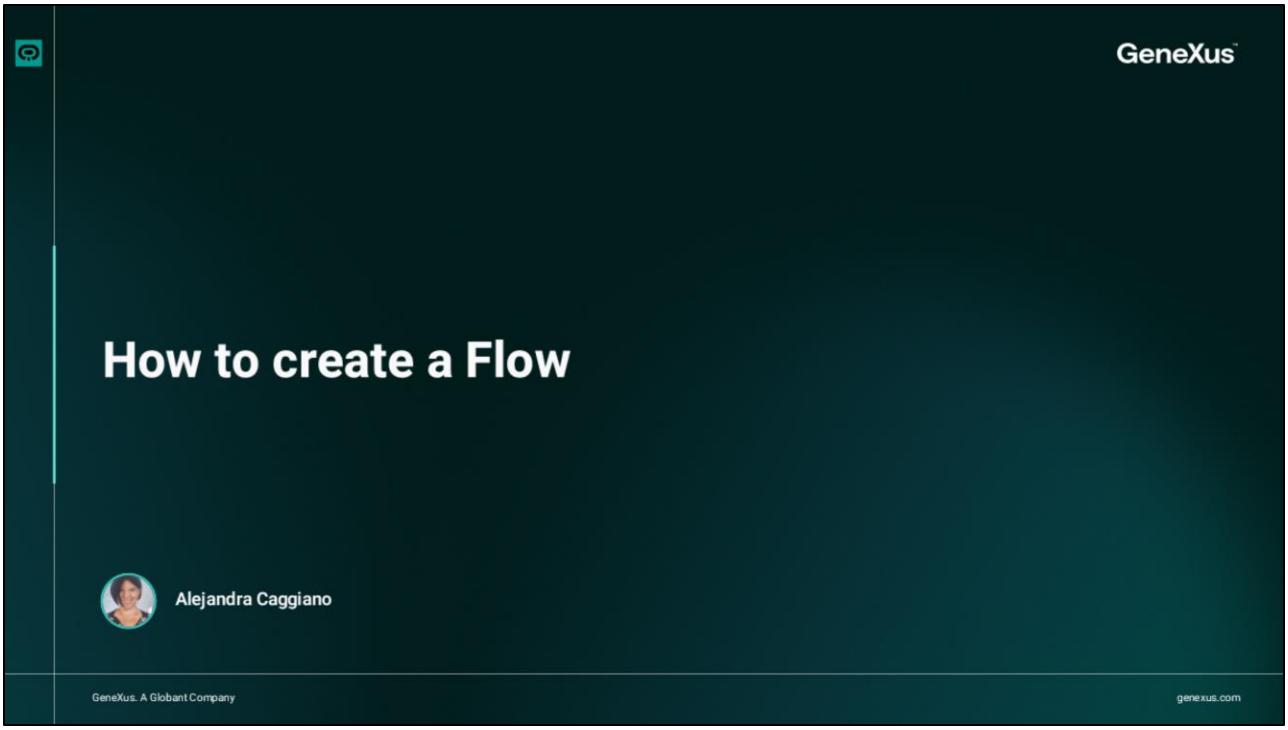
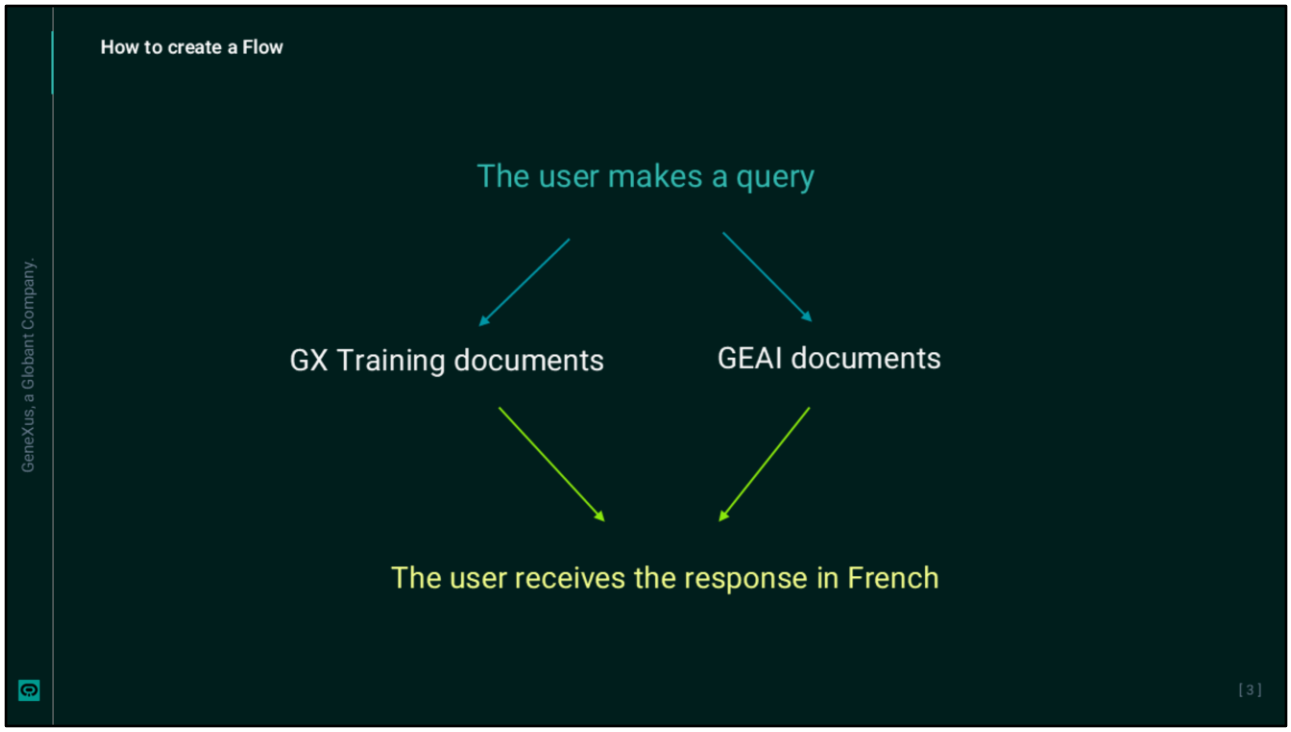
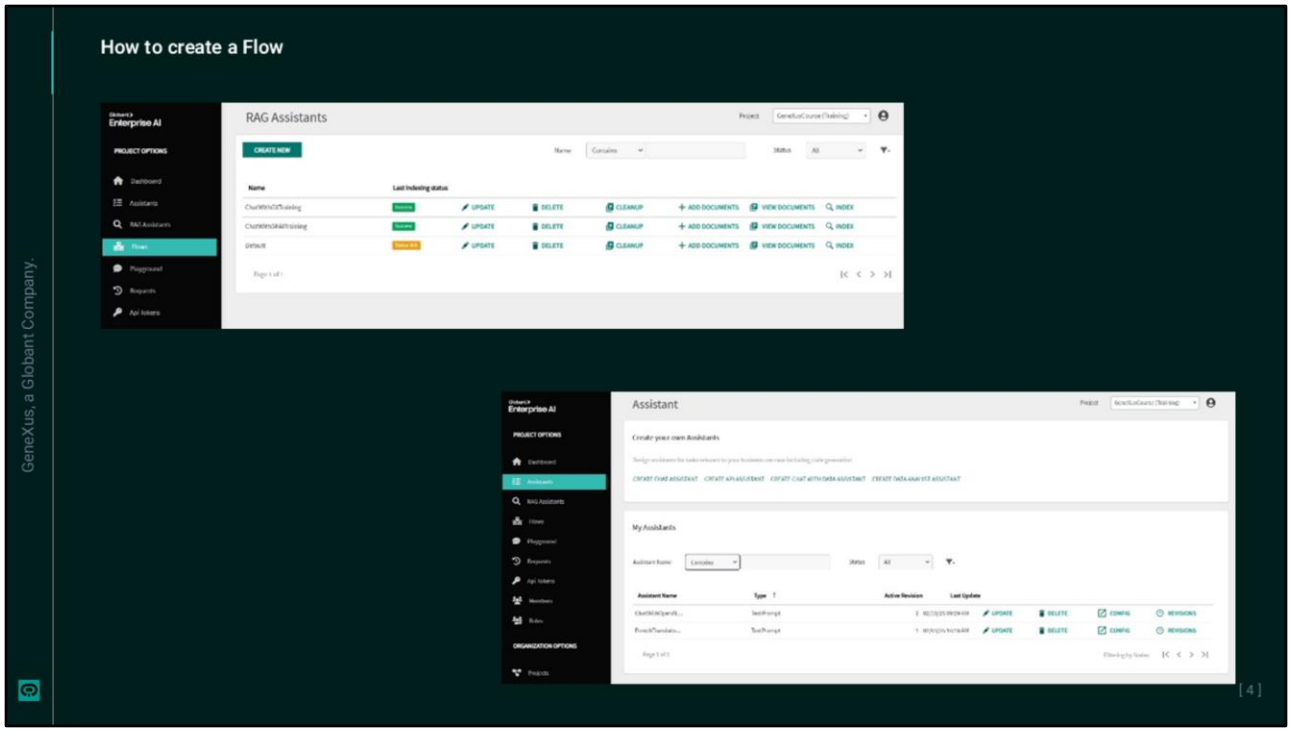

Globant
Enterprise AI



We are going to create our first Flow.



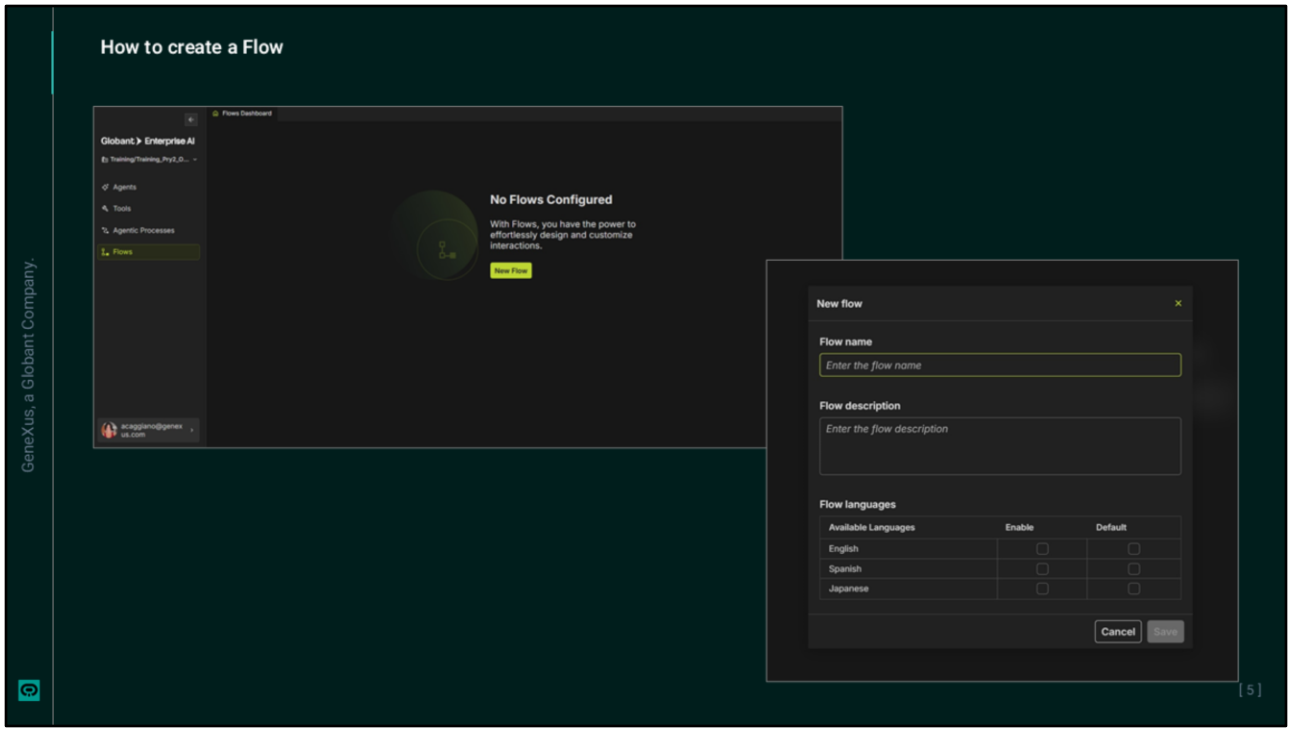
The objective is to enable the end user to make a query on a topic related to GeneXus or Globant Enterprise AI training and have the answer translated into French.



First, let's look at the defined assistants.

We access the backoffice, select the project and see that we have two RAG Assistants defined: one of them interacts with GeneXus Training documents, and the other interacts with Globant Enterprise AI documents.

In addition, we have a Chat Assistant that translates text from English to French.



OK, let's create the Flow.

From the menu, we select **The Lab**, and access a new window (Flow Builder), where we can create and manage or edit the flows associated with the selected project.

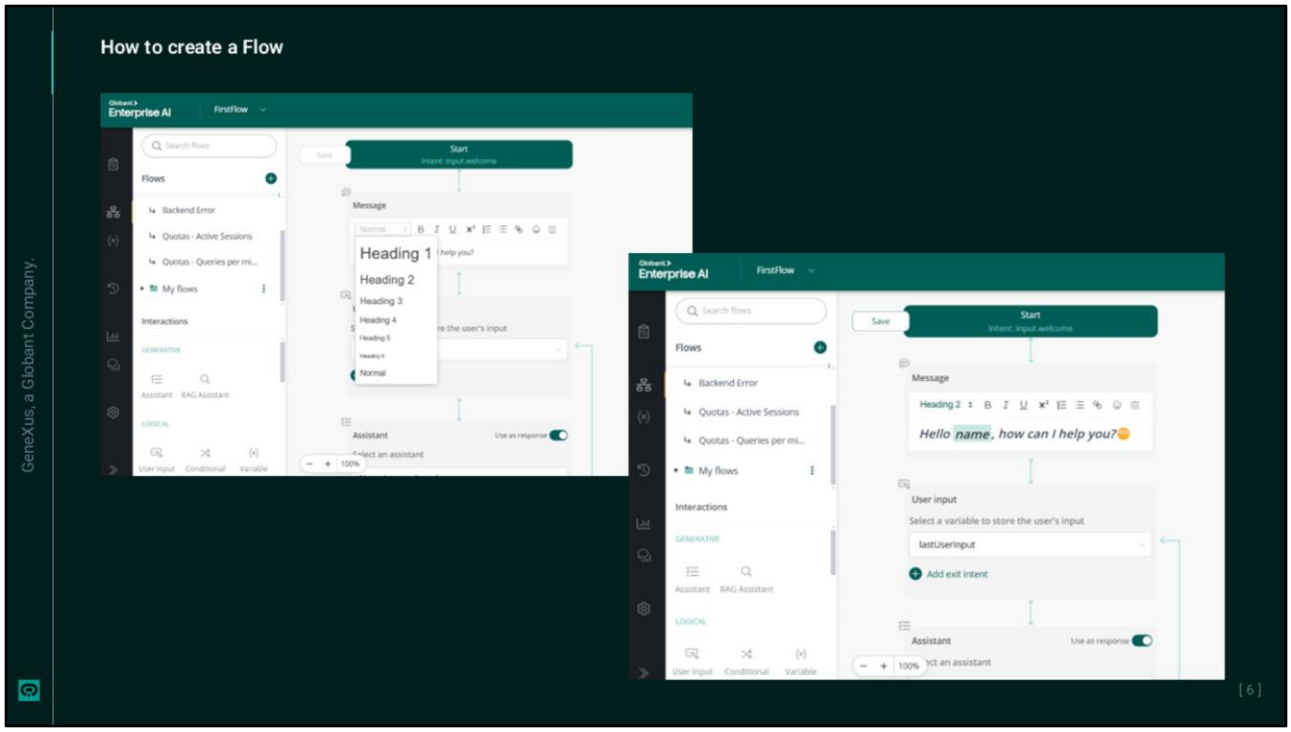
To start creating a new flow, we select **New Flow**. Here we must indicate:

- The name of the flow that will allow it to be easily identified. In our example, we will use “FrenchTrainingDocumentation“
- A description which, although optional, we recommend adding to understand its purpose.

Resolves GeneXus and GEAI queries, and translates their response into French

- Finally, the Languages option sets the language in which the flow will be configured and defines the language of the encoded messages. It is possible to select several languages so that the same message can be available in different languages.

We click on **Save**.



We must now add the assistants and customize their behavior. To do this, we click on **Edit**.

This initial node, called Start, is generated automatically and establishes a basic configuration that can be customized by adding or modifying interactions, messages and assistants as needed. It marks the beginning of the conversation flow. This node is activated each time the flow is triggered and its main purpose is to initiate the flow. From there, the other nodes that manage the interaction with the user are connected. It is possible to change the name of the node from Flows, Rename Flow.

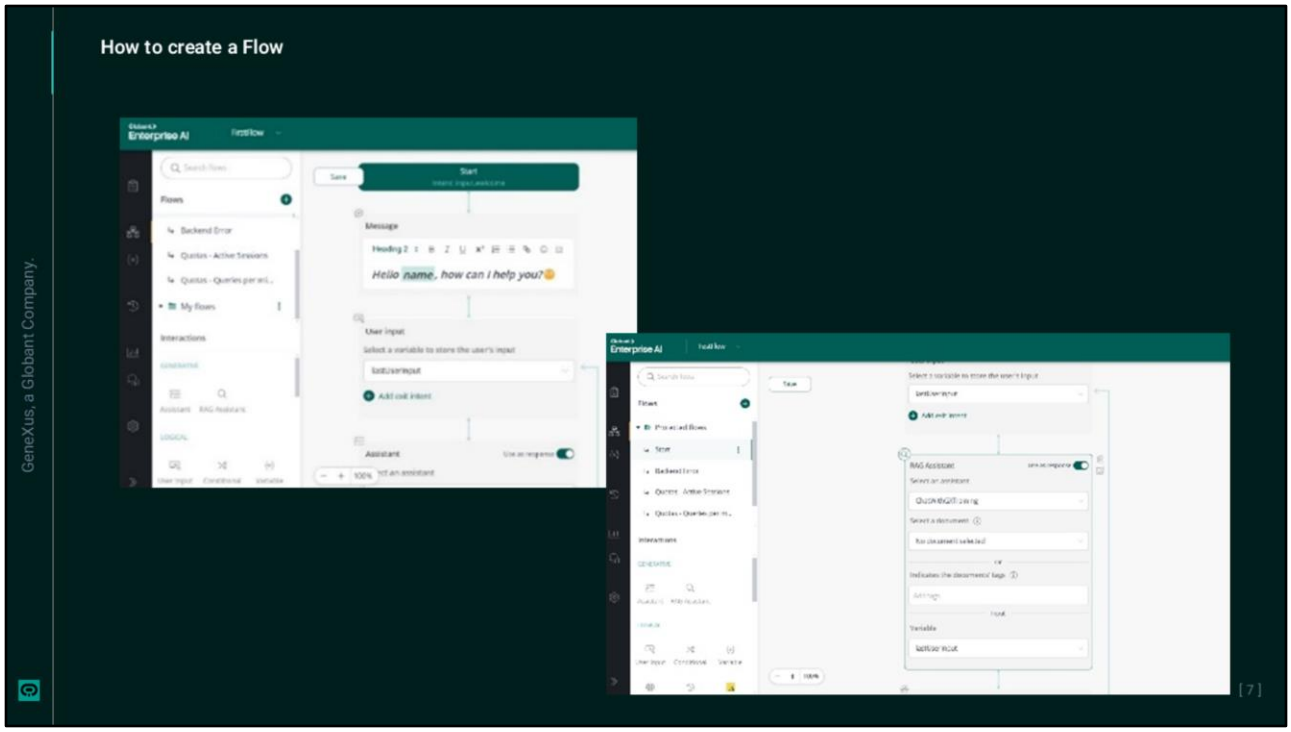
The next node is the Message node, which is responsible for sending the welcome message or any other initial message configured. In this case, the message is ”Hello {name}, how can I help you?” where {name} represents a dynamic variable that is filled with the user's name.

This message is customizable and can also be formatted with different styles, as we can see from here, to add lists, links and emojis.

Let's customize our initial greeting.

This last button allows you to insert dynamic variables in the message.
Selecting it displays a list of the variables available in the system, and it is also possible to define new custom variables.

If we want to delete this Message node, we do it from here.



Next, the flow captures the user input through the User Input node and stores it in the lastUserInput variable for later use.

Then we see the Assistant node, where the assistant must be configured to perform specific actions. This node uses the information stored in the lastUserInput variable that contains the text entered by the end user in the User Input node.

And if we click here, we are offered the assistants defined in the Assistants option of the project where we are positioned in the backoffice.

But for our example we need to choose one of the RAG Assistants. So we delete this node and from this menu we see that we can define interactions with Assistants and with RAG Assistants.

So, we drag the RAG Assistants option to the desired location within the Flow. In our example we place it next to the User Input node. We choose the ChatWithGXTraining assistant, and then this field allows us to select a specific document that will act as a source to respond to the end user's request. If we do so, the assistant will only use that document as a reference.
It is also possible to add tags that will allow the assistant to search only the documents that match the indicated tags in order to respond to the end user's request.

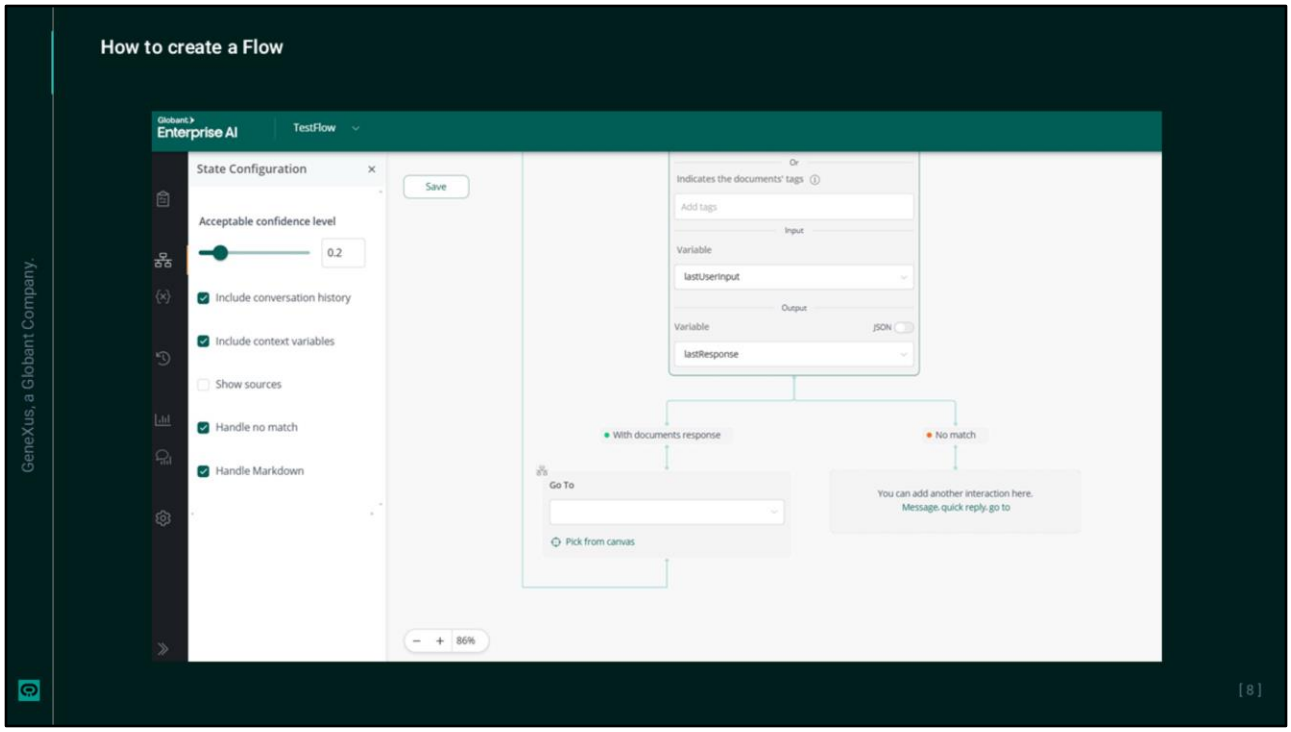
In the Input section there is a field called Variable. The variable to be used as input for the node is defined there; for example, lastUserInput.

As mentioned, this variable contains the most recent user input and will be used as the basis for querying the selected documents or tags.

The assistant's response can be stored in this same variable, lastUserInput, or in a new variable, as appropriate. If we look here, by default the Use as Response option is enabled, which means that the assistant's response is used directly in the lastUserInput variable.

In our example we want to save it in another variable, so we deactivate this option. Note that the Output field is enabled where we can select or create a new variable in which the output generated by the node will be stored.

In this case, we want to save the response of the RAG Assistant in a new variable called lastResponse.



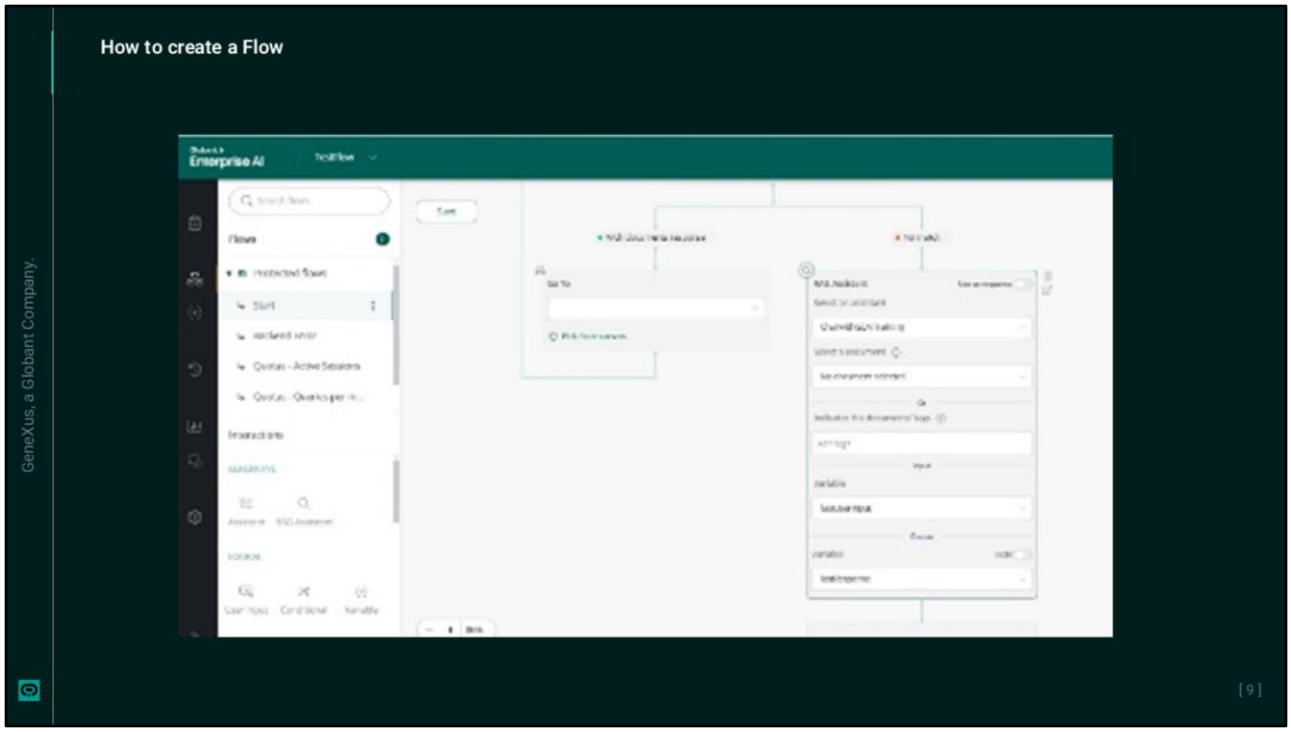
Now let's look at this pencil icon to the right of the node. Selecting it opens the State Configuration menu where it is possible to modify the following:

- Acceptable confidence level: The default value is 0.2, and indicates the confidence level in matching fragments obtained by the RAG Assistant after searching the vector database. Only responses that meet or exceed this confidence level will be considered valid.
- The option to include the conversation history allows the assistant to take previous messages into account when generating responses.
- Including context variables is also enabled by default and allows context variables stored during the conversation to be sent to assistants for use within prompts.
- The Show sources option allows the assistant to display the sources of information used to generate the response, which helps to provide transparency and validate the accuracy of the response.
- In addition, it is possible to Handle empty response. This option is disabled by default and handles situations where the assistant cannot generate a valid response. When enabled, it opens a new flow branch, allowing for the configuration of specific actions to take in case there is no response available. In this new branch, it is possible to add any of the components available in the menu to continue the interaction as needed.

In our example, if the RAG Assistant finds the answer in the GeneXus Training documents, you want the flow to call the corresponding assistant to translate the answer into French and store it in the same input variable (lastResponse).

But if it does not find any answer, the flow will have to pass to a second RAG Assistant configured with the documents used in Globant Enterprise AI courses.

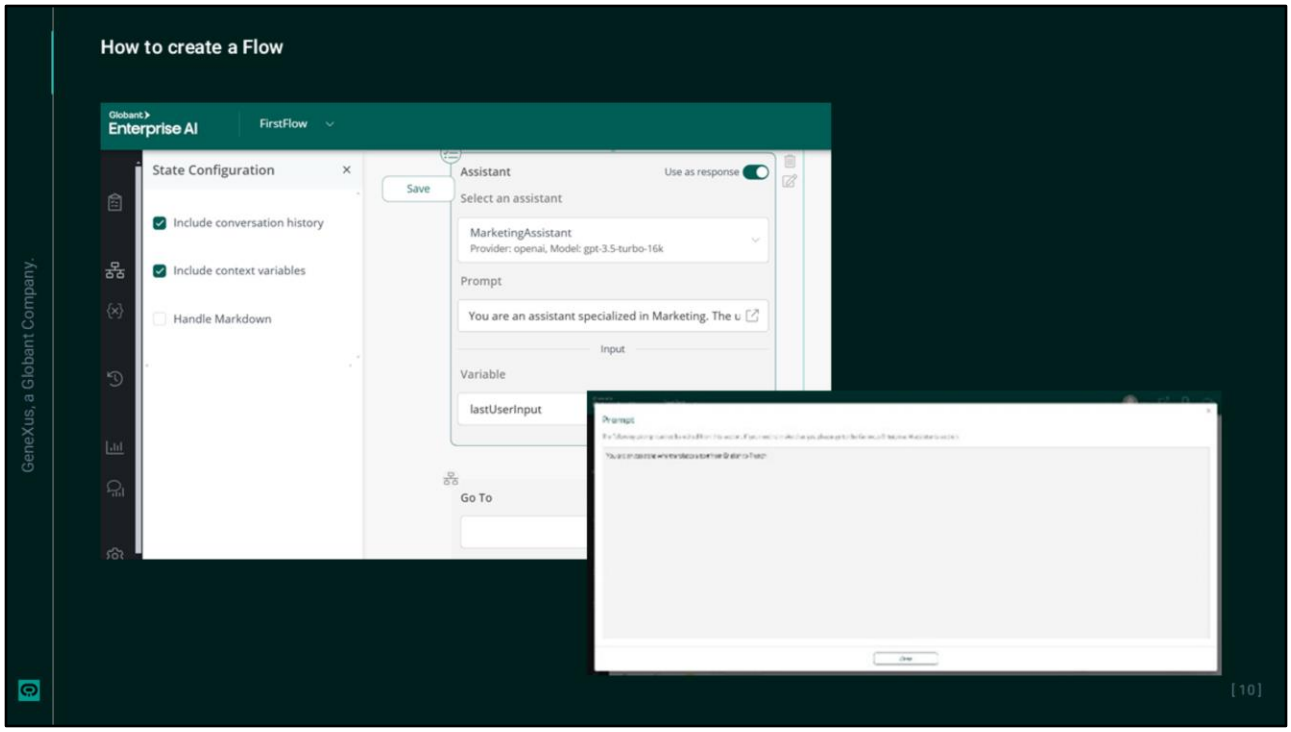
So we select the option Handle no match, and we see that a new branch is opened.



Then, when the user's query does not find an answer in the first RAG Assistant, it should continue on this side.

So we close the configuration box and drag a new interaction with a RAG Assistant. We select the corresponding assistant that will have lastUserInput as input variable which, remember, is the variable that contains the last user input.

The response generated by this RAG Assistant should be stored in the lastResponse variable, so we deactivate this flag to allow us to indicate the output variable.

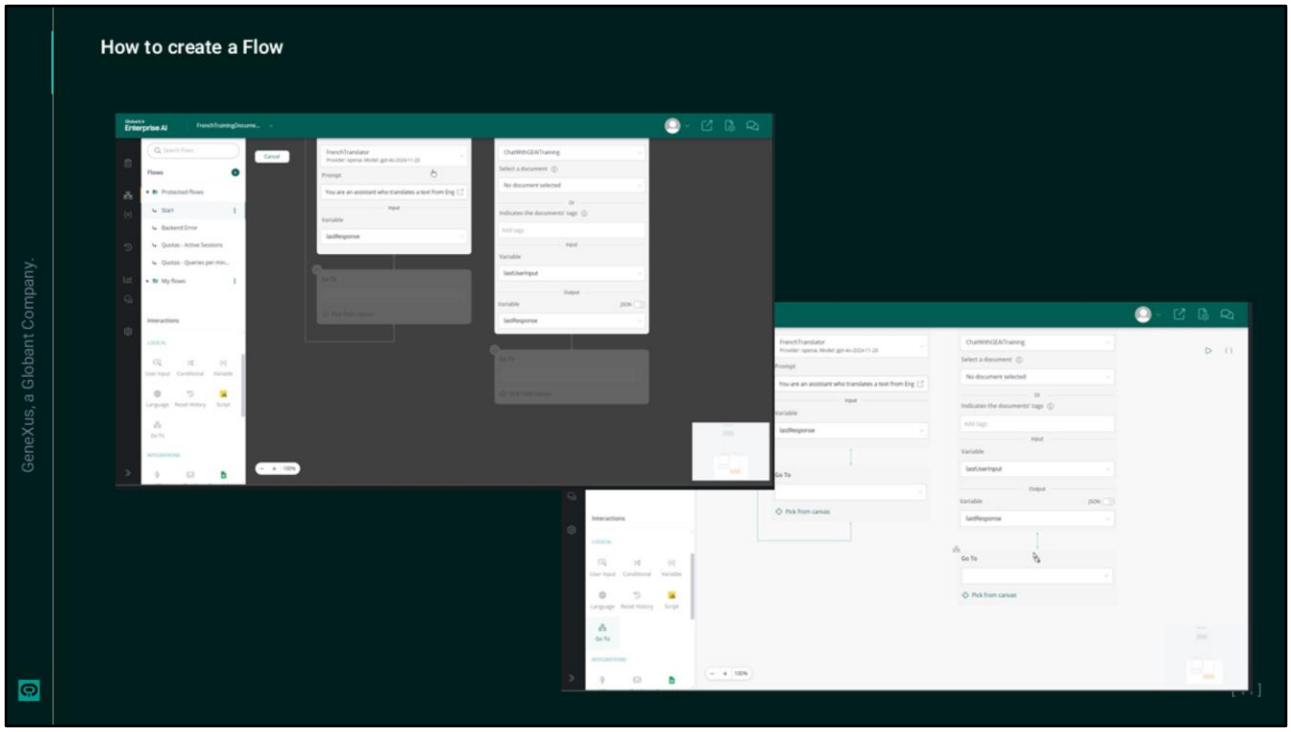


OK, but if the response to the user's query is found with the action of the first RAG Assistant, then it should be translated into French.

So we choose the interaction with an Assistant and drag it to its location in the Flow.

We then choose the Assistant that translates into French and see that the Prompt field displays the message defining the assistant's task. This message is read-only and cannot be edited.

We store its response in the same lastResponse variable.



Finally, the Go to node is used to connect different flows within the interaction. This allows you to select the flow to which you want to redirect the end user based on their current interaction.

Clicking on the blank bar (which corresponds to the drop-down menu), displays the list of available flows:

In addition, the Pick from Canvas option allows you to select a node directly from the canvas, facilitating navigation and connection between different parts of the Flow.
In this case, there are two 'Go to' nodes.

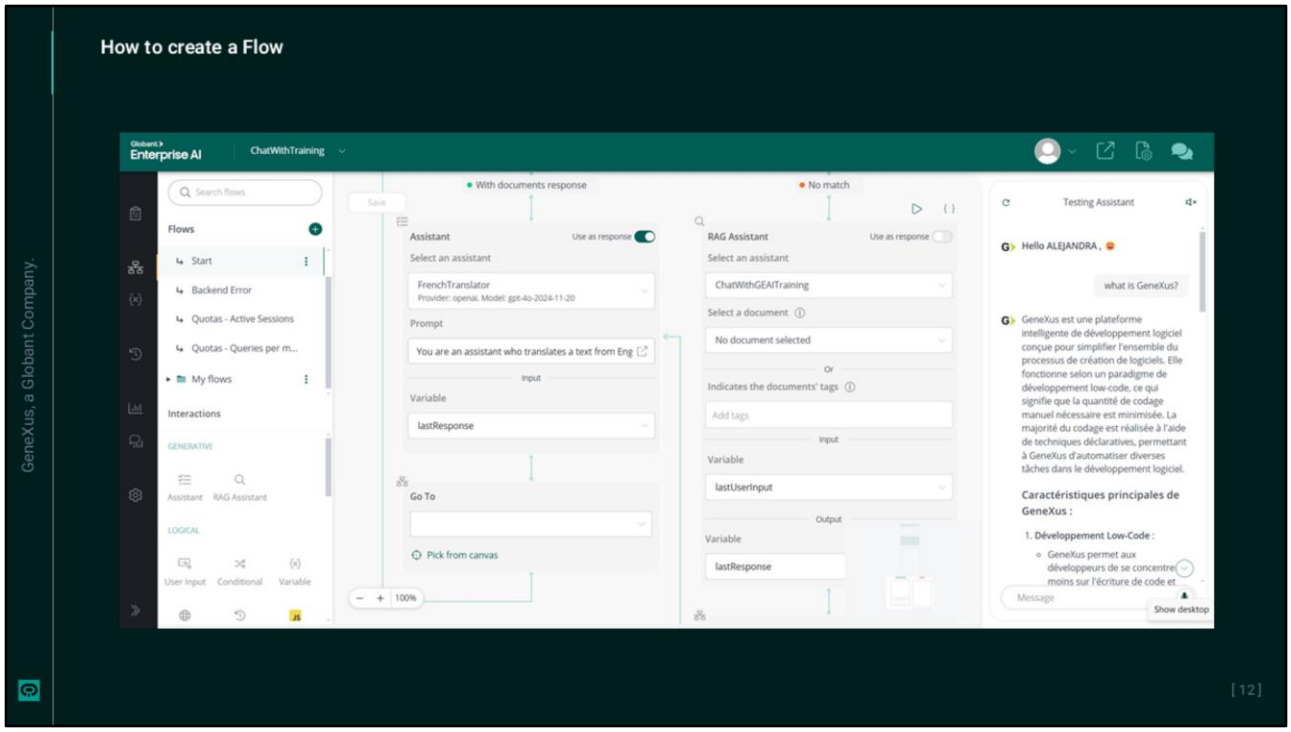
- One of them returns to the User Input node. Since the previous node (which translates into French) does not store the answer in a new variable and simply redirects to the "user input", the answer received by the end user is the actual answer to their question translated into French.
- The other 'Go to' node must send the response of this second RAG Assistant to the node that is in charge of translating into French using the lastResponse variable.

Therefore, from the menu, we choose the Go to interaction, and drag it into position. We choose Pick from Canvas and indicate the Assistant node to which we should redirect this output.

The lastResponse variable contains the response obtained from either of the two RAG Assistants previously executed in the Flow.

Let’s review the sequence of the variables:
The user enters a query and it is received in the lastUserInput variable.
The first RAG Assistant receives that input and if it has the answer, it stores it in the lastResponse variable, which, in turn, is received by the assistant that translates into French.
In case of not having the answer, the user's query in lastUserInput enters the second RAG Assistant that saves its answer in the lastResponse variable and sends it to the Chat Assistant for translation.

In this way, the flow ensures that the end user receives the response in French, regardless of the RAG Assistant that generated it.

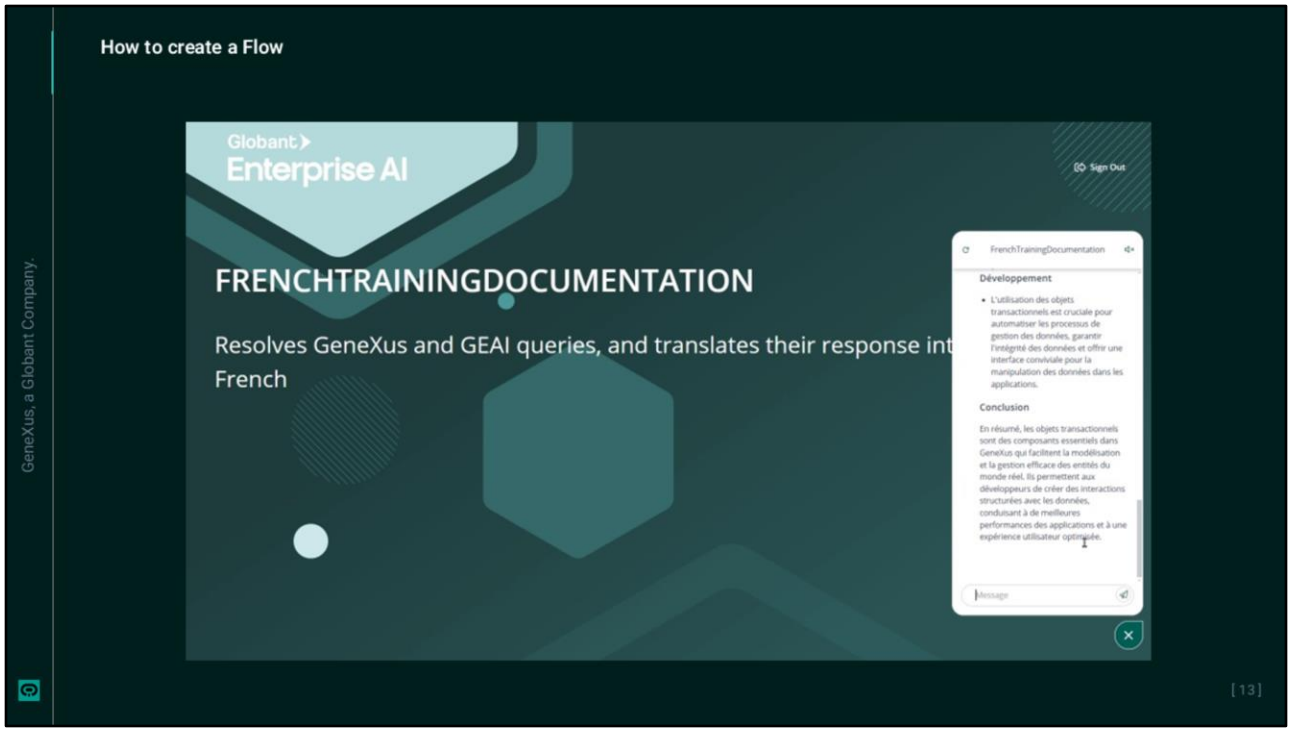


Well, at this point, we can test the Flow so first we save it by pressing Save.

We have three options to test the Flow from this design screen.
These two options open a window on the right side of the screen, while this other option will open a new window in the browser.
Let's test it from the screen itself.

We see the initial message. Let's ask “What is GeneXus?”

We get the answer in French.



Now let's try from this other option…

In this way, we have created and tested our first Flow. Next, we will create a Flow with conditional component.


Globant
Enterprise AI