

# API do Assistente



Alejandra Caggiano

Dentro do conjunto de APIs disponíveis, GeneXus Enterprise AI oferece uma que permite a criação de novos assistentes, a modificação de suas definições e a recuperação de sua informação.

## API do Assistente

### Variáveis genéricas

Variável	Descrição
\$BASE_URL	URL base para a instalação de GeneXus Enterprise AI
\$SAIA_APITOKEN	Um API token gerado para cada projeto

### Métodos disponíveis

Method	Path	Description
GET	/assistant/{id}	Gets assistant data
POST	/assistant	Creates a new assistant
PUT	/assistant/{id}	Updates an assistant
DELETE	/assistant/{id}	Deletes an assistant
POST	/assistant/text/begin	Begins a text conversation with the GeneXus Enterprise AI Assistant
POST	/assistant/text	Sends a text prompt to the GeneXus Enterprise AI Assistant
POST	/assistant/chat	Sends a chat request to the GeneXus Enterprise AI Assistant
GET	/assistant/request/{id}/status	Retrieves the status of a request
POST	/assistant/request/{id}/cancel	Cancela a request

Para utilizar esta API devemos considerar as variáveis genéricas que já conhecemos:

Base\_URL e SAIAAPIToken

E também é necessário um API token de GeneXus Enterprise AI relacionado ao escopo da organização.

Os métodos disponíveis para esta API são os seguintes. Como exemplo, vamos testar o método POST que permite criar um novo assistente.

## API do Assistente: POST assistant

### cURL Sample

```
curl -X POST "$BASE_URL/v1/assistant" \  
-H "Authorization: Bearer $$AIA_PROJECT_API_TOKEN" \  
-H "accept: application/json" \  
-d '{  
  "type": "chat",  
  "name": "TestAssistant",  
  "prompt": "translate the following text to Esperanto"  
}'
```

<https://api.qa.saia.ai/v1/assistant>

O cURL sample correspondente, que lembramos estarem disponíveis na documentação técnica de GeneXus Enterprise AI, indica que a url deve ter a seguinte forma:

Portanto, em nosso exemplo, a url é a seguinte

<https://api.qa.saia.ai/v1/assistant>

onde esta parte corresponde ao conteúdo da variável BASE\_URL: aplicada ao nosso contexto.

## API do Assistente: POST assistant

The image shows two overlapping screenshots. The top one is from the Genexus Enterprise AI interface, displaying the 'Project Api Tokens (TrainingProject)' page. It features a table with columns for Name, Description, and Status. The bottom screenshot is from Postman, showing a POST request to the endpoint 'https://api.qa.saaia.ai/v1/assistant'. The request body is a JSON object with the following structure:

```
1 {
2   "type": "chat",
3   "name": "MyFrenchAssistant",
4   "prompt": "The user proposes a small paragraph, and the expected response is its translation into French. For example, for the input 'My name is Alejandra and I live in Uruguay. We are in autumn.', the expected response is 'Je m'appelle Alejandra et je vis en Uruguay. C'est l'automne.'"
5 }
6
7
8
9
10
11
12
13
```

The response body is also shown in JSON format:

```
1 {
2   "assistantDescription": "MyFrenchAssistant",
3   "assistantId": "910018b0-c704-4a18-92b0-34cb24de5ccf",
4   "assistantName": "MyFrenchAssistant",
5   "assistantPriority": 0,
6   "assistantStatus": 1,
7   "assistantType": "TextPromptAssistant",
8   "intents": [
9     {
10      "assistantIntentDefaultRevision": "1",
11      "assistantIntentDescription": "Default",
12      "assistantIntentId": "01e2270a-5d70-427e-8492-9ceca1283e06",
13      "assistantIntentName": "Default",
14    }
15 ]
16 }
```

### MyFrenchAssistant

"My name is Alejandra and I live in Uruguay. We are in autumn."

"Je m'appelle Alejandra et je vis en Uruguay. C'est l'automne."

Bem. Então entramos no Postman e declaramos o POST.

Vemos que o tipo de autorização é Bearer e é necessário um Project api token

Então, a partir da plataforma, vamos até a opção Api tokens e copiamos o default.

Voltamos então ao Postman e definimos a autorização necessária

Em type colocamos Bearer token, e colamos o token.

Para definir o corpo precisamos indicar o tipo de assistente, seu nome e prompt.

Nossa intenção é criar um assistente para chat que nos retorne a tradução para o francês de um pequeno parágrafo inserido pelo usuário.

Seu nome será "MyFrenchAssistant" e o prompt dirá que se trata de um assistente pessoal. O usuário propõe um pequeno parágrafo e a resposta esperada é sua tradução para o francês. Por exemplo, para o input "Meu chamo Alejandra e moro no Uruguai. Estamos no outono.", a resposta esperada é "Je m'appelle Alejandra et je vis en Uruguay. C'est l'automne."

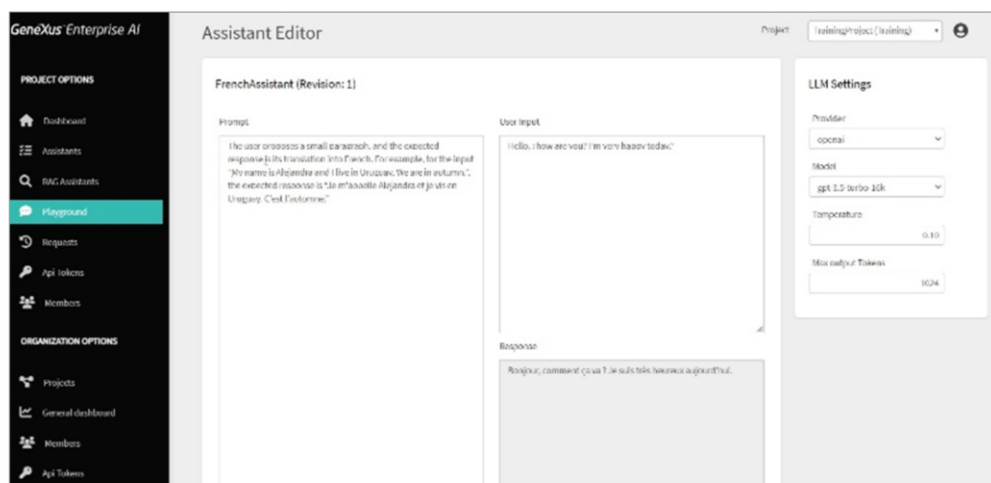
Então na aba body, escolhemos Raw, Json e definimos a estrutura.

O tipo de assistente é Chat, o nome é MyFrenchAssistant e colamos o prompt que mencionamos anteriormente.

Pressionamos Send e vemos a resposta com a criação do assistente.

Entre toda a informação que nos é fornecida, vemos que foi criada a revisão 1, ou seja, a primeira versão da definição de assistente, (lembramos que um assistente pode ter várias revisões), e vamos levar especialmente em conta o Id, pois é um parâmetro necessário na execução de outros métodos da API, como GET, PUT e DELETE.

## Testando o assistente: Edit prompt



Bem, o que queremos agora é testá-lo. E para isso temos várias opções.

Uma opção é entrar diretamente na plataforma e verificar se o assistente MyFrenchAssistant está definido.

Podemos ver o prompt definido, inserir um input e testá-lo.

Por exemplo, vamos dizer "Olá, como vai você? Eu estou muito feliz hoje." Pressionamos Test e vemos sua tradução para o francês.

## Testando o assistente: Playground



Muito bem. Vamos testá-lo também a partir do Playground

Selecionamos o assistente, criamos um novo chat e indicamos desta vez que Uruguai é um país da América do Sul. Sua capital é Montevidéu.”

E vemos sua tradução.

## Testando o assistente: Postman API Platform

### cURL Sample

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

### Chat API

<https://api.qa.saia.ai/chat>

Perfeito. Resta-nos testá-lo também via API, e para isso vamos utilizar a API para chat.

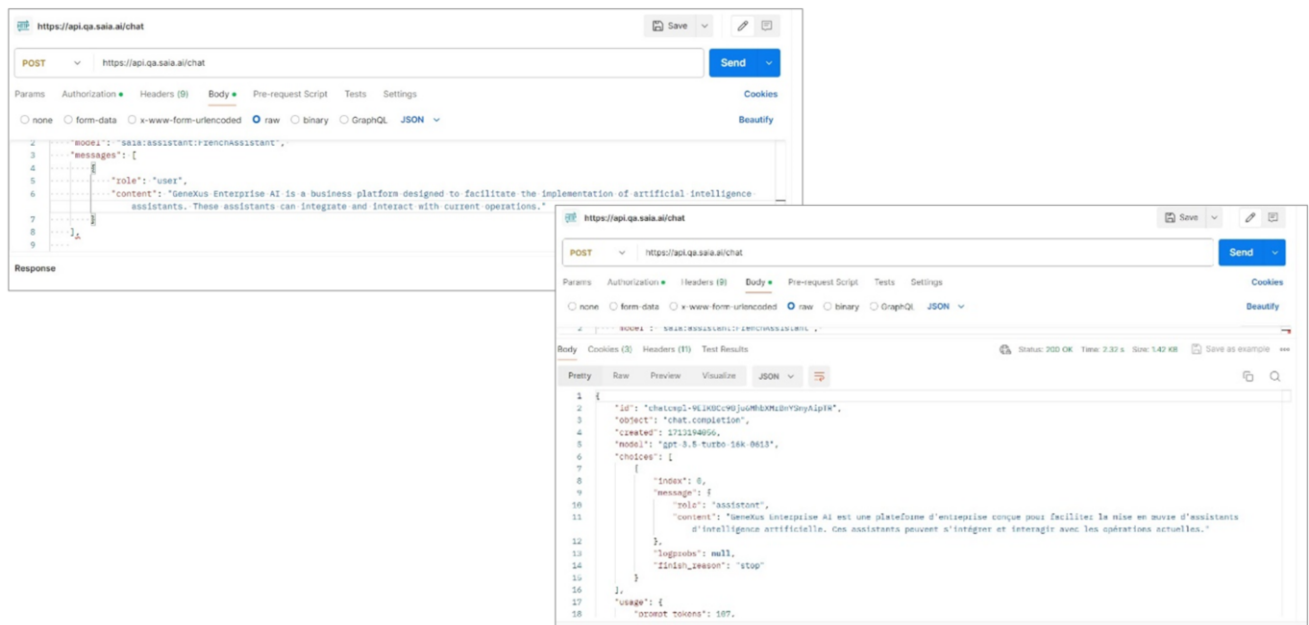
O cURL sample é o que estamos vendo e que já utilizamos anteriormente;

Vamos então definir um POST, e a url será a seguinte:

<https://api.qa.saia.ai/chat>



## Testando o assistente: Postman API Platform



Voltamos ao Postman e definimos o POST.

A autorização é Bearer e precisamos de um Project api token. Então vamos para a plataforma, API tokens, copiamos o Default e o colamos.

Vamos para Body, Raw, Json e definimos o corpo da solicitação.

Lembremos que "Model" corresponde ao tipo de assistente, seguido de seu nome.

Então indicamos o type "assistant" e o nome "MyFrenchAssistant".

Lembremos também que Messages define uma mensagem que você deseja adicionar, onde "content" corresponde ao input do usuário.

Indicaremos como input que "GeneXus Enterprise AI é uma plataforma de negócios desenhada para facilitar a implementação de assistentes de inteligência artificial. Esses assistentes podem ser integrados e interagir com as operações atuais."

Se necessário, poderíamos indicar outros parâmetros, como determinar a revisão do assistente que se deseja utilizar. Neste caso temos apenas uma e é a ativa por padrão, portanto não é necessário indicá-la.

Pressionamos Send e vemos a resposta.

## API do Assistente: DELETE assistant

### cURL Sample

```
curl -X DELETE "$BASE_URL/v1/assistant/{id}" \  
-H "Authorization: Bearer $SAIA_PROJECT_APITOKEN" \  
-H "accept: application/json"
```

```
"assistantId": "be96910c-d008-4697-846d-38accdf220e8",
```

<https://api.qa.saia.ai/v1/assistant/be96910c-d008-4697-846d-38accdf220e8>

Pois bem. Para finalizar, vamos eliminar o assistente que criamos e testamos através de todas as opções.

Para isso vamos utilizar o método Delete, que requer o AssistantId como parâmetro.

O Id de nosso assistente que agora queremos excluir é o que estamos vendo:

Portanto, a url necessária é a seguinte

<https://api.qa.saia.ai/v1/assistant/be96910c-d008-4697-846d-38accdf220e8>

Também precisamos de um Project api token para autorização.

# API do Assistente: DELETE assistant

The image shows two overlapping screenshots. The top one is a Postman interface for a DELETE request to `https://api.qa.saa.ai/v1/assistant/be96910c-d008-4697-846d-38accdf220e8`. The request body is empty, and the authorization is set to Bearer Token. A warning message states: "Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables." The token value is: `default_OuK5BwzqSLNEHwUF-GV0DQCLIZYhFyX8BGAeA8gTCLSa8FgeZKcOrSS0rsvhehojogjRDp6SlcOQObjWelaGcBRmSgX0A8F6nlogwW0N8l6v3UwC0M1-J7UBWdN00X-DXrP2uRkJILUx03`. The bottom screenshot shows the GenDius Assistant management interface. It features a sidebar with navigation options like "My Assistants", "Projects", and "Members". The main area is titled "Assistant" and contains a table of assistants. The table has columns for "Assistant Name", "Type", "Active Revisions", and "Last Update".

Assistant Name	Type	Active Revisions	Last Update	EDIT	DELETE	TEST PREVIEW	ALL REVISIONS
Model@assistant	FrontendChat	1	02/17/24 07:03:49	UPDATE	DELETE	TEST PREVIEW	ALL REVISIONS
Model@assistant	FrontendChat	1	02/17/24 06:52:45	UPDATE	DELETE	TEST PREVIEW	ALL REVISIONS

Vamos ao Postman e definimos a solicitação. Pressionamos Send.

Não há erros e se voltarmos à plataforma podemos verificar que o assistente foi excluído.

**GeneXus**<sup>™</sup>  
by **Globant**

[training.genexus.com](https://training.genexus.com)