GeneXus by Globant

Creation and use of APIs to interact with Assistants

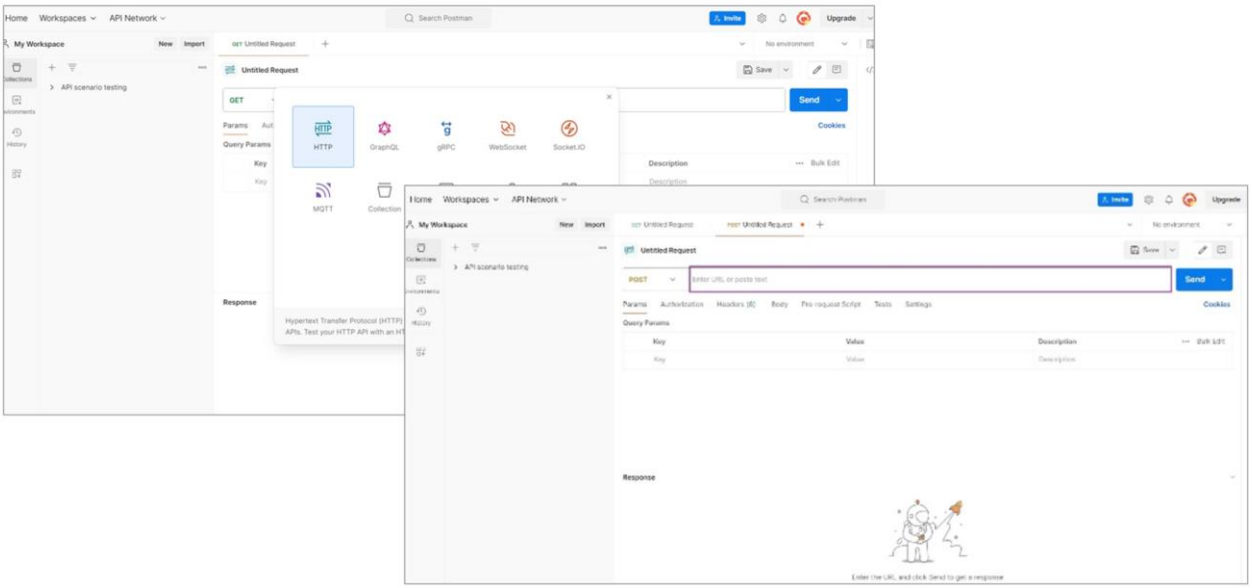
Alejandra Caggiano

Once an assistant has been created, tested at the time of its creation, and tested again from the Playground, we consider it valid.

Postman API Platform

We will now test it via API through Postman API Platform, a platform to create and use APIS. From there, we will be able to use the API in our developments, regardless of the technology.

Postman API Platform

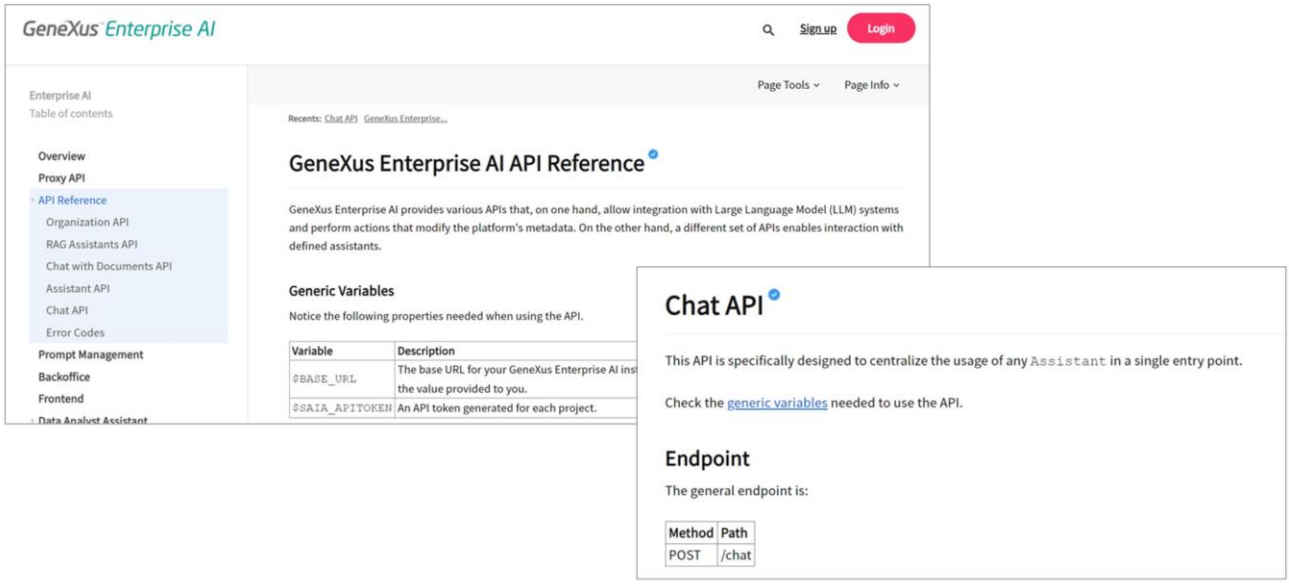


We enter Postman and from My Workspaces we select New... HTTP:

Since we want to send a query and receive a response, what we will do is a POST.

So here we choose POST, and we must indicate the URL to access the server. To determine this URL, we consult the GeneXus Enterprise AI documentation.

Postman API Platform



Our goal is to connect to the chat assistant that we created earlier, called Marketing Assistant, so we need the necessary information to communicate with this type of assistant.

It is important to take this into account because if we wanted to connect, for example, with a RAG Assistant, we would need other parameters.

Good. Therefore, in API References, Chat API, we find the information we need and examples of use:

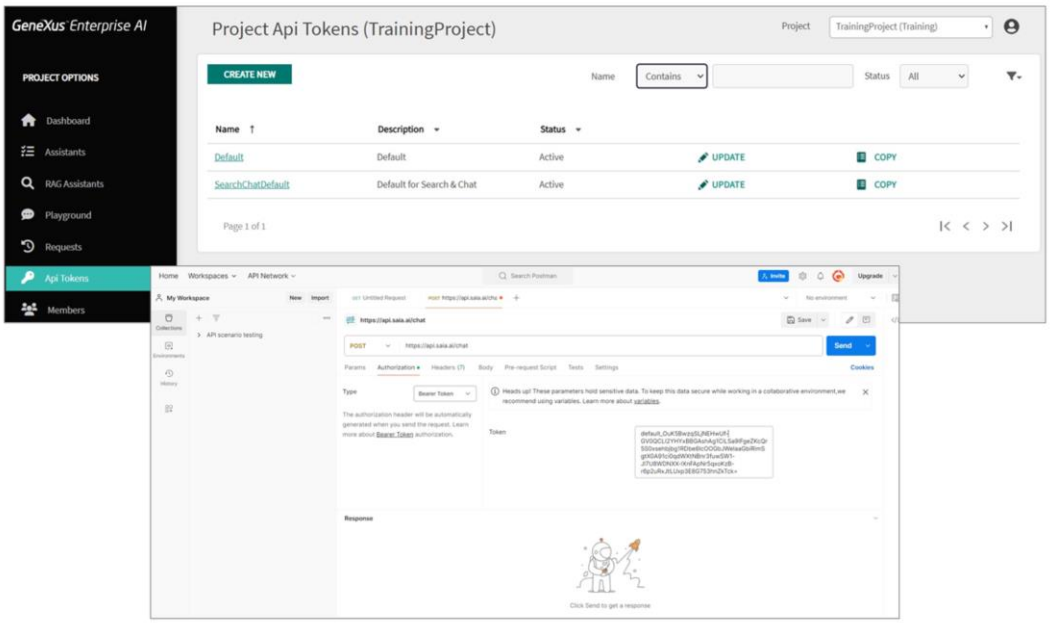
To use the API we need the content of the variables BASE_URL and SAIA_APITOKEN, where BASE_URL corresponds to the installation base URL of GX Enterprise AI and SAIA_APITOKEN corresponds to an API token generated for the project.

OK, so we indicate the base URL corresponding to our scenario:

https://api.qa.saia.ai

We see that the path is completed with /chat, so we add it in the POST statement.

Postman API Platform



We now move on to Authorization, which is Bearer token, so in the Postman Authorization tab we choose that option.

To use the API, we must authenticate each request using an API token. As we have already seen, these tokens are managed in the GeneXus Enterprise AI Backoffice and uniquely identify the sender of the request.

We must then provide that token. For that, we go to the GeneXus Enterprise AI Backoffice, and in the API Tokens option we can select any of the defined tokens, since any of them identifies us in the project.

For example, the Default one. We select Copy.

We go back to Postman, and paste the token.

Postman API Platform

```
curl -X POST "$BASE_URL/chat" \
 -H "Authorization: Bearer $SAIA_PROJECT_APITOKEN" \
 -H "Content-Type: application/json" \
 --data '{
    "model": "saia:assistant:translate-to-spanish",
    "messages": [
        {
            "role": "user",
            "content": "Hi, welcome to GeneXus Enterprise AI!!"
        }
    ],
    "stream": true
}'
```

Good. Now we go to the Body tab, choose RAW, JSON and refer to a basic example of what we should declare here.

We have already indicated the POST, the authorization method and the content type. We must now specify the body of the query, so we copy what corresponds to DATA, and paste it.

“Model” corresponds to the type of assistant, followed by its name. As we have said, depending on the type of assistant, the parameters may vary.

The “Assistant” type identifies a standard assistant, a Data Analyst Assistant and an API Assistant, while the “Search” type identifies a RAG Assistant.

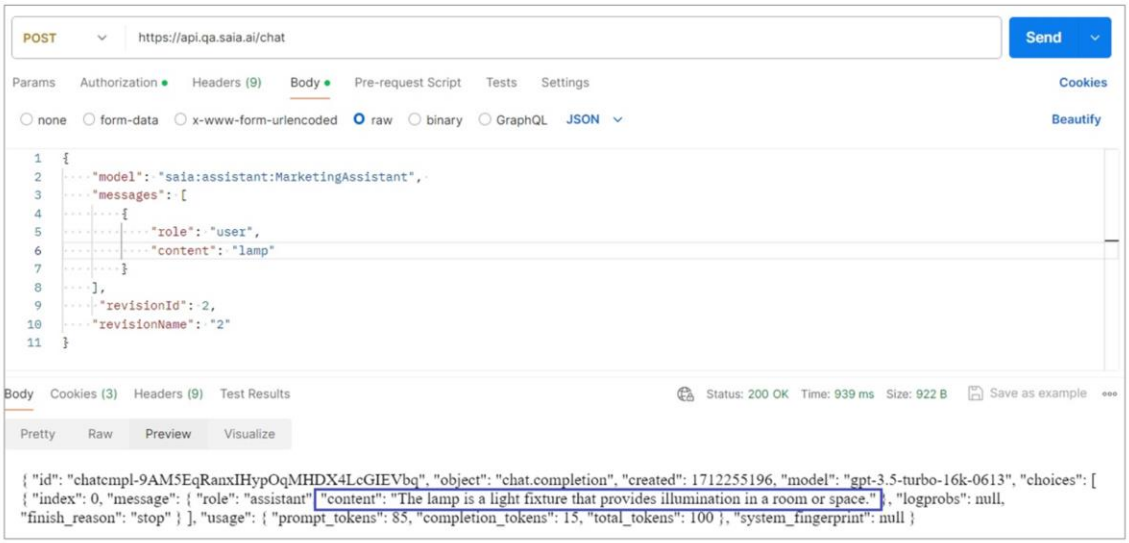
In this case, we want to contact our chat assistant, so the indicated type is “assistant” and the name is “Marketing Assistant”.

Let's look now at the Messages element.

This element defines a message that we want to add. The minimum expression is this one we are declaring, where “content” corresponds to the user input.

In this example, we are going to indicate “lamp” as input, expecting the assistant to return a correct description of this product.

Postman API Platform



Then we can add other parameters, so let's indicate, for example, the revision of the assistant we want to use. Remember that we could have several revisions of the same assistant.

We then specify the identifier and name of the revision.

“revisionId”: 2

“revisionName”: “2”

Now we click on Send and we get the answer:

The assistant says that “A lamp is a lighting fixture that provides illumination in a room or space”.

Very good. Now that we have tested our assistant via API, we can use it in our developments.

Later on, we will see examples of use from a GeneXus Knowledge Base.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com