

Diseño del Prompt para el monitoreo de Logs

Estoy necesitando implementar un proyecto ASP.NET Core Web Api con .NET 10 para el monitoreo de unas Tables de Azure Storage para dos aplicaciones AppSalud y LinaChatbot

El objetivo del proyecto es leer unos Logs que están en unas tablas, para detectar:

- errores de servicios externos (outbound)
- errores internos recurrentes de la aplicación
- tiempos de respuesta elevados según el tipo de servicio

Criterio de aceptación 1: Identificación errores y tiempos de respuestas elevados

Dado que se requiere analizar los Logs en las diferentes tablas del Azure Storage,

Cuando se obtengan los registros de las tablas,

Entonces se debe implementar un LLM que analice los registros obtenidos para identificar errores de servicio externos, internos, y tiempos de respuestas elevados.

Criterio de aceptación 2: Identificación errores y tiempos de respuestas elevados

Dado que se requiere enviar una notificación

Cuando cuando el criterio de aceptación 1 se cumpla

Entonces se debe enviar una notificación ya sea por (Telegram, correo, etc.)

Recursos

Para AppSalud este son los datos que se tienen:

Cuneta de almacenamiento:

DefaultEndpointsProtocol=https;AccountName=appsaluddevstorage;AccountKey=tgw633R2bnHIWaJnuhgkEUaLk0tUU6bv0MSjkplvBQYpz2ZcYSMQWN/ngwQDjMLzTQynp5VgDyOAg==;EndpointSuffix=core.windows.net;

Listado de tablas a analizar: AppLog, Log, LogCsAuthenticate

Para LinaChatbot este son los datos que se tienen:

Cuneta de almacenamiento:

DefaultEndpointsProtocol=https;AccountName=Linadevstorage;AccountKey=tgw633R2bnHIWaJnuhgkEUaLk0tUU6bv0MSjkplvXRTBQYpz2ZcYSMQWN/ngwQDjMLzTQynp5VgDyOAg==;EndpointSuffix=core.windows.net;

Listado de tablas a analizar: AppLog, Log, LogCsAuthenticate

En cuanto al LLM esto es lo que se tiene:

The screenshot displays the Azure AI Foundry console interface for the 'gpt-5-mini' model. The left sidebar contains navigation options such as 'Áreas de juegos', 'Compilación y personalización', 'Agentes', 'Plantillas', 'Ajuste preciso', 'Comprensión de contenidos', 'Observar y optimizar', 'Traza', 'Supervisión', 'Proteger y gobernar', 'Evaluación', 'Barreras de protección y controles', 'Riesgos y alertas', 'Gobernanza', 'Azure OpenAI', 'Almacenes de vectores de asistente', 'Archivos de datos', 'Mis recursos', 'Modelos + puntos de conexión', 'Más', and 'Centro de administración'.

The main content area is titled 'gpt-5-mini' and includes a 'Punto de conexión' section with the URL 'https://orquestador-foundry.cognitiveservices.azure.com/openai/responses/ap...' and a 'Clave' field. Below this is the 'Información de implementación' section, which provides details about the deployment, including the model name, type, creation and modification dates, and the creator's email address.

The right sidebar contains the 'Idioma' (C#), 'SDK' (Azure OpenAI SDK), and 'Tipo de autenticación' (Key Authentication) settings. It also includes an 'Introducción' section with a link to the documentation and a '1. Autenticación mediante la clave de API' section with a code snippet for authenticating the client.

The bottom section shows the '2. Instalar dependencias' section, which includes instructions for installing the dotnet runtime and a code snippet for restoring the packages. The '3. Ejecutar un ejemplo de código básico' section provides a code snippet for making a basic API call to the model.

Área de juegos de chat | Microsoft 365 Copilot | Implementación de AI | Fundación de IA de Azure | AzureAI Foundry Group | Feedback - Microsoft | Overview - Azure AI Foundry | gpt-5-mini - Azure AI | gpt-5-mini - Azure AI |

https://aka.ms/azure-ai-foundry | English as a second... | Paquete de acceso... | AppSulad Firebase... | Outlook calendario... | Inducción de Argu... | Día | IA | Olsacar | Jailbreak

Azure AI Foundry | Monitoreo-Storage-Account | Implementaciones | gpt-5-mini | Documentación | Todos los recursos | Orquestador Foundry (beta2, 50)

Áreas de juegos

Compilación y personalización

Agentes

Plantillas

Ajuste preciso

Comprensión de contenidos

Observar y optimizar

Traza

Supervisión

Proteger y gobernar

Evaluación

Barreras de protección y controles

Riesgos y alertas

Gobernanza

Azure OpenAI

Almacenes de vectores de asistente

Archivos de datos

Mis recursos

Modelos y puntos de conexión

Más

Centro de administración

Parámetro warning: restore ADAD881

```
List(ChatMessage) messages = new List(ChatMessage)();
{
    new SystemChatMessage("You are a helpful assistant."),
    new UserChatMessage("I am going to Paris, what should I see?"),
};

var response = chatClient.CompleteChat(messages, requestOptions);
System.Console.WriteLine(response.Value.Content[0].Text);
```

4. Explorar más ejemplos

Ejecutar una conversación de varios turnos

En este ejemplo se muestra una conversación de varios turnos con la API de finalización de chat. Al usar el modelo para una aplicación de chat, deberá administrar el historial de esa conversación y enviar los mensajes más recientes al modelo.

```
using OpenAI.Chat;
using Azure;
using Azure.AI.OpenAI;

var endpoint = new Uri("https://orquestador-foundry.cognitiveservices.azure.com/");
var deploymentName = "gpt-5-mini";
var apiKey = "your-api-key";

AzureOpenAIClient azureClient = new(
    endpoint,
    new AzureKeyCredential(apiKey));
ChatClient chatClient = azureClient.GetChatClient(deploymentName);

List(ChatMessage) messages = new List(ChatMessage)();
{
    new SystemChatMessage("You are a helpful assistant."),
    new UserChatMessage("I am going to Paris, what should I see?"),
};

var response = chatClient.CompleteChat(messages);
System.Console.WriteLine(response.Value.Content[0].Text);
// Append the model response to the chat history.
messages.Add(new AssistantChatMessage(response.Value.Content[0].Text));
```

Área de juegos de chat | Microsoft 365 Copilot | Implementación de AI | Fundación de IA de Azure | AzureAI Foundry Group | Feedback - Microsoft | Overview - Azure AI Foundry | gpt-5-mini - Azure AI | gpt-5-mini - Azure AI |

https://aka.ms/azure-ai-foundry | English as a second... | Paquete de acceso... | AppSulad Firebase... | Outlook calendario... | Inducción de Argu... | Día | IA | Olsacar | Jailbreak

Azure AI Foundry | Monitoreo-Storage-Account | Implementaciones | gpt-5-mini | Documentación | Todos los recursos | Orquestador Foundry (beta2, 50)

Áreas de juegos

Compilación y personalización

Agentes

Plantillas

Ajuste preciso

Comprensión de contenidos

Observar y optimizar

Traza

Supervisión

Proteger y gobernar

Evaluación

Barreras de protección y controles

Riesgos y alertas

Gobernanza

Azure OpenAI

Almacenes de vectores de asistente

Archivos de datos

Mis recursos

Modelos y puntos de conexión

Más

Centro de administración

System.Console.WriteLine(response.Value.Content[0].Text);

4. Explorar más ejemplos

Ejecutar una conversación de varios turnos

En este ejemplo se muestra una conversación de varios turnos con la API de finalización de chat. Al usar el modelo para una aplicación de chat, deberá administrar el historial de esa conversación y enviar los mensajes más recientes al modelo.

```
using OpenAI.Chat;
using Azure;
using Azure.AI.OpenAI;

var endpoint = new Uri("https://orquestador-foundry.cognitiveservices.azure.com/");
var deploymentName = "gpt-5-mini";
var apiKey = "your-api-key";

AzureOpenAIClient azureClient = new(
    endpoint,
    new AzureKeyCredential(apiKey));
ChatClient chatClient = azureClient.GetChatClient(deploymentName);

List(ChatMessage) messages = new List(ChatMessage)();
{
    new SystemChatMessage("You are a helpful assistant."),
    new UserChatMessage("I am going to Paris, what should I see?"),
};

var response = chatClient.CompleteChat(messages);
System.Console.WriteLine(response.Value.Content[0].Text);
// Append the model response to the chat history.
messages.Add(new AssistantChatMessage(response.Value.Content[0].Text));
// Append new user question.
messages.Add(new UserChatMessage("What is so great about #1?"));

response = chatClient.CompleteChat(messages);
System.Console.WriteLine(response.Value.Content[0].Text);
```