Creation and use of APIs to interact with Assistants



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

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.

	CL gHPC Wedgesen Societ.0	X Second Costee Description Description C Second Se	Junto S A 🚱 Hayes
Ott Parama Ant Outry Parama Kay Kay Kay Magti Calent	CQ. gRPC Weddoore Societ.CO	X Search	Links © û 🚱 Unge
	Contentions	" Int liet ned Request	
Response Hypertext Transfer Protocol (PITTP) Affis. Test your HTTP Affi with as HT	A* At scenario secting Contraction At a contract of secting Attraction Attraction	POST V Enter UNIX or pactor low Neuron Authorizing Hondrin (M) Overy Prevents Overy Prevents	Sand v Coskies
	57	Key Value	Description Bulk Lot
	_	Resona	
	Hypertext Transfer Protocol (PT APIs, Text pour HTTP API with a	Pagantes Pagantest Tandar Protocol (HTT) Aftis. Sami poor HTT PAP with an et al.	Pageness Inguestes france/formation Induces 50 Body 7 frances/active Inguestes france/formation Signal Signal Signal Signal Inguestes france/formation Signal Signal Signal Signal Signal Inguestes france/formation Signal Signal

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.

Enterprise Al			Page Tools ~ Page Info ~		
Table of contents	Recents: Chat API GeneXus Enterprise				
Overview Proxy API	GeneXus	Enterprise AI API Reference	0		
- API Reference Organization API RAG Assistants API Chat with Documents API	GeneXus Enterpri and perform actic defined assistants	se Al provides various APIs that, on one hand, allow int ons that modify the platform's metadata. On the other l 5.	regration with Large Language Model (LLM) systems hand, a different set of APIs enables interaction with		
Assistant API Chat API Error Codes	Generic Variables Notice the following properties needed when using the API.		Chat API [°]		
Prompt Management	Variable	Description	This API is specifically designed to contraliz	o the usage of any Beeri stant in a single entry point	
Backoffice	\$BASE_URL	The base URL for your GeneXus Enterprise AI inst the value provided to you.	This AFTIS specifically designed to centraliz	e the usage of any Assistant in a single entry point.	
Frontend \$SAIA_APITOKEN An API token generated for each project.			Check the <u>generic variables</u> needed to use the API.		
			Endpoint The general endpoint is:		
			Method Path		

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.

GeneXus by Globant

PROJECT OPTIONS	CREATE NEW		Name	Contains ~	Status All	
A Dashboard	Name 1	Description -	Status -			
2프 Assistants	Default	Default	Active	P UPDATE	СОРУ	
Q RAG Assistants	SearchChatDefault	Default for Search & Chat	Active	IPDATE	СОРУ	
Playground Requests	Page 1 of 1				K	< > >
🔎 Api Tokens	Home Workspaces - API Network -		Q. Search Postman	(A sector	🗊 🗘 🍘 Upgrade 🗸	
Members .	A My Recent Average Norw Norw </td <td>at United Reset (and Programme) The set of the set of</td> <td>the sequent forget from the former of the sequence of the</td> <td>A Transpire the data sector while working in a caleboardine stations and the sector of the sector of</td> <td>Remembers V D</td> <td></td>	at United Reset (and Programme) The set of the set of	the sequent forget from the former of the sequence of the	A Transpire the data sector while working in a caleboardine stations and the sector of	Remembers V D	
		Response	Br.			

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.



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.

GX

Params	Authoriz	zation • He	aders (9)	ody • Pre-request Script Tests Se	ttings		Cookie
⊖ nor	ne O form	i-data 🔿 x-v	www-form-urle	ncoded O raw O binary O GraphQL	JSON V		Beautify
1 2 3 4 5 6 7 8 9 10 11	{ "mode "mess	el": "saia:a sages": [["role": "content risionId": 2 isionName":	"user", ": "lamp" , "2"	ketingAssistant",			
dy C	Cookies (3) Raw	Headers (9) Preview	Test Results Visualize		🛱 Status: 2	00 OK Time: 939 ms Size: 922	B 🖾 Save as example
{ "id" { "ind "finis	': "chatcmr lex": 0, "m h_reason":	ol-9AM5EqR essage": { "re "stop" }], "t	anxIHypOqM ole": "assista usage": { "pr	IHDX4LcGIEVbq", "object": "chat.cor at" ["content": "The lamp is a light fixtu mpt_tokens": 85, "completion_tokens"	npletion", "created": 17122551 re that provides illumination in : 15, "total_tokens": 100 }, "sy	96, "model": "gpt-3.5-turbo a room or space." }, "logpi stem_fingerprint": null }	0-16k-0613", "choices": [robs": null,

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.



training.genexus.com