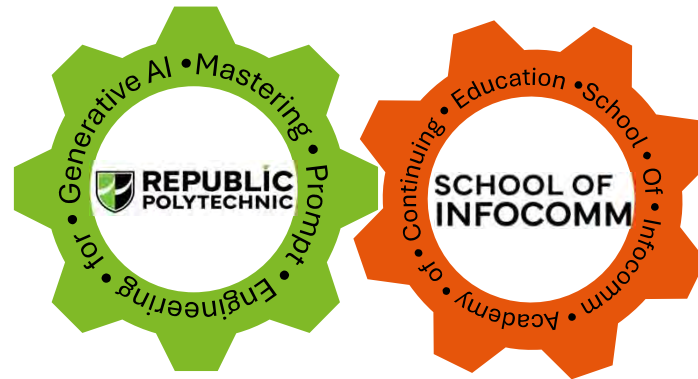


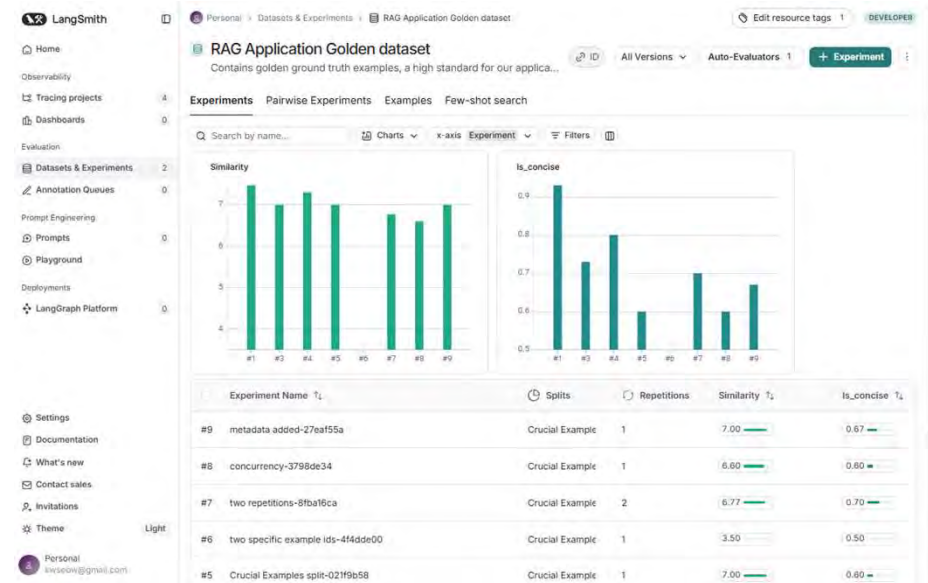
2025



Lesson 15 – Langsmith tracing

LangSmith

- **LangSmith** is a platform for building production-grade LLM applications. It allows you to closely monitor and evaluate your application, so you can ship quickly and with confidence. With LangSmith you can:
 - **Trace LLM Applications:** Gain visibility into LLM calls and other parts of your application's logic.
 - **Evaluate Performance:** Compare results across models, prompts, and architectures to identify what works best.
 - **Improve Prompts:** Quickly refine prompts to achieve more accurate and reliable results.
- LangSmith is a **standalone platform** that can be used on it's own no matter how you're creating your LLM applicatons.



LangSmith

- **Tracing** to give insights into our application as we design our overall architecture
- Quickly **iterate prompts** directly within LangSmith
- Setup programmatic **offline evaluations** in LangSmith
- Introduce a few ways to **collect human feedback**
- **Monitor production performance** in LangSmith and leverage usage patterns to continuously improve your application over time.

Sign up

- Sign up for LangSmith [here](#). You can reference LangSmith docs [here](#).
- Navigate to the Settings page, and generate an API key in LangSmith.
- Create a .env file that mimics the provided .env. Set **LANGCHAIN_API_KEY** in the .env file.
- Set **OPEN_API_KEY** in the .env file.

Activity – Langsmith intro

01

Sign up Langsmith
<https://smith.langchain.com>

02

Generate API Key
Setting. API Key, Create API Key

03

Hook API Key to application
00-rag_application.ipynb

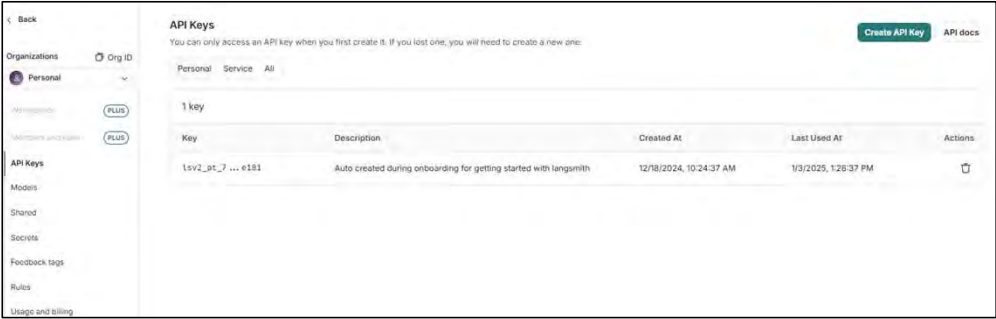
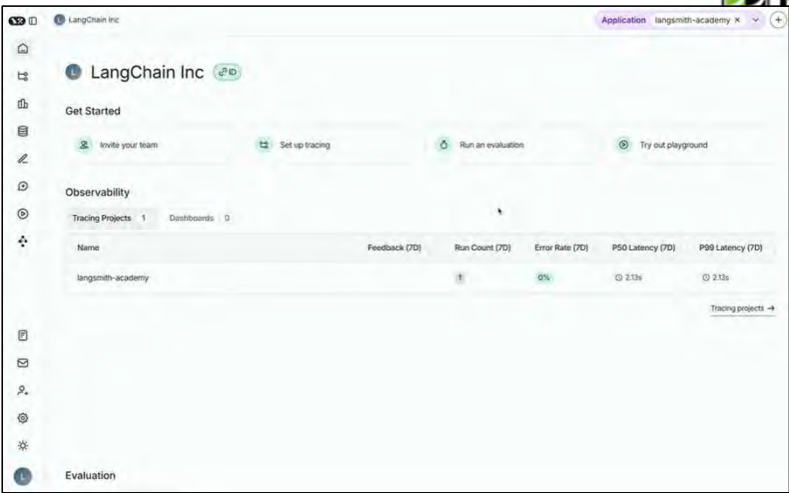
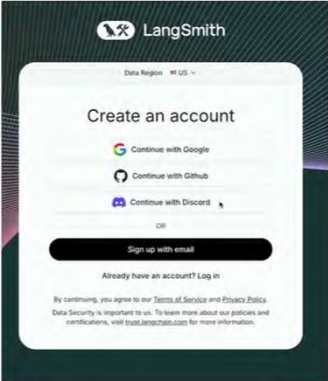
04

Set up Environment variables
.env.txt



10 mins

Official (Closed) \ Non-Sensitive

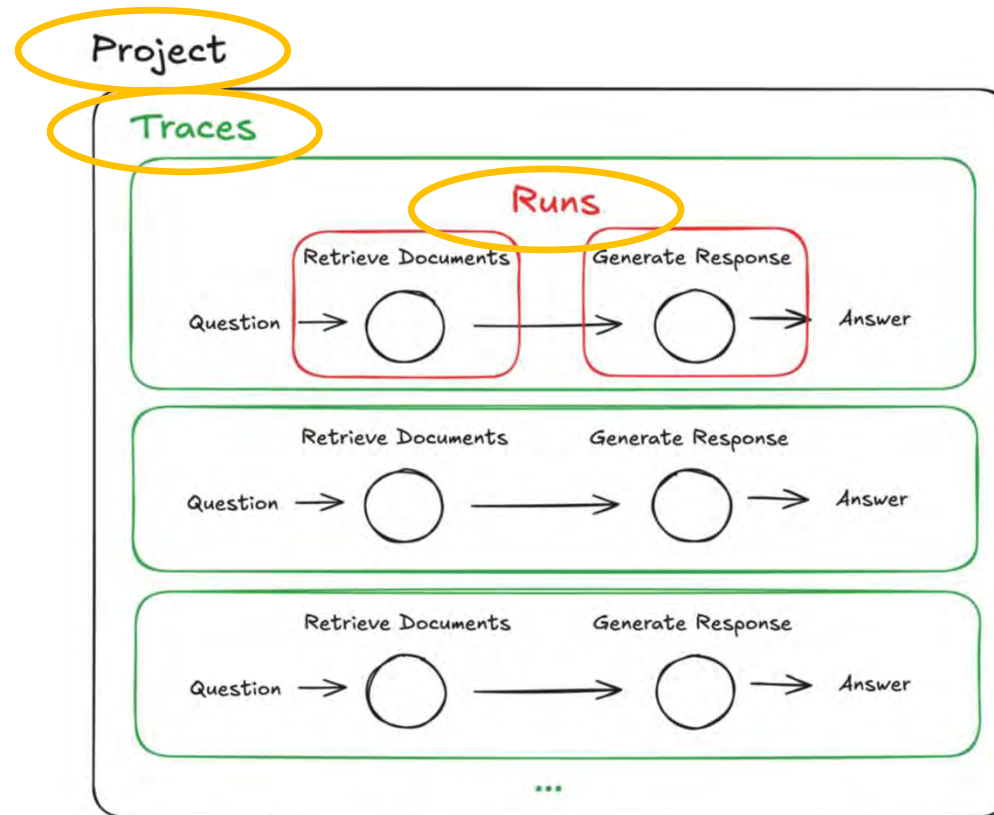


Target to finish by XX:XX

Tracing

- May need to **debug unexpected outputs** or performance bottlenecks
- Tracing helps **pinpoint issues** and track how each part of your application contributes to the output
- Set up tracing and dive deeper into the type of information they capture
- Walk through how you can use Traces to debug your application as you build

Tracing Concepts



Tracing Concepts

Trace

Run Tree

RAG

- └ Retrieve Documents
 - └ SKLearn Vector DB
- └ Generate Response
 - └ OpenAI Chat Completion

Run

Name: Retrieve Documents

- Input: What can I do with LangSmith?
- Output: Document A, Document B ...
- Metadata: {"db_provider": "sklearn"}
- Run Type: "chain"
- Latency: 0.22s
- Feedback: {"docs_are_good": true}

Activity – Tracing basics with decorator

- 01 Set up Environment variables
- 02 Add trace with @traceable
- 03 Add Metadata & filter runs with filter
- 04 Share run

Prepare your RAG/Web Search application for tracing

Share your trace to the padlet



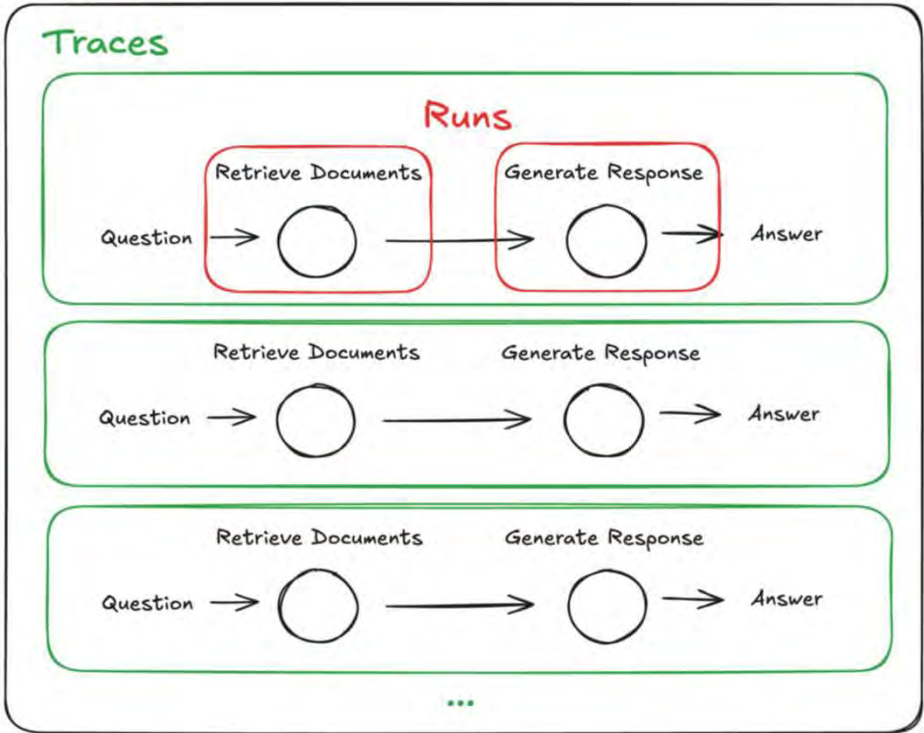
20
mins

L15-01-tracing_basics.ipynb

Official (Closed) \ Non-Sensitive



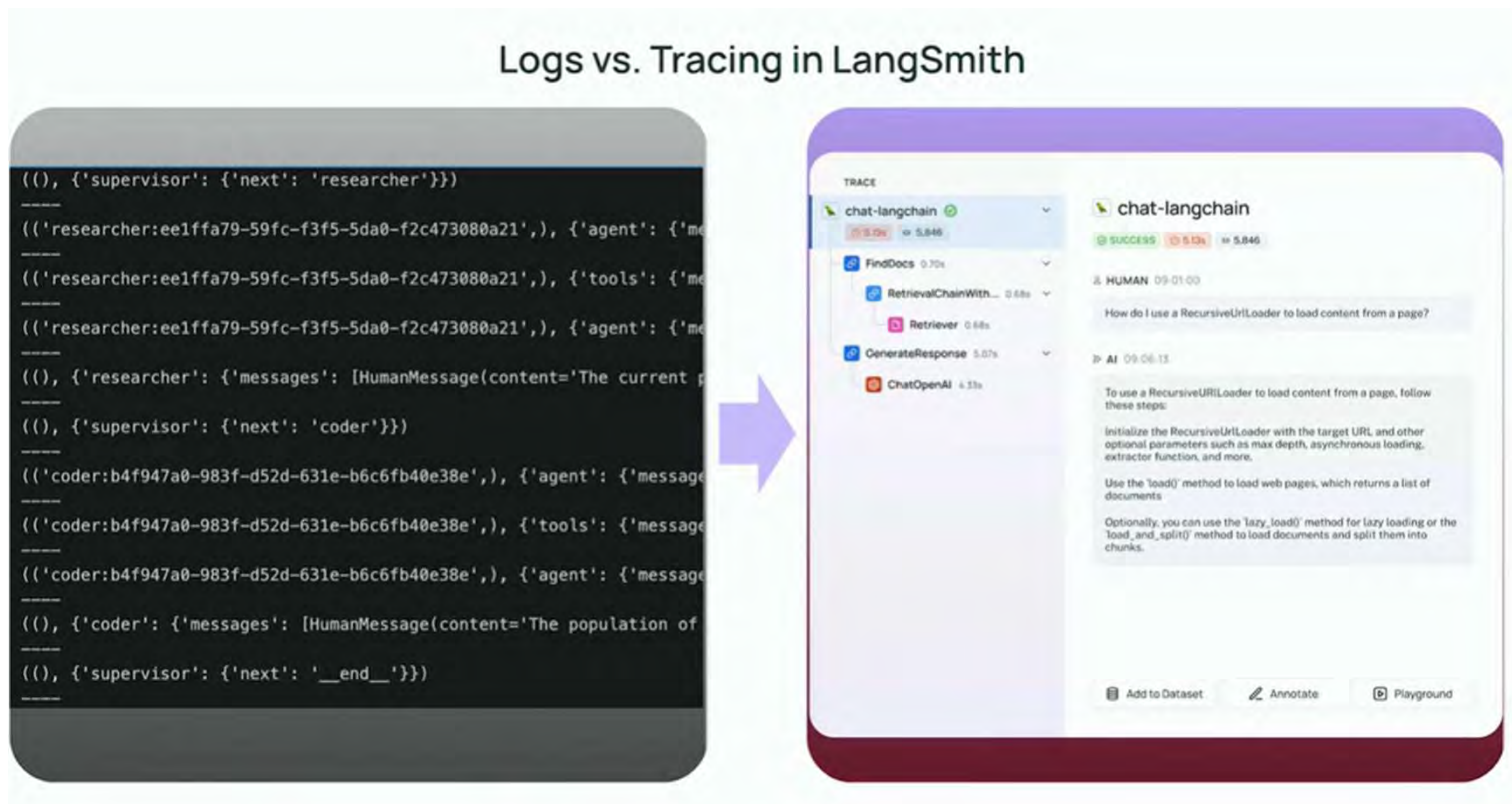
Project









Target to finish by XX:XX

Tracing – Type of runs

- Logs vs. Tracing in LangSmith



Tracing – Type of runs

- LangSmith supports many different types of Runs
- You can specify your Run in the @traceable decorator
- Types of runs:
 - LLM – Invokes an LLM 
 - Retriever – Retrieves documents from databases or other sources 
 - Tool – Executes actions with function calls 
 - Chain – Default type; combines multiple Runs into a larger process 
 - Prompt – Hydrates a prompt to be used with an LLM 
 - Parser – Extracts structured data 

Activity – Types of Runs

01

Set up Environment variables

02

LLM Run Type

link to playground

03

Add Metadata

Is+provider and Is_model_name

04

Handling Streaming LLM Runs

05

Retriever runs + document

06

Tool call

Prepare your RAG application for tracing

Share your trace to the padlet

Target to finish by XX:XX



Official (Closed) \ Non-Sensitive



15 Mins Break

Back by 10:30

Activity – Alternative Ways to Trace

01

Set up Environment variables

02

@traceable

03

LangChain/LangGraph

04

With trace()

05

wrap_openai()

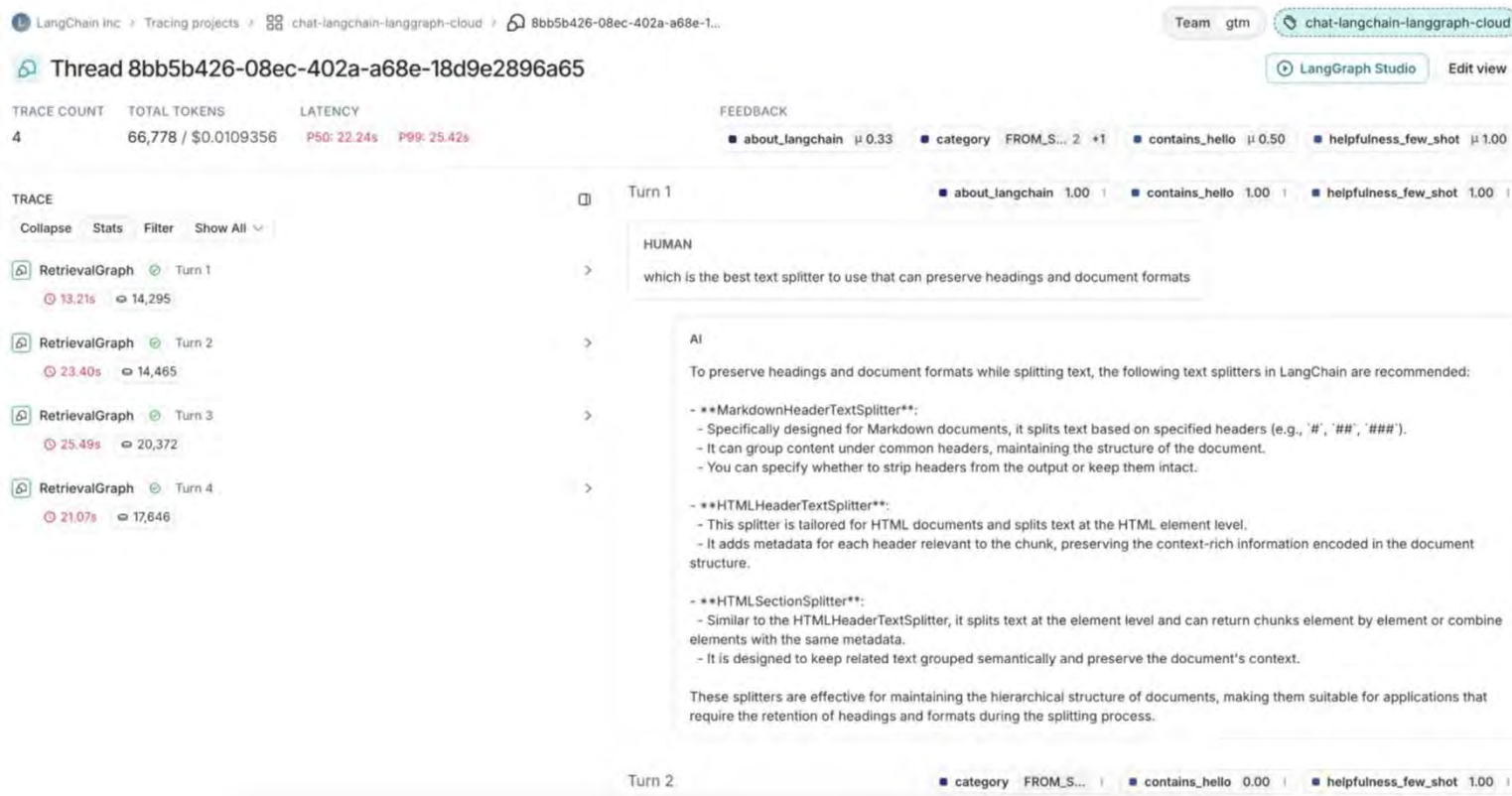
Method	When-to-use
@traceable	<ul style="list-style-type: none"> - Default way to set up tracing - Manages RunTree, inputs, and outputs for you
LangChain/LangGraph	<ul style="list-style-type: none"> - Get tracing out of the box!
with trace():	<ul style="list-style-type: none"> - Want more control over what inputs and outputs get logged - Cannot use decorators or wrappers (not tracing a function)
wrap_openai()	<ul style="list-style-type: none"> - Want to use the OpenAI SDK directly
RunTree	<ul style="list-style-type: none"> - Want more low level control over tracing configuration - Need to know run_id (for adding user feedback or logging to another service)

Target to finish by XX:XX



Tracing – Conversational Threads

- More meaningful to put all chat history into one thread



Activity – Conversational Thread

01

Set up Environment variables

02

Group traces into threads

03

Click on the Thread button to thread view in particular trace



10 mins

Official (Closed) \ Non-Sensitive



LangSmith

Home

Observability

Tracing projects5

Dashboards0

Evaluation

Datasets & Experiments2

Annotation Queues0

Prompt Engineering

Prompts0

Playground

Deployments

LangGraph Platform0

Settings

Documentation

What's new

Contact sales

Invitations

ThemeLight

Personal

kwseow@gmail.com

Personal > Tracing projects > langsmith-rp > a84acb68-4e45-4fc2-a9f8-3...

Add resource tags

DEVELOPER

Edit view

Thread a84acb68-4e45-4fc2-a9f8-32136b759340

TRACE COUNT

TOTAL TOKENS

LATENCY

2

1,288

P50: 2.36sP99: 3.20s

TRACE

Collapse

Stats

Filter

Show All

langsmith_rag

Turn 1

1.50s763

langsmith_rag

Turn 2

3.22s525

Load more

Turn 1

HUMAN

How do I add metadata to a Trace?

Output

To add metadata to a trace in LangSmith, you need to send arbitrary metadata and tags along with the trace. This can include information like the execution environment or the user who initiated the trace. For detailed instructions, refer to the documentation on adding metadata keys to a trace.

Turn 2

HUMAN

How can I add tags to a Trace?

Output

You can add tags to a trace in LangSmith by sending arbitrary metadata and tags along with the trace. Additionally, you can annotate a trace inline by clicking on the "Annotate" button in the upper right corner of the trace view or by sending it to the Annotation Queue. Feedback tags are associated with your workspace and can be used to critique specific parts of the trace.

© 2025 - Republic Polytechnic (Singapore) - All Rights Reserved

RP
16

Official (Closed) \ Non-Sensitive



Thank you!