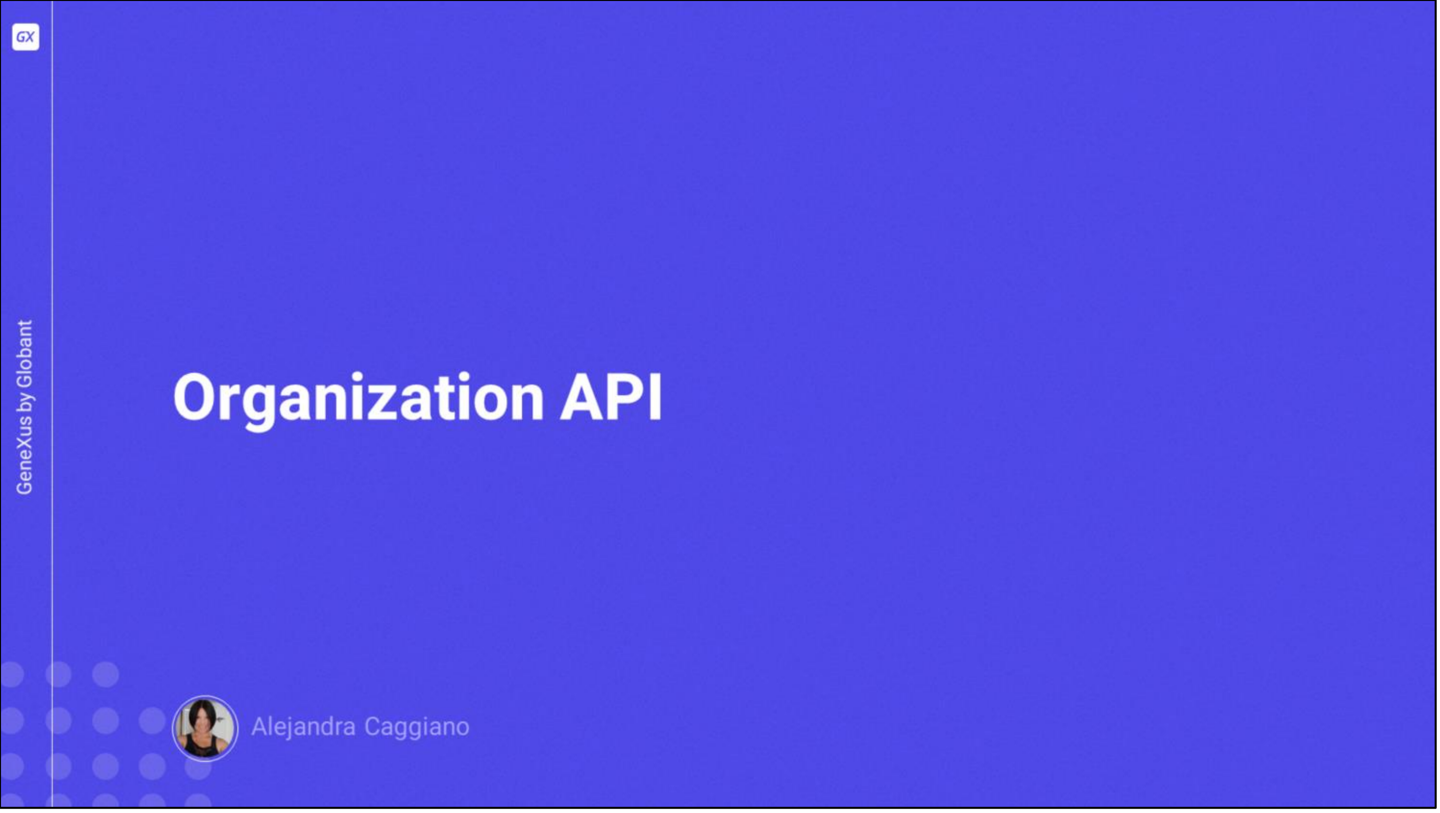


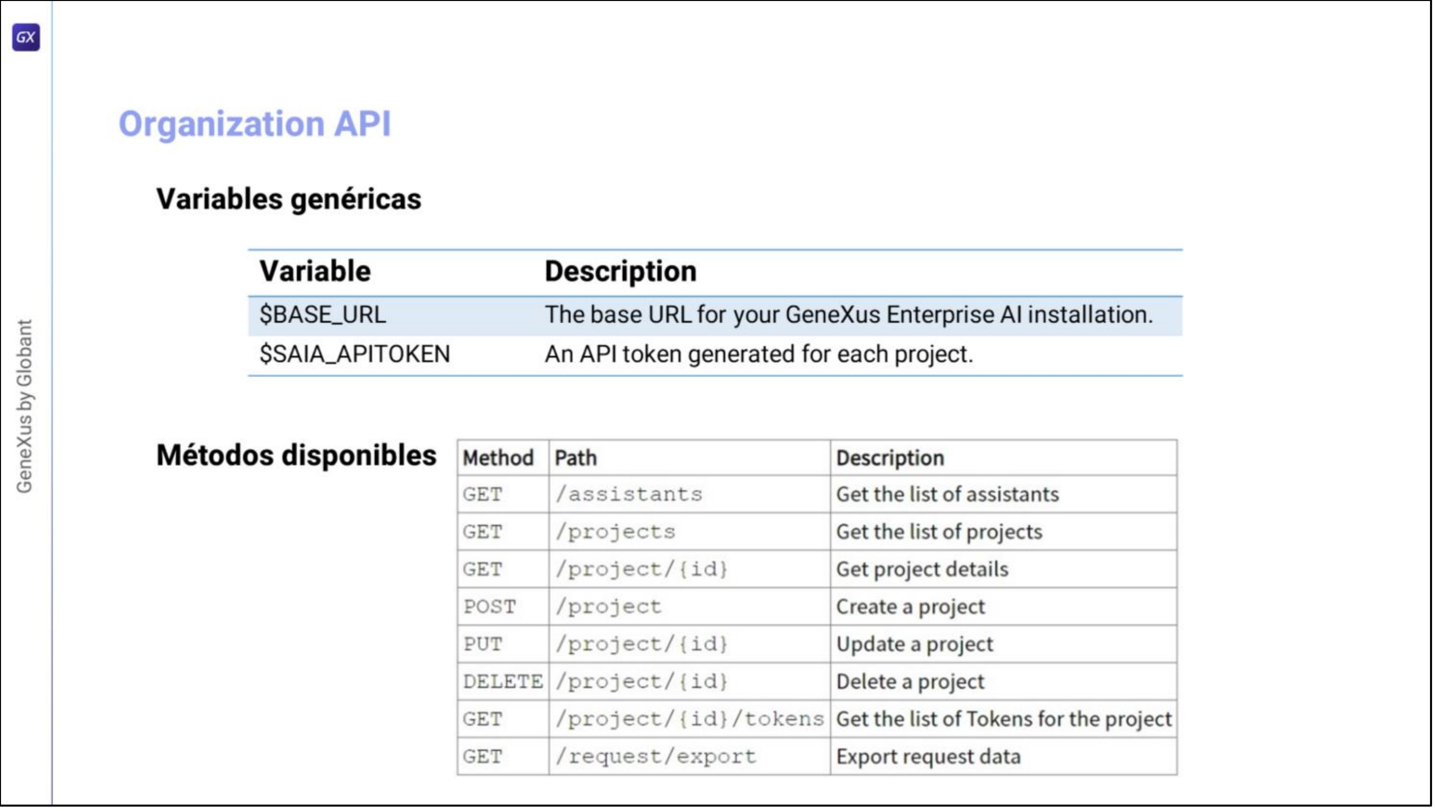
GeneXus by Globant

Organization API

Alejandra Caggiano

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:

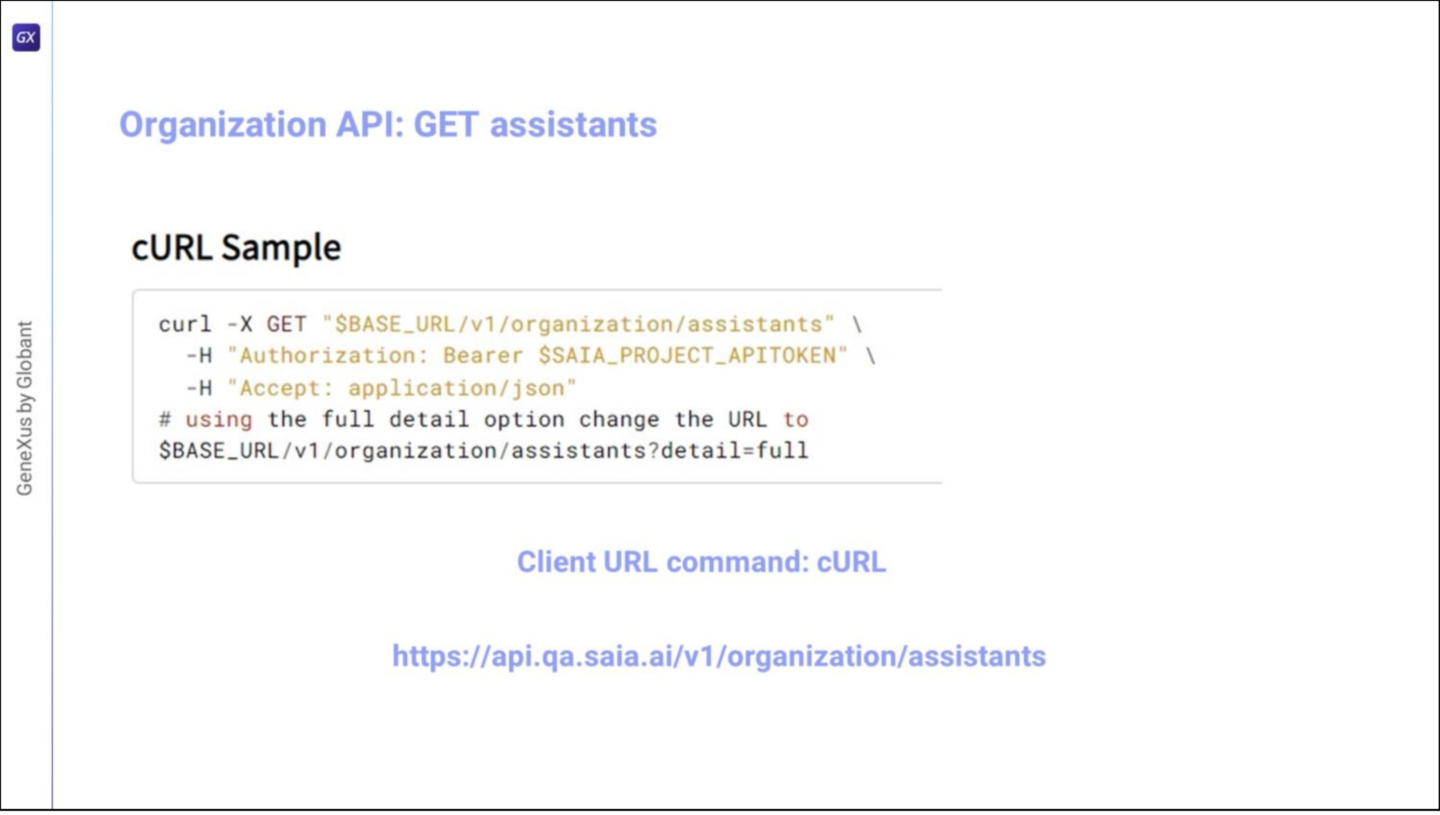
Base_URL and SAIAAPiToken.

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.



GeneXus by Globant

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.

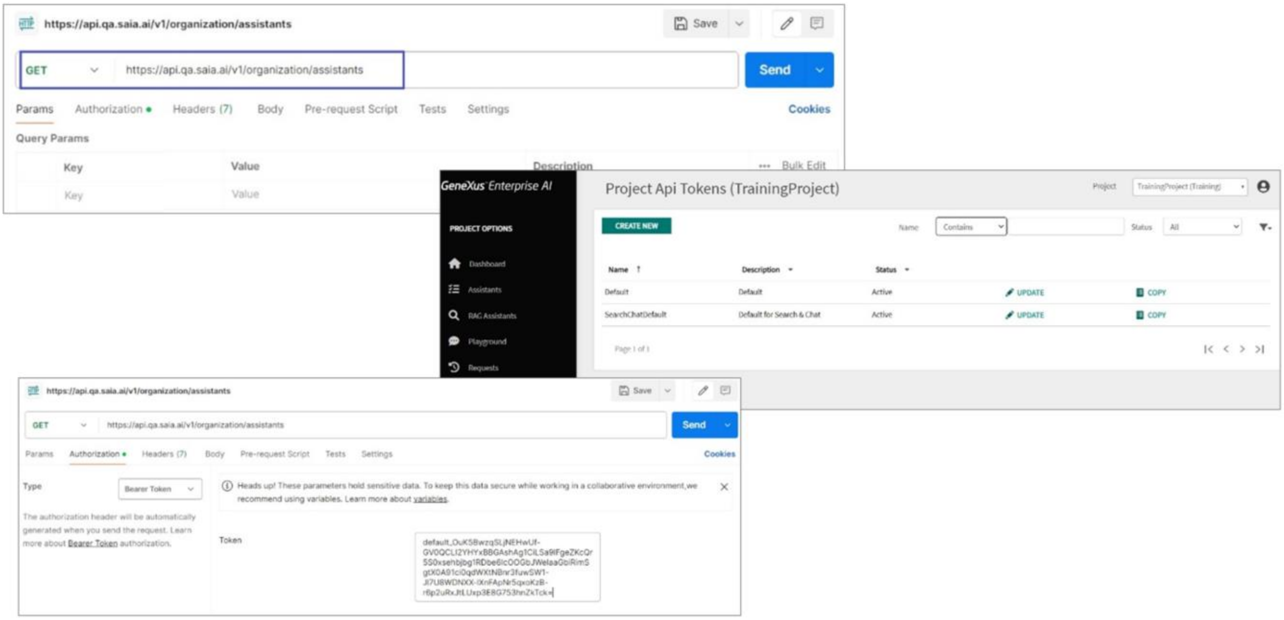
Then we see that the URL must look like this....

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



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

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

Query Params

Key	Value	Description
Key	Value	

Project Api Tokens (TrainingProject)

Project: TrainingProject (Training)

Name: Contains Status: All

Name	Description	Status		
Default	Default	Active	UPDATE	COPY
SearchChatDefault	Default for Search & Chat	Active	UPDATE	COPY

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

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

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about Bearer Token authorization.

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.

Token: default_OuK5BwzqSLjNEHwUf-GV0QCLI2YHYxBBGAshAg1CILSa9lFgeZKcQr5S0xsehbjbg1RDbe6lcOOGbJWelaaGbIRimSgtX0A91ci0qdWXtNBnr3fuwSW1-Jl7U8WDNXX-lXnFApNr5qxoKzB-r6p2uRxJtLUxp3E8G753hnZkTck=

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.

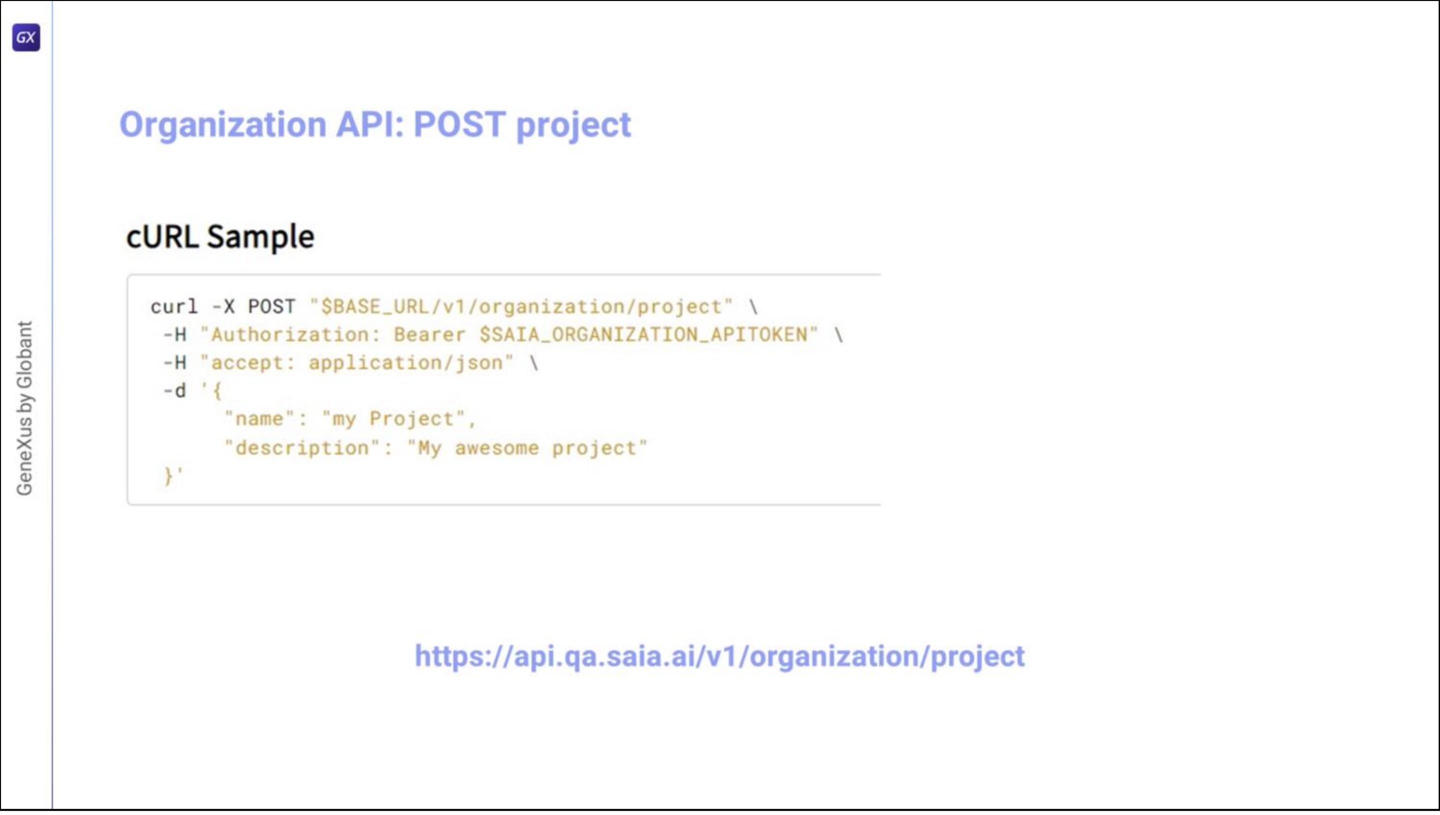
To get the response, we click on Send.

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:



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.

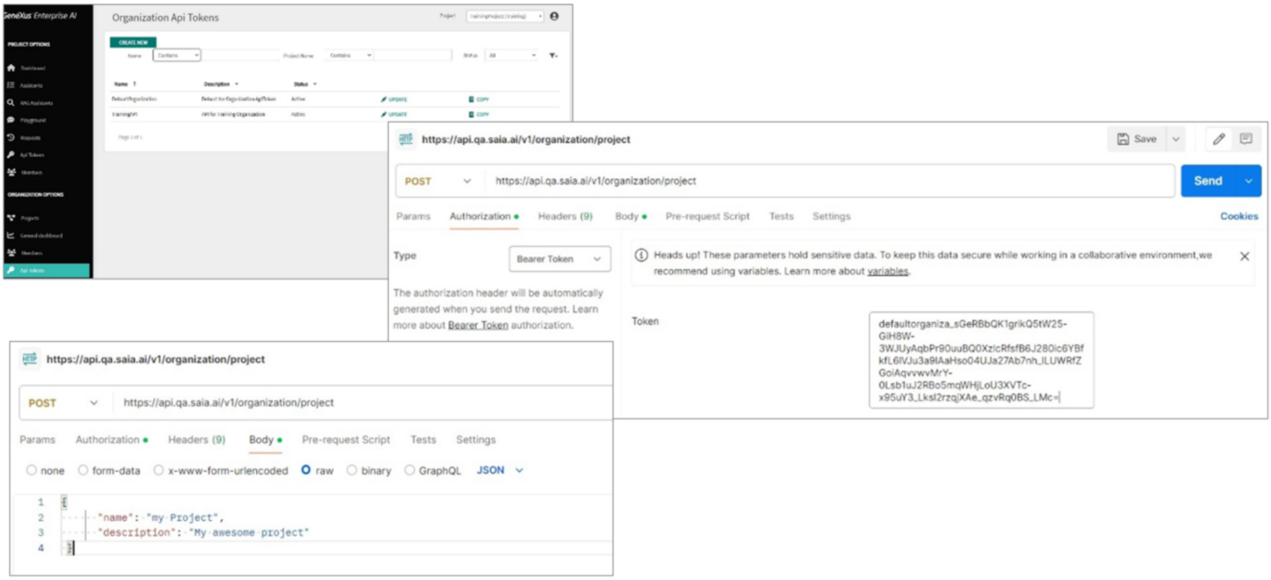
We are going to use the POST method to create a new project.

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



Organization Api Tokens

Project: trainingProject (training)

Name: Contains | Project Name: Contains | Status: All

Name	Description	Status
DefaultOrganization	Default for OrganizationApiToken	Active
trainingAPI	API for training Organization	Active

Page 1 of 1

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

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

Params | Authorization ● | Headers (9) | Body ● | Pre-request Script | Tests | Settings

Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about Bearer Token authorization.

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.

Token: defaultorganiza_sGeRBbQK1grikQ5tW25-GiH8W-3WJUyAqbPr90uuBQ0XzlcRfsfB6J280ic6YBfkfL6lVJu3a9lAaHso04UJa27Ab7nh_lLUWRfZGoiAqvvwvMrY-0Lsb1uJ2RBo5mqWHjLoU3XVTc-x95uY3_Lksl2rzqjXAe_qzvRq0BS_LMc=

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

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

Params | Authorization ● | Headers (9) | Body ● | Pre-request Script | Tests | Settings

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "name": "my Project",
    "description": "My awesome project"
}
```

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

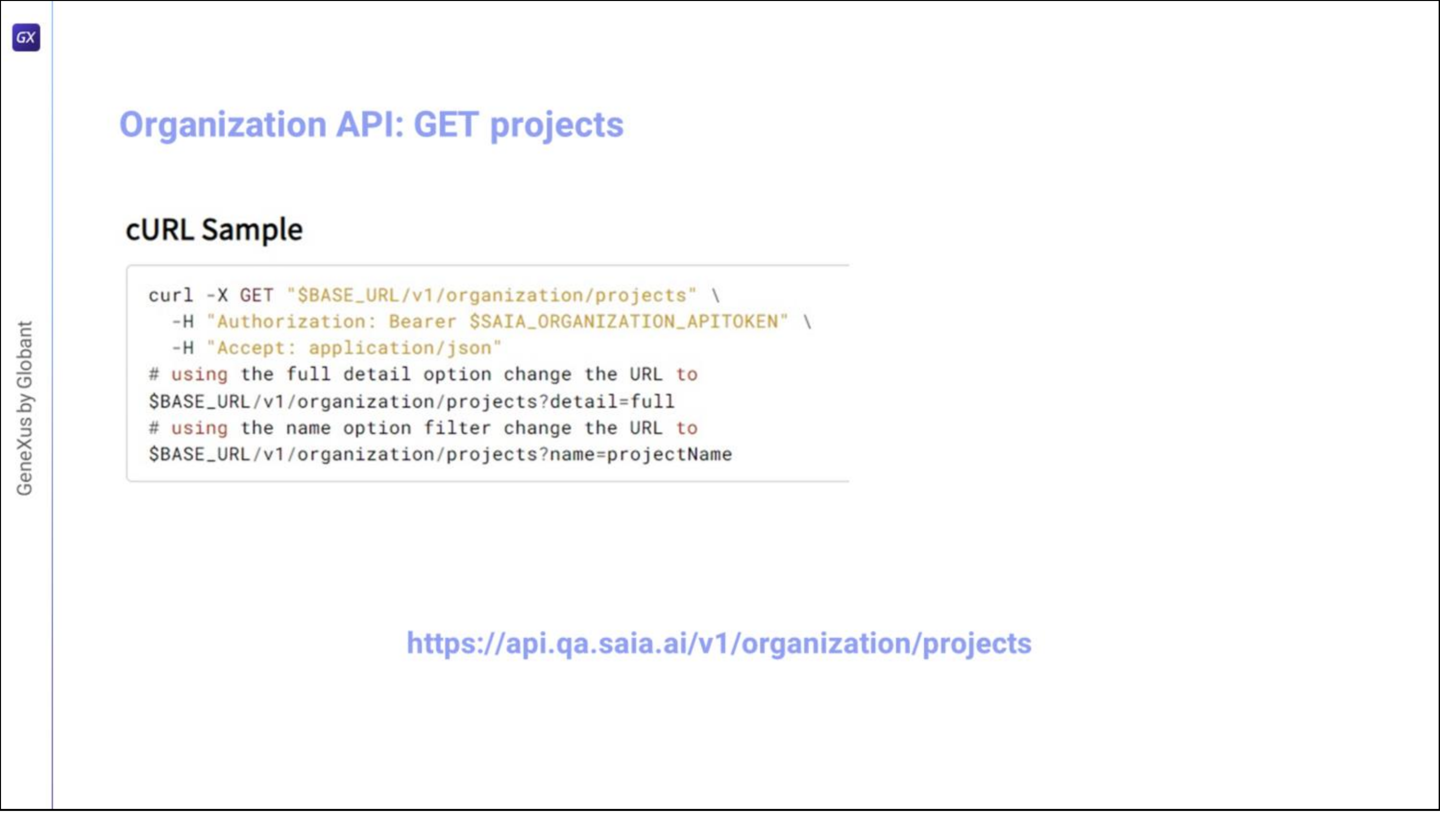
In this case, an Organization Api token is required, so we enter the platform to obtain it. And we paste it.

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.



Organization API: GET projects

cURL Sample

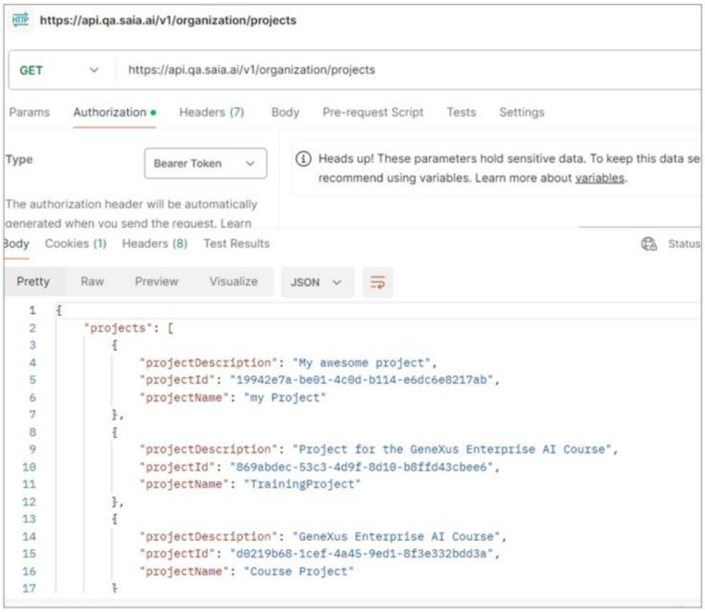
```
curl -X GET "$BASE_URL/v1/organization/projects" \
  -H "Authorization: Bearer $SAIA_ORGANIZATION_APITOKEN" \
  -H "Accept: application/json"
# using the full detail option change the URL to
$BASE_URL/v1/organization/projects?detail=full
# using the name option filter change the URL to
$BASE_URL/v1/organization/projects?name=projectName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

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



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

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

Type: Bearer Token

Heads up! These parameters hold sensitive data. To keep this data se
recommend using variables. Learn more about variables.

The authorization header will be automatically generated when you send the request. Learn

```
{
    "projects": [
        {
            "projectDescription": "My awesome project",
            "projectId": "19942e7a-be01-4c0d-b114-e6dc6e8217ab",
            "projectName": "my Project"
        },
        {
            "projectDescription": "Project for the GeneXus Enterprise AI Course",
            "projectId": "869abdec-53c3-4d9f-8d10-b8ffd43cbee6",
            "projectName": "TrainingProject"
        },
        {
            "projectDescription": "GeneXus Enterprise AI Course",
            "projectId": "d0219b68-1cef-4a45-9ed1-8f3e332bdd3a",
            "projectName": "Course Project"
        }
```

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.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com