Microsoft

Develop Generative AI Solutions with Azure OpenAI Service

© Copyright Microsoft Corporation. All rights reserved.

Agenda

- Get started with Azure OpenAI Service
- Develop apps with Azure OpenAI Service
- Apply prompt engineering with Azure OpenAI Service
- Use your own data with Azure OpenAI Service

© Copyright Microsoft Corporation. All rights reserved.

Get started with Azure OpenAI Service

© Copyright Microsoft Corporation. All rights reserved.

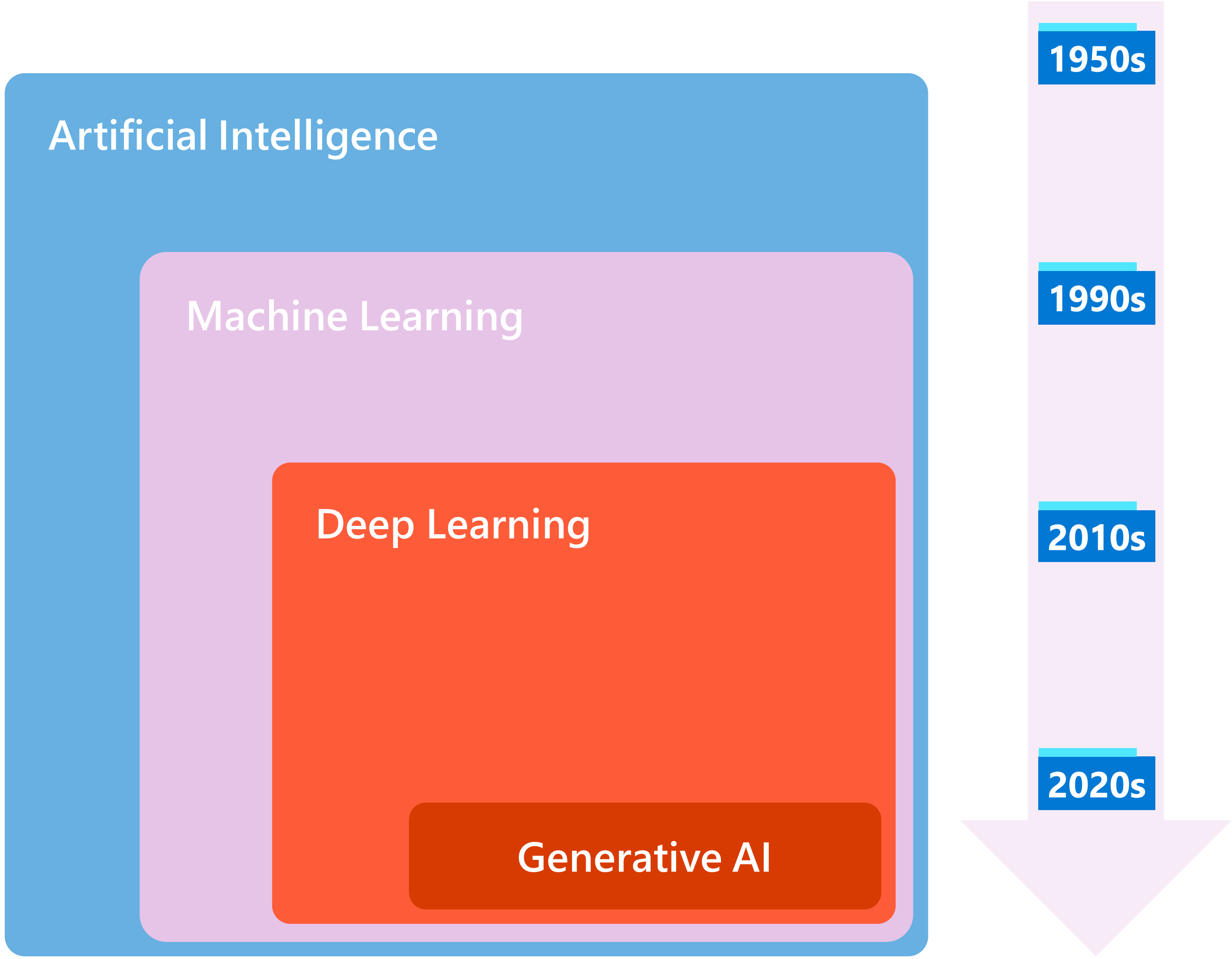
Learning Objectives

After completing this module, you will be able to:

1. Describe what generative AI is
2. Provision a resource and deploy a model
3. Use Azure OpenAI Studio

© Copyright Microsoft Corporation. All rights reserved.

What is generative AI?



Artificial Intelligence
the field of computer science that seeks to create intelligent machines that can replicate or exceed human intelligence

Machine Learning
subset of AI that enables machines to learn from existing data and improve upon that data to make decisions or predictions

Deep Learning
a machine learning technique in which layers of neural networks are used to process data and make decisions

Generative AI
Create new written, visual, and auditory content given prompts or existing data.

© Copyright Microsoft Corporation. All rights reserved.

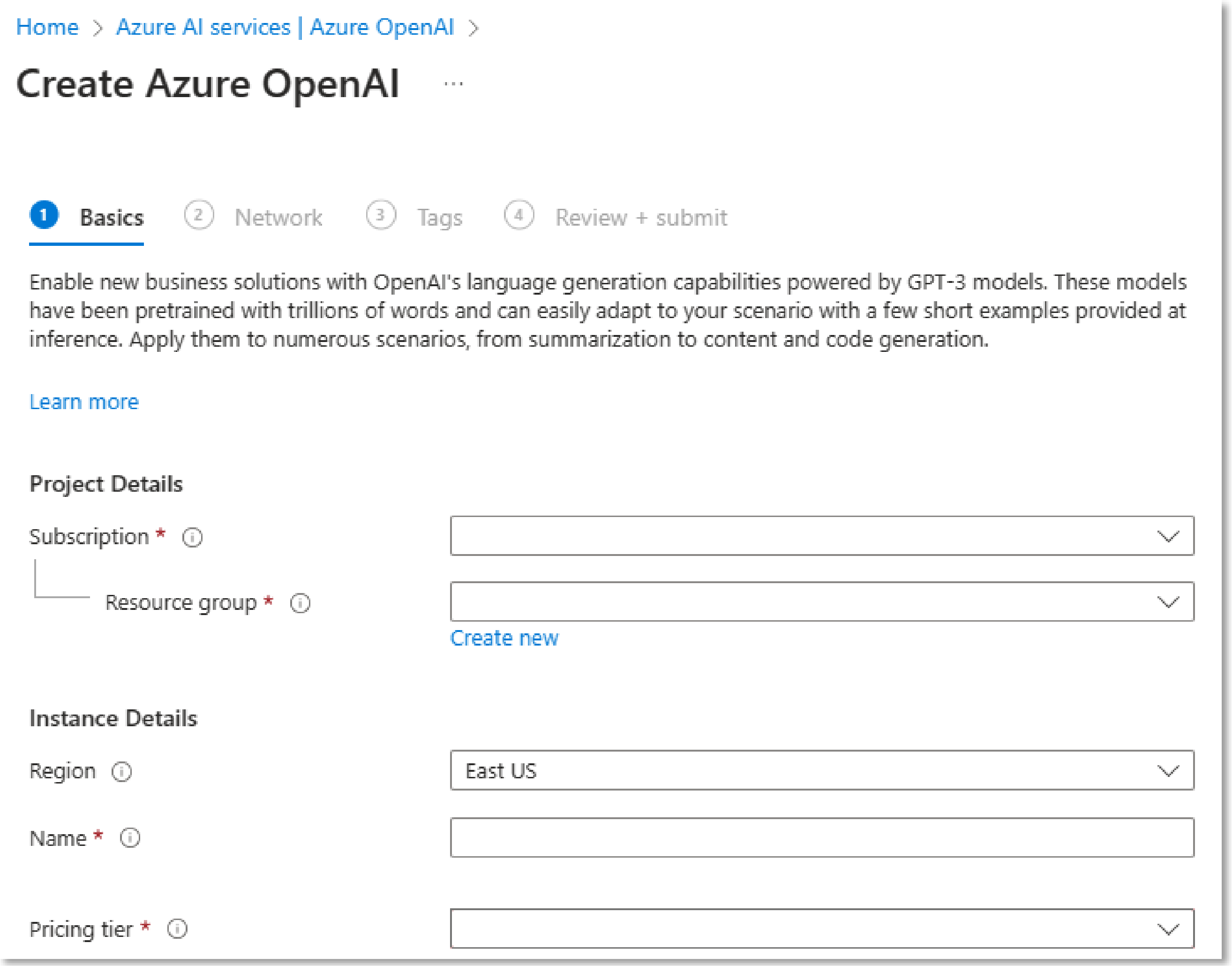
Provision an Azure OpenAI resource in Azure

Deploy a model in Azure OpenAI Studio to use it

1. Apply for access to the Azure OpenAI service: https://aka.ms/oaiapply
2. Create an **Azure OpenAI** resource in the Azure portal

Alternatively, use the Azure CLI

```
az cognitiveservices account create \
-n MyOpenAIResource \
-g MyResourceGroup \
-l eastus \
--kind OpenAI \
--sku s0 \
--subscription subscriptionID
```



Home > Azure AI services | Azure OpenAI >

Create Azure OpenAI ...

1 Basics 2 Network 3 Tags 4 Review + submit

Enable new business solutions with OpenAI's language generation capabilities powered by GPT-3 models. These models have been pretrained with trillions of words and can easily adapt to your scenario with a few short examples provided at inference. Apply them to numerous scenarios, from summarization to content and code generation.

Learn more

Project Details

Subscription *

Resource group *

Create new

Instance Details

Region: East US

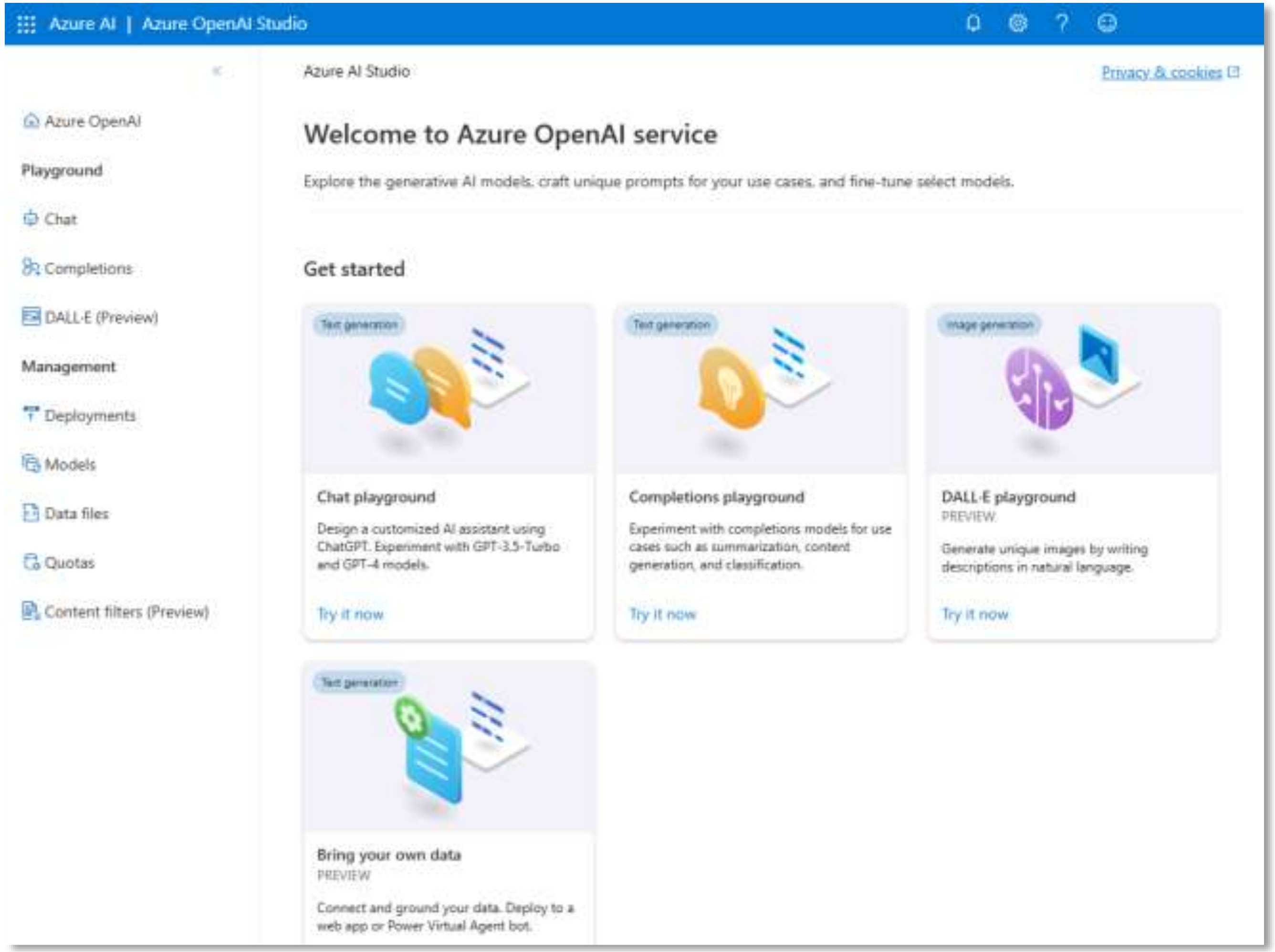
Name *

Pricing tier *

© Copyright Microsoft Corporation. All rights reserved.

Azure OpenAI Studio

- Web portal for working with Azure OpenAI models: https://oai.azure.com/
- View and deploy base models
- Connect your own data source
- Manage fine tuning and data files for custom models
- Test models in visual playgrounds:
  - **Completions** (GPT-3 and earlier models)
  - **Chat** (GPT-3.5-Turbo and later models)



Welcome to Azure OpenAI service

Explore the generative AI models, craft unique prompts for your use cases, and fine-tune select models.

Get started

Chat playground

Design a customized AI assistant using ChatGPT. Experiment with GPT-3.5-Turbo and GPT-4 models.

Completions playground

Experiment with completions models for use cases such as summarization, content generation, and classification.

DALL·E playground
PREVIEW

Generate unique images by writing descriptions in natural language.

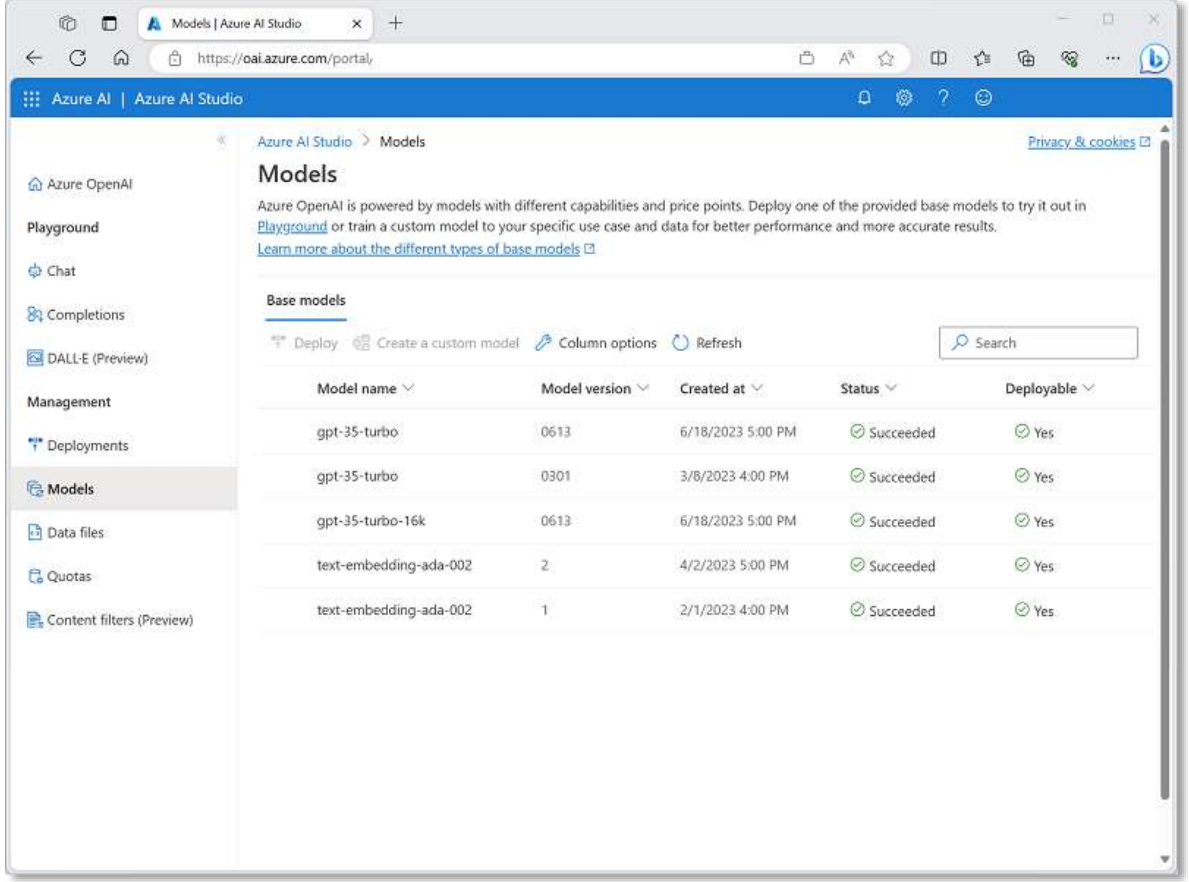
Bring your own data
PREVIEW

Connect and ground your data. Deploy to a web app or Power Virtual Agent bot.

© Copyright Microsoft Corporation. All rights reserved.

Types of generative AI model

Model Family	Description
GPT-4	Newest, most capable chat-based models for language and code generation
GPT-3	Natural language and code-generation models
Embeddings	Models that use embeddings for specific tasks (similarity, text search, and code search)
DALL-E	Image-generation model (*preview, restricted regions*)



Models

Azure OpenAI is powered by models with different capabilities and price points. Deploy one of the provided base models to try it out in Playground or train a custom model to your specific use case and data for better performance and more accurate results.

Learn more about the different types of base models

Model name	Model version	Created at	Status	Deployable
gpt-35-turbo	0613	6/18/2023 5:00 PM	✓ Succeeded	✓ Yes
gpt-35-turbo	0301	3/8/2023 4:00 PM	✓ Succeeded	✓ Yes
gpt-35-turbo-16k	0613	6/18/2023 5:00 PM	✓ Succeeded	✓ Yes
text-embedding-ada-002	2	4/2/2023 5:00 PM	✓ Succeeded	✓ Yes
text-embedding-ada-002	1	2/1/2023 4:00 PM	✓ Succeeded	✓ Yes

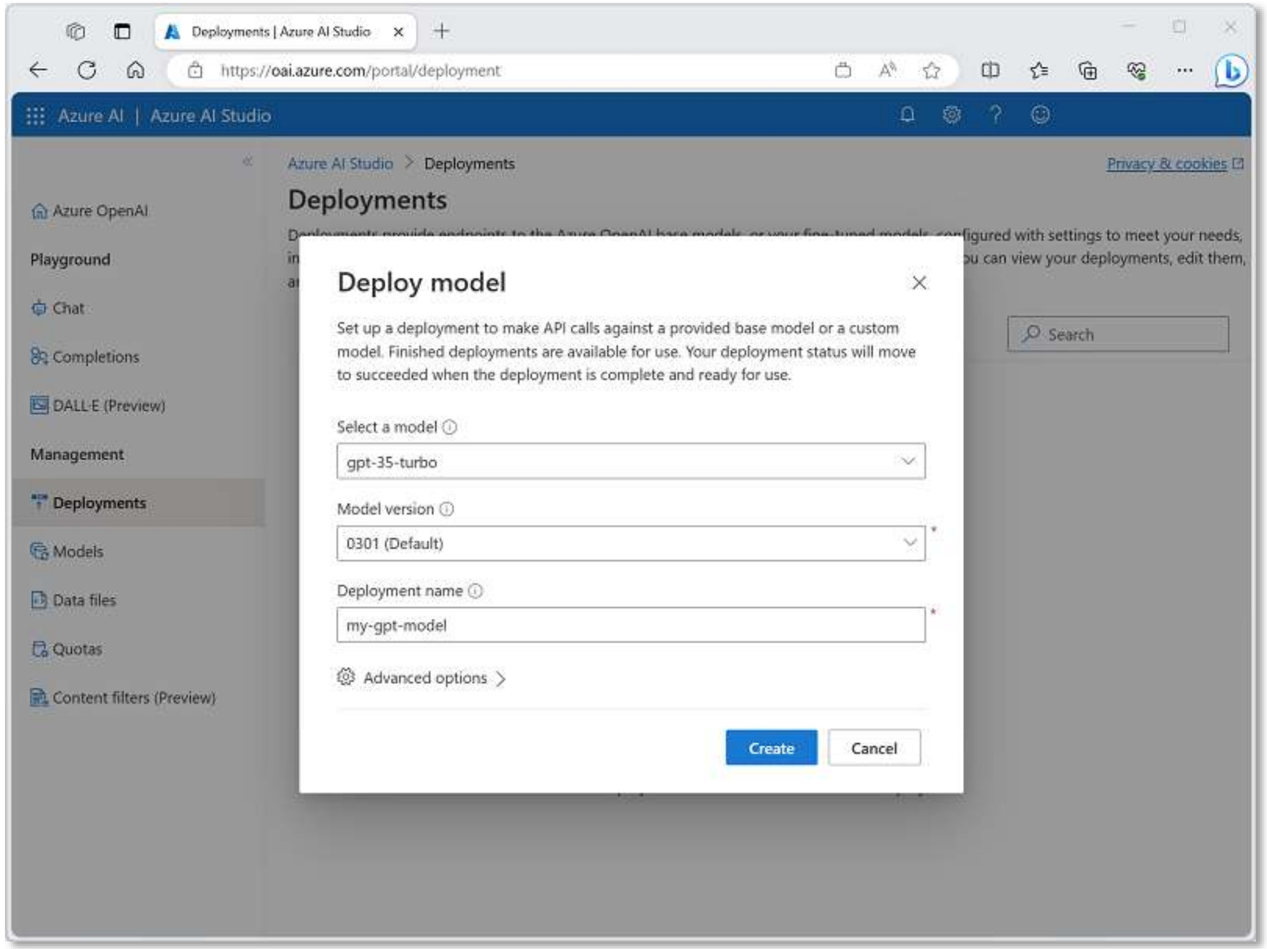
© Copyright Microsoft Corporation. All rights reserved.

Deploying generative AI models

Deploy a model in Azure OpenAI Studio to use it

- You can deploy one or more instances of each available model
- The number of deployments depends on your quota, which you can see in the portal
- Alternatively, use the Azure CLI

```
az cognitiveservices account deployment create \
   -g myResourceGroupName \
   -n MyOpenAIResource \
   --deployment-name my-gpt-model \
   --model-name gpt-35-turbo \
   --model-version “0301"  \
   --model-format OpenAI \
   --scale-settings-scale-type "Standard"
```



Deploy model

Set up a deployment to make API calls against a provided base model or a custom model. Finished deployments are available for use. Your deployment status will move to succeeded when the deployment is complete and ready for use.

Select a model: gpt-35-turbo

Model version: 0301 (Default)

Deployment name: my-gpt-model

© Copyright Microsoft Corporation. All rights reserved.

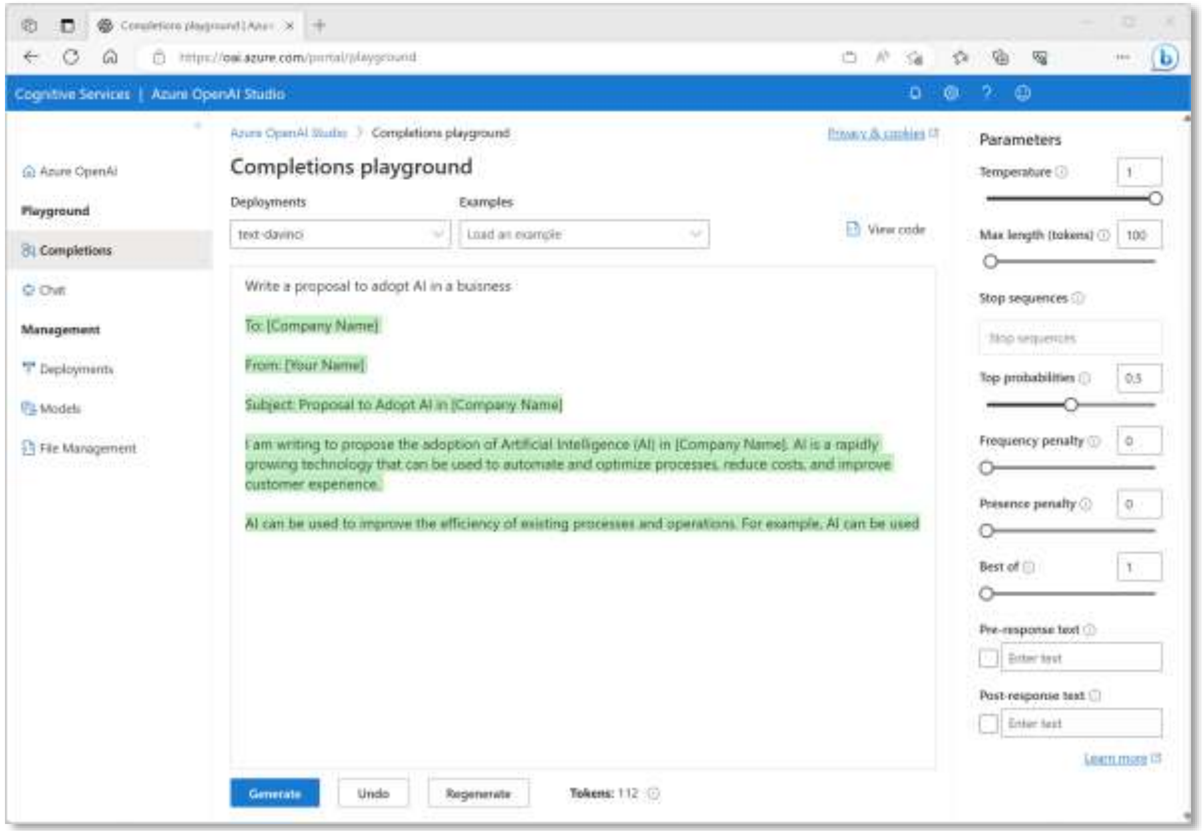
Using prompts to get completions from models

Task	Prompt	Completion
Classifying content	Tweet: I enjoyed the training course. Sentiment:	Positive
Generating new content	Write a poem about databases	Databases, oh databases, You keep our information safe, From the small to the large, You store our data in a place.
Transformation/Translation	English: Hello French:	Bonjour
Summarization	Scotland is [*long description of Scotland…*] Summarize the previous text	Scotland is [*summarized description…*]
Continuation	One way to grow tomatoes is to	start with seeds…
Question answering	How many moons does Earth have?	Earth has one moon.
Chat	*Setup, followed by messages…*	*A sequence of relevant responses*

© Copyright Microsoft Corporation. All rights reserved.

Testing models in Azure OpenAI Studio playgrounds

Completions playground



Completions playground

Deployments: text-davinci

Examples: Load an example

Write a proposal to adopt AI in a buisness

To: [Company Name]

From: [Your Name]

Subject: Proposal to Adopt AI in [Company Name]

I am writing to propose the adoption of Artificial Intelligence (AI) in [Company Name]. AI is a rapidly growing technology that can be used to automate and optimize processes, reduce costs, and improve customer experience.

AI can be used to improve the efficiency of existing processes and operations. For example, AI can be used

Tokens: 112

Parameters

Temperature: 1

Max length (tokens): 100

Stop sequences

Top probabilities: 0.5

Frequency penalty: 0

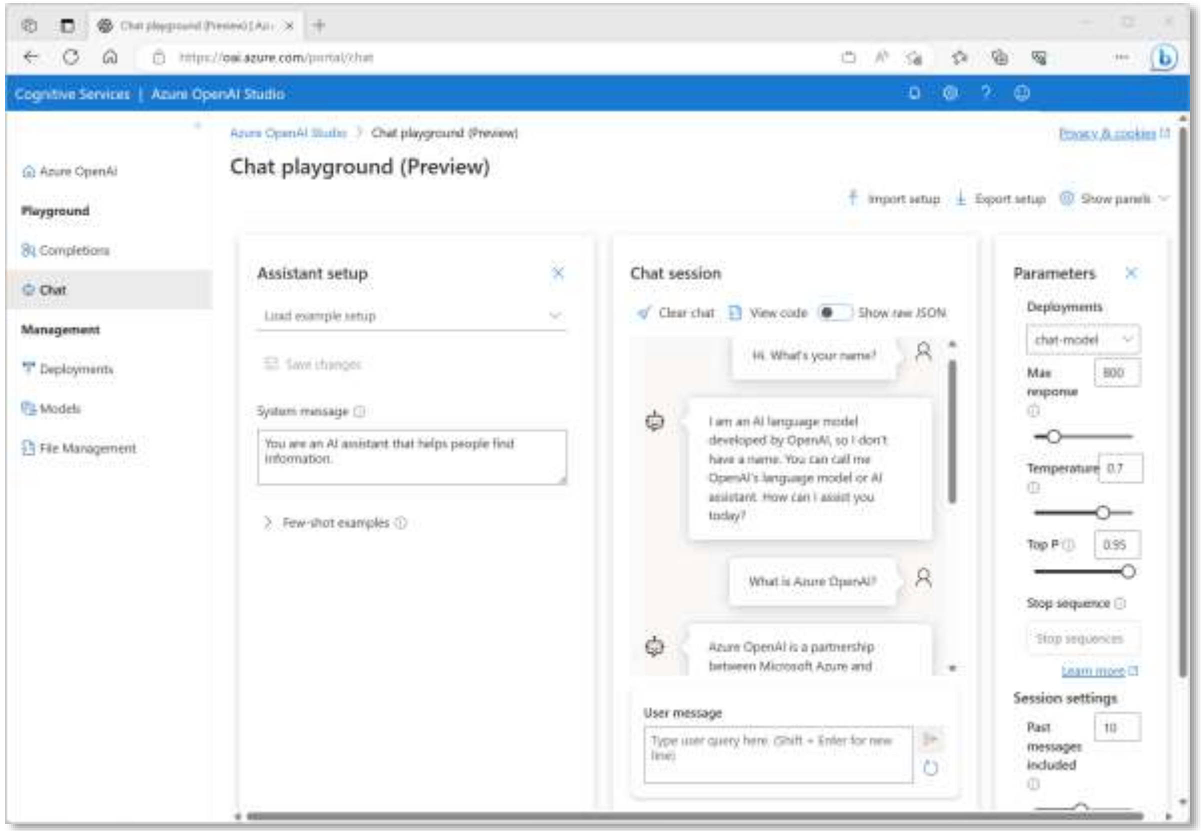
Presence penalty: 0

Best of: 1

Pre-response text: ☐ Enter text

Post-response text: ☐ Enter text

Chat playground



Chat playground (Preview)

Assistant setup

System message: You are an AI assistant that helps people find information.

Chat session

Hi. What's your name?

I am an AI language model developed by OpenAI, so I don't have a name. You can call me OpenAI's language model or AI assistant. How can I assist you today?

What is Azure OpenAI?

Azure OpenAI is a partnership between Microsoft Azure and

User message: Type user query here. (Shift + Enter for new line)

Parameters

Deployments: chat-model

Max response: 800

Temperature: 0.7

Top P: 0.95

Stop sequence: Stop sequences

Session settings

Past messages included: 10

© Copyright Microsoft Corporation. All rights reserved.

Develop apps with Azure OpenAI Service

Learning Objectives

After completing this module, you will be able to:

1. Integrate Azure OpenAI into your app
2. Use the REST API
3. Use language specific SDKs

© Copyright Microsoft Corporation. All rights reserved.

Integrating Azure OpenAI into your app

Applications submit prompts to deployed models. Responses are completions.

Three REST API endpoints:

- **Completion** - model takes an input prompt, and generates one or more predicted completions
- **Embeddings** - model takes input and returns a vector representation of that input
- **ChatCompletion** - model takes input in the form of a chat conversation (where roles are specified with the message they send), and the next chat completion is generated

ChatCompletion will be the endpoint we focus on for this course

Use **Completion** and **Embeddings** with GPT-3 based models
Use **ChatCompletion** with GPT-35-Turbo and later models

© Copyright Microsoft Corporation. All rights reserved.

Using the Azure OpenAI REST API

Completion Endpoint

https://*endpoint*.openai.azure.com/openai/deployments/*deployment*/**completions**

```
{
  "prompt": "Your favorite Shakespeare
             play is",
  "max_tokens": 5
}
```

```
{
    "id": “1234….",
    "object": "text_completion",
    "created": 1679001781,
    "model": “gpt-35-turbo",
    "choices": [
        {
            "text": "Macbeth",
            "index": 0,
            "logprobs": null,
            "finish_reason": "stop"
        }
    ]
}
```

© Copyright Microsoft Corporation. All rights reserved.

Using the Azure OpenAI REST API

Embedding Endpoint

https://*endpoint*.openai.azure.com/openai/deployments/*deployment*/**embeddings**

```
{
  "input": "The food was delicious and
            the waiter was very
            friendly..."
}
```

```
{
  "object": "list",
  "data": [
    {
      "object": "embedding",
      "embedding": [
        0.0172990688066482523,
        ....
        0.0134544348834753042,
      ],
      "index": 0
    }
  ],
  "model": "text-embedding-ada:002"
}
```

© Copyright Microsoft Corporation. All rights reserved.

Using the Azure OpenAI REST API

ChatCompletion Endpoint

https://*endpoint*.openai.azure.com/openai/deployments/*deployment*/**chat/completions**

```
{
  "messages":[
    {"role": "system",
     "content": "You are an assistant
      that teaches people about AI."},
    {"role": "user",
     "content": "Does Azure OpenAI
          support multiple languages?"},
    {"role": "assistant",
     "content": "Yes, Azure OpenAI
          supports several languages."},
    {"role": "user",
     "content": "Do other Cognitive
         Services support translation?"}
  ]
}
```

```
{
    "id": “unique_id", "object": "chat.completion",
    "created": 1679001781, "model": "gpt-35-turbo",
    "usage": { "prompt_tokens": 95,
               "completion_tokens": 84, "total_tokens": 179},
    "choices": [
          {"message":
               {  "role": "assistant",
                   "content": "Yes, other Azure Cognitive
                       Services also support translation…"},
            "finish_reason": "stop",
            "index": 0}
    ]
}
```

© Copyright Microsoft Corporation. All rights reserved.

Using Azure OpenAI SDKs – Microsoft C#

```
using Azure.AI.OpenAI;

OpenAIClient client = new OpenAIClient(new Uri(yourEndpoint), new AzureKeyCredential(yourKey));

// Build completion options object
ChatCompletionsOptions chatCompletionsOptions = new ChatCompletionsOptions()
{
    Messages =
    {
        new ChatMessage(ChatRole.System, "You are a helpful assistant."),
        new ChatMessage(ChatRole.User, "What is Azure OpenAI?"),
    },
    MaxTokens = 120,
    Temperature = 0.7f,
    DeploymentName = oaiModelName
};
// Send request to Azure OpenAI model
ChatCompletions response = client.GetChatCompletions(chatCompletionsOptions);
string completion = response.Choices[0].Message.Content;
Console.WriteLine($"Answer: {completion}");
```

© Copyright Microsoft Corporation. All rights reserved.

Using Azure OpenAI SDKs – Python

```
from openai import AzureOpenAI

# Initialize the Azure OpenAI client
client = AzureOpenAI(
        azure_endpoint = your_endpoint,
        api_key=your_key,
        api_version="2023-05-15")

# Send request to Azure OpenAI model
response = client.chat.completions.create(
   model=azure_oai_model,
   temperature=0.7,
   max_tokens=120,
   messages=[
       {"role": "system", "content": "You are a helpful assistant."},
       {"role": "user", "content": "Summarize the following text in 20 words or less:\n" + text}
   ]
)

print("Summary: " + response.choices[0].message.content + "\n")
```

© Copyright Microsoft Corporation. All rights reserved.

Apply prompt engineering with Azure OpenAI Service

Learning Objectives

After completing this module, you will be able to:

1. Understand what prompt engineering is
2. Understand considerations for different endpoints
3. Explore different techniques of prompt engineering

© Copyright Microsoft Corporation. All rights reserved.

What is Prompt Engineering?

Constructing prompts to:

- Maximize relevancy and accuracy of completions
- Specify formatting and style of completions
- Provide conversational context
- Mitigate bias and improve fairness

© Copyright Microsoft Corporation. All rights reserved.

Considerations for API endpoints

Completions API (*GPT-35-Turbo and earlier*)

Include context in the prompt (as a *meta prompt*)

```
{
  prompt: "You are a professional events
           planner. Write an invitation
           for a party to celebrate the
           launch of a new product."
}
```

ChatCompletions API (*GPT-35-Turbo and later*)

Set context in a system message

```
{
  "messages":[
    {"role": "system", "content": "You are a
        professional events planner."},
    {"role": "user", "content": "Write an
        invitation for a party to celebrate
        the launch of a new product."}
  ]
}
```

© Copyright Microsoft Corporation. All rights reserved.

Providing clear instructions

Write a product description for a new water bottle

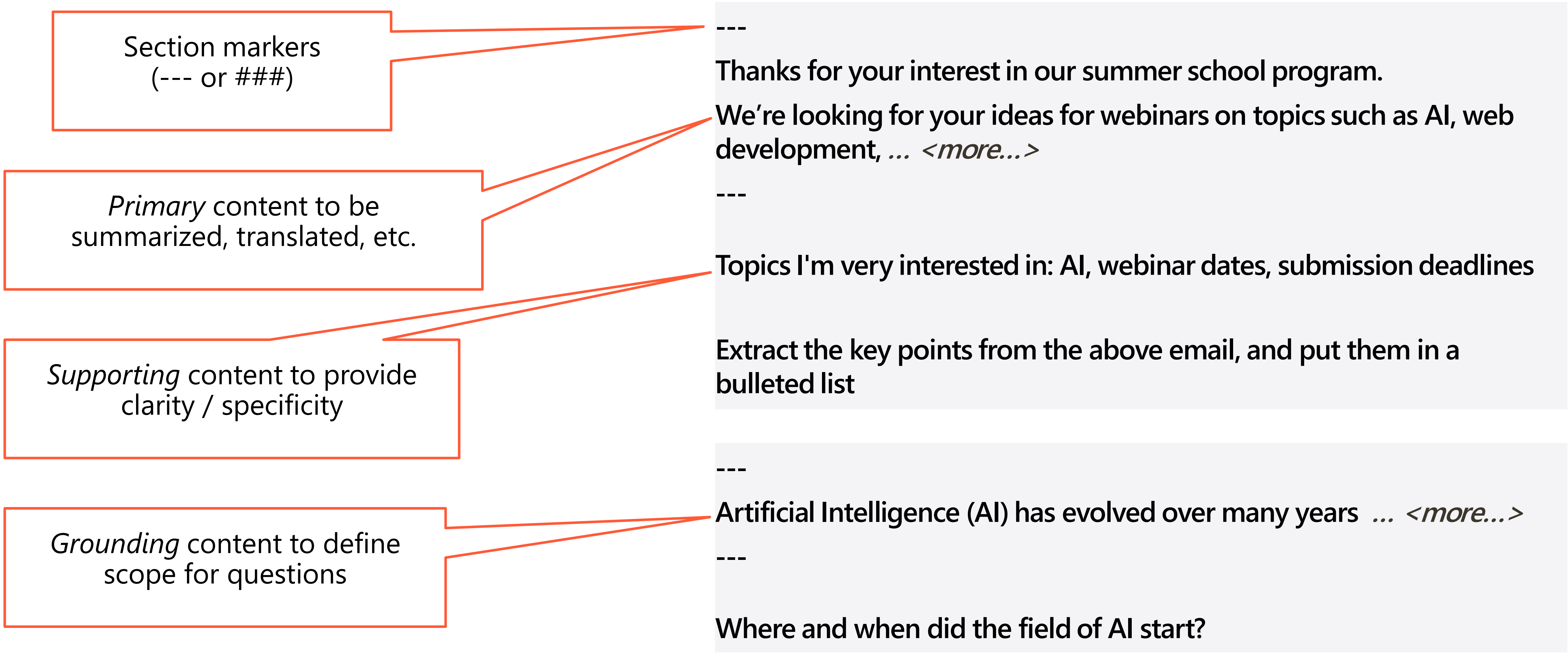
Introducing the latest addition to our product line - the innovative and eco-friendly water bottle. Made from high-quality, BPA-free materials, this bottle is designed to keep your beverages at the perfect temperature for hours. Its sleek and stylish design makes it the perfect accessory for any outdoor activity or workout session. With its convenient carrying handle and leak-proof lid, you can take it with you wherever you go, without worrying about spills or leaks. Plus, its easy-to-clean design ensures that you can use it again and again, making it the perfect choice for anyone looking for a reliable and sustainable water bottle.

Write a product description for a new water bottle that is 100% recycled. Be sure to include that it comes in natural colors with no dyes, and each purchase removes 10 pounds of plastic from our oceans

Introducing our newest water bottle, the eco-friendly choice for the conscious consumer. Made from 100% recycled materials and available in natural colors with no dyes, this bottle is not only stylish but also sustainable. Each purchase of this bottle removes 10 pounds of plastic from our oceans, making it a responsible choice for those who care about the environment. Its durable and leak-proof design makes it perfect for outdoor activities, while its easy-to-carry handle makes it convenient to take on the go. Join us in our mission to reduce plastic waste and choose this eco-friendly water bottle for your daily hydration needs.

© Copyright Microsoft Corporation. All rights reserved.

Primary, *supporting*, and *grounding* content



Section markers (--- or ###)

Primary content to be summarized, translated, etc.

Supporting content to provide clarity / specificity

Grounding content to define scope for questions

```
---
Thanks for your interest in our summer school program.
We're looking for your ideas for webinars on topics such as AI, web
development, … <more…>
---

Topics I'm very interested in: AI, webinar dates, submission deadlines

Extract the key points from the above email, and put them in a
bulleted list
```

```
---
Artificial Intelligence (AI) has evolved over many years … <more…>
---

Where and when did the field of AI start?
```

© Copyright Microsoft Corporation. All rights reserved.

Cues

- Cues provide a starting point on which the completion builds
- Use cues to set the scope and format for the output
- Cues are particularly useful when generating code. For example:
    - Provide the first keyword in a SQL SELECT statement
    - Define a name and parameters for a Python function

```
---
Review: The writing was good but the movie was too long.
Review: The special effects were terrible.
<more>...
---

Summarize the reviews above.
Most common complaints:
-
```

- The movie was too long.
- The special effects were terrible.

© Copyright Microsoft Corporation. All rights reserved.

Requesting output composition

Write a table in markdown with 6 animals in it, with their genus and species

```
| Animal           | Genus          | Species        |
| ---------------- |:--------------:|:--------------:|
| Red Panda        | Ailurus        | fulgens        |
| African Elephant | Loxodonta      | africana       |
| Snow Leopard     | Panthera       | uncia          |
| Arctic Fox       | Vulpes         | lagopus        |
| Chimpanzee       | Pan            | troglodytes    |
| Siberian Tiger   | Panthera       | tigris altaica |
```

© Copyright Microsoft Corporation. All rights reserved.

Using a system message

When using the ChatCompletions API, a system message sets context that can affect completions

```
{"role": "system", "content": "You are a casual, helpful assistant.
                               You will talk like an American old western film character."},
{"role": "user", "content": "Can you direct me to the library?"}
```

```
{"role": "assistant", "content": "Well howdy there, stranger! The library, huh?
                                  Y'all just head down the main road till you hit the town
                                  square. Once you're there, take a left and follow the street
                                  for a spell. You'll see the library on your right, can’t
                                  miss it. Happy trails!"}
```

© Copyright Microsoft Corporation. All rights reserved.

Conversation history and *few shot* learning

- Including previous messages helps retain the context and style of a conversation
- Starting a conversation with user-defined messages and responses (*few shot* learning) sets examples for the expected responses

```
{"role": "system", "content": "You are an assistant that evaluates the sentiment of customer
feedback."},
{"role": "user", "content": "That was an awesome experience"},
{"role": "assistant", "content": "positive"},
{"role": "user", "content": "I won't do that again"},
{"role": "assistant", "content": "negative"},
{"role": "user", "content": "That was not worth my time"},
{"role": "assistant", "content": "negative"},
{"role": "user", "content": "You can't miss this"}
```

```
{"role": "assistant", "content": "positive"},
```

© Copyright Microsoft Corporation. All rights reserved.

Chain of thought

Ask a model to break down its response and explain its reasoning

What sport is easiest to learn but hardest to master? Give a step by step approach of your thoughts, ending in your answer

Step 1: Identify the criteria for "easy to learn" and "hard to master" sports.
For a sport to be considered easy to learn, it should have simple rules and require minimal equipment For a sport to be considered hard to master, it should require years of practice to perfect and have a large variety of techniques and strategies…

Step 2: Consider different sports that fit these criteria.
Some sports that could be considered easy to learn but hard to master include:
- Tennis: Simple rules, easy to pick up, but requires years of practice to master all the different shots and strategies.
- Golf: Basic swing mechanics are easy to learn but perfecting them takes a lifetime of practice.
- …

Step 3: Evaluate each sport based on the criteria.
Tennis and golf both require expensive equipment, which could make them less accessible to beginners. In terms of skill ceiling and time required to master, tennis and golf both have a high skill ceiling and require years of practice to perfect….

Step 4: Make a decision based on the evaluation.
Based on the above criteria and evaluation, I would say that tennis is the sport that is easiest to learn but hardest to master. While it may require expensive equipment, it is still accessible to many people. The basic rules and techniques are easy to learn, but mastering all the different shots and strategies takes years of practice…

© Copyright Microsoft Corporation. All rights reserved.

Use your own data with Azure OpenAI Service

Learning Objectives

After completing this module, you will be able to:

1. Understand how using your own data works
2. Use the REST API
3. Use language specific SDKs

© Copyright Microsoft Corporation. All rights reserved.

How Azure OpenAI can use your data

Set up your data source

- Use an existing data source, such as an Azure search resource
- Use the Azure OpenAI studio to create that data source, if you don't already have one
- When creating the data source, you can use data already in your account such as blob storage

Configure the studio or your app to connect to that data source

- In the studio, set up the connection by pointing it to the data source
- In your app, specify the data source in the prompt parameters

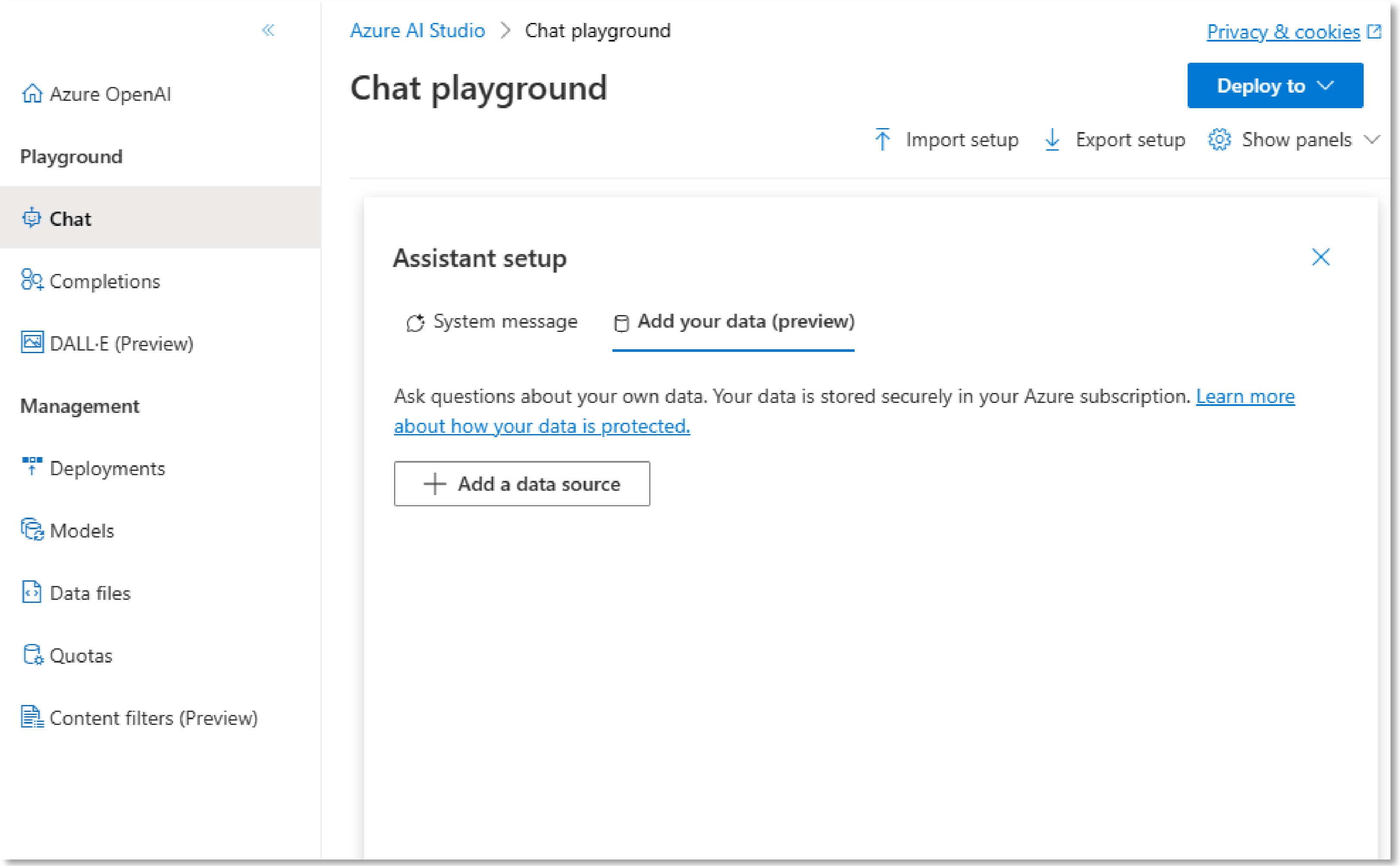
Use the Azure OpenAI model, which now uses your data for grounding

- Chat with the AI models like normal
- If the data source has relevant information about the prompt, it will use that data
- You can specify if the AI model is limited to just your data source

© Copyright Microsoft Corporation. All rights reserved.

Connect to your data source

- Add your data source in the Chat playground, under Assistant setup
- Use an existing data source, or use that wizard to create a new one
- Once connected, a new chat session will start. Chat like normal, and see how the AI model references that data



Ask questions about your own data. Your data is stored securely in your Azure subscription. Learn more about how your data is protected.

© Copyright Microsoft Corporation. All rights reserved.

Using the Azure OpenAI REST API

Using your own data

https://*endpoint*.openai.azure.com/openai/deployments/*deployment*/**extensions/chat/completions**?api-version=*version*

- With each call, you need to specify the data source values, along with the messages array and any other parameters

- The REST endpoint will be different than normal chats

```
{
    "dataSources": [
        {
            "type": "AzureCognitiveSearch",
            "parameters": {
                "endpoint": "<your_search_endpoint>",
                "key": "<your_search_endpoint>",
                "indexName": "<your_search_index>"
            }
        }
    ],
    "messages":[
                ...
    ]
}
```

© Copyright Microsoft Corporation. All rights reserved.

Using Azure OpenAI SDKs – Microsoft C#

```
AzureCognitiveSearchChatExtensionConfiguration ownDataConfig = new()
{
        SearchEndpoint = new Uri(searchEndpoint),
        IndexName = searchIndex
};
ownDataConfig.SetSearchKey(searchKey);

ChatCompletionsOptions chatCompletionsOptions = new ChatCompletionsOptions()
{
    Messages =
    {
        new ChatMessage(ChatRole.User, “I want to go to New York. Where should I stay?”)
    },
    MaxTokens = 600,
    Temperature = 0.9f,
    DeploymentName = modelName,
    AzureExtensionsOptions = new AzureChatExtensionsOptions()
    {
        Extensions = {ownDataConfig}
    }
};
```

© Copyright Microsoft Corporation. All rights reserved.

Using Azure OpenAI SDKs – Python

```
# Create extension config for own data
extension_config = dict(dataSources = [
  {
    "type": "AzureCognitiveSearch",
    "parameters": {
      "endpoint":search_endpoint,
      "key": search_key,
      "indexName": search_index,
    }
  }])
response = client.chat.completions.create(
  model = azure_oai_model,
  temperature = 0.5,
  max_tokens = 1000,
  messages = [
    {"role": "user", "content": "I want to go to New York. Where should I stay?"}}
  ],
  extra_body = extension_config
)
```

© Copyright Microsoft Corporation. All rights reserved.

Learning Recap

In this learning path, we:

Created and deployed Azure OpenAI resources

Integrated Azure OpenAI into your application through REST APIs and SDKs

Explored prompt engineering techniques to improve model responses

Connected your own data for grounding an Azure OpenAI model

© Copyright Microsoft Corporation. All rights reserved.

Microsoft

© Copyright Microsoft Corporation. All rights reserved.