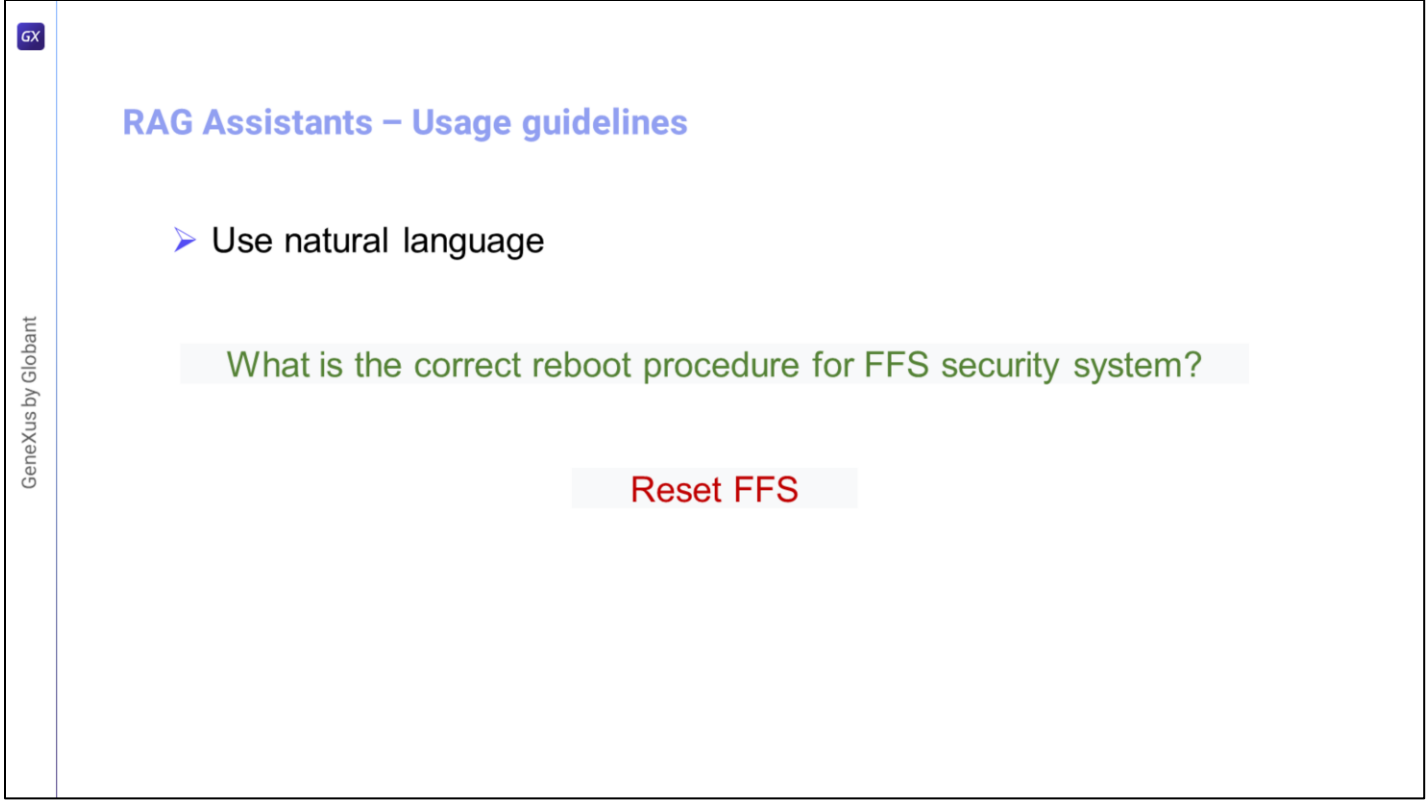
GeneXus by Globant

RAG Assistants – Usage and Configuration Guidelines

Alejandra Caggiano

Now that we know how to create and test an assistant, we'll see some guidelines on how to ask the right questions to get information that was part of a given RAG Assistant's document ingestion.

Let's look at some general considerations:

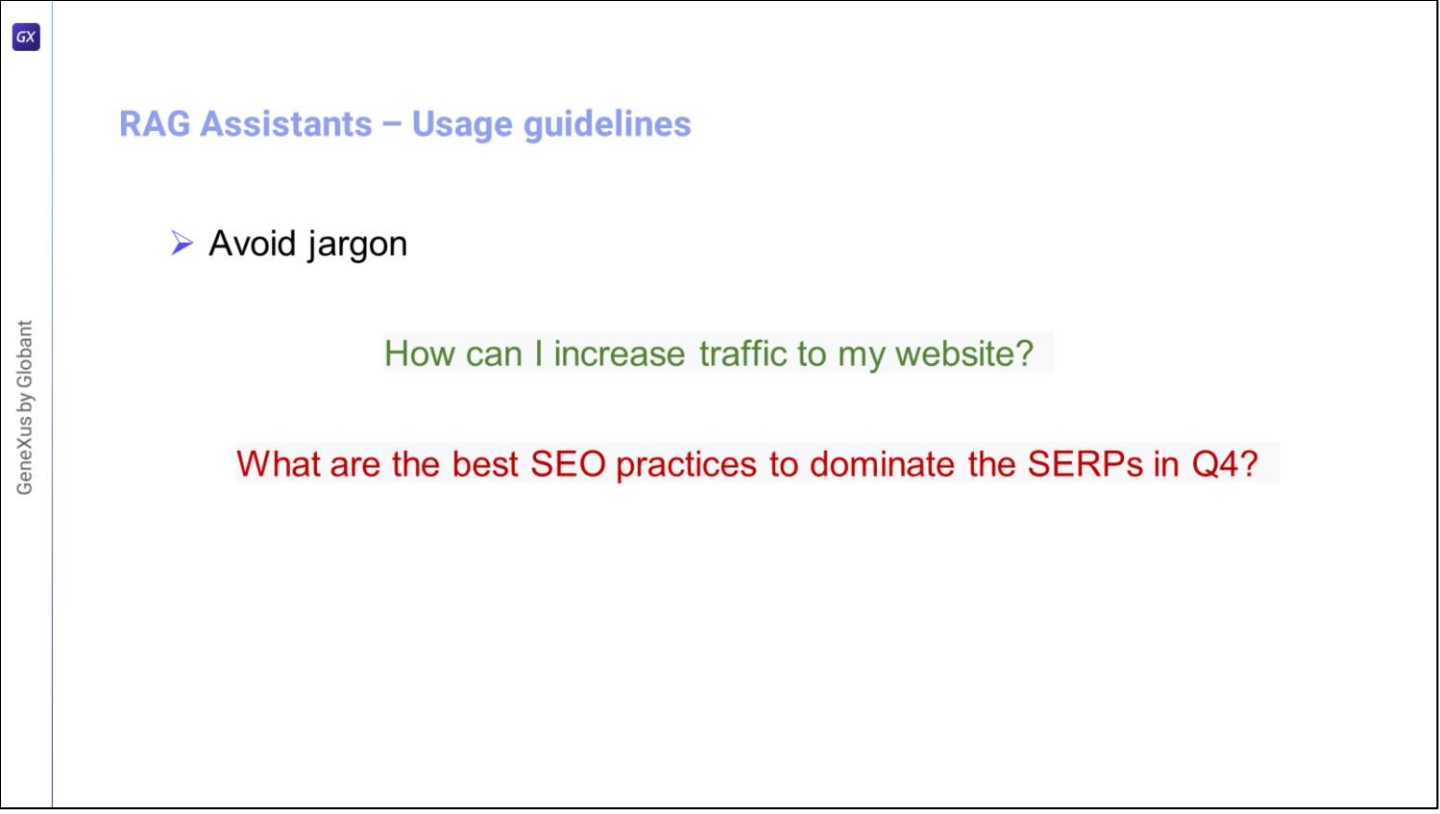


The first consideration is to **use natural language**: This means writing questions naturally, just as if you were talking to a person.

It doesn't matter if you make a spelling mistake or use synonyms. What's important is to convey your intention correctly.

To give an example, a correct question might be: Which is the correct restart procedure for the FFS security system?

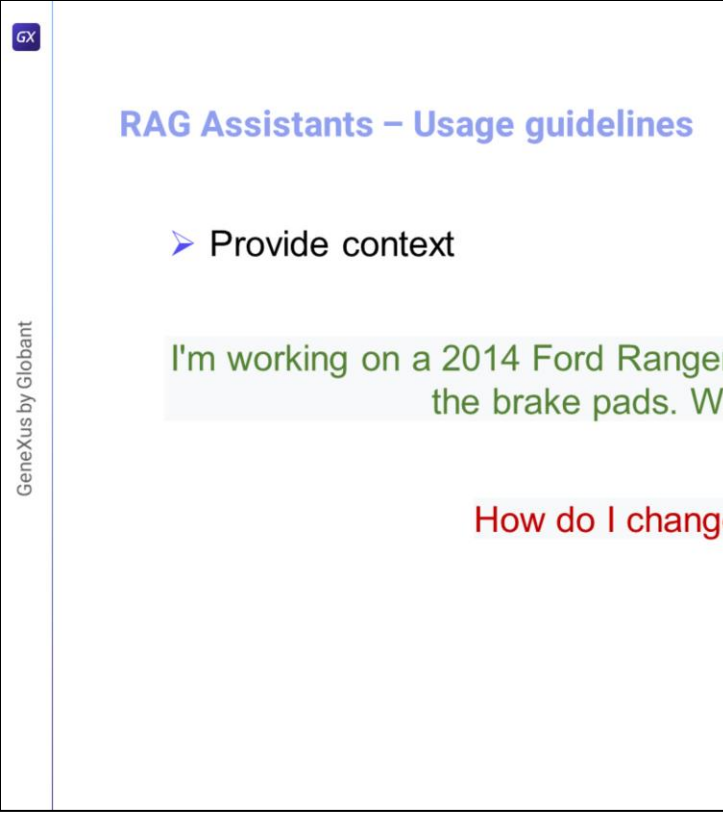
Meanwhile, an incorrect expression would be to simply write: Reset FFS.



Good. Another consideration to keep in mind is to **avoid jargon**: While the assistant may understand technical terms, simpler language can often provide more accurate results.

For example, a correct expression would be: How can I increase traffic to my website?

On the other hand, an incorrect expression would be: What are the best SEO practices for mastering SERPs in Q4?



Another important consideration is to **provide context**: Adding a bit of context or background to the query can help the assistant better understand the scope of the request.

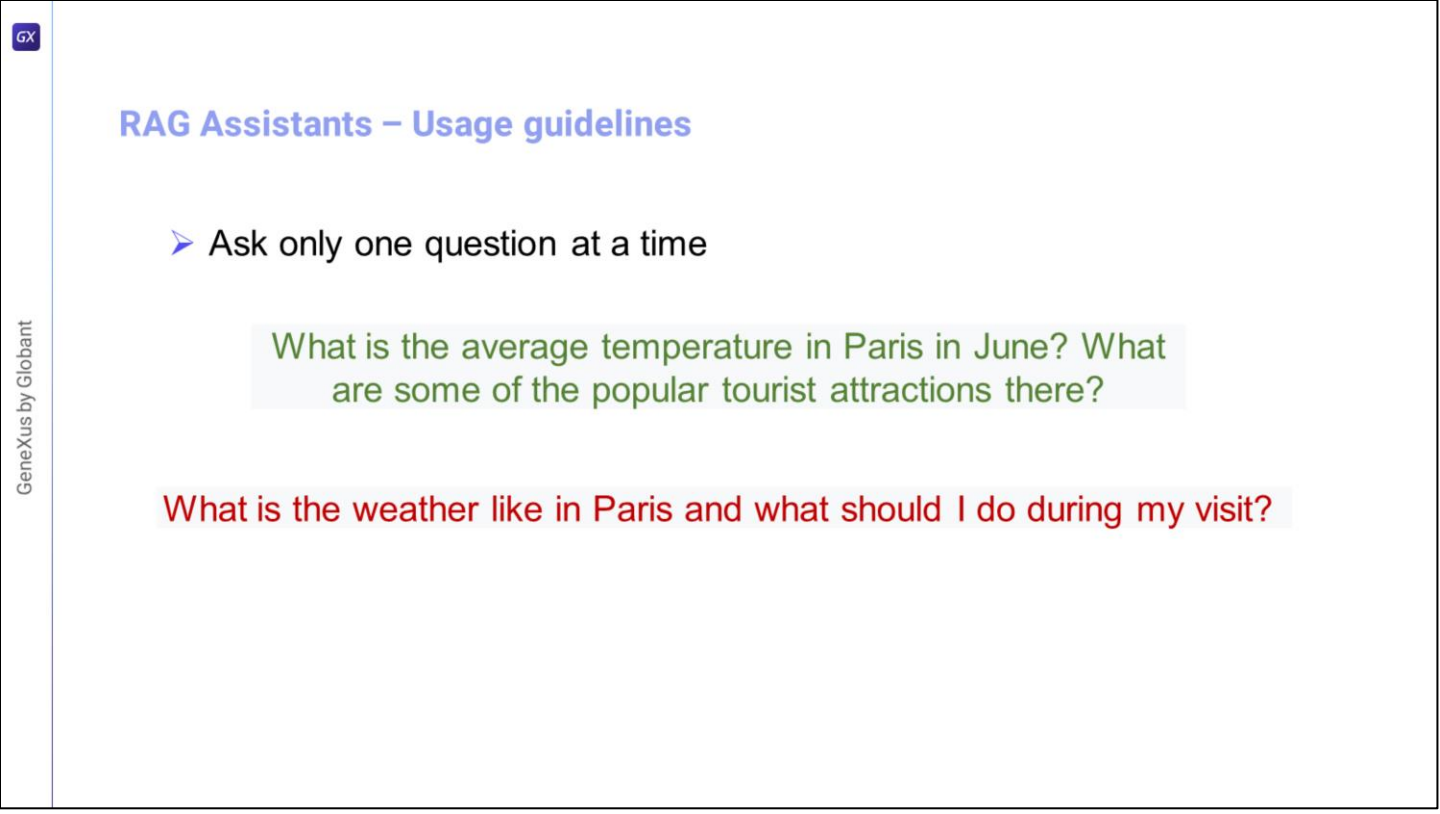
A correct expression would be: I'm working on a 2014 Ford Ranger and need to know how to replace the brake pads. What are the steps?

Meanwhile, it would not be good to simply ask: How do I change the brake pads?

Another recommended guideline is to use **clear and concise language**: It is important to be specific and clearly articulate what you are looking for in order to receive more accurate and relevant information. Avoiding ambiguity helps the assistant provide accurate answers.

A correct example would be to ask: What is the process for Uruguayan citizens to apply for a Japan tourist visa?

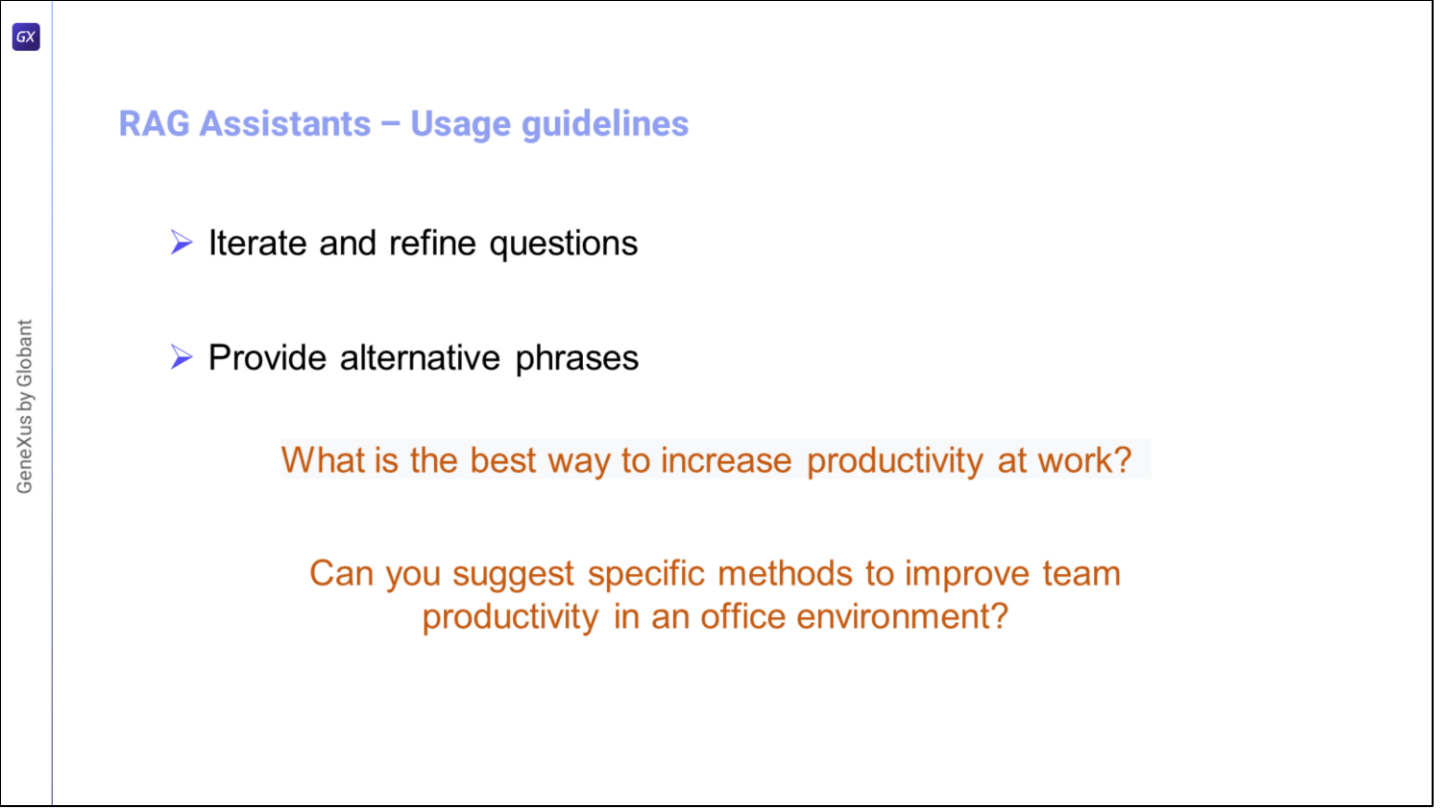
On the other hand, it would not be recommended to simply ask: How do I travel to Japan?



Another useful guideline is to **ask only one question at a time**: If we must ask a complex query, it is good to consider breaking it down into simpler parts.

A correct example would be: What is the average temperature in Paris in June? What are some of the popular tourist attractions there?

On the other hand, it would not be advisable to ask: What is the weather like in Paris and what should I do during my visit?



Another good consideration is to **iterate and refine questions**: If the initial answer is different from what we need, we should rephrase the question according to the answer provided by the assistant.

Providing alternative phrases is also a good idea: This means trying different ways of phrasing a query, including synonyms and variations in order to explore the assistant's understanding.

Let's look at the following query as an example:

What is the best way to increase productivity at work? If the assistant doesn't understand the question, it could answer with general productivity strategies rather than specific workplace strategies.

It can then be rephrased as follows:

Can you suggest specific methods for improving team productivity in an office environment? This rephrasing, then, clarifies that the focus is on team productivity in a professional environment, not on individual productivity tips.

RAG Assistants – Questions that the assistant cannot answer

- Avoid asking for information related to dates

What was the contract between March and April?

- The Assistant cannot provide information that is not found in the documents
- The Assistant cannot summarize or count items

Good. Just as we have seen usage guidelines and considerations when communicating with an assistant, it is also important to keep in mind that **there are certain questions that the assistant cannot answer**.

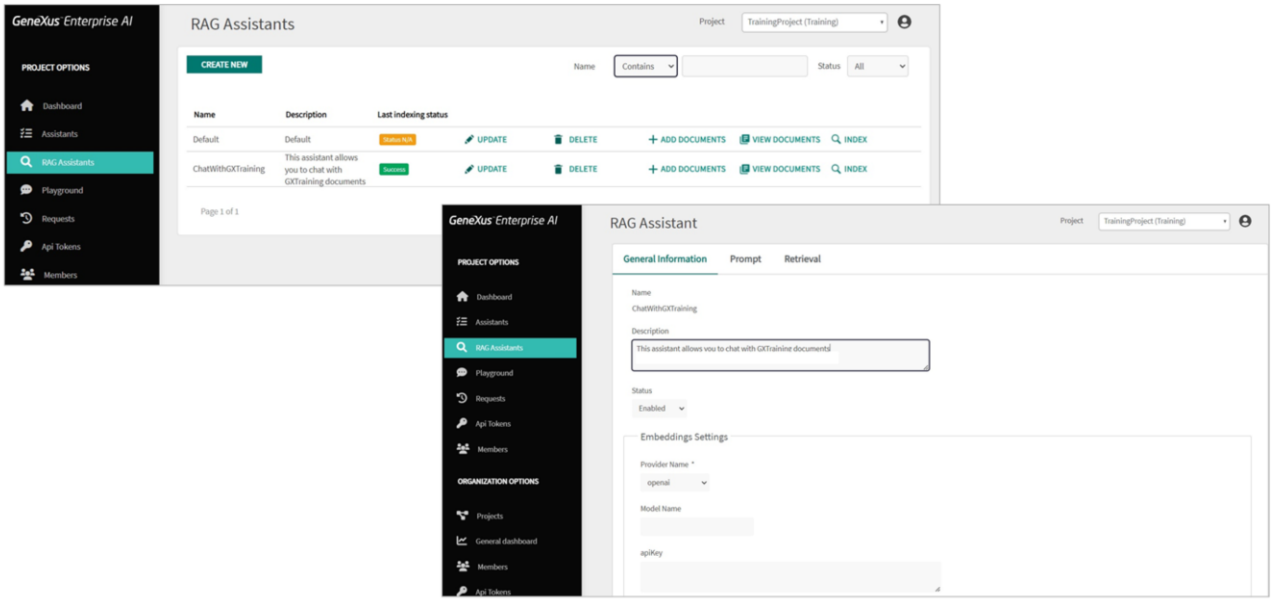
For example, we should avoid asking for information related to dates. Something along these lines would not be advisable: “What was the contract between March and April?”

Comparisons or contrasting information are not expected to yield good results.

Another query that the assistant cannot answer involves information that is not part of the documents that were used to give context to the RAG Assistant.

It cannot summarize or count items, nor can it process text appearing in images.

RAG Assistants – Configuration



OK. Now that we know the considerations to take into account to communicate effectively with a RAG assistant, let's take a closer look at its settings.

As we have seen before, every interaction with the Search and Chat component is configured through the RAG Assistants section. A default RAG assistant is created during initialization and then you can modify it or create new ones to change its behavior.

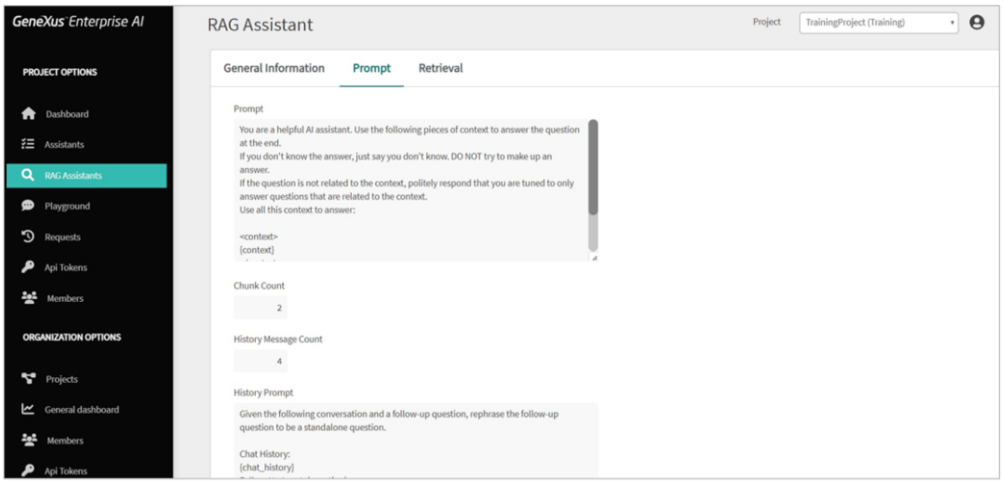
Once created, we access its options through the Update button, where we know that this set of options is organized in three sections:

The **General Information** section, which indicates details that allow us to quickly understand the assistant's capabilities and features.

The **Prompt** section, where information is provided on how the assistant is configured to receive and process user queries.

And the **Retrieval** section, which specifies how the information is retrieved.

RAG Assistants – Configuration



Let's go to the **Prompt** section.

As we already know, here you can configure instructions to guide the assistant on how to approach and answer questions. These instructions establish clear guidelines for the assistant to provide relevant and useful answers based on the context provided. The default value is the one displayed.

The context and question variables must be kept configured because they will be replaced with the associated information before the interaction.

Good. This next option defines how many fragments are retrieved to increase the context.

The Number of messages in history option sets the number of historical messages that are taken into account in the conversation.

It is useful for tracking the interaction history and understanding the context gathered in the conversation. It should be taken into account that this value is related to the end user's question and the associated answer. That is, if it is set to 4, it means that the last 4 interactions should be considered:

•Previous question

•Previous answer

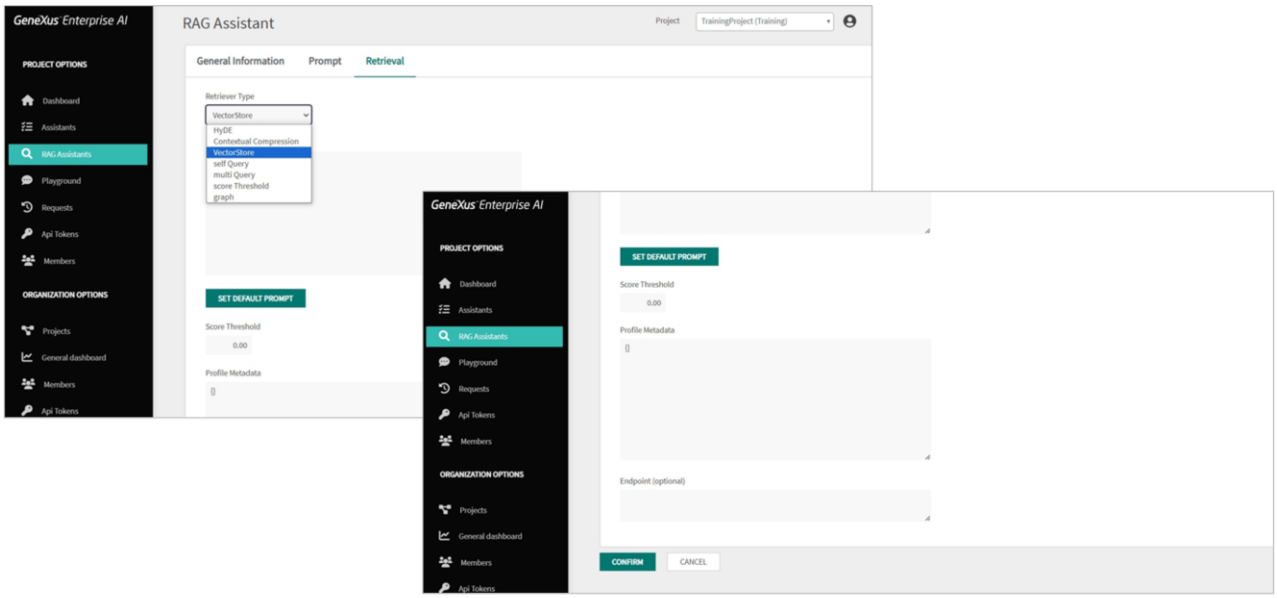
•Last question

•and Last answer

The minimum value it can take is 0, which indicates that the conversation history is of no interest.

As for the LLM configuration options, they include the configuration of the model used by the assistant to generate the response, including the service provider, model name, temperature, maximum token limit, and other parameters that affect how answers are generated.

RAG Assistants – Configuration



Let's move on to the **Retrieval** tab.

This section allows you to specify how to retrieve the information sent to the context.

Here you can indicate the type of retriever used to get information. The default value is VectorStore, which directly uses the defined VectorStore without additional pre-processing.

The other values it can take are the following:

Hypothetical Document Embeddings method: This method uses embedding techniques to answer queries, generate hypothetical answers, embed them in the generated document and then use it as a final example.

Another possible retrieval method is **Contextual Compression**. This method seeks to improve the answers returned by document similarity searches by better taking into account the context of the query.

Another option is **Self-Query.** In this type of retrieval, it first queries itself to retrieve filter information based on the natural language query. Then it runs a second query to the LLM with the query and filters applied based on the first one.

Multi-Query can also be selected. This type of retriever automates the fast tuning process by using an LLM to generate multiple queries from different perspectives for a certain initial user query. For each query, it retrieves a set of relevant documents and takes the unique union across all queries to obtain a larger set of potentially relevant documents.

Next, the Score Threshold option uses what is called recursive similarity search. All similar query matches will be returned according to the specified minimum score threshold.

Finally, the **Graph** retriever type uses a graph-based information representation approach for retrieval.

Then, in this box you can specify the query that is sent to the retriever to search for information. This query can be a question or a specific request.

This **Score Threshold** box defines the minimum valid value to consider the information as valid when it is retrieved. If there are no valid documents, no interaction with the LLM occurs. The default value is 0.0.

The **profile metadata** box may contain additional metadata related to the retriever's profile. The default value is an empty object, but in certain cases, it is convenient to be able to refine each access to the LLM to obtain the desired result.

Finally, the **Endpoint** box may contain the URL pointing to the specific server or service where the retrieval methods or models are hosted.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com