



Arize Calhacks Hacker Starter Pack

Arize is an AI engineering platform focused on evaluation and observability. It helps engineers develop, evaluate, and observe AI applications and agents.

Arize has both Enterprise and OSS products to support this goal:

- Arize AX — an enterprise AI engineering platform from development to production, with an embedded Alyx
- Phoenix — a lightweight, open-source project for tracing, prompt engineering, and evaluation

Why Observability Matters

Observability matters because it gives you insight into what's really happening under the hood with your AI applications/agents. It's how you spot when your agent goes off track, starts hallucinating, or gives a confusing response. For example, if users keep getting frustrated when talking to a chatbot, observability can show you whether the issue came from missing context, bad retrieval, or a broken tool call.

[Arize Phoenix](#)

At Calhacks, we'll be looking at Phoenix, our open source observability product!

[Phoenix Cloud](#) offers free, ready-to-use Phoenix instances preconfigured with 10 GiB of storage, more than enough for any project pre-deployment.

Observability unlocks tracing and evaluation pipelines.

What is Tracing?

[Tracing Quickstart](#)

LLM tracing records the paths taken by agents or LLM apps as they propagate through multiple steps of a request. For example, when a user interacts with an LLM application, tracing can capture the sequence of operations, such as tools calls, LLM response generation, and document retrieval to provide a detailed timeline of the request's execution.

The screenshot displays the OpenAI Tracing interface, which provides a detailed view of the execution path of an LLM application. The interface is divided into several sections:

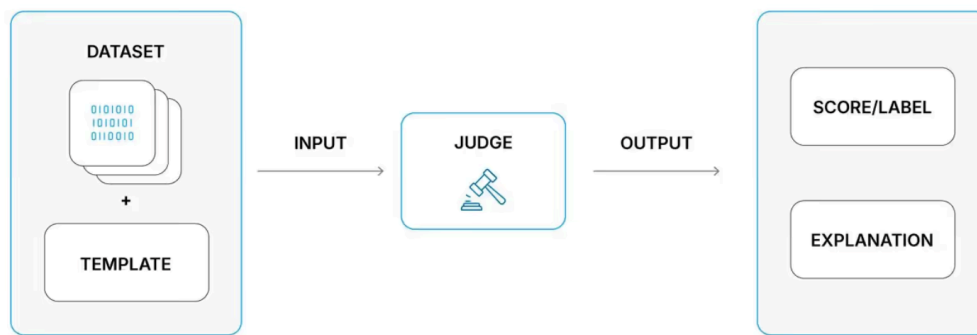
- Trace Details:** A sidebar on the left showing a tree view of the trace steps. The steps include: BaseQueryEngine.query (1.00s), RetrieverQueryEngine.query (0.91s), BaseRetriever.retrieve (0.15s), VectorIndexRetriever.retrieve (0.14s), BaseEmbedder.get_query_embedding (0.11s), OpenAIEmbedder.get_query_embedding (0.11s), CohereReranker.postprocess_nodes (0.15s), BaseSynthesizer.synthesize (0.59s), CompactAndRefine.get_response (0.58s), TokenTextSplitter.split_text (4.19ms), Refine.get_response (0.57s), TokenTextSplitter.split_text (4.14ms), DefaultRefineProgram.call (0.55s), LLM.predict (0.55s), and OpenAI.chat (1631, 0.54s).
- Trace Status:** A top bar indicating the overall status of the trace. It shows "OK" with a green checkmark, a latency of 1.00s, and a feedback button.
- Input Messages:** A section showing the input messages to the LLM. It includes a system message with instructions and a user message with context information and a request for the PHOENIX_API_KEY.
- Output:** A section showing the output of the LLM, which is a JSON object containing the API key.
- Summary:** A right sidebar providing summary information about the trace, including the status (OK), start time (10/8/2024 16:09:52 PM), end time (10/8/2024 16:09:52 PM), latency (0.54s), and total tokens (1631).

What are Evals?

[Evals Quickstart](#)

LLMs are non-deterministic, and their execution and outputs can vary even with the same input. To ensure application performance, you need a way to judge the quality of your LLM outputs.

There are many ways to go about this, but the most common is using LLM-as-a-Judge. This is an evaluation method that uses an LLM to judge the output of another LLM.



This approach works great for evaluations because you can define any custom criteria and describe it in plain English. From there, you can feed in existing data, run it through the Judge, and review aggregate metrics to understand overall performance.

Total Traces	Total Cost	Latency P50	Latency P99	relevance	tone
10	<\$0.01	4.43s	8.64s	μ 1.00	μ 1.00
Spans	Traces	Sessions	Metrics	Config	
Q filter condition (e.x. span_kind == 'LLM')					
☐	status	kind	name	input	output
☐	⊖	agent	customer_support_sessio	What's your refund po...	--
				relevance	μ 1.00
				tone	μ 1.00
10/					
☐	⊖	agent	customer_support_sessio	Can I downgrade to a ...	--
				relevance	μ 1.00
				tone	μ 1.00
10/					
☐	⊖	agent	customer_support_sessio	When can I upgrade m...	--
				relevance	μ 1.00
				tone	μ 1.00
10/					

Cookbooks and Tutorials

These notebooks/tutorials are to help you getting started with implementing Tracing/Evals and building more robust Agents and LLM Applications.

- [End-to-End Tutorial](#)
- [Creating a custom LLM Judge evaluator](#)

Videos

- [Tracing and Evaluating OpenAI Agents](#)
- [Trace Level Evals](#)
- [Session Level Evals](#)

Career Opportunities

- [Open Roles](#)