

Criação e uso de APIs para interagir com Assistentes



Alejandra Caggiano

Uma vez criado um assistente, testado no momento da sua criação e depois testado também a partir do Playground, o consideramos válido.

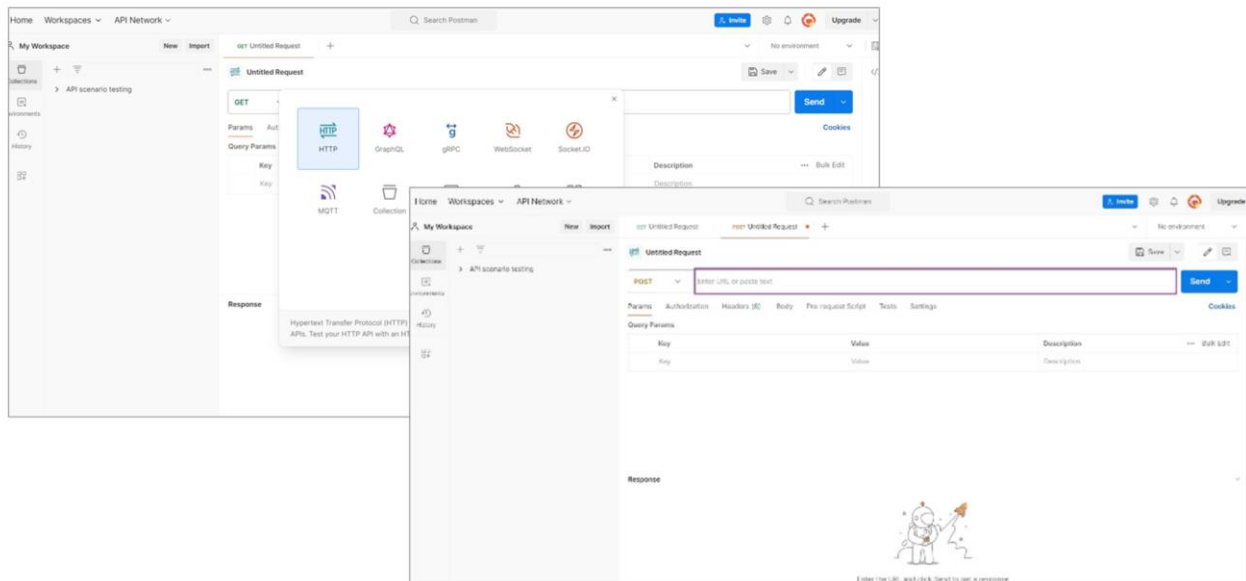
Postman API Platform

Passaremos agora a testá-lo via API através de Postman API Platform, uma plataforma para criar e utilizar APIs.

A partir daí, podemos utilizar a API em nossos desenvolvimentos, independentemente da tecnologia

.

Postman API Platform

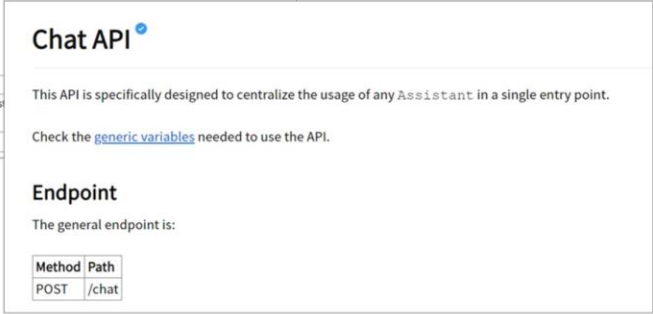
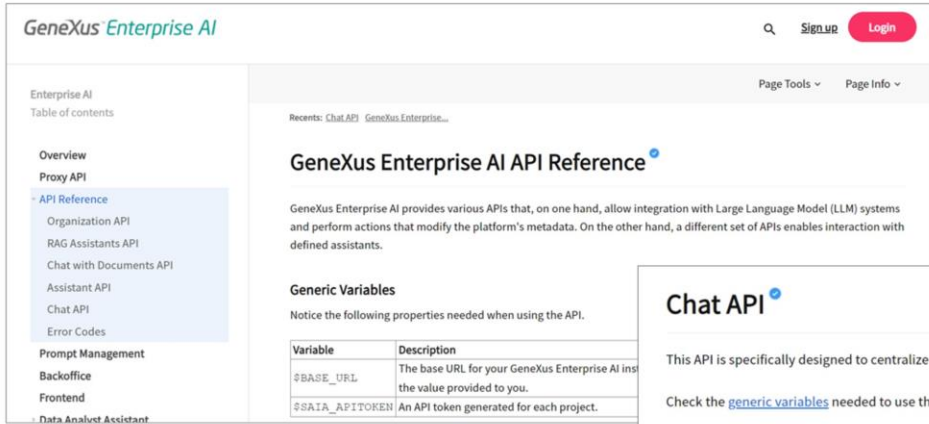


Entramos então no Postman e a partir de My Workspaces selecionamos New... HTTP:

Como nosso objetivo é enviar, postar uma consulta para receber uma resposta, o que faremos é um POST.

Então aqui escolhemos POST, e devemos indicar a URL

de acesso ao server. Para determinar esta URL, consultamos a documentação de GeneXus Enterprise AI



Nosso objetivo é nos conectar com o assistente para chat que criamos anteriormente, denominado Marketing Assistant, portanto, precisamos da informação necessária para nos comunicarmos com este tipo de assistentes.

É importante levar isso em consideração, pois se a ideia fosse conectar, por exemplo, com um RAG Assistant, precisaríamos de outros parâmetros.

Bem. Assim, em API References, Chat API, encontramos a informação que precisamos e também exemplos de uso:

Vemos que para utilizar a API precisamos do conteúdo das variáveis `BASE_URL` e `SAIA_API_TOKEN`

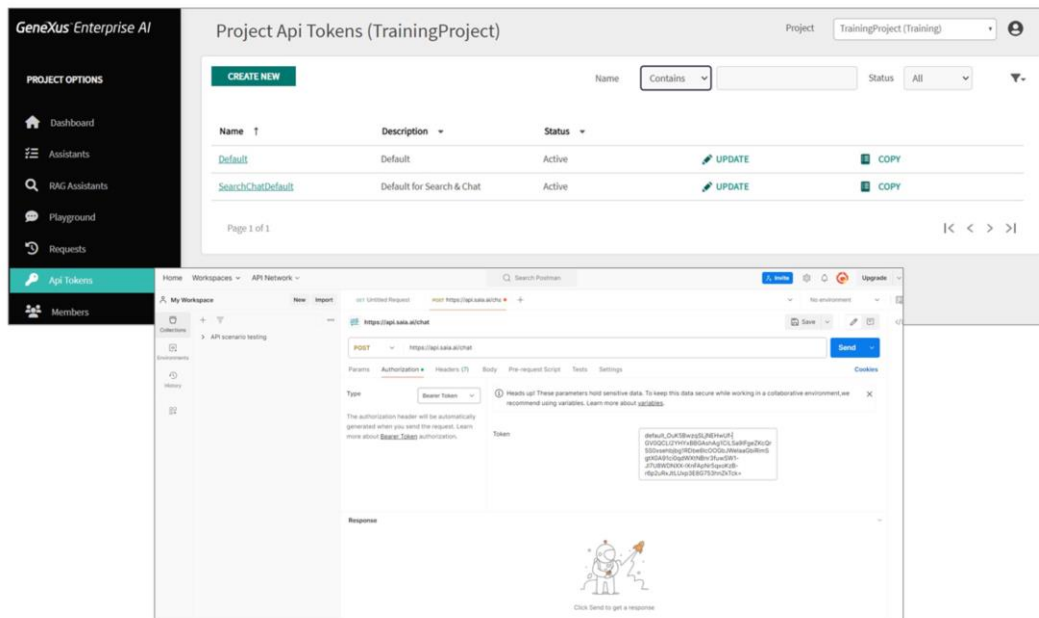
Onde `BASE_URL` corresponde à URL base de instalação do GX Enterprise AI. e `Saia_apitoken` corresponde a um api token gerado para o projeto.

Bem, então indicamos a url base correspondente ao nosso cenário

```
https://api.qa.saia.ai
```

vemos que o path se completa com `/chat`, então o adicionamos na declaração do POST.

Postman API Platform



Passamos agora para a Autorização, que é Bearer token, então na aba Authorization do Postman escolhemos essa opção.

Para utilizar a API, devemos autenticar cada solicitação através de um API token. Como já vimos, estes tokens são gerenciados no Backoffice de GeneXus Enterprise AI e identificam de forma única o remetente da solicitação.

Devemos então fornecer esse token. Para isso, vamos ao Backoffice de GeneXus Enterprise AI, à opção API Tokens e podemos selecionar qualquer um dos tokens definidos, já que qualquer um deles nos identifica no projeto.

Por exemplo, o Default. Pressionamos Copy.

Voltamos ao Postman e colamos o 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  
'
```

Bem. Vamos agora para a aba Body, escolhemos RAW, JSON e vamos consultar um exemplo base do que devemos declarar aqui.

Já indicamos o POST, a forma de autorização e o tipo de conteúdo. O que devemos indicar agora é o corpo da consulta, então copiamos o que corresponde a DATA e colamos

“Model” corresponde ao tipo de assistente, seguido de seu nome. Como já mencionamos, dependendo do tipo de assistente os parâmetros podem variar.

O type “Assistant” identifica um assistente padrão, um Data Analyst Assistant e um API Assistant, enquanto o type “Search” identifica um RAG Assistant

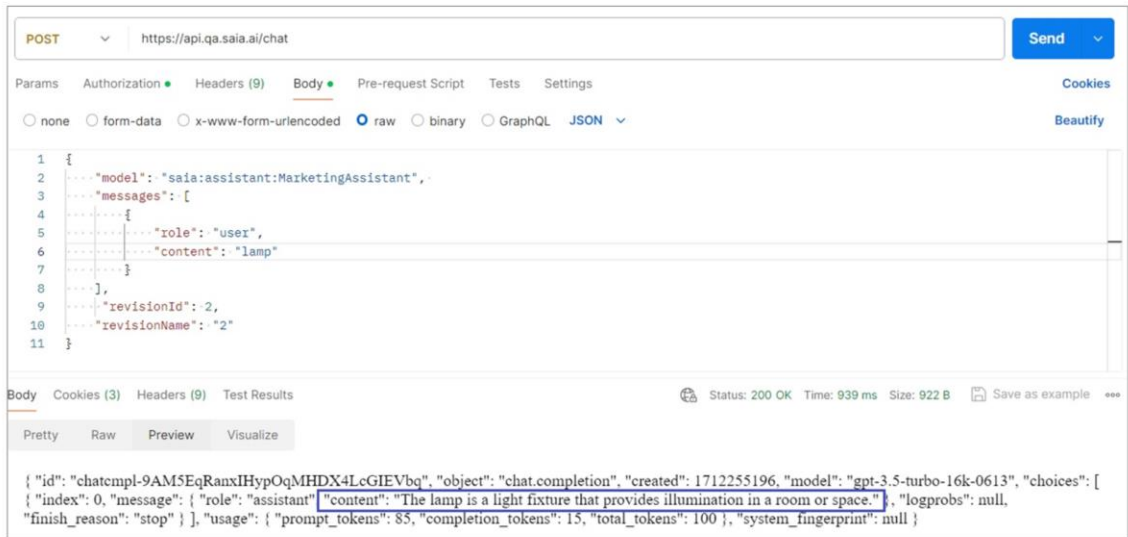
No nosso caso queremos comunicar com nosso assistente para chat, então o type indicado é “assistant” e o nome é “Marketing Assistant”

Vejamos agora o elemento Messages.

Este elemento define uma mensagem que você deseja adicionar. A expressão mínima é esta que estamos declarando, onde “content” corresponde ao input do usuário.

No nosso exemplo, como input, vamos indicar “lâmpada”, lamp, esperando que o assistente nos devolva uma boa descrição deste produto.

Postman API Platform



Depois podemos adicionar outros parâmetros, por isso vamos indicar, por exemplo, a revisão do assistente que queremos utilizar. Lembremos que poderíamos ter várias revisões de um mesmo assistente.

Indicamos então o identificador e nome da revisão

“revisionId”: 2

“revisionName”: “2”

Pressionamos agora Send e obtemos a resposta:

O assistente nos diz que “A lâmpada é um dispositivo de iluminação que fornece luz em um ambiente ou lugar.”

Muito bem. Agora que testamos nosso assistente via API, podemos utilizá-lo em nossos desenvolvimentos.

Mais adiante veremos exemplos de uso a partir de uma base de conhecimento GeneXus.

GeneXus[™]
by **Globant**

training.genexus.com