# API da Organização



GeneXus Enterprise AI oferece uma API que fornece endpoints para recuperar dados da organização, como projetos e solicitações. Permite recuperar detalhes de um projeto e também exportar dados de solicitações.

# API da Organização

#### Variáveis genéricas

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

Métodos disponíveis	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

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

Base\_URL e SAIAAPiToken

Os métodos disponíveis para esta API são os seguintes:

Vamos então testar alguns desses métodos.

Em primeiro lugar, vamos testar o método GET para obter a lista de assistentes definidos na Organização.

Isto requer um API token de GeneXus Enterprise AI relacionado ao escopo da organização.

## API da Organização: 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

O parâmetro requerido por este método é o nível de detalhe que se requer na resposta. Os valores possíveis são "summary", que é o valor padrão, ou "full"

A opção de "resumo", que como dissemos é a opção padrão, mostrará apenas o primeiro nível. Por outro lado, a opção completa, "full", mostrará também os detalhes da revisão e a composição do assistente.

Bem. Tomaremos como base o cURL sample que estamos vendo e que está disponível na documentação técnica de GeneXus Enterprise AI.

Lembremos que o comando Client URL (cURL) é uma ferramenta de linha de comandos que permite a troca de dados. Você especifica a URL do servidor, ou seja, o local para onde deseja enviar uma solicitação e os dados que deseja enviar para essa URL do servidor.

Vemos então que a URL deve ter esta forma....

Então nossa url será a seguinte:

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

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

## **Postman API Platform**

GET	<ul> <li>https://api.qa.sai</li> </ul>	a.ai/v1/organization/assistants			Send ~			
Params	Authorization • Header	s (7) Body Pre-request Script	Tests Settings		Cookies			
•	Cey	Value	Description		+++ Bulk Edit			
1	Key	Value	GeneXus Enterprise Al	Project Api Tokens (	(TrainingProject)			Project (Training)
			PROJECT OPTIONS	CREATE NEW		Name Conta	m v]	Status All 👻
			ft Dashboard	Name 1	Description +	Status +		
			2 Assistants	Default	Default	Active	J UPDATE	Сору
			Q RAG Assistants	SearchChatDefault	Default for Search & Chat	Active	& UPDATE	COPY
			Playground	Page 1 of 1				I< < > >
			'9 Requests					
m https	://api.qa.sala.al/v1/organization/assis	tants		🖾 Save 🗸 🥒 💬				
DET https	://api.qa.sala.al/v1/organization/assis ~ https://api.qa.sala.al/v1/org	tants anization/assistants		🖾 Save 🗸 🥒 🖾				
0ET Params	//api.qa.sala.al/v1/organization/assis v https://api.qa.sala.al/v1/org Authorization + Headers (7) B	tants anization/assistants ody Pre-request Script Tests Settings		Save V C				
GET Params Type	(/api.qa.sala.al/Vforganization/assis v https://api.qa.sala.al/Vforg Authorization + Headers (7) B Bears Token v	tants inization/assistants ody Pre-request Script Tests Settings O Heads up! These parameters hold sensitive recommend using variables. Learn more ab	data. To keep this data secure while working in a co	Save V P C				
DE https OET Params Type The authorit generated y	(/api,qa.sala.al/V/forganization/assis v https://epi.qa.sala.al/V/forg Authorization • Headers (7) B itearer Token v auton header will be automatically entron used with the result. Learn	tants wrization/assistants only Pre-request Soript Tests Settings () Heads of These parameters hold sensitive recommend using variables. Learn more abo	data. To keep this data secure while working in a co of satisfies	Send Cookies				

Bem. Entramos então no Postman e declaramos o GET:

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 ao Postman e definimos a autorização requerida

Em type colocamos Bearer token, e colamos o token

Para obter a resposta pressionamos Send

Vemos a descrição, id, nome e tipo de cada assistente definido, e vemos também os dados do projeto correspondente, que no nosso caso é TrainingProject

Contudo. Conforme indicado no sample, se quisermos receber o detalhe completo na resposta, devemos modificar a url do GET desta forma:

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

Fazemos. Pressionamos Send novamente e vemos agora o detalhe completo:

**GeneXus by Globant** 

# API da Organização: 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

Bem. Vejamos outro exemplo.

Vamos utilizar o método POST para criar um novo projeto.

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

O cURL sample que precisamos neste caso é o seguinte:

Portanto, a url que devemos indicar fica desta forma:

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

## **Postman API Platform**

Asistanta	Name 1 De	scription .	Salus *								
RNG-haldsterks	Betauthgarbarian Beta	aut tor deputation igificant	Action	/ UPDATE	COM						
Pagesard Report AcTrinov	Page Left	or hennig ciganasole	Adre	https://	api.qa.saia.ai/v1/organi:	tion/proje	ct		Save	× .	0 E
Mandan				POST	https://api.qa.sa	a.ai/v1/org	anization/project			Sen	d
Projects General clashboard				Params A	uthorization • Heade	(9) E	ody  Pre-request Script Tests	Settings			Cookie
Api teles				Туре	Bearer Toke	~	Heads up! These parameters hold recommend using variables. Learn	sensitive data. To keep this data secure while working in a co more about <u>variables</u> .	llaborative enviro	nment,we	×
				The authorizating generated whi more about Be	tion header will be auton en you send the request earer Token authorization	itically Learn	Token	defaultorganiza_sGeRBbQK1grikQStW25- GiH8W-	1		
https://api.	qa.saia.ai/v1/organizat	tion/project						3WJUyAqbPr90uuBQ0XzicRfsfB6J280ic6YBf kfL6NJu3a9IAaHso04UJa27Ab7nh_ILUWRfZ GolAqvvwvMrY- 0Lsb1uJ2RB05mcWHiLoU3XVTc-			
POST ~	https://api.qa.saia.	.ai/v1/organizatio	on/project					x95uY3_Lksl2rzq XAe_qzvRq0BS_LMc=			
Params Autho	orization • Headers	(9) Body •	Pre-request Sc	ript Tests ary O GraphQ	Settings L JSON ~						
1 1	name": "my Project	awesome proj	ect"								

Novamente no Postman, indicamos POST e colamos a url

Vemos que neste caso é necessário um api token de organização, então entramos na plataforma para obtê-lo.

E o colamos. Para definir o corpo precisamos indicar o nome e descrição do projeto a criar.

Então vamos para a aba Body, escolhemos Raw, Json e definimos os dados.

Pressionamos Send e vemos a resposta com os dados da criação do novo projeto:

Como forma de verificar efetivamente a criação deste novo projeto também via API, vamos aplicar rapidamente o método GET que permite obter os projetos definidos na Organização

GeneXus by Globant

# API da Organização: 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

De acordo com o cURL sample correspondente, a url que devemos indicar é a seguinte:

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

# **Postman API Platform**

GET	https://api.qa.saia.ai/v1/organization/projects
Params	Authorization Headers (7) Body Pre-request Script Tests Settings
Туре	Bearer Token  V Heads up! These parameters hold sensitive data. To keep this data recommend using variables. Learn more about <u>variables</u> .
The autho	prization header will be automatically
aenerated	J when you send the request. Learn
	avies (1) meaners (0) meaning 460 area
Pretty	Raw Preview Visualize JSON V 🚍
1	
2	"projects": [
3	4
4	"projectDescription": "My awesome project",
	"projectId": "19942e7a-be01-4c0d-b114-e6dc6e8217ab",
5	
5	"projectName": "my Project"
5 6 7	"projectName": "my Project" },
5 6 7 8	"projectName": "my Project" }, {
5 6 7 8 9	<pre>"projectName": "my Project" }, {     "     projectDescription": "Project for the GeneXus Enterprise AI Course",</pre>
5 6 7 8 9 10	<pre>"projectName": "my Project" },  ' projectDescription": "Project for the GeneXus Enterprise AI Course", "projectId": "869abdec-53c3-4d9f-8d10-b8ffd43cbee6",</pre>
5 6 7 8 9 10 11	<pre>"projectName": "my Project" }, {     "projectDescription": "Project for the GeneXus Enterprise AI Course",     "projectId": "869abdec-53c3-4d9f-8d10-b8ffd43cbee6",     "projectName": "TrainingProject"</pre>
5 6 7 8 9 10 11 12	<pre>"projectName": "my Project" },  projectDescription": "Project for the GeneXus Enterprise AI Course",     "projectId": "869abdec-53c3-409f-8d10-b8ffd43cbee6",     "projectName": "TrainingProject" },</pre>
5 6 7 8 9 10 11 12 12	<pre>"projectName": "my Project" },  ' projectDescription": "Project for the GeneXus Enterprise AI Course",     "projectId": "869abdec-53c3-4d9f-8d10-b8ffd43cbee6",     "projectName": "TrainingProject" }, </pre>
5 6 7 8 9 10 11 12 13 14	<pre>"projectName": "my Project" }, {     "projectDescription": "Project for the GeneXus Enterprise AI Course",     "projectId": "869abdec-53c3-4d9f-8d18-bBffd43cbee6",     "projectName": "TrainingProject" }, {     "projectDescription": "GeneXus Enterprise AI Course",     "projectDescription": "GeneXus En</pre>
5 6 7 8 9 10 11 12 13 14 15	<pre>"projectName": "my Project" },  projectDescription": "Project for the GeneXus Enterprise AI Course",     "projectId": "869abdec-53c3-409f-8d10-b8ffd43cbee6",     "projectName": "TrainingProject" },  projectDescription": "GeneXus Enterprise AI Course",     "projectDescription": "GeneXus Enterprise AI Course",     "projectId": "d8219b68-1cef-4a45-9ed1-8f3a332bdd3a",</pre>
5 6 7 8 9 10 11 12 13 14 15 16	<pre>"projectName": "my Project" },  * projectDescription": "Project for the GeneXus Enterprise AI Course",     "projectId": "869abdec-53c3-4d9f-8d10-b8ffd43cbee6",     "projectName": "TrainingProject" },  * projectDescription": "GeneXus Enterprise AI Course",  * projectId16": "Ge19bb8-1cef-4a45-9ed1-8f3e332bdd3a",  * projectId16": "Course Project" </pre>

Então, novamente a partir do Postman, definimos a solicitação:

Também neste caso, conforme indicado no sample, é necessário um api token de Organização, então o obtemos e completamos a definição da solicitação.

Pressionamos Send e verificamos se o novo projeto faz parte da lista de projetos retornada na resposta.

Desta forma então, é possível testar os restantes métodos disponíveis nesta API.

Lembremos que depois poderemos integrar o uso dessas APIs em nossos desenvolvimentos independentemente da tecnologia.



training.genexus.com