

GeneXus application that interacts with a RAG assistant



Alejandra Caggiano

We have previously seen how to interact with a Chat assistant from a GeneXus knowledge base.

GeneXus application that interacts with a RAG assistant

The screenshot displays the GeneXus Enterprise AI Course interface. On the left, a dark sidebar contains the GeneXus logo and a navigation menu with items: UNANIMO, Products, and Chat with GeneXus Training (which is highlighted). The main content area is titled "GeneXus Enterprise AI Course" and "Chat with GeneXus Training". It features a "Question" input field containing the text "What is a transaction object?" and a blue "ASK THE ASSISTANT" button. Below the input, the "Assistant response" is displayed in a white box with a light border. The response text is as follows:

Assistant response

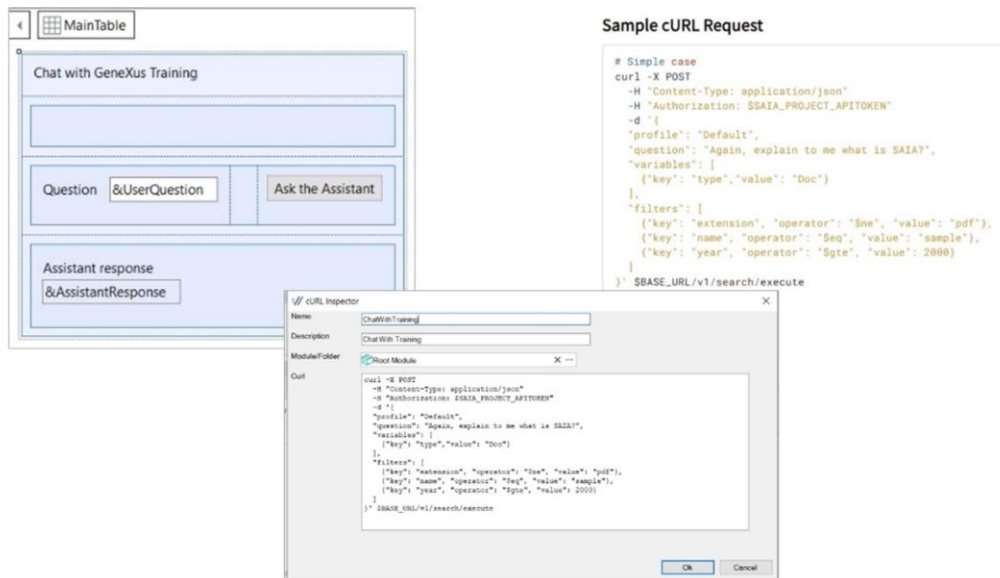
A transaction object in the context provided refers to a specific type of object created in GeneXus to represent real-world entities or actions that need to be recorded or processed within a system. In this context, it is mentioned that for every object from reality identified, a transaction object will be created. These transaction objects are used to model and manage data related to customers, attractions, countries, and other entities mentioned.

In the design of the first transaction, it is important to define the attributes and properties that will be associated with the transaction object. These attributes will capture relevant information about the entity or action being represented, such as customer details, attraction information, country data, etc. By creating transaction objects, developers can effectively map real-world concepts into the system and perform operations on them, such as data retrieval, manipulation, and storage.

Overall, transaction objects play a crucial role in structuring and organizing data within a system, allowing for the representation and management of various entities and actions. They serve as a bridge between the real world and the digital system, enabling the system to interact with and process information related to different aspects of reality.

We are now going to extend that same knowledge base so that it can also interact with our RAG Assistant named ChatWithGXTraining. This way we will enable the end user to ask questions about this content.

GeneXus application that interacts with a RAG assistant



If we go to GeneXus, we have already created the web panel named ChatGXTraining where the end user can enter a query and ask the assistant for the answer. We have two variables:

The &UserQuestion variable, on which the user will submit their query, and the &AssistantResponse variable that will be in charge of displaying the answer received.

Our objective then is to code the event associated with the button so that when it is pressed a procedure is called to send the query and receive the response.

To create this procedure that has the base of the request structure, we are going to load the cURL sample through the menu Tools / Application Integration / cURL Inspector.

We name it ChatWithTraining and paste the sample, available in the GeneXus Enterprise AI technical documentation.

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

In this case, we need to interact with the API that allows us to chat with documents, so the sample that we paste corresponds to the cURL Request.

GeneXus application that interacts with a RAG assistant

```
you | Rules * | Conditions | Variables | Help | Documentation |
|
| Param(in:&UserQuestion, out:&AssistantResponse);
|
|
| 1 //curl -X POST -H "Content-Type: application/json" -H "Authorization: $SAIA_PROJECT_APITOKEN" -d '{ "profile": "Default
| 2
| 3 &HttpClient.Secure = 1
| 4 &HttpClient.Host = "api.qa.saia.ai"
| 5
| 6 &HttpClient.AddHeader("Authorization", "Bearer default_OuK5BwzqSLjNEHwUf-GV0QCLi2YHYxBBGAshAg1CilSa9lFgeZKcQr5S0xsehbjbg
| 7 &HttpClient.AddHeader("Content-Type", "application/json")
| 8
| 9 &HttpClient.AddString('{"profile": "ChatWithGXTraining", "question": "' + &UserQuestion.Trim() + '" }')
| 10
| 11 &HttpClient.Execute("POST", "/v1/search/execute")
| 12
```

We click on OK, and we see the procedure created with the base structure we need. The first thing we indicate is the connection protocol. And for that, as we know, we declare the Secure method with the value 1 that corresponds to the HTTPS protocol.

Then we indicate the Host corresponding to the content of the &BASE_URL variable, which in turn depends on the GeneXus Enterprise AI installation.

Good. Next, we need to indicate the Project Api Token, which as we already know, we copy it from the platform and paste it.

Let's go to the body of the query.

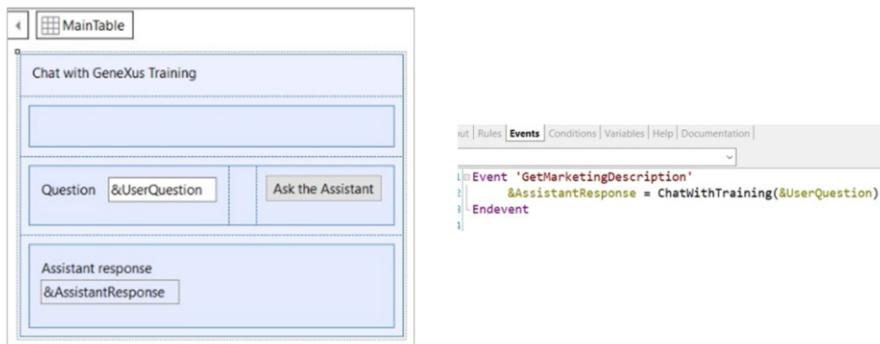
First, we indicate the "profile", that is, the name of the assistant, which in this case is ChatWithGXTraining.

As we discussed in the previous example, this procedure should receive the query from the end user to send it to the assistant and get the answer.

So we declare the corresponding Parm rule, where the input parameter is the user's question, and the output parameter, or return value, is the assistant's answer.

We may also set filters for the query, but we won't do it in this case. Next, we start running the POST, filling in the path.

GeneXus application that interacts with a RAG assistant



The image shows a GeneXus web panel and its associated event code. The web panel, titled "MainTable", contains a section "Chat with GeneXus Training". It features a text input field, a "Question" label, a variable field containing "&UserQuestion", and a button labeled "Ask the Assistant". Below this is an "Assistant response" section with a variable field containing "&AssistantResponse". To the right, the event code for "GetMarketingDescription" is shown, which calls the procedure "ChatWithTraining" with the user's question as a parameter and receives the assistant's response in the variable "&AssistantResponse".

```
1 | Event 'GetMarketingDescription'  
2 |   &AssistantResponse = ChatWithTraining(&UserQuestion)  
3 | Endevent  
4 |
```

Now we must complete its call from the web panel. For that, we go to the event associated with the button and call the procedure, sending as parameter the variable &UserQuestion with the user's question, and receiving its answer in the variable &AssistantResponse, both present in the form.

To test it, we press F5, and run the web panel.

GeneXus application that interacts with a RAG assistant

GeneXus GeneXus Enterprise AI Course

UNANNOUNCED

Products

Chat with GeneXus Training

Question: What is a GeneXus knowledge base? [ASK THE ASSISTANT](#)

Assistant response

A GeneXus knowledge base (KB) is a central repository of information and resources that is used in GeneXus development projects. It contains all the necessary data, structures, rules, and logic required for building applications using the GeneXus platform. The KB serves as a single source of truth for developers working on a project, allowing them to access and modify the same set of information.

In the context provided, it is mentioned that GeneXusGen allows for centralizing the knowledge base, which facilitates collaboration among multiple developers working on the same project. Each developer has their own local copy of the KB, which is connected to the centralized KB managed by GeneXusGen. This setup enables developers to work independently on their tasks while ensuring that all modifications are consolidated and synchronized through the central KB.

By using GeneXusGen to manage the knowledge base, developers can easily share development tasks, send and receive modifications, and work together on the project even if they are in different geographical locations. This centralized approach helps in maintaining a history log of activities performed on the project and streamlines the management of the project as a whole.

In summary, a GeneXus knowledge base is a crucial component in GeneXus development projects, serving as a centralized repository of information that enables collaboration, version control, and efficient management of the project.

GeneXus GeneXus Enterprise AI Course

UNANNOUNCED

Products

Chat with GeneXus Training

Question: What is a formula attribute in GeneXus? [ASK THE ASSISTANT](#)

Assistant response

A formula attribute in GeneXus is a type of attribute that is calculated based on a formula or expression defined by the user. Instead of storing a specific value like a regular attribute, a formula attribute dynamically calculates its value based on the values of other attributes or external factors. This allows for more flexibility and automation in the data model, as the calculated value is always up-to-date based on the defined formula.

In the context provided, the discussion is focused on how GeneXus determines which attributes best describe a transaction based on data types. While formula attributes are not explicitly mentioned in this context, it is important to understand that they play a crucial role in defining dynamic calculations within the data model. By using formula attributes, users can create complex calculations, validations, or business rules that are automatically applied to the data without manual intervention.

Overall, formula attributes in GeneXus provide a powerful tool for developers to create dynamic and efficient data models that can adapt to changing requirements and conditions.

GeneXus GeneXus Enterprise AI Course

UNANNOUNCED

Products

Chat with GeneXus Training

Question: What is a transaction object? [ASK THE ASSISTANT](#)

Assistant response

A transaction object in the context provided refers to a specific type of object created in GeneXus to represent real-world entities or actions that need to be recorded or processed within a system. In this context, it is mentioned that for every object from reality identified, a transaction object will be created. These transaction objects are used to model and manage data related to customers, attractions, courses, and other entities mentioned.

In the design of the first transaction, it is important to define the attributes and properties that will be associated with the transaction object. These attributes will capture relevant information about the entity or action being represented, such as customer details, attraction information, country data, etc. By creating transaction objects, developers can effectively map real-world concepts into the system and perform operations on them, such as data retrieval, manipulation, and storage.

Overall, transaction objects play a crucial role in structuring and organizing data within a system, allowing for the representation and management of various entities and actions. They serve as a bridge between the real world and the digital system, enabling the system to interact with and process information related to different aspects of reality.

Remember that our RAG assistant was fed with only 18 documents, which is not a lot of information, and for that reason we will ask simple questions whose answers are in the scope of those documents. This will allow us to test the application.

First, let's ask: What is a GeneXus knowledge base? Let's see the answer.

Good. Let's ask now: What is a transaction object?

Very good. And if we ask: what is a formula attribute in GeneXus?

Perfect. The answers received were thorough and correct. We consider it valid.

Now let's look at the menu. We have access to the chat assistant and the assistant to chat with documents.

In this way, we have achieved a GeneXus application that interacts with two artificial intelligence assistants created through GeneXus Enterprise AI.

GeneXus[™]
by **Globant**

training.genexus.com