

Configuración en API Assistants

Los API Assistants, además de interactuar con usuarios como hemos visto hasta ahora, también tienen la capacidad de interactuar con otros sistemas. Esto significa que pueden enviar y recibir información a través de diferentes canales, como el correo electrónico, WhatsApp u otras conexiones que se pueden configurar en Make.

En este capítulo, exploraremos una funcionalidad muy importante de Make que nos permite configurar y gestionar nuestros API Assistants de manera eficiente.

Make: una herramienta para configurar API Assistants

Veremos cómo utilizar Make para configurar nuestros API Assistants a través de un ejemplo práctico. Imaginemos que estamos construyendo una plataforma educativa que utiliza personajes históricos como asistentes virtuales. En esta plataforma, tenemos 80 asistentes diferentes, cada uno representando a un personaje histórico.

Gestión de múltiples asistentes con Make

Para gestionar estos 80 asistentes de manera eficiente, utilizaremos la plataforma de OpenAI Assistants. Cada asistente está diseñado para interactuar con dos tipos de usuarios: niños (NI) y docentes o adultos (DA). Por lo tanto, en realidad, tenemos 40 personajes históricos representados por dos asistentes cada uno.

Estructura del "Prompt"

Para que cada asistente se comporte de manera coherente con el personaje que representa, es crucial definir un "Prompt" adecuado. Este "Prompt" incluye información sobre el rol que debe interpretar el asistente, así como el contexto clave del personaje, como su biografía, sus logros y su época.

Aspectos Clave del Prompt:

- **Rol a interpretar:** Se especifica claramente quién es el personaje que el asistente debe interpretar.
- **Información contextual:** Se proporciona información relevante sobre la vida del personaje, como fechas de nacimiento y muerte, nacionalidad, logros y contexto histórico.
- **Fecha actual:** Es importante incluir la fecha actual en el "Prompt", ya que la información de los asistentes puede estar limitada a la fecha de su entrenamiento.
- **Tipo de usuario:** Se indica si el asistente interactuará con niños (NI) o docentes y adultos (DA).

- **Límite de palabras:** Se establece un límite de palabras para las respuestas del asistente (por ejemplo, 100 palabras).
- **Tono y estilo:** Se define el tono y estilo de comunicación del asistente de acuerdo con la personalidad del personaje.

Actualización de los "Prompts" con Make

Para facilitar la gestión de los 80 "Prompts", utilizaremos una hoja de cálculo de Google Sheets. En esta hoja, tendremos una lista de todos los personajes con sus respectivos "Prompts". Para actualizar los "Prompts" de manera eficiente, utilizaremos Make para conectar la hoja de cálculo con la plataforma de OpenAI Assistants.

Ejemplo: Actualización del "Prompt" de Walt Disney

Imaginemos que queremos actualizar el "Prompt" del asistente que interpreta a Walt Disney. Para ello, modificaremos el "Prompt" en la hoja de cálculo y activaremos una casilla de verificación. Make detectará este cambio y actualizará automáticamente el "Prompt" en OpenAI Assistants.

Automatización con Make: una gran ayuda

Este proceso de actualización automática nos ahorra mucho tiempo y esfuerzo. No tenemos que modificar cada "Prompt" individualmente en la plataforma de OpenAI Assistants. Además, podemos utilizar fórmulas sencillas de Excel para concatenar texto e información en los "Prompts", lo que facilita la creación y edición de los mismos.

Conexión de Excel con OpenAI Assistants mediante Make

Para conectar la hoja de cálculo con OpenAI Assistants, crearemos un escenario en Make. Este escenario ejecutará una función que activará un Webhook en Make cada vez que se modifique la hoja de cálculo.

Funcionalidad del Escenario:

- **Webhook:** Se activa al modificar la hoja de cálculo.
- **Identificación de cambios:** Make identifica si se ha modificado un solo personaje o todos los asistentes.
- **Actualización de "Prompts":** Make actualiza los "Prompts" en OpenAI Assistants según los cambios en la hoja de cálculo.

Actualización masiva de "Prompts"

Además de la actualización individual de "Prompts", Make nos permite actualizar todos los "Prompts" de manera masiva con una sola operación.

Ejemplo de Actualización Masiva:

1. **Borrado de valores:** Make borra todos los valores en un rango de la hoja de cálculo que contiene la lista de asistentes.
2. **Obtención de datos:** Make realiza una llamada de API a OpenAI para obtener la lista completa de asistentes.
3. **Acumulación de información:** Make utiliza un iterador para acumular la información de cada asistente.
4. **Unificación de datos:** Make utiliza un "Text Aggregator" para unir la información de todos los asistentes en una sola variable de texto.
5. **Actualización en la hoja:** Make actualiza la hoja de cálculo con la información de todos los asistentes en una sola operación.

Eficiencia y Rentabilidad

Este proceso de actualización masiva nos permite ahorrar mucho tiempo, especialmente cuando trabajamos con una gran cantidad de asistentes. Además, la eficiencia en la gestión de los "Prompts" nos ayuda a optimizar el rendimiento de la plataforma y a utilizar los recursos de manera más rentable.

Make: una herramienta poderosa para API Assistants

Make no solo es un intermediario entre OpenAI Assistants y los usuarios, sino que también es una herramienta poderosa para configurar y gestionar los "Prompts" de manera eficiente.

Recomendación:

Les animo a explorar las diferentes funciones de Make y a experimentar con la configuración de API Assistants. Make les ofrece una gran flexibilidad para automatizar tareas y optimizar el funcionamiento de sus proyectos con OpenAI.