

Asistente de API



Alejandra Caggiano

API Assistant es una funcionalidad de GeneXus Enterprise AI que permite definir asistentes para chatear e interactuar con cualquier API REST externa que cumpla con la especificación OpenAPI 3.0 o superior.

Asistente de API

Este asistente aprovecha los modelos de inteligencia artificial y las capacidades de llamada de funciones para proporcionar una forma dinámica de chatear con APIs utilizando lenguaje natural.

Este asistente aprovecha los modelos de inteligencia artificial y las capacidades de llamada de funciones para proporcionar una forma dinámica de chatear con APIs utilizando lenguaje natural.

Asistente de API

- Compatibilidad con OpenAPI 3.0
- Instrucciones personalizables
- Configuración del Modelo
- Seguridad
- Llamadas a funciones automatizadas:
- Múltiples llamadas a distintas APIs.

Las características claves son las siguientes:

Compatibilidad con OpenAPI 3.0: Es posible interactuar con cualquier API que esté construida siguiendo la especificación OpenAPI 3.0 o posterior.

También las Instrucciones personalizables: el usuario puede proporcionar instrucciones de texto específicas para guiar al asistente de inteligencia artificial en la toma de decisiones y el manejo preciso de las solicitudes.

Configuración del modelo: es posible ajustar la configuración del modelo de inteligencia artificial, como ser la temperatura y el número máximo de tokens generados, para mejorar y personalizar las respuestas del asistente según las necesidades del usuario.

Seguridad: Garantiza la seguridad de las interacciones mediante el uso de API tokens. Estos tokens, que pueden ser de tipo Basic o Bearer, se utilizan en el encabezado de Autorización para autenticar y autorizar solicitudes, protegiendo así la información y los datos.

También las llamadas a funciones automatizadas: devuelve respuestas de manera eficiente al realizar solo las llamadas necesarias a la API. Esto se logra gracias al modelo inteligente de GeneXus Enterprise AI y a las llamadas a funciones automatizadas que manejan las solicitudes de los usuarios finales.

Y también las múltiples llamadas a distintas APIs: La inteligencia del asistente se muestra realizando llamadas secuenciales a diferentes API para obtener resultados complejos. Por ejemplo, al recibir una pregunta como '¿Cómo está el clima en Tokio y Montevideo?' el asistente identifica la necesidad de consultar la API meteorológica de cada ciudad mencionada. Esta capacidad de manejar automáticamente múltiples solicitudes garantiza una experiencia perfecta para el usuario, que no necesita intervenir en el proceso.

Asistente de API

Requisitos previos

- Configurar las solicitudes.

Por ejemplo:

```
-H "Authorization: Bearer $OPENAI_API_KEY"
```

- Verificar que la especificación de la API esté documentada de acuerdo a OpenAPI 3.0 o superior, en formato JSON.

Bien. Antes de comenzar a utilizar este tipo de asistente, es importante asegurarse de que se cumplan ciertos requisitos previos:

Si la API con la que se desea interactuar está autenticada con un API token (ya sea basic o bearer), se debe configurar las solicitudes para incluir este token en el encabezado de autorización, especificando el tipo correcto.

```
-H "Authorization: Bearer $OPENAI_API_KEY"
```

De esta forma, al ejecutar la llamada, el token será utilizado para la interacción.

También se debe verificar que la especificación de la API esté documentada según OpenAPI 3.0 o superior, en formato JSON. Esto es importante, ya que es posible que las APIs que no cumplan con esta especificación no sean totalmente compatibles.

Bien. Ahora sí, veamos un ejemplo

Asistente de API

Ejemplo: Weather API

Obtener la especificación OpenAPI 3.0 para la API con la cual se desea interactuar.

```
{
  "openapi": "3.0.0",
  "info": {
    "title": "Weather API",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "https://weathergpt.vercel.app",
      "description": "Test"
    }
  ],
  "host": "https://weathergpt.vercel.app",
  "paths": {
    "/api/weather": {
      "get": {
        "summary": "Get current weather information",
        "operationId": "checkWeatherUsingGET",
        "parameters": [
          {
            "name": "location",
            "in": "query",
            "required": true,
            "description": "Location for which to retrieve weather information.",
            "schema": {
              "type": "string"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Current weather information",
            "content": {
              "application/json": {
                "schema": {
                  "type": "object",

```

Ln 1, Col 1 7.330 caracteres.

Lo primero que debemos hacer es obtener las especificaciones de OpenAPI3.0 para la api con la cual queremos interactuar, que en nuestro caso será la **Weather API**.

Asistente de API

The screenshot shows the 'Assistant Api Editor' interface. On the left, the 'New Assistant' section is active, with a 'General Information' sub-section. The 'Api Description (Instructions)' field contains the following text: 'The final response must strictly comply with the following: It must be in the same language as the user's request. The response text must be formatted as a Table, with 2 columns and rows. In case it exists, it must return the LINK to view more details of the Weather. Draw an emoji that roughly describes the weather condition, in the Condition column along with the description.' On the right, the 'Authentication' section shows 'Authentication Type' set to 'Api Key'. The 'LLM Settings' section shows 'Provider' set to 'openai', 'Model' set to 'gpt-3.5-turbo-16k', 'Temperature' set to '0.10', and 'Max output Tokens' set to '1024'.

Ingresamos entonces al Backoffice de GeneXus Enterprise AI, seleccionamos el proyecto sobre el cual vamos a trabajar y en el menú seleccionamos Assistants, Create API assistant

Se nos pide una descripción para la API. Aunque se trata de algo opcional, es útil que el asistente comprenda el propósito y la funcionalidad de la API con la cual se desea interactuar. Es posible incluir cualquier instrucción que se considere relevante para guiar al asistente de manera efectiva.

Por ejemplo, para la Weather API, puede ser algo como lo siguiente:

The final response must strictly comply with the following:

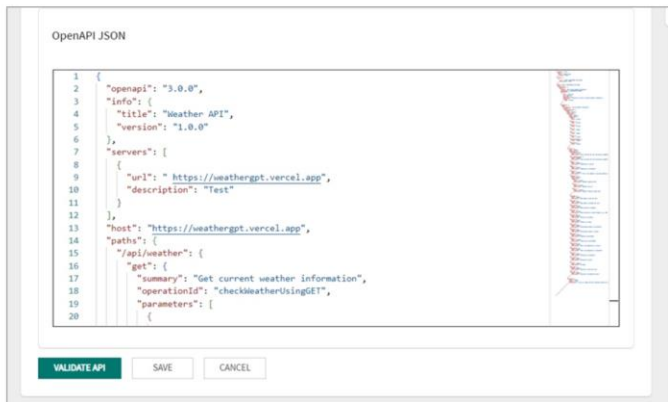
It must be in the same language as the user's request.

The response text must be formatted as a Table, with 2 columns and rows.

In case it exists, it must return the LINK to view more details of the Weather.

Draw an emoji that roughly describes the weather condition, in the Condition column along with the description.

Asistente de API



```
{
  "openapi": "3.0.0",
  "info": {
    "title": "Weather API",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "https://weathergpt.vercel.app",
      "description": "Test"
    }
  ],
  "host": "https://weathergpt.vercel.app",
  "paths": {
    "/api/weather": {
      "get": {
        "summary": "Get current weather information",
        "operationId": "checkWeatherUsingGET",
        "parameters": [
          {
            "name": "location",
            "in": "query",
            "required": true,
            "description": "Location for which to retrieve weather information.",
            "schema": {
              "type": "string"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Current weather information",
            "content": {
              "application/json": {
                "schema": {
                  "type": "object",

```

Ln 1, Col 1 7,330 caracteres

El paso siguiente es completar el archivo OpenAPI JSON con la especificación OpenAPI 3.0 o superior.

Aquí es donde se deben especificar los detalles técnicos de la API, incluyendo rutas, métodos HTTP, parámetros de consulta, cuerpos de solicitud y respuesta, entre otros. Este archivo actúa como una especificación detallada de la API, lo que permite una interacción precisa y consistente con el Asistente.

La opción Validate api realiza una verificación básica del json para verificar si está formado correctamente y si tiene algún servicio definido. También comprueba si la URL del servicio está correctamente definida.

Vale mencionar que esta verificación no ejecuta una prueba contra la api.

Asistente de API

The screenshot shows the 'Assistant Api Editor' interface. On the left, under 'New Assistant', there is a 'General Information' section with an 'Api Description (Instructions)' text area containing the following text: 'The final response must strictly comply with the following: It must be in the same language as the user's request. The response text must be formatted as a Table, with 2 columns and rows. In case it exists, it must return the LINK to view more details of the Weather. Draw an emoji that roughly describes the weather condition, in the Condition column along with the description.' On the right, there are two configuration panels. The 'Authentication' panel has 'Authentication Type' set to 'Api Key' (selected over 'None'). The 'LLM Settings' panel has 'Provider' set to 'openai', 'Model' set to 'gpt-3.5-turbo-16k', 'Temperature' set to '0.10', and 'Max output Tokens' set to '1024'. The top right corner shows 'Project' set to 'TrainingProject (Training)'.

El paso siguiente es configurar la Autenticación

En este paso, es posible definir si la API requiere autenticación y, de ser así, qué tipo de autenticación se utilizará. GeneXus Enterprise AI ofrece soporte para autenticación basada en ApiKeys, las cuales pueden ser Basic o Bearer.

Si la API no requiere autenticación, se debe seleccionar la opción None. De lo contrario, debe establecer el valor de la API key.

Esta clave única brinda acceso exclusivo a la API.

Además, se deberá especificar el Tipo de autenticación, donde se deberá seleccionar entre los valores Basic o Bearer.

El siguiente paso consiste en personalizar las configuraciones relacionadas con el modelo de lenguaje para adaptarlo a las necesidades específicas y obtener un rendimiento óptimo.

Proveedor: Determina que el proveedor del modelo de lenguaje es OpenAI.

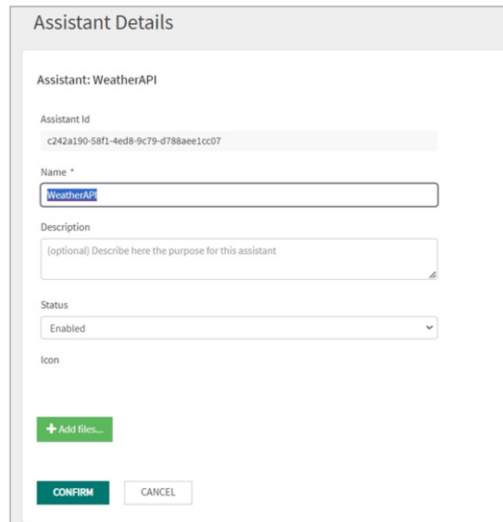
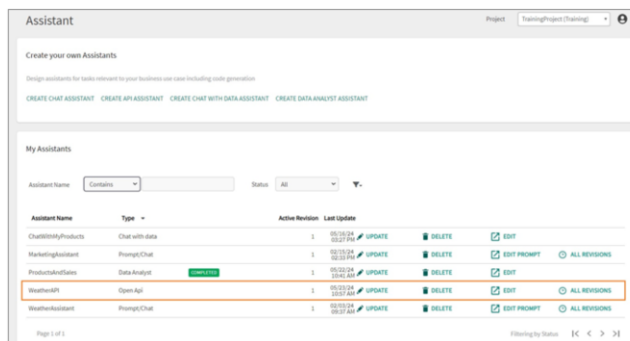
Modelo: Se debe seleccionar el modelo de idioma específico que desea utilizar. Puede elegir entre una variedad de opciones, desde GPT-3.5 hasta GPT-4, que tienen sus propias características y capacidades únicas.

Temperatura: Ajusta la temperatura del modelo para controlar la creatividad y diversidad en las respuestas generadas. Los valores más bajos producen respuestas más conservadoras, mientras que los valores más altos fomentan la creatividad. El valor predeterminado es 0,10.

Máxima salida de Tokens: Define el número máximo de tokens permitidos en cada respuesta. Esta configuración determina la longitud máxima de las respuestas generadas por el modelo de lenguaje. El valor predeterminado es 1024.

Es importante tener en cuenta que estas configuraciones son opcionales y pueden dejarse con sus valores predeterminados.

Asistente de API



Presionamos Save.

Esto abre una ventana donde podemos ingresar el nombre y la descripción de la API. También podemos agregar un archivo para el icono.

Una vez completada toda esta información presionamos Confirm.

Vemos el asistente creado, y si presionamos Update, podemos ver el identificador de versión con la que se guardó, podemos cambiar el nombre, la descripción, configurarlo como habilitado o deshabilitado, y agregar un ícono si se desea.

Asistente de API

Assistant

Project: TrainingProject (Training)

Create your own Assistants

Design assistants for tasks relevant to your business use case including code generation

CREATE CHAT ASSISTANT CREATE API ASSISTANT CREATE CHAT WITH DATA ASSISTANT CREATE DATA ANALYST ASSISTANT

My Assistants

Assistant Name: Contains Status: All

Assistant Name	Type	Active Revision	Last Update			
ChatWithProducts	Chat with data	1	05/23/24 09:27:01	UPDATE	DELETE	EDIT
MarketingAssistant	PromptChat	1	05/23/24 09:29:04	UPDATE	DELETE	EDIT PROMPT ALL REVISIONS
ProductsAndSales	Data Analyst	1	05/22/24 19:42:04	UPDATE	DELETE	EDIT
WeatherAPI	Open Api	1	05/23/24 10:57:01	UPDATE	DELETE	EDIT ALL REVISIONS
WeatherAssistant	PromptChat	1	05/23/24 09:27:01	UPDATE	DELETE	EDIT PROMPT ALL REVISIONS

Page 1 of 1

WeatherAPI Assistant

Project: TrainingProject (Training)

Go to list of Assistants

Revision Id	Revision Name	Provider Name	Model Name	Model Type	Time Stamp		
2	2	openai	gpt-3.5-turbo-16k	Chat	05/23/24 11:00 AM	EDIT/TEST	SET AS ACTIVE
1	1	openai	gpt-3.5-turbo-16k	Chat	05/23/24 10:57 AM	EDIT/TEST	

Page 1 of 1

Si en cambio, se selecciona Edit, es posible administrar y mantener diferentes versiones.

Podemos sobrescribir la versión existente, o crear una nueva versión con los cambios realizados. Para esto debemos seleccionar Save as a new revision.

De esta manera, podemos controlar y organizar las modificaciones según sea necesario.

Desde aquí podemos ver todas las versiones del Asistente.

Asistente de API

compare please the weather in Montevideo and Buenos Aires

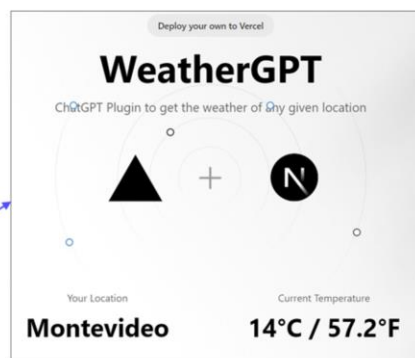
Here is a comparison of the weather between Montevideo and Buenos Aires:

Location	Condition	Description
Montevideo	 Moderate rain	Moderate rain
Buenos Aires	 Partly cloudy	Partly cloudy

You can find more details about the weather in Montevideo [here](#) and the weather in Buenos Aires [here](#).

Regenerate response

Type a message



Finalmente, probar el nuevo asistente desde la opción Playground del menú

Consultamos, por ejemplo, el estado actual del tiempo en Venecia...

Y queremos comparar el estado actual del tiempo en Montevideo y Buenos Aires

GeneXusTM
by **Globant**

training.genexus.com