Open on GitHub

Introduction

LangChain is a framework for developing applications powered by large language models (LLMs).

LangChain simplifies every stage of the LLM application lifecycle:

- **Development**: Build your applications using LangChain's open-source components and third-party integrations. Use LangGraph to build stateful agents with first-class streaming and human-in-the-loop support.
- **Productionization**: Use LangSmith to inspect, monitor and evaluate your applications, so that you can continuously optimize and deploy with confidence.
- **Deployment**: Turn your LangGraph applications into production-ready APIs and Assistants with LangGraph Platform.

LangChain implements a standard interface for large language models and related technologies, such as embedding models and vector stores, and integrates with hundreds of providers. See the integrations page for more.

Select chat model: OpenAI ▾

```
pip install -qU langchain-openai
```

```
import getpass
import os

if not os.environ.get("OPENAI_API_KEY"):
  os.environ["OPENAI_API_KEY"] = getpass.getpass("Enter API key for OpenAI: ")

from langchain_openai import ChatOpenAI
```

```python
model = ChatOpenAI(model="gpt-4o-mini")
```

```python
model.invoke("Hello, world!")
```

> NOTE
>
> These docs focus on the Python LangChain library. Head here for docs on the JavaScript LangChain library.

Architecture

The LangChain framework consists of multiple open-source libraries. Read more in the Architecture page.

- **`langchain-core`**: Base abstractions for chat models and other components.
- **Integration packages** (e.g. `langchain-openai`, `langchain-anthropic`, etc.): Important integrations have been split into lightweight packages that are co-maintained by the LangChain team and the integration developers.
- **`langchain`**: Chains, agents, and retrieval strategies that make up an application's cognitive architecture.
- **`langchain-community`**: Third-party integrations that are community maintained.
- **`langgraph`**: Orchestration framework for combining LangChain components into production-ready applications with persistence, streaming, and other key features. See LangGraph documentation.

Guides

Tutorials

If you're looking to build something specific or are more of a hands-on learner, check out our tutorials section. This is the best place to get started.

These are the best ones to get started with:

- Build a Simple LLM Application

- Build a Chatbot
- Build an Agent
- Introduction to LangGraph

Explore the full list of LangChain tutorials here, and check out other LangGraph tutorials here. To learn more about LangGraph, check out our first LangChain Academy course, *Introduction to LangGraph*, available here.

How-to guides

Here you'll find short answers to “How do I….?” types of questions. These how-to guides don't cover topics in depth – you'll find that material in the Tutorials and the API Reference. However, these guides will help you quickly accomplish common tasks using chat models, vector stores, and other common LangChain components.

Check out LangGraph-specific how-tos here.

Conceptual guide

Introductions to all the key parts of LangChain you'll need to know! Here you'll find high level explanations of all LangChain concepts.

For a deeper dive into LangGraph concepts, check out this page.

Integrations

LangChain is part of a rich ecosystem of tools that integrate with our framework and build on top of it. If you're looking to get up and running quickly with chat models, vector stores, or other LangChain components from a specific provider, check out our growing list of integrations.

API reference

Head to the reference section for full documentation of all classes and methods in the LangChain Python packages.

Ecosystem

🦜🛠️ LangSmith

Trace and evaluate your language model applications and intelligent agents to help you move from prototype to production.

🦜🕸️ LangGraph

Build stateful, multi-actor applications with LLMs. Integrates smoothly with LangChain, but can be used without it.

Additional resources

Versions

See what changed in v0.3, learn how to migrate legacy code, read up on our versioning policies, and more.

Security

Read up on security best practices to make sure you're developing safely with LangChain.

Contributing

Check out the developer's guide for guidelines on contributing and help getting your dev environment set up.

Edit this page

Was this page helpful?

👍 👎

Next

Tutorials »

Community

Twitter

GitHub

Organization

Python

JS/TS

More

Homepage

Blog

YouTube