

# OpenAI Conversation

The OpenAI integration adds a conversation agent powered by [OpenAI](#) in Home Assistant.

Controlling Home Assistant is done by providing the AI access to the Assist API of Home Assistant. You can control what devices and entities it can access from the [exposed entities page](#). The AI is able to provide you information about your devices and control them.

This integration does not integrate with [sentence triggers](#).

This integration requires an API key to use, [which you can generate here](#). This is a paid service, we advise you to monitor your costs in the [OpenAI portal](#) closely and configure [usage limits](#) to avoid unwanted costs associated with using the service.

## Configuration

To add the **OpenAI Conversation** service to your Home Assistant instance, use this My button:

ADD INTEGRATION TO MY 

Manual configuration steps



# Generate an API Key

The OpenAI key is used to authenticate requests to the OpenAI API. To generate an API key take the following steps:

- Log in to the [OpenAI portal](#) or sign up for an account.
- Enable billing with a valid credit card
- Configure [usage limits](#).
- Visit the [API Keys page](#) to retrieve the API key you'll use to configure the integration.

## Options

Options for OpenAI Conversation can be set via the user interface, by taking the following steps:

- Browse to your Home Assistant instance.
- Go to [Settings > Devices & Services](#).
- If multiple instances of OpenAI Conversation are configured, choose the instance you want to configure.
- Select the integration, then select **Configure**.

### Instructions

Instructions for the AI on how it should respond to your requests. It is written using [Home Assistant Templating](#).

### Control Home Assistant

If the model is allowed to interact with Home Assistant. It can only control or provide information

If the model is allowed to interact with Home Assistant, it can only control or provide information about entities that are [exposed](#) to it.

---

## Recommended settings

If enabled, the recommended model and settings are chosen.

If you choose to not use the recommended settings, you can configure the following options:

### Model

The GPT language model is used for text generation. You can find more details on the available models in the [GPT-4o Documentation](#). The default is “gpt-4o-mini”.

---

### Maximum Tokens to Return in Response

The maximum number of words or “tokens” that the AI model should generate in its completion of the prompt. For more information, see the [OpenAI Completion Documentation](#).

---

### Temperature

A value that determines the level of creativity and risk-taking the model should use when generating text. A higher temperature means the model is more likely to generate unexpected results, while a lower temperature results in more deterministic results. See the [OpenAI Completion Documentation](#) for more information.

---

### Top P

An alternative to temperature, top\_p determines the proportion of the most likely word choices the model should consider when generating text. A higher top\_p means the model will only consider the most likely words, while a lower top\_p means a wider range of words, including less likely ones, will be considered. For more information, see the [OpenAI Completion API Reference](#).

---

### Enable web search

Enable OpenAI-provided [Web search tool](#). Note that it is only available for gpt-4o and gpt-4o-mini models.

---

### Search context size

The search is performed with a separate fine-tuned “gpt-4o-search-preview” or “gpt-4o-mini-search-preview” model with its own context and its own [pricing](#) . This parameter controls how much context is retrieved from the web to help the tool formulate a response. The tokens used by the search tool do not affect the context window of the main model. These tokens are also not carried over from one turn to another — they’re simply used to formulate the tool response and then discarded. This parameter would affect the search quality, cost, and latency.

**Include home location**

This parameter allows using the location of your Home Assistant instance during search to provide more relevant search results.

# Talking to Super Mario over the phone

You can use an OpenAI Conversation integration to [talk to Super Mario and, if desired, have it control devices](#) in your home.

## Actions

### ACTION OPENAI\_CONVERSATION.GENERATE\_IMAGE

Allows you to ask OpenAI to generate an image based on a prompt. This action populates [Response Data](#) with the requested image.

Data attribute	Optional	Description	Example
<a href="#">config_entry</a>	no	Integration entry ID to use.	
<a href="#">prompt</a>	no	The text to turn into an image.	Picture of a dog
<a href="#">size</a>	yes	Size of the returned image in pixels. Must be one of <a href="#">1024x1024</a> , <a href="#">1792x1024</a> , or <a href="#">1024x1792</a> . defaults to <a href="#">1024x1024</a> .	1024x1024

<code>quality</code>	yes	The quality of the image that will be generated. <code>hd</code> creates images with finer details and greater consistency across the image.	standard
<code>style</code>	yes	The style of the generated images. Must be one of <code>vivid</code> or <code>natural</code> . Vivid causes the model to lean towards generating hyper-real and dramatic images. Natural causes the model to produce more natural, less hyper-real looking images.	vivid

```

action: openai_conversation.generate_image
data:
  config_entry: abce6b8696a15e107b4bd843de722249
  prompt: "Cute picture of a dog chasing a herd of cats"
  size: 1024x1024
  quality: standard
  style: vivid
response_variable: generated_image

```

The response data field `url` will contain a URL to the generated image and `revised_prompt` will contain the updated prompt used.

## EXAMPLE USING A GENERATED IMAGE ENTITY

The following example shows an automation that generates an image and displays it in a image template entity. The prompt uses the state of the weather entity to generate a new image of New York in the current weather state.

The resulting image entity can be used in, for example, a card on your dashboard.

The `config_entry` is installation specific. To get the value, make sure the integration has been installed. Then, go to [Developer Tools > Actions](#). Ensure you are in UI mode and enter the following below:

Developer tools

YAML

STATES

SERVICES

TEMPLATE

EVENTS

STATISTICS

ASSIST

The service dev tool allows you to call any available service in Home Assistant.

Service

OpenAI Conversation: Generate image

×

▼

Turn a prompt into an image

Config Entry

The config entry to use for this service

Integration \*

OpenAI Conversation

×

▼

Prompt

The text to turn into an image

☒

Size

The size of the image to generate

☒

1024x1024

☐

1024x1792

☐

1792x1024

☒

Quality

The quality of the image that will be generated

☒

standard

☐

hd

☒

Style

The style of the generated image

☒

vivid

☐

natural

GO TO YAML MODE

CALL SERVICE

Select **YAML Mode** to reveal the `config_entry` value to be used in the below example automation.

Developer tools

YAML

STATES

SERVICES

TEMPLATE

EVENTS

STATISTICS

ASSIST

The service dev tool allows you to call any available service in Home Assistant.

Service

OpenAI Conversation: Generate image

×

▼

```
1 service: openai_conversation.generate_image
2 data:
3   size: 1024x1024
4   config_entry: abce6b8696a15e107b4bd843de722249
5   quality: standard
6   style: vivid
7
```

GO TO UI MODE

CALL SERVICE

All available parameters

?

Parameter	Description	Example
config_entry	The config entry to use for this service	

prompt	The text to turn into an image	A photo of a dog
size	The size of the image to generate	1024x1024
quality	The quality of the image that will be generated	standard
style	The style of the generated image	vivid

FILL EXAMPLE DATA

```
automation:
- alias: "Update image when weather changes"
  triggers:
    - trigger: state
      entity_id: weather.home
  actions:
    - alias: "Ask OpenAI to generate an image"
      action: openai_conversation.generate_image
      response_variable: generated_image
      data:
        config_entry: abce6b8696a15e107b4bd843de722249
        size: "1024x1024"
        prompt: >-
          New York when the weather is {{ states("weather.home") }}

    - alias: "Send out a manual event to update the image entity"
      event: new_weather_image
      event_data:
        url: '{{ generated_image.url }}'

template:
- trigger:
    - alias: "Update image when a new weather image is generated"
      trigger: event
      event_type: new_weather_image
  image:
    - name: "AI generated image of New York"
      url: "{{ trigger.event.data.url }}"
```

## SERVICE OPENAI\_CONVERSATION.GENERATE\_CONTENT

Allows you to ask OpenAI to generate a content based on a prompt. This service populates [Response Data](#) with the response from OpenAI.

- **Service data attribute:** `config_entry`

- **Description:** Integration entry ID to use.
- **Example:**
- **Optional:** no
- **Service data attribute:** `prompt`
  - **Description:** The text to generate content from.
  - **Example:** Describe the weather
  - **Optional:** no
- **Service data attribute:** `image_filename`
  - **Description:** List of file names for images to include in the prompt.
  - **Example:** /tmp/image.jpg
  - **Optional:** yes

```
service: openai.generate_content
data:
  config_entry: abce6b8696a15e107b4bd843de722249
  prompt: >-
    Very briefly describe what you see in this image from my doorbell camera.
    Your message needs to be short to fit in a phone notification. Don't
    describe stationary objects or buildings.
  image_filename:
    - /tmp/doorbell_snapshot.jpg
  response_variable: generated_content
```

The response data field `text` will contain the generated content.

Another example with multiple images:

```
service: openai.generate_content
data:
  prompt: >-
    Briefly describe what happened in the following sequence of images
    from my driveway camera.
  image_filename:
    - /tmp/driveway_snapshot1.jpg
```



- /tmp/driveway\_snapshot2.jpg
- /tmp/driveway\_snapshot3.jpg
- /tmp/driveway\_snapshot4.jpg

`response_variable`: generated\_content

## Related topics

- [Exposing entities to assist](#)
- [Create an ai personality](#)

## Related links

- [OpenAI API key](#)
- [OpenAI](#)

### HELP US IMPROVE OUR DOCUMENTATION

Suggest an edit to this page, or provide/view feedback for this page.

[Edit](#)

[Provide feedback](#)

[View pending feedback](#)



The OpenAI Conversation service was introduced in Home Assistant 2023.2, and it's used by [4.2%](#) of the active installations.

Its IoT class is [Cloud Polling](#).

[View source on GitHub](#)

[View documentation](#)