

6.6 - Generate images with Azure OpenAI Service

- [Overview](#)
 - [Introduction](#)
 - [What is DALL-E?](#)
 - [Explore DALL-E in Azure AI Studio](#)
 - [Use the Azure OpenAI REST API to consume DALL-E models](#)
 - [Exercise - Generate images with a DALL-E model](#)
 - [Provision an Azure OpenAI resource](#)
 - [Deploy a model](#)
 - [Use the REST API to generate images](#)
 - [Prepare to develop an app in Visual Studio Code](#)
 - [Configure your application](#)
 - [View application code](#)
 - [Run the app](#)
 - [Knowledge Check](#)
 - [Summary](#)
-

Overview

The Azure OpenAI service includes the DALL-E model, which you can use to generate original images based on natural language prompts.

By the end of this module, you'll be able to:

- Describe the capabilities of DALL-E in the Azure OpenAI service
 - Use the DALL-E playground in Azure AI Studio
 - Use the Azure OpenAI REST interface to integrate DALL-E image generation into your apps
-

Introduction

The Azure OpenAI service enables you to use language models to generate content based on natural language prompts. One of these models is the DALL-E image generation model, which is capable of creating original graphical content based on natural language descriptions of a desired image.

The ability to use AI to generate graphics has a number of applications; including the creation of illustrations or photorealistic images for articles or marketing collateral, generation of unique product or company logos, or any scenario where a desired image can be described.

In this module, you'll learn how to use the DALL-E model in an instance of the Azure OpenAI service, and retrieve the images it generates.

What is DALL-E?

DALL-E is a neural network based model that can generate graphical data from natural language input. Put more simply, you can provide DALL-E with a description and it can generate an appropriate image.

For example, you might submit the following natural language prompt to DALL-E:

A squirrel on a motorcycle

This prompt could result in the generation of graphical output such as the following image:



The images generated by DALL-E are original; they are not retrieved from a curated image catalog. In other words, DALL-E is not a search system for *finding* appropriate images - it is an artificial intelligence (AI) model that *generates* new images based on the data on which it was trained.

Explore DALL-E in Azure AI Studio

To experiment with DALL-E, you can provision an Azure OpenAI Service resource in an Azure subscription and use the *Images playground* to submit prompts and view the resulting generated images.

The screenshot shows the Azure OpenAI service playground interface. On the left, there's a sidebar with navigation links like Overview, Get started, Model catalog, Model benchmarks, Prompt catalog, AI Services, Playgrounds (Chat, Assistants, Images), Tools (Fine-tuning, Batch jobs), Shared resources (Deployments, Quota, Content filters, Data files), and a preview section. The main area is titled "Images playground". It has tabs for Deployments (set to Dalle3) and View Code. A search bar and tile size selector are also present. A "Prompt" input field contains the text "A 3D digital render of squirrel on a motorcycle wearing a helmet". Below it, two generated images are displayed side-by-side. The first image is a 3D digital render of a squirrel wearing a black motorcycle helmet and riding a sleek black motorcycle. The second image is a more realistic, colorful photograph of a squirrel sitting on a dark motorcycle. Both images have their respective captions below them: "A 3D digital render of squirrel on a motorcycle wearing a helmet" and "A squirrel on a motorcycle". At the bottom of each image card are five small blue icons for file operations.

When using the playground, you can adjust the **settings** to specify:

- The resolution (size) of the generated images. Available sizes are `1024x1024` (which is the default value), `1792x1024`, or `1024x1792`.
- The image style to be generated (such as `vivid` or `natural`).
- The image quality (choose from `standard` or `hd`).

Use the Azure OpenAI REST API to consume DALL-E models

You can use the Azure OpenAI service REST API to consume DALL-E models from applications.

To make a REST call to the service, you need the endpoint and authorization key for the Azure OpenAI Service resource you have provisioned in Azure.

You initiate the image generation process by submitting a POST request to the service endpoint with the authorization key in the header. The request must contain the following parameters in a JSON body:

- **prompt**: The description of the image to be generated.
- **n**: The number of images to be generated. DALL-E 3 only supports n=1.
- **size**: The resolution of the image(s) to be generated (`1024x1024`, `1792x1024`, or `1024x1792` for DALL-E 3; `256x256`, `512x512`, or `1024x1024` for DALL-E 2).
- **quality Optional**: The quality of the image (`standard` or `hd`). Defaults to `standard`.
- **style Optional**: The visual style of the image (`natural` or `vivid`). Defaults to `vivid`.

For example, the following JSON could be used to generate an 512 x 512 image of a badger wearing a tuxedo:

```
{
  "prompt": "A badger wearing a tuxedo",
```

```
"n": 1,  
"size": "512x512",  
"quality": "hd",  
"style": "vivid"  
}
```

If you're using an older generation model such as DALL-E 2, the initial request does not immediately return the results of the image generation process. Instead, the response includes an **operation-location** header with a URL for a callback service that your application code can poll until the results of image generation are ready.

With DALL-E 3, the result from the request is processed synchronously with the response containing the URL for the generated image. The response is similar to the following JSON:

```
{  
  "created": 1686780744,  
  "data": [  
    {  
      "url": "<URL of generated image>",  
      "revised_prompt": "<prompt that was used>"  
    }  
  ]  
}
```

The **data** element includes the **url** value, which references a PNG image file generated from the prompt that you can then view or download.

The response also contains a **revised prompt** that was used to generate the image, which was updated by the system to achieve the most desirable results. In this example, the image might look similar to the following image:



Exercise - Generate images with a DALL-E model

The Azure OpenAI Service includes an image-generation model named DALL-E. You can use this model to submit natural language prompts that describe a desired image, and the model will generate an original image based on the description you provide.

In this exercise, you'll use a DALL-E version 3 model to generate images based on natural language prompts.

This exercise will take approximately **25** minutes.

Provision an Azure OpenAI resource

Before you can use Azure OpenAI to generate images, you must provision an Azure OpenAI resource in your Azure subscription. The resource must be in a region where DALL-E models are supported.

1. Sign into the **Azure portal** at <https://portal.azure.com>.
2. Create an **Azure OpenAI** resource with the following settings:
 - **Subscription:** Select an Azure subscription that has been approved for access to the Azure OpenAI service, including DALL-E
 - **Resource group:** Choose or create a resource group
 - **Region:** Choose either **East US**, **Australia East**, or **Sweden Central***
 - **Name:** A unique name of your choice
 - **Pricing tier:** Standard S0
3. Wait for deployment to complete. Then go to the deployed Azure OpenAI resource in the Azure portal.

- DALL-E 3 models are only available in Azure OpenAI service resources in the **East US**, **Australia East**, and **Sweden Central** regions.

Deploy a model

Next, you will create a deployment of the **dalle3** model from the CLI. In the Azure portal; select **Cloud Shell** icon from the top menu bar and ensure that your terminal is set to **Bash**. Refer to this example and replace the following variables with your own values:

```
az cognitiveservices account deployment create \
    -g *your resource group* \
    -n *your Open AI resource* \
    --deployment-name dall-e-3 \
    --model-name dall-e-3 \
    --model-version 3.0 \
    --model-format OpenAI \
    --sku-name "Standard" \
    --sku-capacity 1
```

> /* Sku-capacity is measured in thousands of tokens per minute. A rate limit of 1,000 tokens per minute is more than adequate to complete this exercise while leaving capacity for other people using the same subscription.

Use the REST API to generate images

The Azure OpenAI service provides a REST API that you can use to submit prompts for content generation - including images generated by a DALL-E model.

Prepare to develop an app in Visual Studio Code

Now let's explore how you could build a custom app that uses Azure OpenAI service to generate images. You'll develop your app using Visual Studio Code. The code files for your app have been provided in a GitHub repo.

Tip: If you have already cloned the **mslearn-openai** repo, open it in Visual Studio code. Otherwise, follow these steps to clone it to your development environment.

1. Start Visual Studio Code.
2. Open the palette (SHIFT+CTRL+P) and run a **Git: Clone** command to clone the `https://github.com/MicrosoftLearning/mslearn-openai` repository to a local folder (it doesn't matter which folder).
3. When the repository has been cloned, open the folder in Visual Studio Code.

Note: If Visual Studio Code shows you a pop-up message to prompt you to trust the code you are opening, click on **Yes, I trust the authors** option in the pop-up.

4. Wait while additional files are installed to support the C# code projects in the repo.

Note: If you are prompted to add required assets to build and debug, select **Not Now**.

Configure your application

Applications for both C# and Python have been provided. Both apps feature the same functionality. First, you'll add the endpoint and key for your Azure OpenAI resource to the app's configuration file.

1. In Visual Studio Code, in the **Explorer** pane, browse to the **Labfiles/03-image-generation** folder and expand the **CSharp** or **Python** folder depending on your language preference. Each folder contains the language-specific files for an app into which you're going to integrate Azure OpenAI functionality.
2. In the **Explorer** pane, in the **CSharp** or **Python** folder, open the configuration file for your preferred language
 - **C#**: `appsettings.json`
 - **Python**: `.env`
3. Update the configuration values to include the **endpoint** and **key** from the Azure OpenAI resource you created (available on the **Keys and Endpoint** page for your Azure OpenAI resource in the Azure portal).
4. Save the configuration file.

View application code

Now you're ready to explore the code used to call the REST API and generate an image.

1. In the **Explorer** pane, select the main code file for your application:
 - **C#**: `Program.cs`
 - **Python**: `generate-image`
2. Review the code that the file contains, noting the following key features:

- The code makes an HTTPS request to the endpoint for your service, including the key for your service in the header. Both of these values are obtained from the configuration file.
- The request includes some parameters, including the prompt from on the image should be based, the number of images to generate, and the size of the generated image(s).
- The response includes a revised prompt that the DALL-E model extrapolated from the user-provided prompt to make it more descriptive, and the URL for the generated image.

Important: If you named your deployment anything other than the recommended `dalle3`, you'll need to update the code to use the name of your deployment.

Run the app

Now that you've reviewed the code, it's time to run it and generate some images.

1. Right-click the **CSharp** or **Python** folder containing your code files and open an integrated terminal. Then enter the appropriate command to run your application:

C#

```
dotnet run
```

Python

```
pip install requests
python generate-image.py
```

3. When prompted, enter a description for an image. For example, *A giraffe flying a kite*.
4. Wait for the image to be generated - a hyperlink will be displayed in the terminal pane. Then select the hyperlink to open a new browser tab and review the image that was generated.

TIP: If the app doesn't return a response, wait a minute and try again. Newly deployed resources can take up to 5 minutes to become available.

5. Close the browser tab containing the generated image and re-run the app to generate a new image with a different prompt.

Knowledge Check

1. You want to use a model in Azure OpenAI to generate images. Which model should you use? *

DALL-E

✓ The DALL-E model is used to generate images based on natural language prompts.

GPT-35-Turbo

Text-Davinci

2. Which playground in Azure AI Studio should you use to utilize the DALL-E model? *

Completions

Chat

Images

✓ The Images playground is used to explore image generation models.

3. In a REST request to generate images, what does the n parameter indicate? *

The description of the desired image.

The number of images to be generated

✓ The number of images to be generated is specified in the n parameter.

The size of the image to be generated

Summary

This module described the DALL-E image generation model, and how you can use it in the Azure OpenAI service to generate images based on natural language prompts.

You can explore DALL-E using the playground in Azure AI Studio, and you can use the REST API to build applications that use DALL-E to generate new images.

Tip: To learn more about using DALL-E in the Azure OpenAI service, see [Quickstart: Get started generating images using Azure OpenAI Service](#) in the Azure openAI service documentation.