



INflow Labs

ChatGPT in Microsoft Teams

by Ryan Malani 📅 February 8, 2023 ⌚ 4 min read

It goes without saying that the talk of the town is the recently released ChatGPT of OpenAI.¹ Despite the recent large investment from Microsoft,² we imagine it will take a while to implement the fully-fledged functionality across their product suite. However, in about an hour of combined research and implementation we devised a solution to implement the completions functionality of their most powerful GPT-3 model, text-davinci-003,³ as a chatbot available through our Microsoft Teams instance using the OpenAI API [application programming interface].

getting started

There are a couple of things that you'll need to get started on this project. The first is obviously Microsoft Teams [<https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>], second is Power Virtual Agents [<https://powervirtualagents.microsoft.com/en-us/>] and Power Automate [<https://powerautomate.microsoft.com/en-us/>], and lastly would be an OpenAI account [<https://platform.openai.com/signup>] capable of generating API keys.

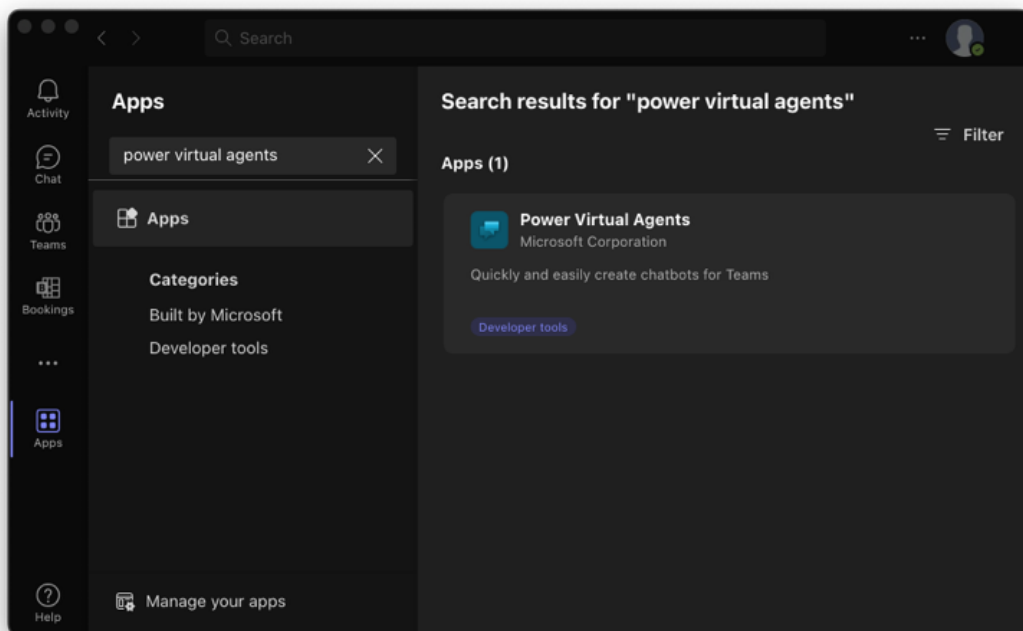
generating an API key

Log into your OpenAI account, click on the organization or profile listed in the navigation ribbon, and click on **View API keys**. From there, you can just click **Create new secret key** and copy your newly generated key.

+ Create new secret key

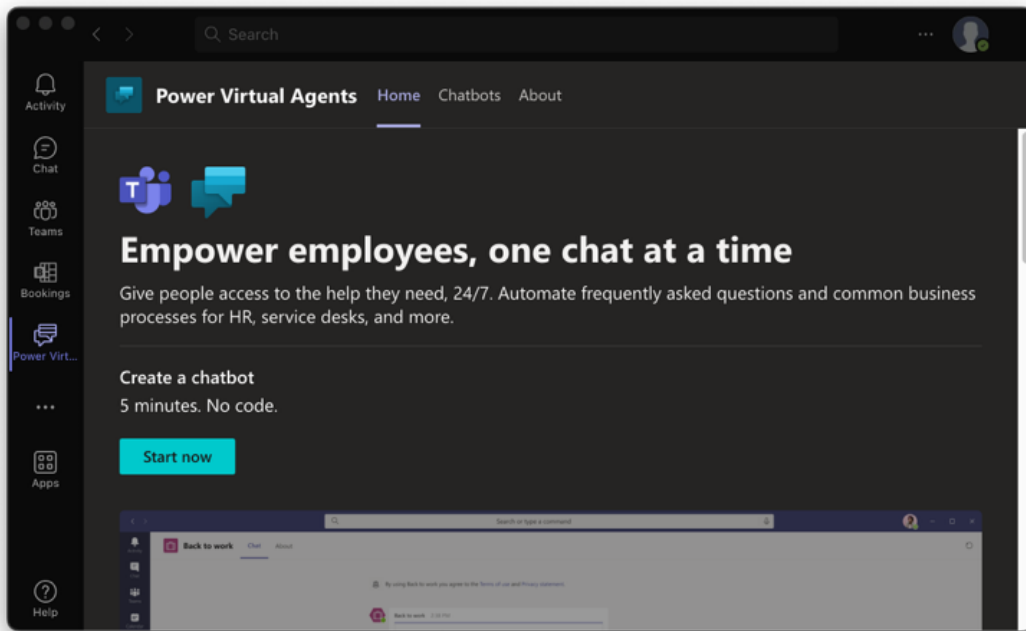
creating a new virtual agent

From within your Teams environment, click the **Apps** button in the side navigation menu and search for “Power Virtual Agents”.



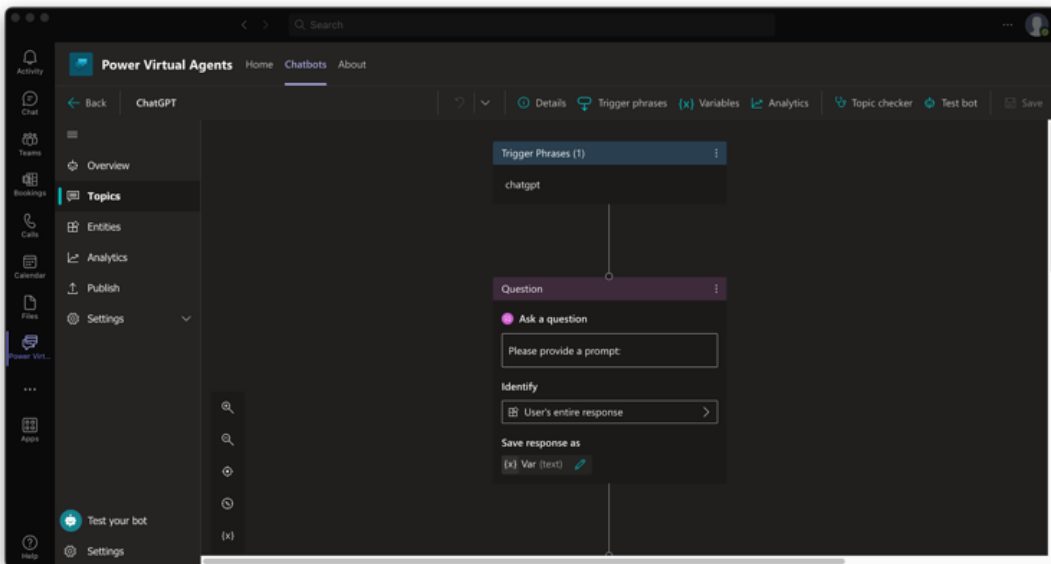
If this is your first time using it, you'll be prompted to add it to your organization's environment. Once you've done that, click **Open** to access the tool. Once it's open, click **Start now** to make a new

chatbot, and select the appropriate team you want to have access to it.



From there, click into **Topics** on the left. You can switch the status of the default topics to **off** and use the **New topic** dropdown at the top to make a new topic from scratch. Once you have a fresh topic opened up, you'll do the following:

1. Adding some trigger phrases that you'll use to call the functionality of the chatbot - in our case we just used "chatgpt" to keep it simple.
2. Add an "Ask a question" node. Prompt the user with a question, Identify the User's entire response, and save the user's entire response to a text variable that will be used as an input to our Power Automate flow.



creating the power automate flow

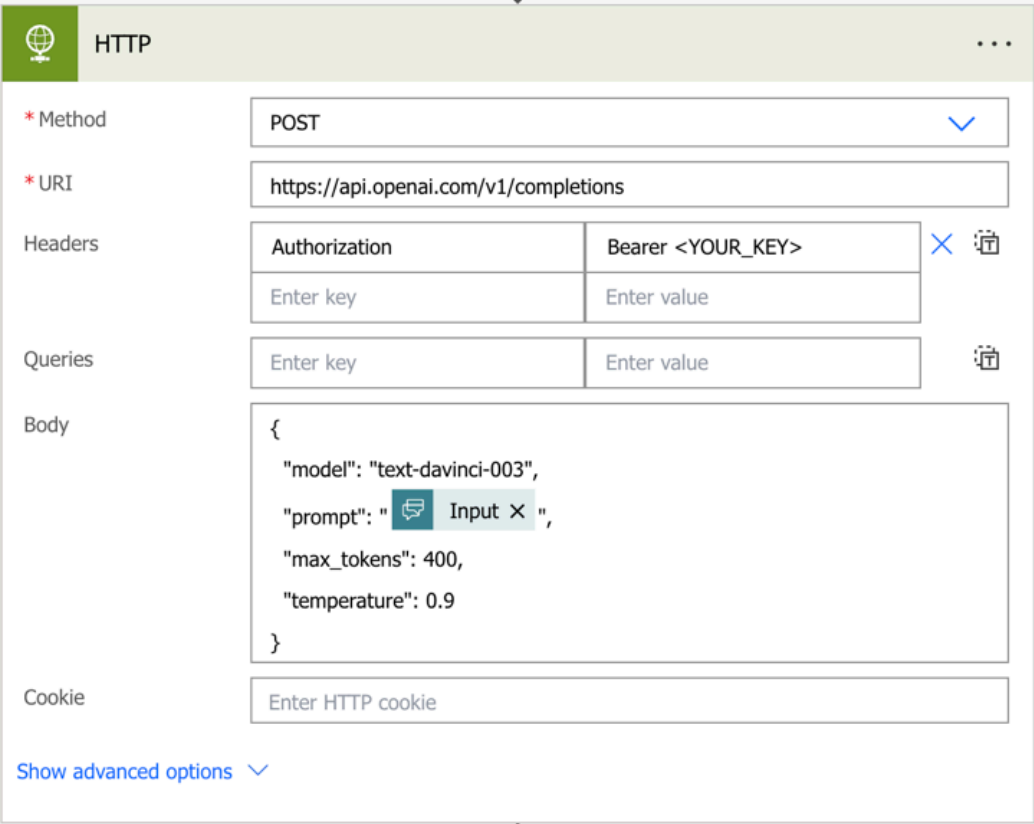
Next, we'll create a node to "Call an action" and "Create a flow". I've already created the API request flow, so I'll skip ahead and explain how it works.

First, using the input from the virtual agent, we'll make an HTTP request to the API at the completions endpoint (<https://api.openai.com/v1/completions>). Since it's a POST request, we'll have to pass in a request body:

```
{
  "model": "text-davinci-003",
  "prompt": "@{triggerBody()['text']}",
  "max_tokens": 400,
  "temperature": 0.9
}
```

In this example, we're using the davinci model, prompting using the input from the user, using no more than 400 tokens per request [this number can be increased to 4096⁴], and a temperature of 0.9

to provide a more verbose response [more randomized as per the docs⁴]. In the headers that are passed, be sure to replace <YOUR_KEY> with the secret key you just generated from OpenAI.



The screenshot shows an HTTP client interface with the following fields:

- Method:** POST
- URI:** https://api.openai.com/v1/completions
- Headers:**
 - Authorization: Bearer <YOUR_KEY>
 - Enter key: Enter value
- Queries:**
 - Enter key: Enter value
- Body:**

```
{
  "model": "text-davinci-003",
  "prompt": "Input X",
  "max_tokens": 400,
  "temperature": 0.9
}
```
- Cookie:** Enter HTTP cookie

At the bottom, there is a link: [Show advanced options](#)

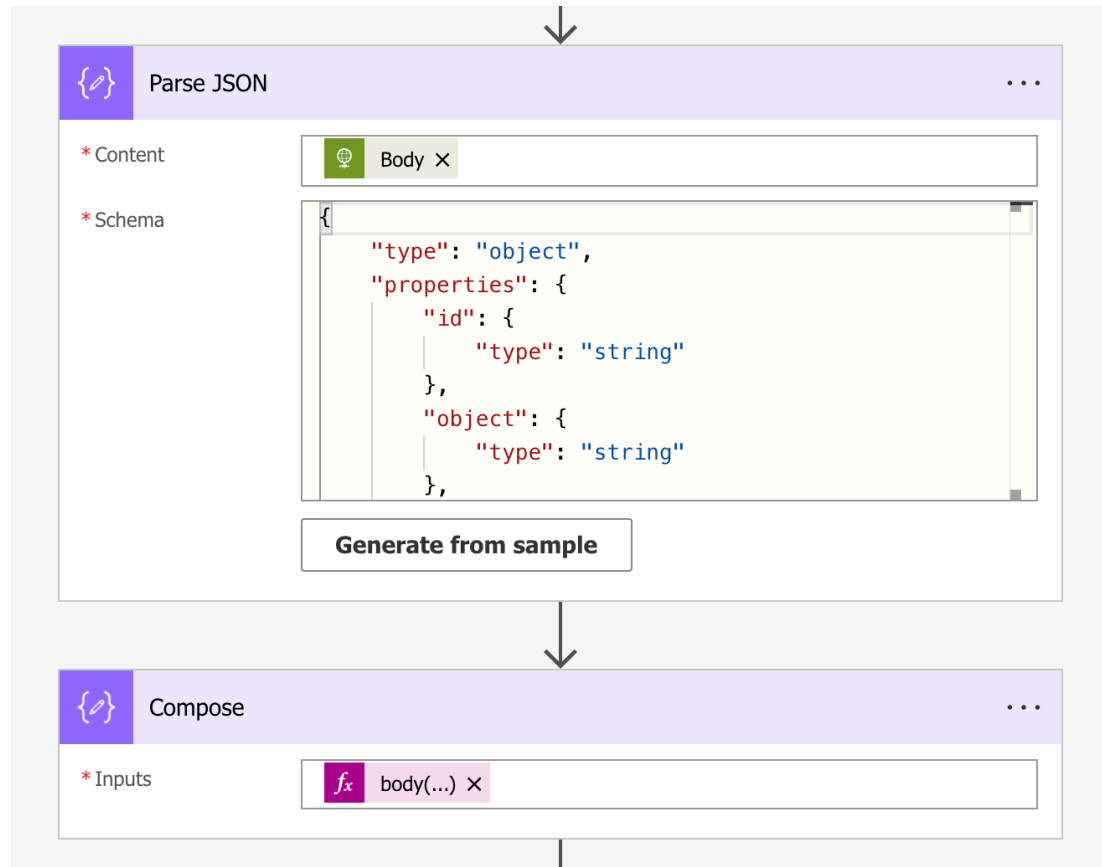
Next, you'll have to parse the JSON response that you receive from the API, using a built-in "Parse JSON" step. I'll save you the hassle of generating the schema:

```
{
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "object": {
      "type": "string"
    },
    "created": {
```

```
    "type": "integer"
  },
  "model": {
    "type": "string"
  },
  "choices": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "text": {
          "type": "string"
        },
        "index": {
          "type": "integer"
        },
        "logprobs": {},
        "finish_reason": {
          "type": "string"
        }
      },
      "required": ["text", "index", "logprobs", "finish_reason"]
    }
  },
  "usage": {
    "type": "object",
    "properties": {
      "prompt_tokens": {
        "type": "integer"
      },
      "completion_tokens": {
        "type": "integer"
      },
      "total_tokens": {
        "type": "integer"
      }
    }
  }
}
```

Follow that up with a built in “Compose” step, utilizing the following as a “Expression” to access the first element of the “choices” array generated by the API:

```
body('Parse_JSON')?['choices']?[0]['text']
```

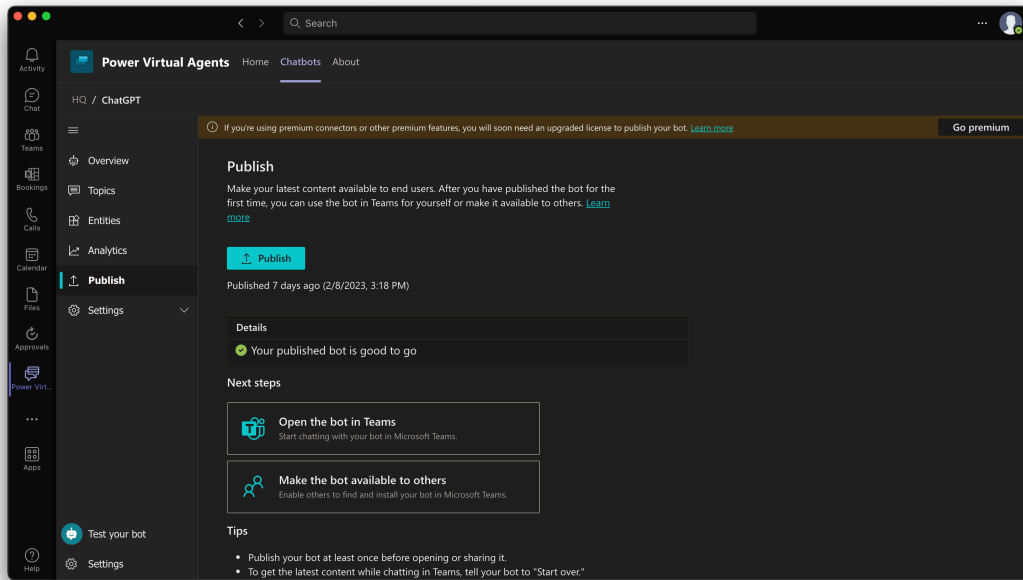


Finally, send the output of the output step of the flow to the virtual agent and share that with the user in the “Message” node.

publishing the virtual agent

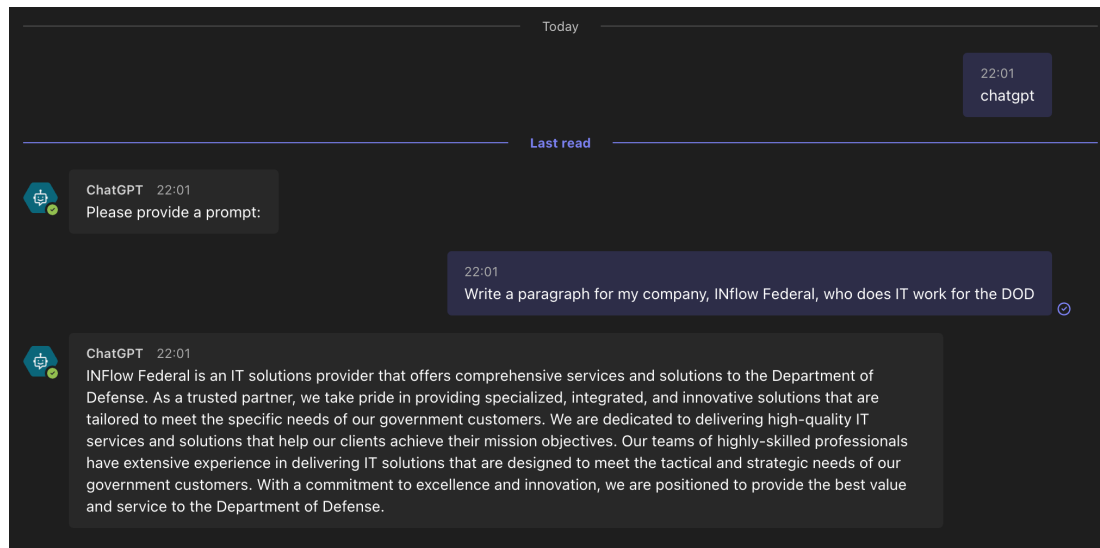
You can use the “Test bot” functionality at the top to ensure that your API request is being routed correctly and providing you with a proper output. As long as everything looks good, you’re ready to publish your bot for use by the rest of your team! Here’s how:

1. Click “Save” at the top to ensure all your changes are included.
2. Click “Publish” in the left side menu, and then the “Publish” button. Once your bot has been published, you should get a confirmation message like the one below.
3. If you need to, use the “Make the bot available to others” button and follow the prompts.



usage

Congrats on deploying your ChatGPT powered Virtual Agent! Just prompt by sending “chatgpt” and it can now be used to generate text:



1. <https://openai.com/blog/chatgpt/> (<https://openai.com/blog/chatgpt/>) ↩
2. <https://www.bloomberg.com/news/articles/2023-01-23/microsoft-makes-multibillion-dollar-investment-in-openai>
(<https://www.bloomberg.com/news/articles/2023-01-23/microsoft-makes-multibillion-dollar-investment-in-openai>) ↩
3. <https://platform.openai.com/docs/models/gpt-3>
(<https://platform.openai.com/docs/models/gpt-3>) ↩
4. <https://platform.openai.com/docs/api-reference/completions/create#completions/create-prompt>
(<https://platform.openai.com/docs/api-reference/completions/create#completions/create-prompt>) ↩ ↩²