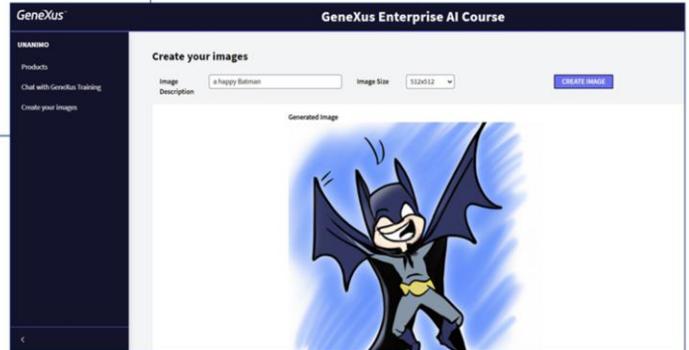


Aplicativo GX que interage com IA para gerar imagens



Alejandra Caggiano

Aplicativo GX que interage com IA para gerar imagens



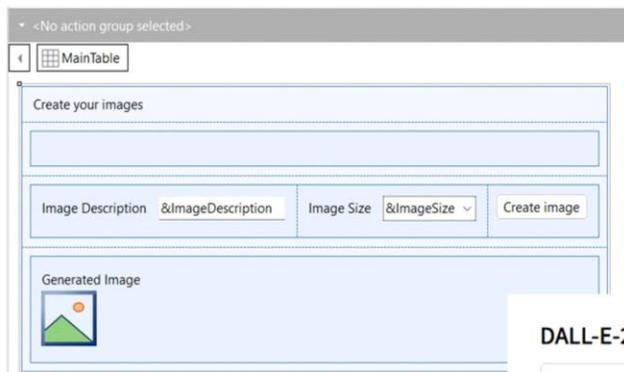
Já temos nossa aplicação GeneXus que interage com um Chat assistant e com um RAG assistant. Vamos habilitar agora a possibilidade de gerar imagens.

Aplicativo GX que interage com IA para gerar imagens

- GeneXus Enterprise AI: Proxy API
- DALL-E-2

Para isso vamos utilizar a Proxy API para interagir com um sistema de inteligência artificial que nos permita gerar imagens a partir de uma descrição em linguagem natural.

Aplicativo GX que interage com IA para gerar imagens



DALL-E-2 - cURL

```
curl --location 'https://api.saia.ai/proxy/openai/v1/images/generations' \
-H 'Content-Type: application/json' \
-H 'Authorization: $SAIA_API_TOKEN' \
-d '{
  "model": "dall-e-2",
  "prompt": "a halloween pumpkin",
  "size": "1024x1024"
}'
```

Vamos então ao GeneXus, à nossa base de conhecimento.

Já criamos um web panel chamado CreatelImages, onde permitimos ao usuário final inserir uma breve descrição e selecionar o tamanho da imagem a ser gerada.

A funcionalidade então que queremos resolver é o código do evento associado ao botão Create Image. E para isso devemos criar um procedimento que receba a descrição e tamanho da imagem como parâmetros de entrada e retorne a url da imagem gerada. A partir dessa url mostraremos a imagem na variável &GeneratedImage presente no form do web panel.

Vamos para Tools / Application Integration / cURL Inspector.

Colocamos como nome GeneratelImages e carregamos o cURL sample, que neste caso corresponde à interação da Proxy API com DALL-E-2

Aplicativo GX que interage com IA para gerar imagens

```
Layout Rules * Conditions Variables Help Documentation
1 Parm(in: &ImageDescription, in: &ImageSize, out:&ImageURL);
2
3
4 //curl --location 'https://api.saia.ai/proxy/openai/v1/images/generations' -H 'Content-Type: application/json'
5 &HttpClient.Secure = 1
6 &HttpClient.Host = "api.qa.saia.ai"
7 &HttpClient.AddHeader("Content-Type", "application/json")
8 &HttpClient.AddHeader("Authorization", "Bearer default_OuK5BwzqSLjNEHwUf-GV0QCLIZYHYxBBGAsAg1CiLSa91F")
9 &HttpClient.AddString("{\"model\":\"dall-e-2\", \"prompt\": \"' + &ImageDescription.Trim() + '\" , \"size\": \"' + &ImageSize + '\""});
10
11 &HttpClient.Execute("POST", !"/proxy/openai/v1/images/generations")
12
```

Como já sabemos, isso gera o procedimento base onde completaremos a definição da conexão com os dados necessários, de acordo com o nosso contexto.

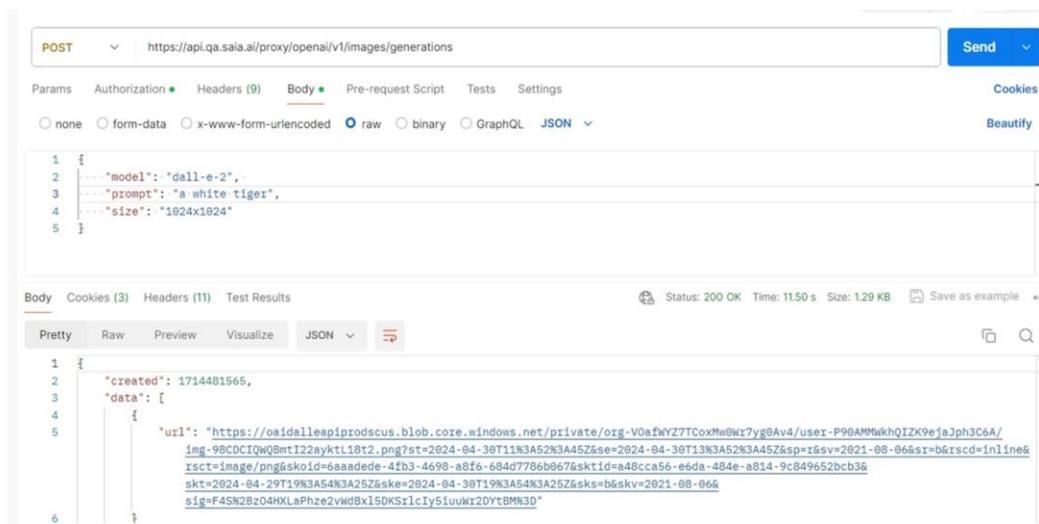
Em primeiro lugar, declaramos a regra Parm para que receba a descrição e tamanho da imagem, e retorne a url da imagem gerada.

Vamos analisar o source:

Definimos o protocolo de conexão HTTPS, definimos o host, o project api token e definimos o corpo da solicitação de forma que considere os parâmetros de entrada que correspondem à descrição e tamanho da imagem requerida pelo usuário final.

Então definimos o POST para sua execução.

Aplicativo GX que interage com IA para gerar imagens



The screenshot displays a Postman interface for a POST request to the OpenAI API endpoint: `https://api.qa.saia.ai/proxy/openai/v1/images/generations`. The request body is a JSON object with the following content:

```
1 {
2   ... "model": "dall-e-2",
3   ... "prompt": "a white tiger",
4   ... "size": "1024x1024"
5 }
```

The response status is 200 OK, with a time of 11.50 s and a size of 1.29 KB. The response body is a JSON object with the following content:

```
1 {
2   "created": 1714481565,
3   "data": [
4     {
5       "url": "https://oaidalleapiprodscus.blob.core.windows.net/private/org-V0afWY27TCoxMw0W17yg@Av4/user-P96AMWkhQIZK9ejaJph3C6A/img-98CDIQ0BmtI22aykt118t2.png?st=2024-04-30T19%3A52%3A45Z&se=2024-04-30T19%3A52%3A45Z&sp=r&sv=2021-08-06&sr=b&rscd=inline&rsct=image/png&skoid=6aaadede-4fb3-4698-a8f6-684d7786b067&sktid=a48cca56-e6da-484e-a814-9c849652bcb3&skt=2024-04-29T19%3A54%3A25Z&skne=2024-04-30T19%3A54%3A25Z&skns=b6skv=2021-08-06&sig=F45N2Bz04HXLaPhze2vhd8x15DKSr1cIy5Iuuwr2DyTBMN3D"
6     }
7   ]
8 }
```

O que acontece com a resposta recebida?

Repetimos o mesmo processo que realizamos nos exemplos anteriores. A partir do Postman salvamos a resposta da solicitação e importamos o arquivo para nossa base de conhecimento, utilizando a opção Tools / Application integration / Json import.

Aplicativo GX que interage com IA para gerar imagens

Name	Type
MyImages	
• created	Numeric(10.5)
data	
dataItem	
• url	VarChar(100)

```
1 //curl --location 'https://api.saia.ai/proxy/openai/v1/images/generations' -H 'Content-Type: application
2
3 &HttpClient.Secure = 1
4 &HttpClient.Host = "api.qa.saia.ai"
5
6 &HttpClient.AddHeader("Content-Type", "application/json")
7 &HttpClient.AddHeader("Authorization", "Bearer default_OuK5BwzqSLjNEHwUf-GV0QCLi2YHYxBBGAg1CilSa9lF
8
9 &HttpClient.AddString(!{"model":"dall-e-2", "prompt": " " + &ImageDescription.Trim() + " " , "size": " " +
10
11 &HttpClient.Execute("POST", !"/proxy/openai/v1/images/generations")
12
13
14 &MyImages.FromJson(&HttpClient.ToString())
15
16 &ImageUrl = &MyImages.data.Item(1).url
17
```

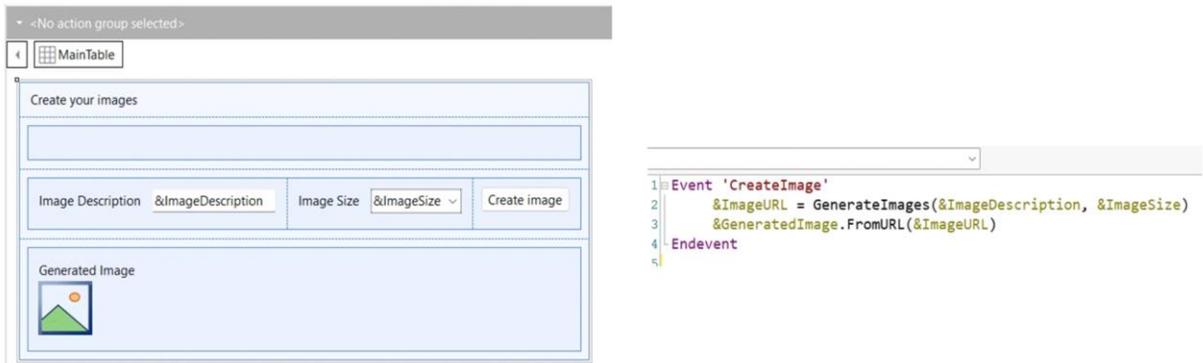
Colocamos como nome MyImages e isso gera o seguinte tipo de dado estruturado:

onde o elemento url do primeiro item da coleção "data" corresponderá à url da imagem gerada e será o parâmetro de saída de nosso procedimento.

Então definimos a variável &MyImages, baseada no sdt, e a carregamos com a resposta da solicitação aplicando o método FromJson, como já vimos nos exemplos anteriores.

Finalmente carregamos a variável &ImageUrl com o valor do elemento "url" do primeiro item da coleção "data".

Aplicativo GX que interage com IA para gerar imagens



The screenshot shows a GX application window titled "MainTable" with a "Create your images" section. The interface includes a text input for "Image Description" (labeled &ImageDescription), a dropdown menu for "Image Size" (labeled &ImageSize), and a "Create image" button. Below the input fields is a "Generated Image" section displaying a small image of a landscape with a sun and mountains. To the right, the code for the "CreateImage" event is shown:

```
1 Event 'CreateImage'  
2     &ImageURL = GenerateImages(&ImageDescription, &ImageSize)  
3     &GeneratedImage.FromURL(&ImageURL)  
4 Endevent  
5
```

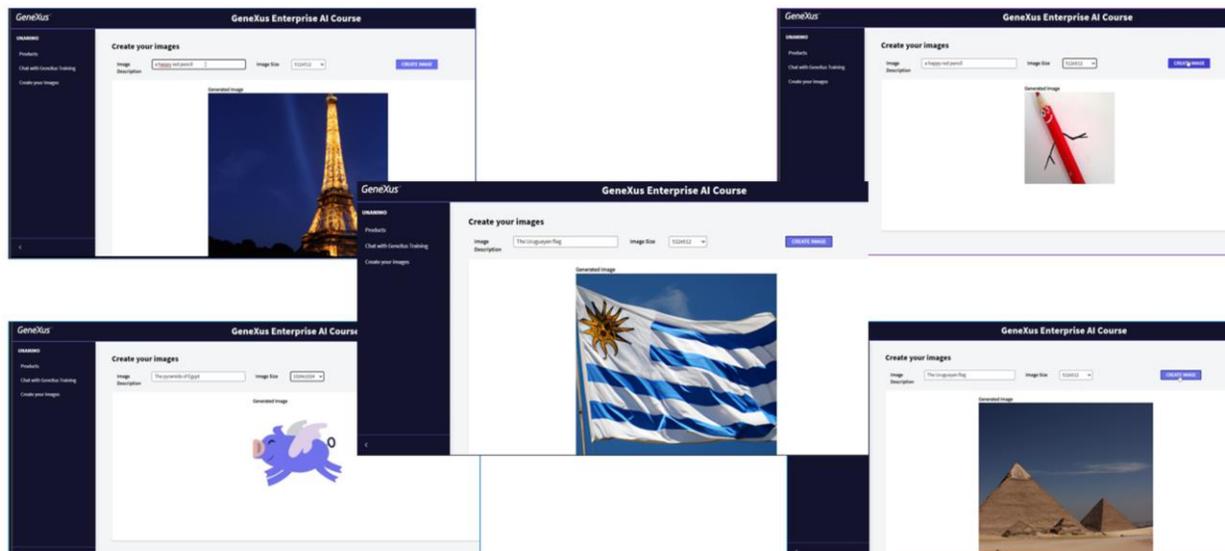
Uma vez definido este procedimento, devemos chamá-lo a partir do evento associado ao botão do web panel.

Então fazemos a chamada e recebemos sua saída na variável &ImageURL.

Por último, a variável &GeneratedImage, presente no form, carregamos com a imagem obtida da url recebida.

Para testá-lo, pressionamos F5.

Aplicativo GX que interage com IA para gerar imagens



Em primeiro lugar vamos pedir uma imagem da Torre Eiffel à noite, e escolhemos como tamanho 512x512

Se a imagem que recebemos não nos agrada, pedimos outra.

Vejamos outros exemplos mais criativos, como um lápis vermelho sorrindo... Ou um porquinho azul voando, de tamanho 256x256... As pirâmides do Egito... e a bandeira Uruguaia

GeneXusTM
by **Globant**

training.genexus.com