Date :  October 12th, 2023

Openai ADAPTER (1.0.0)

# Overview

The **OpenAI adapter** for HCL Volt Foundry’s Integration service integrates Volt Foundry with the very popular OpenAI platform and its operations for intelligent text generation, conversation completion, image generation, and text moderation. This adapter allows developers to add AI assistant very quickly to Volt MX middleware services and to Volt Iris multi-channel apps to enable services such as chat-bot attendant, FAQ, or the smart comparison of competitive products by querying this AI.

An example use case for developing Foundry services using this **OpenAI adapter** is a multi-channel app which allows the end-user to submit a question, needing an answer for a particular topic, have that question be checked for moderation (offensive language), generate an intelligent answer to the question, and to generate an icon to associate with the question/answer pair. The combination of the above services will enable a rich and adaptive FAQ that a business can use to help its users or customers.

A screenshot of a chat

Description automatically generated

The **OpenAI adapter** supports the following operations:

1. **Create Chat Completions**, https://api.openai.com/v1/chat/completions - useful to complete a conversation between a user and the AI assistant.
2. **Create Completions**, https://api.openai.com/v1/completions - useful to answer a question, to generate texts on a topic, or to generate a snippet of code in a particular programming language.
3. **Create Image,** https://api.openai.com/v1/images/generations - useful to generate an image (copyright free) or an icon based on the user provided prompt.
4. **Create Moderations**, https://api.openai.com/v1/moderations - useful to check if the input texts violate the specified OpenAI content policy.
5. **List Models**, https://api.openai.com/v1/models - Lists the currently available OpenAI models, and provides basic information about each one such as the owner and availability.
6. **Retrieve Models**, https://api.openai.com/v1/models/<model-name> - Retrieves a model instance, providing basic information about the model such as the owner and permission.
7. **Delete fine-tune model**, https://api.openai.com/v1/models/<fine-tune-model-name> - Delete a fine-tuned model. You must have the Owner role in your organization to delete a model.

For more detailed information, please follow this URL to view the OpenAI API reference: https://platform.openai.com/docs/api-reference

# Getting Started

## Prerequisites

* Volt Foundry
* OpenAI API key is required in the input header of each of the OpenAI adapter operations (https://platform.openai.com/account/api-keys)

## Third Party Dependency

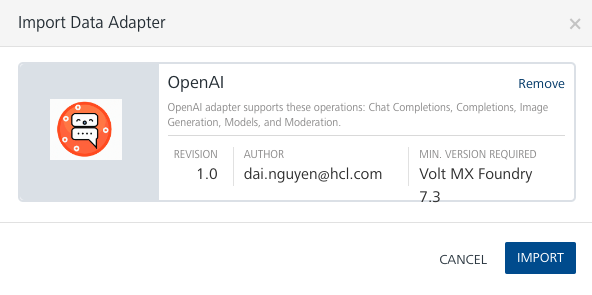
## At the time of this **OpenAI adapter** creation (10/12/2023), OpenAI API is at version 1. The information provided herein is dependent on OpenAI permission for its API usage and thusly is subject to change by OpenAI.

## Importing the adapter

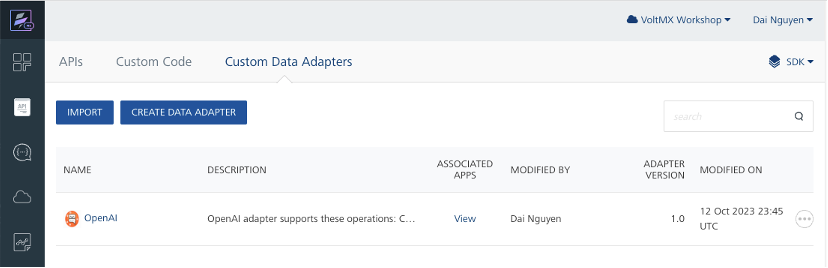
**To import the Data Adapter to Volt Foundry, do the following:**

1. Sign in to the  [HCL Foundry](https://manage.hclvoltmx.com/).
2. From the left navigation menu, select **API Management**.
3. In **API Management**, select **Custom Data Adapters**.  
   
4. Click **IMPORT** to import a custom data adapter.  
   
5. On the Import Data Adapter dialog box, click browser to import.  
   
6. Select OpenAI.zip zip file and click **IMPORT**.

After you import the data adapter, Volt Foundry opens a window that shows the metadata of the data adapter.



After you import the data adapter, you can view it on the Custom Data Adapters page and use it to create services on Volt Foundry.



## Creating an Integration service

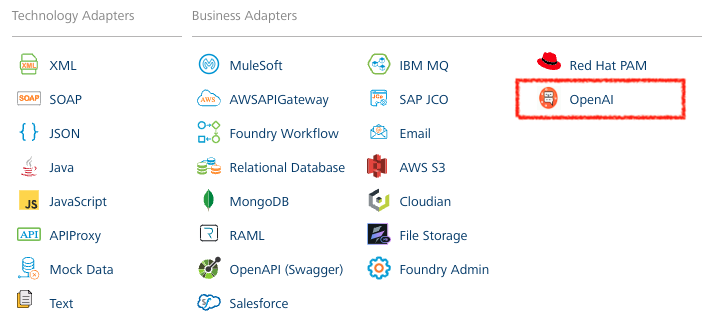
After you import the data adapter into Volt Foundry, you can use it to create an Integration Service.

Follow the given steps to create an Integration service using the OpenAI Adapter.

1. Sign in to the [HCL Foundry](https://manage.hclvoltmx.com/).
2. From the left navigation menu, select **API Management**.
3. In **API Management**, select **Integration**.



1. To create a new service, click the **+** button or the **CONFIGURE NEW** button.  
   
2. On the Service Definition tab, select the service type as OpenAI, and click **SAVE**.



Alternatively, you can also create a Foundry app and create an Integration service inside it.

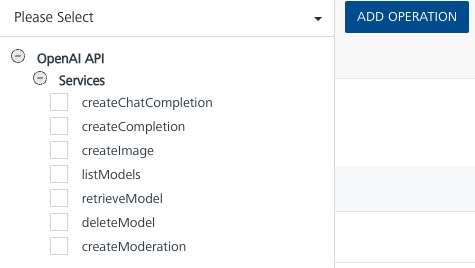
E. Creating and Executing operations



After you create an integration service, you can create and execute operations using the service.

#### Creating an Operation

* In **API Management/Foundry app you created**, in the **Integration** section, select the service that you created.
* After you select the service, navigate to the **Operation List** tab.  
  
* From the drop down list, select an operation that you want to execute, and click **ADD OPERATION**.



#### Executing an Operation

* From the **Operations List** tab, in the **Configured Operations** section, click on the operation you want to execute, for example **createChatCompletion**.

A screenshot of a chat application

Description automatically generated

* In the initial display of the selected operation in the Operation Page, the Request Input tab will list all the input parameters available to an operation. **You may delete the input parameters you do not plan to offer or support**. See below and note the scroll bar:

A screenshot of a computer

Description automatically generated

* On the Operation Page, in the Request Input tab, enter all the needed test values to execute the operation. For example, **createChatCompletion** operation will need: 1) messages collection which contains the role and the content properties, and 2) the model. See below:

A screenshot of a computer

Description automatically generated

* You must provide the OpenAI API Key from your OpenAI’s account. The key must be inserted into **each operation’s input header**. Set the key into the input header’s Authorization parameter, as a “**constant**” bearer-token value. See below:

A screenshot of a computer

Description automatically generated

* Select a run-time environment and click **Save and Fetch Response** to get a response based on your inputs.  
  A screenshot of a computer

  Description automatically generated

## Publishing your application

If you want to use the services in client applications, you need to publish an app to a run-time environment. You can create the service (as described above) in an application or import the service into an application and publish the application.

# Example input and output of each supported openai operations

## createChatComplete operation

### Example of the initial Request Input parameters to test:

### Note that “messages” is an array/collection. For the initial test in Foundry, where you provide the input parameters and click “Save and Fetch Response” button, you can only provide 1 set of **role and content**. The example below shows the role of “system” with content as “You are a beauty consultant”. See below:

### 

### Example execution and Output Result:

### A screenshot of a computer program Description automatically generated

### Example of a retest of Request Input parameters:

### After the first test, shown above, you can alter the input parameters and click the green “play” icon, at the top right of the “Request Payload” panel to retest.

### Note that “messages” is an array/collection in which you can insert a series of conversation between different roles, “system, “assistant”, and “user” to retest.

### 

## createComplete operation

### Example of the initial Request Input parameters to test:

## A screenshot of a test Description automatically generated

### Example execution and Output Result:

## A screenshot of a computer code Description automatically generated

## createImage operation

### Example of the initial Request Input parameters to test:

## 

### Example execution and Output Result:

## 

## listModels operation

### No Input parameter is needed to list the models. Click the **Save and Fetch Response** button to test.

### Example execution and Output Result:

## A screenshot of a computer Description automatically generated

## retrieveModel operation

### Example of the initial Request Input parameters to test:

### A screenshot of a computer Description automatically generated

### Example execution and Output Result:

## A screenshot of a computer Description automatically generated

## deleteModel operation

### Example of the initial Request Input parameters to test:

### 

### Example execution and Output Result:

## 

## createModeration operation

### Example of the initial Request Input parameters to test:

### 

### Example first execution and Output Result:

### A screenshot of a computer Description automatically generated

### Example execution and Output Result of a retest where the input payload is changed:

### Note: “input” parameter, shown above, is a string. However, the operation will accept an array of strings as the input payload. Also note that the result of the moderation has changed because of the second input string “Then I will look for unicorns in the cloud”. See below:

## 

# References

## List of API Endpoints

1. **Create Chat Completions**, https://api.openai.com/v1/chat/completions.
2. **Create Completions**, https://api.openai.com/v1/.
3. **Create Image,** https://api.openai.com/v1/images/generations.
4. **Create Moderations**, https://api.openai.com/v1/moderations.
5. **List Models**, https://api.openai.com/v1/models.
6. **Retrieve Models**, https://api.openai.com/v1/models/<model-name>.

# Revision History

Adapter version 1.0.0:

## Known Issues

-List known issues

## Limitations

-List limitations