

Globant ▶
Enterprise AI



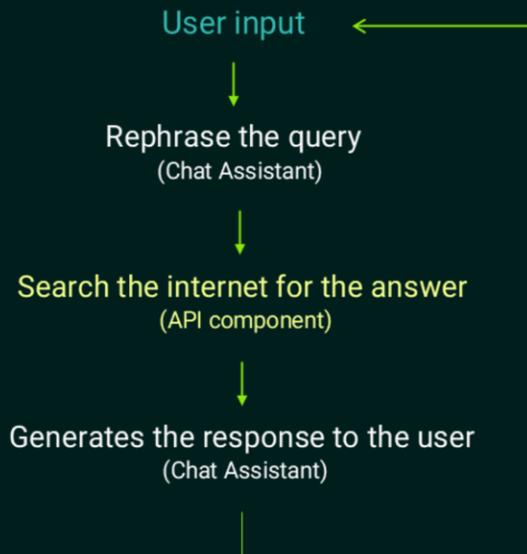
Flow with Internet search



Alejandra Caggiano

Vamos a crear un nuevo flujo que gestiona dos chat assistants, e incluye un componente API de búsqueda.

Flow with Internet search



Uno de los asistentes responderá preguntas mediante resultados de búsqueda en internet, y el otro será capaz de reformular las preguntas, basadas en el historial de conversaciones, como si fueran preguntas independientes.

Por su parte el componente API realizará la búsqueda en internet, utilizando un nuevo motor de búsqueda y será el nexo entre ambos asistentes.

Flow with Internet search

ResponseFromSearch assistant

Prompt

INSTRUCTIONS

Your task is to generate a response for the user based on the results of an internet search.
DO NOT USE PRIOR KNOWLEDGE AND DO NOT MAKE UP INFORMATION USE THE SEARCH RESULTS.
Given the following results from an internet search and NOT PRIOR KNOWLEDGE answer the user question the best you can.

SEARCH RESULTS

{searchResults}

RESPONSE FORMAT

For the response you can generate RICH TEXT USING HTML if it is needed.
DO NOT USE MARKDOWN TO FORMAT YOUR RESPONSE.
FOR RICH TEXT DO NOT USE MARKDOWN USE HTML INSTEAD.

LLM Settings

Provider

Google VertexAI

Model

gemini-1.5-pro

Temperature

0.10

Max output Tokens

8192



Accedemos al backoffice, y en nuestro proyecto vemos los dos asistentes creados: Observemos el prompt del asistente ResponseFromSearch. Establece que este asistente deberá generar una respuesta para el usuario basándose en los resultados de una búsqueda en internet.

NO DEBE USAR CONOCIMIENTOS PREVIOS NI INVENTAR INFORMACIÓN, DEBE USAR LOS RESULTADOS DE LA BÚSQUEDA.

Con los resultados de una búsqueda en internet y SIN CONOCIMIENTOS PREVIOS, responderá a la pregunta del usuario lo mejor posible. El resultado se guardará en la variable SearchResults, y para la respuesta, puede generar texto enriquecido con HTML si es necesario.

Flow with Internet search

FollowUpRephrase assistant

Prompt

Given the conversation history and not prior knowledge, and a follow-up question, rephrase the follow-up question to be a standalone question IF IT IS NEEDED. If the question works as a standalone question, just repeat the question.

RESPONSE ONLY WITH THE STANDALONE QUESTION AND NOTHING MORE;
DO NOT INCLUDE ANY OTHER TEXT

LLM Settings

Provider

Google VertexAI

Model

gemini-1.5-pro

Temperature

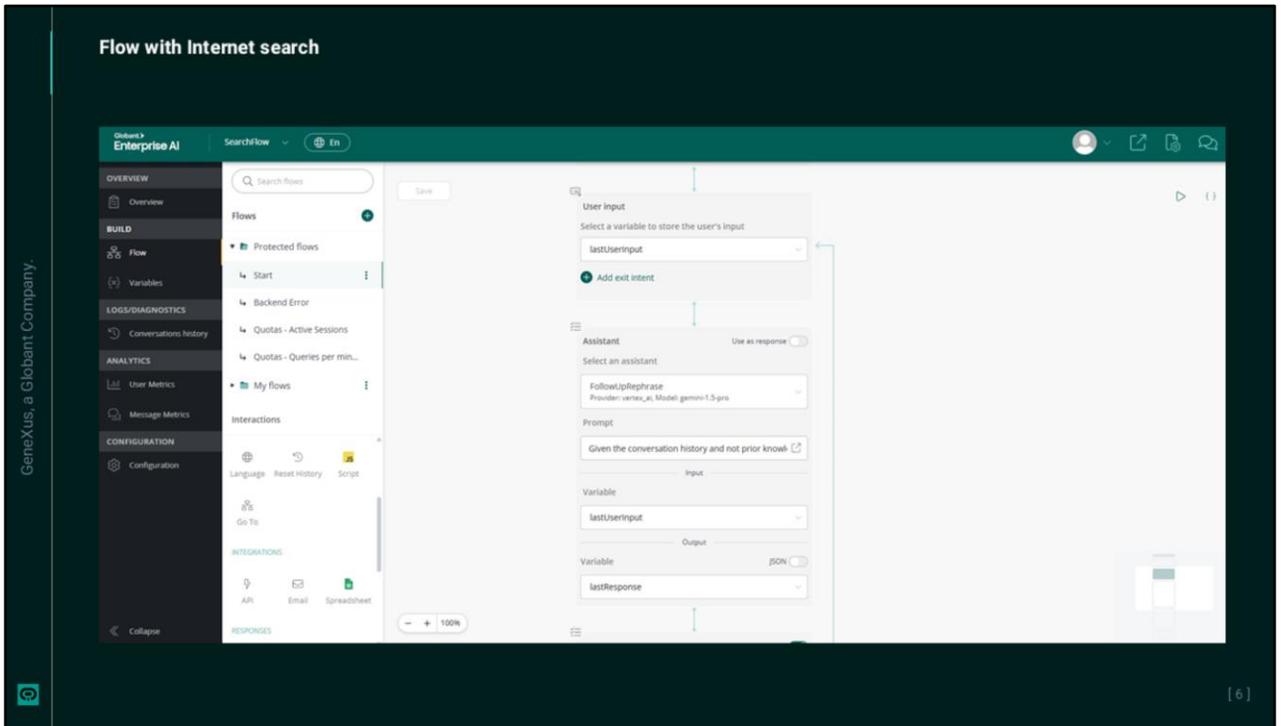
0.10

Max output Tokens

8192

Observemos ahora la definición del otro chat assistant de nombre FollowUpRephrase.

Establece que deberá reformular una pregunta teniendo en cuenta el historial de la conversación y no los conocimientos previos. Deberá responder solamente con la pregunta independiente, sin incluir ningún otro texto.



Bien. Comencemos a definir el flujo.

Accedemos a Flows, y creamos un nuevo flujo de nombre InternetSearchFlow. Agregamos una descripción e indicamos los idiomas español e inglés.

Luego del nodo UserInput que es quien recibe la consulta del usuario, conectamos con el asistente FollowUpRephrase. Este asistente recibe la consulta en la variable lastUserinput, pero su respuesta será guardada en una nueva variable. Así que desmarcamos esta opción e indicamos la variable lastResponse en la salida.

Este asistente devolverá una pregunta cuya respuesta será buscada a través de un motor de búsqueda, para lo cual necesitamos un componente API que permita accederlo y actúe como nexo entre ambos asistentes.

Vamos a las interacciones, seleccionamos API y arrastramos hasta su posición en el flujo.

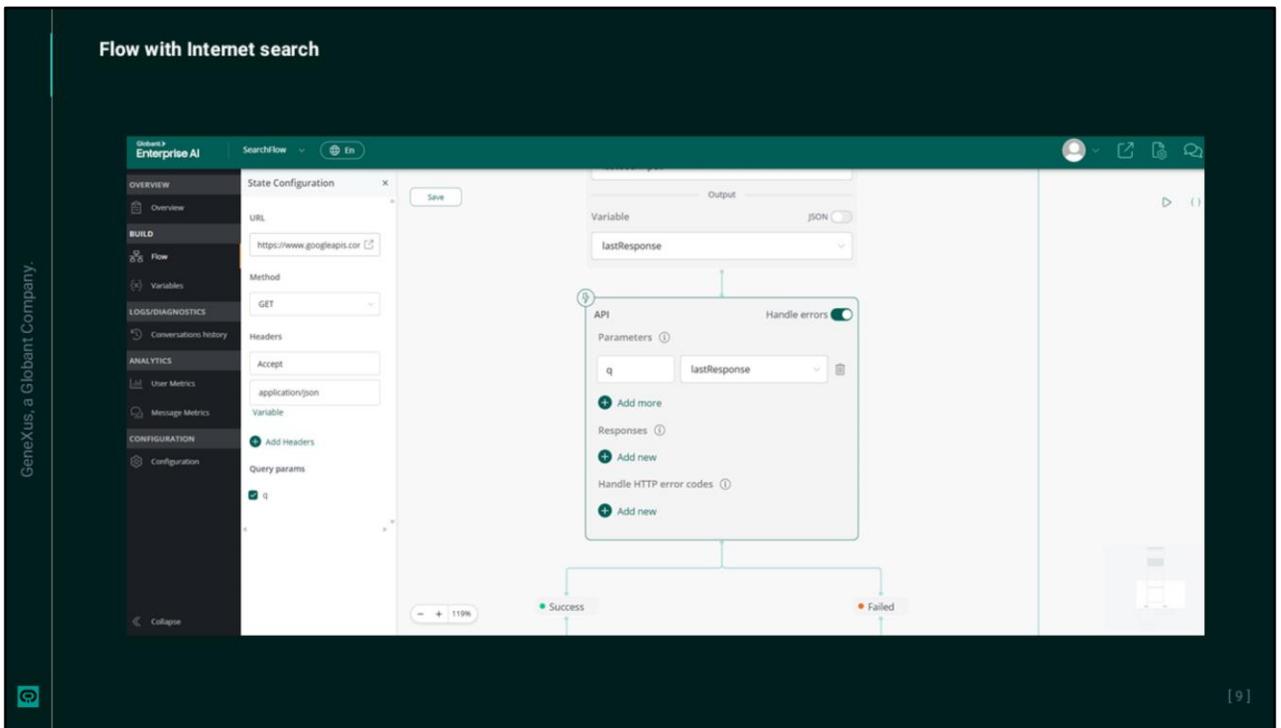
Flow with Internet search

The screenshot shows a web page titled "Custom Search JSON API: Introduction" from the Programmable Search Engine documentation. The page has a blue header with navigation links: Home, Guides, Reference, and Support. A left sidebar contains a "Filter" input and a list of categories: More Callback Examples, Look and Feel, Structured Data, JSON API (with sub-items: Overview, Introduction, Using REST, Performance Tips, Libraries and Samples, Site Restricted JSON API), and Advanced Topics. The main content area has a breadcrumb trail: Home > Products > Programmable Search Engine > Guides. The page title is "Custom Search JSON API: Introduction" with a "Send feedback" button. The text on the page includes: "This document will help you to get familiar with Custom Search JSON API and its usage." followed by sections: "Before you start", "Create Programmable Search Engine" (with a paragraph explaining the need for an existing instance and a link to the Control Panel), and "Identify your application to Google with API key" (with a paragraph explaining the need for an API key and a "Get a Key" button).

Para poder configurar la información necesaria debemos definir la API de búsqueda, y para eso vamos a crear un nuevo motor de búsqueda programable.

El primer paso es obtener la API key, así que accedemos al sitio ([Google Search API documentation](#)) y presionamos Get Key.

Seleccionamos o creamos un proyecto. La clave que se nos muestra debemos guardarla porque la necesitaremos para el componente API en el flujo.



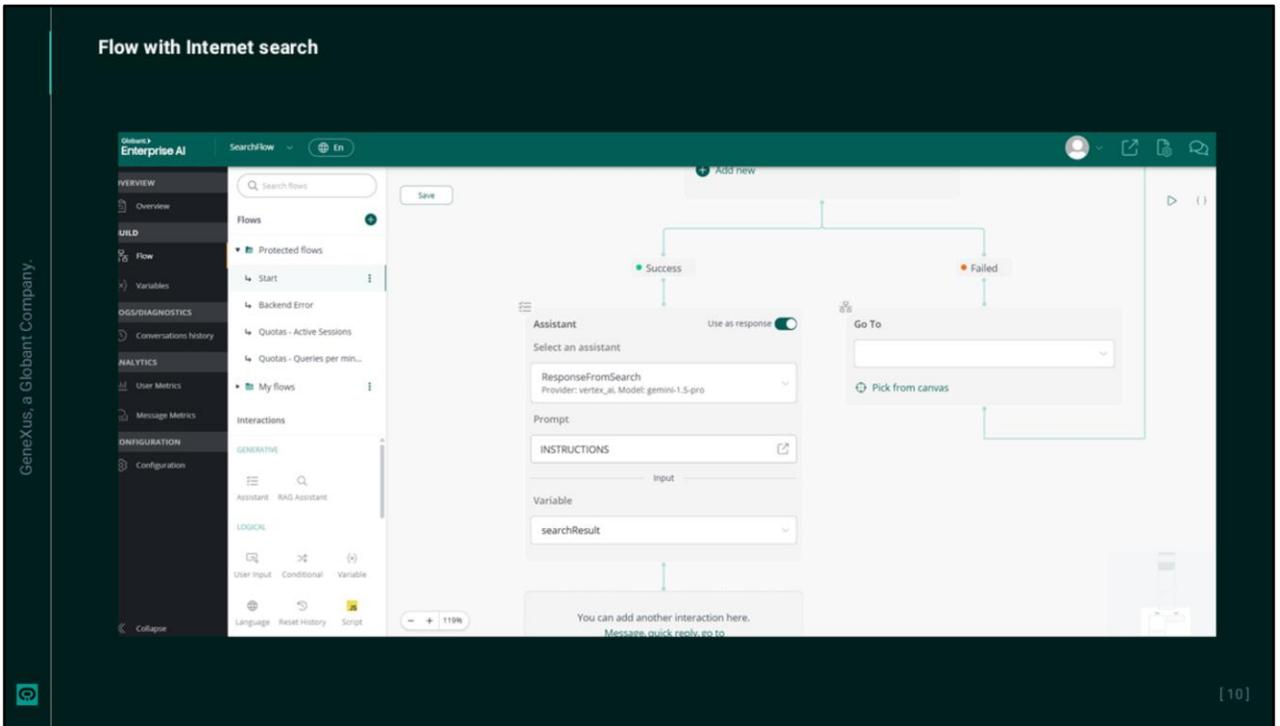
Bien. Ya tenemos todo lo necesario para continuar con la definición del flujo, así que volvemos y al costado del nodo API, seleccionamos Edit.

En el menú de la izquierda, debemos indicar la URL, teniendo en cuenta de referenciar los valores obtenidos para la la API key y el Id del motor de búsqueda

En el nodo API, vamos a la sección Parameters, Add new, y configuramos el parámetro con la siguiente información. Como nombre del parámetro ponemos "q", e indicamos lastResponse como valor, ya que será el valor obtenido de la respuesta anterior. Luego en el menú de la izquierda marcamos "q" como parámetro de la consulta.

En la sección Response del nodo, asociamos la propiedad items a la variable searchResults. Si la variable no existe, la creamos.

Bien. Una vez definido el nodo API, debemos asegurarnos de que el flujo siga su curso y que la secuencia de las variables sea correcta. El nodo API devuelve el resultado de la búsqueda en la variable searchResults indicada como respuesta. Así que esta variable deberá ser la entrada en el asistente ResponseFromSearch, que será quien devuelva la respuesta al usuario.



Así que, desde el cuadro de interacciones, arrastramos otro nodo Assistant e indicamos al asistente ResponseFromSearch. La entrada de este asistente será la variable searchResults recibida a través del nodo API.

Bien. Agregamos ahora el correspondiente nodo Go to para volver al nodo UserInput.

Hemos completado entonces la definición del flujo, así que guardamos los cambios, y estamos en condiciones de probar su comportamiento.

Flow with Internet search

GeneXus, a Globant Company.

Globant >
Enterprise AI

INTERNETSEARCHFLOW

Holds a conversation by rephrasing the user's questions and searching the internet for answers.

InternetSearchFlow

Specializes in digital transformation, using engineering, innovation, design, and AI solutions to help organizations.

how many offices does the company have?

Globant is a publicly traded IT and software development company with headquarters in Luxembourg. They have offices in many locations around the world, including a new Innovation Center in Malaga, Spain.

how many offices?

In a 2019 blog post, Globant stated that they had over 40 offices worldwide.

[11]

Vamos a la página de prueba.

Preguntamos ¿Qué es Globant? ¿Cuántas oficinas tiene?

Y así podemos continuar la conversación.

Globant ▶
Enterprise AI