

# API Assistant



Alejandra Caggiano

API Assistant is a feature of GeneXus Enterprise AI that allows you to define assistants to chat and interact with any external REST API that complies with the OpenAPI 3.0 specification or higher.

## API Assistant

This assistant leverages AI models and function calling capabilities to provide a dynamic way to chat with APIs using natural language.

This assistant leverages artificial intelligence models and function calling capabilities to provide a dynamic way to chat with APIs using natural language.

## API Assistant

- Compatibility with OpenAPI 3.0
- Customizable instructions
- Model settings
- Security
- Automated function calls:
- Multiple API calls

Its key features are as follows:

**Compatibility with OpenAPI 3.0:** It is possible to interact with any API that is built following the OpenAPI 3.0 specification or later.

**Customizable Instructions:** The user can provide specific text instructions to guide the AI assistant in making decisions and accurately handling requests.

**Model settings:** It is possible to adjust the AI model settings, such as temperature and the maximum number of tokens generated, to enhance and customize the assistant's answers according to the user's needs.

**Security:** Ensures the security of interactions through the use of API tokens. These tokens, which can be of Basic or Bearer type, are used in the Authorization header to authenticate and authorize requests, thus protecting information and data.

**Automated function calls:** It returns answers efficiently by making only the necessary API calls. This is achieved through GeneXus Enterprise AI's intelligent model and automated function calls that handle end user requests.

**Multiple calls to different APIs:** The assistant's intelligence is shown by making sequential calls to different APIs to obtain complex results. For example, upon receiving a question such as "What is the weather like in Tokyo and Montevideo?" the assistant identifies the need to query the weather API for each city mentioned. This ability to automatically handle multiple requests ensures a seamless experience for the user, who doesn't need to be involved in the process.

## API Assistant

### Prerequisites

- Configure your requests.

**For example:**

```
-H "Authorization: Bearer $OPENAI_API_KEY"
```

- Verify that the API Specification is documented according to OpenAPI 3.0 or higher, in JSON format.

Good. Before we start using this type of assistant, it is important to make sure that certain prerequisites are met:

If the API you want to interact with is authenticated with an API token (either Basic or Bearer), you need to configure the requests to include this token in the authorization header, specifying the correct type.

```
-H "Authorization: Bearer $OPENAI_API_KEY"
```

In this way, when the call is executed, the token will be used for the interaction.

You should also verify that the API specification is documented according to OpenAPI 3.0 or higher, in JSON format. This is important because APIs that do not comply with this specification may not be fully supported.

## API Assistant

### Example: Weather API

Get the OpenAPI 3.0 specification for the API you want to call.

```
{
  "openapi": "3.0.0",
  "info": {
    "title": "Weather API",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "https://weathergpt.vercel.app",
      "description": "Test"
    }
  ],
  "host": "https://weathergpt.vercel.app",
  "paths": {
    "/api/weather": {
      "get": {
        "summary": "Get current weather information",
        "operationId": "checkWeatherUsingGPT",
        "parameters": [
          {
            "name": "location",
            "in": "query",
            "required": true,
            "description": "Location for which to retrieve weather information.",
            "schema": {
              "type": "string"
            }
          }
        ]
      },
      "responses": {
        "200": {
          "description": "Current weather information",
          "content": {
            "application/json": {
              "schema": {
                "type": "object",

```

Ln 1, Col 1 7.330 caracteres.

Now, let's see an example.

The first thing to do is to obtain the OpenAPI3.0 specifications for the API we want to interact with, which in our case will be the Weather API.

## API Assistant

The screenshot shows the 'Assistant API Editor' interface. At the top right, the 'Project' is set to 'TrainingProject (Training)'. The main area is titled 'New Assistant' and contains a 'General Information' section with an 'Api Description (Instructions)' field. The description text is: 'The final response must strictly comply with the following: It must be in the same language as the user's request. The response text must be formatted as a Table, with 2 columns and rows. In case it exists, it must return the LINK to view more details of the Weather. Draw an emoji that roughly describes the weather condition, in the Condition column along with the description.' To the right, there are two panels: 'Authentication' with 'Authentication Type' set to 'Api Key', and 'LLM Settings' with 'Provider' set to 'openai', 'Model' set to 'gpt-3.5-turbo-16k', 'Temperature' set to '0.10', and 'Max output Tokens' set to '1024'.

We go to the GeneXus Enterprise AI Backoffice, select the project we are going to work on and in the menu we select Assistants, Create API assistant.

We are asked for a description for the API. Although this is optional, it is useful for the assistant to understand the purpose and functionality of the API we want to interact with. We can include any instructions that we consider relevant to guide the assistant effectively.

For example, for the Weather API, it may be something like the following:

*The final response must strictly comply with the following:*

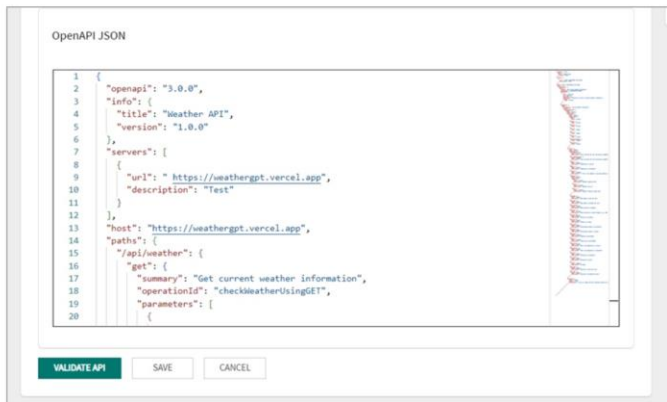
*It must be in the same language as the user's request.*

*The response text must be formatted as a Table, with 2 columns and rows.*

*In case it exists, it must return the LINK to view more details of the Weather.*

*Draw an emoji that roughly describes the weather condition, in the Condition column along with the description.*

## API Assistant



```
{
  "openapi": "3.0.0",
  "info": {
    "title": "Weather API",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "https://weathergpt.vercel.app",
      "description": "Test"
    }
  ],
  "host": "https://weathergpt.vercel.app",
  "paths": {
    "/api/weather": {
      "get": {
        "summary": "Get current weather information",
        "operationId": "checkWeatherUsingGET",
        "parameters": [
          {
            "name": "location",
            "in": "query",
            "required": true,
            "description": "Location for which to retrieve weather information.",
            "schema": {
              "type": "string"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "Current weather information",
            "content": {
              "application/json": {
                "schema": {
                  "type": "object",

```

Ln 1, Col 1 7,330 characters

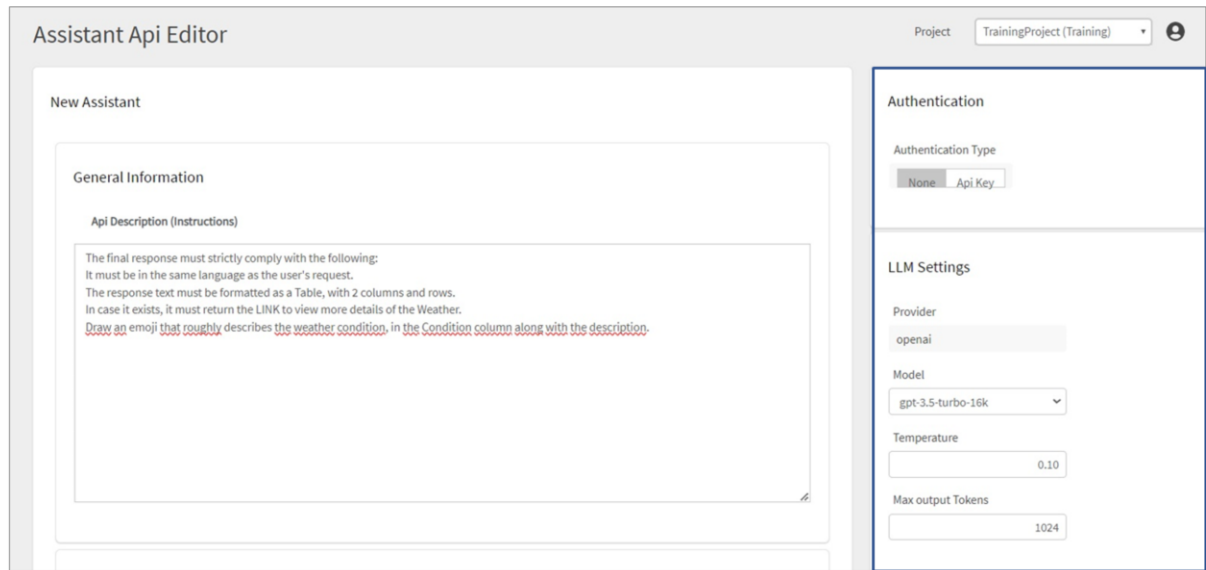
The next step is to complete the OpenAPI JSON file with the OpenAPI 3.0 specification or higher.

This is where the technical details of the API should be specified, including paths, HTTP methods, query parameters, request and response bodies, among others. This file acts as a detailed API specification, allowing for accurate and consistent interaction with the Assistant.

The Validate API option performs a basic check of the JSON to verify if it is correctly formed and if it has any services defined. It also checks if the service URL is correctly defined.

It is worth mentioning that this verification does not run a test against the API.

# API Assistant



The next step is to set up Authentication.

In this step, it is possible to define whether the API requires authentication and, if so, what type of authentication will be used. GeneXus Enterprise AI supports authentication based on ApiKeys, which can be Basic or Bearer.

If the API does not require authentication, the None option must be selected. Otherwise, the API key value must be set.

This unique key provides exclusive access to the API.

In addition, the Type of authentication must be specified, selecting between the Basic or Bearer values.

The next step is to customize the settings related to the language model to adapt it to specific needs and obtain optimal performance.

**Provider:** Determines that the language model provider is OpenAI.

**Model:** The specific language model to be used must be selected. You can choose from a variety of options, from GPT-3.5 to GPT-4, have their own unique features and capabilities.

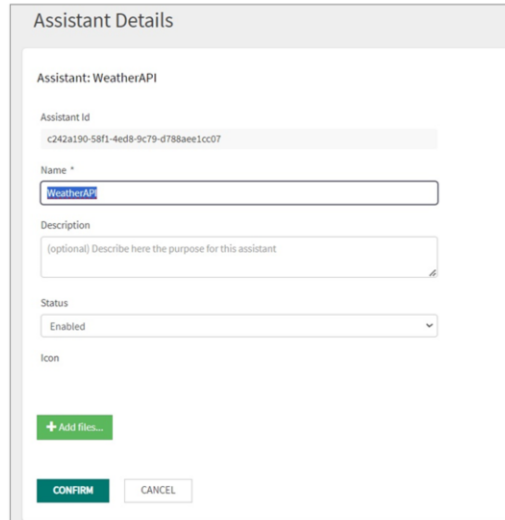
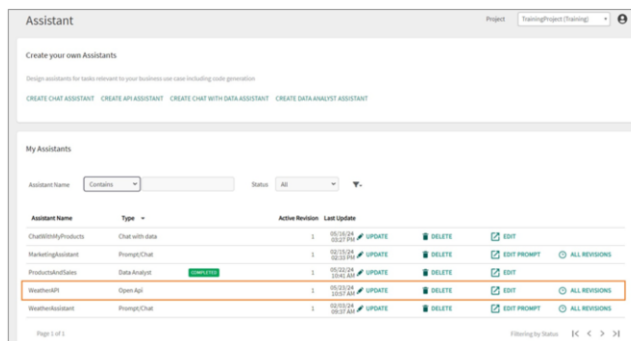


**Temperature** : Adjusts the temperature of the model to control creativity and diversity in the responses generated. Lower values produce more conservative answers and higher values encourage creativity. The default value is 0.10.

**Max Token Output**: Defines the maximum number of tokens allowed in each response. This setting determines the maximum length of the responses generated by the language model. The default value is 1024.

It is important to note that these settings are optional and can be left with their default values.

## API Assistant



We click on Save.

This opens a window to enter the name and description of the API. We can also add a file for the icon.

Once all this information has been completed, we click on Confirm.

We see the assistant created, and if we click on Update, we can see the version identifier with which it was saved, change the name and description, set it as enabled or disabled, and add an icon if we want.

## API Assistant

Assistant

Project: TrainingProject (Training)

Create your own Assistants

Design assistants for tasks relevant to your business use case including code generation

CREATE CHAT ASSISTANT CREATE API ASSISTANT CREATE CHAT WITH DATA ASSISTANT CREATE DATA ANALYST ASSISTANT

My Assistants

Assistant Name:  Status: All

Assistant Name	Type	Active Revision	Last Update			
ChatWithProducts	Chat with data	1	05/23/24 10:27:01	UPDATE	DELETE	EDIT
MarketingAssistant	PromptChat	1	05/23/24 10:29:01	UPDATE	DELETE	EDIT PROMPT ALL REVISIONS
ProductsAndSales	Data Analyst	1	05/22/24 19:42:01	UPDATE	DELETE	EDIT
WeatherAPI	Open Api	1	05/23/24 10:57:01	UPDATE	DELETE	EDIT ALL REVISIONS
WeatherAssistant	PromptChat	1	05/23/24 09:27:01	UPDATE	DELETE	EDIT PROMPT ALL REVISIONS

Page 1 of 1

WeatherAPI Assistant

Project: TrainingProject (Training)

Go to list of Assistants

Revision Id	Revision Name	Provider Name	Model Name	Model Type	Time Stamp		
2	2	openai	gpt-3.5-turbo-16k	Chat	05/23/24 11:00 AM	EDIT/TEST	SET AS ACTIVE
1	1	openai	gpt-3.5-turbo-16k	Chat	05/23/24 10:57 AM	EDIT/TEST	

Page 1 of 1

If Edit is selected instead, it is possible to manage and maintain different versions.

We can overwrite the existing version, or create a new version with the changes made. To do this, we select Save as a new revision.

In this way, we can control and organize the modifications as needed.

From here, we can see all the versions of the Assistant.

## API Assistant

compare please the weather in Montevideo and Buenos Aires

Here is a comparison of the weather between Montevideo and Buenos Aires:

Location	Condition	Description
Montevideo	 Moderate rain	Moderate rain
Buenos Aires	 Partly cloudy	Partly cloudy

You can find more details about the weather in Montevideo [here](#) and the weather in Buenos Aires [here](#).

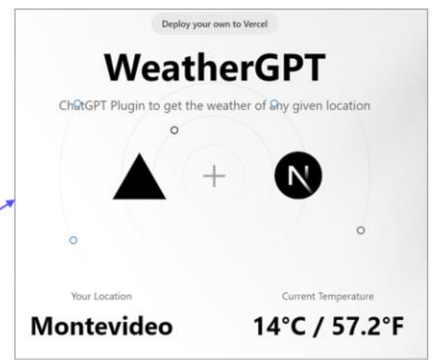
Regenerate response

Type a message

Deploy your own to Vercel

### WeatherGPT

ChatGPT Plugin to get the weather of any given location



Your Location: **Montevideo**

Current Temperature: **14°C / 57.2°F**

The interface features a circular graphic with a triangle, a plus sign, and a circle with an 'N'.

Deploy your own to Vercel

### WeatherGPT

ChatGPT Plugin to get the weather of any given location



Your Location: **Buenos Aires**

Current Temperature: **13°C / 55.4°F**

The interface features a circular graphic with a triangle, a plus sign, and a circle with an 'N'.

Finally, we test the new assistant from the Playground option of the menu.

We check, for example, the current weather conditions in Venice.

And we want to compare the current weather conditions in Montevideo and Buenos Aires.

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

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