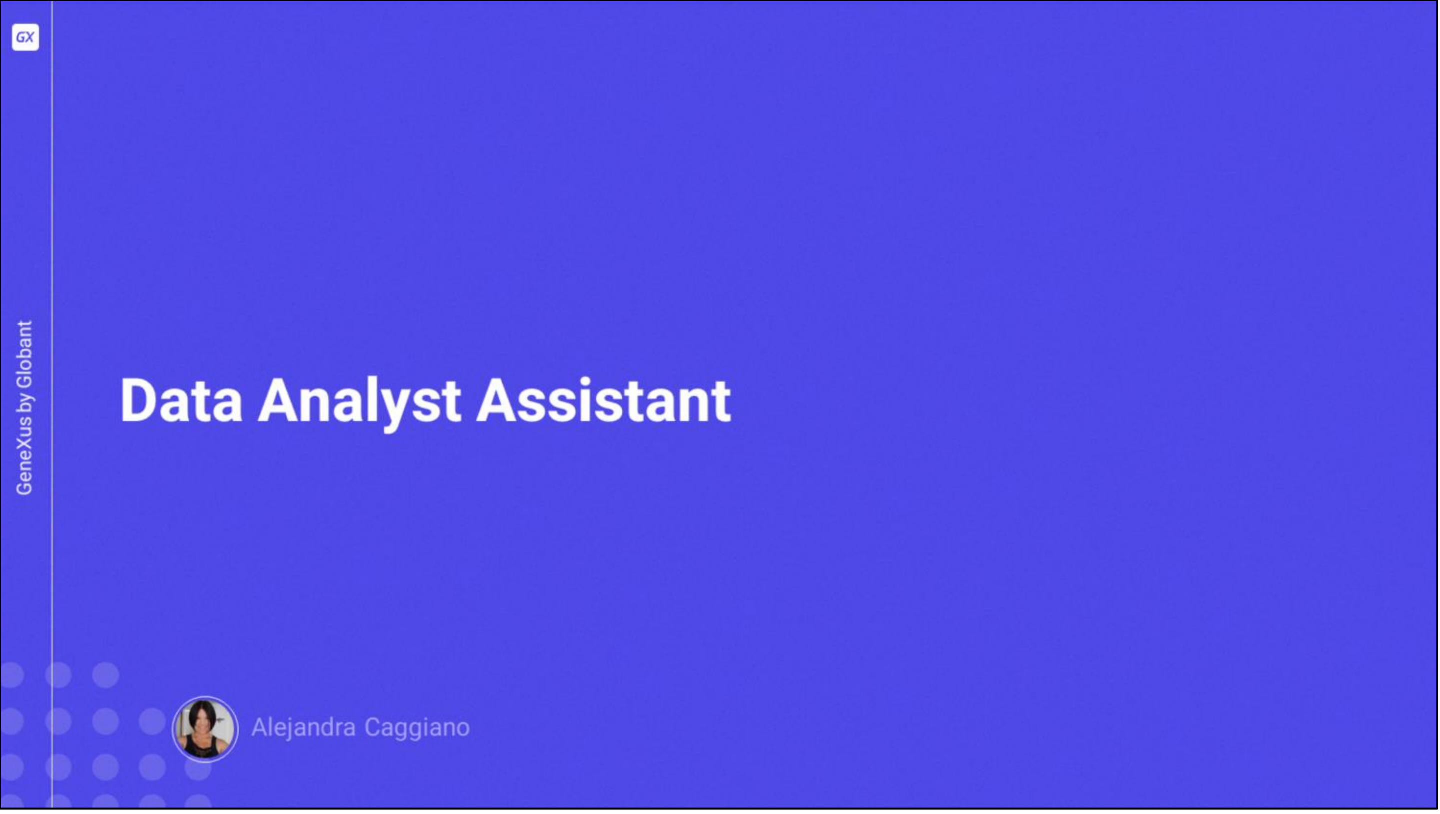
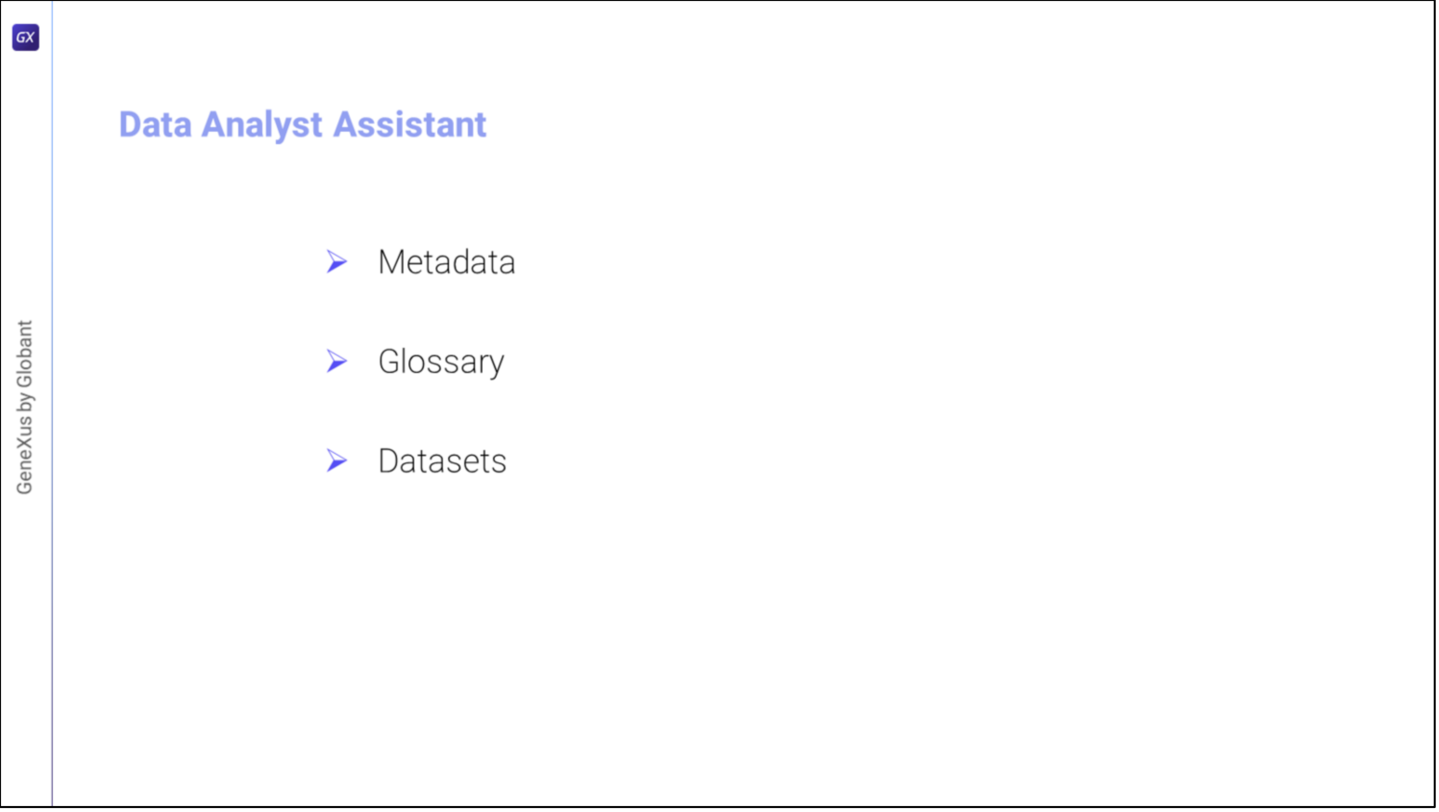


GeneXus by Globant

Data Analyst Assistant

Alejandra Caggiano

The Data Analyst Assistant is a GeneXus Enterprise AI assistant that uses artificial intelligence to analyze data and generate useful lists, summaries, and descriptions. It provides specific answers in short timeframes to support decision-making.



Data Analyst Assistant

- Metadata
- Glossary
- Datasets

To define an assistant of this type, the following information must be available:

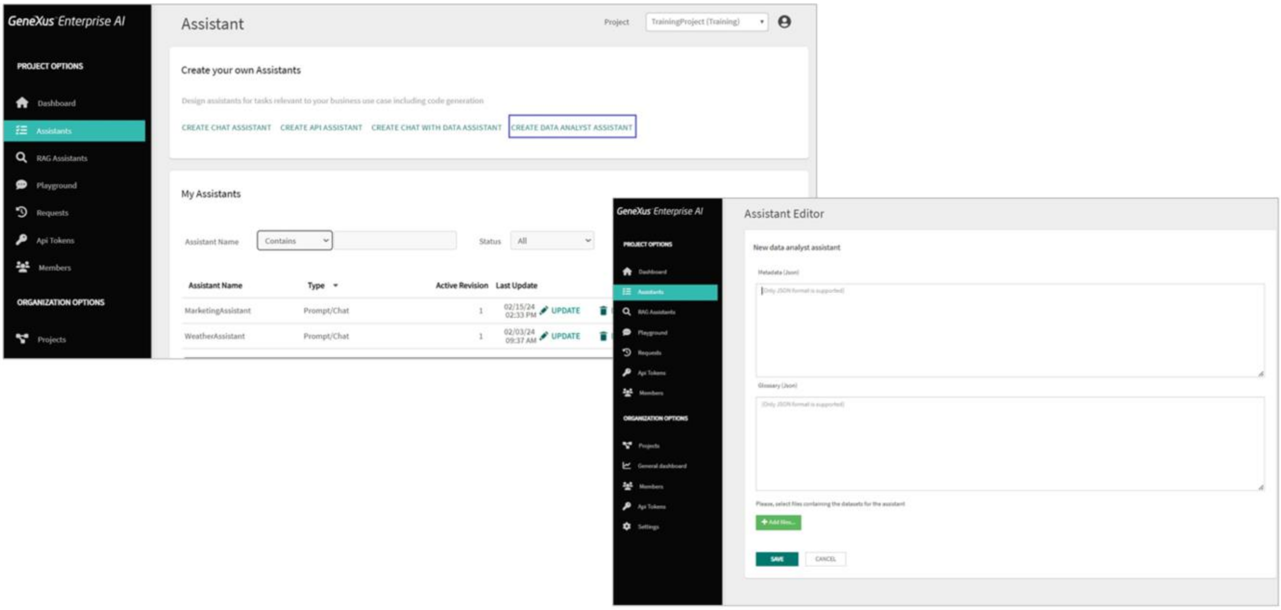
● **Metadata**: This includes a description of each data set, its columns (including possible data types and values), and considerations for the encoder and interpreter.

● The **Glossary** should be indicated: It means a list of terms used in the company and/or the user's domain that would be relevant for the LLM to understand the questions.

● And also the **Dataset**, which is a set of CSV-formatted files that follow the descriptions added as metadata.

All this information is then used to communicate with the API, enabling end users to interact with it through GeneXus Enterprise AI.

Data Analyst Assistant



Assistant

Project: TrainingProject (Training)

Create your own Assistants

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

My Assistants

Assistant Name: Contains

Status: All

Assistant Name	Type	Active Revision	Last Update
MarketingAssistant	Prompt/Chat	1	02/15/24 02:33 PM
WeatherAssistant	Prompt/Chat	1	02/03/24 09:37 AM

Assistant Editor

New data analyst assistant

Metadata (Json)

[Only JSON format is supported]

Glossary (Json)

[Only JSON format is supported]

Please, select files containing the datasets for the assistant

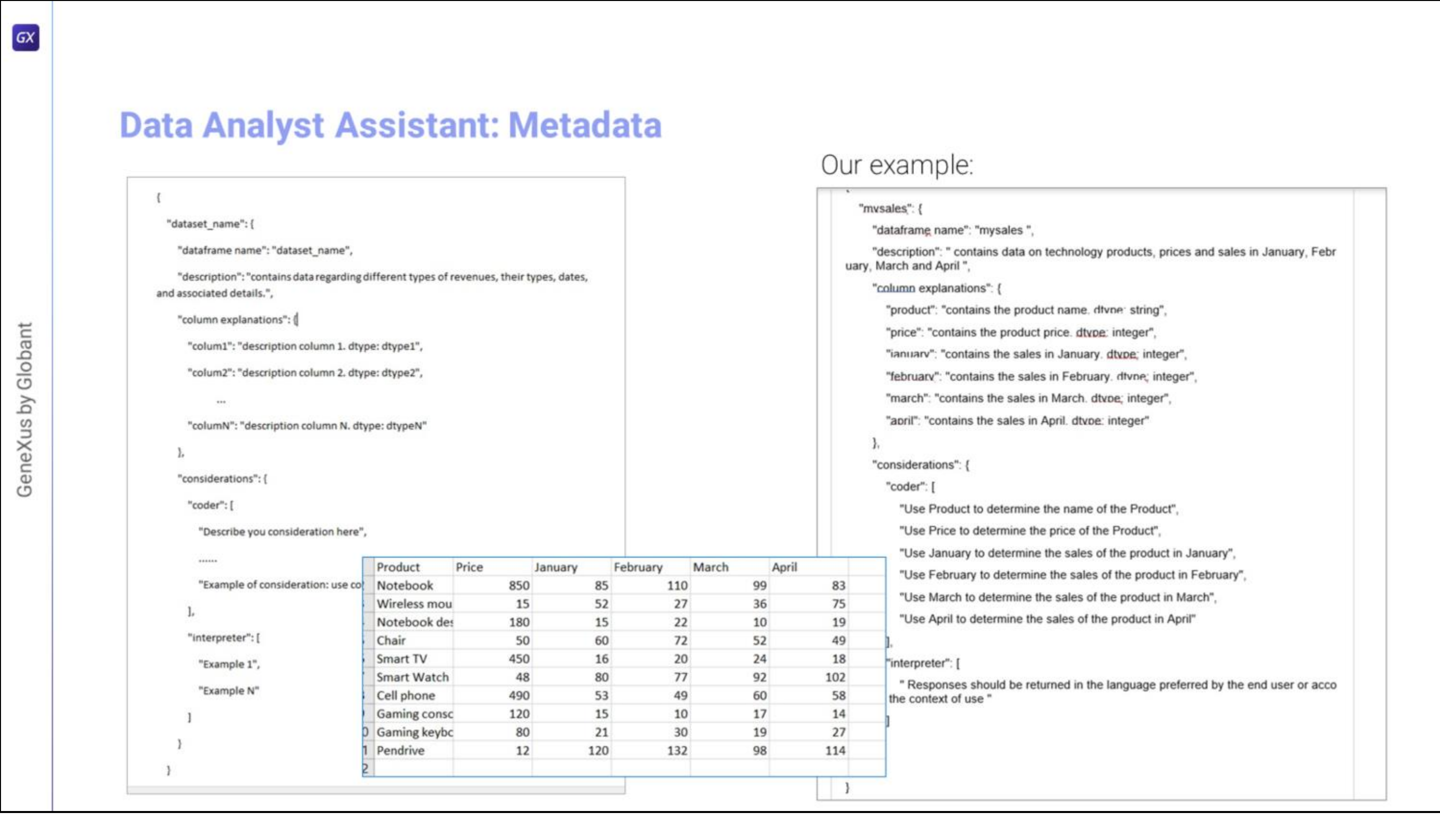
Let's see an example of how to create this type of assistant.

First, we go to the GeneXus Enterprise AI Backoffice

We select the project we will work on, and in the Assistants option of the menu, we choose Create Data Analyst Assistant.

Next, we must define the metadata.

This is a JSON with descriptive information about the information contained in a dataset. It is used to provide context and better understand the available data.



GX

GeneXus by Globant

Data Analyst Assistant: Metadata

```
{
  "dataset_name": {
    "dataframe name": "dataset_name",
    "description": "contains data regarding different types of revenues, their types, dates,
and associated details.",
    "column explanations": {
      "colum1": "description column 1. dtype: dtype1",
      "colum2": "description column 2. dtype: dtype2",
            ...
      "columN": "description column N. dtype: dtypeN"
    },
    "considerations": {
      "coder": [
        "Describe you consideration here",
        ......
        "Example of consideration: use co
      ],
      "interpreter": [
        "Example 1",
        "Example N"
      ]
    }
  }
```

Product	Price	January	February	March	April
Notebook	850	85	110	99	83
Wireless mou	15	52	27	36	75
Notebook des	180	15	22	10	19
Chair	50	60	72	52	49
Smart TV	450	16	20	24	18
Smart Watch	48	80	77	92	102
Cell phone	490	53	49	60	58
Gaming consc	120	15	10	17	14
Gaming keybc	80	21	30	19	27
Pendrive	12	120	132	98	114

Our example:

```
  "mvsales": {
    "dataframe name": "mysales ",
    "description": " contains data on technology products, prices and sales in January, Febr
uary, March and April ",
    "column explanations": {
      "product": "contains the product name. dtvne: string",
      "price": "contains the product price. dtvpe: integer",
      "ianuarv": "contains the sales in January. dtvpe; integer",
      "februarv": "contains the sales in February. dtvne; integer",
      "march": "contains the sales in March. dtvpe; integer",
      "april": "contains the sales in April. dtvpe: integer"
    },
    "considerations": {
      "coder": [
        "Use Product to determine the name of the Product",
        "Use Price to determine the price of the Product",
        "Use January to determine the sales of the product in January",
        "Use February to determine the sales of the product in February",
        "Use March to determine the sales of the product in March",
        "Use April to determine the sales of the product in April"
      ],
      "interpreter": [
        " Responses should be returned in the language preferred by the end user or acco
the context of use "
      ]
  }
```

This **metadata** includes details such as the name of the dataset, a description of its contents and the structure of its columns (including data types).

To complete this metadata correctly, a JSON must be defined with the following structure:

Where:

The "dataframe_name" must contain the name of the dataset and correspond to the name of one of the .csv files to be loaded.

The "description" must provide a description of the dataset, including the data types of its columns and possible values.

The "column explanations" must include a description of each column in the dataset, specifying its purpose and the type of data it contains.

It is important to consider that "colum1" to "columN": correspond to the exact names (without blanks or special characters) of the columns in the .csv files to be loaded.

The number of columns in the .csv files must match the number of columns provided in this metadata. In addition, it is important to avoid duplication of column names.

As for the “considerations”, they should describe specific considerations for the encoder and the interpreter.

The “encoder” generates the code necessary to process the data. In addition, it uses the metadata information to guide the process of extracting relevant information from the data.

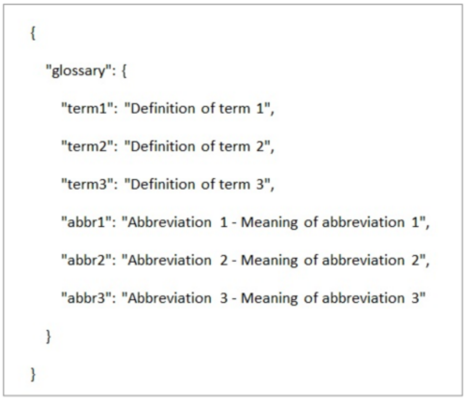
As for the “interpreter”, its role is to generate the final response with the information obtained from the data processing performed by the encoder. It analyzes the results to better understand the data and provide relevant answers.

It may include considerations for understanding the technical terms and abbreviations used in the data, as well as specifying that the answers must be provided in the language selected by the end user or according to the context.

In this example, we have a very simple dataset, where technology products are recorded, including the respective number of units sold in January to April.

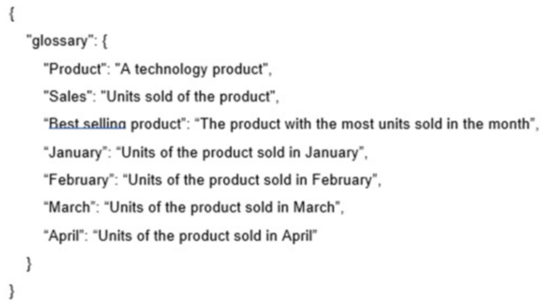
We load the metadata taking into account everything we mentioned above.

Data Analyst Assistant: Glossary



```
{
  "glossary": {
    "term1": "Definition of term 1",
    "term2": "Definition of term 2",
    "term3": "Definition of term 3",
    "abbr1": "Abbreviation  1 - Meaning of abbreviation 1",
    "abbr2": "Abbreviation  2 - Meaning of abbreviation 2",
    "abbr3": "Abbreviation  3 - Meaning of abbreviation 3"
  }
}
```

Our example:



```
{
  "glossary": {
    "Product": "A technology product",
    "Sales": "Units sold of the product",
    “Best selling product": “The product with the most units sold in the month",
    “January": “Units of the product sold in January",
    “February": “Units of the product sold in February",
    “March": “Units of the product sold in March",
    “April": “Units of the product sold in April"
  }
}
```

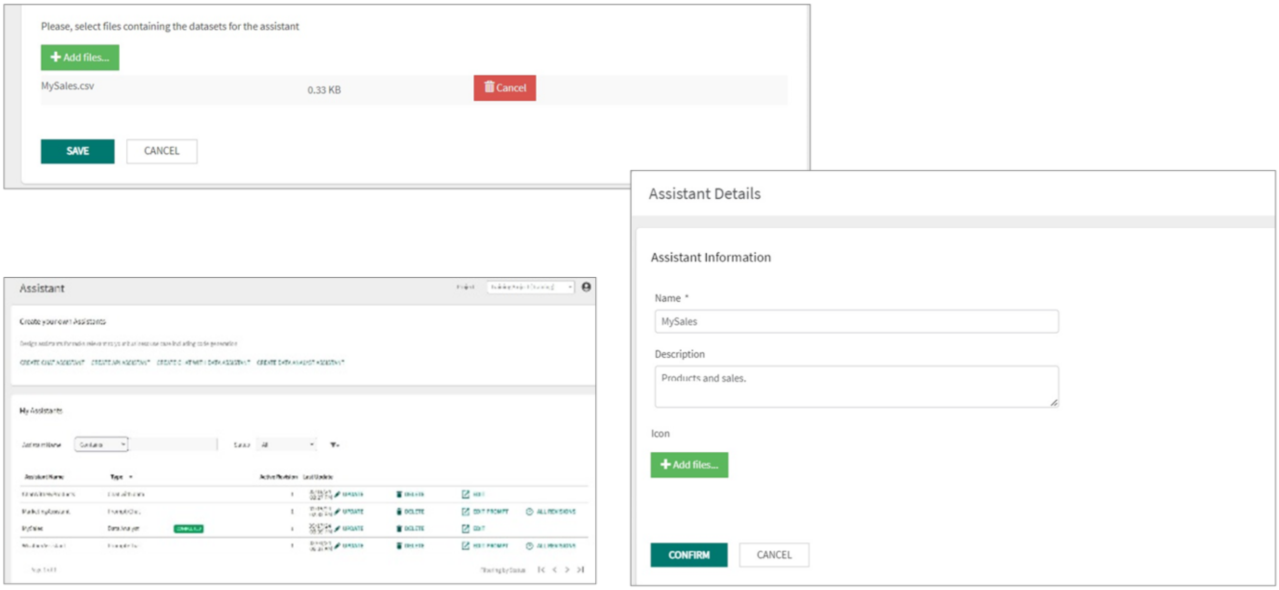
Good. The next step is to define the **glossary**.

This is also a JSON containing a list of terms used in the end user's company or domain, along with their corresponding definitions. These terms and definitions are relevant for the LLM to understand the questions and provide more accurate answers.

For example, it may include abbreviations that are frequently used in the company and their meanings for better understanding in a format like the following:

We load the glossary associated with our example.

Data Analyst Assistant: Creation



Please, select files containing the datasets for the assistant

MySales.csv 0.33 KB

Assistant Details

Assistant Information

Name *: MySales

Description: Products and sales.

Icon

OK. At this point, the next step is to **add files**.

For that, we must load a file from the dataset by selecting Add files...

It is important to make sure that the CSV file follows the descriptions and metadata added earlier.

We load our file.

Finally, to create the assistant, we select Save.

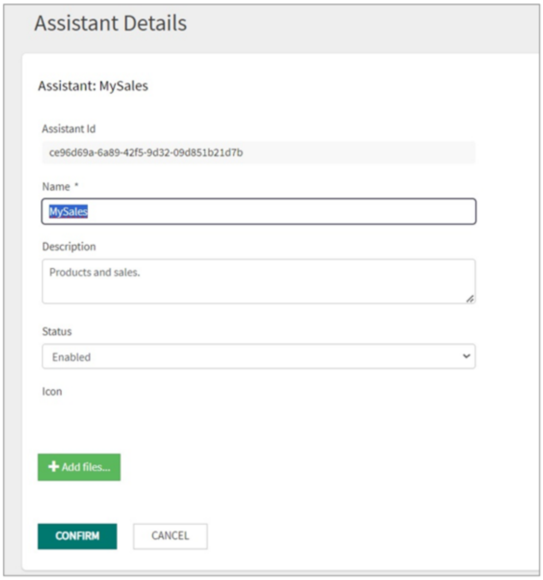
A window is displayed to enter a Name, Description and Icon if desired.

We click on Confirm... ...and go back to the page showing the current status of the assistant.

This status is related to the loading process of the datasets with the metadata and glossary information.

The possible statuses are FAILED, COMPLETED, or PROCESSING (indicating the progress percentage).

Data Analyst Assistant: Update



Assistant Details

Assistant: MySales

Assistant Id

ce96d69a-6a89-42f5-9d32-09d851b21d7b

Name *

MySales

Description

Products and sales.

Status

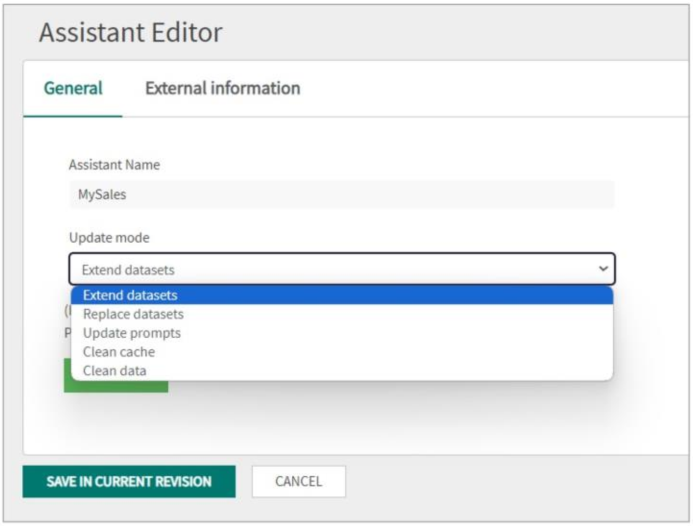
Enabled

Icon

Once the loading status is completed, we can select Update and see the version identifier with which it was saved, change the name and description, set it as enabled or disabled, or add an icon.

GeneXus by Globant

Data Analyst Assistant: Edition



Assistant Editor

Assistant Name: MySales

Update mode: Extend datasets

It is also possible to edit it, to extend or replace the dataset, or to update the prompt.

Expanding the dataset expands the information available without affecting the original structure of the dataset.

The files selected for integration must meet the same initial conditions, such as having the same number of columns and file names.

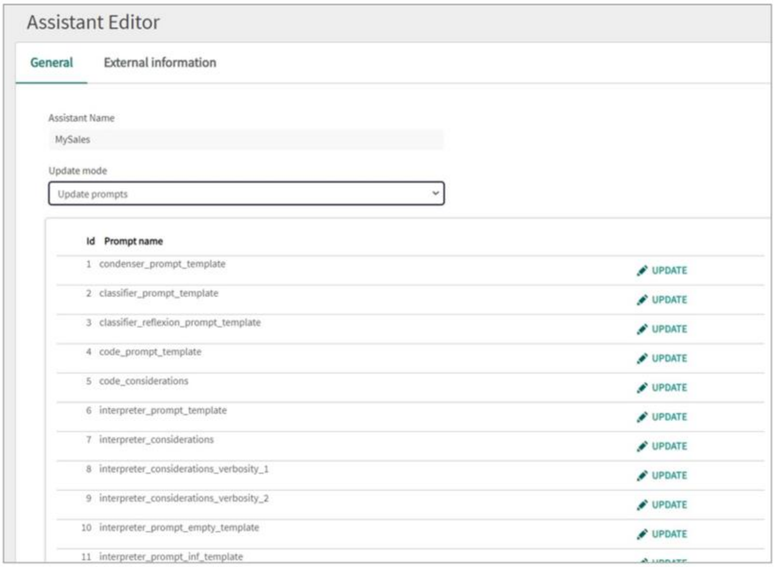
The files used for the extension must not contain the same rows that are already loaded in the assistant, as this could generate duplicate keys and cause problems when running the queries.

It should also be noted that **when increasing the dataset it is not possible to make changes to the metadata or the glossary.** This function focuses only on increasing existing datasets without affecting other aspects of the assistant.

Replacing the dataset, on the other hand, allows you to completely replace the data in the existing dataset, as well as modify the metadata and associated glossary.

This is useful when you need to update the information completely or make significant changes to the structure and description of the dataset.

Data Analyst Assistant: Update prompt



Assistant Editor

General | External information

Assistant Name: MySales

Update mode: Update prompts

Id	Prompt name	
1	condenser_prompt_template	UPDATE
2	classifier_prompt_template	UPDATE
3	classifier_reflexion_prompt_template	UPDATE
4	code_prompt_template	UPDATE
5	code_considerations	UPDATE
6	interpreter_prompt_template	UPDATE
7	interpreter_considerations	UPDATE
8	interpreter_considerations_verbosity_1	UPDATE
9	interpreter_considerations_verbosity_2	UPDATE
10	interpreter_prompt_empty_template	UPDATE
11	interpreter_prompt_inf_template	

If Update prompt is selected, a list of the 28 default prompts will be displayed.

It is possible to query and update each of the prompts individually by clicking on Update.

This requires the Administrator, ProjectRole or OrganizationRole role.

It is important to note that these updates are made from the list of prompts and not at the general level of the assistant.

When generating a new version of the assistant and making changes with the Extend Datasets or Replace Datasets options, changes made to the prompts in previous versions of the assistant will be ignored.

As usual, we can finally test the created assistant using the Playground option in the Backoffice menu.

GX

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com