Microsoft

Copilot Studio and Azure AI Workshop

Lab 2: Copilot studio Topics and Actions

Hands-on Lab Step-by-Step Guide

April 2025

Lab Overview and Pre-requisites

Learning Objectives

This lab is designed to enhance the Sales Buddy agent created in LAB 1, by adding **Actions** and **Topics**.

In Microsoft Copilot Studio, **Topics** define how an agent handles conversations. Each topic represents a specific subject or task, guiding the agent's responses to user inputs to ensure interactions are relevant and coherent.
Actions are integral components that enable your agent to perform specific tasks in response to user inputs or events.

Topics guide the conversation while **Actions** perform tasks. Topics can include actions as part of their conversation flow.

In this Lab, you will be creating an Action and Topic to perform 2 separate tasks

Lab 2a. Create new leads using a connector Action

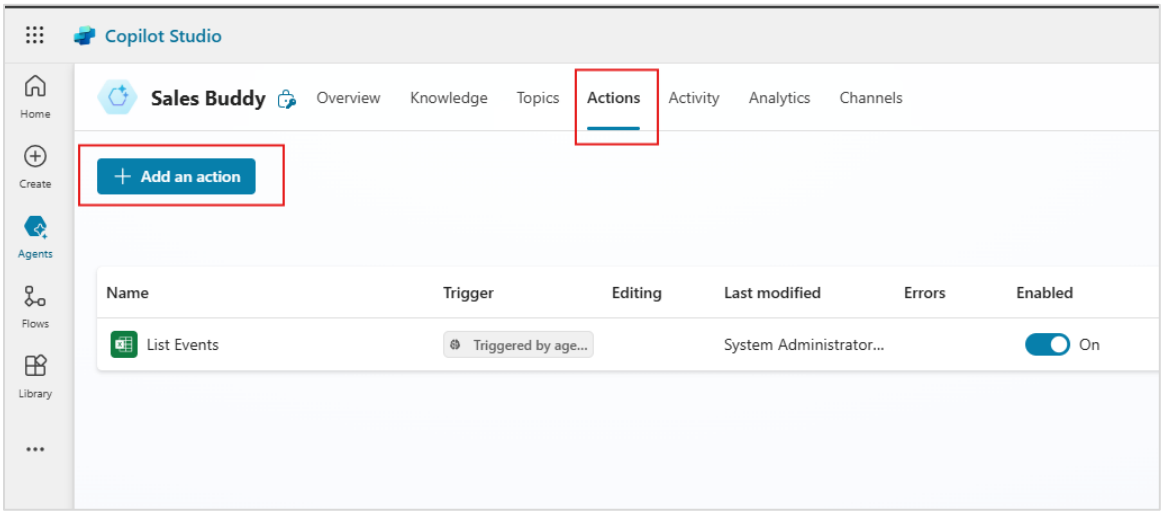
Lab 2b. Create new registration for customers/leads using a Topic + Action

Pre-requisites

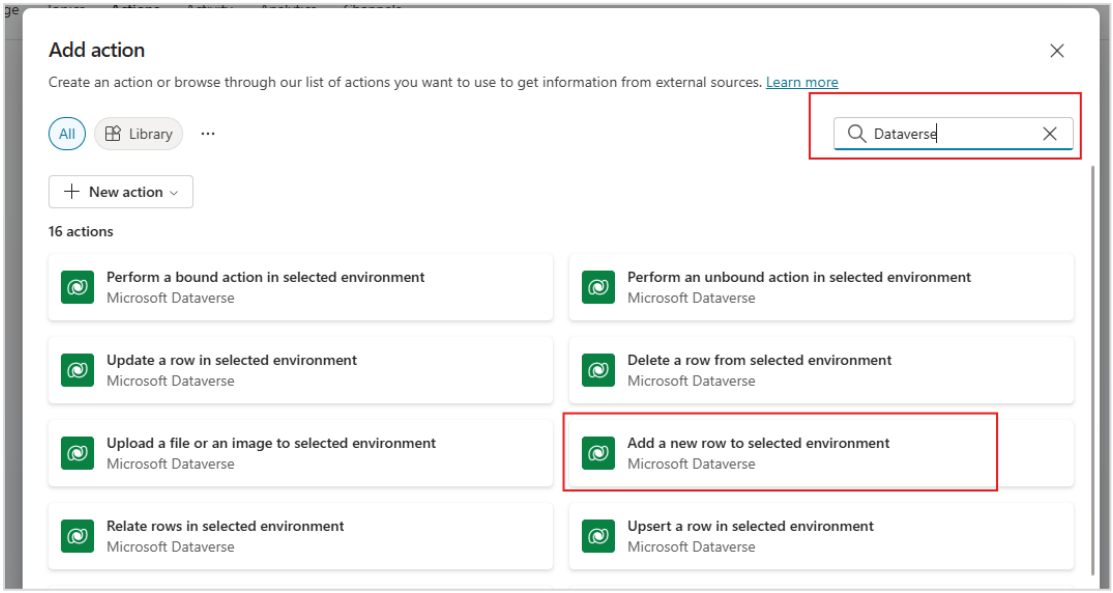
- You will need to have completed Lab 1 so you have the Sales Buddy agent created that we will be utilizing for this lab.

Lab 2a: Create new leads in CRM/Dataverse using a Connector Action

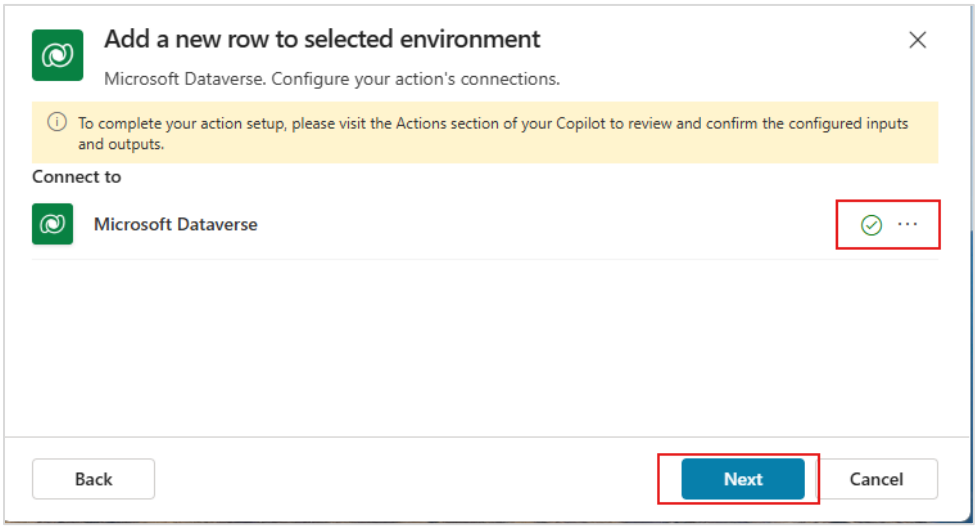
1. Open the Sales Buddy Agent created in Lab 1. Go to Actions Tab and click **+ Add an action**



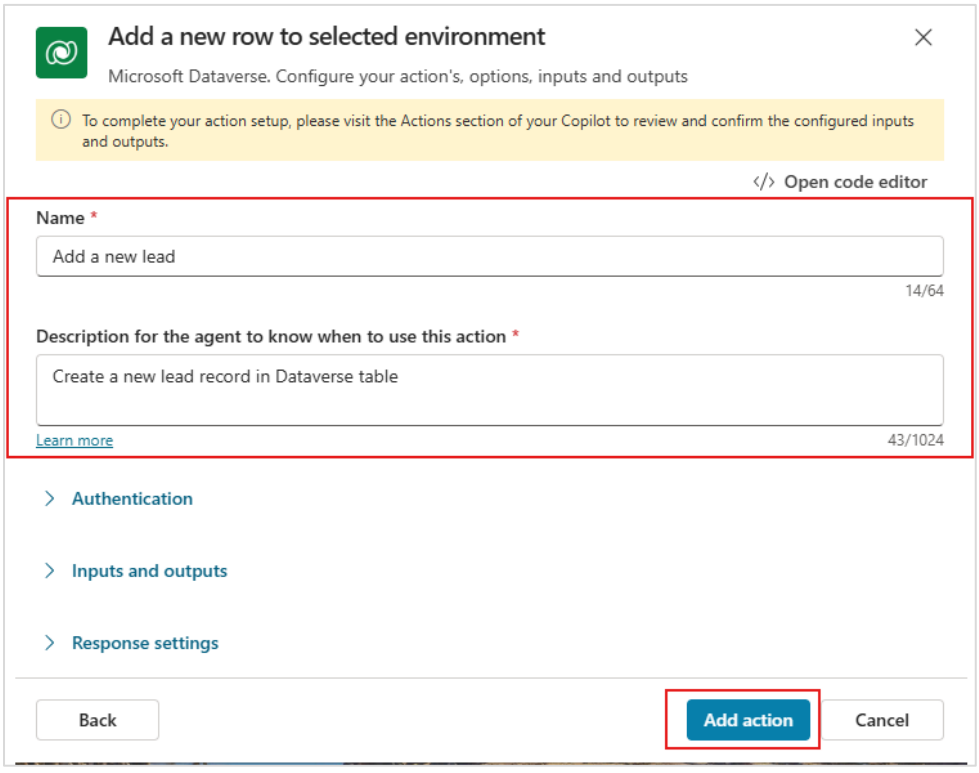
2. Search for **Dataverse** connectors and select **Add a new row to selected environment**



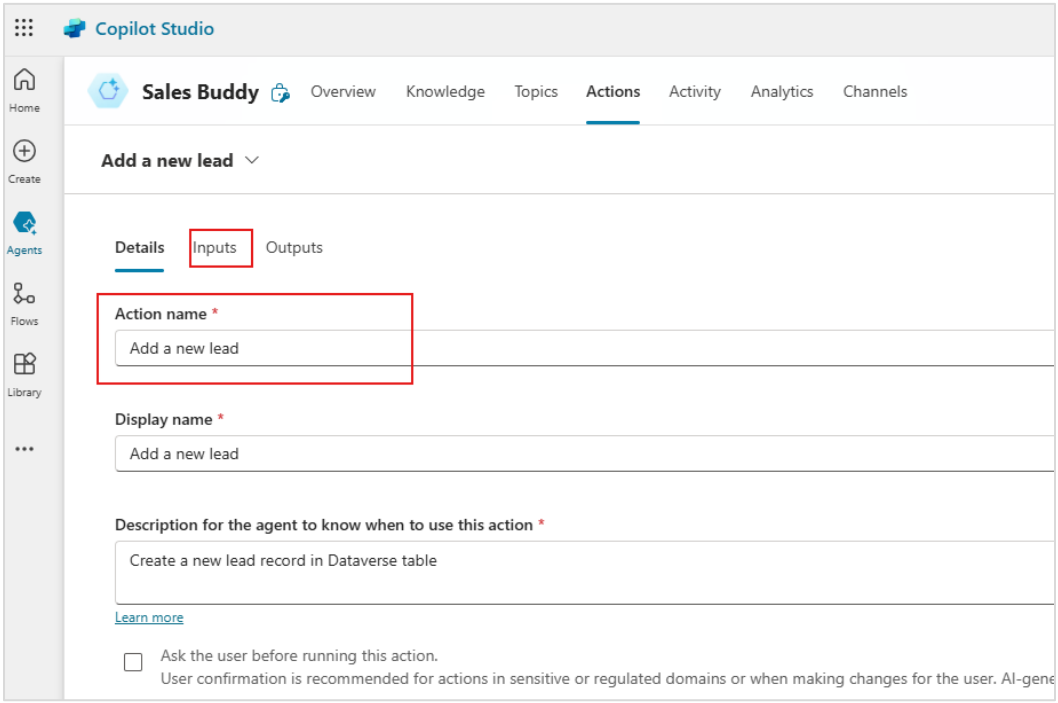
3. Ensure the connection is already established and click **Next**



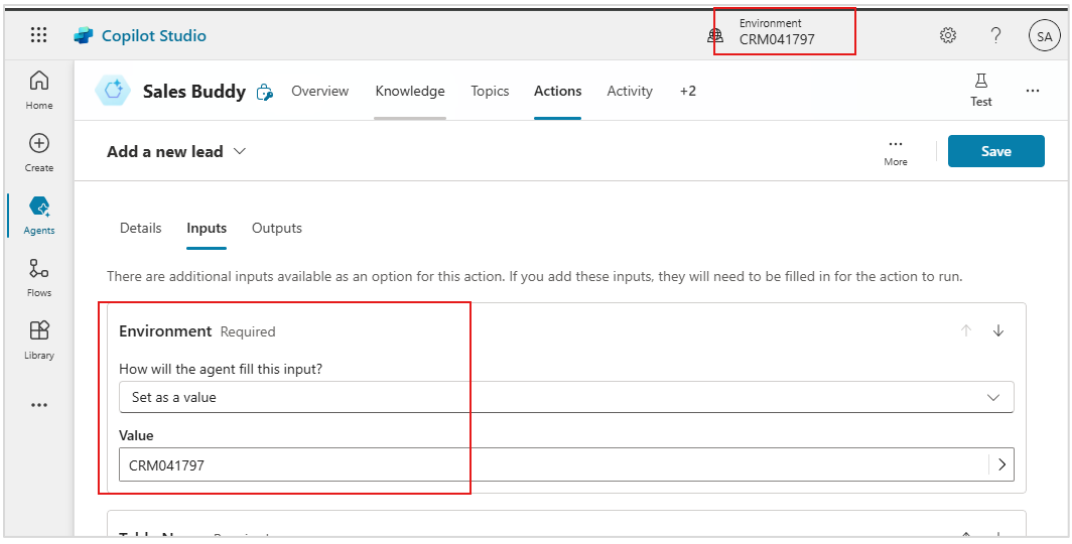
4. Update the following
   Name = Add a new lead,
   Description = Create a new lead record in Dataverse table
   and select **Add action**



5. You should find the newly added Action under the **Actions** tab. Open it again and update the **Action Name** = **Add a new lead.** Next click on **Inputs:**



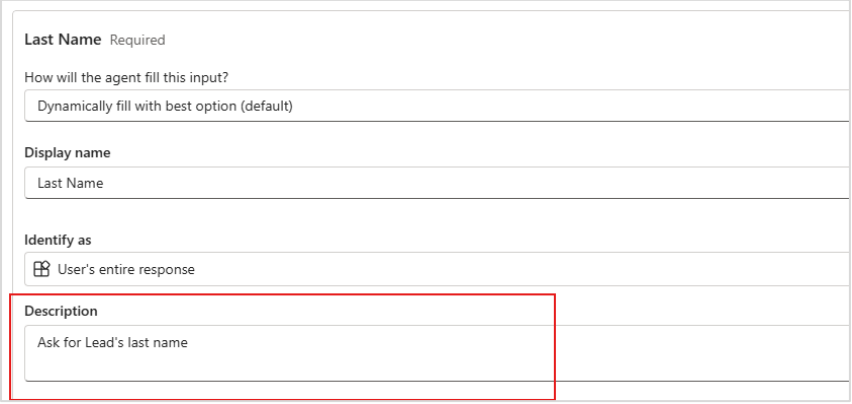
6. Under **Environment,** choose **Set as a value** and select your environment as **Value**



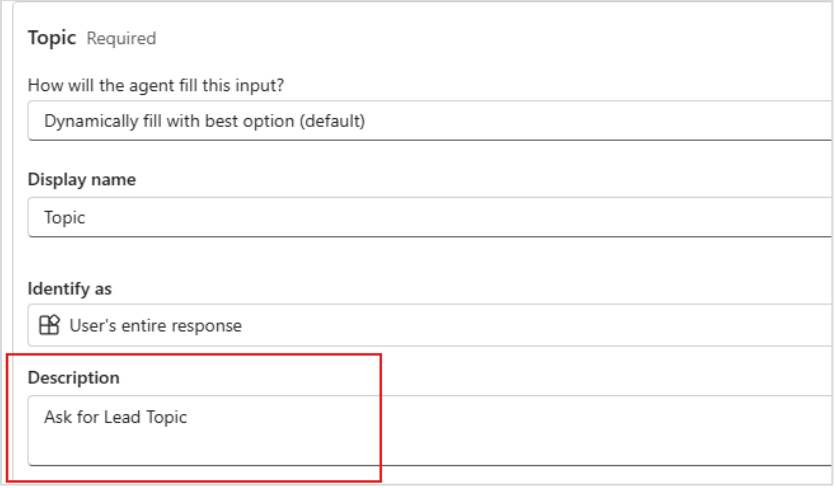
7. Under **Table Name**, choose **Set as a value** and select the **Value** as **Leads**

8. You will find 2 more required inputs for mandatory fields in Leads table below. For these 2 inputs, we will not set any value by default, but we'll describe how Agent can receive these values from the user.

   Under **Last Name,** update the **description** as – **Ask for Lead's last name**

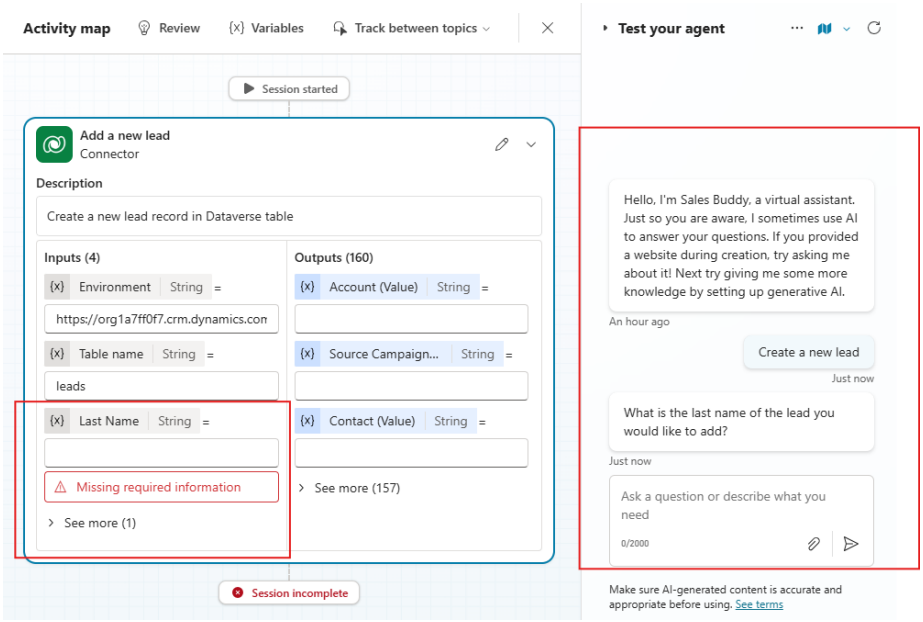


   Similarly, under **Topic,** update the **description** as – **Ask for Lead Topic**

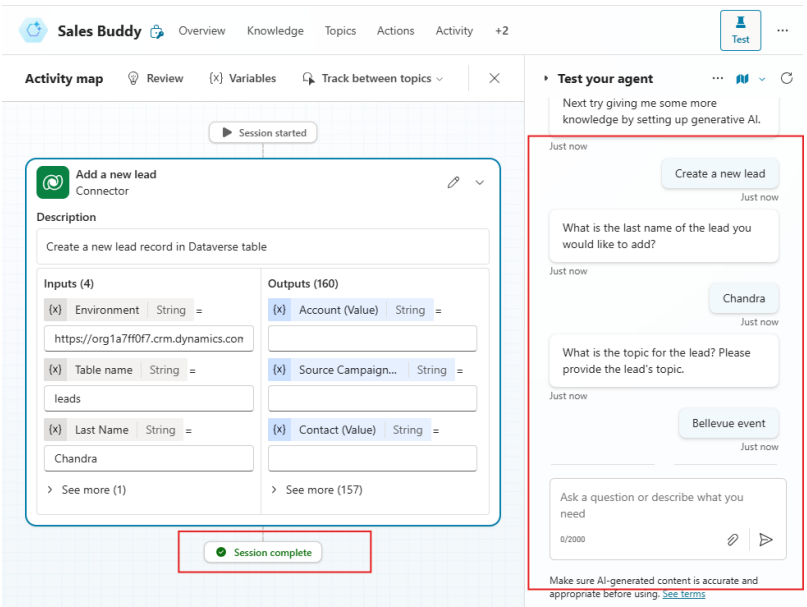


9. Click on **Save** to save the Action.
10. Now that our Action is configured with the required inputs, let's test this out. Open the **Test** window and send the following message:
    - Create a new lead

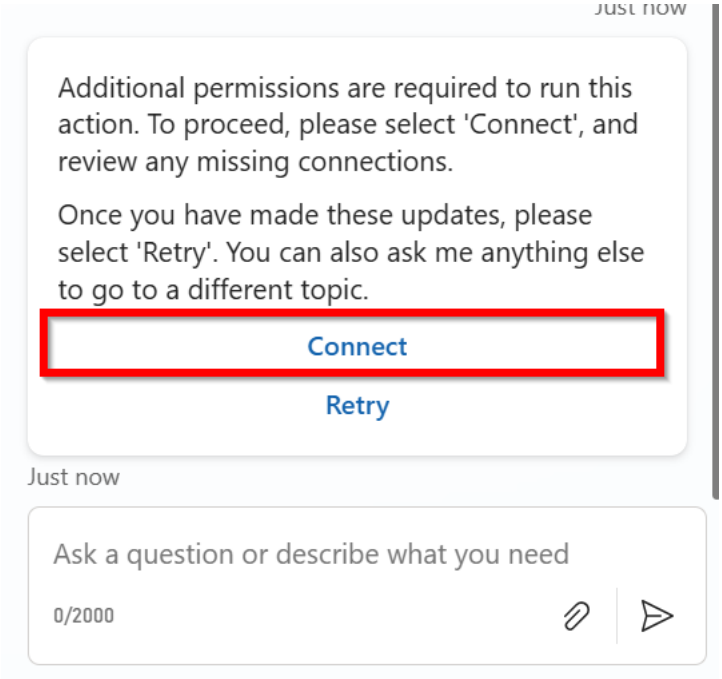
    You can find that Agent requests for missing information (Last Name and Topic) from the end user



11. Respond to the Agent query by providing the last name and Topic. Once it is complete, the new Lead record will be created in the Dataverse table.

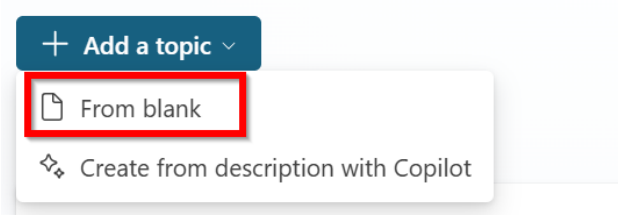


12. (Optional) You might need to create a connection for the first time. Click on Connect to follow the steps. Use the credentials from your demo tenant.

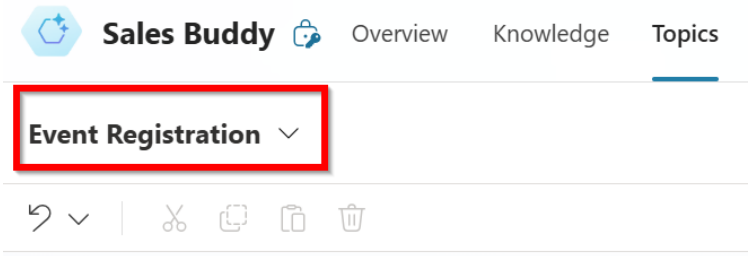


Lab 2b: Create New Event Registration using Topic

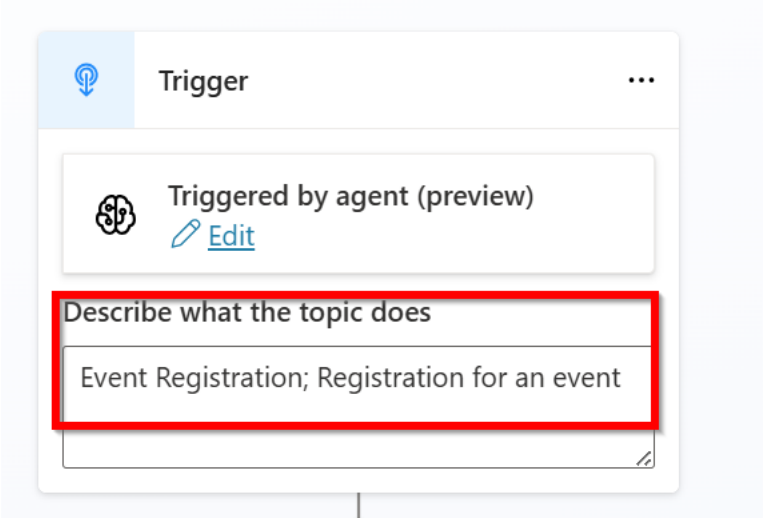
1. Go to the Topics Tab to add a topic From Blank.



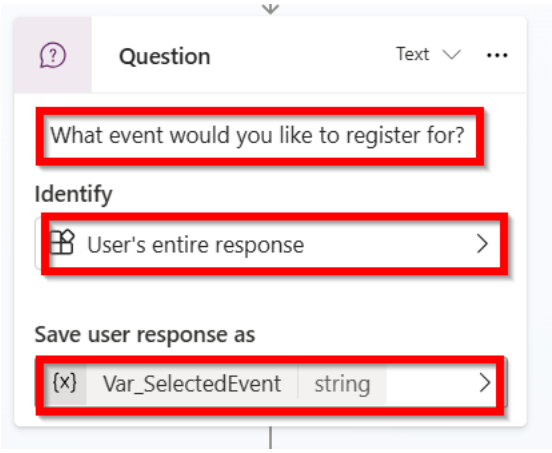
2. Rename the topic to "Event Registration"



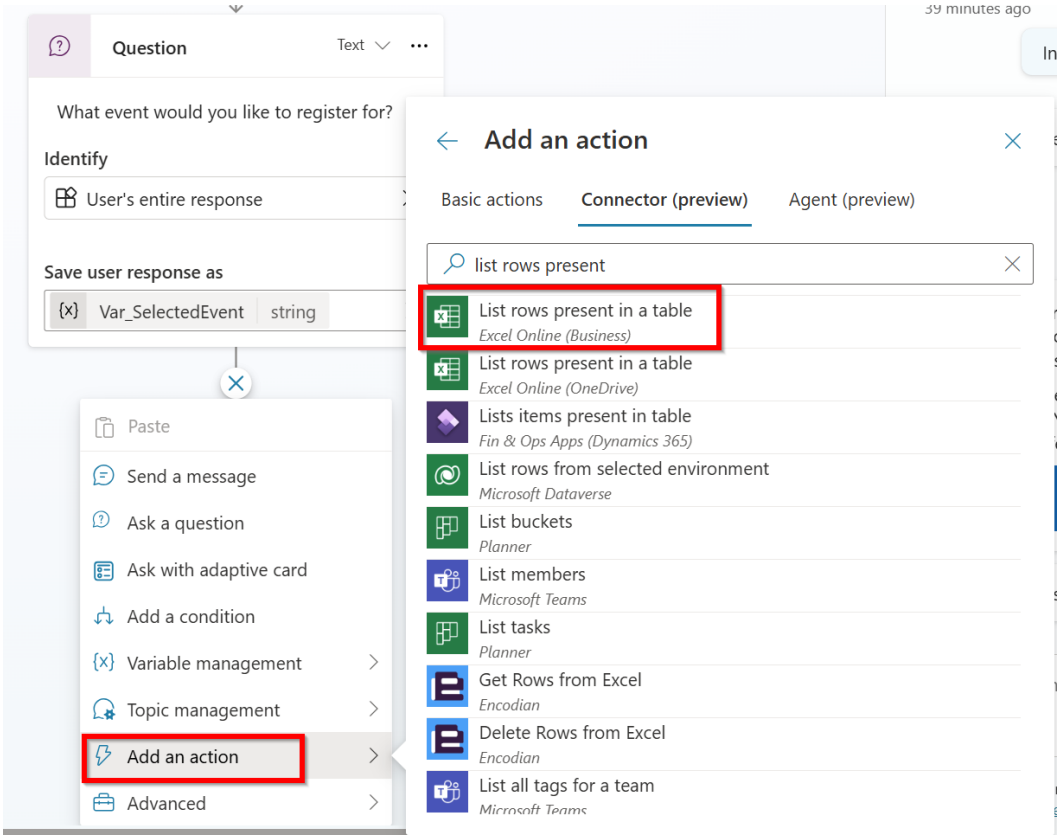
3. Enter trigger description "Event Registration; Register for an event"



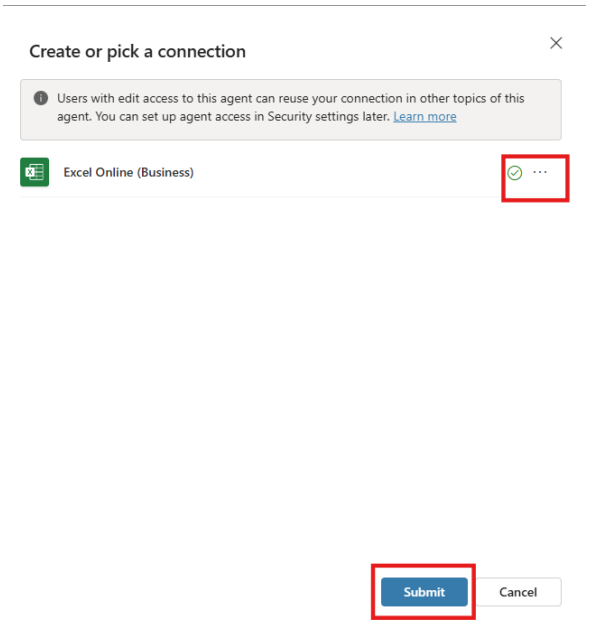
4. Add a node to ask a question. In the text field, enter "Which event would you like to register for?". In the Identify field, pick "User's entire response". In the Save user response as field, create a new variable renamed to Var_SelectedEvent.



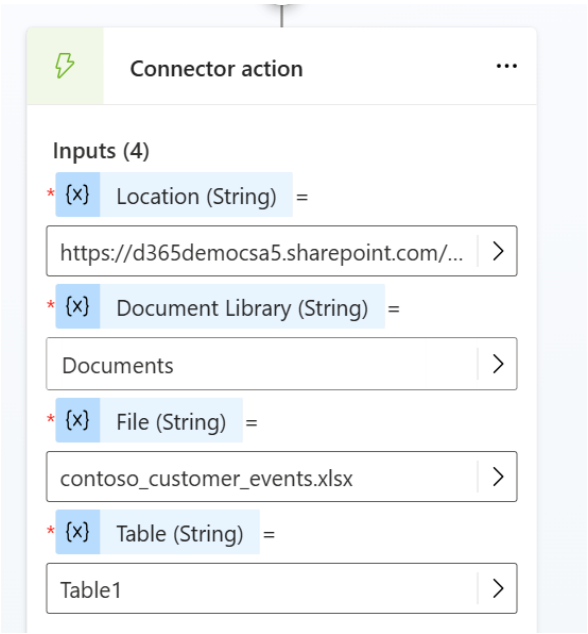
5. Add an action, go to Connector (preview) tab, and search for "list rows present in a table" connector.



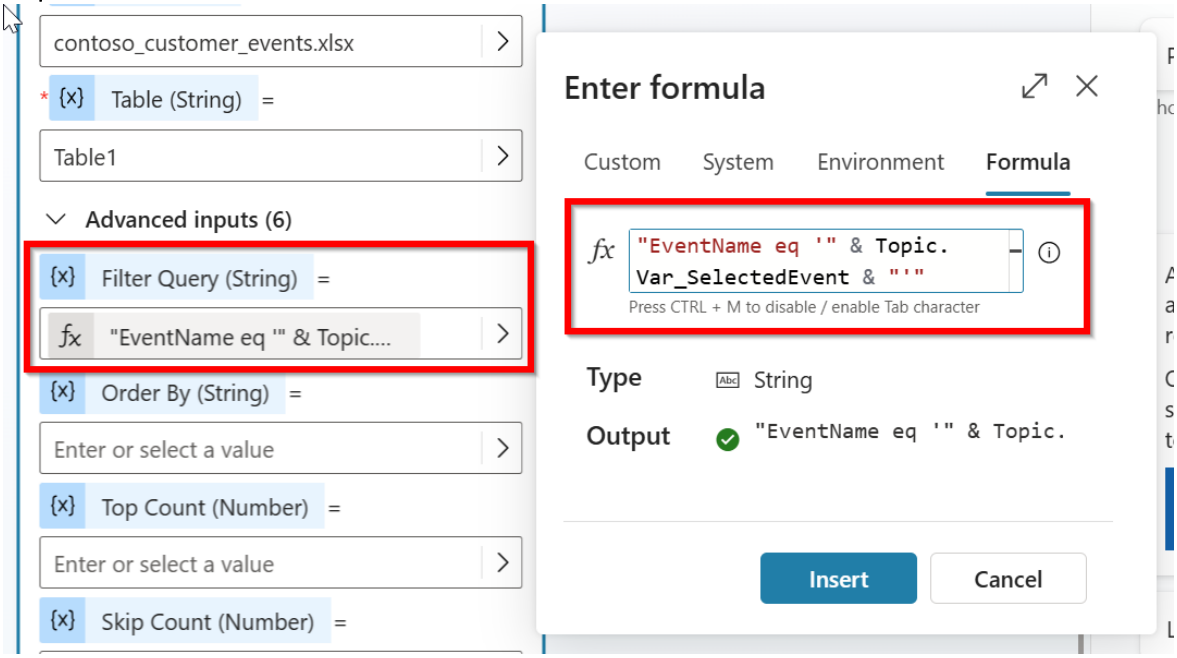
6. Create or pick a connection. Confirm you have the correct connection and click Submit.



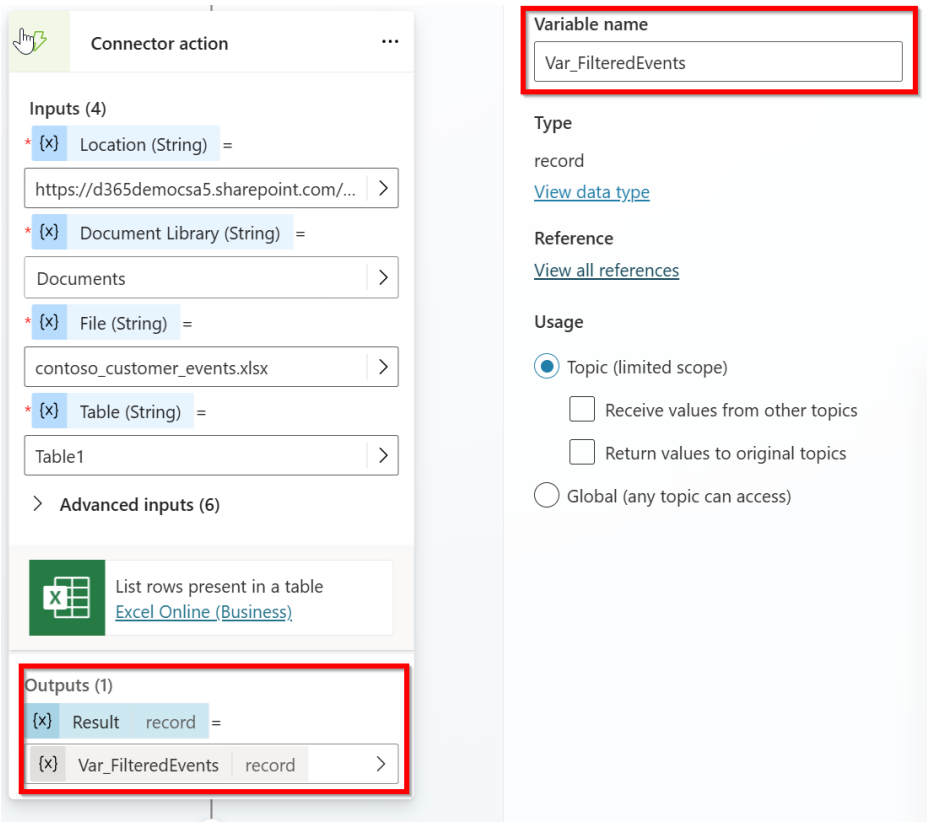
7. Fill out the input parameters. **Location** field is the SharePoint site root url in the format of https://yourenvironmentname.sharepoint.com/sites/SalesTeam/. **Document Library** field is Documents. **File** name field is contoso_customer_events.xlsx. **Table** field is Table1.



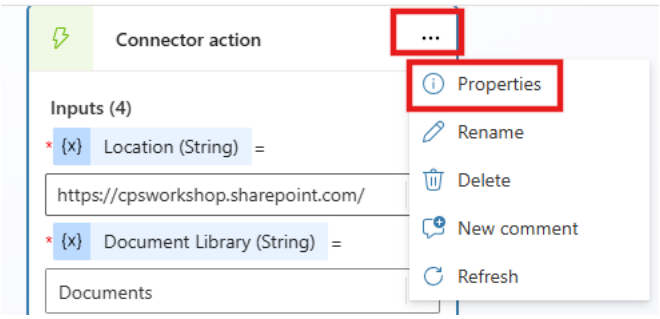
8. Fill out advanced settings **Filter Query** field. Enter the formula as **"EventName eq '" & Topic.Var_SelectedEvent & "'** then click **Insert.**



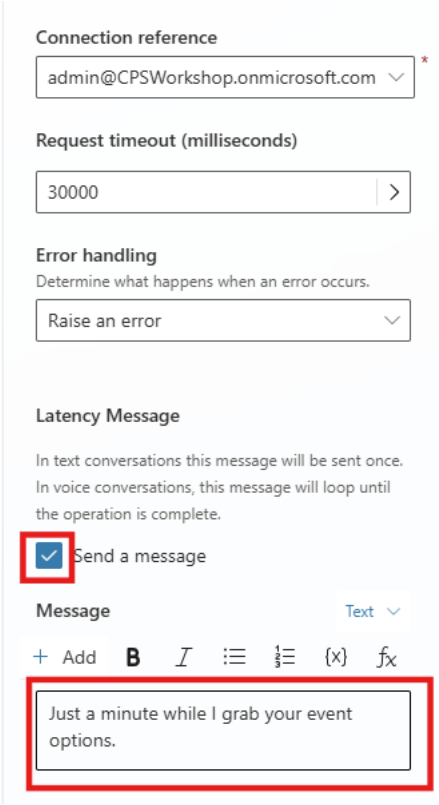
9. **Rename the output parameter** GetItems to "Var_FilteredEvents"



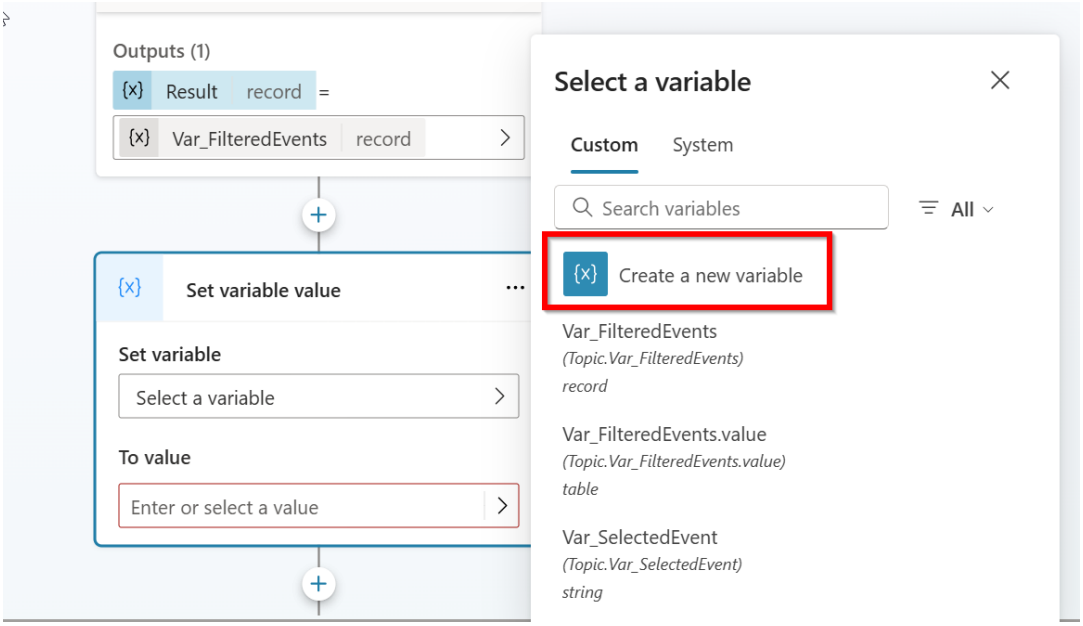
10. Add a latency message. Best practice when calling an action that may take some time is to set a latency message so the users knows the agent is working in the background. In the Connector action, click the **(...)** and select **Properties.**

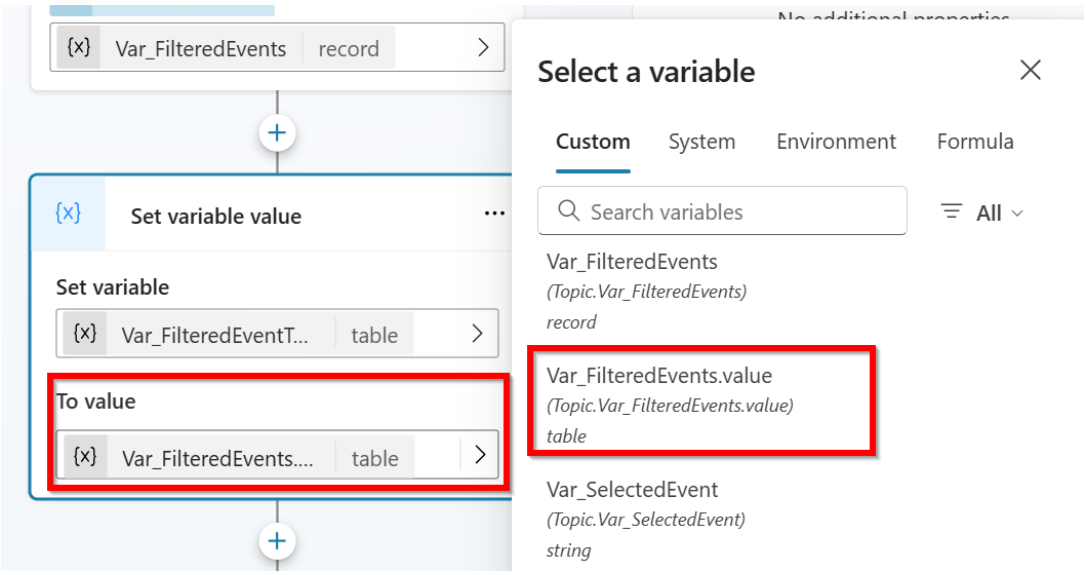


11. Scroll down to **Latency Message** and click the **Send a message** box. Add the following message - **Just a minute while I grab your event options.** Then **close** the Properties box.

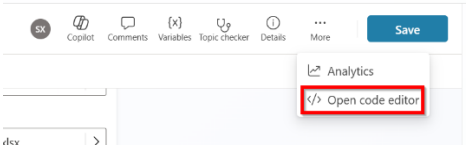


12. Add a variable management node , pick set a variable value. In the Set variable dropdown, create a new variable named "Var_FilteredEventText". In the To value dropdown, pick the existing variable "Var_FilteredEvents.value" that is the table.





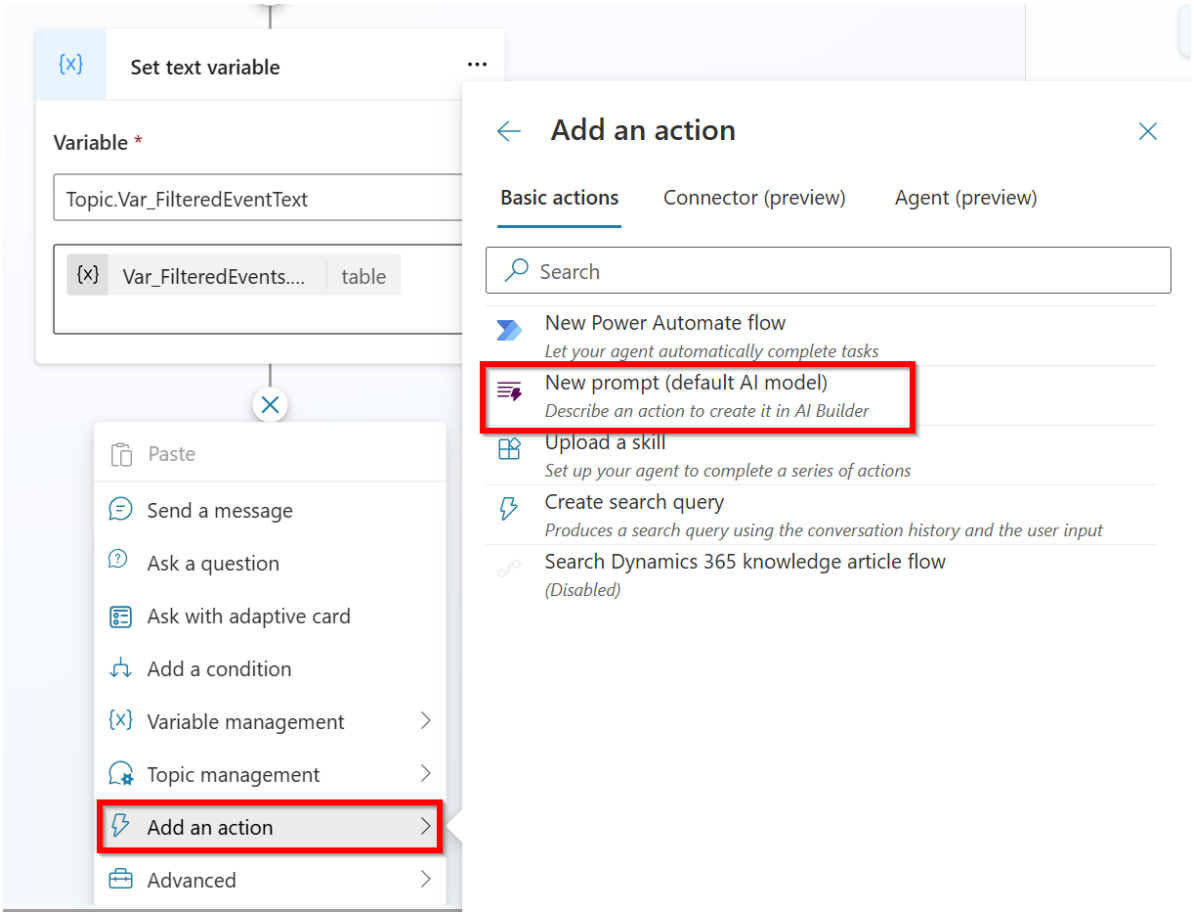
13. Change the variable **Var_FilteredEvents.value** type from **table** to **text** by using **code editor**. In the upper ribbon, click on **More** and select **Open Code editor**.



14. Scroll to the bottom of the code editor, change **kind** to SetTextVariable, change **value** to "{Topic.Var_FilteredEvents.value}". **Save** and **close** the code editor

```
- kind: SetTextVariable
  id: setVariable_YoHLZ5
  variable: Topic.Var_FilteredEventText
  value: "{Topic.Var_FilteredEvents.value}"
```

15. **Add an action**, pick **New prompt (default AI model)**



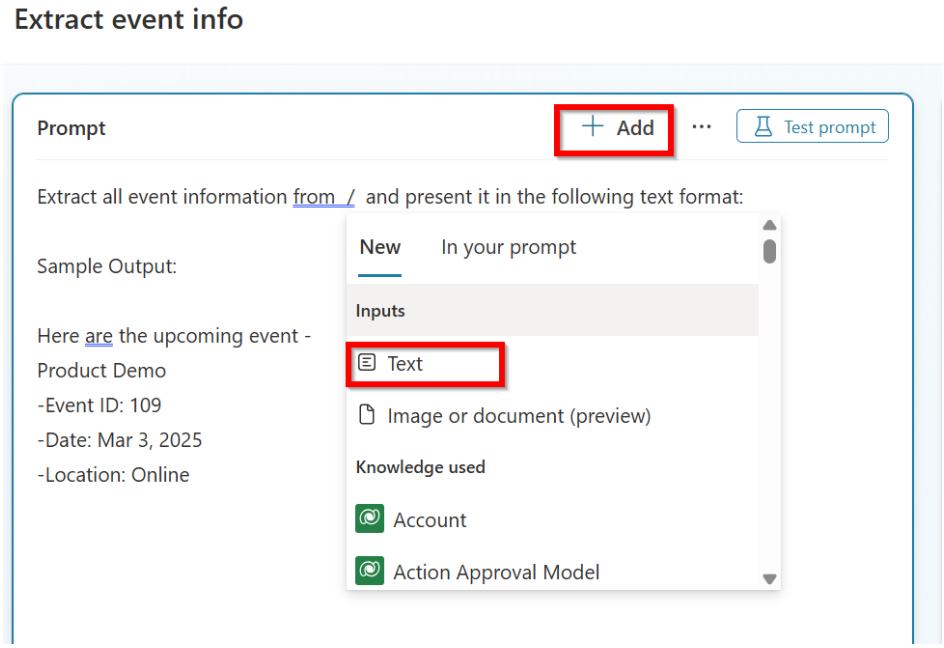
16. Name the prompt "**Extract event info**". In the Prompt, enter the following and Save.

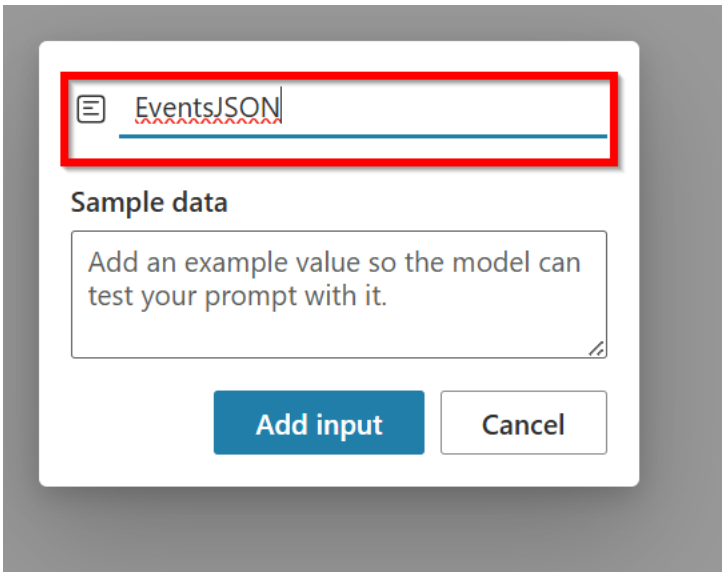
    Extract all event information from EventsJSON *and present it in the following text format:*

    Sample Output:

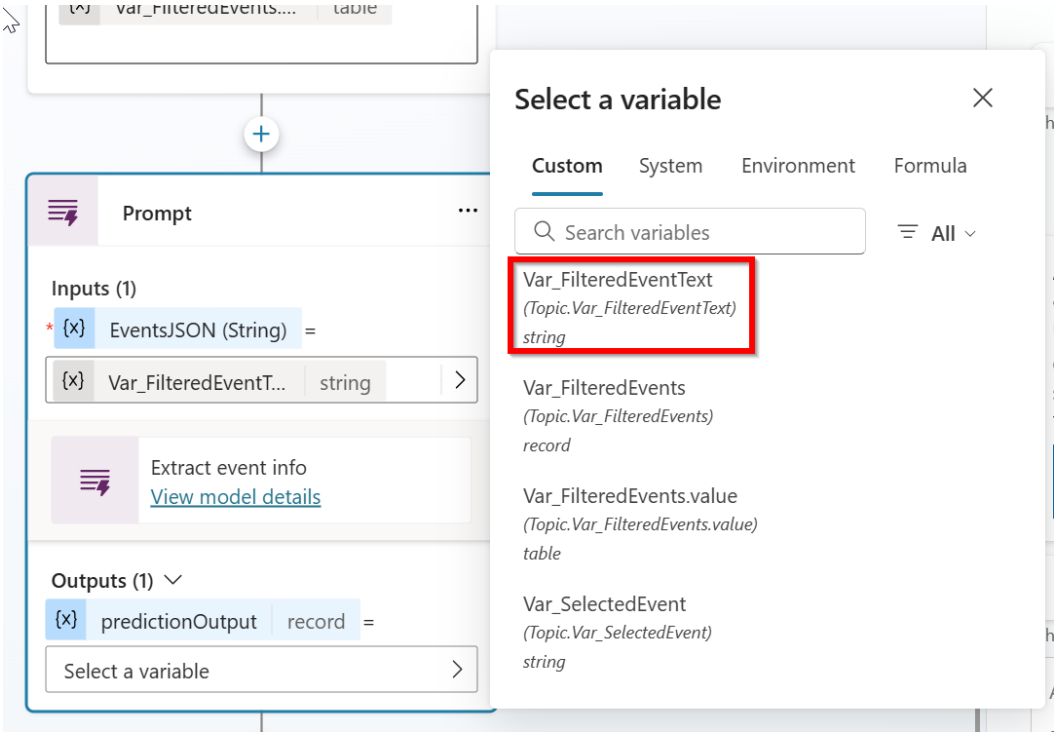
    Here are the upcoming events –
    Product Demo
    -Event ID: 109
    -Date: Mar 3, 2025
    -Location: Online

    EventsJSON is added by clicking on **+Add** button, pick **Text** under the **Inputs**. Enter the Name as "EventsJSON"

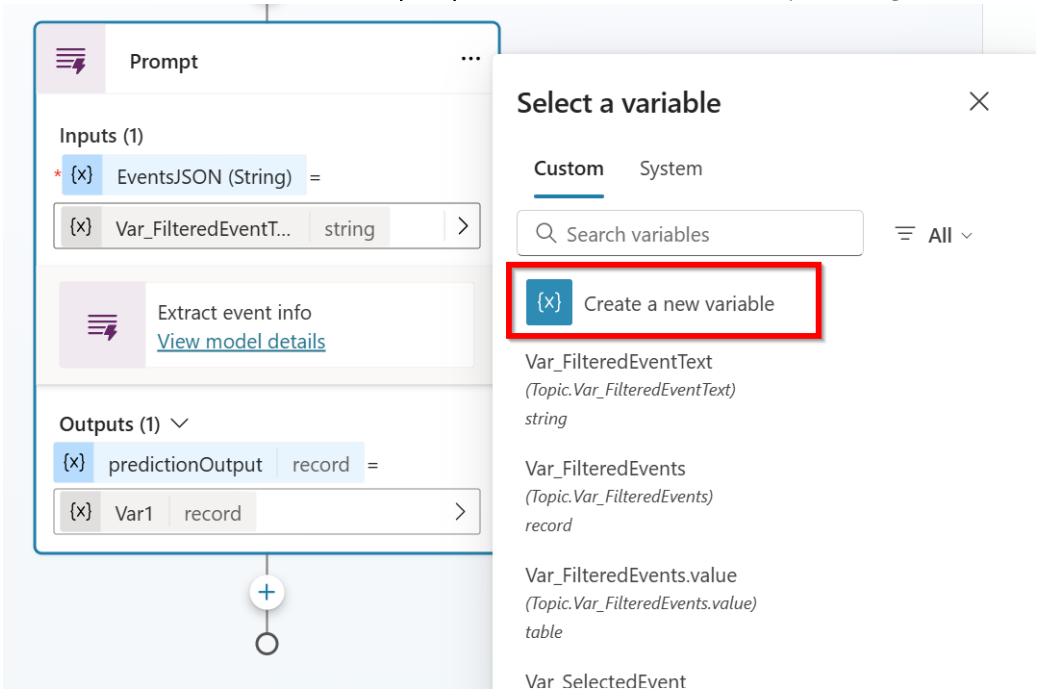




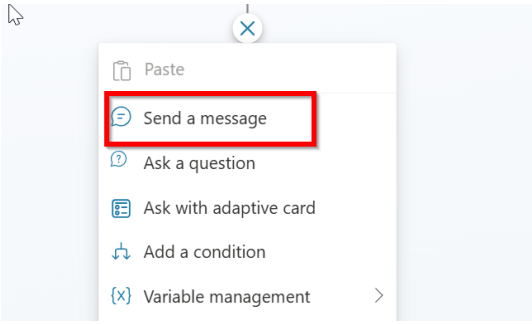
17. Fill out the **input parameter** EventsJSON to be variable Var_FilteredEventText

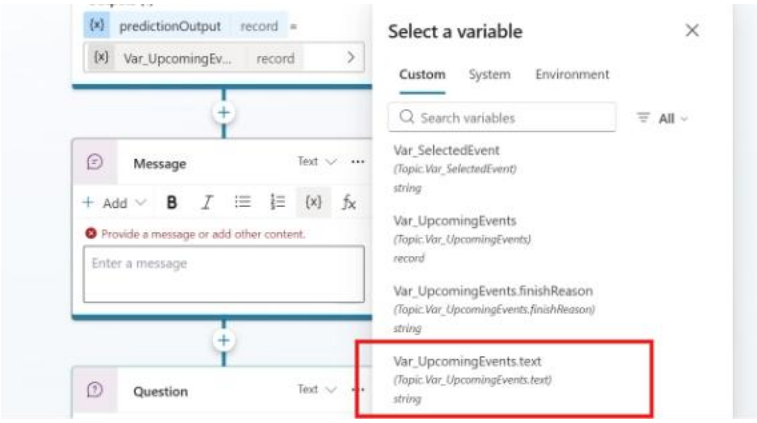


18. Create a new variable for **Output parameter**, named Var_UpcomingEvents.

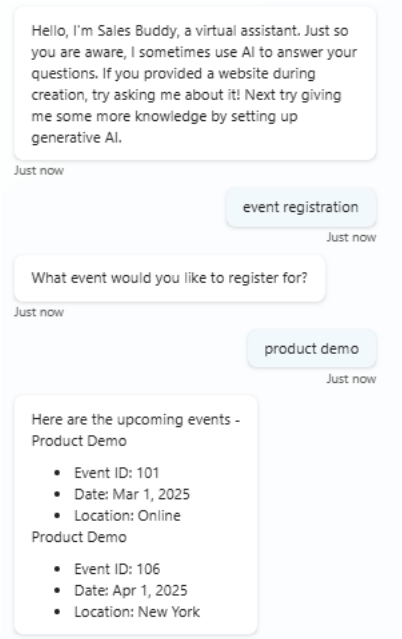


19. Add a **new node to send a message**. Pick Var_upcomingevents.text from the variable list.

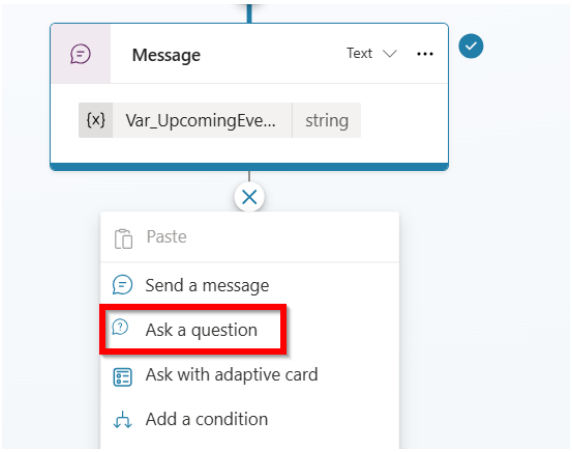


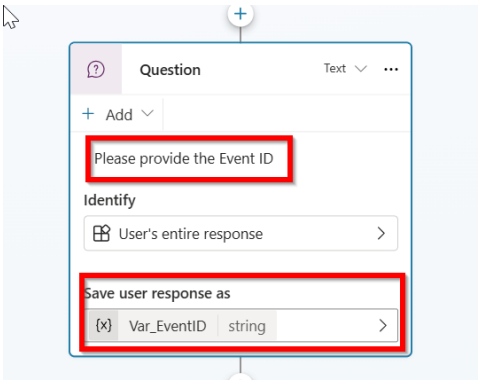


20. Save the agent and test it. You should see the conversation flow as following:

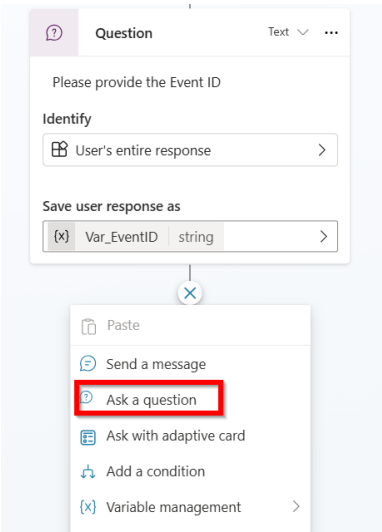


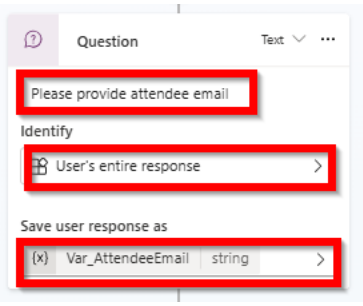
21. Add a question node to capture EventID. In the text field, enter "Please provide the Event ID". In the Identify field, pick User's entire response. In the Save user response as field, rename the variable to "Var_EventID".



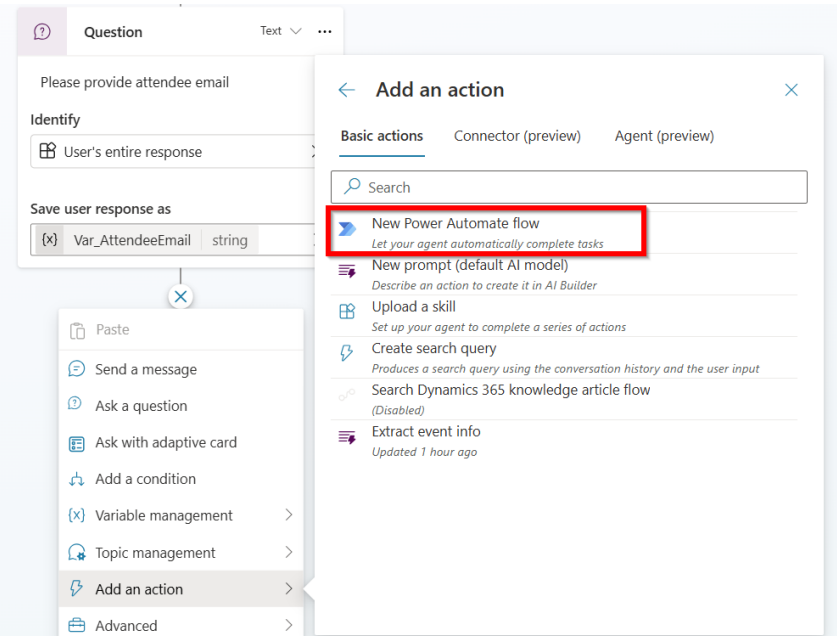


22. Add a new node to capture attendee email. In the text field, enter "Please provide attendee email". In the Identify field, pick User's entire response. In the Save user response as field, rename the variable to Var_AttendeeEmail



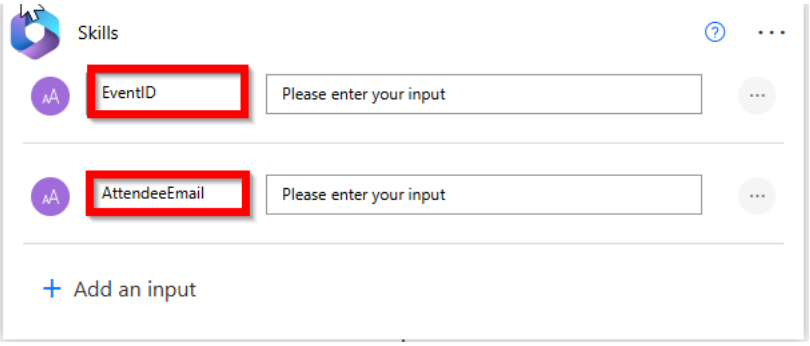


23. Now, we need to create a **power automate flow** to log attendee email and event id registration to the Excel sheet.
    a. **Add an action node** and pick New **Power Automate flow** in the Basic actions tab.

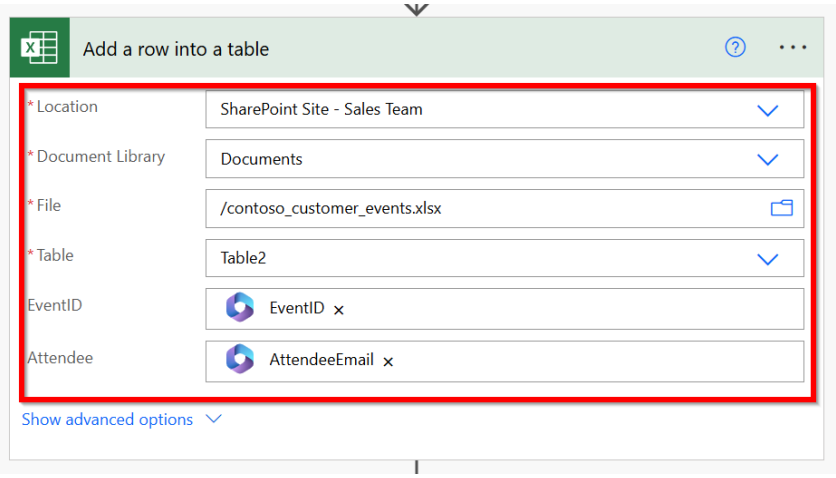


b. It will take you to the **Flow Designer page**. Switch to the old designer view by clicking on the toggle button on the top right. New designer (Turn it off)

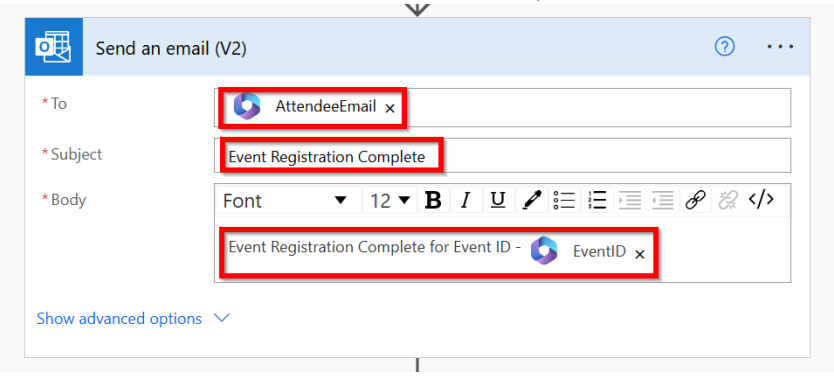
c. Clicking on the first Skill tile. **Add two input variables**: EventID and AttendeeEmail



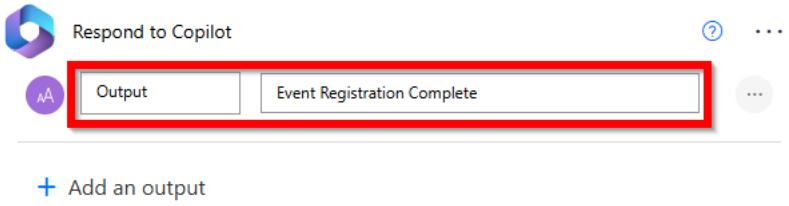
d. **Add an excel action** to add a row into a table. Update the fields as follows.



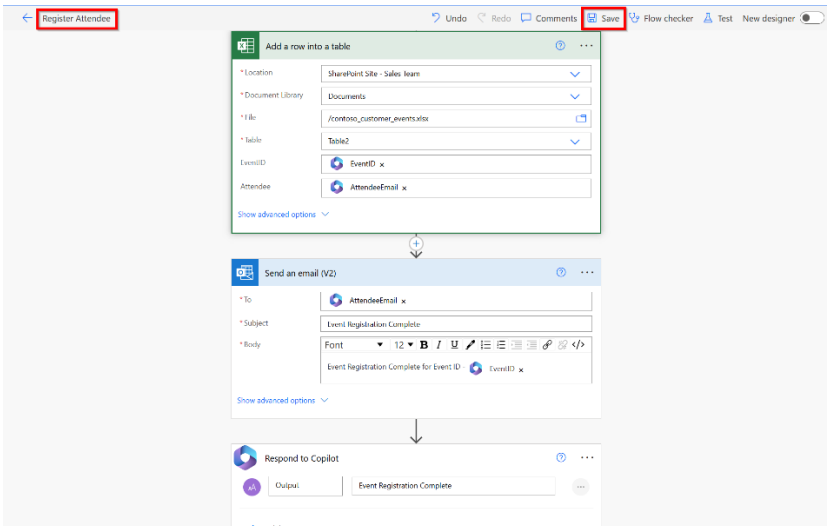
e. Add an outlook action to send emails. Update the fields as follows.



f. In the respond action, add an output parameter. Set the value to "Event Registration Complete."

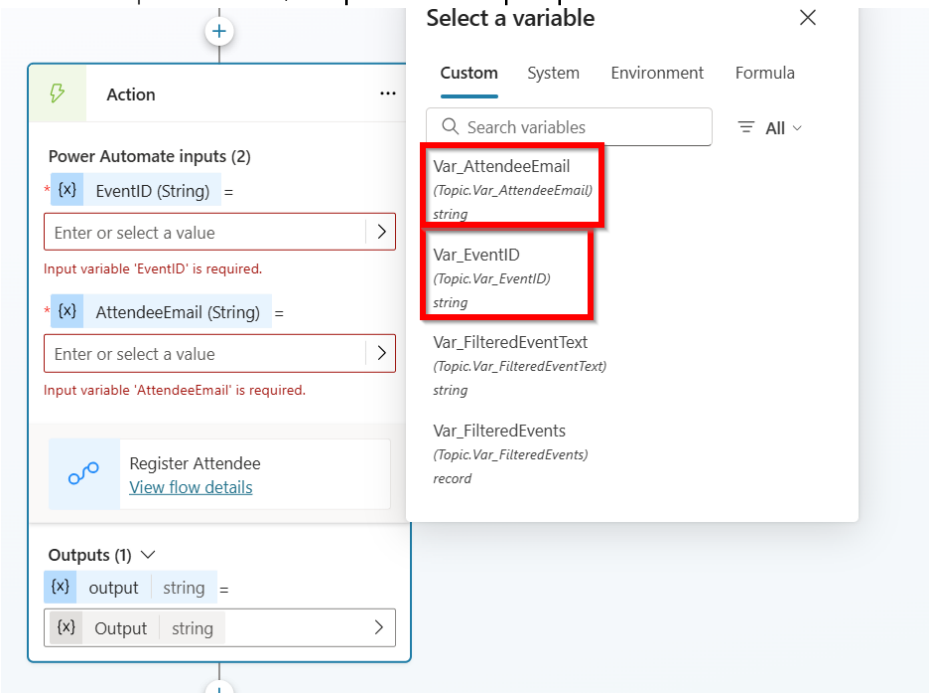


g. Finally, rename the flow to Register Attendee.

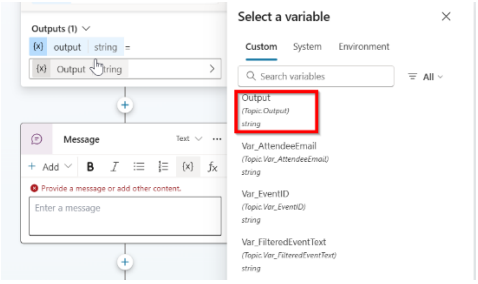


h. Save the flow in the old designer. Save and Publish if you are using the new designer.

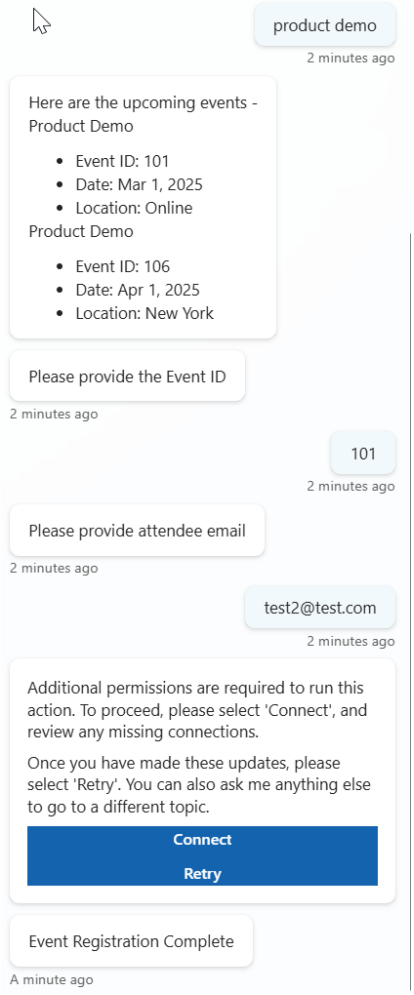
24. Back in copilot studio, map the two input parameters to variables as follows:



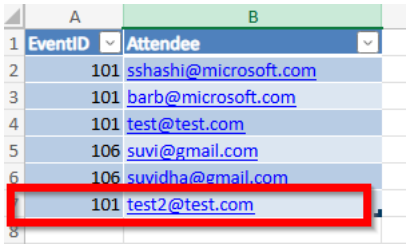
25. Add a node to send confirmation message. In the text field, pick the Output variable from previous step.



26. Test the agent. See an example below. You might need to create a connection for the 1st time. Click on Connect and follow the steps.



27. Open Excel and go to the 2nd tab. Verify that the attendee registration is logged properly.



	A	B
1	EventID	Attendee
2	101	sshashi@microsoft.com
3	101	barb@microsoft.com
4	101	test@test.com
5	106	suvi@gmail.com
6	106	suvidha@gmail.com
7	101	test2@test.com
8		

Congratulations! You have successfully completed Lab 2. You can now proceed to Lab 3.