

langchain-ai / langgraph

<> Code

Issues 82

Pull requests 24

Discussions

Actions

Projects 2

Security

Insights

Build resilient language agents as graphs.

[langchain-ai.github.io/langgraph/](#)

MIT license

Security policy

14.9k stars

2.6k forks

95 watching

Branches

Activity

Custom properties

Tags

Public repository

71 Branches

361 Tags

Go to file

Go to file

+

Add file

<> Code

joaquin-borggio-lc

docs: Add pre-req for egress to control plane (#5241)

✓

84e14f4 · 2 days ago

.github

Revert "Reapply "docs: deploy from v0 ..."

3 days ago

docs

docs: Add pre-req for egress to control...

2 days ago

examples

docs: Time travel and breakpoints (#52...

2 days ago

libs

Bump to 0.0.87

2 days ago

.gitignore

Improve type checking on graph init ...

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 c...

last year

Makefile

lint: use pep 604 union syntax and pep ...

last month

README.md

Remove gitmcp badge (#5055)

3 weeks ago

security.md

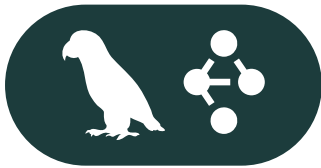
x (#210)

last year

README

MIT license

Security



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"}]}
)
```



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](#).


Additional resources

- [Guides](#): Quick, actionable code snippets for topics such as streaming, adding memory & persistence, and design patterns (e.g. branching, subgraphs, etc.).
- [Reference](#): Detailed reference on core classes, methods, how to use the graph and checkpointing APIs, and higher-level prebuilt components.
- [Examples](#): Guided examples on getting started with LangGraph.
- [LangChain Academy](#): Learn the basics of LangGraph in our free, structured course.
- [Templates](#): Pre-built reference apps for common agentic workflows (e.g. ReAct agent, memory, retrieval etc.) that can be cloned and adapted.
- [Case studies](#): Hear how industry leaders use LangGraph to ship AI applications at scale.

Acknowledgements

LangGraph is inspired by [Pregel](#) and [Apache Beam](#). The public interface draws inspiration from [NetworkX](#).
LangGraph is built by LangChain Inc, the creators of LangChain, but can be used without LangChain.

Releases 351

 **sdk==0.1.72** Latest
3 days ago

+ 350 releases

Packages

No packages published

Used by 31.4k



Contributors 212



+ 198 contributors

Deployments 500+

- ✓ Preview 29 minutes ago
- ✓ Production 2 days ago

+ more deployments

Languages

