Getting Started

This guide will walk you through the process of integrating Coralogix's AI Observability solution with the OpenAI platform to monitor and gain insights into your LLM applications. By following these steps, you'll be able to start sending AI observability data to Coralogix AI Center in just a few minutes.

> **NOTE**
>
> For instructions on instrumenting interactions with Bedrock-hosted models, refer to the Coralogix Amazon Bedrock documentation.

Requirements

To integrate your application with Coralogix's AI Observability solution using OpenAI, ensure you have the following:

- Python 3.8 or higher
- A Coralogix account with an API key
- An OpenAI API key

AI Center processes **only trace data**, not logs, and retrieves it **exclusively from your S3 archive**. Data stored in Frequent Search will be ignored. To ensure full compatibility, instrument your observability data as traces and route it to archive storage.

Install the SDK

Install the LLM Tracekit library using `pip`.

```
pip install llm-tracekit[openai]
```

Set up environment variables

Configure the necessary environment variables.

```
# Coralogix credentials
export CX_TOKEN="your-coralogix-api-key"
export CX_ENDPOINT="your-coralogix-region-endpoint"  # Replace with your
region endpoint (e.g. https://ingress.staging.coralogix.net:443)
```

```
# OpenAI API key (for the example)
export OPENAI_API_KEY="your-openai-api-key"
```

Create a simple application

Create a new Python file (e.g., `ai_center_demo.py`) using the following code:

```
import os
from openai import OpenAI
from llm_tracekit import OpenAIInstrumentor, setup_export_to_coralogix

# Configure export to Coralogix
setup_export_to_coralogix(
    service_name="ai-demo-service",
    application_name="ai-demo-app",
    subsystem_name="getting-started"
)

# Instrument OpenAI client
OpenAIInstrumentor().instrument()

# Initialize OpenAI client
client = OpenAI()

# Send a request to OpenAI
def generate_content():
    print("Sending request to OpenAI...")
    response = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=[
            {"role": "system", "content": "You are a helpful assistant."},
            {"role": "user", "content": "Explain what AI observability is
in one sentence."},
        ],
    )

    # Display a nicer formatted response
    print("\n" + "="*50)
    print(" AI RESPONSE:")
    print(f"{response.choices[0].message.content}")
```

```
    print("="*50)

    # Confirmation about traces
    print("\n Traces have been successfully sent to Coralogix AI Center!")
    print("View your data in the Coralogix AI Center dashboard.\n")

if __name__ == "__main__":
    generate_content()
```

Run the application

Execute your Python script.

```
python ai_center_demo.py
```

The output should resemble the following:

```
Sending request to OpenAI...

==================================================
 AI RESPONSE:
AI observability refers to the tools and practices used to monitor,
analyze, and understand the behavior and performance of AI models and
systems in real-time, ensuring they operate effectively and align with
intended outcomes.
==================================================

 Traces have been successfully sent to Coralogix AI Center!
View your data in the Coralogix AI Center dashboard.
```

View your data in Coralogix AI Center

1. Log into your Coralogix account.
2. Go to **AI Center > Application Catalog** to see your new service.
3. Click on your application to view its detailed information.
4. Navigate to the LLM Calls section to see the trace for your request.

Capture tool calls

If your application uses OpenAI's function calling capabilities, these will be automatically captured as part of the trace data.

Last updated: June 8, 2025

Was this helpful?

Leave your feedback here.

◯ **Yes** ◯ **No**

Send