Informatica® Cloud Application Integration
September 2024

Azure OpenAI Chat with File using Guide

Informatica Cloud Application Integration Azure OpenAI Chat with File using Guide
September 2024

Publication Date: 2024-12-06

Table of Contents

Preface

Use *Azure OpenAI Chat with File using Guide* to learn how to ask questions to the Azure OpenAI Large Language Model (LLM) based on an uploaded file's contents using a guide. This guide assumes that you have an understanding of the Azure OpenAI Connector concepts.

CHAPTER 1

Introduction to Azure OpenAI Chat with File using Guide recipe

The Azure OpenAI Chat with File using Guide recipe is based on REST and SOAP APIs. You can ask questions to the Azure OpenAI Large Language Model (LLM) based on an uploaded file's contents.

The recipe allows you to upload a file with context, provide a user prompt, read the text from the file, and answer the user's questions based on the contents of the file using the Azure OpenAI Large Language Model (LLM).

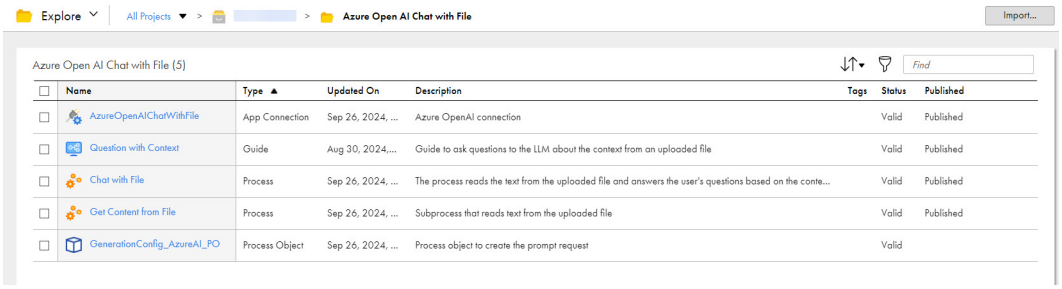
You can upload files of the following formats:

`.json`, `.js`, `.txt`

Azure OpenAI Chat with File using Guide recipe contents

The Azure OpenAI Chat with File using Guide recipe contains a process object, an app connection, a guide, and two processes.

The following image shows the assets that the Azure OpenAI Chat with File using Guide recipe package contains:



Azure OpenAI Chat with File using Guide recipe assets

The following table lists the assets that the Azure OpenAI Chat with File using Guide recipe package contains:

Asset Name	Asset Type	Description
GenerationConfig_AzureAI_PO	Process object	Creates the prompt request.
AzureOpenAIChatWithFile	App connection	Azure OpenAI connection.
Get Content from File	Process	Subprocess that reads the text from the uploaded file.
Chat with File	Process	Process that reads the text from the uploaded file and answers the user's questions based on the contents of the file using the Azure OpenAI Large Language Model (LLM).
Question with Context	Guide	Guide that allows users to ask questions to the LLM about the context from an uploaded file.

CHAPTER 2

Using the Azure OpenAI Chat with File using Guide recipe

To use the Azure OpenAI Chat with File using Guide recipe, you must perform the following steps manually:

Step 1: Copy and access the recipe

Step 2: Configure and publish the Azure OpenAI connection

Step 3: Configure and publish the processes

Step 4: Publish and run the guide

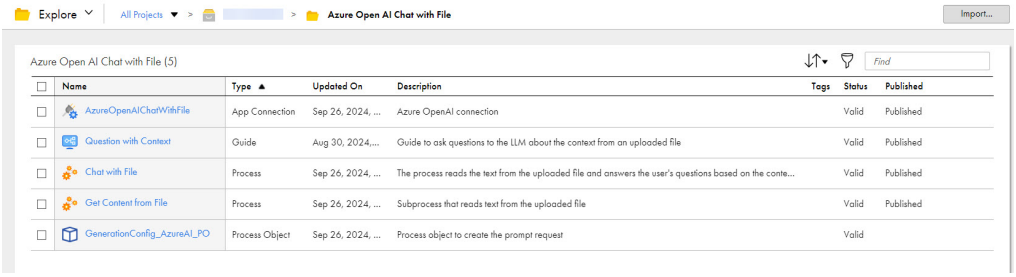
Step 1: Copy and access the recipe

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

1. Open the **Azure OpenAI Chat with File using Guide** 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 Azure OpenAI connection

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

1. Open the **AzureOpenAIChatWithFile** connection.
2. 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**.

3. Save and publish the connection.

Step 3: Configure and publish the processes

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

1. To publish the **Get Content from File** process, click **Actions** in the row that contains the process and select **Publish**.
2. Open the **Chat with File** process.
3. On the **Temp Fields** tab of the Start step, enter values for the following fields:
   - In the **deployment_id** field, enter the deployment ID of the LLM model.
   - In the **api_version** field, enter the API version of the LLM model.
4. Optionally, in the **Configure Request Parameters** step, enter the prompt instructions in the **Assignments** field by updating the **Prompt_Configuration** field using the Expression Editor, as shown in the following sample code:

```
For Prompt_Configuration:
 <GenerationConfig_AzureAI>
   <temperature>0.5</temperature>
   <topP>1</topP>
   <max_tokens>1000</max_tokens>
</GenerationConfig_AzureAI>
```

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

Property	Description
temperature	Controls the randomness of the model's output. A lower value close to 0 makes the output more deterministic, while a higher value close to 1 increases randomness and creativity. For example, if `temperature` is set to 0.5, the model 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.
max_tokens	Defines the token count of your prompt. The max_tokens can't exceed the model's context length. Most of the models have a context length of 2048 tokens.

5. Optionally, in the **Prepare Request** step, enter the prompt instructions in the **Assignments** field by updating the **Request** field using the Expression Editor, as shown in the following sample code:

```
For Request:
<CreateChatCompletionRequest>
 <temperature>{$temp.Prompt_Configuration[1]/temperature  }</temperature>
   <top_p>{$temp.Prompt_Configuration[1]/top_p }</top_p>
   <max_tokens>{$temp.Prompt_Configuration[1]/max_tokens }</max_tokens>
<messages>
    <role>user</role>
    <content>You are a helpful assistant.{$temp.Content_From_File}</content>
</messages>
<messages>
    <role>user</role>
    <content>{$input.User_Prompt}</content>
</messages>
</CreateChatCompletionRequest>
```

6. Save and publish the process.

Step 4: Publish and run the guide

You can ask questions to the LLM about the context from an uploaded file. You can use files of the following formats:

`.json, .js, .txt`

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

1. Open the **Question with Context** guide.
2. On the **Start** tab of the Start step, ensure that the **Run As** field is set to **Current User**.
3. Save and publish 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, select the context file that you want to upload from your machine.
6. In the **Enter your query below** field, enter a question based on the file contents.
7. Click **Continue**.
   You will receive a response from the LLM.
8. To ask the next question, click **Next Question**.

9. Enter new questions with the same context and click **Continue**.

Note: If the uploaded file is empty or the question is not related to the file context, an error occurs.

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