Develop a vision-enabled generative AI application

1. Introduction

https://learn.microsoft.com/en-us/training/modules/develop-generative-ai-vision-apps/1-introduction

Introduction

Completed

- 1 minute

Generative AI models enable you to develop chat-based applications that reason over and respond to input. Often this input takes the form of a text-based prompt, but increasingly multimodal models that can respond to visual input are becoming available.

In this module, we'll discuss vision-enabled generative AI and explore how you can use Microsoft Foundry to create generative AI solutions that respond to prompts that include a mix of text and image data.

2. Deploy a multimodal model

https://learn.microsoft.com/en-us/training/modules/develop-generative-ai-vision-apps/2-deploy-multimodal-model

Deploy a multimodal model

Completed

- 3 minutes

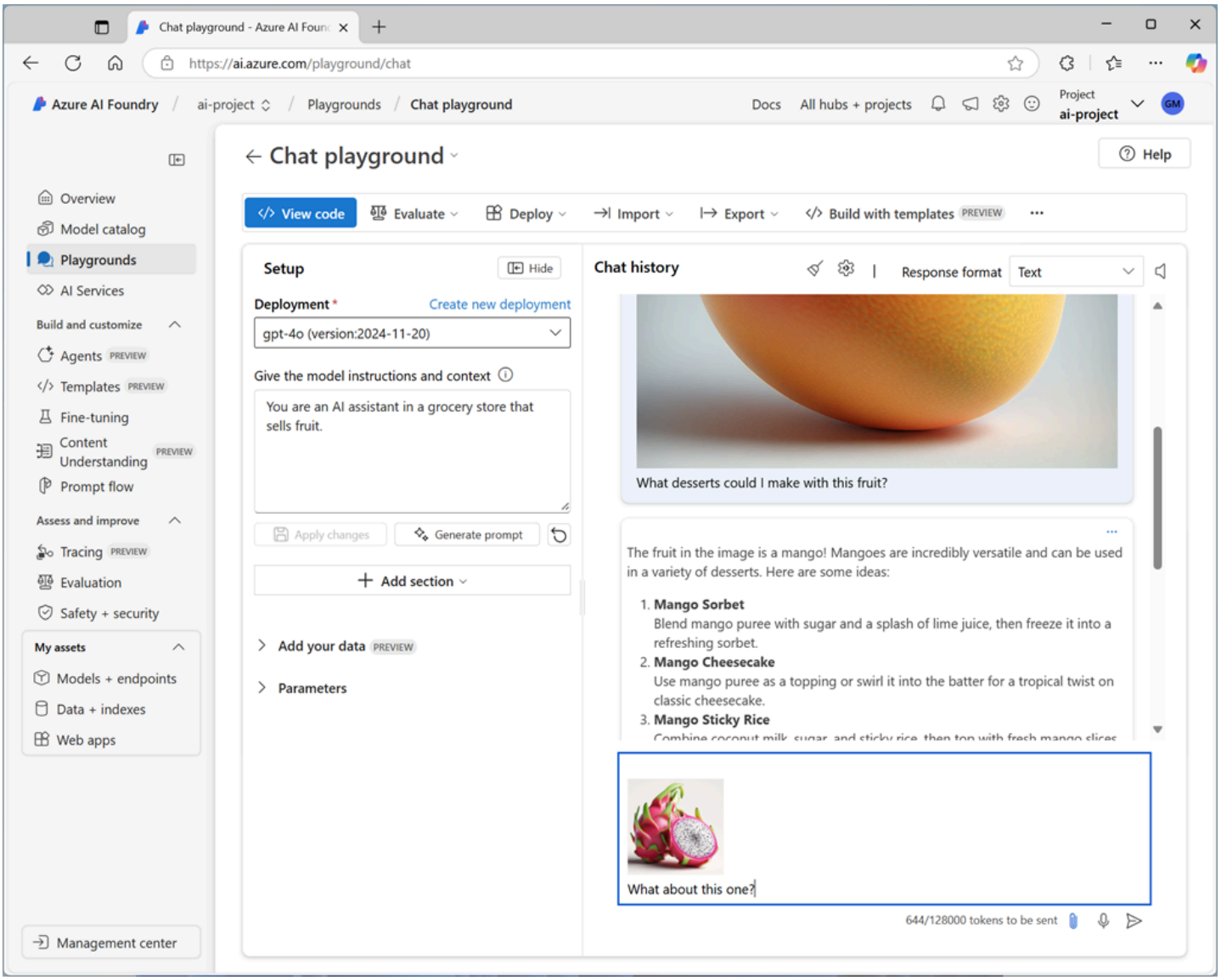
To handle prompts that include images, you need to deploy a *multimodal* generative AI model - in other words, a model that supports not only text-based input, but image-based (and in some cases, audio-based) input as well. Multimodal models available in Microsoft Foundry include (among others):

- Microsoft **Phi-4-multimodal-instruct**
- OpenAI **gpt-4o**
- OpenAI **gpt-4o-mini**

> **Tip**
>
> To learn more about available models in Microsoft Foundry, see the **Model catalog and collections in Microsoft Foundry portal** article in the Microsoft Foundry documentation.

Testing multimodal models with image-based prompts

After deploying a multimodal model, you can test it in the chat playground in Microsoft Foundry portal.



In the chat playground, you can upload an image from a local file and add text to the message to elicit a response from a multimodal model.

3. Develop a vision-based chat app

https://learn.microsoft.com/en-us/training/modules/develop-generative-ai-vision-apps/3-develop-visual-chat-app

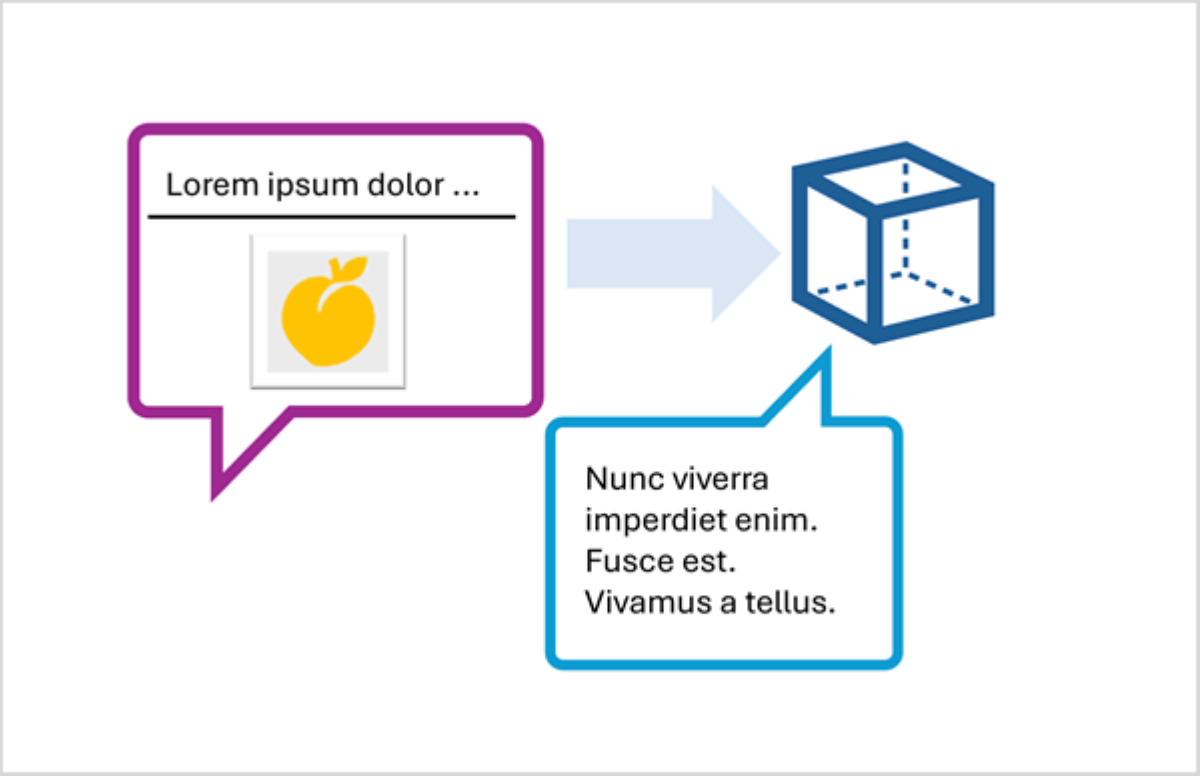
Develop a vision-based chat app

Completed

- 5 minutes

To develop a client app that engages in vision-based chats with a multimodal model, you can use the same basic techniques used for text-based chats. You require a connection to the endpoint where the model is deployed, and you use that endpoint to submit prompts that consists of messages to the model and process the responses.

The key difference is that prompts for a vision-based chat include multi-part user messages that contain both a *text* (or *audio* where supported) content item and an *image* content item.



The JSON representation of a prompt that includes a multi-part user message looks something like this:

```
{
    "messages": [
        { "role": "system", "content": "You are a helpful assistant." },
        { "role": "user", "content": [
            {
                "type": "text",
                "text": "Describe this picture:"
            },
            {
                "type": "image_url",
                "image_url": {
                    "url": "https://....."
                }
            }
        ] }
    ]
}
```

The image content item can be:

- A URL to an image file in a web site.
- Binary image data

When using binary data to submit a local image file, the **image_url** content takes the form of a base64 encoded value in a data URL format:

```
{
    "type": "image_url",
    "image_url": {
       "url": "<binary_image_data>"
    }
}
```

Depending on the model type, and where you deployed it, you can use Microsoft Azure AI Model Inference or OpenAI APIs to submit vision-based prompts. These libraries also provide language-specific SDKs that abstract the underlying REST APIs.

In the exercise that follows in this module, you can use the Python or .NET SDK for the Azure AI Model Inference API and the OpenAI API to develop a vision-enabled chat application.

4. Exercise - Develop a vision-enabled chat app

https://learn.microsoft.com/en-us/training/modules/develop-generative-ai-vision-apps/4-exercise

Exercise - Develop a vision-enabled chat app

Completed

- 30 minutes

If you have an Azure subscription, you can complete this exercise to develop a vision-enabled chat app.

> **Note**

> If you don't have an Azure subscription, you can sign up for an account, which includes credits for the first 30 days.

Launch the exercise and follow the instructions.

Launch Exercise

5. Module assessment

https://learn.microsoft.com/en-us/training/modules/develop-generative-ai-vision-apps/5-knowledge-check

Module assessment

Completed

- 3 minutes

6. Summary

https://learn.microsoft.com/en-us/training/modules/develop-generative-ai-vision-apps/6-summary

Summary

Completed

- 1 minute

In this module, you learned about vision-enabled generative AI models and how to implement chat solutions that include image-based input.

Vision-enabled models let you create AI solutions that can understand images and respond to related questions or instructions. Beyond just identifying objects in pictures, some models can also use reasoning based on what they see. For instance, they can interpret a chart or assess if an object is damaged.

> **Tip**
>
> For more information about working with multimodal models in Microsoft Foundry, see **How to use image and audio in chat completions with Azure AI model inference** and **Quickstart: Use images in your AI chats**.