

Promtior technical challenge - Federico Traiman

Tech stack

- LangChain 0.1.0: Orchestration framework
- Ollama: Local LLM server
- Llama 3: Language model
- ChromaDB: Vector store/database
- LangServe: API framework
- FastAPI: Web server
- BeautifulSoup4: Web scraping library

Key Interaction Flow

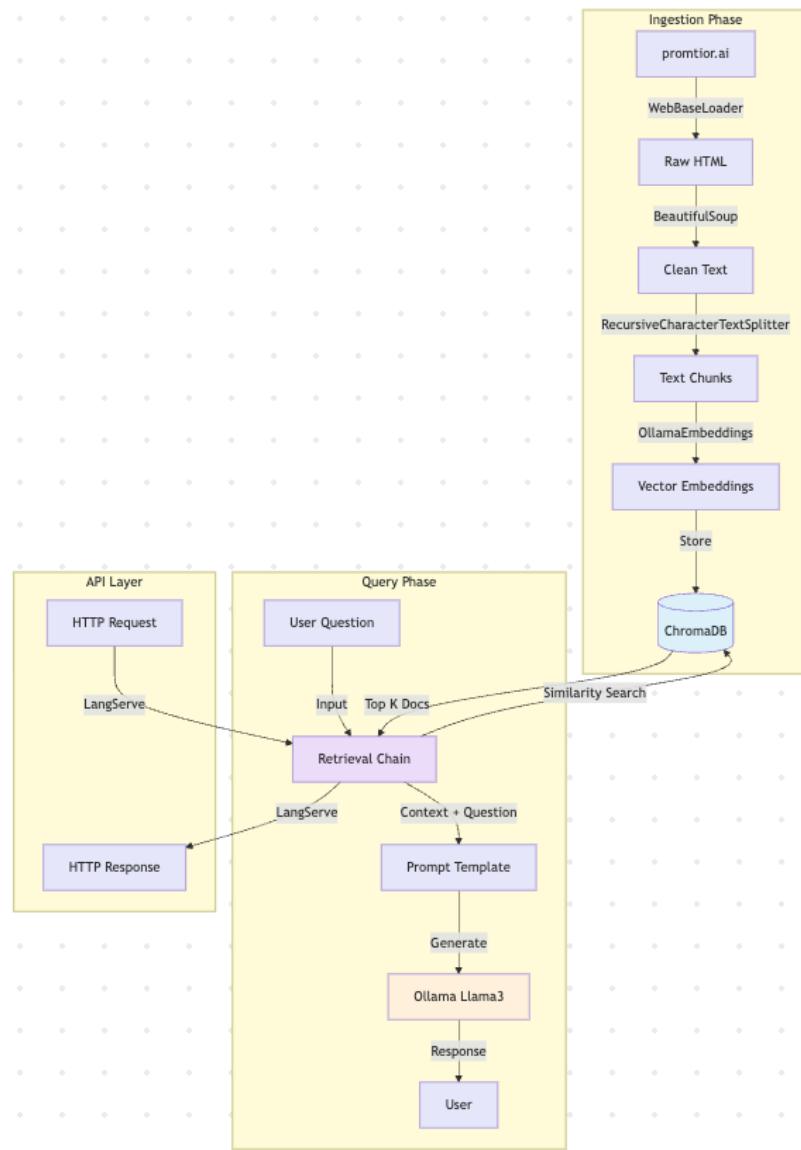
1. Ingestion (Pre-processing)

- Input: The system scrapes promtior.ai.
- Embeddings: Text is split into chunks and converted into vectors using OllamaEmbeddings.
- Storage: Vectors are stored locally in ChromaDB.

2. Runtime (Chat)

- Receive Question: User sends a question via the API or Playground.
- Retrieval: The system searches ChromaDB for the most relevant text chunks.
- Generation: The question + relevant context are sent to the Ollama (Llama 3) model running locally.
- Response: The model's answer is streamed back to the user.

Implementation diagram

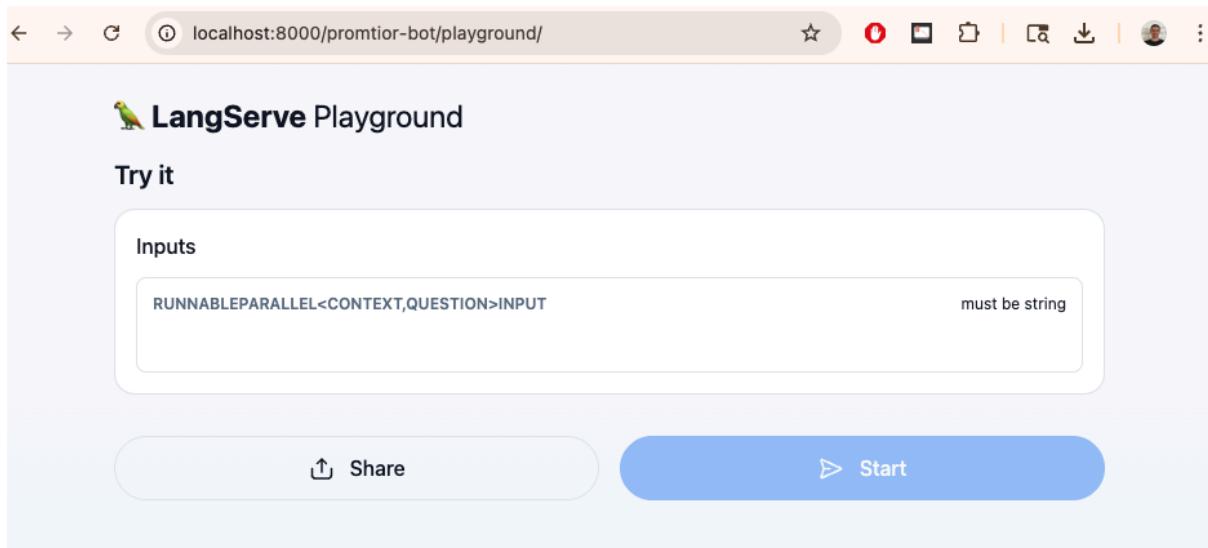


Instructions to execute

Execute the following commands in the terminal:

1. `python3 -m pip install -r requirements.txt` (install dependencies)
2. `python3 -m app.ingest` (initiate data ingestion)
3. `python3 -m app.server` (start the server app)
4. <http://localhost:8000/promtior-bot/playground/> (open visual interface)

Visual interface



Testing the app

Data ingestion

```
[federicotraiman@34HW016989 promtior-ollama-bot % python3 -m app.ingest
/Users/federicotraiman/Library/Python/3.9/lib/python/site-packages/urllib3/_init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3
'. See: https://github.com/urllib3/urllib3/issues/3020
warnings.warn(
USER_AGENT environment variable not set, consider setting it to identify your requests.
🌐 Loading data from https://promtior.ai/...
✓ Loaded 1 document(s)

📄 Splitting text into chunks...
✓ Created 4 chunks

📁 Removing old vector store...

🔮 Creating embeddings with Ollama (llama3)...
/Users/federicotraiman/Projects/Promtier/promtior-ollama-bot/app/ingest.py:44: LangChainDeprecationWarning: The class `OllamaEmbeddings` was deprecated in LangChain 0.3.1 and will be removed in 1.0.0. An updated version of the class exists in the :class:`~langchain-ollama` package and should be used instead. To use it run `pip install -U :class:`~langchain-ollama` and import as `from :class:`~langchain_ollama import OllamaEmbeddings``.
embeddings = OllamaEmbeddings(model=OLLAMA_MODEL, base_url=OLLAMA_BASE_URL)
💾 Storing in ChromaDB at /Users/federicotraiman/Projects/Promtier/promtior-ollama-bot/data/chroma_db...

✅ Ingestion complete! 4 chunks stored in vector database.
federicotraiman@34HW016989 promtior-ollama-bot % ]
```

Starting the app

```
federicotraiman@34HW016989 promtior-ollama-bot % python3 -m app.server

/Users/federicotraiman/Library/Python/3.9/lib/python/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'Libressl 2.8.3'.. See: https://github.com/urllib3/urllib3/issues/3020
    warnings.warn(
/Users/federicotraiman/Projects/Promtier/promtior-ollama-bot/app/chain.py:24: LangChainDeprecationWarning
: The class `OllamaEmbeddings` was deprecated in LangChain 0.3.1 and will be removed in 1.0.0. An updated
version of the class exists in the :class:`~langchain-ollama` package and should be used instead. To use
it run `pip install -U :class:`~langchain-ollama` and import as `from :class:`~langchain_ollama import OllamaEmbeddings``.
    embeddings = OllamaEmbeddings(model=OLLAMA_MODEL, base_url=OLLAMA_BASE_URL)
/Users/federicotraiman/Projects/Promtier/promtior-ollama-bot/app/chain.py:27: LangChainDeprecationWarning
: The class `Chroma` was deprecated in LangChain 0.2.9 and will be removed in 1.0. An updated version of
the class exists in the :class:`~langchain-chroma` package and should be used instead. To use it run `pip
install -U :class:`~langchain-chroma` and import as `from :class:`~langchain_chroma import Chroma``.
    vectorstore = Chroma(
/Users/federicotraiman/Projects/Promtier/promtior-ollama-bot/app/chain.py:35: LangChainDeprecationWarning
: The class `ChatOllama` was deprecated in LangChain 0.3.1 and will be removed in 1.0.0. An updated versi
on of the class exists in the :class:`~langchain-ollama` package and should be used instead. To use it run
`pip install -U :class:`~langchain-ollama` and import as `from :class:`~langchain_ollama import ChatOlla
ma``.
    llm = ChatOllama(model=OLLAMA_MODEL, base_url=OLLAMA_BASE_URL, temperature=0)
✓ Server initialized successfully!
 Endpoint:
 - Playground: http://localhost:8000/promtior-bot/playground
 - Invoke: http://localhost:8000/promtior-bot/invoke
 - Stream: http://localhost:8000/promtior-bot/stream
INFO:     Started server process [86136]
INFO:     Waiting for application startup.
```



```
LANGSERVE: Playground for chain "/promtior-bot/" is live at:
LANGSERVE:   ↳ /promtior-bot/playground/
LANGSERVE: See all available routes at /docs/
INFO:     Application startup complete.
```

Sample questions

What services does Promtior offer?

The screenshot shows a web browser window for 'LangServe Playground' at the URL 0.0.0.0:8000/promtior-bot/playground/. The browser's address bar includes a 'Not Secure' warning. The main interface has sections for 'Inputs' and 'Output'. In the 'Inputs' section, there is a text area containing the question 'What services does Promtior offer?'. Below this, the 'Output' section displays a response in Spanish: 'Basándome en el contexto proporcionado, puedo responder que Promtior ofrece los siguientes servicios:' followed by a bulleted list of services categorized by industry. The list includes:

- * GenAI Department as a service
- * GenAI Adoption Consulting
- * Transform your Business with Generative AI (Operations)
 - + Automate repetitive tasks.
 - + Optimize workflow processes.
 - + Reduce manual errors.
- * Technology
 - + Auto-generate and review code.
 - + Automate software testing.
 - + Optimize IT resources.
- * Sales
 - + Personalize customer communications.
 - + Predict customer needs.
 - + Automate lead follow-ups.
- * Legal
 - + Quickly extract key information from documents.
 - + Generate draft contracts.
 - + Comparative analysis of regulations.
- * Customer Service
 - + Provide automated and personalized responses.
 - + Real-time sentiment analysis.
 - + Proactive case and ticket management.
- * Finance
 - + Predictive analysis of financial trends.
 - + Automatically generate financial reports.
 - + Detect fraud and anomalies.

Below the service list, a note states: 'Además, Promtior también ofrece GenAI Product Delivery, que incluye predictive analytics, intelligent automation y otros servicios.'

At the bottom of the interface, there are buttons for 'Intermediate steps 8' and 'Share' (with a link icon), and a large blue button labeled 'Start' with a right-pointing arrow icon.

When was the company founded?

The screenshot shows a web-based playground interface for LangServe. At the top, the URL bar displays "localhost:8000/promtior-bot/playground/". Below the URL is a header with a parrot icon and the text "LangServe Playground". A "Try it" button is visible. The main area is divided into "Inputs" and "Output" sections. In the "Inputs" section, there is a text input field containing the question "When was the company founded?". A "Reset" button is located at the top right of this section. In the "Output" section, there is a text box containing the response "No tengo suficiente información para responder esa pregunta.". Below the output box is a link "Intermediate steps 8 >". At the bottom, there are two buttons: a white "Share" button with a blue outline and a large blue "Start" button.