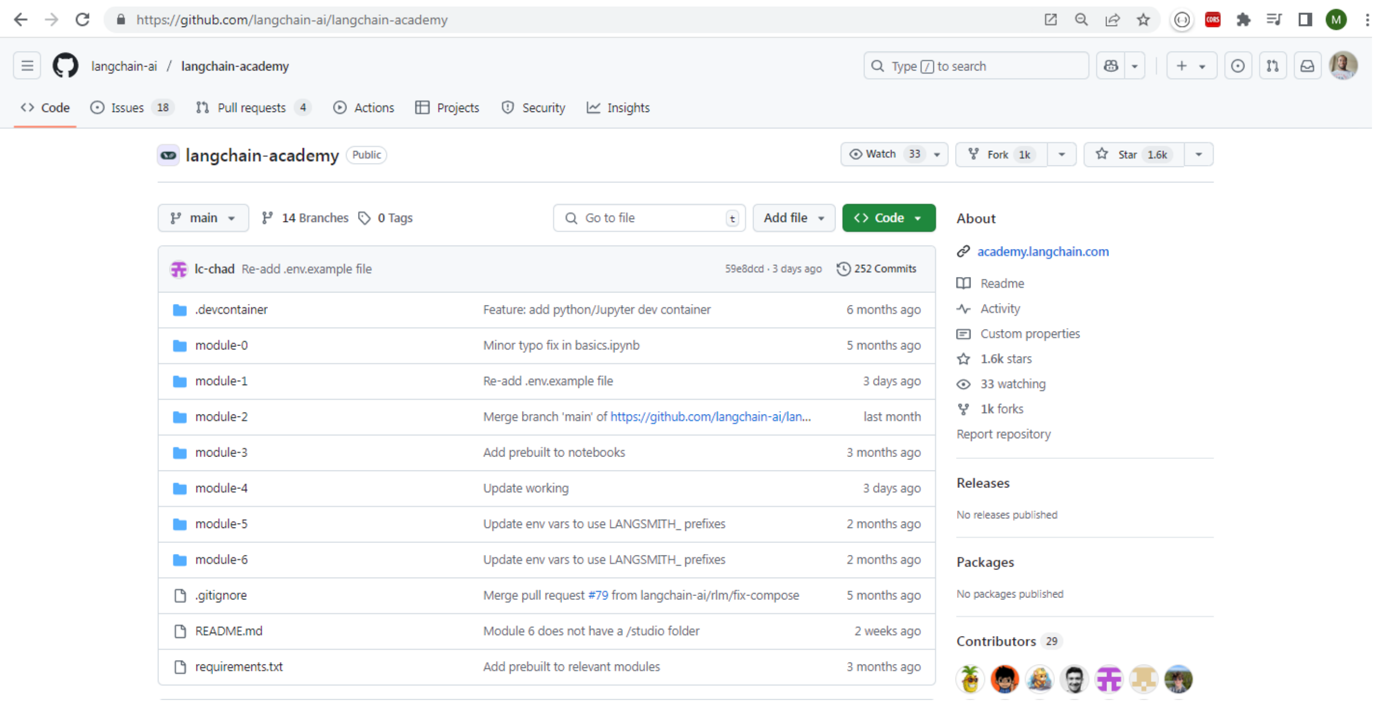
Ambient agents represent a shift from chat-based AI interactions to AI that works quietly in the background, monitoring and analyzing information without requiring constant user attention. While chat interfaces limit us to one task at a time, ambient agents can handle multiple tasks simultaneously, working diligently without the pressure of real-time response expectations.

6 Use-cases

Multi-agent supervisor: Research assistant



https://github.com/langchain-ai/langchain-academy/blob/main/module-4/research-assistant.ipynb



langchain-academy Public

14 Branches 0 Tags

lc-chad Re-add .env.example file 59e8dcd · 3 days ago 252 Commits

.devcontainer	Feature: add python/Jupyter dev container	6 months ago
module-0	Minor typo fix in basics.ipynb	5 months ago
module-1	Re-add .env.example file	3 days ago
module-2	Merge branch 'main' of https://github.com/langchain-ai/lan...	last month
module-3	Add prebuilt to notebooks	3 months ago
module-4	Update working	3 days ago
module-5	Update env vars to use LANGSMITH_ prefixes	2 months ago
module-6	Update env vars to use LANGSMITH_ prefixes	2 months ago
.gitignore	Merge pull request #79 from langchain-ai/rlm/fix-compose	5 months ago
README.md	Module 6 does not have a /studio folder	2 weeks ago
requirements.txt	Add prebuilt to relevant modules	3 months ago

About

academy.langchain.com

- Readme
- Activity
- Custom properties
- 1.6k stars
- 33 watching
- 1k forks

Report repository

Releases

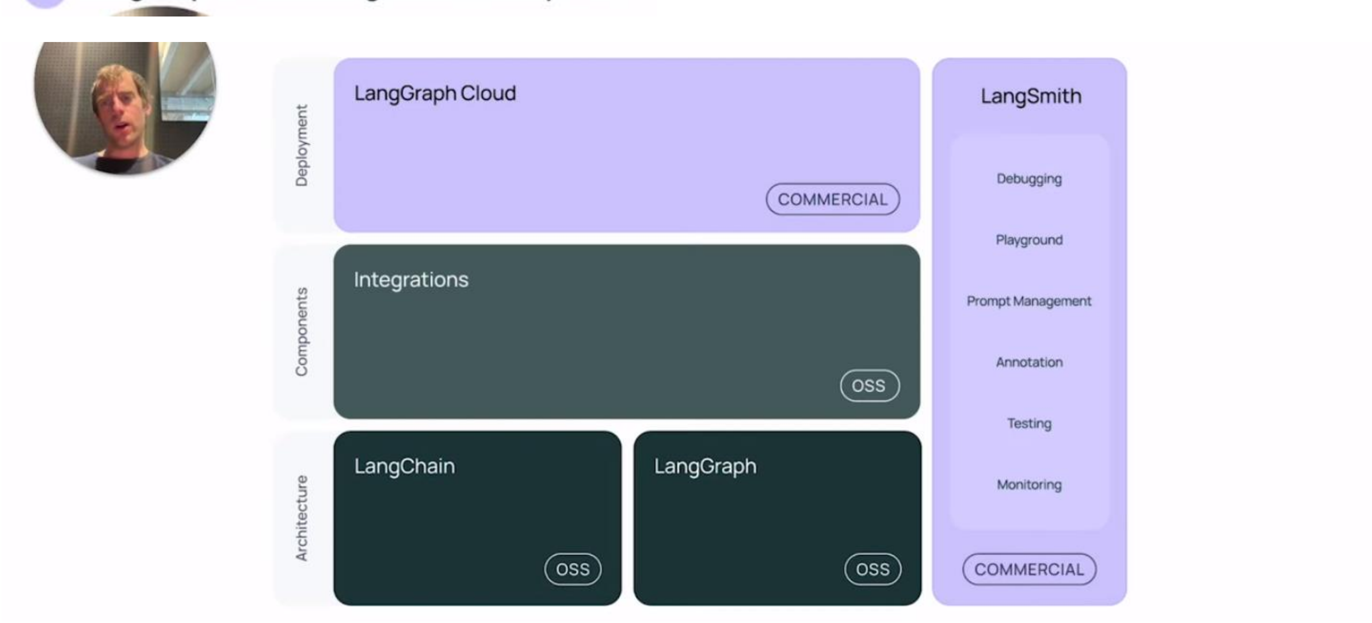
No releases published

Packages

No packages published

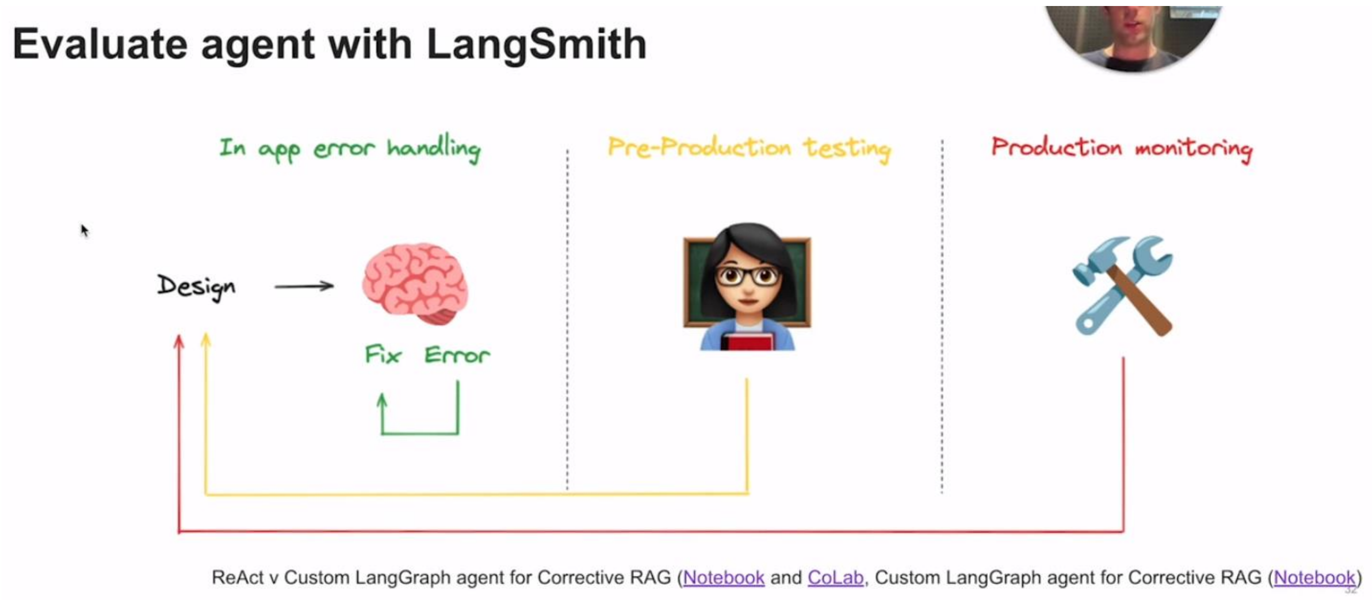
Contributors 29

7 LangGraph within LangChain's Ecosystem



Deployment	LangGraph Cloud COMMERCIAL	LangSmith Debugging Playground Prompt Management Annotation Testing Monitoring COMMERCIAL
Components	Integrations OSS	
Architecture	LangChain OSS LangGraph OSS	

8 LangGraph + LangSmith

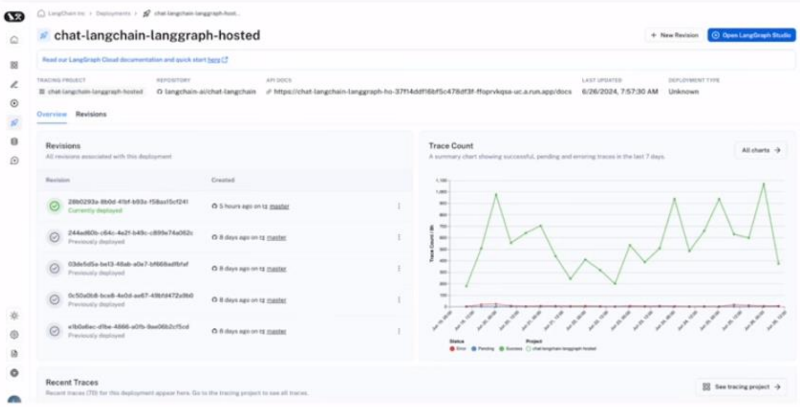


Evaluate agent with LangSmith

ReAct v Custom LangGraph agent for Corrective RAG (Notebook and CoLab, Custom LangGraph agent for Corrective RAG (Notebook)

9 Deployment

Deploy agents with LangGraph Cloud



LangChain Inc › Deployments › chat-langchain-langgraph-host...

chat-langchain-langgraph-hosted

Read our LangGraph Cloud documentation and quick start here

TRACING PROJECT	REPOSITORY	API DOCS	LAST UPDATED	DEPLOYMENT TYPE
chat-langchain-langgraph-hosted	langchain-ai/chat-langchain	https://chat-langchain-langgraph-ho-37f14ddf16bf5c478df3f-ffoprvkqsa-uc.a.run.app/docs	6/26/2024, 7:57:30 AM	Unknown

Revisions

All revisions associated with this deployment

Revision	Created
28b0293a-8b0d-41bf-b93a-f58aa15cf241 Currently deployed	5 hours ago on master
244ad60b-c64c-4e2f-b49c-c899e74a062c Previously deployed	8 days ago on master
03de5d5a-be13-48ab-a0e7-bf668adfbfaf Previously deployed	8 days ago on master
0c50a0b8-bce8-4e0d-ae67-49bfd472a9b0 Previously deployed	8 days ago on master
e1b0a6ec-d1be-4866-a0fb-8ae06b2cf5cd Previously deployed	8 days ago on master

Trace Count

A summary chart showing successful, pending and erroring traces in the last 7 days.

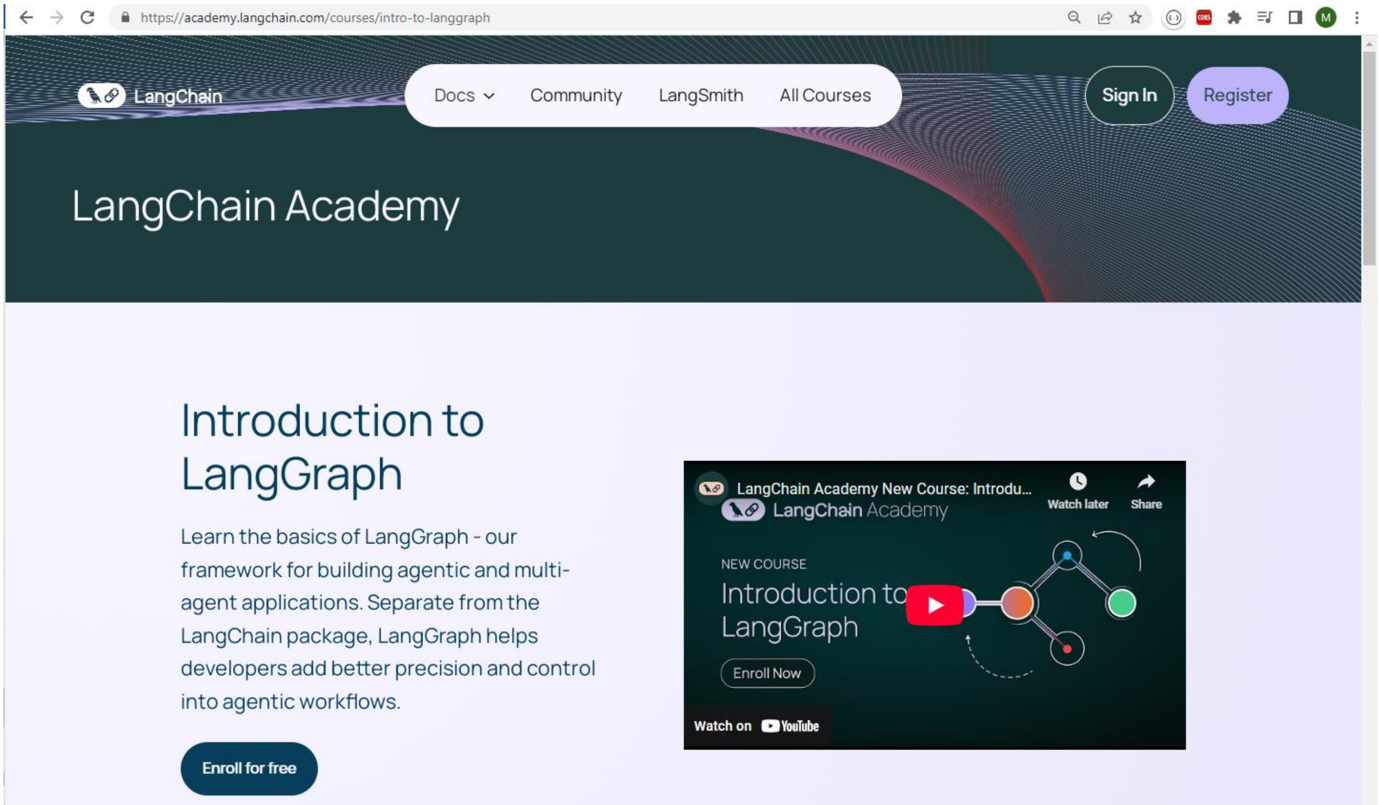
Recent Traces

Recent traces (70) for this deployment appear here. Go to the tracing project to see all traces.

Reference: https://blog.langchain.dev/langgraph-cloud/

Free Course!

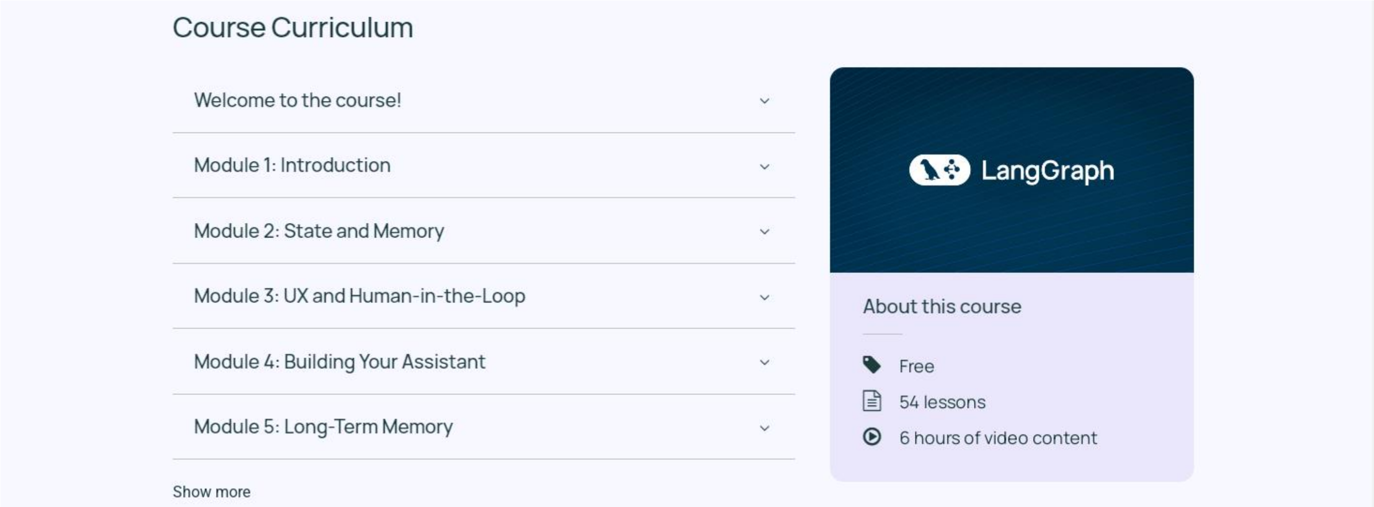
https://academy.langchain.com/courses/intro-to-langgraph



LangChain Academy

Introduction to LangGraph

Learn the basics of LangGraph - our framework for building agentic and multi-agent applications. Separate from the LangChain package, LangGraph helps developers add better precision and control into agentic workflows.



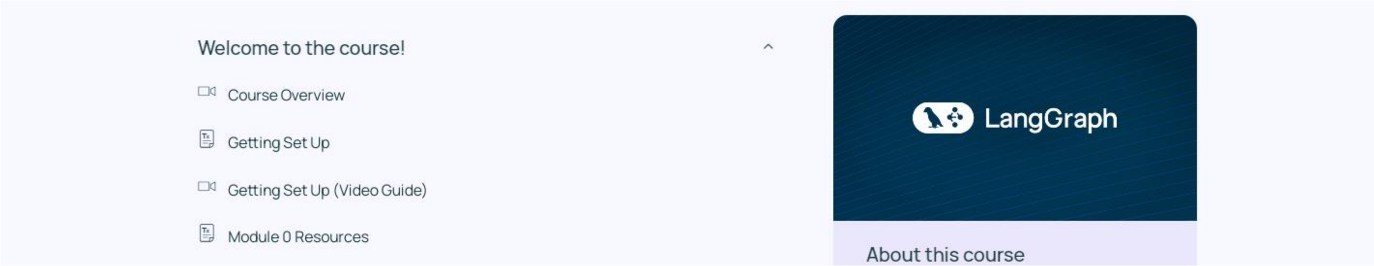
Course Curriculum

- Welcome to the course!
- Module 1: Introduction
- Module 2: State and Memory
- Module 3: UX and Human-in-the-Loop
- Module 4: Building Your Assistant
- Module 5: Long-Term Memory

Show more

About this course

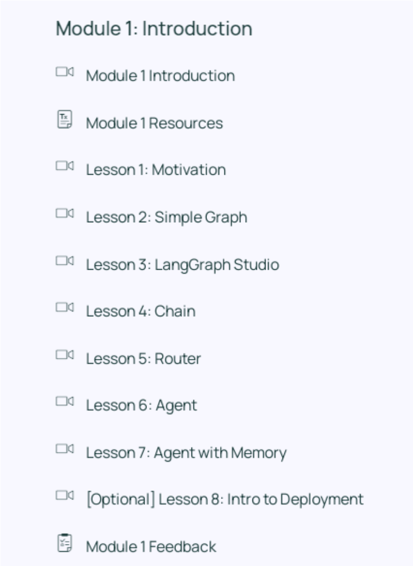
- Free
- 54 lessons
- 6 hours of video content



Welcome to the course!

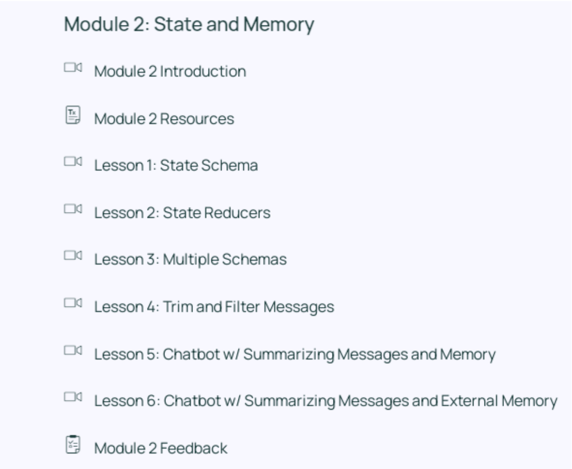
- Course Overview
- Getting Set Up
- Getting Set Up (Video Guide)
- Module 0 Resources

About this course



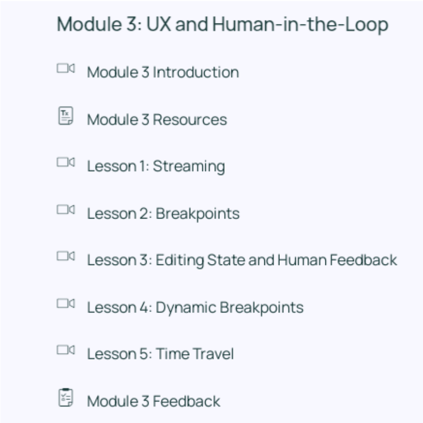
Module 1: Introduction

- Module 1 Introduction
- Module 1 Resources
- Lesson 1: Motivation
- Lesson 2: Simple Graph
- Lesson 3: LangGraph Studio
- Lesson 4: Chain
- Lesson 5: Router
- Lesson 6: Agent
- Lesson 7: Agent with Memory
- [Optional] Lesson 8: Intro to Deployment
- Module 1 Feedback



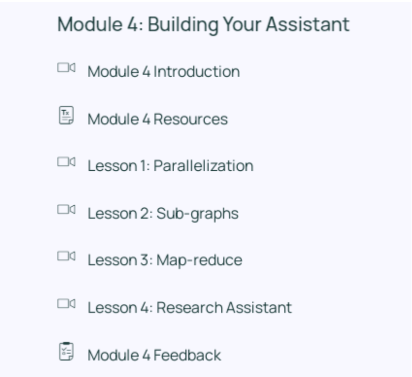
Module 2: State and Memory

- Module 2 Introduction
- Module 2 Resources
- Lesson 1: State Schema
- Lesson 2: State Reducers
- Lesson 3: Multiple Schemas
- Lesson 4: Trim and Filter Messages
- Lesson 5: Chatbot w/ Summarizing Messages and Memory
- Lesson 6: Chatbot w/ Summarizing Messages and External Memory
- Module 2 Feedback



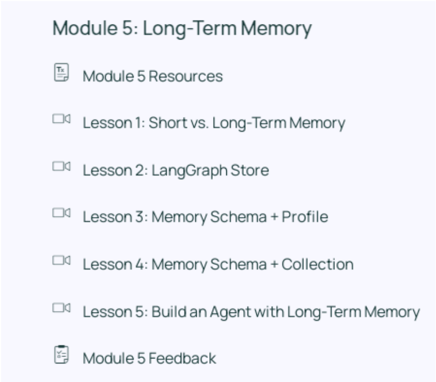
Module 3: UX and Human-in-the-Loop

- Module 3 Introduction
- Module 3 Resources
- Lesson 1: Streaming
- Lesson 2: Breakpoints
- Lesson 3: Editing State and Human Feedback
- Lesson 4: Dynamic Breakpoints
- Lesson 5: Time Travel
- Module 3 Feedback



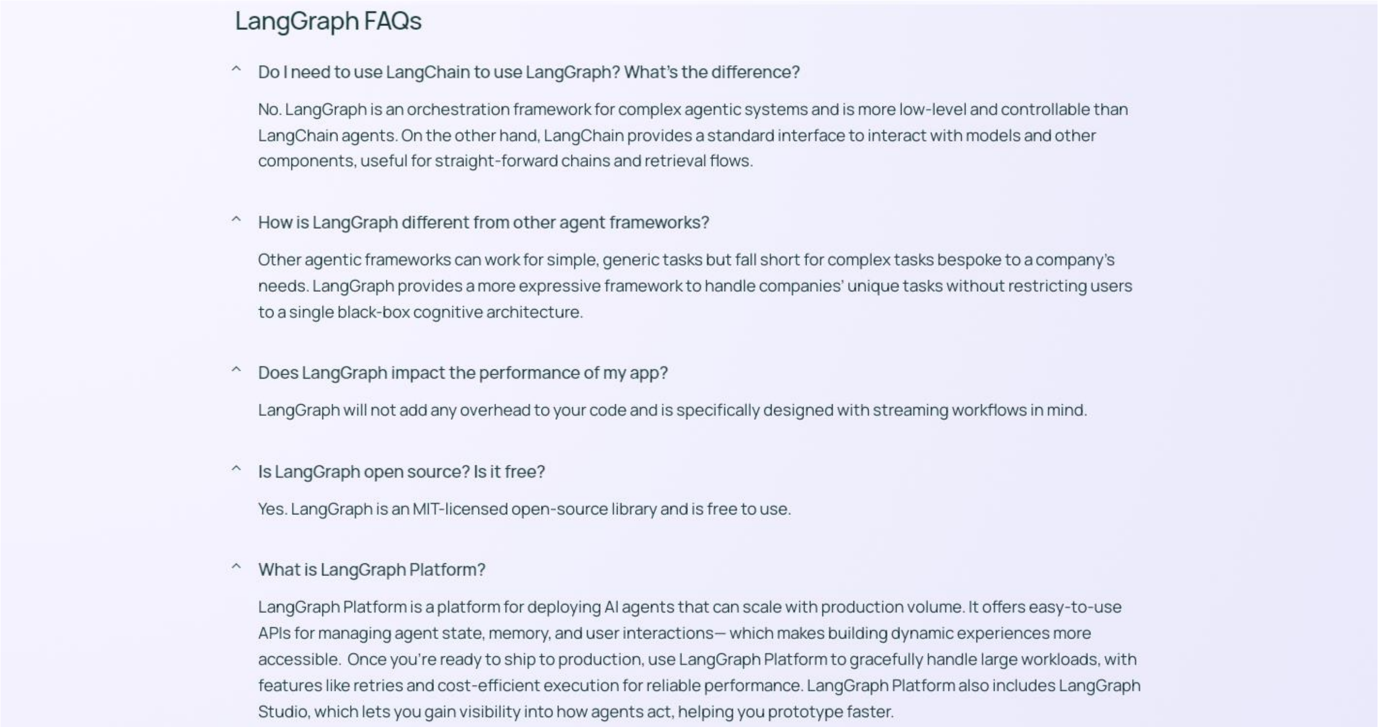
Module 4: Building Your Assistant

- Module 4 Introduction
- Module 4 Resources
- Lesson 1: Parallelization
- Lesson 2: Sub-graphs
- Lesson 3: Map-reduce
- Lesson 4: Research Assistant
- Module 4 Feedback



Module 5: Long-Term Memory

- Module 5 Resources
- Lesson 1: Short vs. Long-Term Memory
- Lesson 2: LangGraph Store
- Lesson 3: Memory Schema + Profile
- Lesson 4: Memory Schema + Collection
- Lesson 5: Build an Agent with Long-Term Memory
- Module 5 Feedback



LangGraph FAQs

^ Do I need to use LangChain to use LangGraph? What's the difference?

No. LangGraph is an orchestration framework for complex agentic systems and is more low-level and controllable than LangChain agents. On the other hand, LangChain provides a standard interface to interact with models and other components, useful for straight-forward chains and retrieval flows.

^ How is LangGraph different from other agent frameworks?

Other agentic frameworks can work for simple, generic tasks but fall short for complex tasks bespoke to a company's needs. LangGraph provides a more expressive framework to handle companies' unique tasks without restricting users to a single black-box cognitive architecture.

^ Does LangGraph impact the performance of my app?

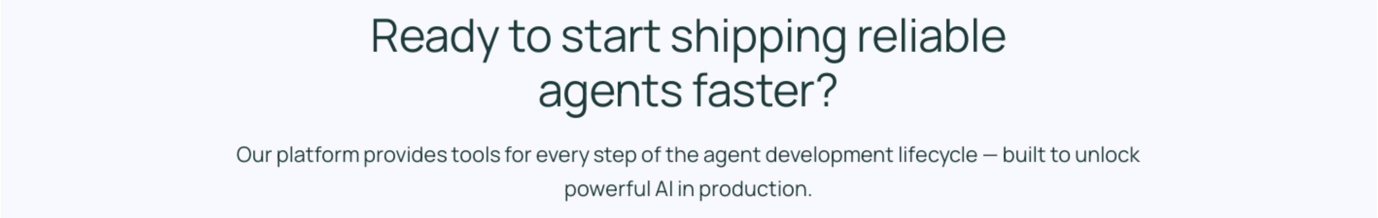
LangGraph will not add any overhead to your code and is specifically designed with streaming workflows in mind.

^ Is LangGraph open source? Is it free?

Yes. LangGraph is an MIT-licensed open-source library and is free to use.

^ What is LangGraph Platform?

LangGraph Platform is a platform for deploying AI agents that can scale with production volume. It offers easy-to-use APIs for managing agent state, memory, and user interactions— which makes building dynamic experiences more accessible. Once you're ready to ship to production, use LangGraph Platform to gracefully handle large workloads, with features like retries and cost-efficient execution for reliable performance. LangGraph Platform also includes LangGraph Studio, which lets you gain visibility into how agents act, helping you prototype faster.

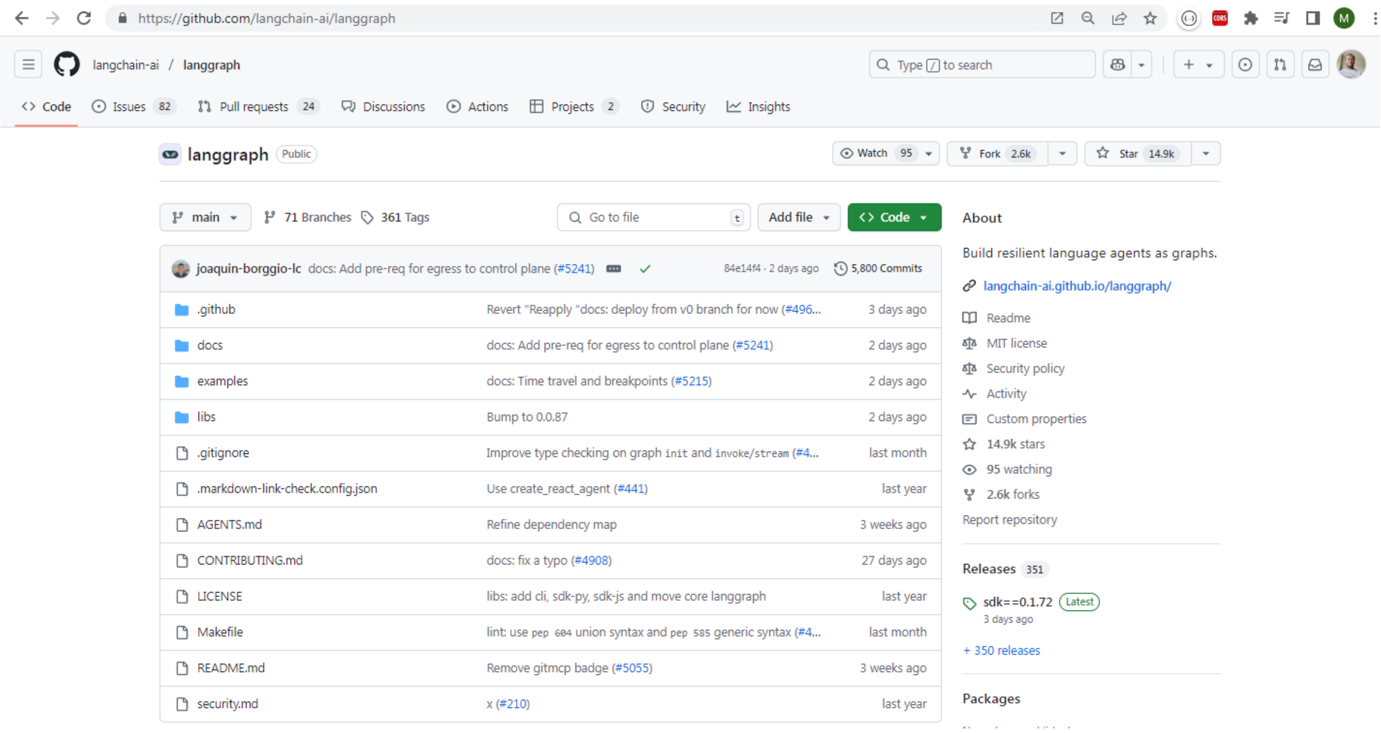


Ready to start shipping reliable agents faster?

Our platform provides tools for every step of the agent development lifecycle — built to unlock powerful AI in production.

Bring LangChain Academy to Your Company

Nominate your team for a hands-on training—
available in-person at your office or remotely, free of charge.



joaquin-borggio-lc docs: Add pre-req for egress to control plane (#5241) ✓ 84e14f4 · 2 days ago 5,800 Commits

.github	Revert "Reapply "docs: deploy from v0 branch for now (#496...	3 days ago
docs	docs: Add pre-req for egress to control plane (#5241)	2 days ago
examples	docs: Time travel and breakpoints (#5215)	2 days ago
libs	Bump to 0.0.87	2 days ago
.gitignore	Improve type checking on graph `init` and `invoke/stream` (#4...	last month
.markdown-link-check.config.json	Use create_react_agent (#441)	last year
AGENTS.md	Refine dependency map	3 weeks ago
CONTRIBUTING.md	docs: fix a typo (#4908)	27 days ago
LICENSE	libs: add cli, sdk-py, sdk-js and move core langgraph	last year
Makefile	lint: use `pep 604` union syntax and `pep 585` generic syntax (#4...	last month
README.md	Remove gitmcp badge (#5055)	3 weeks ago
security.md	x (#210)	last year

About

Build resilient language agents as graphs.

langchain-ai.github.io/langgraph/

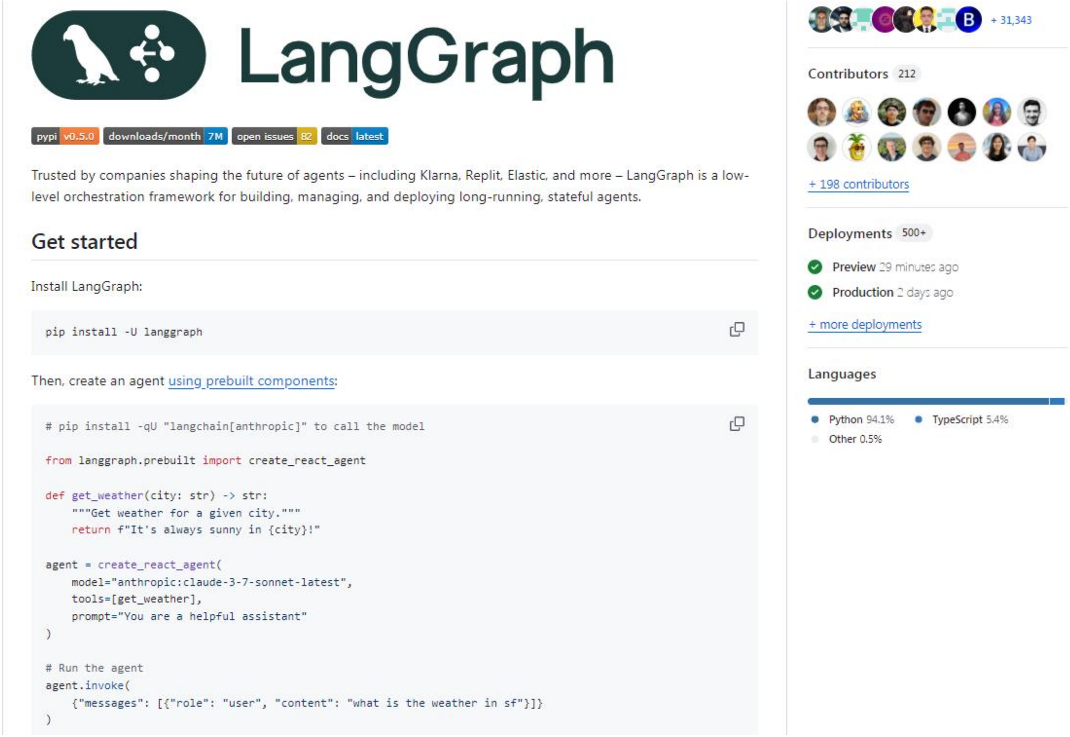
Readme
MIT license
Security policy
Activity
Custom properties
14.9k stars
95 watching
2.6k forks
Report repository

Releases 351

sdk==0.1.72 Latest
3 days ago

+ 350 releases

Packages



LangGraph

pypi v0.5.0 downloads/month 7M open issues 82 docs latest

Trusted by companies shaping the future of agents – including Klarna, Replit, Elastic, and more – LangGraph is a low-level orchestration framework for building, managing, and deploying long-running, stateful agents.

Get started

Install LangGraph:

```
pip install -U langgraph
```

Then, create an agent using prebuilt components:

```
# pip install -qU "langchain[anthropic]" to call the model

from langgraph.prebuilt import create_react_agent

def get_weather(city: str) -> str:
    """Get weather for a given city."""
    return f"It's always sunny in {city}!"

agent = create_react_agent(
    model="anthropic:claude-3-7-sonnet-latest",
    tools=[get_weather],
    prompt="You are a helpful assistant"
)

# Run the agent
agent.invoke(
    {"messages": [{"role": "user", "content": "what is the weather in sf"}]}
)
```

+ 31,343

Contributors 212

+ 198 contributors

Deployments 500+

✓ Preview 29 minutes ago

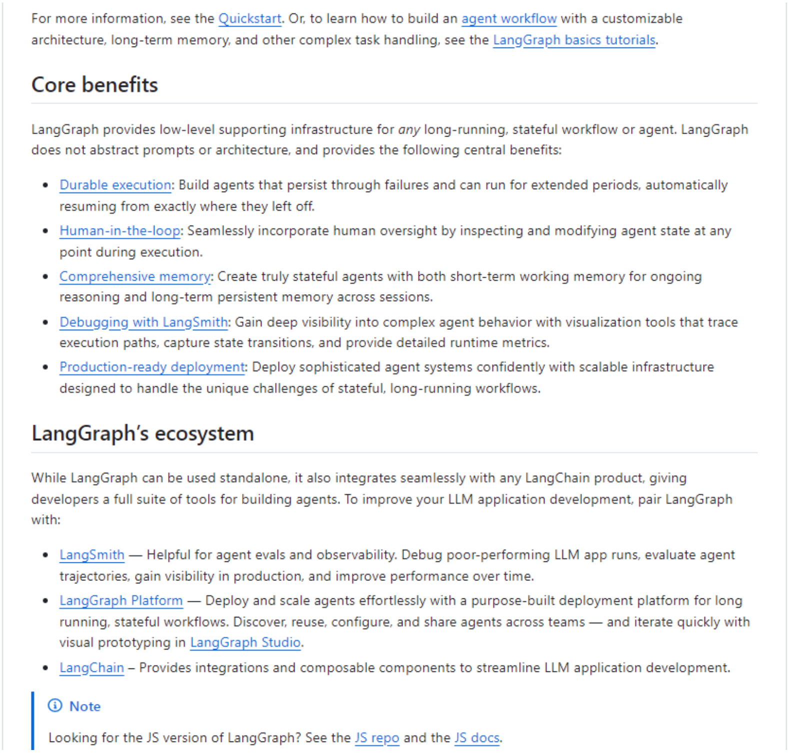
✓ Production 2 days ago

+ more deployments

Languages

● Python 94.1% ● TypeScript 5.4%

● Other 0.5%



For more information, see the Quickstart. Or, to learn how to build an agent workflow with a customizable architecture, long-term memory, and other complex task handling, see the LangGraph basics tutorials.

Core benefits

LangGraph provides low-level supporting infrastructure for *any* long-running, stateful workflow or agent. LangGraph does not abstract prompts or architecture, and provides the following central benefits:

- Durable execution: Build agents that persist through failures and can run for extended periods, automatically resuming from exactly where they left off.
- Human-in-the-loop: Seamlessly incorporate human oversight by inspecting and modifying agent state at any point during execution.
- Comprehensive memory: Create truly stateful agents with both short-term working memory for ongoing reasoning and long-term persistent memory across sessions.
- Debugging with LangSmith: Gain deep visibility into complex agent behavior with visualization tools that trace execution paths, capture state transitions, and provide detailed runtime metrics.
- Production-ready deployment: Deploy sophisticated agent systems confidently with scalable infrastructure designed to handle the unique challenges of stateful, long-running workflows.

LangGraph's ecosystem

While LangGraph can be used standalone, it also integrates seamlessly with any LangChain product, giving developers a full suite of tools for building agents. To improve your LLM application development, pair LangGraph with:

- LangSmith — Helpful for agent evals and observability. Debug poor-performing LLM app runs, evaluate agent trajectories, gain visibility in production, and improve performance over time.
- LangGraph Platform — Deploy and scale agents effortlessly with a purpose-built deployment platform for long running, stateful workflows. Discover, reuse, configure, and share agents across teams — and iterate quickly with visual prototyping in LangGraph Studio.
- LangChain – Provides integrations and composable components to streamline LLM application development.

> ⓘ Note
>
> Looking for the JS version of LangGraph? See the JS repo and the JS docs.