

Lab 2- Intro to Azure OpenAI Studio / Prompt Engineering

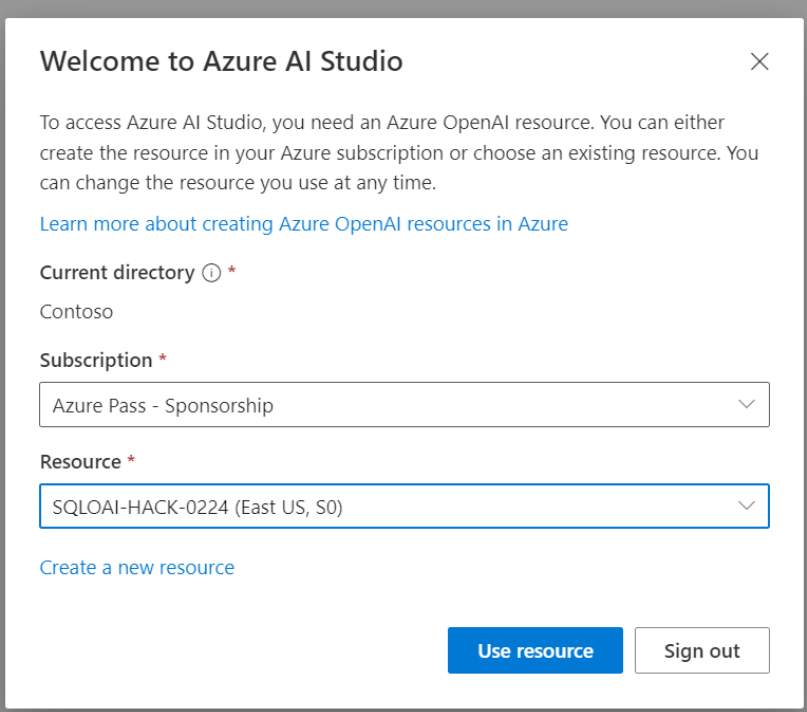
V1.0

Contents

1. Introduction to Azure OpenAI Studio	2
3. Prompt Engineering	7

1. Introduction to Azure OpenAI Studio

In this section, we will look at the Azure OpenAI Studio and test some of the possible natural language processing capabilities.

Narrative	Screenshot	Notes
<p>In the browser (can be outside of your VM), navigate to the Azure OpenAI studio: https://oai.azure.com/portal</p> <p>Select the following:</p> <ul style="list-style-type: none"> • Directory: Contoso • Subscription: Azure Pass – Sponsorship • Azure OpenAI resource: SQLOAI-HACK-0224 <p>Click Use Resource.</p>	 <p>Welcome to Azure AI Studio [Close]</p> <p>To access Azure AI Studio, you need an Azure OpenAI resource. You can either create the resource in your Azure subscription or choose an existing resource. You can change the resource you use at any time.</p> <p>Learn more about creating Azure OpenAI resources in Azure</p> <p>Current directory ⓘ *</p> <p>Contoso</p> <p>Subscription *</p> <p>Azure Pass - Sponsorship [Dropdown Arrow]</p> <p>Resource *</p> <p>SQLOAI-HACK-0224 (East US, S0) [Dropdown Arrow]</p> <p>Create a new resource</p> <p>Use resource Sign out</p>	

Only for your info:

To use the Azure OpenAI models, it is required to have model deployments. For this lab, **models have been deployed** upfront for you to use.

Some resources to read up:

- Overview of models: [Azure OpenAI Service models - Azure OpenAI | Microsoft Learn](#)
- Create resource: [How-to: Create and deploy an Azure OpenAI Service resource - Azure OpenAI | Microsoft Learn](#)
- Deploy models: [How-to: Create and deploy an Azure OpenAI Service resource - Azure OpenAI | Microsoft Learn](#)
- Information on quota/capacity management: [Azure OpenAI Service quotas and limits - Azure AI services | Microsoft Learn](#)

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 ⓘ

Auto-update to default

Deployment name ⓘ

gpt-35-turbo

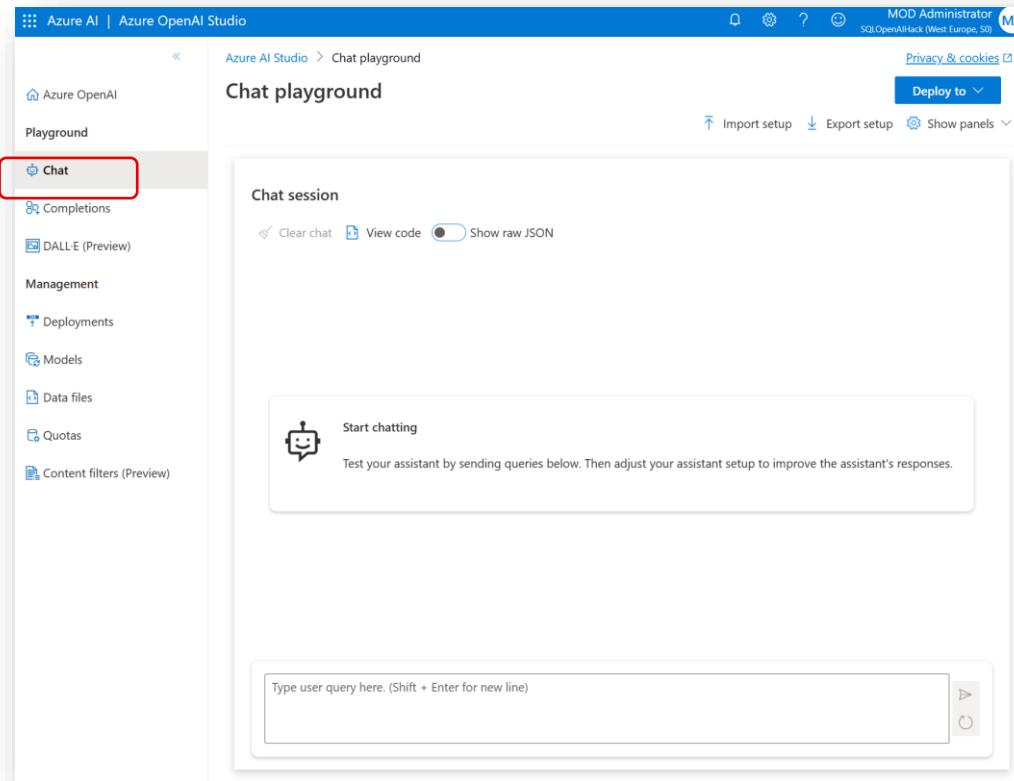
⚙️ Advanced options >

Create Cancel

Select **Chat Playground** on the left to test the chat experience.

Here you can start exploring Azure OpenAI capabilities with a no-code approach. It's simply a text box where you can submit a prompt (instruction) to generate a completion (output). From this page, you can quickly iterate and experiment with the capabilities.

You can try out the following sample prompts by pasting them in the chat field and ask follow-up questions on your query.



Sample 1: Natural Language to SQL

Prompt (paste in the chat field):

Postgres SQL tables, with their properties:

#

Employee(id, name, department_id)

Department(id, name, address)

Salary_Payments(id, employee_id, amount, date)

#

A query to list the names of the departments which employed more than 10 employees in the last 3 months

SELECT

Postgres SQL tables, with their properties:

#

Employee(id, name, department_id)

Department(id, name, address)

Salary_Payments(id, employee_id, amount, date)

#

A query to list the names of the departments which employed more than 10 employees in the last 3 months

SELECT

To list the names of the departments which employed more than 10 employees in the last 3 months, you would need to perform a query that joins the `Employee` and `Department` tables, filters the employee count based on the `date` column from the `Salary_Payments` table, and then groups the results by department to count the employees. The query would look something like this:

```
SELECT d.name
FROM Department d
JOIN Employee e ON d.id = e.department_id
JOIN Salary_Payments sp ON e.id = sp.employee_id
WHERE sp.date >= CURRENT_DATE - INTERVAL '3 months'
GROUP BY d.name
HAVING COUNT(DISTINCT e.id) > 10;
```

This query assumes that if an employee has received a salary payment in the last 3 months, they are considered employed in that time frame. Adjustments might be needed depending on the exact definition of "employed" in this context.

Sample 2: Explain SQL Query

Explain what the below SQL query does. Also answer why someone might be interested in this time period, and why a company might be interested in this SQL query.

```
SELECT c.customer_id
FROM Customers c
JOIN Streaming s
ON c.customer_id = s.customer_id
WHERE c.signup_date BETWEEN '2020-03-01' AND '2020-03-31'
AND s.watch_date BETWEEN c.signup_date AND DATE_ADD(c.signup_date, INTERVAL 30 DAY)
GROUP BY c.customer_id
HAVING SUM(s.watch_minutes) > 50 * 60
```

Explanation:

This SQL query is used to identify customers who have watched more than 50 hours (or 3000 minutes, as the time is represented in minutes in the database) of content within the first 30 days of their signup date.

The query operates in several steps:

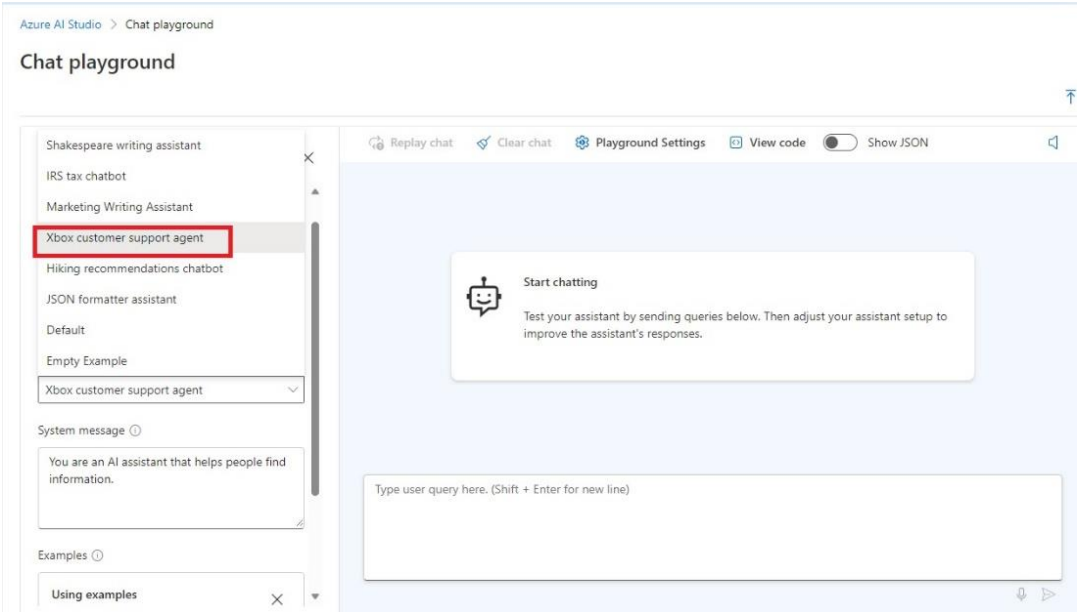
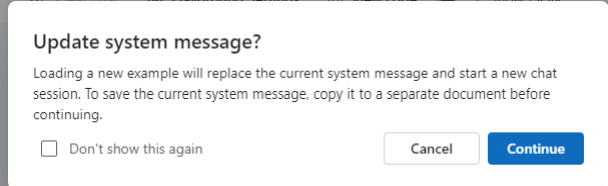
1. Selects the customer id from the Customers table (represented as 'c').
2. Joins the Customers table with the Streaming table (represented as 's') on the common field 'customer_id'.
3. Applies a filter on the signup date of the customers, selecting only those who signed up between the 1st and the 31st of March, 2020.
4. Further filters the streaming data to consider only the watch dates within the first 30 days of each customer's signup date.
5. Finally, it groups the results by customer id and only includes those who have a sum of watch minutes greater than 3000 minutes (50 hours).

Someone might be interested in this time period if, for example, they were doing a study on customer engagement or a promotional campaign that took place in March 2020.

From a business perspective, a company might be interested in this SQL query to understand the behavior of new customers, to evaluate the success of a marketing campaign, or to assess the effectiveness of their content in engaging new subscribers. The 50 hours threshold might be a key performance indicator (KPI) for customer engagement in the early stages of subscription.

<p>Prompt:</p> <p>Explain what the below SQL query does. Also answer why someone might be interested in this time period, and why a company might be interested in this SQL query.</p> <p>Details: Temperature = 0.8; MaxTokens = 150+; Stop sequences: #</p> <pre>SELECT c.customer_id FROM Customers c JOIN Streaming s ON c.customer_id = s.customer_id WHERE c.signup_date BETWEEN '2020-03-01' AND '2020-03-31' AND s.watch_date BETWEEN c.signup_date AND DATE_ADD(c.signup_date, INTERVAL 30 DAY) GROUP BY c.customer_id HAVING SUM(s.watch_minutes) > 50 * 60</pre> <p>Explanation:</p>		
---	--	--

3. Prompt Engineering

Narrative	Screenshot	Notes
<p>Still in the chat playground, on the left side under “Setup”, you can use the Assistant setup to select a few pre-loaded System message examples to get started.</p> <p>System messages give the model instructions about how it should behave and any context it should reference when generating a response. You can describe the assistant's personality, tell it what it should and shouldn't answer, and tell it how to format responses.</p> <p>Few-shot examples allows you to provide conversational examples that are used by the model for in-context learning.</p> <p>From the assistant setup drop-down, select Xbox customer support agent.</p>		
<p>You'll be prompted asking if you want to update the system message, select Continue.</p>		

Check the updates that were automatically performed in [System Message Field](#) and [Examples](#).

Assistant setup

System message ⓘ

You are an Xbox customer support agent whose primary goal is to help users with issues they are experiencing with their Xbox devices. You are friendly and concise. You

Examples ⓘ

Using examples

Add examples to show the chat what responses you want. It will try to mimic any responses you add here so make sure they match the rules you laid out in the system message.

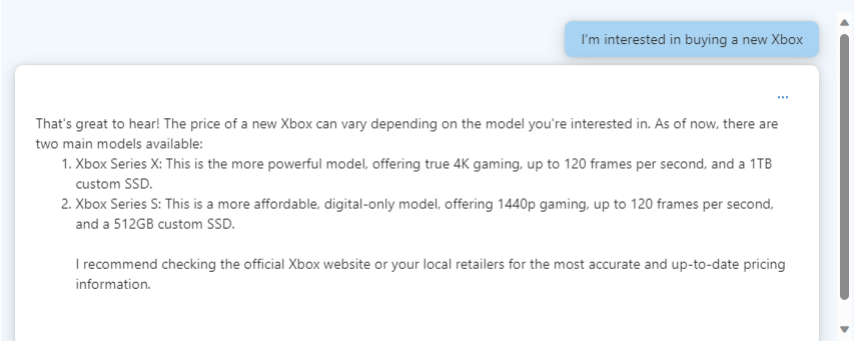
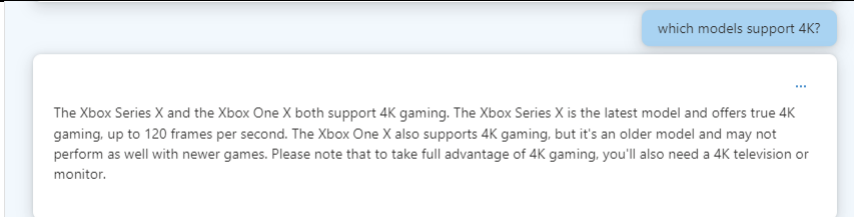
User:

How much is a PS5?

Assistant:

I apologize, but I do not have information about the prices of other gaming devices.

Azure OpenAI x Databases Hack

<p>In the chat session pane, enter the following question: "I'm interested in buying a new Xbox", and select Send.</p>	 <p>The screenshot shows a chat window with a light blue header. The user's input is in a blue bubble: "I'm interested in buying a new Xbox". The AI's response is in a white bubble with a blue border. It starts with "That's great to hear! The price of a new Xbox can vary depending on the model you're interested in. As of now, there are two main models available:" followed by a numbered list: <ol style="list-style-type: none"> 1. Xbox Series X: This is the more powerful model, offering true 4K gaming, up to 120 frames per second, and a 1TB custom SSD. 2. Xbox Series S: This is a more affordable, digital-only model, offering 1440p gaming, up to 120 frames per second, and a 512GB custom SSD. The response ends with "I recommend checking the official Xbox website or your local retailers for the most accurate and up-to-date pricing information." </p>
<p>Enter a follow-up question like: "which models support 4K?"</p>	 <p>The screenshot shows the same chat window. The user's input is in a blue bubble: "which models support 4K?". The AI's response is in a white bubble with a blue border. It says: "The Xbox Series X and the Xbox One X both support 4K gaming. The Xbox Series X is the latest model and offers true 4K gaming, up to 120 frames per second. The Xbox One X also supports 4K gaming, but it's an older model and may not perform as well with newer games. Please note that to take full advantage of 4K gaming, you'll also need a 4K television or monitor." </p>

Now that you have a basic conversation select **View code** setup and you'll have a replay of the code behind the entire conversation so far:

Chat playground

The screenshot displays the 'Chat playground' interface. On the left, the 'Assistant setup' panel is open, showing a system message: 'only provide factual answers to queries, and do not provide answers that are not related'. Below this, an 'Examples' section contains a conversation snippet where a user asks 'How much is a PS5?' and the assistant responds with a question about Xbox issues. The main chat area on the right shows a previous assistant response about Xbox pricing and a new user query: 'which models support 4K?'. At the top of the chat area, the 'View code' button is highlighted with a red box. Other buttons like 'Replay chat', 'Clear chat', and 'Playground Settings' are also visible.

Clear Chat and select **Default** as a System Message template.

In the next exercises, you will practice prompt engineering. For each task, you'll get some input text and then an expected completion. Your task is to write the prompt to achieve the expected completion.

Narrative	Screenshot	Notes
<p>Staying in the Default System message template. Now write a prompt that generates the expected completion.</p> <p>Input text: I was enjoying the sun, but then a huge cloud came and covered the sky.</p> <p>Expected completion: Ich genoss die Sonne, aber dann kam eine riesige Wolke und bedeckte den Himmel.</p>		
<p>Write a prompt that generates the expected completion.</p> <p>Input text: I was enjoying the sun, but then a huge cloud came and covered the sky.</p> <p>Expected completion: I was not enjoying the sun, and then a huge cloud did not come and cover the sky.</p>		
<p>Write a prompt that generates the expected completion.</p> <p>Input text: Not much to write about here, but it does exactly what it's supposed to. filters out the pop sounds. now my</p>		

Azure OpenAI x Databases Hack

<p>recordings are much more crisp. it is one of the lowest prices pop filters on amazon so might as well buy it, they honestly work the same despite their pricing.</p> <p>Expected completion (or similar): Positive: 0.75 Neutral: 0.20 Negative: 0.05</p>		
<p>Write a prompt that generates the expected completion</p> <p>Input text: I was enjoying the sun, but then a huge cloud came and covered the sky.</p> <p>Expected completion:</p> <pre>{ "translated": "Ich genoss die Sonne, aber dann kam eine riesige Wolke und bedeckte den Himmel.", "negated": "I was not enjoying the sun, and no huge cloud came and covered the sky.", "third_person": "She was enjoying the sun, but then a huge cloud came and covered the sky." }</pre>		
<p>Write a prompt that generates the expected completion</p> <p>Input text: Hello, my name is Mateo Gomez. I lost my Credit card on August 17th, and I</p>		

<p>would like to request its cancellation. The last purchase I made was of a Chicken parmigiana dish at Contoso Restaurant, located near the Hollywood Museum, for \$40.</p> <p>Below is my personal information for validation:</p> <p>Profession: Accountant Social Security number is 123-45-6789 Date of birth: 9-9-1989 Phone number: 949-555-0110 Personal address: 1234 Hollywood Boulevard Los Angeles CA Linked email account: mateo@contosorestaurant.com Swift code: CHASUS33XXX</p> <p>Expected completion:</p> <pre>{ "reason": "Lost card", "classified_reason": "lost_card", "name": "Mateo Gomez", "ssn": "123-45-6789", "dob": "09/09/1989" }</pre>		
---	--	--