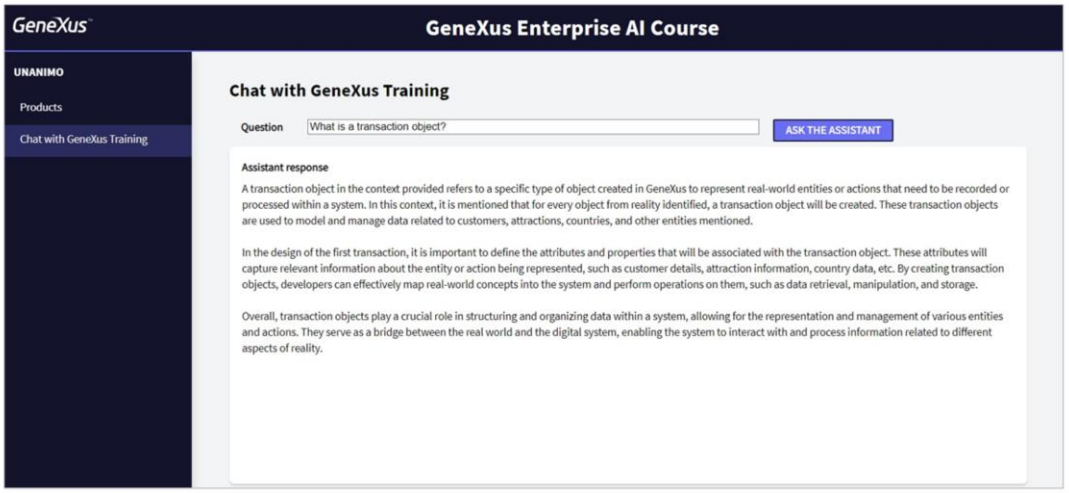
GeneXus by Globant

Aplicación GeneXus que interactúa con un RAG assistant

Alejandra Caggiano

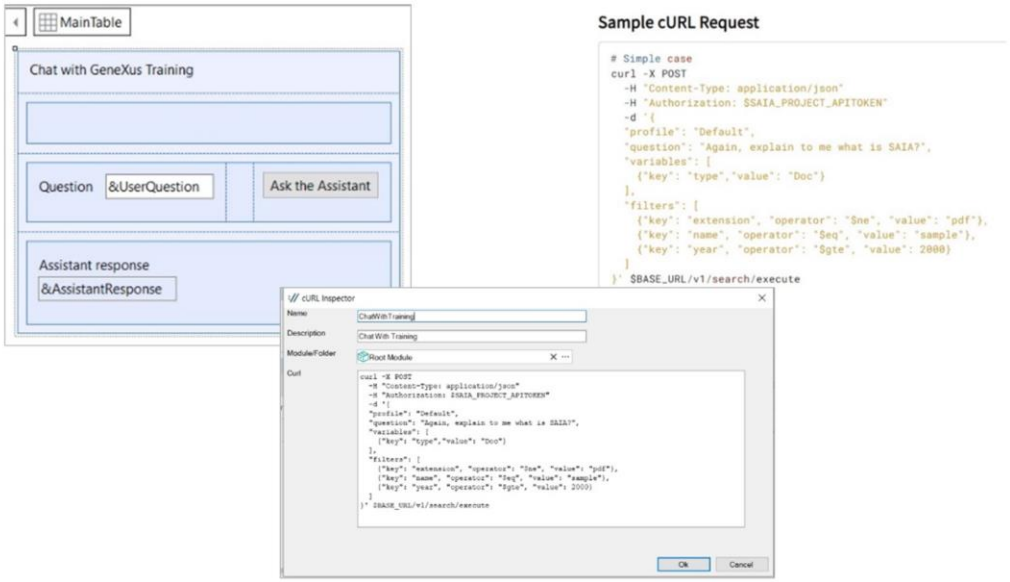
Hemos visto anteriormente cómo interactuar desde una base de conocimiento GeneXus con un Chat assistant.

Aplicación GeneXus que interactúa con un RAG assistant



Vamos a ampliar ahora esa misma base de conocimiento para que también pueda interactuar con nuestro RAG Assistant de nombre ChatWithGXTraining. De esta forma habilitaremos al usuario final a realizar preguntas sobre dicho contenido.

Aplicación GeneXus que interactúa con un RAG assistant



Si vamos a GeneXus, ya hemos creado el web panel de nombre ChatGXTraining., donde el usuario final puede ingresar una consulta y preguntar al asistente para obtener la respuesta. Tenemos dos variables:

La variable &UserQuestion, sobre la cual el usuario planteará su consulta, y la variable &AssistantResponse que será la encargada de mostrar la respuesta recibida.

Nuestro objetivo entonces es codificar el evento asociado al botón para que al presionarlo se llame a un procedimiento que envíe la consulta y reciba la respuesta.

Para crear ese procedimiento ya con la base de la estructura del request, vamos a cargar el cURL sample a través del menú Tools / Application Integration / cURL Inspector.

Ponemos como nombre ChatWithTraining y pegamos el sample, disponible en la documentación técnica de GeneXus Enterprise AI.

https://wiki.genexus.com/enterprise-ai/wiki?8,Table+of+contents%3AEnterprise+AI,

En este caso necesitamos interactuar con la api que nos permite chatear con documentos, así que el sample que pegamos corresponde al cURL Request.

Aplicación GeneXus que interactúa con un RAG assistant

Presionamos Ok, y vemos creado el procedimiento con la estructura base que necesitamos.

Lo primero que indicamos es el protocolo de conexión. Y para eso, como yo sabemos, declaramos el método Secure con el valor 1 que corresponde al protocolo HTTPS.

Luego indicamos el Host que corresponde al contenido de la variable &BASE_URL, y que depende a su vez de la instalación de GeneXus Enterprise AI.

Bien. Necesitamos luego indicar el Project api token, que como también ya sabemos, lo copiamos de la plataforma y lo pegamos.

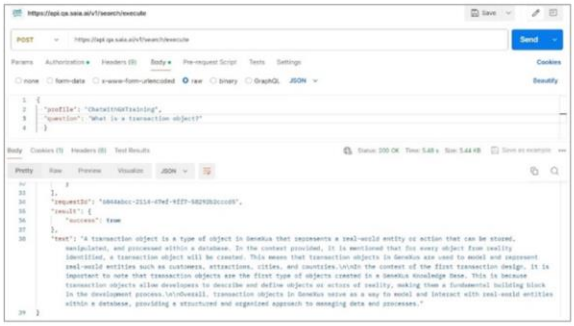
Vamos ahora al cuerpo de la consulta. En primer lugar, indicamos el “profile”, o sea, el nombre del asistente que en este caso es ChatWithGXTraining.

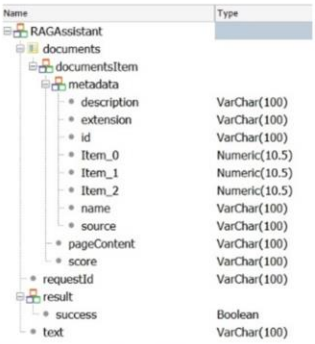
Al igual que analizamos en el ejemplo anterior, este procedimiento deberá recibir la consulta del usuario final para enviársela al asistente y que se obtenga la respuesta.

Así que declaramos la correspondiente regla Parm, donde el parámetro de entrada es la pregunta del usuario, y el parámetro de salida, o retorno, es la respuesta del asistente, Podríamos también indicar filtros a la consulta, pero no lo haremos en este caso.

Bien. Luego indiciamos la ejecución del POST, completando el path

Aplicación GeneXus que interactúa con un RAG assistant






```
//curl -X POST  -H "Content-Type: application/json"  -H "Authorization: $SAIA_PROJECT_APITOKEN"  -d '{  "profile": "Defaul

&HttpClient.Secure = 1
&HttpClient.Host = "api.qa.saia.ai"

&HttpClient.AddHeader(!"Authorization", !"Bearer default_OuK5BwzqSLjNEHwUf-GV0QCLI2YHYxBBGAshAg1CiLSa9lFgeZKcQr5S0xsehbjbg
&HttpClient.AddHeader(!"Content-Type", !"application/json")

&HttpClient.AddString(!'{ "profile": "ChatWithGXTraining", "question": "' + &UserQuestion.Trim() + '" }')

&HttpClient.Execute(!"POST", "/v1/search/execute")

&RAGAssistant.FromJson(&HttpClient.ToString())

&AssistantResponse = &RAGAssistant.text
```

Hasta acá entonces el envío de la consulta. Ahora debemos recibir la respuesta del asistente. Si recordamos la salida que vimos desde Postman cuando testeamos este asistente via api, vemos que esta salida corresponde a una estructura, y la respuesta del asistente queda en el item “text”.

Así que la salvamos como un archivo json, para luego importarlo en la KB y generar automáticamente el tipo de dato estructurado que necesitamos para recibir la respuesta.

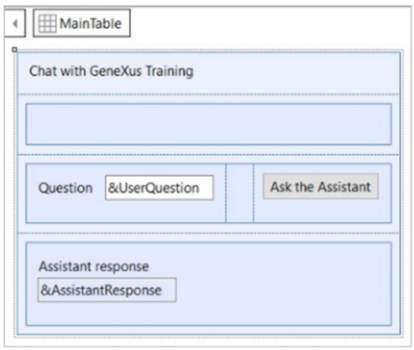
Volvemos entonces a la KB y lo importamos. Vamos a Tools / Aplication Integration / Json import.

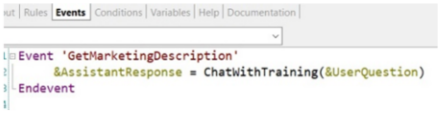
Ponemos como nombre RAGAssistant y cargamos el archivo. Vemos que se creó el tipo de dato estructurado. Lo abrimos, y observamos el item “text”, donde recibiremos la respuesta del asistente.

¿Qué debemos hacer entonces? Debemos crear una variable basada en este tipo de dato y cargar allí la respuesta del asistente. En el procedimiento, definimos entonces la variable &RAGAssistant y cargamos la respuesta from Json. Ya tenemos la respuesta completa, pero lo que necesitamos es el contenido del item “text. Así que a nuestra variable de retorno, &AssistantResponse, la cargamos con ese contenido.

Bien. Ya hemos completado el procedimiento.

Aplicación GeneXus que interactúa con un RAG assistant





Ahora debemos completar su llamada desde el web panel. Para eso, vamos al evento asociado al botón y llamamos al procedimiento, enviándole como parámetro la variable &UserQuestion con la pregunta del usuario, y recibiendo su respuesta en la variable &AssistantResponse, ambas presentes en el form.

Para probarlo, presionamos F5, y ejecutamos el web panel.

Aplicación GeneXus que interactúa con un RAG assistant

Recordemos que nuestro RAG assistant fue alimentado solamente con 18 documentos, lo cual no es mucha información, y por ese motivo haremos preguntas simples cuyas respuestas se encuentren en el alcance de esos documentos. Esto nos permitirá testear la aplicación.

En primer lugar, Bvamos a preguntar: ¿Qué es una base de conocimiento GeneXus? Veamos la respuesta.

Bien. Preguntemos ahora: ¿Qué es un objeto transacción?

Muy bien. Y si preguntamos ¿qué es un atributo fórmula en GeneXus?

Perfecto. Las respuestas recibidas fueron prolijas y correctas. Lo damos por válido.

Ahora observemos el menú. Tenemos acceso al asistente para chat y al asistente para chatear con documentos.

De esta forma entonces hemos logrado una aplicación GeneXus que interactúa con dos asistentes de inteligencia artificial creados a través de GeneXus Enterprise AI.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com