Bloom Institute
of Technology

LangServe

Seamlessly deploy and manage LangChain-based agents using LangServe, Docker, LangChain CLI, and Render for efficient cloud operations.

Core Competencies

The student must demonstrate…

1. Understanding LangServe (5 min)

2. Deploying app to Render (10 min)

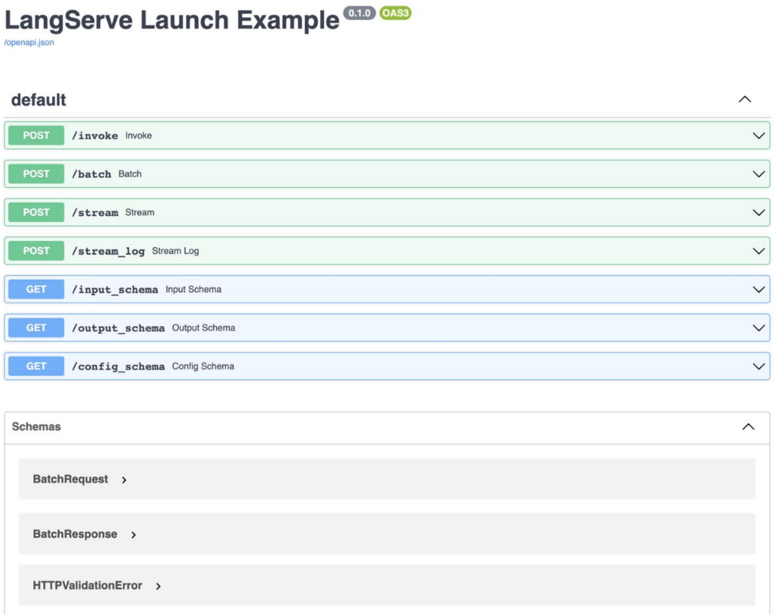
3. Set up feedback loop (10 min)

4. Scoring responses and adding them to a dataset (5 min)

5. Understanding how to use agents and graphs on LangServe (10 min)

LangServe

Langserve is a framework designed to easily deploy LLM chains as scalable web services. It leverages FastAPI and other reliable libraries to provide robust API endpoints and seamless integration with various LLM chains.



LangServe Launch Example 0.1.0 OAS3

/openapi.json

default

POST /invoke Invoke

POST /batch Batch

POST /stream Stream

POST /stream_log Stream Log

GET /input_schema Input Schema

GET /output_schema Output Schema

GET /config_schema Config Schema

Schemas

BatchRequest ›

BatchResponse ›

HTTPValidationError ›

How it Works: Use a conversational retrieval chain (or any other chain/agent) and pass it to `add_routes`.

Features

1. **Scalable Python Web Server:** Automatically inferred and enforced input/output schemas with rich error messages.

2. **API Documentation:** /docs endpoint serves API docs with JSON Schema and Swagger.

3. **Invoke Endpoint:** /invoke endpoint accepts JSON input and returns JSON output, supporting concurrent requests.

4. **Batch Endpoint:** /batch endpoint produces output for several inputs in parallel, batching calls to LLMs where possible.

5. **Streaming Endpoint:** /stream endpoint sends output as it become available, using SSE (like OpenAI Streaming API).

6. **Stream Log Endpoint:** /stream_log endpoint for streaming all (or some) intermediate steps from your chain/agent.

Deploy Agents On Render

Objective: Deploy a LangChain-based agent/chain, including installation, package management, optional LangSmith setup, and Docker deployment by linking Render to a public Git repository, configuring instance type and environment variables for seamless deployment.

Overview of Tasks:

1. Create a Render account.
2. Click "New" and select "Web Service."
3. Switch to "Public Git Repository."
4. Fork the provided template repo.
5. Paste the link to your forked repository.
6. Update the instance type to "Free."
7. Add your environment variables based on the .env.sample file.

Hands-On Homework.

Start brainstorming your final project goals:

1. **Identify a Problem:** Collaborate to find a problem suitable for automation or AI solutions.

2. **Design the Solution:** Plan the implementation, specifying the agent's tasks, training strategy, and performance metrics.