

# **GenAI Tool: Entity Extractor**

Connected System Plugin for **Appian**

# Appian Corporation

Version 1.0.9

## Table of Contents

Overview	3
Features	3
Connected System Configuration	4-10
Integration	10-12

## Overview

The GenAI Tool: 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.

### Privacy Policy

All information passed through AI tools will be processed and may remain with the organizations that develop those tools. Please exercise caution with what information is disclosed to the AI tool for this reason.


## Features

- Extract Entities from a text or document

# Connected System Configuration

## 1. OpenAI Embedding

### Connected System Properties



**GenAI Tool: Entity Extractor**  
Extract entities from text using ChatGPT  
Version: 1

**Name \***

**Description**

**GenAI Tool: 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 \***

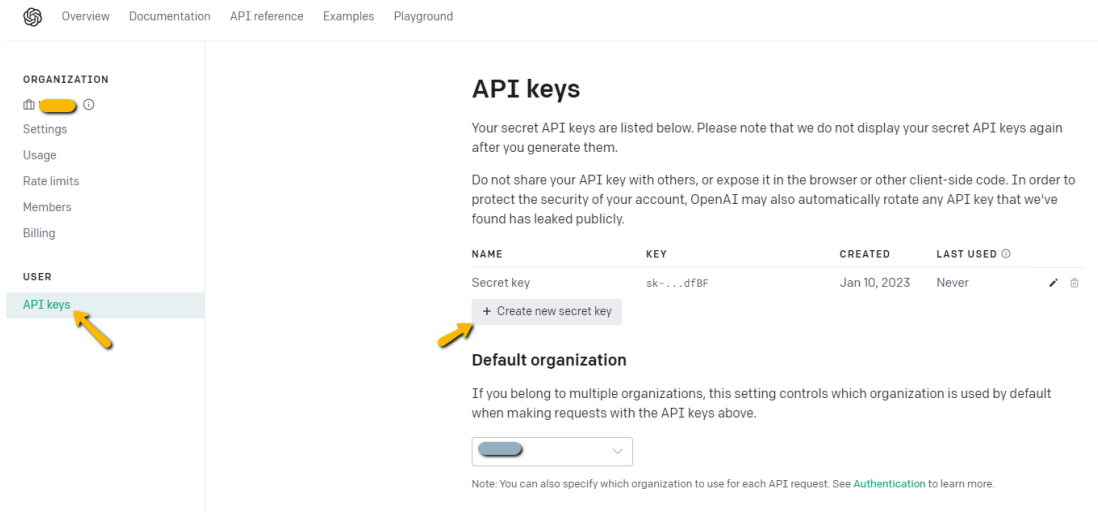
Provide the name of the model to use for text completion. Example: gpt-3.5-turbo for GPT 3.5 Turbo model, gpt-4 for GPT 4 model. Visit <https://platform.openai.com/docs/models/model-endpoint-compatibility> and use one of the models listed under /v1/chat/completions endpoint.

Connection successful

**TEST CONNECTION**

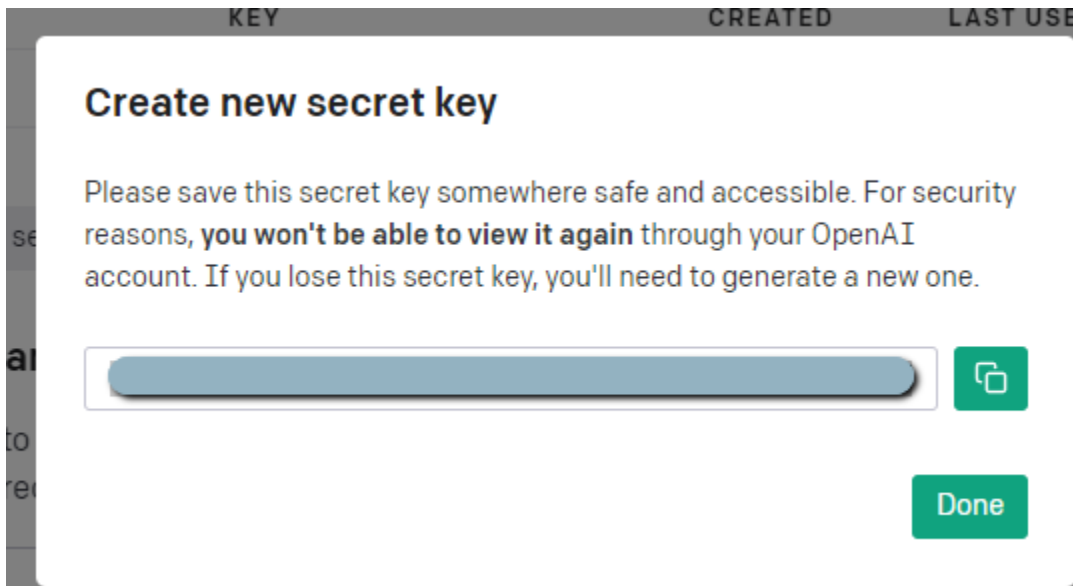
The Entity Extraction connected system with OpenAI authentication requires the following credentials :OpenAI API Key and Completion model.

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



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


- 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.



- Provide the completion model you want to use in the Completion Model field. 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.

## 2. Azure OpenAI Embedding

### Connected System Properties



**GenAI Tool: Entity Extractor**  
Extract entities from text using ChatGPT  
Version: 1

**Name \***

**Description**

**GenAI Tool: 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.

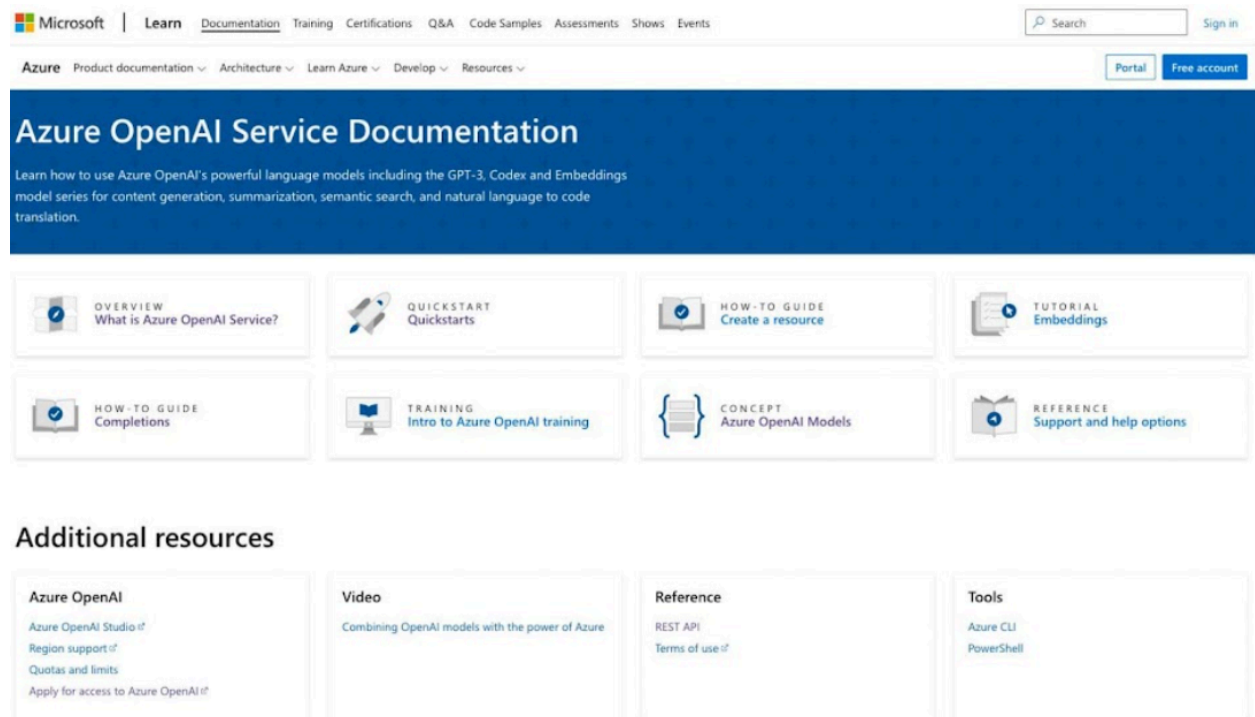
Connection successful

**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.

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.



Microsoft | Learn Documentation Training Certifications Q&A Code Samples Assessments Shows Events Search Sign in

Azure Product documentation Architecture Learn Azure Develop Resources Portal Free account

## Azure OpenAI Service Documentation

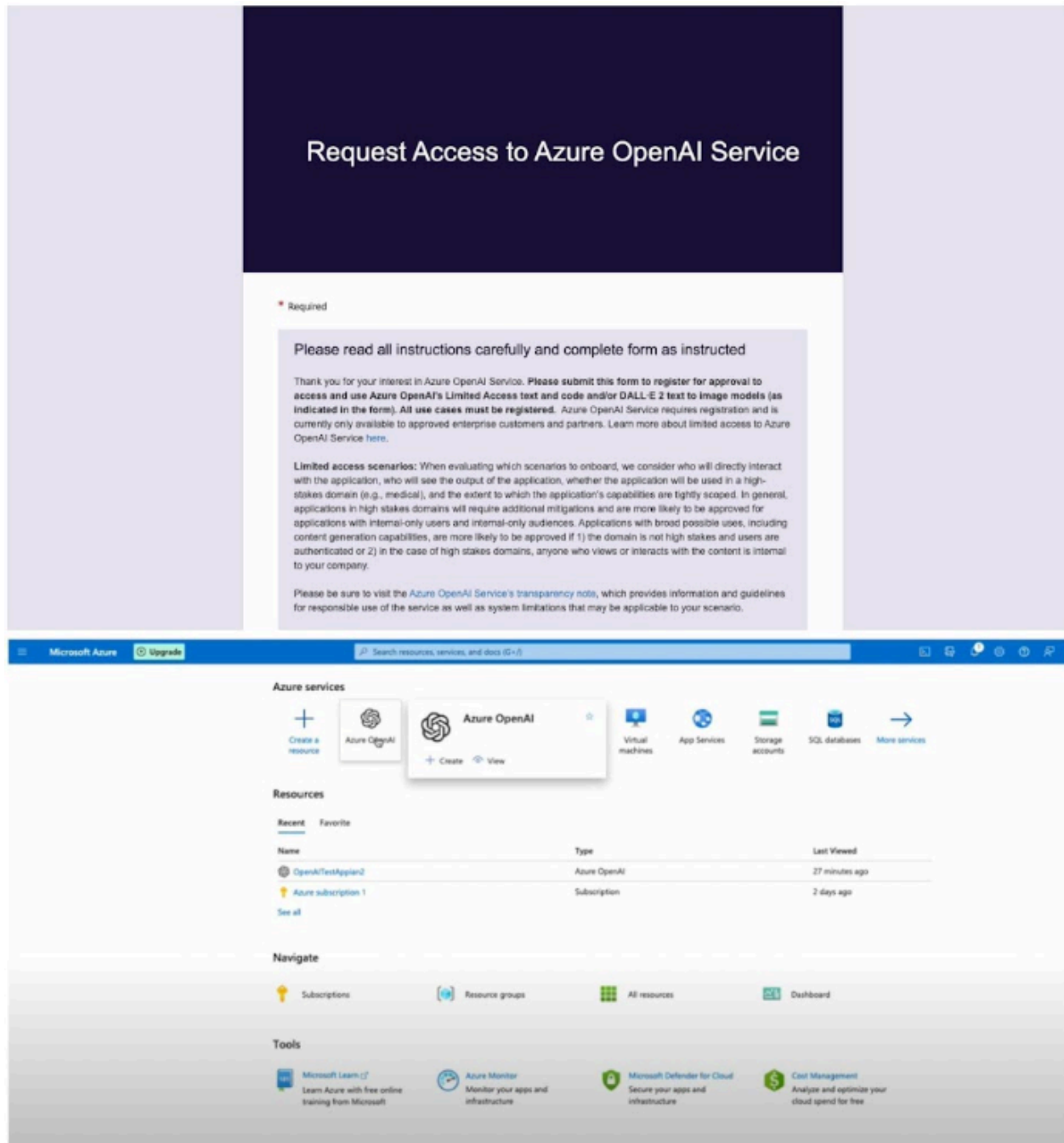
Learn how to use Azure OpenAI's powerful language models including the GPT-3, Codex and Embeddings model series for content generation, summarization, semantic search, and natural language to code translation.

- 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
- REFERENCE: Support and help options

### Additional resources

- Azure OpenAI**
  - Azure OpenAI Studio
  - Region support
  - Quotas and limits
  - Apply for access to Azure OpenAI
- Video**
  - Combining OpenAI models with the power of Azure
- Reference**
  - REST API
  - Terms of use
- Tools**
  - Azure CLI
  - PowerShell

3. Create a service and set domain name.



4. Within your service, create and access API keys through "Keys and Endpoints" under Resource Management.
5. Deploy OpenAI models through the Azure OpenAI Studio. Information generated from these deployments will be needed to use the connected system. This information can be accessed through the "Deployments" tab in Azure OpenAI Studio.



Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > Cognitive Services

## Cognitive Services | Azure OpenAI

Search  + Create Manage deleted resources Manage view Refresh Export to CSV Open query Assign tags Delete

Filter for any field... Subscription equals all Type equals all Resource group equals all Location equals all Add filter

Showing 1 to 1 of 1 records

Name	Kind	Location	Custom Domain Name	Pricing tier	Status
OpenAITestAppian2	OpenAI	East US	openaitestappian2	50	Succeeded

Left sidebar navigation:

- Overview
- All Cognitive Services
- Azure OpenAI
- Speech
  - Speech service
- Language
  - Language service
  - Translator
  - Language understanding (classic)
  - QnA maker (classic)
- Vision
  - Computer vision
  - Custom vision
  - Face API
- Decision
  - Anomaly detector
  - Content moderator
  - Personalizer
  - Health Insights

Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > Cognitive Services | Azure OpenAI > OpenAITestAppian2

## OpenAITestAppian2 | Keys and Endpoint

Search  Regenerate Key1 Regenerate Key2

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource Management

- Keys and Endpoint
- Model deployments
- Pricing tier
- Networking
- Identity
- Cost analysis
- Properties
- Locks

Monitoring

- Alerts
- Metrics
- Diagnostic settings
- Logs

Automation

These keys are used to access your Cognitive Service API. Do not share your keys. Store them securely-- for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1

KEY 2

Location/Region

Endpoint

Model name	Model version	Created at	Status	Deployable
code-davinci-002	1	7/10/2022 8:00 PM	Succeeded	Yes
gpt-35-turbo	0301	3/8/2023 7:00 PM	Succeeded	Yes
gpt-4	0314	3/20/2023 8:00 PM	Succeeded	Yes
gpt-4-32k	0314	3/20/2023 8:00 PM	Succeeded	No
text-ada-001	1	2/28/2022 7:00 PM	Succeeded	Yes
text-curie-001	1	2/28/2022 7:00 PM	Succeeded	Yes
text-davinci-002	1	1/21/2022 7:00 PM	Succeeded	Yes
text-davinci-003	1	9/29/2022 8:00 PM	Succeeded	Yes
text-embedding-ada-002	2	4/2/2023 8:00 PM	Succeeded	Yes
text-embedding-ada-002	1	2/1/2023 7:00 PM	Succeeded	Yes
text-similarity-ada-001	1	5/19/2022 8:00 PM	Succeeded	Yes
text-similarity-curie-001	1	5/19/2022 8:00 PM	Succeeded	Yes

## 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. The response should not contain anything outside of 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 in List of Text format. Provide null to extract all the entities.

EES\_INT\_entityExtractorDocument

SAVE CHANGES

appia

Connected System \*

EES CS Entity Extractor

Input Type \*

Document

Provide the input type for entity extraction. Valid values are: TEXT, DOCUMENT

Document to be summarized

inputDocument

Upload PDF files with maximum 5 pages.

Entities to extract

1 r!entities

TEST REQUEST

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

Set as default test values

Value: Result

- 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 document in List of Text format. Provide null to extract all the entities.

EES\_INT\_entityExtractorText

Connected System \*  
EES CS Entity Extractor

Input Type \*  
Text

Provide the input type for entity extraction. Valid values are : TEXT, DOCUMENT

Input Text  
rlltext

Provide the text to be extracted.

Entities to extract

```
1 rllentities
```

Place cursor on function, rule, or constant to display help

Provide the as List of Text

TEST REQUEST

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"

Set as default test values

Result Request Response

Success!

Time  
1,101 ms  
Prepare: < 1 ms - Execute: 1,101 ms (Send/Wait/Receive: 1 ms) - Transform: < 1 ms

Value: Result

- Dictionary
  - 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
}
```