Build a Chatbot with PromptFlow endpoint + PVA

What is PromptFlow?

Prompt flow is a powerful feature within Azure Machine Learning that streamlines the development, evaluation, and continuous integration and deployment (CI/CD) of prompt engineering projects. It empowers data scientists and LLM application developers with an experience that combines natural language prompts, templating language, a list of built-in tools and Python code. You can learn more about it here: What is Azure Machine Learning prompt flow

What is PVA (Power Virtual Agents)?

Microsoft Power Virtual Agents empowers everyone to create intelligent conversational bots - from citizen developers, business users to professional developers alike - to seamlessly build secure and scalable virtual agents in an integrated building platform. You can learn more about it here: Power Virtual Agents overview

How to get started

1. You will first need to setup your PromptFlow endpoint. To do this, please setup your custom environment here with your use case: Deploy a prompt flow using Docker | Github. You can test your endpoint using Postman client. Keep your configuration handy. For example:

   Request:

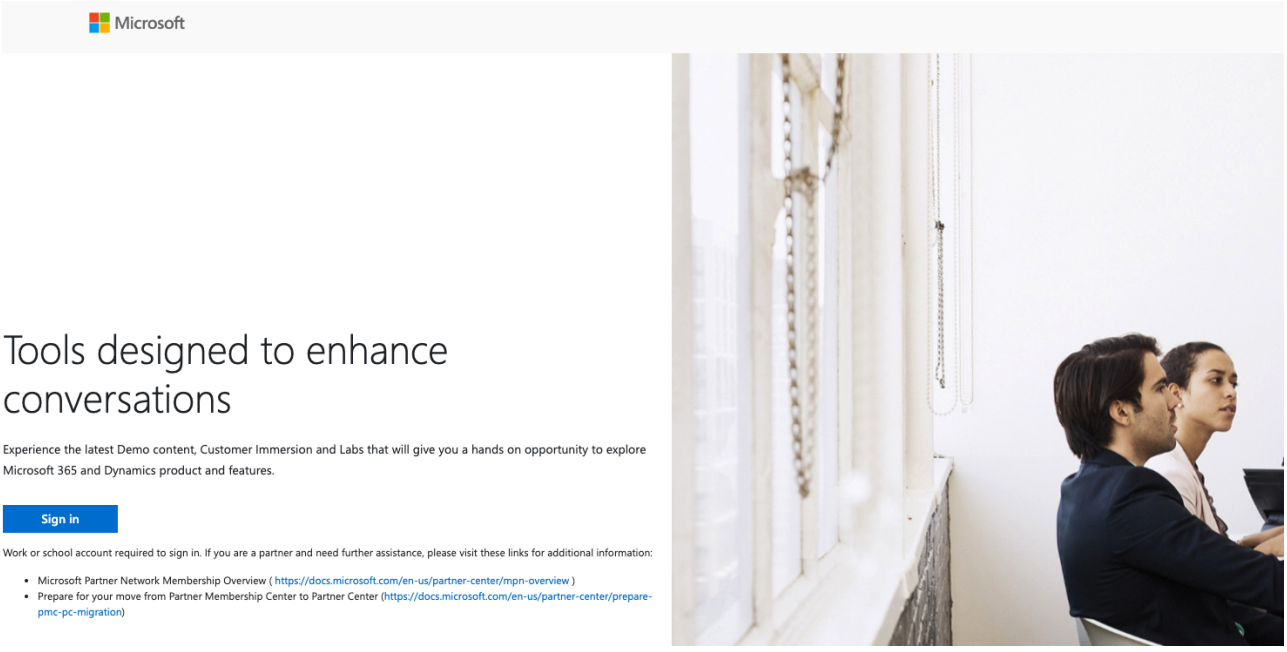
   ```
   POST https://promptflow-webapp-hack.azurewebsites.net/score

   body:
   { "chat_input": "How to verify a customer?", "chat_history": [] }
   ```

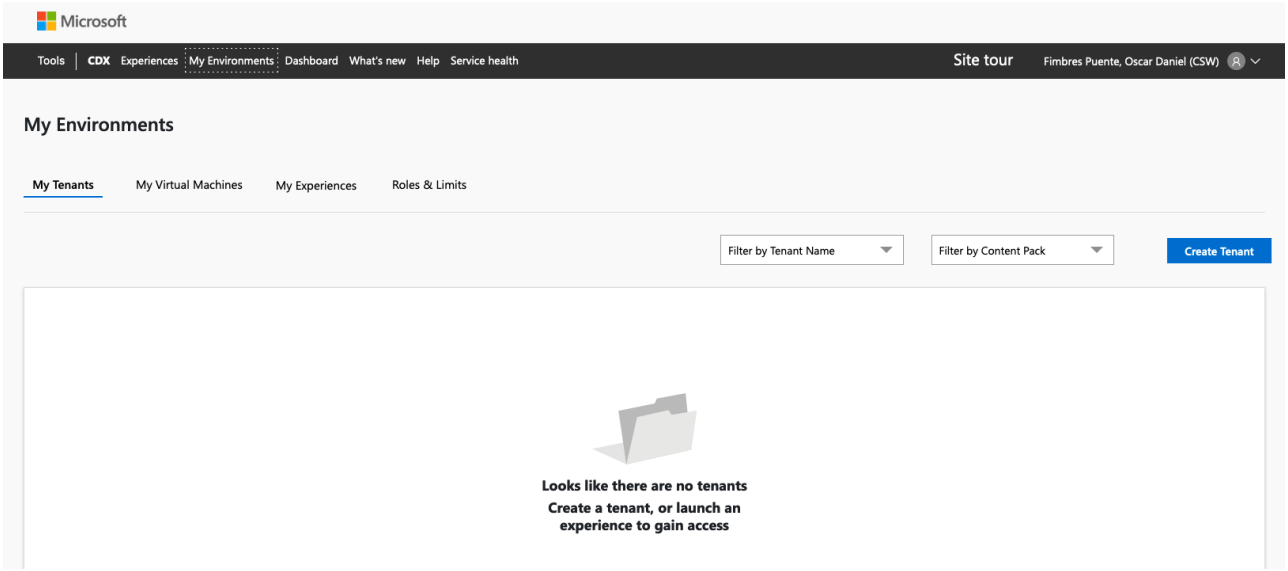
   Response:

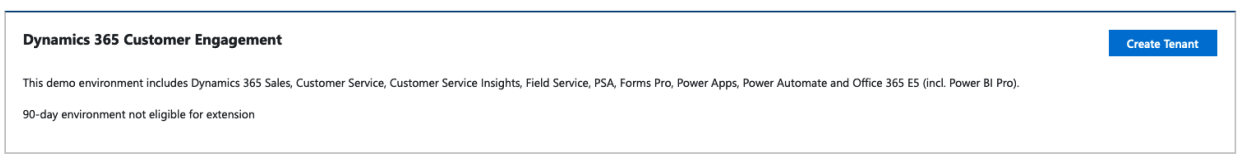
   ```
   body:
   {
       "answer": "To verify a customer, you need to follow the Account
   Verification Process - BCSS. The process involves determining the
   verification method based on the caller's type and workgroup, asking
   the caller to provide their full name, CTN, account name or company
   name, and validating the information provided using FaST or Clarify.
   If the caller is unable to complete all verification requirements,
   they are unverified and have General Access. (Source: Account
   Verification Process - BCSS)",
       "context": "Content: What I Need to Know\n\nAdhere to all VID
   Business Rules found in Account Verification Process -
   BCSS.\n\nDetermine Verification Method\n\nAsk the caller to
   provide:¶\r\nCaller's full nameATTUID¶\r\nIf Screen pop is available,
   the ATTUID can be confirmed instead of requesting it be
   provided.Identify the caller's Business Unit in Webphone. If the
   ```

```
caller does not have a Business Unit listed, move up in the hierarchy
until a Business Unit is visible.¶\r\n¶\n\nContinue verification based
on the Business Unit.¶\r\nWhen the Business Unit starts with AT&T
Business, follow AT&T Business (BCSS/Enterprise/GBS) or Finance.When
the Business Unit does not start with AT&T Business, follow Non-AT&T
Business - All Retail/Virtual Sales Experience (formerly DMDR/Mobility
Sales)/AT&T Right to You.¶\n\nNon-AT&T Business - All Retail/Virtual
Sales ..."
}
```

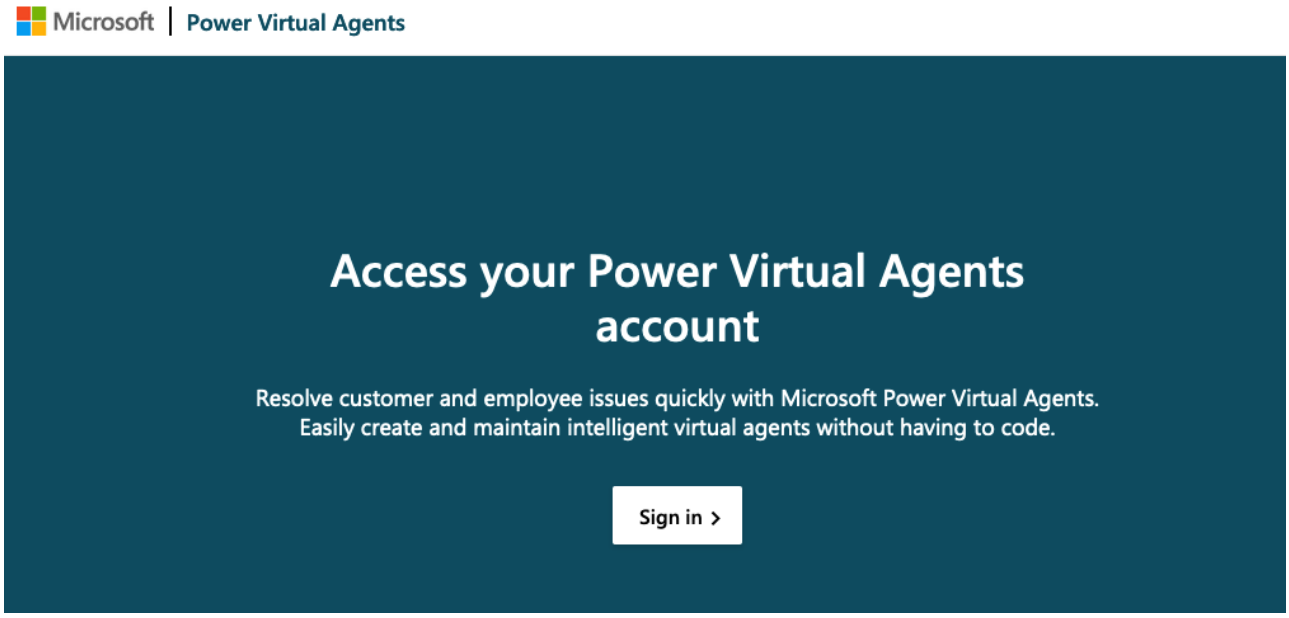
2. As a next step, you will need a PVA environment. Easiest way to get access to such an environment is to create a tenant from demos.microsoft.com. Go to My Environments|Create Tenant. You want a D365 Customer Engagement tenant - it comes with the licenses you need.





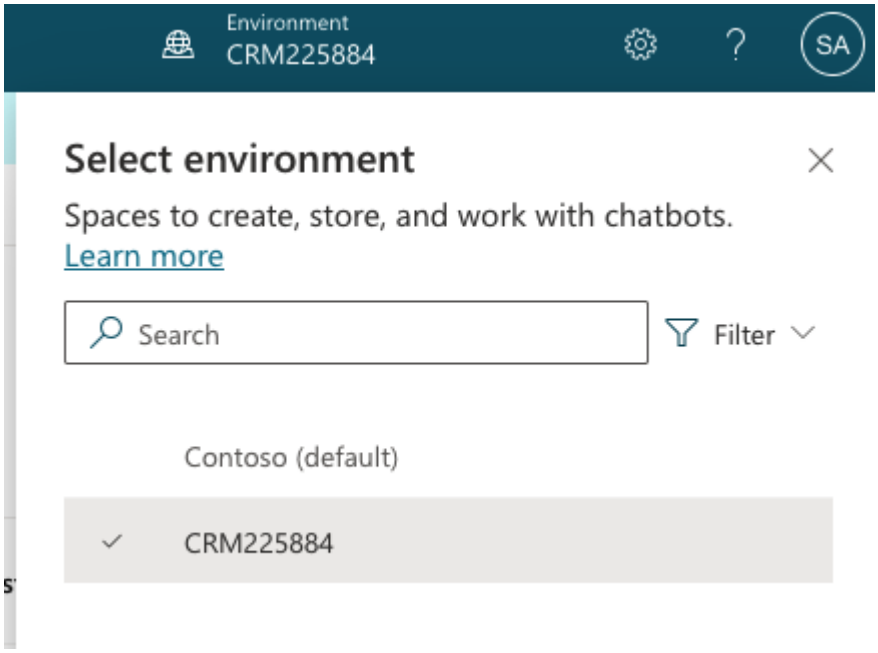


You will receive a set of credentials to access your custom environment. Go to powervirtualagents.microsoft and use these new credentials.

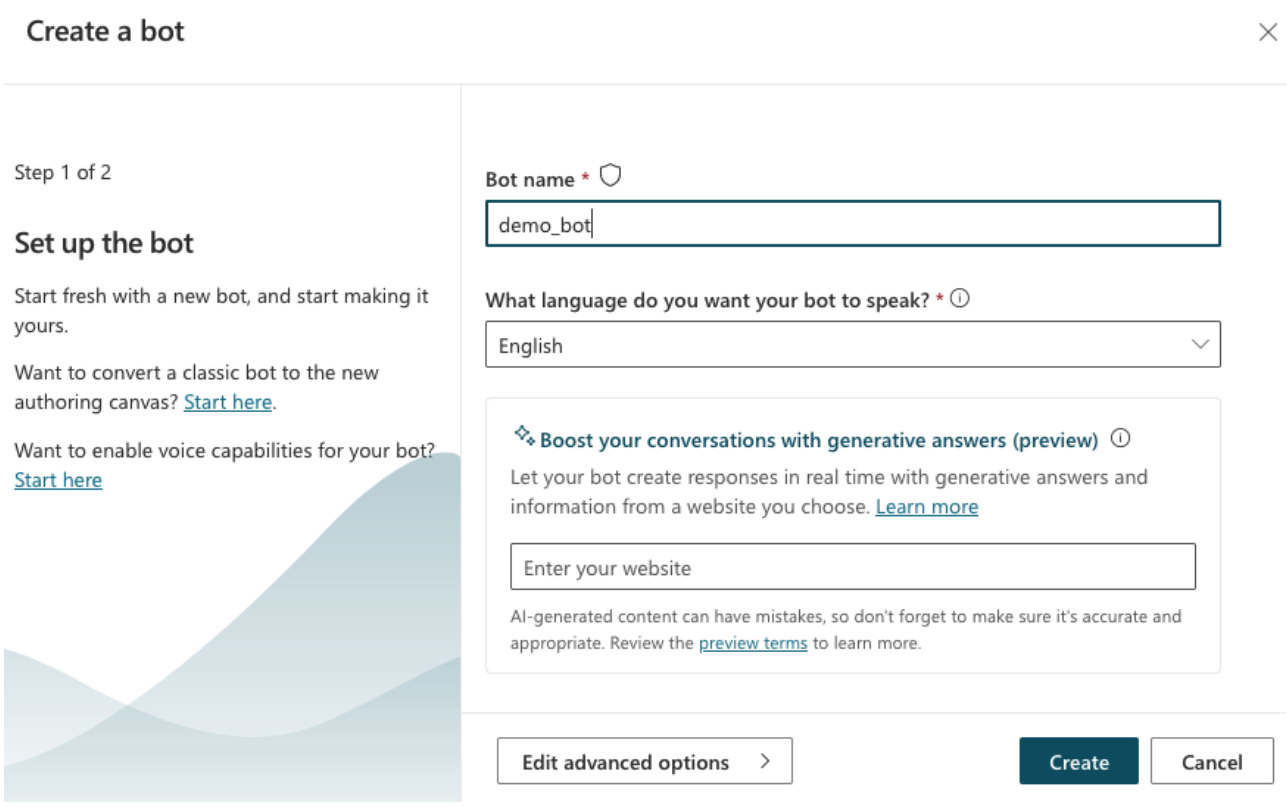


Now that you are all set-up, let's get started

1. Sign-in the Power Virtual Agents page with your existing account. Make sure your new environment is selected.

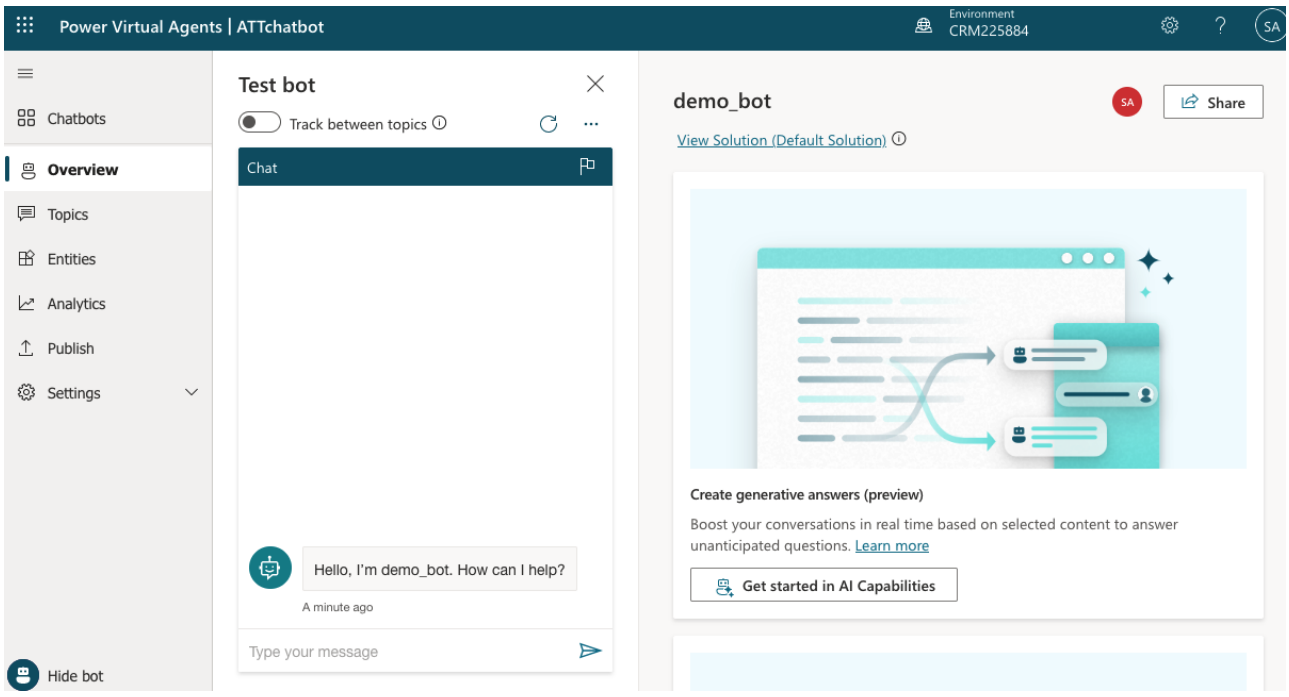


2. One the next step, click "Create" to create a new bot. Then, "Try the unified canvas (preview)". Provide a name for your bot. E.g. "demo_bot", then "Create".

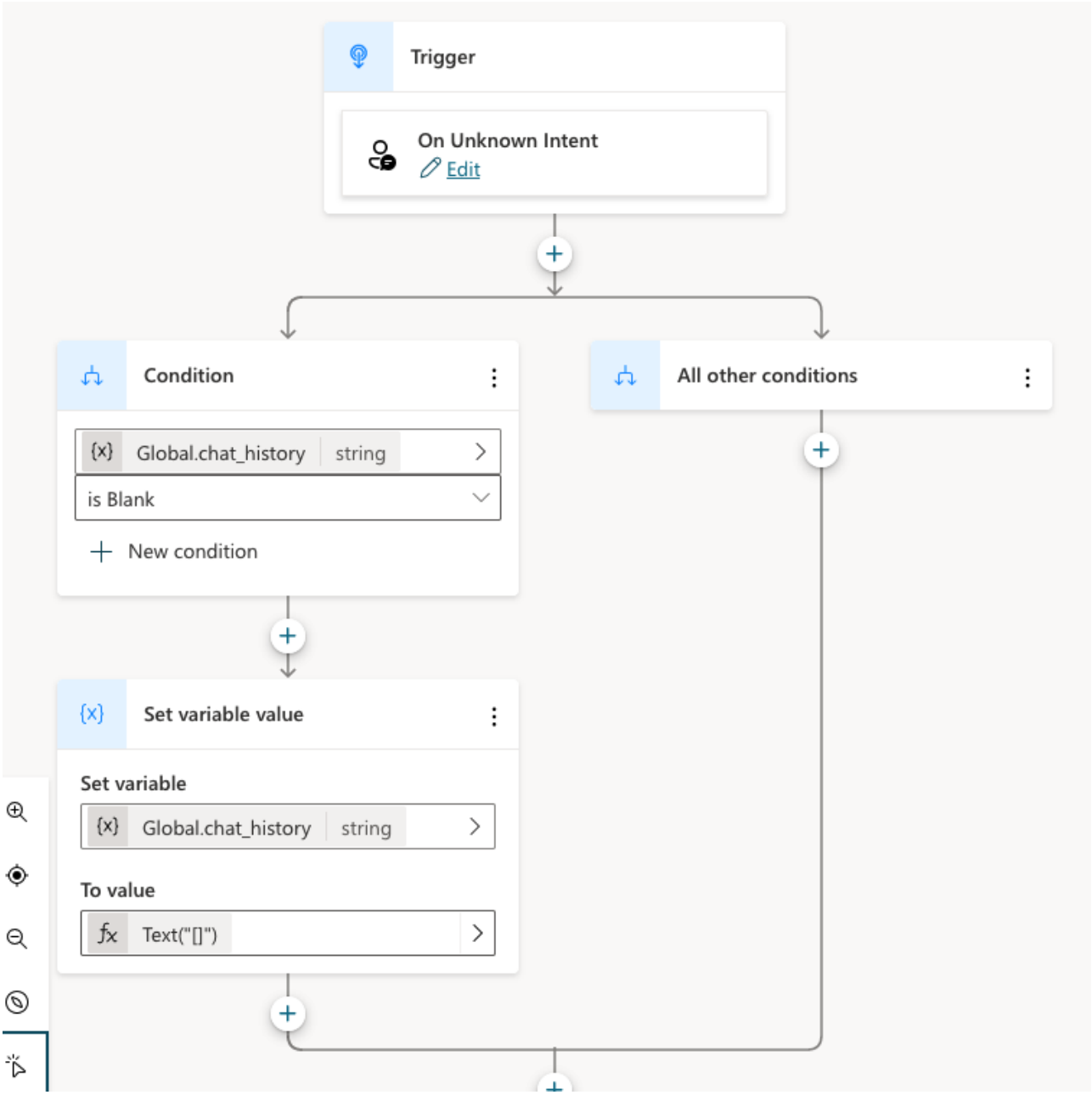


Your new bot will be created after a few moments.

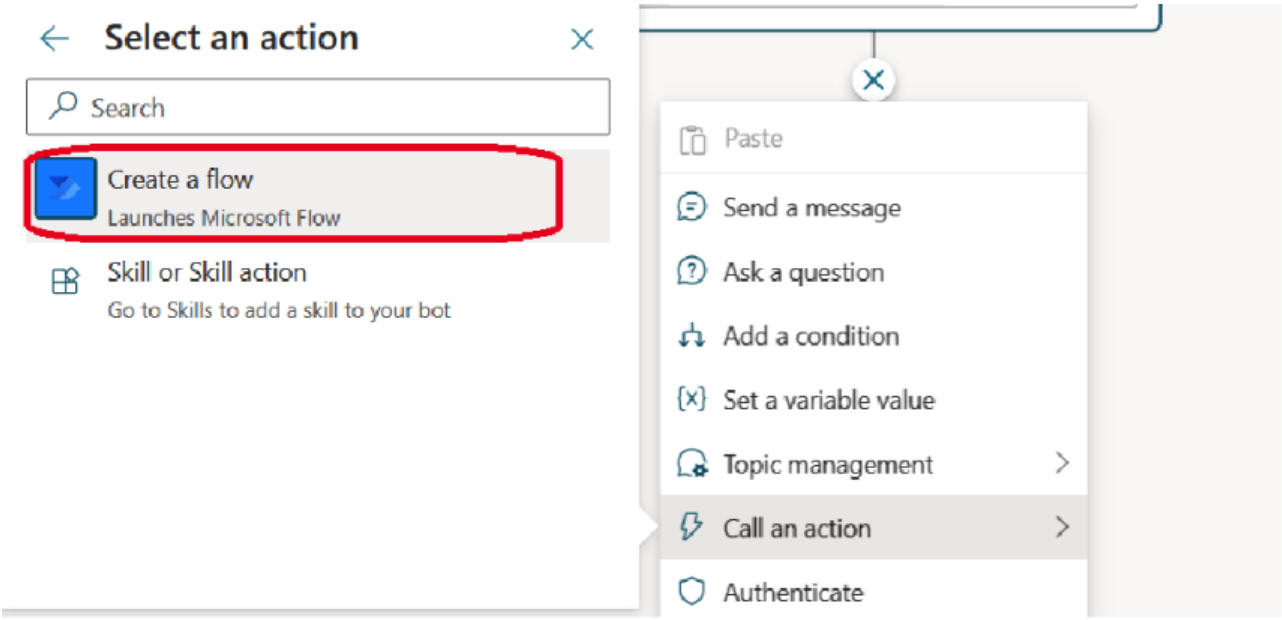
3. Once it's created, you will land on the following page.



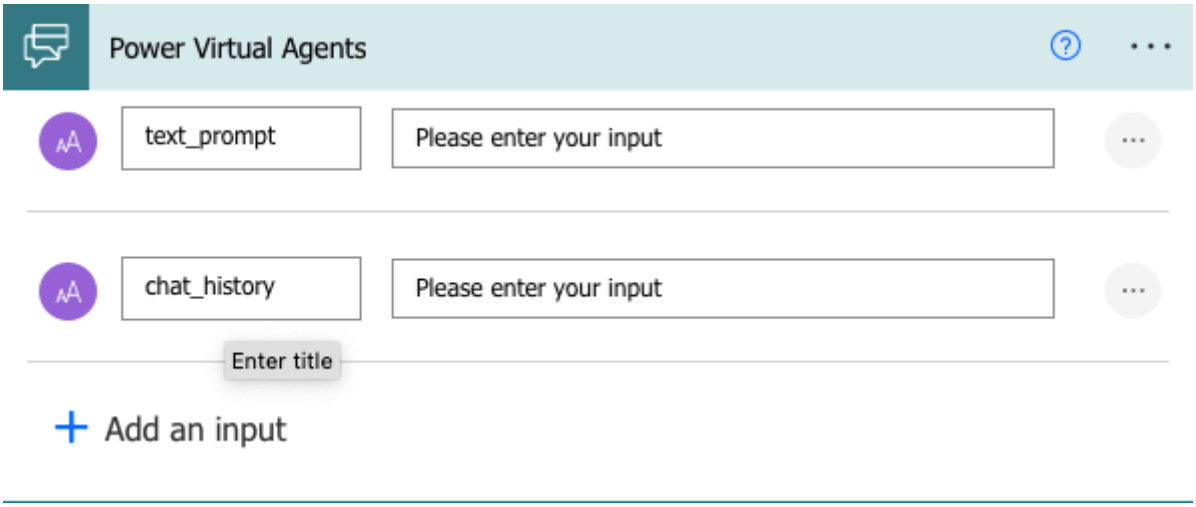
4. On the Topics menu, select "System", and select "Fallback". We need to create a global variable for the chat_history like the following.



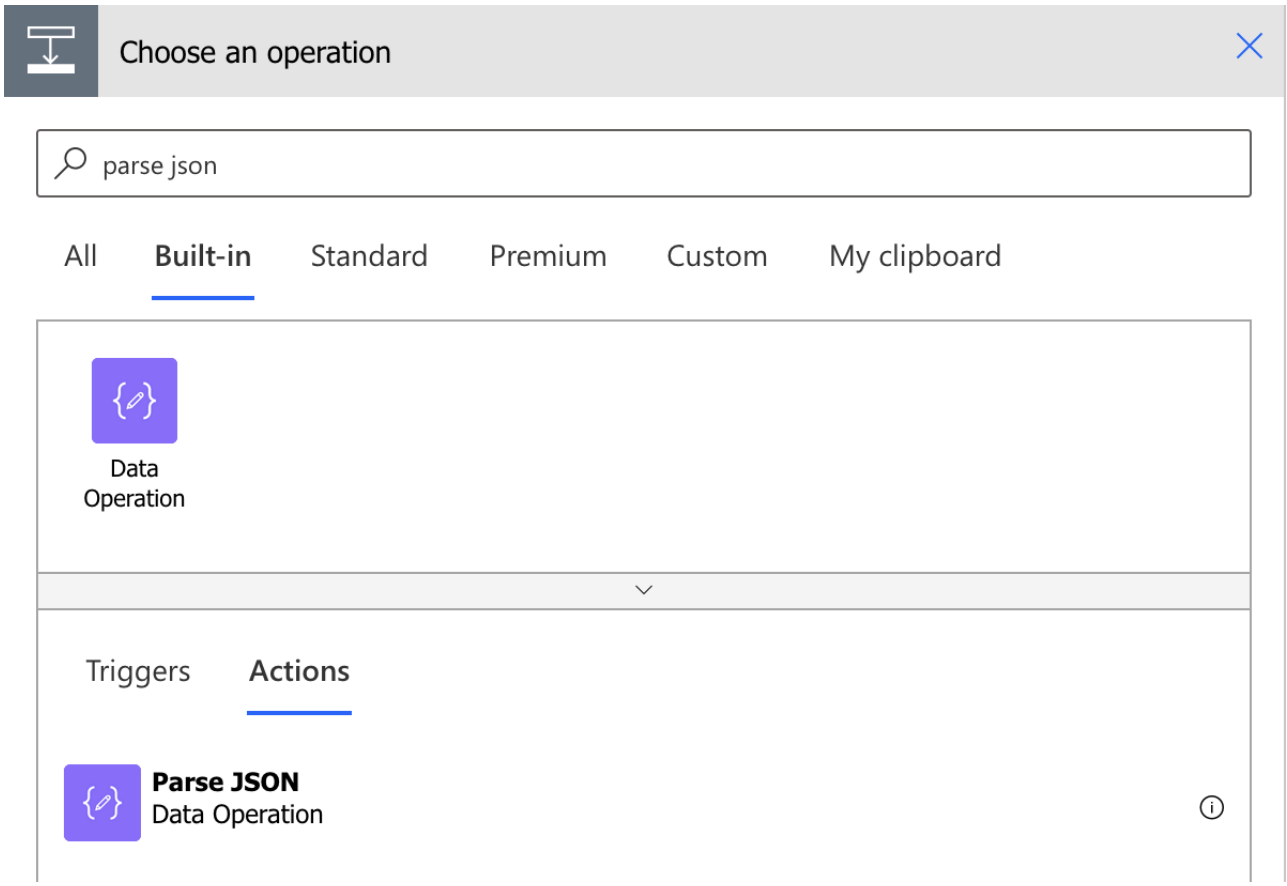
5. At the end of this workflow, click on the "+" symbol under the node you just added. Go to "Call an action > Create a flow".

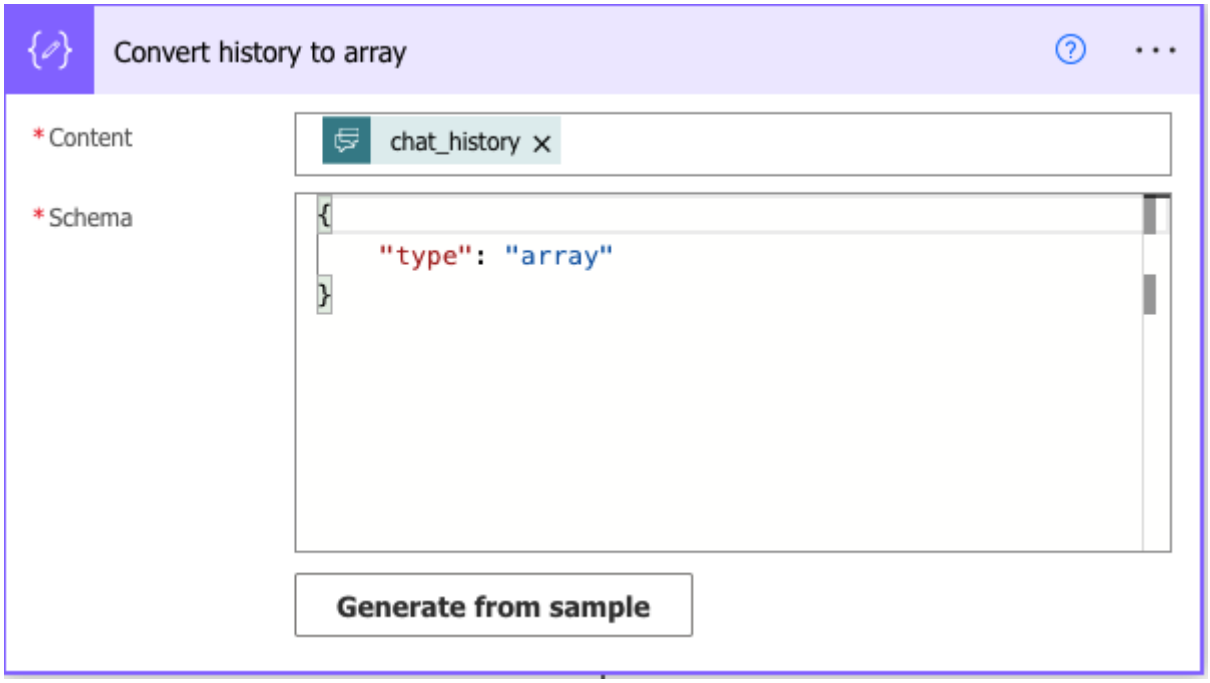


6. This action will redirect you to Power Automate, a service that helps you create automated workflows between your apps and services to synchronize files, get notifications, collect data, and more (find out more here: Get Started with Power Automate - Power Automate | Microsoft Learn), in order create your flow. On the top "Power virtual agents", add two inputs and provide a name for it, e.g. text_prompt and chat_history. Leave the values of the variables as-is.

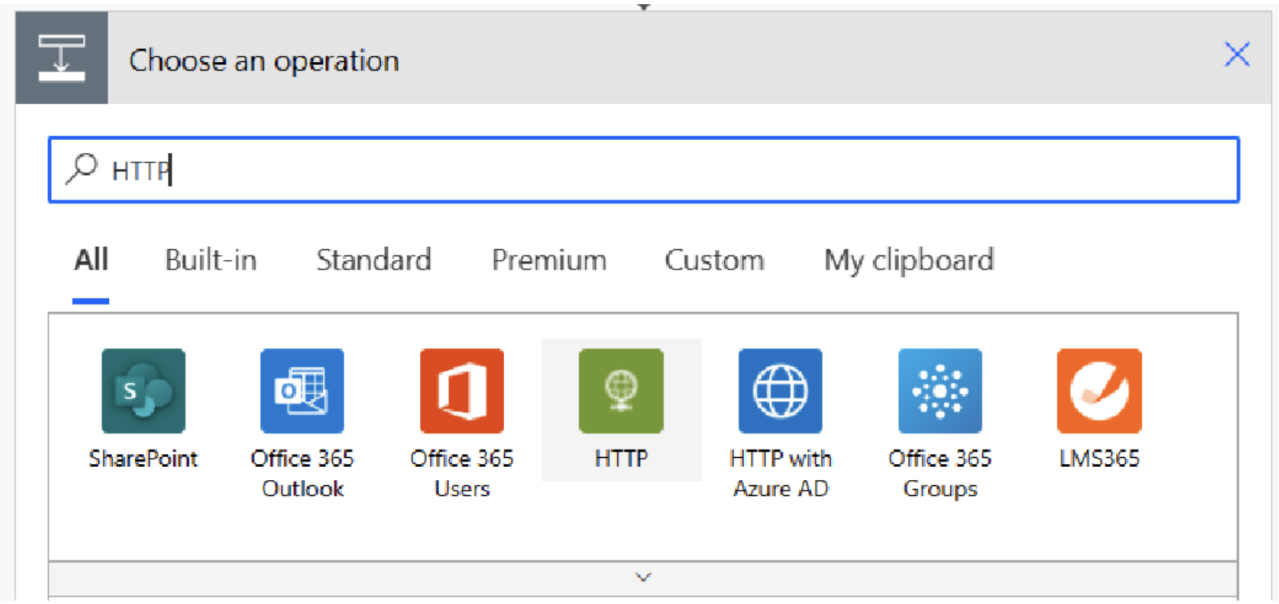


7. Since Power Automate inputs does not understand stringified arrays, we need to parse it before using it as an array. Search for "Parse Json". You can add the ""

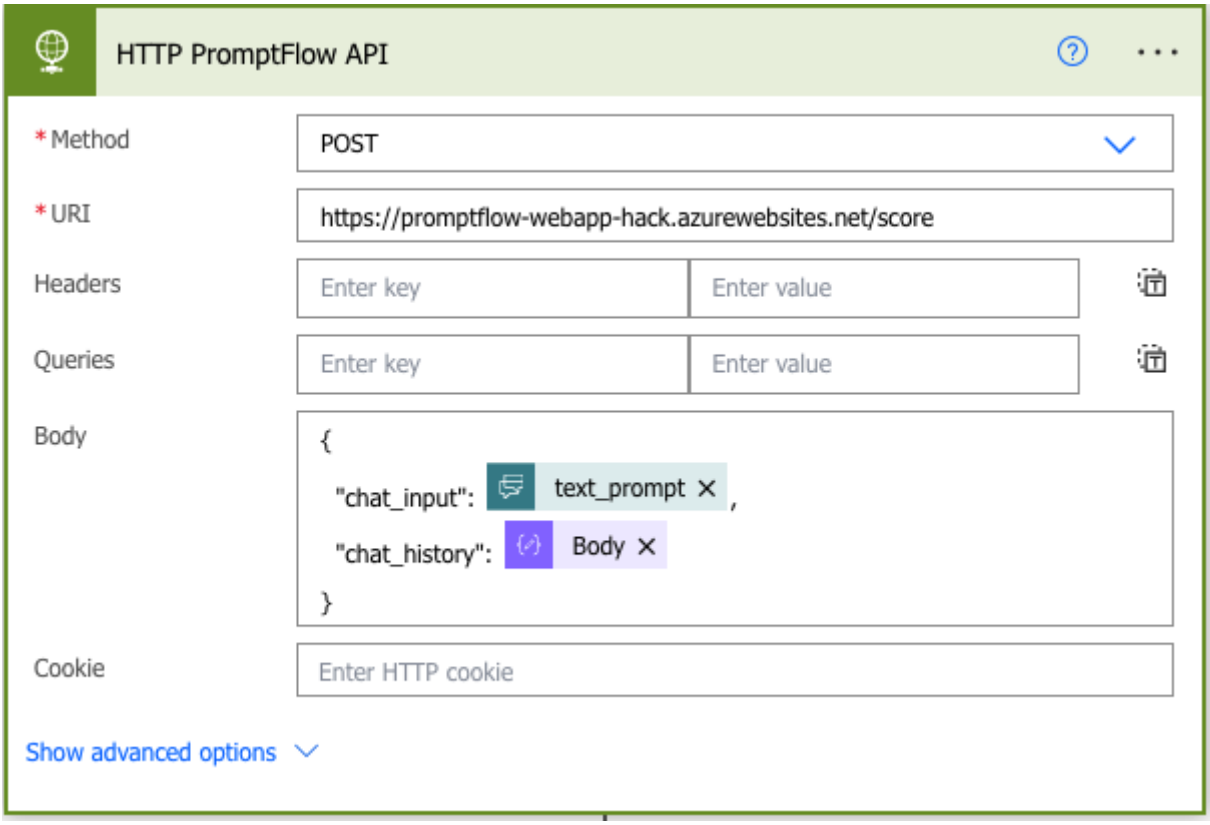




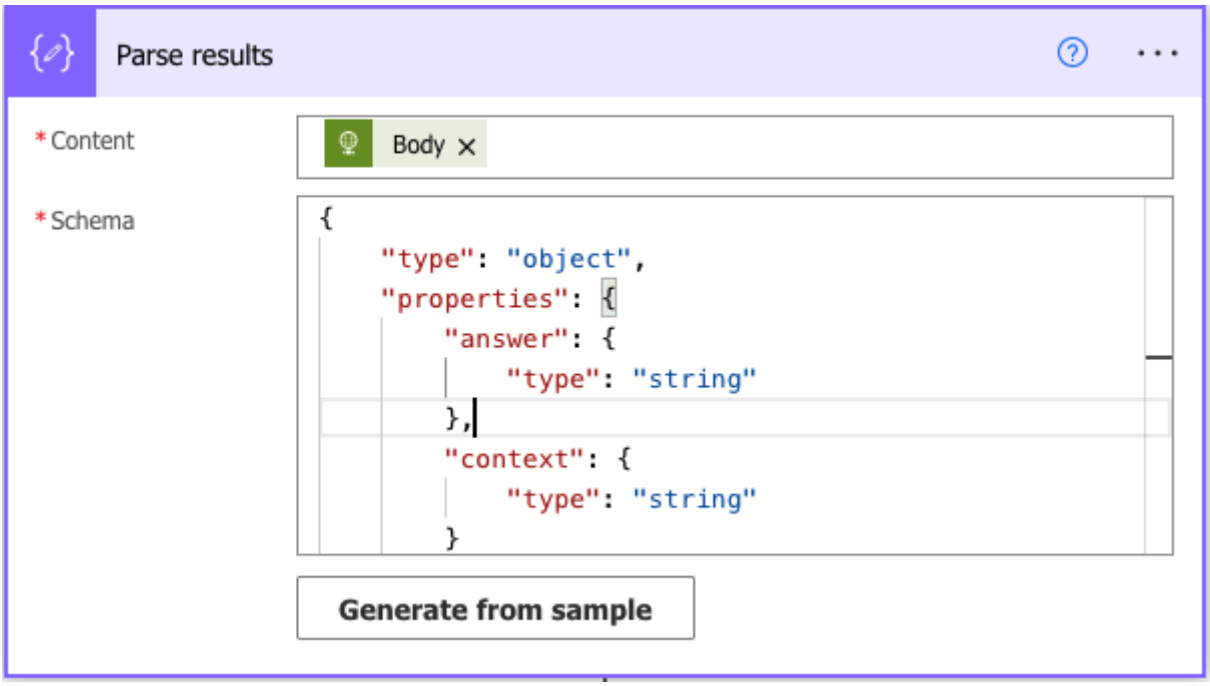
8. Click the + symbol after the node you just modified, select "Add an action", type "HTTP" and select the simple option available.



9. Populate with the PromptFlow details as in the image below. Make sure the "Body" and the text_prompt variables are specified.

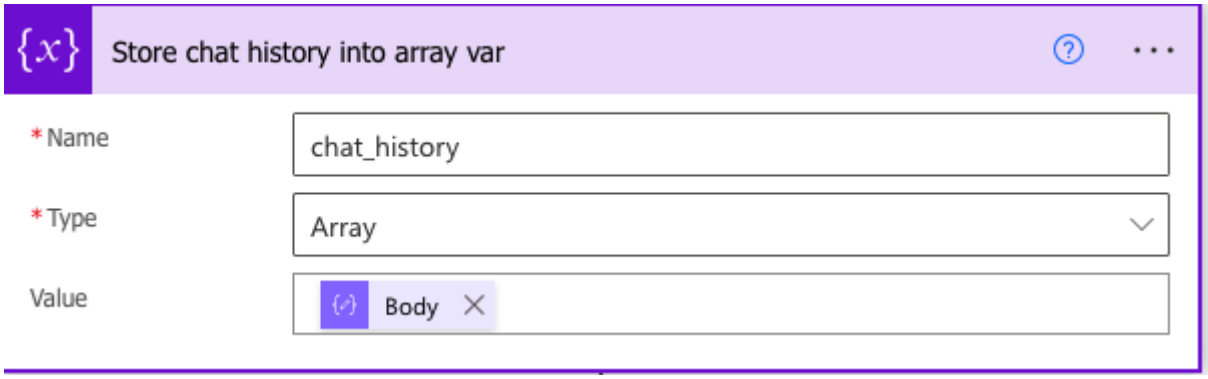


10. Next, click on the "+" icon to add a new action and search for "Parse Json". As in the previous steps, add the "Body" variable in the content. For the Schema, add the following (you don't need to click on "Generate from sample" option):

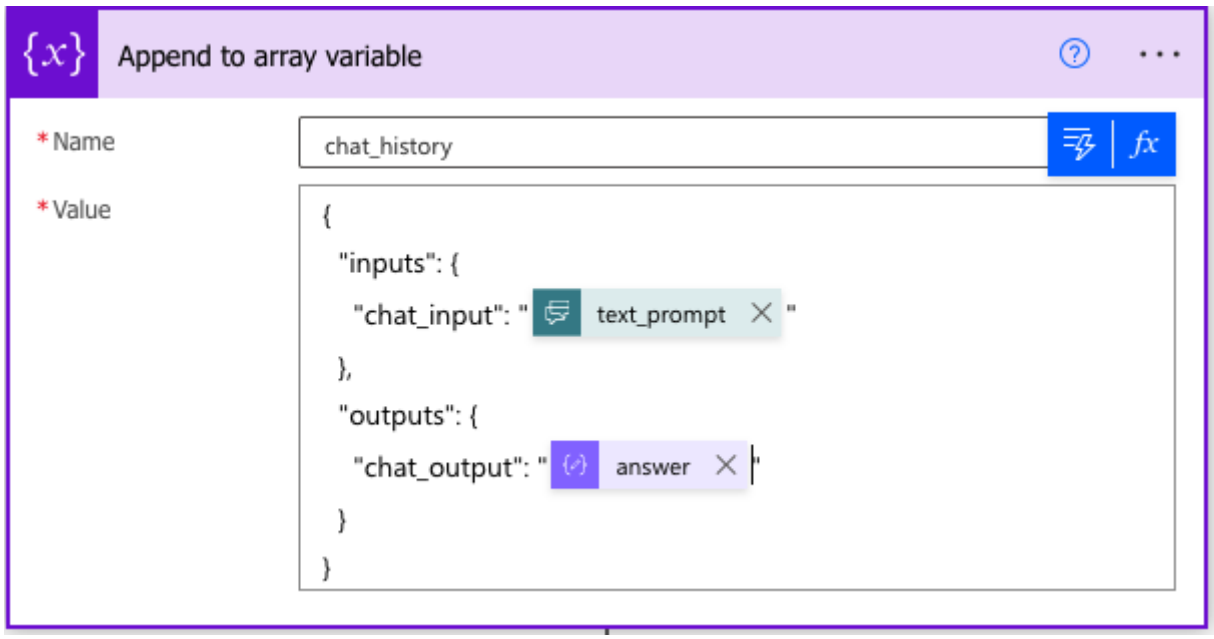


11. In order to build up the chat history, we need to perform a few variable operations.

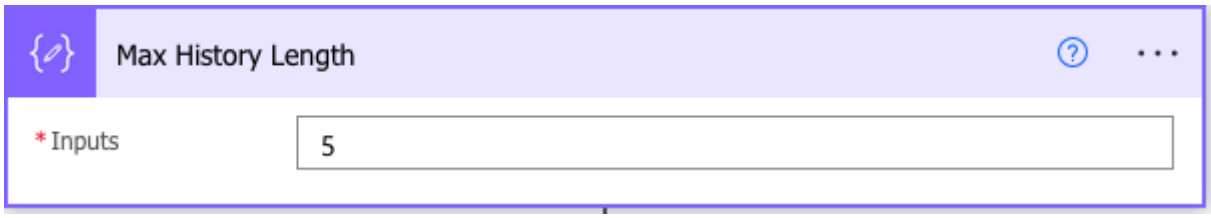
    Search for initialize variable and cast the array variable from "Convert history to array" output



Search for append to array variable using the text_prompt input and the Parse Json output



Search for compose and initialize the max history length constant

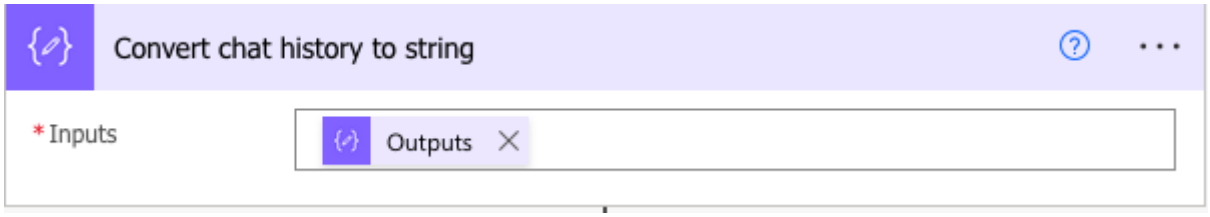


Search for compose and initialize the max history length constant. Use `length(variables('chat_history'))`

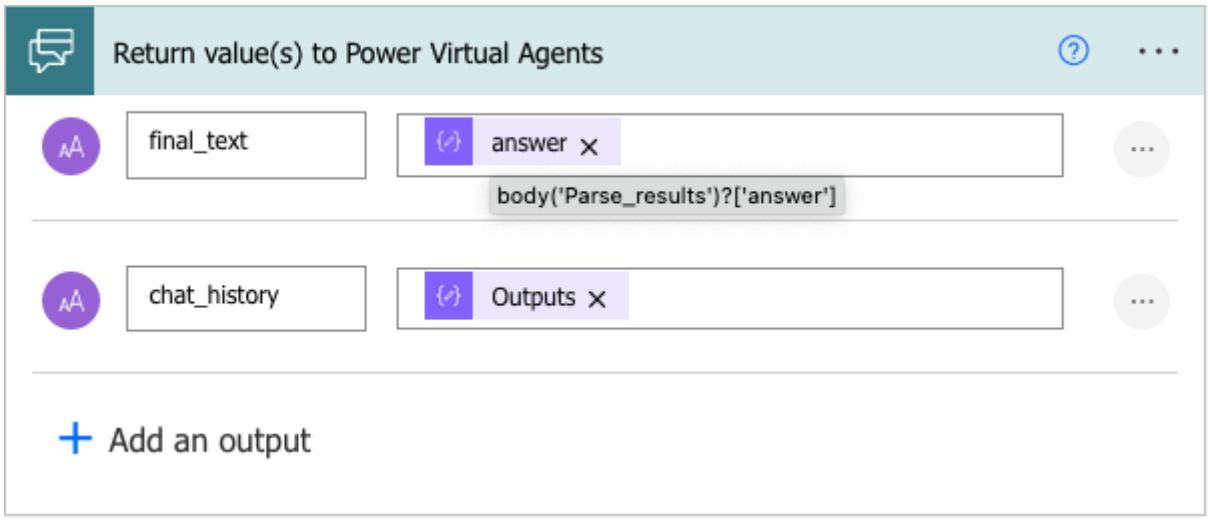
Search for compose and calculate the number of items to skip. Use `max(0, sub(outputs('Current_History_Length'), outputs('Max_History_Length')))`

Search for compose and skip the number items of the previous image. Use `skip(variables('chat_history'), outputs('Skip_items'))`

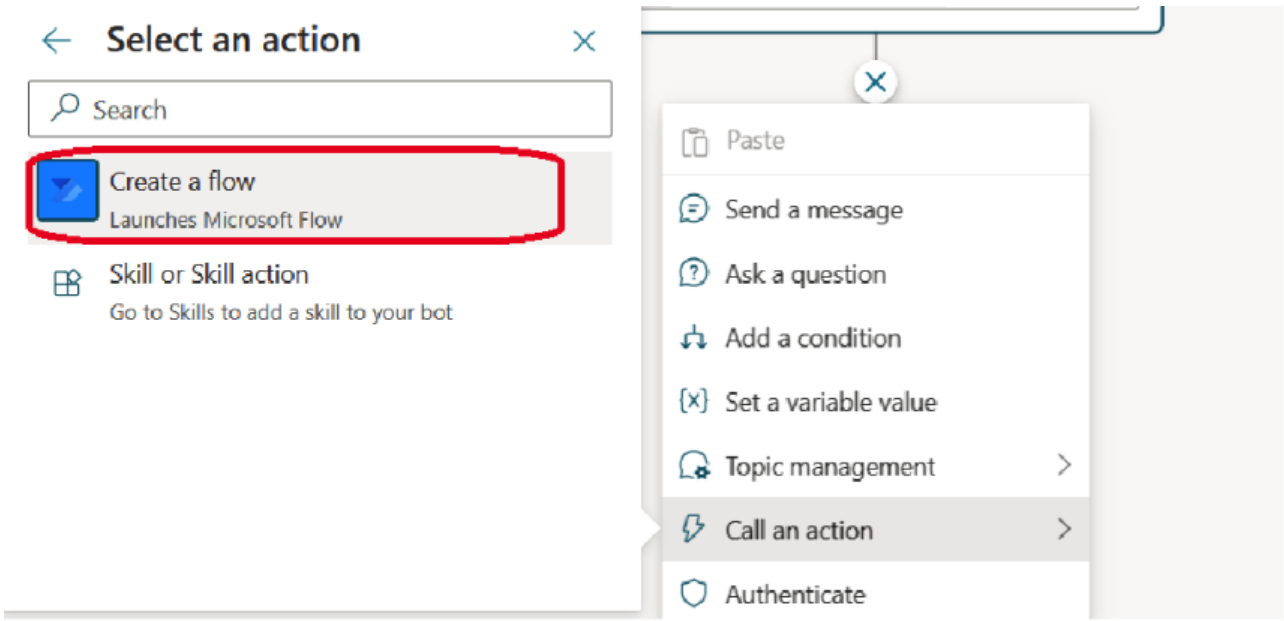
Search for compose and create a new array for chat_history.



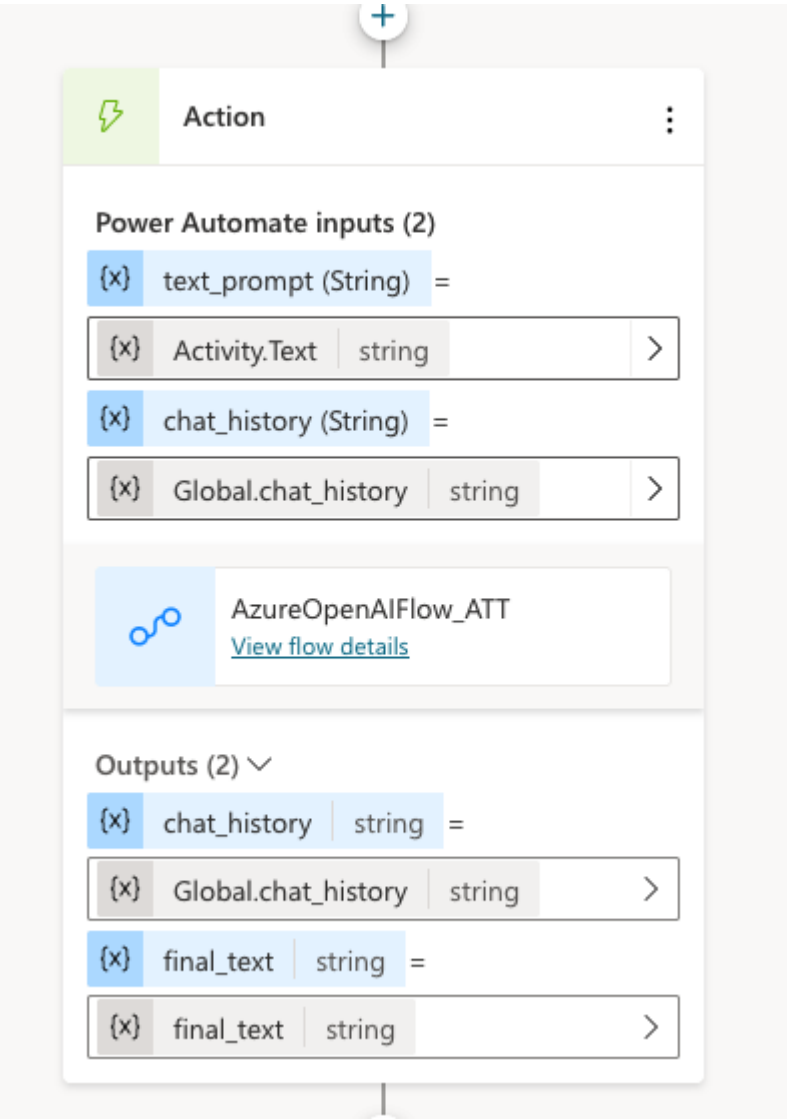
12. The final step is to write the output of the REST API request into a variable, in this case we called it "final_text". Click on the text field for assigning the value, right next to it. Do the same with "chat_history" using last output. A new window will pop-up.



13. Now, coming back to Power Virtual Agents page, in fallback topic sets the inputs and outputs to call the workflow.



14. Print the "final_text" message at the end of the workflow.



Let's try the bot

Click on the Test your bot option on the bottom left of your screen. You can ask whatever you want. The following screenshot shows an example of a conversation.