GenAI Tool: Data Formatter

Connected System Plugin for Appian
Table of Contents

Overview 3
Features 3
Connected System Configuration 4
  Chat Completion Model: OpenAI 4
  Chat Completion Model: Azure OpenAI 7
Integration 11
  1. Record Field Input 11
  2. Text Input Type 12
Data Formatter Sample App Setup 13
Overview

The GenAI Tool: Document Formatter Connected System allows developers to manipulate inputted data to fit a format specified by the user in simple, natural language.

After entering either OpenAI or AzureOpenAI credentials, developers are able to format the values of a specific field of a record type or the values of a passed in array. This documentation outlines the process of obtaining and leveraging these credentials within the Appian platform. It also gives additional instructions on how to configure the integrations and the sample app which is downloadable from the Appian AppMarket alongside the plug-in and documentation.

Note: This AI Tool can utilize external LLMs such as OpenAI and Azure OpenAI. As with all plug-ins, data sent to these external organizations is subject to the external organization’s security policies, procedures, and pricing. Please check with your administrator and your organization’s AI policy before sending any sensitive information to external services.

Features

- Format the values of a record type’s field.

- Format the values of an array of data.
**Connected System Configuration**

**Chat Completion Model: OpenAI**

### Connected System Properties

**GenAI Tool: Data Generator**
Generate sample Record data with ChatGPT. The RecordType should have at least one row for the plugin to generate data.
Version: 1

**Name**
DGS CS Data Generation

**Description**
Connected System for Data Generator

### GenAI Tool: Data Generator Configuration

#### Authentication
OpenAI Services

Use the OpenAI services for Chat Completion

**OpenAI API Key**

```
............................
```

Enter your OpenAI API Key. Visit https://beta.openai.com/account/api-keys to get an API key for your account.

**Completion Model**

gpt-3.5-turbo

Provide the name of the model to use for text completion. Example: gpt-3.5-turbo for GPT 3.5 Turbo model, gpt-4 for GPT 4 model. gpt-4 is the most consistent model in determining the size of output while gpt-3.5-turbo is faster than gpt-4. Visit https://platform.openai.com/docs/models/model-endpoint-compatibility and use one of the models listed under V1/chat/completions endpoint.

Connection successful
The Document Summarization Connected system with OpenAI authentication requires the following credentials: **OpenAI API Key** and **Chat Completion model**.

To retrieve your OpenAI API Key:

1. Go to the [OpenAI console](https://console.openai.com/). Make sure that the **API keys** menu is selected.

   ![OpenAI API keys](https://console.openai.com/

   2. Click on **Create new secret key** to generate a new API key.

   3. Copy the value and save it separately as we won’t be able to access it again. Paste the API key in the connected system dialog box.

   ![Create new secret key](https://console.openai.com/)

To find the appropriate Chat Completions model:
1. Visit https://platform.openai.com/docs/models/model-endpoint-compatibility and use one of the models listed under /v1/chat/completions endpoint. Example: gpt-3.5-turbo for GPT 3.5 Turbo model, gpt-4 for GPT 4 model.
   
a. Each model has unique strengths so try to select the most appropriate for your use. If you would like to prioritize consistency in the size and format of your generated summary, we recommend you use a GPT 4 model. If you need to prioritize speed of generation, GPT 3.5 Turbo might be better suited.
This authentication requires the following credentials: Azure Region, Azure Deployment ID and Azure API Key. Follow these steps to get the Azure credentials.
Set up your Azure OpenAI Account

1. Navigate to Azure’s OpenAI API docs and ensure you have met the listed prerequisites. View the prerequisites by selecting “Quickstarts.” If you have not already done so, create an Azure subscription.

2. Apply for access to Azure OpenAI services by completing the form here. You will need your subscription ID from the previous step.

3. Create a service and set your domain name.
Create and Access API Keys
4. Within your service, create and access API keys through "Keys and Endpoints" under Resource Management. The “Location/Region” listed in this window will be used as your Azure Region in the Connected System configuration.

Deploy OpenAI Models and Set your Deployment ID
5. Visit the Azure OpenAI Studio to deploy OpenAI models under your created resource.
6. Click into your resource to enter the Azure OpenAI Studio. Navigate to “Models” seen as a tab on the left side of the screen. After selecting the best OpenAI Model for your data and use case, deploy your selected chat completions model. The deployment ID you create during deployment will be used when configuring your Connected System.
**Integration**

Returns a list of formatted data from a list of text input or a Record Field.

ChatGPT Prompt used:
“You are a highly intelligent Data formatting assistant that formats the given data based on the given description. The input is in the format of Array of String. The output result should be an Array of Formatted Text. Description: `<Format Description>`. Format the input `<Input List>`:

1. Record Field Input

**Inputs:**

**Record Field Values** (Text) - Required - Values of the record type field to be formatted. Use the `getrecordfieldvalues()` function to query and input the data you would like to format. `getrecordfieldvalues()` receives a record type, a record field, a start index, and a batch size. The `recordType` must be wrapped in a `toxml()` function, and the record field must be wrapped in a `tostring()` function. The start index input represents the index of the data from which you would like to begin querying and formatting. The batch size represents how many items you would like to format after the indicated start index.

Example:

```java
getrecordfieldvalues(
    toxml(recordType!{recordTypeName}),
    tostring(recordType!{recordType}.fields.{fieldName}),
    1,
    10
)
```

**Format Description** (Text) – Required - Describe how the data should be formatted.

Example: "Numbers rounded to two decimal places. Return as a string."
Output: Dictionary

{result: {"01-10-23", "02-10-23", "03-10-23"}}

2. Text Input Type

Inputs:

**Input List** (List of Text) - Required - Provide the items you want to format as a list of text items. Example: 

**Format Description** (Text) – Required - Describe how the data should be formatted. Example: 
"Date formatted as dd-mm-yyyy"
Output: Dictionary

{result: {25.49, 37.49}}

Data Formatter Sample App Setup

To configure the sample app to begin experimenting with the GenAI Tool: Data Formatter, complete the following steps.

1. **Download the necessary files**: Find the Data Formatter Connected System from Appian’s App Market. Select download and open the zip file to access the Sample Application (.zip), an SQL script (.sql), a Sample App Properties file (.properties), and the plug-in jar file (.jar).

2. **Import the sample data**: In your Appian Cloud Database, select the database where you would like to create a new table for the sample record data. Select Import and
“Choose File.” Select the SQL file from the downloaded package and hit the “Import” button.

3. **Configure the properties file:** In the properties file downloaded from the AppMarket, add your API key to either the OpenAI or Azure OpenAI API key variable, depending on your preference and your models available.

4. **Import the sample application:** In your Appian designer, select the import button and upload the downloaded sample app zip file and the configured properties file.

5. **Finish configuring sample data:** In each of the DFS Sample Record, configure the data source by selecting “Change Data Source” in the top right of Data Model view. Connect to the table in your database that was created with the SQL script in step #2.

6. **Explore the tool:** You can now freely experiment with the Data Formatter tool from the sample application. Both input types are available for use from the interface of the sample application.
   a. Record Field input type: can query and manipulate the data from the Sample Record record type
   b. List of Text input type: can input the data items to be manipulated, separating each item by a new line in the text field