Organization API



GeneXus Enterprise AI offers an API that provides endpoints to retrieve data about the organization, such as projects and requests.

It allows retrieving project details and exporting request data.

Organization API

Variables genéricas

Variable	Description
\$BASE_URL	The base URL for your GeneXus Enterprise AI installation.
\$SAIA_APITOKEN	An API token generated for each project.

Métodos disponibles	Method	Path	Description
	GET	/assistants	Get the list of assistants
	GET	/projects	Get the list of projects
	GET	/project/{id}	Get project details
	POST	/project	Create a project
	PUT	/project/{id}	Update a project
	DELETE	/project/{id}	Delete a project
	GET	/project/{id}/tokens	Get the list of Tokens for the project
	GET	/request/export	Export request data

To use this API, we must consider the generic variables we already know:

Base_URL and SAIAAPiToken.

The methods available for this API are as follows:

Let's test some of these methods.

First, let's test the GET method to get the list of assistants defined in the Organization.

This requires a GeneXus Enterprise AI API token related to the scope of the organization.

Organization API: GET assistants

cURL Sample

```
curl -X GET "$BASE_URL/v1/organization/assistants" \
    -H "Authorization: Bearer $SAIA_PROJECT_APITOKEN" \
    -H "Accept: application/json"
# using the full detail option change the URL to
    $BASE_URL/v1/organization/assistants?detail=full
```

Client URL command: cURL

https://api.qa.saia.ai/v1/organization/assistants

The parameter required by this method is the level of detail required in the response. The possible values are "summary", which is the default, or "full".

The "summary" option, which as we said is the default option, will only show the first level. On the other hand, the "full" option will also show the details of the review and the composition of the assistant.

Good. We will take as a basis the cURL sample we are looking at, which is available in the technical documentation of GeneXus Enterprise AI.

Remember that the Client URL (cURL) command is a command line tool for data exchange. We specify the server URL, that is, the location where we want to send a request, and the data to be sent to that server URL.

Then we see that the URL must look like this....

Therefore, our URL will be as follows:

https://api.qa.saia.ai/v1/organization/assistants

...where this part corresponds to the content of the variable BASE_URL applied to our context.

Postman API Platform

GET v https://api.qa.sa	aia.ai/v1/organization/assistants			Send ~			
arams Authorization • Heade	rs (7) Body Pre-request Script	Tests Settings		Cookies			
Key	Value	Description		+++ Bulk Edit			
Key	Value	GeneXus Enterprise Al	Project Api Tokens	(TrainingProject)			Project (TrainingProject (Training)
		PROJECT OPTIONS	CREATE NEW		Name	ains 💙	Status All 👻
		ft Dathboard	Name 1	Description +	Status +		
		注重 Assistants	Default	Default	Active	J UPDATE	Сору
		Q RAG Assistants	SearchChatDefault	Default for Search & Chat	Active	J UPDATE	COPY
		 Playground Remuests 	Page 1 of 1				1< < > >
of https://api.qa.sala.al/v1/organization/assi	istants	 Playground Bisquests 	Page 1 of 1				K < > >
양호 https://api.qa.sala.al/v1/organization/assi OET	istants ganization/assistants	 Playground Bequests 	Page 1 of 1				K < > >
Bittps://api.ga.sala.al/v/forganization/assi OET v https://api.ga.sala.al/v/forganization/assi Parama Authorization + Headers (7)	latants ganization/assistants Dody Pre-request Script Tests Settings	 Plagnoud Requests 	Page 1 of 1				К 🗘 Х
Intrps://api.qa.sala.al/vflorganization/assi OET v Intps://goi.qa.sala.al/vflor Parama Authorization • Headers (7) Type Bearer Token v	stants gerization/assistants Body Pre-request Script Tests Settings () Heads up! These parameters hold sensitive recommend using variable.	Plagnout To keep this data secure while working in a contractions.	Page 1 of 1				K < > 3
https://api.ga.sala.al/vf.lorganization/assi OET v https://api.ga.sala.al/vf.lorg Params Authorization + Headers (7) Type Reserv Token v The authorization header with the automaticality	astants Body Pre-request Script Tests Settings () Heads up? These parameters hold sensitive recommend using variables. Learn more abo	Plagnout To Repute	Page 1 of 1 Distance - Page 1 of 1 Send - Cookle Itaborative environment, are				(< > >)

Good. We enter Postman and declare the GET:

Note that the authorization type is Bearer and a Project Api Token is required.

So, from the platform, we go to the Api Tokens option and copy the default.

We go back to Postman and define the authorization required.

In type, we set Bearer token and paste the token.

To get the response, we click on Send.

We see the description, ID, name and type of each assistant defined, as well as the data of the corresponding project, which in our case is TrainingProject.

OK. As indicated in the sample, if we want to receive the complete details in the response, we must modify the URL of the GET in this way:

https://api.qa.saia.ai/v1/organization/assistants?detail=full

We do it. We click on Send again and now we see the complete details:

GeneXus by Globant

Organization API: POST project

cURL Sample

```
curl -X POST "$BASE_URL/v1/organization/project" \
    -H "Authorization: Bearer $SAIA_ORGANIZATION_APITOKEN" \
    -H "accept: application/json" \
    -d '{
        "name": "my Project",
        "description": "My awesome project"
}'
```

https://api.qa.saia.ai/v1/organization/project

Good. Let's see another example.

We are going to use the POST method to create a new project.

A GeneXus Enterprise AI API token related to the scope of the organization is also required. The cURL sample we need in this case is the following:

Therefore, the URL we need to indicate looks like this:

https://api.qa.saia.ai/v1/organization/project

Postman API Platform

Assistants	Name 1	Description *	Setus +								
RNG-had-stands	Takingte	Anthor Linking Organization	Active	UPDATE							
Playpount Reports ApThiers	Page L of L			m https://a	pi.qa.saia.ai/v1/organiz	ation/proje	ct		Save	× 0	9 E
Alambes				POST	https://api.qa.sa	ia.ai/v1/org	anization/project			Send	d -
Projects General decisioned				Params Au	thorization • Header	rs (9) I	ody • Pre-request Script Tests	Settings			Cookie
Api Mon				Туре	Bearer Toker	n ~	Heads up! These parameters hole recommend using variables. Lear	d sensitive data. To keep this data secure while working in a co rn more about <u>variables</u> .	Ilaborative enviro	nment,we	×
				The authorizati generated whe more about Be	on header will be autom n you send the request. arer Token authorization	Learn	Token	defaultorganiza_sGeRBbQK1grikQ5tW25- GiHRW-	1		
https://api	.qa.saia.ai/v1/organi	ization/project						3WJUyAqbPr90uuBQ0XzicRfsfB6J280ic6YBf kfL6i/Ju3a9IAaHso04UJa27Ab7nh_ILUWRfZ GolAqvvwvMY- 0Lsb1u2RBo5mqWHjLoU3XVTc-			
POST ~	https://api.qa.s	aia.ai/v1/organizati	on/project					x95uY3_Lksl2rzqJXAe_qzvRq0BS_LMc=	J		
Params Auth	orization • Heade	ers (9) Body •	Pre-request	Script Tests S binary O GraphQL	iettings JSON V						
1 4 2 ····· 3 ·····	"name": • "my · Proj "description": • "	ect", My-awesome-proj	ect"								

Once again in Postman, we indicate POST and paste the URL.

In this case, an Organization Api token is required, so we enter the platform to obtain it. And we paste it.

To define the body we need to indicate the name and description of the project to create.

So we go to the Body tab, choose Raw, JSON, and define the data.

We click on Send, and get the answer containing the new project creation information:

In order to effectively verify the creation of this new project also via API, let's quickly apply the GET method to obtain the projects defined in the Organization.

GeneXus by Globant

Organization API: GET projects

cURL Sample

```
curl -X GET "$BASE_URL/v1/organization/projects" \
    -H "Authorization: Bearer $SAIA_ORGANIZATION_APITOKEN" \
    -H "Accept: application/json"
# using the full detail option change the URL to
    $BASE_URL/v1/organization/projects?detail=full
# using the name option filter change the URL to
    $BASE_URL/v1/organization/projects?name=projectName
```

https://api.qa.saia.ai/v1/organization/projects

According to the corresponding cURL sample, we must use the following URL:

https://api.qa.saia.ai/v1/organization/projects

Postman API Platform

GET	~	https://ap	i.qa.saia.ai/v1/orç	ganization/pro	jects				
Params	Autho	erization •	Headers (7)	Body Pre-r	equest Scrij	ot Tests	Settings		
Туре		Bear	er Token 🗸	(1) Heads recom	s up! These nmend using	parameters I variables. L	hold sensitive da earn more about	ta. To keep this d variables.	lata se
The aut	horization	header will be	automatically						
denerate	ed when v	ou send the re	auest. Learn					A	Ctobu
souy (JOOKIES (1	Headers (c	5) Test Results					49	Statu
Pretty	Raw	Preview	Visualize	JSON V	=				
1	£								
2	"pi	ojects": [
2	"pi	ojects": [{							
2 3 4	"pi	ojects": [{ proje	ctDescription	: "My awes	ome projec	:t",			
2 3 4 5	*p1	ojects": [{ proje "proje	ctDescription ctId": "199420	': "My aweso 97a-be01-4c0	ome projec 0d-b114-e6	t", dc6e8217al	b",		
2 3 4 5 6	"pi	ojects": [{ "proje "proje "proje	ctDescription ctId": "199420 ctName": "my F	": "My aweso 97a-be01-4c0 Project"	ome projec 0d-b114-e6	t", dc6e8217al	b",		
2 3 4 5 6 7	"pı	ojects": [{	ctDescription ctId": "199420 ctName": "my f	": "My aweso 97a-be01-4c0 Project"	ome projec 9d-b114-ec	t", dc6e8217at	b",		
2 3 4 5 6 7 8	"pı	ojects": [{ proje "proje proje }, {	ctDescription ctId": "199420 ctName": "my F	": "My aweso 97a-be01-4c0 Project"	ome projec 0d-b114-ec	t", dc6e8217al	b",		
2 3 4 5 6 7 8 9	"pi	ojects": [{ "proje "proje }, { "proje	ctDescription' ctId": "199420 ctName": "my f ctDescription'	": "My aweso 97a-be01-4co Project" ": "Project	ome projec Od-b114-ec	dc6e8217al	b", terprise AI Co	purse*,	
2 3 4 5 6 7 8 9 10	"pī	ojects": [{	ctDescription ctId": "199420 ctName": "my f ctDescription ctId": "86abo	: "My aweso 97a-be01-4co Project" : "Project dec-53c3-4d	ome projec Od-b114-ec for the (9f-8d10-b2	dc6e8217al dc6e8217al eneXus Ent	b", terprise AI Co 6",	ourse",	
2 3 4 5 6 7 8 9 10 11	"pi	ojects": [{ *proje *proje }, { *proje *proje *proje	ctDescription ctId": "199420 ctName": "my f ctDescription ctId": "869abc ctName": "Tra:	": "My aweso 97a-be01-4c0 Project" ": "Project dec-53c3-4d iningProject	ome projec Od-b114-ec for the (9f-8d10-b8 t"	eneXus Ent ffd43cbeed	b", terprise AI Co 6",	ourse",	
2 3 4 5 6 7 8 9 10 11 12	"pi	ojects": [{ proje proje proje }, { uproje proje proje },	ctDescription ctId": "19942 ctName": "my f ctDescription ctId": "869ab ctName": "Tra:	": "My awess a7a-be01-4ct Project" ": "Project dec-53c3-4dt IningProject	ome projec 9d-b114-ed for the C 9f-8d10-b8	eneXus Ent ffd43cbeed	b", terprise AI Co	burse",	
2 3 4 5 6 7 8 9 10 11 12 13	"pi	ojects": [{ "proje "proje "proje }, { "proje "proje "proje "proje "proje "proje "proje "proje }, {	ctDescription ctId": "19942/ ctName": "my f ctDescription ctId": "869abd ctName": "Tra:	": "My awess #7a-be01-4ct Project" ": "Project dec-53c3-4dt iningProject	ome projec 0d-b114-e6 for the (9f-8d10-b8 t"	t", dc6e8217al eneXus Ent iffd43cbeed	b", terprise AI Cc	ourse",	
2 3 4 5 6 7 8 9 10 11 12 13 14	"p1	ojects": [{ "proje "proje "proje }, { "proje "proje "proje }, { "proje	ctDescription ctId": "199424 ctName": "my f ctDescription ctId": "869abd ctName": "Tra: ctDescription	": "My awess a7a-be01-4c0 Project" ": "Project dec-53c3-4d IningProject ": "GeneXus	ome projec 0d-b114-ec for the (9f-8d10-b2 t" Enterpris	eneXus Ent ieneXus Ent iffd43cbeed	b", terprise AI Cc 6", se",	ourse",	
2 3 4 5 6 7 8 9 10 11 12 13 14 15	*p1	ojects*: [{ *proje *proje *proje *proje *proje *proje *proje	ctDescription ctId": "19942d ctName": "my i ctDescription ctId": "869abd ctName": "Tra: ctDescription ctId": "462191	": "My aweso 97a-be01-4ci Project" ": "Project dec-53c3-4d iningProject ": "GeneXus 568-1cef-4a	ome projec 9d-b114-ec for the 0 9f-8d10-b2 t" Enterpris 45-9ed1-81	eneXus Ent idc6e8217al iffd43cbeed ie AI Court 3e332bdd3d	b", terprise AI Cc 6", se", a",	ourse",	

So again from Postman we define the request:

Also in this case, as indicated in the sample, an Organization Api Token is required, so we get it and complete the definition of the request.

We click on Send and check that the new project is part of the list of projects returned in the response.

In this way, it is possible to test the other methods available in this API.

Remember that later on we will be able to integrate these APIs into our development work regardless of the technology.



training.genexus.com