Informatica® Cloud Application Integration
September 2024

Azure OpenAI Prompt Chaining

Informatica Cloud Application Integration Azure OpenAI Prompt Chaining
September 2024

Publication Date: 2024-09-30

Table of Contents

Preface

Use Azure OpenAI Prompt Chaining to learn how to design prompt chains and provide the desired responses from a Large Language Model (LLM). This guide assumes that you have an understanding of the Azure OpenAI Connector concepts.

CHAPTER 1

Introduction to Azure OpenAI Prompt Chaining recipe

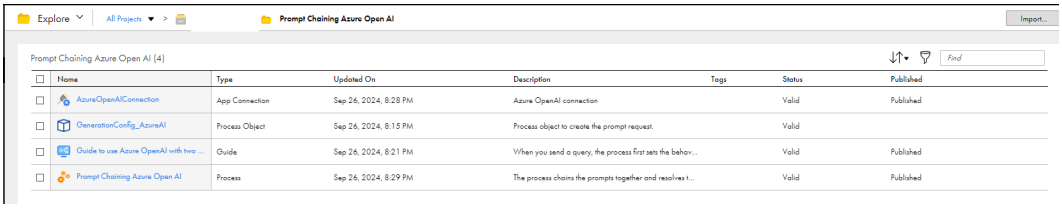
The Azure OpenAI Prompt Chaining recipe is initiated using a guide or an HTTP request.

The process chains the prompts together and resolves them in the sequence in which they're provided. Chaining of these prompts augments the ability of the language model being used to deliver a highly curated response. When you send a query, the process first sets the behavior and topic, and then provides the additional instructions to answer the query. After receiving the response, you can ask a new question without changing the topic.

Azure OpenAI Prompt Chaining recipe contents

The Azure OpenAI Prompt Chaining recipe contains multiple assets, such as a process object, an app connection, a guide, and a process.

The following image shows the assets that the Azure OpenAI Prompt Chaining recipe package contains:



Azure OpenAI Prompt Chaining recipe assets

The following table lists the assets that the Azure OpenAI Prompt Chaining recipe package contains:

Asset Name	Asset Type	Description
AzureOpenAIConnection	App connection	Azure OpenAI connection
GenerationConfig_AzureAI	Process object	Settings for the prompt request

Asset Name	Asset Type	Description
Guide to use Azure OpenAI with two requests	Guide	When you send a query, the process first sets the behavior and topic, and then provides the additional instructions to answer the query. After receiving the response, you can ask a new question without changing the topic.
Prompt Chaining Azure OpenAI	Process	The process chains the prompts together and resolves them in the sequence in which they're provided. Chaining of these prompts augments the ability of the language model being used to deliver a highly curated response.

CHAPTER 2

Using the Azure OpenAI Prompt Chaining recipe

To use the Azure OpenAI Prompt Chaining recipe, you must perform the following steps manually:

Step 1: Copy and access the recipe

Step 2: Configure and publish the AzureOpenAI connection

Step 3: Configure and publish the process

Step 4: Publish and run the guide

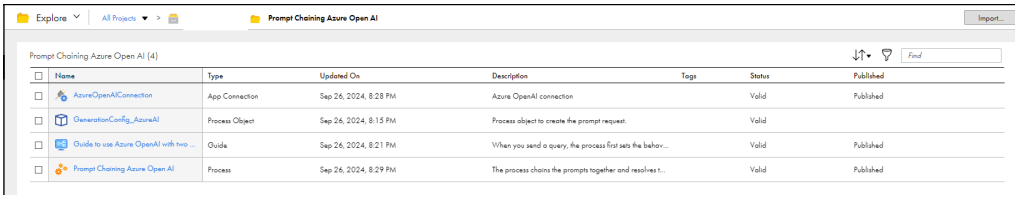
Step 1: Copy and access the recipe

Copy the pre-configured assets in the recipe to a separate project or folder.

1. Open the **Azure OpenAI Prompt Chaining** recipe and click **Use**.
2. Select the location where you want to copy the recipe, and then click **Continue**.
3. In the **Copying the recipe** dialog box, click **OK**.

   It might take some time for the recipe to get copied. You will receive a notification when the recipe is ready for use.
4. After the recipe is copied, click **Explore** to access the recipe content.
5. Navigate to the project or folder where you copied the recipe or enter the recipe name in the **Find** box.

   All the assets in the recipe are displayed as shown in the following image:



Step 2: Configure and publish the AzureOpenAI connection

Configure the endpoint URL and API key in the AzureOpenAI connection, and then publish the connection.

1. Open the **AzureOpenAI** connection.
2. In the **Type** field, select **AzureOpenAI**.
3. In the **Run On** field, select **Cloud Server or any Secure Agent**.
4. In the **Connection Properties** section, enter values for the following properties:

Property	Description
Endpoint_URL	The REST API endpoint for Azure OpenAI. You can find this value in the **Keys & Endpoint** section when examining your resource from the Azure portal. Alternatively, you can find the value in **Azure OpenAI Studio** > **Playground** > **Code View**.
API_Key	The API key to authenticate Azure OpenAI connection requests. You can find this value in the **Keys & Endpoint** section when examining your resource from the Azure portal. Alternatively, you can find the value in **Azure OpenAI Studio** > **Playground** > **Code View**.

5. Save, test, and publish the connection.

Step 3: Configure and publish the process

Configure the deployment details of the LLM model and publish the processes.

1. Open the **Prompt Chaining Azure OpenAI** process.
2. On the **Temp Fields** tab of the **Start** step, enter values for the following fields:
   - In the **api_version** field, enter the API version of the LLM model. Default is **2024-06-01**. You can optionally edit the api version.
   - In the **deployment_id** field, enter the user-specific deployment ID.
3. Optionally, in the **Configure Request Parameters** step, configure the prompt instructions in the **Assignments** field by updating the **Prompt_Configuration** field using the Expression Editor, as shown in the following sample code:

```
<GenerationConfig_AzureAI>
     <topP>1</topP>
     <max_tokens>500</max_tokens>
     <temperature>0.5</temperature>
  </GenerationConfig_AzureAI>
```

For the **Prompt_Configuration** field, enter values for the following properties:

Property	Description
max_tokens	Defines the maximum number of tokens that the model can generate in its response. Setting a limit ensures that the response is concise and fits within the desired length constraints.
temperature	Controls the randomness of the model's output. A lower value makes the output more deterministic, while a higher value increases randomness and creativity. For example, a temperature of 0.5 balances between deterministic and creative outputs.
topP	Determines the cumulative probability threshold for token selection. The model considers the smallest set of tokens whose cumulative probability meets or exceeds topP. For example, if topP is set to 0.1, the model considers only the top 10% most probable tokens at each step.

4. In the **Create Prompt 1** step, enter the prompt instructions in the **Assignments** field by updating the **Prompt_Request** field using the Expression Editor as shown in the following sample code:

```
<CreateChatCompletionRequest>
    <temperature>{$temp.Prompt_Configuration[1]/temperature }</temperature>
   <top_p>{$temp.Prompt_Configuration[1]/top_p }</top_p>
   <max_tokens>{$temp.Prompt_Configuration[1]/max_tokens }</max_tokens>
     <messages>
      <role>system</role>
      <content>{$input.First_System_Prompt  } </content>
    </messages>
    <messages>
      <role>user</role>
      <content>{$input.First_User_Prompt  }</content>
    </messages>
</CreateChatCompletionRequest>
```

After configuring the prompt instructions, the process sends the details to the LLM to fetch the required response, and then stores the first response.

5. In the **Create Prompt 2** step, in the **Assignments** field, update the **Prompt_Request** field using the Expression Editor as shown in the following sample code:

```
<CreateChatCompletionRequest>
    <temperature>{$temp.Prompt_Configuration[1]/temperature }</temperature>
    <top_p>{$temp.Prompt_Configuration[1]/top_p }</top_p>
    <max_tokens>{$temp.Prompt_Configuration[1]/max_tokens }</max_tokens>
       <messages>
         <role>system</role>
         <content> {$input.First_System_Prompt} </content>
    </messages>
     <messages>
      <role>user</role>
      <content>{ $input.First_User_Prompt  }</content>
    </messages>
     <messages>
      <role>assistant</role>
      <content>{ $temp.Prompt_Response[1]/choices[1]/message[1]/content    }</
content>
    </messages>
    <messages>
      <role>user</role>
      <content>{$input.Second_User_Prompt  }</content>
    </messages>
</CreateChatCompletionRequest>
```

The LLM uses both the requests as an instruction to prepare the final response.

6. Save and publish the process.

Step 4: Publish and run the guide

When you send a query to the LLM, the process first sets the behavior and topic, and then provides the additional instructions to answer the query.

To publish and run the guide, perform the following steps:

1. Open the **Guide to use Azure OpenAI with two requests** guide.
2. On the **Start** tab of the **Start** step, ensure that the **Run As** field is set to **Current User**.
3. Save and run the guide.
4. On the **Actions** menu, click **Run**. Alternatively, you can copy the execution URL from the **Properties Details** dialog box to run the guide.
5. On the **Instructions** page, enter the requests in the **Set AI Behavior** and **First User Prompt** fields. The LLM uses these requests as instructions to prepare the final response to your query.
6. Click **Continue**.
7. On the next screen, enter any additional instruction for your request and ask your query in the **Second User Prompt** field.

   Note: You can't edit the behavior and topic fields on this page.
8. Click **Continue**.
   The final response appears.
9. Click **New question** to ask another query, or click **End** to finish.

You can also use the embed code to embed the guide into an HTML document of a third-party application.