

# **Entity Extractor**

Connected System Plugin for **Appian**

# Appian Corporation

Version 1.0.0

## 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. Document Input Type	11
2. Text Input Type	12

## Overview

The Entity Extractor Connected System allows developers to extract specific entities with the help of OpenAI services. It allows users to extract all or specific entities from a text or an Appian document.

An entity refers to a specific piece of information that is categorized and extracted from a text. In the context of Natural Language Processing (NLP) and text analysis, entities are typically nouns that have a distinct and well-defined meaning. They often represent real-world objects, such as people, locations, organizations, dates, products, etc.

For example, in the sentence "Apple Inc. was founded by Steve Jobs in Cupertino on April 1, 1976," the entities might be:

"Apple Inc." (Organization)

"Steve Jobs" (Person)

"Cupertino" (Location)

"April 1, 1976" (Date)

Entity extraction, also known as Named Entity Recognition (NER), is the process of identifying and classifying these entities into predefined categories. It's a crucial task in many NLP applications, including information retrieval, question answering, and summarization.

Developers can leverage their entity extraction capabilities through Appian with this connected system by entering the credentials retrieved from either OpenAI or Azure OpenAI Studio. This documentation outlines the process of obtaining and leveraging these credentials within the Appian platform.

**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

- Extract user specified entities from a given text or document

# Connected System Configuration

## Chat Completion Model: OpenAI

### Connected System Properties



#### Entity Extractor

Extract entities from text using ChatGPT  
Version: 1

**Name \***

**Description**

#### Entity Extractor Configuration

**Authentication**

Use the OpenAI services for Chat Completion

**OpenAI API Key**

\*\*\*\*\* (Clear)

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

**Completion Model \***

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. Visit <https://platform.openai.com/docs/models/model-endpoint-compatibility> and use one of the models listed under /v1/chat/completions endpoint.

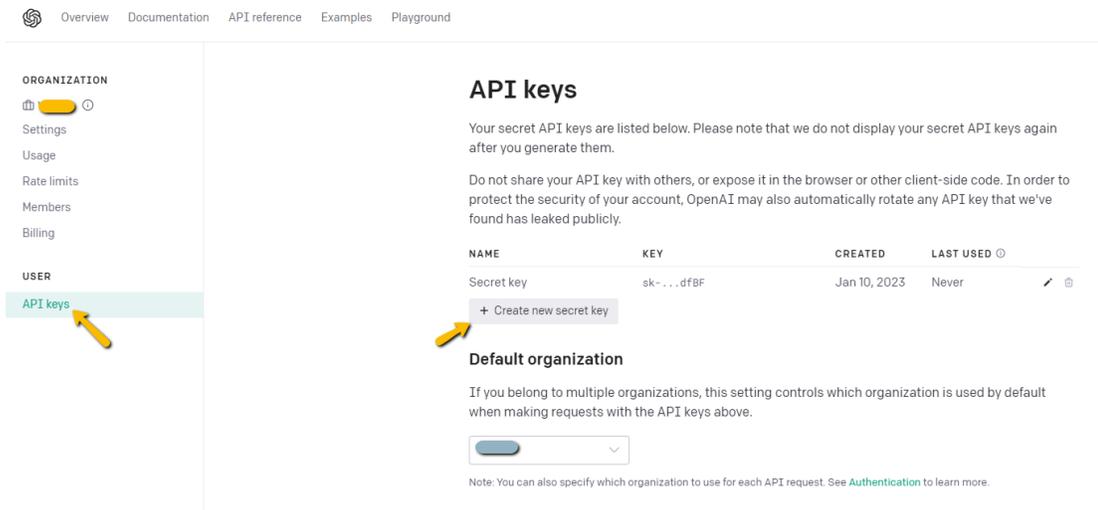
**TEST CONNECTION**

**CANCEL** **USE IN NEW INTEGRATION** **SAVE**

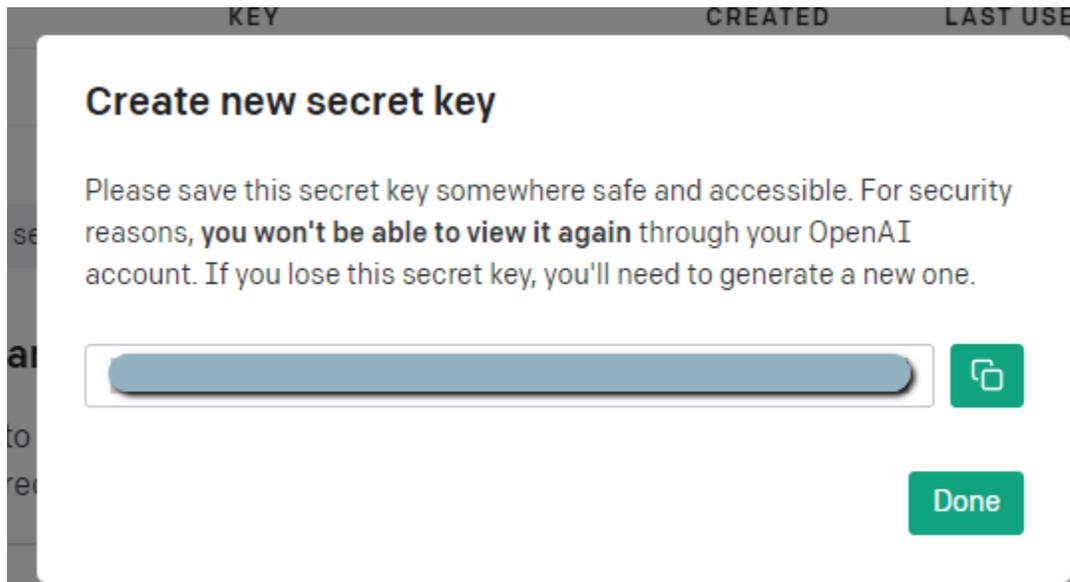
The Entity Extraction connected system with OpenAI authentication requires the following credentials: **OpenAI API Key** and **Chat Completion model**.

To retrieve your OpenAI API Key:

1. Go to the [OpenAI console](#). Make sure that the **API keys** menu is selected.



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.



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 model's behavior, we recommend you use a GPT 4 model. If you need to prioritize speed of generation, GPT 3.5 Turbo might be better suited.

## Chat Completion Model: Azure OpenAI

### Connected System Properties



#### Entity Extractor

Extract entities from text using ChatGPT  
Version: 1

#### Name \*

#### Description

#### Entity Extractor Configuration

##### Authentication

Use the Azure Open AI services for Chat Completion

##### Azure Region \*

Provide the Azure region.

##### Deployment ID \*

Provide the Deployment ID.

##### Azure API Key \*

Provide the API Key obtained from Azure OpenAI

Connection successful

TEST CONNECTION

CANCEL

USE IN NEW INTEGRATION

SAVE

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

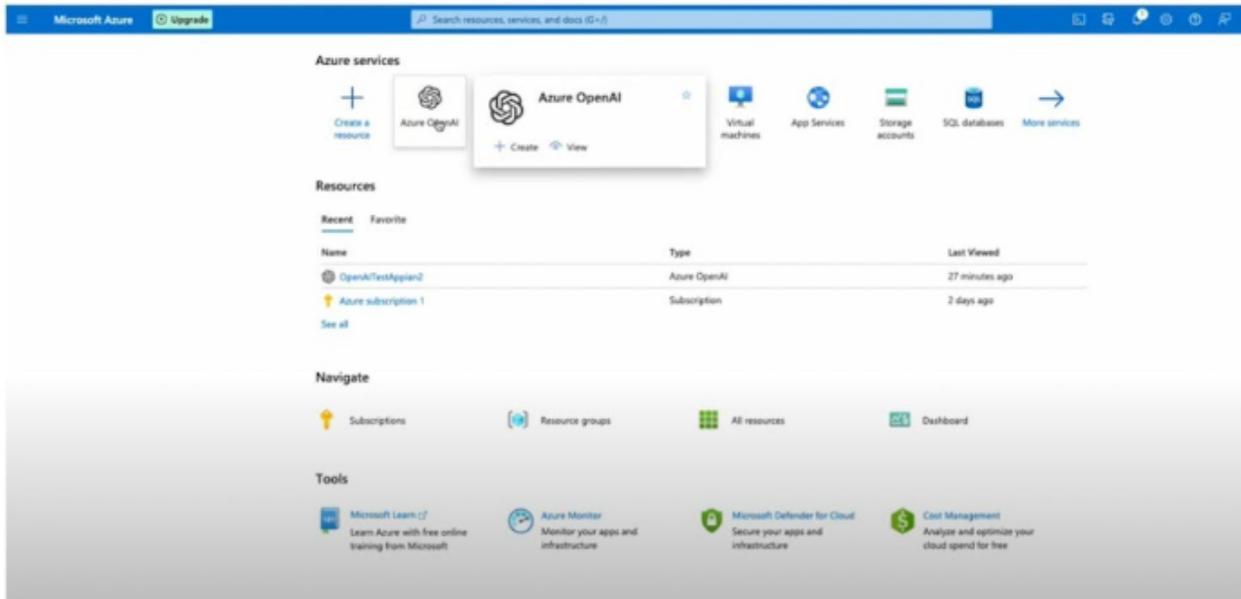
1. Navigate to [Azure's OpenAI API docs](#) and ensure you have met the listed prerequisites. View the prerequisites by selecting "Quickstarts." If you have not already done so, [create an Azure subscription](#).

The screenshot shows the Azure OpenAI Service Documentation page. At the top, there is a navigation bar with the Microsoft logo and links for Learn, Documentation, Training, Certifications, Q&A, Code Samples, Assessments, Shows, and Events. A search bar and a 'Sign in' link are also present. Below the navigation bar, there is a blue header with the text 'Azure OpenAI Service Documentation' and a sub-header 'Learn how to use Azure OpenAI's powerful language models including the GPT-3, Codex and Embeddings model series for content generation, summarization, semantic search, and natural language to code translation.' Below the header, there are eight tiles representing different sections: OVERVIEW (What is Azure OpenAI Service?), QUICKSTART (Quickstarts), HOW-TO GUIDE (Create a resource), TUTORIAL (Embeddings), HOW-TO GUIDE (Completions), TRAINING (Intro to Azure OpenAI training), CONCEPT (Azure OpenAI Models), and REFERENCE (Support and help options). Below these tiles, there is a section titled 'Additional resources' with four sub-sections: Azure OpenAI (links to Azure OpenAI Studio, Region support, Quotas and limits, and Apply for access to Azure OpenAI), Video (link to Combining OpenAI models with the power of Azure), Reference (links to REST API and Terms of use), and Tools (links to Azure CLI and PowerShell).

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

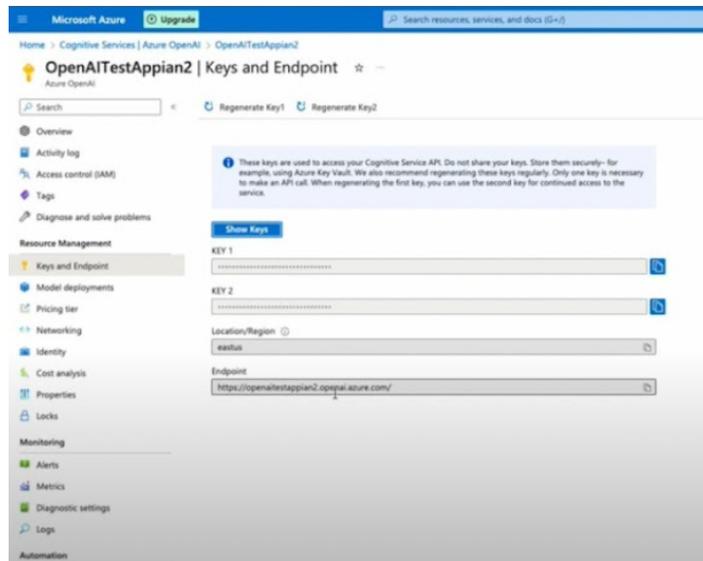
The screenshot shows the 'Request Access to Azure OpenAI Service' form. The title is 'Request Access to Azure OpenAI Service'. Below the title, there is a red asterisk and the word 'Required'. The main content of the form is a text box with the following text: 'Please read all instructions carefully and complete form as instructed. Thank you for your interest in Azure OpenAI Service. Please submit this form to register for approval to access and use Azure OpenAI's Limited Access text and code and/or DALL-E 2 text to image models (as indicated in the form). All use cases must be registered. Azure OpenAI Service requires registration and is currently only available to approved enterprise customers and partners. Learn more about limited access to Azure OpenAI Service here. Limited access scenarios: When evaluating which scenarios to onboard, we consider who will directly interact with the application, who will see the output of the application, whether the application will be used in a high-stakes domain (e.g., medical), and the extent to which the application's capabilities are tightly scoped. In general, applications in high stakes domains will require additional mitigations and are more likely to be approved for applications with internal-only users and internal-only audiences. Applications with broad possible users, including content generation capabilities, are more likely to be approved if 1) the domain is not high stakes and users are authenticated or 2) in the case of high stakes domains, anyone who views or interacts with the content is internal to your company. Please be sure to visit the Azure OpenAI Service's transparency note, which provides information and guidelines for responsible use of the service as well as system limitations that may be applicable to your scenario.'

3. Create a service and set your domain name.



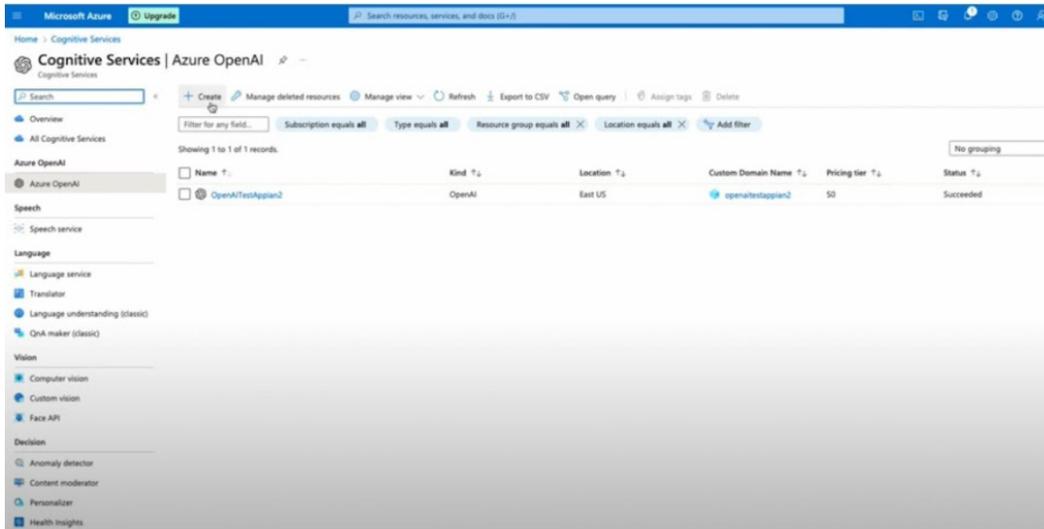
## Create and Access API Keys

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.

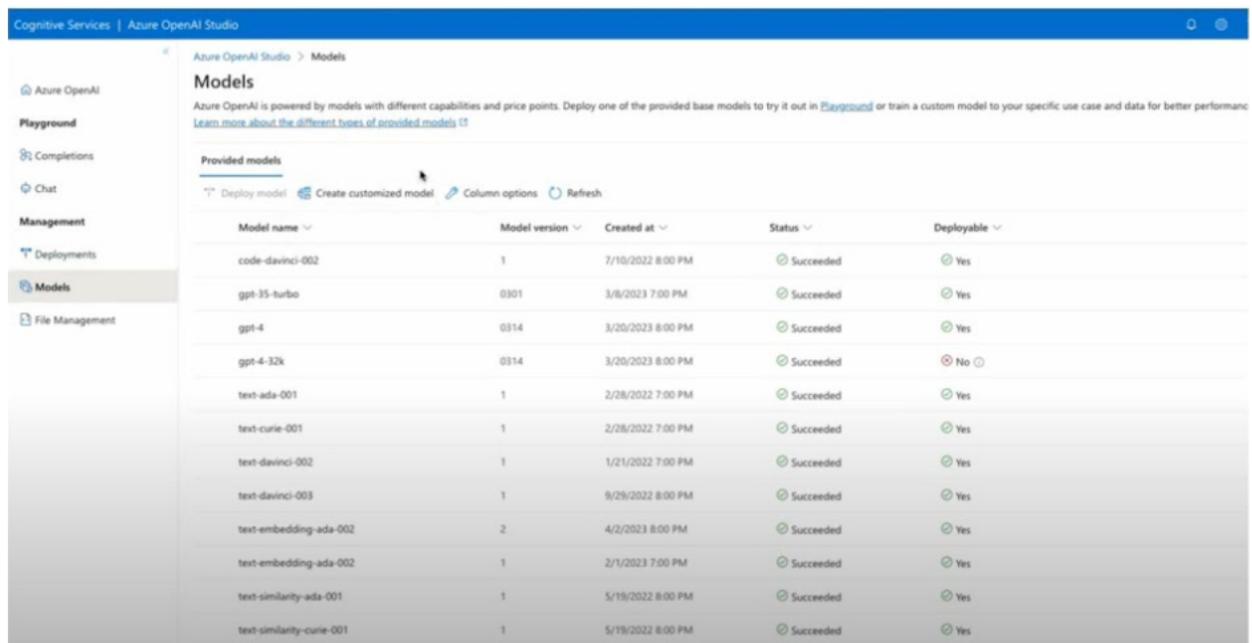


## Deploy OpenAI Models and Set your Deployment ID

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

Extract entities from a text or document.

ChatGPT Prompt used:

You are a highly intelligent entity extractor that displays only the entities in `[{'entity':'entity_name','value':'entity_value'}]` JSON Array format and return empty JSON Array if you are unable to find the entities. An entity is a valuable piece of information contained within the text. Entities will be given as Array of String. Extract *<all the entities / entities list>* from the following text

## 1. Document Input Type

### Inputs:

**Document** (Document) - Required - Provide the document from which the entities are extracted.

**Batch Number** (Number(Integer)) – Optional - Provide the batch number. Default:1

**Number of Pages** (Number(Integer)) – Optional - Provide the number of pages (Batch size) to be selected for entity extraction. Default:1

**Entities to extract** (List of Text) – Optional - Provide the entities to be extracted from the document as a list of text items. Provide null to extract all the entities.

The screenshot shows the Appia EES\_INT\_entityExtractorDocument interface. On the left, the configuration panel includes:

- Connected System \***: EES CS Entity Extractor
- Input Type \***: Document (with a dropdown menu)
- Document to be summarized**: `!!inputDocument` (with a link to upload PDF files with maximum 5 pages)
- Entities to extract**: `!!entities` (with a list editor)

The main workspace displays a table with columns: Rule Input Name, Expression, and Value.

Rule Input Name	Expression	Value
inputDocument (Document)	1	DD_PDF_CHATGPT...
entities (List of Text String)	<pre>2 { 3   "launch date", 4   "organization", 5   "language written" }</pre>	List of Text String: 3 items "launch date" "organization" "language written"

Below the table, the **Value: Result** section shows a dictionary structure:

- Dictionary
  - entities List of Dictionary - 3 items
    - Dictionary
      - value "November 30, 2022" (Text)
      - entity "launch date" (Text)
    - Dictionary
      - value "OpenAI" (Text)
      - entity "organization" (Text)
    - Dictionary
      - value "Python" (Text)
      - entity "language written" (Text)
    - success true (Boolean)

**Output:** Dictionary

{

```

entities: {
  {
    value: "November 30, 2022",
    entity: "launch date"
  },
  { value: "OpenAI", entity: "organization" },
  {
    value: "Python",
    entity: "language written"
  }
},
success: true
}

```

## 2. Text Input Type

### Inputs:

**Input Text** (Text) - Required - Provide the text from which the entities are extracted.

**Entities to extract** (List of Text) – Optional - Provide the entities to be extracted from the text as a list of text items. Provide null to extract all the entities.

The screenshot displays the configuration and execution results for the EES INT\_entityExtractorText rule. The interface is divided into several sections:

- Connected System:** Shows the rule name "EES\_INT\_entityExtractorText".
- Input Type:** Set to "Text".
- Input Text:** A text area containing the text "r!entities".
- Entities to extract:** A list of text items, currently empty.
- Table:** A table with three columns: Rule Input Name, Expression, and Value.
 

Rule Input Name	Expression	Value
text (Text)	1	ChatGPT is an AI-powered language model developed by OpenAI, capable of generating human-like...
entities (List of Text String)	1 * { 2 "organization" 3 }	List of Text String: 1 item "organization"
- Result:** A green bar indicating "Success!".
- Time:** 1,101 ms. Prepare: <1 ms - Execute: 1,101 ms (Send/Wait/Receive 1 ms) - Transform: <1 ms.
- Value: Result:** A dictionary containing:
  - entities: List of Dictionary - 1 item
    - Dictionary
      - value "OpenAI" (Text)
      - entity "organization" (Text)
      - success true (Boolean)

**Output:** Dictionary

```
{  
  entities: { { value: "OpenAI", entity: "organization" } },  
  success: true  
}
```