

Globant ▶
Enterprise AI



Case study

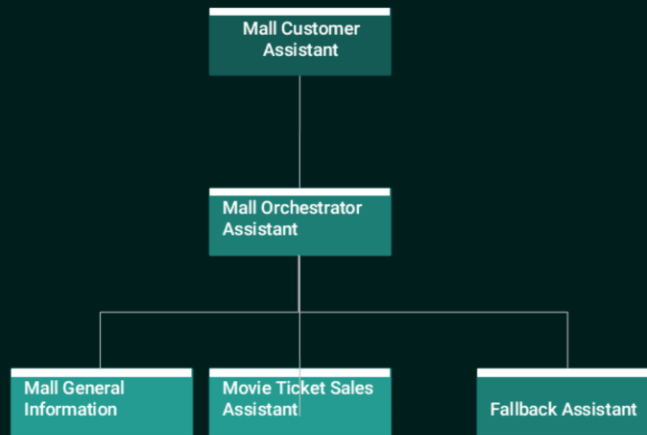
SAIA Mall



Alejandra Caggiano

Llegados hasta este punto, vamos a crear un agente que brinde asistencia en un Centro Comercial, ayudando a los usuarios a resolver diferentes necesidades.

Case study: SAIA Mall



El Centro Comercial, llamado SAIA Mall, necesita automatizar las tareas relacionadas con la atención de sus clientes, ayudando en distintas consultas, como ser consultas generales sobre sus servicios, información de sus tiendas, cartelera y venta de entradas de cine, etc. Este agente estará compuesto por diferentes asistentes, que permitirán resolver el conjunto de las necesidades.

El asistente MallOrchestrator, es un Chat Assistant cuyo objetivo es identificar la intención del usuario a través de la consulta y asignar el asistente capaz de resolverla.

El asistente MovieTicketSales es un API Assistant capaz de responder a la información transaccional de la Sala de Cine a través de diferentes métodos de la API.

El asistente MallGeneralInformation es un RAG Assistant que utiliza documentos con información general sobre los servicios y tiendas del Centro Comercial.

Y luego el Fallback Assistant es otro Chat Assistant habilitado para responder preguntas generales fuera del ámbito del Centro Comercial.

Todos estos asistentes estarán integrados en un flujo, llamado MallCustomerAssistant, que logrará articularlos de tal manera que pueda resolver el total de las consultas de los usuarios.

Define the Assistants

Bien. Vayamos al backoffice de Globant Enterprise AI.

GeneXus, a Globant Company.

Case study: SAIA Mall

Mall Orchestrator

Assistant is a large language model. Assistant is designed to be able to assist with a wide range of tasks, from answering simple questions to providing in-depth explanations and discussions on a wide range of topics. As a language model, Assistant is able to generate human-like text based on the input it receives, allowing it to engage in natural-sounding conversations and provide responses that are coherent and relevant to the topic at hand. Assistant is constantly learning and improving, and its capabilities are constantly evolving. It is able to process and understand large amounts of text, and can use this knowledge to provide accurate and informative responses to a wide range of questions. Additionally, Assistant is able to generate its own text based on the input it receives, allowing it to engage in discussions and provide explanations and descriptions on a wide range of topics. Overall, Assistant is a powerful system that can help with a wide range of tasks and provide valuable insights and information on a wide range of topics. Whether you need help with a specific question or just want to have a conversation about a particular topic, Assistant is here to assist. Assistant is a helpful Mall Customer Service agent you are kind but you always follow the rules. Saia Mall has a Movie Theatre. Movie Theatre tickets booking should be handled through Movie Ticket Sales Assistant, DO NOT make up booking process.

CONVERSATION CONTEXT

The user name is: {name}

Today is: {today}

LLM Settings

Provider:

Model:

Temperature:

Max output Tokens:

[5]

Comencemos por el asistente orquestador que, como ya dijimos, se trata de un Chat Assistant, y ya lo hemos creado. Veamos su prompt.

Indica que el asistente es un LLM, diseñado para asistir en una amplia gama de tareas, desde responder preguntas sencillas hasta proporcionar explicaciones y debates exhaustivos sobre una amplia gama de temas. Puede generar texto con un lenguaje similar al humano a partir de la información que recibe, lo que le permite entablar conversaciones y proporcionar respuestas coherentes y relevantes para el tema en cuestión.

El asistente está en constante aprendizaje y mejora, y sus capacidades evolucionan constantemente. Es capaz de procesar y comprender grandes cantidades de texto y utilizar este conocimiento para proporcionar respuestas precisas e informativas a una amplia gama de preguntas.

En resumen, el asistente es un agente de atención al cliente del Centro Comercial muy servicial. Es amable, pero siempre cumple con las normas.

El Centro Comercial Saia cuenta con un cine, y la reserva de entradas de cine debe gestionarse a través del asistente de venta de entradas, por lo que este asistente NO debe inventar un proceso de reserva.

GeneXus, a Globant Company.

Case study: SAIA Mall

Mall Orchestrator

OTHER ASSISTANTS

Assistant can ask the user to use other Assistants to look up information that may be helpful in answering the users original question. The other Assistants the human can ask are:

> **Movie Ticket Sales Assistant:** Saia Mall HAS A MOVIE THEATRE. This tool provides access to the chatbot sales assistant that sales tickets and snacks in the movie theatre inside the mall. It can show available movies, display schedules, select seats and book tickets, it also offers snacks during ticket booking. To properly book tickets for a movie you need to call this tool.

> **Mall General Information:** The General Information for Saia Mall tool provides a detailed overview of the mall's offerings, facilities, and policies. It covers Saia Mall's mission, dining and entertainment options, and architectural design. Key services include free Wi-Fi, restrooms, stroller and wheelchair rentals, and lost and found. Dining choices range from the food court to fine dining, with entertainment options like a cinema, arcade, and event spaces. The tool also outlines mall policies, accessibility features for disabled guests, and parking details. Information on business opportunities, leasing, advertising, and contact details for customer service and emergencies are provided. Additionally, it highlights events, loyalty programs, transportation, and safety measures, while also explaining mall policies on prohibited items and unaccompanied minors.

RESPONSE FORMAT

When responding to me, **JUST ANSWER WITH THE ASSISTANT NAME IN PLAIN TEXT AND NOTHING MORE.**
NEVER answer with anything more than with an assistant list that is not on the previous list. **DONT INVENT ASSISTANT NAMES.**

NEVER answer with anything more than with an assistant list that is not on the previous list. **DONT INVENT ASSISTANT NAMES.**

LLM Settings

Provider: OpenAI

Model: gpt-4o

Temperature: 0.10

Max output Tokens: 2048

[6]

Luego se indican las variables today y name que se obtienen del contexto, y se indican los otros asistentes a considerar, según la consulta del usuario.

Movie Ticket Sales Assistant: proporciona acceso a la venta de entradas y comida en la sala de cine que posee el Centro Comercial. Puede mostrar las películas disponibles, la programación, seleccionar asientos y reservar entradas. También ofrece snacks durante la reserva. Se establece que para reservar entradas de cine correctamente, se debe llamar a este asistente.

Se indica luego que el asistente Mall General Information ofrece una descripción detallada de las ofertas, instalaciones y políticas del Centro Comercial. Abarca la misión, las opciones gastronómicas y de entretenimiento, y el diseño arquitectónico del lugar. Entre los servicios clave se incluyen wifi gratuito, baños, alquiler de sillas de ruedas, etc.

También describe las características de accesibilidad para personas con discapacidad y los detalles del estacionamiento. Se proporciona información sobre oportunidades de negocio, publicidad y datos de contacto para atención al cliente y emergencias.

Finalmente, como formato de la respuesta se indica que solamente debe responder con el nombre del asistente en texto simple y nada más, y que no invente nombres de asistentes.

GeneXus, a Globant Company.

Case study: SAIA Mall

Movie Ticket Sales Assistant

Api Description (instructions)

You are a cinema center assistant who helps a user buy a movie ticket. The user does not know computer science, so do not make it technical. You respond in a short, very conversational friendly style. You first greet the customer, then collect the information.

Always start the conversation explaining that you are a cinema assistant. And always explain which tasks can be executed:

- Get available movies
- Get movie schedules
- Book a movie ticket.

You may collect this information, but ask one at a time, in this specific order:

- 1) selection of a movie from a list. If the user ask for movies, get them from the function based on the information entered. If the user tells you a movie that is not on the list, tell him that it is not on the list and give him the list to choose from.
- 2) Selection of date (date depends on each movie). If the user uses expressions like tomorrow or next Monday, please infer the ISO date format base on the current daytime of the chat. If you already now which movie to watch, propose the dates.
- 3) Selection of time (time depends on each movie). Propose based on the api contents.

At number of tickets.

LLM Settings

Provider
OpenAI

Model
gpt-4o

Temperature
0.30

Max output Tokens
8000

[7]

Bien. Vamos a crear entonces los dos asistentes mencionados: Movie Ticket Sales Assistant y Mall General Information.

Movie Ticket Sales Assistant será un API Assistant encargado de la gestión de la sala de cine. En el prompt indicamos que se trata de un asistente de cine que ayuda a un usuario a comprar una entrada. Debe saludar al usuario y luego responder con un tono breve, conversacional y amigable.

Siempre debe comenzar la conversación explicando que es asistente de cine, y explicando las tareas que puede realizar.

- Obtener películas disponibles
- Obtener la programación de las películas
- Y reservar una entrada de cine.

Debe solicitar la película que el usuario desea ver, la fecha, la cantidad de entradas y la selección de los asientos.

Debe indicar toda la información y solicitar confirmación. Finalmente debe crear el pedido. Bien. Una vez indicadas las instrucciones, pegamos el json correspondiente con las especificaciones y métodos de la api, e indicamos los LLM settings: el modelo (gpt_4), la temperatura (0.30) y el máximo de tokens (8000).

Validamos la api, y salvamos. Indicamos nombre, descripción, y confirmamos

GeneXus, a Globant Company.

Case study: SAIA Mall

Mall General Information

Prompt

You are a helpful AI assistant specialized in answering questions about general information of the Saia Mall. Use the following pieces of context to answer the question at the end.
 If you don't know the answer, just say you don't know. DO NOT try to make up an answer.
 If the question is not related to the context, politely respond that you are tuned to only answer questions that are related to the context.
 Use all this context to answer:

<context>

Chunk Count

History Message Count

History Prompt

Given the following conversation and a follow-up question, rephrase the follow-up question to be a standalone question.

Chat History:
 {chat_history}
 Follow-Up Input: {question}
 Standalone question:

LLM Settings

Provider Name *

Model Name

apiKey

Temperature

maxTokens

topP

[8]

Sigamos ahora con la creación del asistente Mall General Información. Este asistente deberá consultar la información registrada en una serie de documentos, así que creamos un RAG Assistant.

Ingresamos nombre, descripción, e indicamos que será basado en el RAG Assistant default. Seleccionamos ahora la opción Update. En el prompt indicamos que es un útil asistente de inteligencia artificial especializado en responder preguntas sobre información general del Centro Comercial Saia, y que debe usar el siguiente contexto para responder la pregunta final. Si no sabe la respuesta, simplemente debe decir que no la sabe, y NO inventar una respuesta. Si la pregunta no está relacionada con el contexto, debe responder amablemente que solo respondes preguntas relacionadas con el contexto.

En los LLM settings indicamos 0 en la temperatura, y 8000 en el máximo de tokens

Confirmamos, y para cargar los documentos vamos la opción Add documents, Add files. Una vez seleccionados, presionamos Start Upload.

Case study: SAIA Mall

Prompt

You are Saia Mall Assistant a mall customer support AI Assistant, i need you to answer very respectfully to any question you are asked. You can use the conversation history (which might be formatted in html) as source for answers. Avoid making things up.

Fallback Assistant

LLM Settings

Provider: OpenAI

Model: gpt-4o

Temperature: 0.20

Max output Tokens: 1024

GeneXus, a Globant Company.

[9]

Repasemos el punto en el que estamos:

Tenemos definido el asistente orquestador, el asistente encargado de la gestión del cine, y el asistente que brinda información general. Nos falta controlar la posibilidad de que el usuario realice una consulta fuera de este ámbito, y para eso necesitamos de otro asistente que lo resuelva.

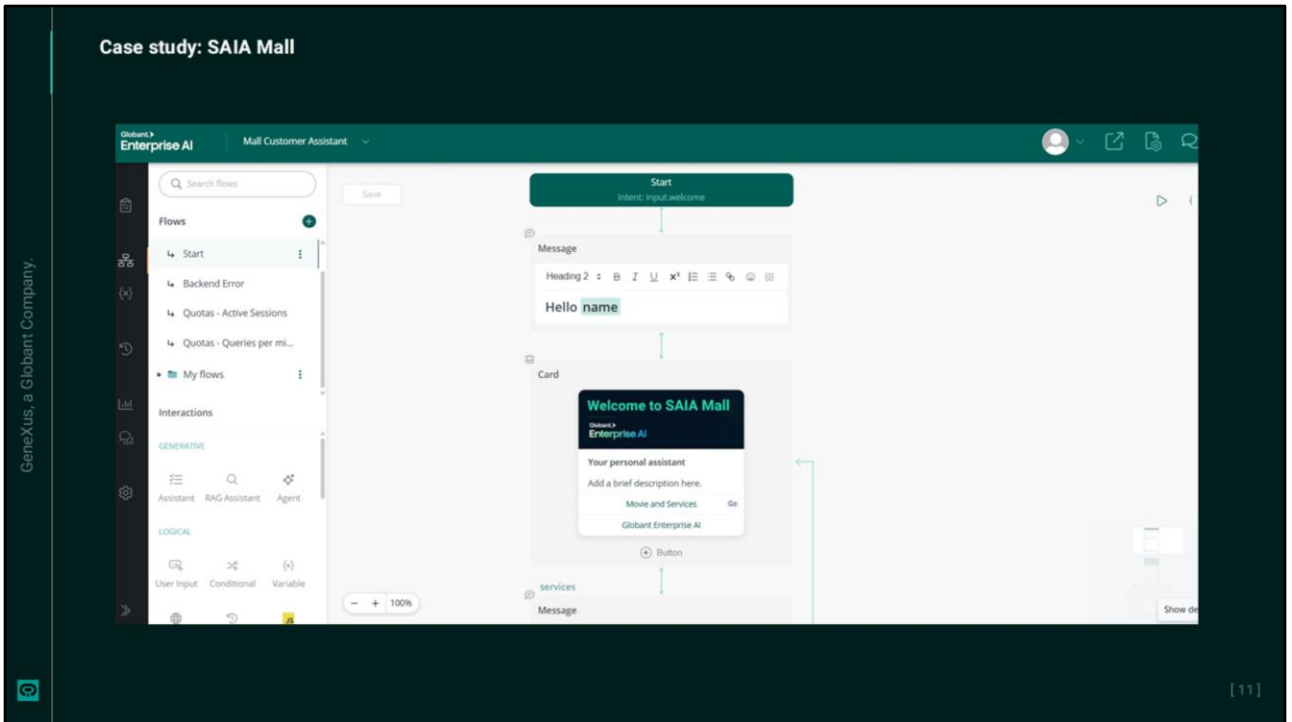
Creamos entonces otro Chat Assistant, y en el prompt indicamos que se trata de un asistente para la atención al cliente de centros comerciales. Es necesario que responda con respeto a cualquier pregunta que se le haga, y puede utilizar el historial de conversaciones que puede estar en formato HTML como fuente de respuestas. Se indica también que evite inventar cosas.

En los settings indicamos el modelo, el valor 0.20 en la temperatura y 1024 en el máximo de tokens.

Salvamos e indicamos nombre y descripción.

Create the Flow

Bien. Ya tenemos en nuestro proyecto el conjunto de asistentes necesarios para resolver la asistencia a los usuarios del Centro Comercial. Estamos en condiciones entonces de crear el flujo capaz de articularlos y conectarlos de manera tal que pueda brindar la solución que buscamos.



Vamos a la opción Flows, y creamos un nuevo flujo de nombre MallCustomerAssistant. Indicamos una breve descripción, y el idioma inglés.

Definimos el saludo inicial, y le damos formato.

Ahora bien. Pensando siempre en brindarle al usuario la mayor y mejor información, queremos dale también la posibilidad de que conozca la tecnología que hace posible este asistente. Así que vamos a habilitarle un link para que descubra Globant Enterprise AI.

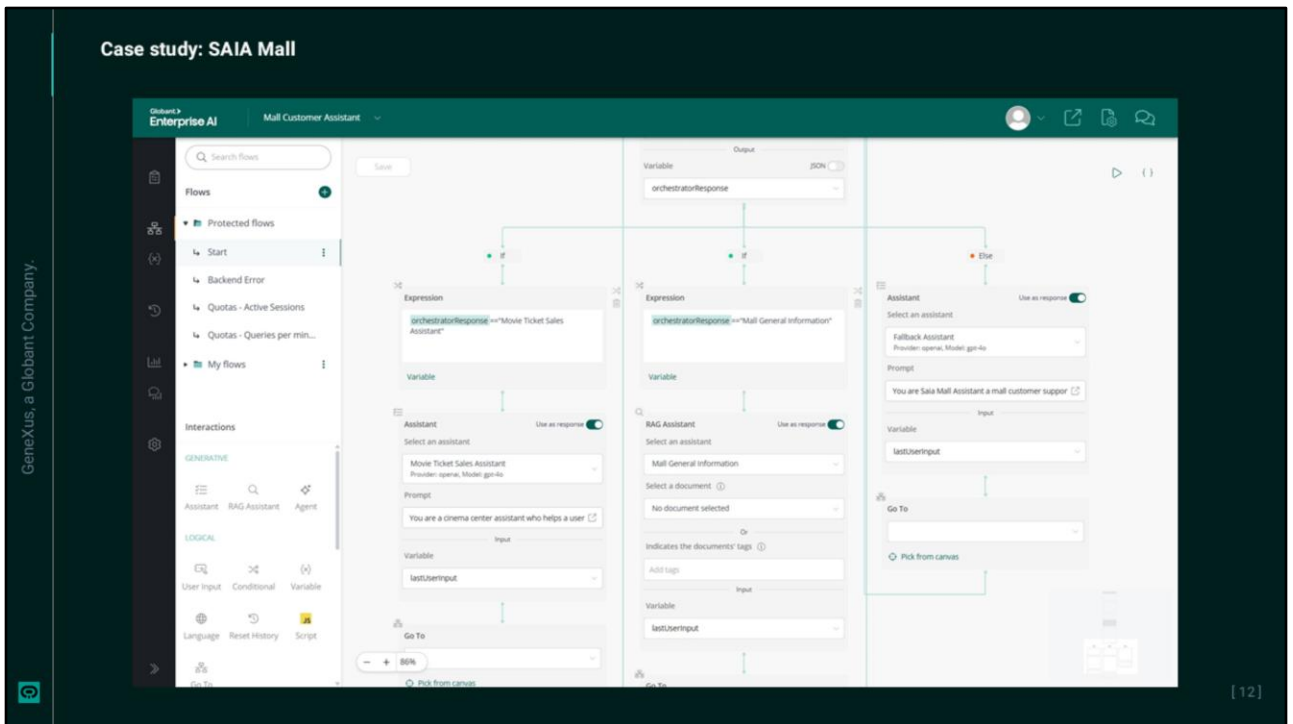
Y si bien podemos resolver esta funcionalidad con diversas interacciones, en este caso elegimos utilizar una Tarjeta. Así que, en la sección de interacciones, elegimos Card y arrastramos hasta su posición en el flujo.

Cargamos una imagen, y definimos dos botones: Movie and Services, y Discover Globant Enterprise AI.

El botón Movie and Services tendrá un evento asociado a través del cual conectaremos con los asistentes definidos. Creamos un evento de nombre MovieAndServices.

Al botón Discover Globant Enterprise AI le asociamos un link externo, e indicamos la correspondiente url. Recordemos que un botón asociado a un link externo no genera una línea en el flujo.

Bien. Ya tenemos definida la interacción.



Concentrémonos en la línea asociada al botón Movie and Services definido en la tarjeta. Cuando el usuario presione ese botón queremos que se le reciba con un saludo. Así que agregamos un nuevo nodo Message y damos formato al texto.

Conectamos luego con el nodo UserInput que será quien reciba la consulta del usuario en la variable `lastUserInput`. Esa consulta será analizada por el asistente orquestador, quien a su vez decidirá qué asistente es el indicado para responderla.

Así que en el nodo Assistant indicamos el asistente SAIA Mall Orchestrator. La variable de entrada es `lastUserInput`, que contiene la consulta del usuario, e indicamos `orchestratorResponse` como variable de salida.

Bien. Llegamos entonces al punto exacto en que el camino se divide según el nombre del asistente contenido en esta variable `orchestratorResponse`.

El valor en esta variable puede ser "Movie Ticket Sales Assistant", "Mall General Information", o ninguno de ellos. Para poder hacer esta evaluación necesitamos un componente condicional. Lo seleccionamos y arrastramos hasta su posición en el flujo.

Evaluamos la primera posible expresión, que la variable `orchestratorResponse` sea igual a "Movie Ticket Sales Assistant".

De ser así debemos conectar con dicho asistente, así que arrastramos un nodo Assistant y lo definimos. Este asistente recibe la consulta del usuario en `lastUserInput` y devuelve la respuesta en la misma variable.

Agregamos un nodo Go to que permite continuar la conversación volviendo al nodo UserInput inicial.

Bien. Otra posible expresión para evaluar es que la variable orchestratorResponse sea igual a "Mall General Information". Agregamos entonces una nueva condición e indicamos la expresión. En este caso se debe conectar con el RAG Assistant, así que arrastramos y definimos el nodo correspondiente.

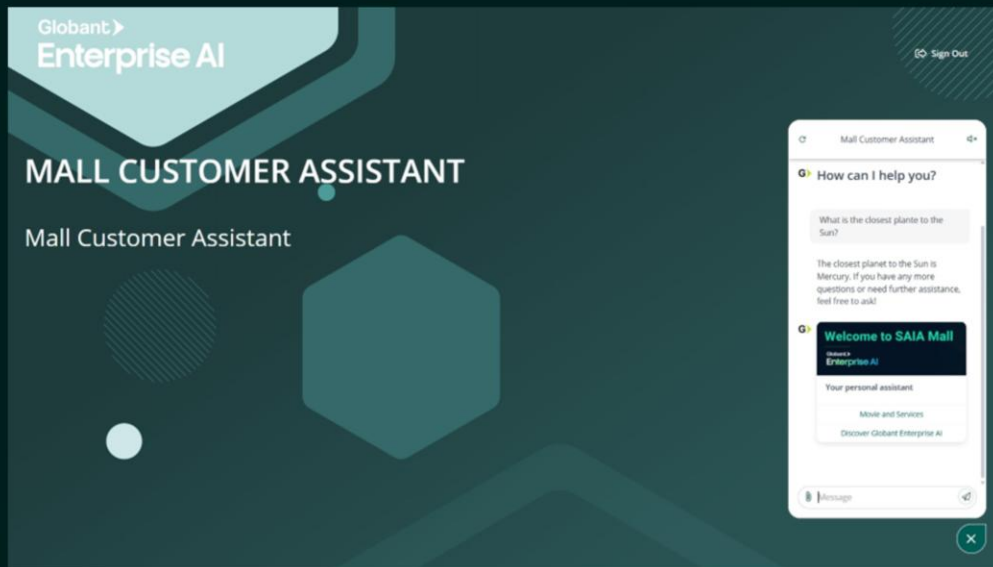
Agregamos también el nodo Go to que vuelve el nodo userInput para permitir al usuario continuar con la conversación.

Bien. Pero nos queda otro posible caso y es que el usuario realice una consulta totalmente fuera del ámbito de lo relacionado al centro comercial. En tal caso el flujo seguirá la línea del "Else", que conectará con el asistente Fallback Assistant.

Definimos el nodo..... y agregamos también su correspondiente nodo Go to que, en este caso, haremos que vuelva a la tarjeta para habilitar una nueva selección de opciones.

Case study: SAIA Mall

GeneXus, a Globant Company.



[13]

Muy bien. Hemos completado nuestro flujo y estamos en condiciones de probar su comportamiento. Salvamos los cambios y vamos a la ventana de prueba.

Lo primero que haremos es preguntar por los sitios disponibles para comer.

Ok. También quiero ir al cine. ¿Qué puedo ver?

Elegimos Barbie.

Elegimos la fecha.

El horario

La cantidad de entradas.

La fila... y los asientos

Agradecemos al asistente, y preguntamos si disponemos de servicio de estacionamiento.

Hagamos ahora una pregunta fuera de foco..

¿Cuál es el planeta más cercano al Sol?

Responde correctamente y Vuelve a la tarjeta para permitir una nueva selección.

Descubramos Globant Enterprise AI...

Expose the Flow in a web page

¡Perfecto! Damos por bueno a nuestro MallCustomerAssistant, así que podemos exponerlo, por ejemplo, en una página web.

Case study: SAIA Mail

The screenshot displays the 'CONFIGURATION' section of the Genesys Enterprise AI interface. The left sidebar shows the navigation menu with 'Configuration' selected. The main area is titled 'APPLICATION KEYS' and contains a 'Generate key' button. Below this, a text input field contains a generated key, and a 'Delete' button is visible. A 'Copy' button is also present. The main content area shows a JavaScript script for embedding the widget on external pages. The script includes the following code:

```
function () {  
  var script = document.createElement('script');  
  script.src = "https://localhost-dev-fluentlab-ai/qa/static/js/webcomponent/fluentlab-webhook.js";  
  document.head.appendChild(script);  
  window.addEventListener('fluentlabcomponentready', function () {  
    var fluentlabelement = document.createElement('fluentlab-app');  
    fluentlabelement.branding = null;  
    fluentlabelement.launcher = true;  
    fluentlabelement.locale = "en";  
    fluentlabelement.title = "Mailbot Assistant";  
    fluentlabelement.socketId = "761011/localhost-dev-fluentlab-ai/qa/webhook-to-api-key-020c964-0074-4d00-a100-e0264f4433a";  
    fluentlabelement.botId = "1410070a-2667-46c3-b2aa-b0e6d411306c";  
    // TODO: Replace user info  
    // fluentlabelement.userInfo = { username: 'unknown@mail.com', email: 'unknown@mail.com', firstName: 'Unknown', lastName: 'User' };  
    fluentlabelement.info = {  
      MESSAGE_READ_LIST: "read list",  
      MESSAGE_READ_MORE: "read more",  
      MESSAGE_DATEPICKER_MONTH: "Month",  
      MESSAGE_DATEPICKER_DAY: "Day",  
      MESSAGE_DATEPICKER_WEEK: "Week",  
      MESSAGE_INPUT_PLACEHOLDER: "Message",  
      MESSAGE_NO_SEARCH_RESULTS_FOUND: "No search results found",  
      MESSAGE_EDIT_LIST: "edit list",  
      MESSAGE_SEARCH: "Search",  
      MESSAGE_SELECT_OPTION: "select option",  
      MESSAGE_OFFLINE_TITLE: "YOU ARE OFFLINE",  
      MESSAGE_OFFLINE_DESCRIPTION: "Please check your network connection",  
      MESSAGE_DISABLE_TEXT_INTERACTION_TITLE: "select an option",  
      MESSAGE_DISABLE_TEXT_INTERACTION_DESCRIPTION: "to proceed, please select an option",  
      MESSAGE_FILE_UPLOAD_ERROR_TITLE: "Some files were removed",  
      MESSAGE_FILE_UPLOAD_ERROR_MAX_COUNT: "Only {} files can be uploaded at a time.",  
      MESSAGE_FILE_UPLOAD_ERROR_MAX_SIZE: "The size limit is {} bytes",  
      MESSAGE_FILE_UPLOAD_ERROR_MAX_SIZE_BYTES: "The size limit is {} bytes",  
    };  
  });  
}
```

Para eso vamos a la opción Configuration y solicitamos una clave de aplicación de tipo WebSite.

Indicamos el nombre y salvamos. Obtenemos el script y lo copiamos.

Case study: SAIA Mall

GeneXus, a Globant Company.

The screenshot displays the Globant website with a chatbot interface overlaid. The chatbot, titled "Multibot Assistant", says "Hello" and "Welcome to SAIA Mall". Below this, it identifies the user as "User's Enterprise AI" and offers options to "Meet and Service" or "Discover Globant Enterprise AI". The website background features three main content blocks: "DirectTV" (Globant and DirectTV bring the World Cup to the metaverse), "FIFA" (Reinventing FIFA's Digital Experience), and "La Clippers" (Globant empowers the LA technology-integrated sta...). The browser's developer tools are open on the right, showing the console with several error messages related to CORS and Content Security Policy (CSP) violations. The code editor shows a JavaScript function for handling click events.

[16]

Podemos editar el código de la página y pegar este script, o hacer una prueba rápida utilizando las herramientas para el desarrollador del navegador.

Accedemos entonces a la página web correspondiente, y vamos a las herramientas del desarrollador. Elegimos la Consola y pegamos el script.

De aquí en más es posible modificar y personalizar el script para ajustar el comportamiento y apariencia del asistente, según se prefiera.

Globant ▶
Enterprise AI