 Microsoft Copilot Studio

Lab 01

Hands-on lab step-by-step

February 2025

Microsoft Copilot Studio Workshop

Contents

[Microsoft Copilot Studio 1](#_Toc182512048)

[Goals for this lab 1](#_Toc182512049)

[Prerequisites 1](#_Toc182512050)

[Fundamental Knowledge: Understanding Power Automate 2](#_Toc182512051)

[Exercise 1: Build a basic Power Automate cloud flow 3](#_Toc182512052)

[Summary 26](#_Toc182512053)

[Terms of Use 27](#_Toc182512054)

# Microsoft Copilot Studio

This lab is subject to the Terms of Use found at the end of this document.

## Goals for this lab

|  |  |
| --- | --- |
| After this lab you will be able to:   * Understand the basics of Power Automate * Use Copilot Studio to request data from another data source using Power Automate in a basic use case (using the ServiceNow connector) and return the data in a conversational dialog with a customer or user | The time to complete this lab is [60] minutes. |

## Prerequisites

Labs have been designed to be completed with only a Microsoft Copilot Studio trial. You can start most labs without having to complete the previous module but note that some exercises may reference previous labs. To fully experience the features and functionality of the product, it is recommended that you make sure to have completed all pre-requisites below before starting this lab.

For this lab you need:

* A computer with internet access.
* Be able to log into the provided Microsoft tenant (some companies enforce users to only connect to their company tenant) or your own enterprise tenant with a Copilot Studio User License (or trial)
* **Generative AI should be set to “classic” (in Settings, Generative AI)**
* **Access to Power Automate (make.powerautomate.com)**
* **Access to an active ServiceNow instance (URL, login and password) – don’t forget to wake the instance if you use a trial**

## Fundamental Knowledge: Understanding Power Automate

Power Automate is a cloud-based service that makes it practical and simple for line of business users to build workflows that automate time-consuming business tasks and processes across applications and services.

Power Automate is part of a powerful and adaptable business application platform that includes Power Apps, Microsoft Dataverse, Dynamics 365, and Office 365. This platform allows our customers, our partners, and our ISV partners to create purpose-built solutions for their own companies, their industry, for functional roles or even for specific geographies. Line-of-business users, who understand their business needs best, can now easily analyze, compose, and streamline data and processes. Professional developers can easily extend the automation, analytics and apps line-of-business to leverage Azure services like Functions, App Service, and Logic Apps. API connectors, gateways, and Microsoft Dataverse make it possible to get more value out of services or data already in use, either in the cloud or on-premises.

Here are a few examples of what you can do with Power Automate.

* Automate business processes.
* Send automatic reminders for past due tasks.
* Move business data between systems on a schedule.
* Connect to more than 1000 data sources or any publicly available API.
* You can even automate tasks on your local computer like computing data in Excel.

Microsoft Copilot Studio connects easily with Power Automate, being able to pass the variables from user’s responses and retrieve data from several different data sources, perform complex operations on that data and return to Microsoft Copilot Studio to share that data with the user. Being able to operate on and retrieve data from almost any data source accessible via an API is one of the most valuable benefits of Copilot Studio.

Alternatively, Microsoft Copilot Studio can also call the same connectors, HTTP requests, or custom connectors as in Power Automate, directly from a topic or from a plugin action.

As part of this Microsoft Copilot Studio lab, it will not include an extensive introduction to Power Automate but does cover a basic scenario of how you can retrieve data from an external data source and use it in the conversational experience of Copilot Studio. To learn more specifically about Power Automate, you can review the [Microsoft Docs on Power Automate](https://learn.microsoft.com/en-us/power-automate/) and also review the in a day material for Microsoft Power Apps, which includes Power Automate.

### Create an agent

1. From the Microsoft Copilot Studio **Home** page, **describe** your agent to create it.

|  |  |  |
| --- | --- | --- |
|  | I want to build an agent for my customer support. It is an assistant for Contoso customers, helping to answer common questions and help with common tasks, like checking a ticket status |  |

A screenshot of a computer

Description automatically generated

1. You will get redirected to a conversational experience to further customize your agent. You can provide further details to the description (you can **decline** to do so), and you will also be prompted for a **tone of voice**.

|  |  |  |
| --- | --- | --- |
|  | Playful tone, joyful, customer focus, but professional |  |

1. You will get asked to **provide a name** for your agent.

|  |  |  |
| --- | --- | --- |
|  | Call this agent [insert name] |  |

1. You will also get asked for **publicly accessible websites** to get information from.

|  |  |  |
| --- | --- | --- |
|  | Information should come from https://learn.microsoft.com/en-us/microsoft-copilot-studio/ and from https://www.microsoft.com/en-us/microsoft-copilot/ |  |

1. Don’t forget to confirm the utilization of the website in the chat panel.

A screenshot of a chat

AI-generated content may be incorrect.

1. You will also get asked **topics or tasks** the agent **shouldn’t help with or talk about**.

|  |  |  |
| --- | --- | --- |
|  | We don't want to discuss other brands like Fabrikam. Never provide product comparisons with competitor technologies. |  |

1. When prompted, you can answer **I’m done**
2. Before you create your copilot, go the “**…**” and “**Edit advanced settings**” menu to update the agent **schema name by appending your First and Last name** (because contrary to the agent *display* name, the schema name is a technical property that can’t be changed afterwards and must be **unique across your environment**) and save it a **solution** (e.g. “Copilot Studio Workshop”) to be able to export it later and inherit from the solution publisher prefix).

|  |  |
| --- | --- |
| Warning outline | **Make sure you pick a unique name for your custom agent by appending your First and Last name to both display name and schema name.** |

1. You can then choose “**Create**”.

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Lights On outline | 1. You can choose to avoid the conversational creation experience by selecting “**Skip to configure**” 2. You can set the copilot’s primary language in the “**Settings**” menu. For the lab, be sure to remain in English (en-US). 3. It is best practice to always configure your agent in the **context of your own solution** and **publisher**, so that the agent is created with the desired publisher prefix, and so that you can easily **export it and deploy it to other environments**. |

## Exercise 2: Take a quick tour of the user interface

Microsoft Copilot Studio makes it easier for you to build basic to advanced Copilots. The following section reviews the main pages of the maker experience for Microsoft Copilot Studio.

### Main interface

A screenshot of a computer

Description automatically generated

* 1. **Home** – Displays Microsoft Copilot Studio home page. This is the page where you initially landed. You can start creating new copilots from here, it contains the list of recent copilots, a list of templates to avoid creating new copilots from scratch, as well as learning resources.

**Create** – This menu gets you to the conversational agent creation experience.  
**Agents** – List of all the agents your user has access to in the environment.  
**Library** – List of connectors available for the extension of Microsoft 1st-party copilots.

* 1. **Agents**– List of available agents that you can customize and quickly navigate to (you can pin/unpin this panel).

|  |  |
| --- | --- |
| Lights On outline | **Pro tip**: when you work on a single agent, you should **unpin** **the list of agents** to get more screen real estate for your authoring. |

* 1. **Menu** – Tabbed navigation between the most useful Copilot Studio capabilities.  
     **Overview** – Description of the copilot, its instructions, and quick view of its configuration (knowledge sources, topics, actions, publish status, etc.)  
     **Knowledge** – Where you manage the agent knowledge sources (website, files, etc.)  
     **Topics** - Where you manage custom and system topics. Topics are the core building blocks of a copilot. Topics can be seen as the agent competencies: they define how a conversation plays out. Topics are discrete conversation paths that, when used together, allow for users to have a conversation that feels natural and flows appropriately.  
     **Actions** – Where you manage action. Actions are pieces of logic with inputs and outputs. They leverage Power Platform components such as connectors, Power Platform cloud flows, AI Builder custom prompts, or Bot Framework skills. Actions are useful to leverage generative AI to both prompt the user for the necessary inputs but also to summarize the output of the action in the desired format.  
     **Analytics** – Where you can view metrics to monitor how well your agent is serving your users and identify ways to improve it.  
     **Channels** – Where you configure how your agent is being made available to your users (e.g. Teams, website, etc.)
  2. **Overview –** Where you can edit the agent description, its generative AI instructions, and where you can have a quick view of its configuration (knowledge sources, topics, actions, publish status, etc.)
  3. **Environment** – Where you can identify the Power Platform environment you’re working from. You would typically create and author a agent in a development environment and deploy it to test and production environments.
  4. **Publish** – Where you can make the latest version of your agent available to your users. Apart from the test pane, changes are not reflected to your end-users until you publish the agent.  
     **Settings** – Where you can manage your agent configuration (advanced settings, security, language, etc.)
  5. **Test your agent** – The test pane allows you to immediately test your agent and your customizations, even without needing to save.

### Settings interface

A screenshot of a computer

Description automatically generated

1. **Agent details** – Where you can update the agent display name, icon, and modify advanced settings (e.g. configure the Azure Application Insights integration)
2. **Generative AI** – Where you can choose to replace the more classic natural language understanding approach for topic triggering and entity extraction with one that’s based on a large language model to do multi-intent detection and more complex entity extraction. This is also where you can configure content moderation setting for knowledge sources (to reduce risks of hallucinations).
3. **Security** – This is where you configure end-user authentication settings (the type of authentication and whether it is enforced or not), and web channel security, that allows you to further secure the Direct Line channel that is used for any web or custom application deployment.
4. **Authoring Canvas** – Enable Optimized Canvas for topics with a high number of nodes, improving performance and usability.
5. **Entities** – Copilot Studio comes with a lot of pre-built entities to help identify key information in a user utterance (e.g. a city, date, number, etc.). This menu is also where you can define your own closed-list entities or regular expression entities.
6. **Skills** – Where you register external Bot Framework skills that your Copilot Studio agent can call, or where you can configure how existing Azure Service Bot can use your Copilot Studio agent as a skill.
7. **Voice** – Make sure your agent works for you with voice-first features like advanced speech recognition and dual-tone multi-frequency (DTMF) input.
8. **Languages** – Where you can configure additional languages your agent can be used in and localized into.
9. **Language understanding** – Where you can configure custom language models developed and trained on Azure AI Language, in Azure Conversational Language Understanding (CLU). When configured, this effectively replaces the out-of-the-box natural language understanding model (NLU) for intent detection and can also replace entity detection and extraction.
10. **Component collections** – Group components together in a collection to distribute across copilots.

## Exercise 3: Test your copilot

1. To show the **Test agent** pane, in the top-right corner of the screen, select **Test**.

A screenshot of a phone

Description automatically generated

1. The **Test agent** pane shows that a message has already been sent to you from the copilot. This message was sent from the **Conversation Start** topic, which begins automatically. At the **Ask a question or describe what you need** prompt, at the bottom of the **Test agent** pane, enter **Hello** and then select the **Send** button.

A screenshot of a chat

Description automatically generated

The Copilot will offer a greeting in the **Test Copilot** pane

|  |  |
| --- | --- |
| Lights On outline | **Pro tips:**   * Click on a message in the test pane to get redirected to the exact topic and node it was used in. * When in a topic, you can access the variables values (**Topic** variables, **Global** variables or even **Environment** variables) in the Variables menu (It’s OK if you don’t have any environment variables on your environment). Within this menu, the Test tab displays the current value at run time * You can **search** through your variable names by clicking the magnifying glass |

A screenshot of a computer

Description automatically generated

## Exercise 1: Build a basic Power Automate cloud flow

Connecting to data provides companies with some of the most benefits as it provides information and insight to users that is up to date and often the relevant for customer or user questions.

In this exercise, you will go through creating a new topic, adding a simple Power Automate action to retrieve information from an external service, and display that data back to the user.

### Task 1: Create a new topic

1. Open the Copilot Studio authoring canvas’ **Topics** page and click **Add a topic** drop down at the top of the screen, select the **From blank** option. Name your topic:

Check Ticket Status

1. Add some **trigger phrases** that a user may ask such as the below ones (if you don’t see trigger phrases, make sure your “Generative AI” settings are set to “classic”)

|  |  |  |
| --- | --- | --- |
|  | What is the status of my ticket INC0008001  Can you get me information on my ticket status  Could you check the status of my ticket  Status update on ticket INC0009005  What’s happening with my ticket INC1234567 |  |

A screenshot of a computer

Description automatically generated

1. Create a new **Question** node and enter text:

Of course, I can get you the status of your ticket. What is your ticket number?

1. In **Identify**, choose **Create an Entity** of type **Regular expression (Regex)**, call it

Ticket Number

and with this pattern:

INC[0-9]{7}

1. Rename the **Var1** variable to

TicketNumber

A screenshot of a chat

Description automatically generated

1. **Save** your topic.

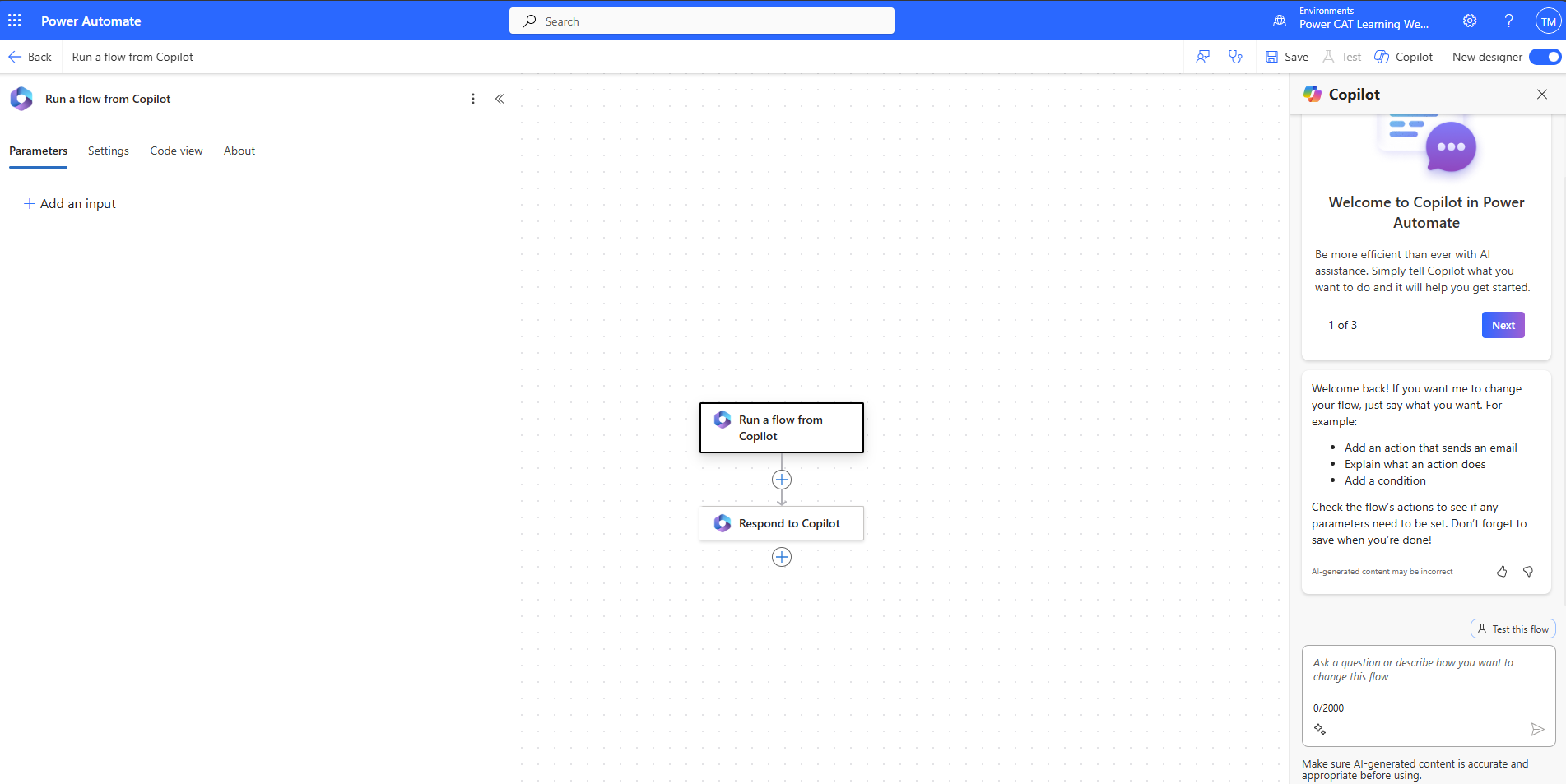
### Task 2: Create your Power Automate cloud flow

1. Click on the **(+)** new node button under the **Question** node, and select **Call an action** and then click **Create a flow** underneath **Basic actions** as shown in the screenshot below.

A screenshot of a computer

Description automatically generated

1. This will open **Power Automate** in a new browser window and includes the scaffolding pre and post actions for a new Power Automate cloud flow to interact with Copilot Studio, as shown in the screenshot below:



|  |  |
| --- | --- |
| Lights On outline | **Pro tip:** Make sure the **New designer** is enabled in Power Automate, in the top right corner |

1. Click on the **Run a flow from Copilot** trigger, in the left-hand pane click **Add an input**  
     
   A screenshot of a computer

   Description automatically generated
2. Add an **input** of type **Text**, and call it TicketNumber  
     
   
3. Click **Add an action** in between the trigger and the action you just added

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Lights On outline | **Note**: in the next steps, **if the ServiceNow** **environment** **isn’t active** and if the connection cannot be properly established – this is typically the case if the **Record Type** doesn’t return a list with **Incident** as an option in step #8 – then **skip to step #11**. |

1. Search for ServiceNow List Recordsin the search bar and select **List Records**

A screenshot of a computer

Description automatically generated

Create a connection for **ServiceNow.** Name the connection to something unique, for example  
ServiceNow - {Your User Name}**.**

Set the **Authentication type** to **Basic Authentication**Set the **Instance** to <https://dev258107.service-now.com/> Replace your instance ID   
Set the **Username** to CopilotStudioServiceAccount  
Set the **Password** to:

|  |  |  |
| --- | --- | --- |
|  | irlOu,i{<{1aczPK+Gnm{(K-\_<EJIr{FXZe&^bLs:C?C!7O]mXfJP89hvKeqhcVK-\_n@L8nCLK&yi.xJ]24I(yOe%3}EDz+;Eo |  |

Then select **Create** **New**

**![A screenshot of a computer screen

Description automatically generated**

1. Now for **Record Type**, select Incident (you can search to make it easier to find Incident in the list).   
   Select **Show all** in **Advanced parameters**.  
   Set the **Display System References** to Yes to show actual values.

Leave the **Exclude Reference Links** to Yes.   
In **Query**, type numberCONTAINS and select the TicketNumber input from the dynamic content (⚡icon).  
**Ensure there are no spaces** between **numberCONTAINS** and the **variable** you reference.  
Alternatively, you can also paste the below in the **Query** field:

|  |  |  |
| --- | --- | --- |
|  | numberCONTAINS@{triggerBody()?['text']} |  |

Set Limit to 1.

A screenshot of a computer

Description automatically generated

1. Click on the **Respond to Copilot** action in the cloud flow and add an **output** of type **Text** and call it SNTicketInfo**.**
2. For the output **value**, use the formula button **𝒇𝓍** in order to get a string version of the first returned record of the result array from the List Records body.  
   **Paste** the below formula, and select **Add**.

A screenshot of a computer

Description automatically generated

|  |  |  |
| --- | --- | --- |
|  | string(first(outputs('List\_Records')?['body/result'])) |  |

A screenshot of a computer

Description automatically generated

1. **Optional step (if ServiceNow isn’t working):** for the output **value**, paste a the below hardcoded payload.  
   This represents an example of what ServiceNow would typically return.

|  |  |
| --- | --- |
| Lights On outline | If you struggle copying the below text, go the Misc folder in Lab files and open ServiceNow Sample JSON Payload.txt |

|  |  |  |
| --- | --- | --- |
|  | {  "parent": "",  "made\_sla": "true",  "caused\_by": "",  "watch\_list": "",  "upon\_reject": "Cancel all future Tasks",  "sys\_updated\_on": "2018-12-12 23:18:55",  "child\_incidents": "0",  "hold\_reason": "",  "origin\_table": "",  "task\_effective\_number": "INC0009005",  "approval\_history": "",  "number": "INC0009005",  "resolved\_by": "",  "sys\_updated\_by": "admin",  "opened\_by": "System Administrator",  "user\_input": "",  "sys\_created\_on": "2018-08-31 21:35:45",  "sys\_domain": "global",  "state": "New",  "route\_reason": "",  "sys\_created\_by": "admin",  "knowledge": "false",  "order": "",  "calendar\_stc": "",  "closed\_at": "",  "cmdb\_ci": "",  "delivery\_plan": "",  "contract": "",  "impact": "1 - High",  "active": "true",  "work\_notes\_list": "",  "business\_service": "",  "business\_impact": "",  "priority": "1 - Critical",  "sys\_domain\_path": "/",  "rfc": "",  "time\_worked": "",  "expected\_start": "",  "opened\_at": "2018-08-31 21:35:21",  "business\_duration": "",  "group\_list": "",  "work\_end": "",  "caller\_id": "David Miller",  "reopened\_time": "",  "resolved\_at": "",  "approval\_set": "",  "subcategory": "Email",  "work\_notes": "2018-12-12 23:18:42 - System Administrator (Work notes)\nupdated the priority to high based on the criticality of the Incident.\n\n",  "universal\_request": "",  "short\_description": "Email server is down.",  "correlation\_display": "",  "delivery\_task": "",  "work\_start": "",  "assignment\_group": "",  "additional\_assignee\_list": "",  "business\_stc": "",  "cause": "",  "description": "Unable to send or receive emails.",  "origin\_id": "",  "calendar\_duration": "",  "close\_notes": "",  "notify": "Do Not Notify",  "service\_offering": "",  "sys\_class\_name": "Incident",  "closed\_by": "",  "follow\_up": "",  "parent\_incident": "",  "sys\_id": "ed92e8d173d023002728660c4cf6a7bc",  "reopened\_by": "",  "incident\_state": "New",  "urgency": "1 - High",  "problem\_id": "",  "company": "",  "reassignment\_count": "0",  "activity\_due": "2018-12-13 01:18:55",  "assigned\_to": "",  "severity": "3 - Low",  "comments": "",  "approval": "Not Yet Requested",  "sla\_due": "UNKNOWN",  "comments\_and\_work\_notes": "2018-12-12 23:18:42 - System Administrator (Work notes)\nupdated the priority to high based on the criticality of the Incident.\n\n",  "due\_date": "",  "sys\_mod\_count": "3",  "reopen\_count": "0",  "sys\_tags": "",  "escalation": "Normal",  "upon\_approval": "Proceed to Next Task",  "correlation\_id": "",  "location": "",  "category": "Software"  } |  |

1. The cloud flow is almost complete. It needs to be **renamed** before we move on to best practices, so it is easily found in Copilot Studio and by administrators.   
   Click the template **title** as shown in the screenshot below and rename it to Get Ticket Status {YourUserName}
2. Click **Publish** and wait a moment until you see the green banner as shown in the screenshot below.

|  |  |
| --- | --- |
| Lights On outline | **Pro tip:** if you don’t see the **Publish** button, select **Save**. |

A screenshot of a computer

Description automatically generated

We have completed our work in Power Automate. Now let’s switch back to Copilot Studio.

### Task 3: Call your Power Automate cloud flow from Copilot Studio

1. Open your existing topic in Copilot Studio and go back to the bottom of your flow, as shown below. Click **Call an action**, and you should see your new Power Automate cloud flow in the list under the **Basic actions** tab. From the list, select your **Get Ticket Status** flow (exact name may differ based on participants).

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Lights On outline | **Pro tip:** If you don’t see the cloud flow you’ve created, **Save** the topic, and **refresh** the page to try again. |

1. When you select your **Get Ticket Status** flow, you will see a new **Action** node is automatically added.  
   If the flow requires an input, it requests the value to be selected. As the flow you created in the previous steps requires the **TicketNumber** input, we need to add this input into the Power Automate action by selecting the variable the value is stored in from the user, which is **TicketNumber** from the previous steps earlier in the lab.
2. Select **Enter or select a value** and choose your **TicketNumber** variable you created in the previous steps of this lab. This is now connected to the Power Automate flow, and outputs the result from Power Automate into the **SNTicketInfo** variable.

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Lights On outline | **Pro tips:**   * If a latency is expected from your integration, go the action’s properties and add a Latency Message, for example: I’m getting these details for you. Hold on… * Consider using HTTP requests and connectors directly in Microsoft Copilot Studio to avoid the added latency of invoking and running a cloud flow in Power Automate. |

1. To use the **author’s permissions** (i.e., the connection of the user who created the cloud flow to connect to ServiceNow), and **not the end-user permissions**, follow the below steps. Otherwise, you’ll get the below error:

A screenshot of a message

Description automatically generated

1. In your action step, click on **View flow details**.
2. Edit the **Run only users** option.
3. Switch to “**Use this connection (ServiceNow - {Your User Name})**”
4. Select **Save**.

A screenshot of a computer error

Description automatically generated

1. As **ServiceNow** will return the full details of the incident in a technical, **JSON**, format, you need to parse it so that Copilot Studio fully understand its content based on its schema.
2. To do this, **go back to your “Check Ticket Status” topic**, go to the end, **click on (+)**, select **Variable management**, and then **Parse value**.

A screenshot of a computer

Description automatically generated

To parse the JSON you can use the Rest API Explorer in ServiceNow to get the structure of the body, or get the schema from a sample payload. For the lab, we’re providing sample ServiceNow data later in this chapter.

|  |  |
| --- | --- |
| Lights On outline | If you struggle copying the below text, go the Misc folder in Lab files, and open ServiceNow Sample JSON Payload.txt |

1. Now we will need to select the value to parse by selecting the **SNTicketInfo** variable from the Power Automate flow action.
2. Next we will select the **Data type** to **From sample data**  
     
   A screenshot of a computer

   Description automatically generated
3. Now paste in the JSON sample provided below (next page), after selecting **Get schema from sample JSON** and select **Confirm**.

A screenshot of a computer

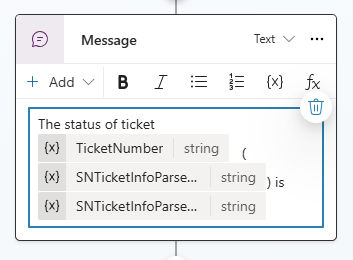
Description automatically generated

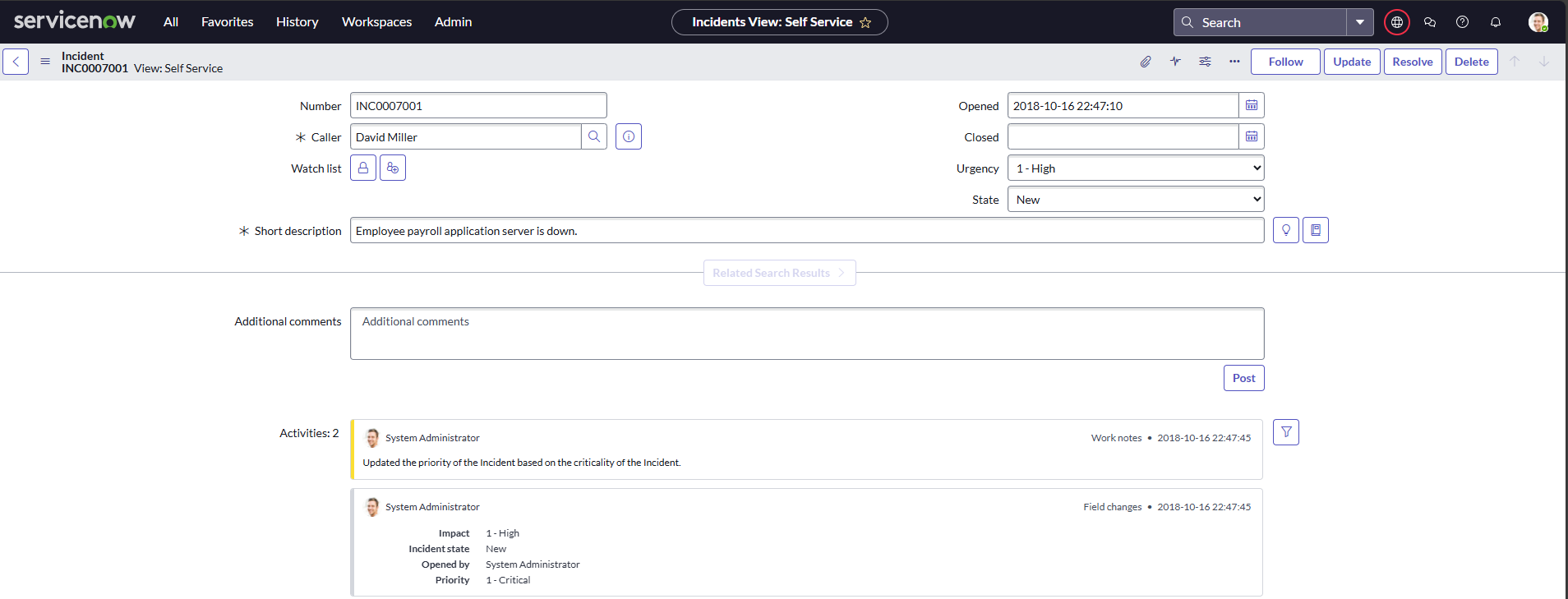
|  |  |  |
| --- | --- | --- |
|  | {  "parent": "",  "made\_sla": "true",  "caused\_by": "",  "watch\_list": "",  "upon\_reject": "Cancel all future Tasks",  "sys\_updated\_on": "2018-12-12 23:18:55",  "child\_incidents": "0",  "hold\_reason": "",  "origin\_table": "",  "task\_effective\_number": "INC0009005",  "approval\_history": "",  "number": "INC0009005",  "resolved\_by": "",  "sys\_updated\_by": "admin",  "opened\_by": "System Administrator",  "user\_input": "",  "sys\_created\_on": "2018-08-31 21:35:45",  "sys\_domain": "global",  "state": "New",  "route\_reason": "",  "sys\_created\_by": "admin",  "knowledge": "false",  "order": "",  "calendar\_stc": "",  "closed\_at": "",  "cmdb\_ci": "",  "delivery\_plan": "",  "contract": "",  "impact": "1 - High",  "active": "true",  "work\_notes\_list": "",  "business\_service": "",  "business\_impact": "",  "priority": "1 - Critical",  "sys\_domain\_path": "/",  "rfc": "",  "time\_worked": "",  "expected\_start": "",  "opened\_at": "2018-08-31 21:35:21",  "business\_duration": "",  "group\_list": "",  "work\_end": "",  "caller\_id": "David Miller",  "reopened\_time": "",  "resolved\_at": "",  "approval\_set": "",  "subcategory": "Email",  "work\_notes": "2018-12-12 23:18:42 - System Administrator (Work notes)\nupdated the priority to high based on the criticality of the Incident.\n\n",  "universal\_request": "",  "short\_description": "Email server is down.",  "correlation\_display": "",  "delivery\_task": "",  "work\_start": "",  "assignment\_group": "",  "additional\_assignee\_list": "",  "business\_stc": "",  "cause": "",  "description": "Unable to send or receive emails.",  "origin\_id": "",  "calendar\_duration": "",  "close\_notes": "",  "notify": "Do Not Notify",  "service\_offering": "",  "sys\_class\_name": "Incident",  "closed\_by": "",  "follow\_up": "",  "parent\_incident": "",  "sys\_id": "ed92e8d173d023002728660c4cf6a7bc",  "reopened\_by": "",  "incident\_state": "New",  "urgency": "1 - High",  "problem\_id": "",  "company": "",  "reassignment\_count": "0",  "activity\_due": "2018-12-13 01:18:55",  "assigned\_to": "",  "severity": "3 - Low",  "comments": "",  "approval": "Not Yet Requested",  "sla\_due": "UNKNOWN",  "comments\_and\_work\_notes": "2018-12-12 23:18:42 - System Administrator (Work notes)\nupdated the priority to high based on the criticality of the Incident.\n\n",  "due\_date": "",  "sys\_mod\_count": "3",  "reopen\_count": "0",  "sys\_tags": "",  "escalation": "Normal",  "upon\_approval": "Proceed to Next Task",  "correlation\_id": "",  "location": "",  "category": "Software"  } |  |

1. Then set a variable to store the parsed record for later user.  
   Create a variable called SNTicketInfoParsed. Its type will automatically be set based on its schema (record).
2. Now, add a response to give the user of your agent a formatted response to tell them the status of the ticket. Click the **(+)** to add a new node and select **Send a message**.   
   **Bold** the key information either with the command bar or by surrounding the text with \*\*.

|  |  |  |
| --- | --- | --- |
|  | The status of ticket {Topic.TicketNumber} ({Topic.SNTicketInfoParsed.short\_description}) is {Topic.SNTicketInfoParsed.state} |  |

|  |  |
| --- | --- |
| Lights On outline | **Pro tip:** Copilot Studio and some channels support [Markdown](https://www.markdownguide.org/) for simple formatting. |



You can look above at the sample JSON to see what data would be returned in what value, however below is a screenshot of the ticket we will lookup later to help you with structuring your message:  
  


1. To end the conversation, select **(+) to add a new node**, **Topic management** and **Go to another topic** and select **End of Conversation**.
2. **Save** your topic and **test** your copilot.

|  |  |  |
| --- | --- | --- |
|  | What is the status of my ticket INC0007001? |  |

1. Make sure you check the status of ticket number **INC0007001**:

A screenshot of a chat

Description automatically generated

You have successfully created a Power Automate cloud flow and a new topic in Microsoft Copilot Studio that used the flow to provide real time data from an external service to the user!

### Task 4: Display the ServiceNow ticket information in an adaptive card

1. Go to your **Check Ticket Status** topic
2. **Delete** the text message variable for your message

A screenshot of a computer

Description automatically generated

1. Select **Add**, **Adaptive Card**

A screenshot of a phone

Description automatically generated

1. Toggle from **Edit JSON** to **Formula**, so that you can make the adaptive card dynamic and author it in   
   Power Fx language.

A screenshot of a computer program

Description automatically generated

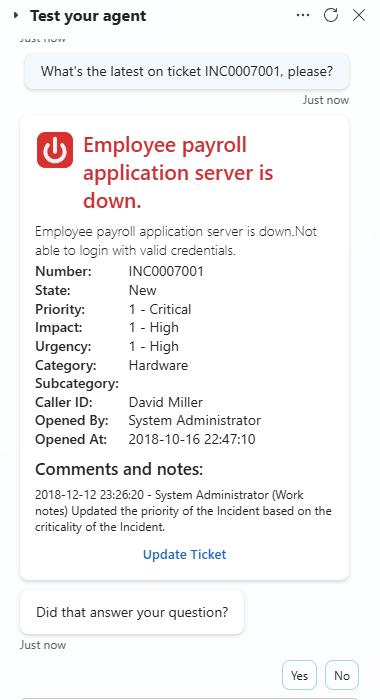
1. **Paste** the below Power Fx formula that already contains the references to the ServiceNow ticket information

|  |  |
| --- | --- |
| Lights On outline | If you struggle copying the below text, go the Misc folder in Lab files, and open ServiceNow Adaptive Card Power Fx.txt |

|  |  |  |
| --- | --- | --- |
|  | {  type: "AdaptiveCard",  version: "1.5",  body: [  {  type: "ColumnSet",  columns: [  {  type: "Column",  width: "auto",  items: [  {  type: "Image",  url: "https://upload.wikimedia.org/wikipedia/commons/6/67/Shutdown\_button\_red\_wikimedia.svg",  size: "Small"  }  ]  },  {  type: "Column",  width: "stretch",  items: [  {  type: "TextBlock",  text: Topic.SNTicketInfoParsed.short\_description,  weight: "Bolder",  size: "Large",  wrap: true,  color: "Attention",  horizontalAlignment: "Left"  }  ],  verticalContentAlignment: "Center",  horizontalAlignment: "Center"  }  ]  },  {  type: "TextBlock",  text: Topic.SNTicketInfoParsed.description,  weight: "Lighter",  wrap: true  },  {  type: "FactSet",  facts: [  {  title: "Number:",  value: Topic.SNTicketInfoParsed.number  },  {  title: "State:",  value: Topic.SNTicketInfoParsed.state  },  {  title: "Priority:",  value: Topic.SNTicketInfoParsed.priority  },  {  title: "Impact:",  value: Topic.SNTicketInfoParsed.impact  },  {  title: "Urgency:",  value: Topic.SNTicketInfoParsed.urgency  },  {  title: "Category:",  value: Topic.SNTicketInfoParsed.category  },  {  title: "Subcategory:",  value: Topic.SNTicketInfoParsed.subcategory  },  {  title: "Caller ID:",  value: Topic.SNTicketInfoParsed.caller\_id  },  {  title: "Opened By:",  value: Topic.SNTicketInfoParsed.opened\_by  },  {  title: "Opened At:",  value: Topic.SNTicketInfoParsed.opened\_at  }  ],  spacing: "Small"  },  {  type: "TextBlock",  text: "Comments and notes:",  weight: "Bolder",  size: "Medium",  wrap: true  },  {  type: "TextBlock",  text: Topic.SNTicketInfoParsed.comments\_and\_work\_notes,  wrap: true,  size: "Small"  }  ],  actions: [  {  type: "Action.OpenUrl",  title: "Update Ticket",  url: "https://dev204932.service-now.com/nav\_to.do?uri=incident.do?sys\_id=" & Topic.SNTicketInfoParsed.sys\_id & "%26sysparm\_view=ess"  }  ],  '$schema': "http://adaptivecards.io/schemas/adaptive-card.json"  } |  |

1. **Save**.
2. **Test** your agent.

|  |  |  |
| --- | --- | --- |
|  | What's the latest on ticket INC0007001, please? |  |



|  |  |
| --- | --- |
| Lights On outline | If the link to update your ticket does not work you might need to replace the ServiceNow URL in the adaptive card by the latest one used in your workshop. |

## Summary

Thank you for completing the lab 1. You have successfully:

* Created a new Power Automate cloud flow
* Called the Power Automate cloud flow into your topic
* Set input and output variables
* Displayed dynamic data back to the user in Copilot Studio

## Terms of Use

By using this document, in whole or in part, you agree to the following terms:

### **Notice**

Information and views expressed in this document, including (without limitation) URL and other Internet Web site references, may change without notice. Examples depicted herein, if any, are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product.

### **Use Limitations**

Copying or reproduction, in whole or in part, of this document to any other server or location for further reproduction or redistribution is expressly prohibited. Microsoft provides you with this document for purposes of obtaining your suggestions, comments, input, ideas, or know-how, in any form, ("Feedback") and to provide you with a learning experience. You may use this document only to evaluate its content and provide feedback to Microsoft. You may not use this document for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this document or any portion thereof. You may copy and use this document for your internal, reference purposes only.

### **Feedback**

If you give Microsoft any Feedback about this document or the subject matter herein (including, without limitation, any technology, features, functionality, and/or concepts), you give to Microsoft, without charge, the right to use, share, and freely commercialize Feedback in any way and for any purpose. You also give third parties, without charge, the right to use, or interface with, any Microsoft products or services that include the Feedback. You represent and warrant that you own or otherwise control all rights to such Feedback and that no such Feedback is subject to any third-party rights.

### **DISCLAIMERS**

CERTAIN SOFTWARE, TECHNOLOGY, PRODUCTS, FEATURES, AND FUNCTIONALITY (COLLECTIVELY "CONCEPTS"),

INCLUDING POTENTIAL NEW CONCEPTS, REFERENCED IN THIS DOCUMENT ARE IN A SIMULATED ENVIRONMENT

WITHOUT COMPLEX SET-UP OR INSTALLATION AND ARE INTENDED FOR FEEDBACK AND TRAINING PURPOSES

ONLY. THE CONCEPTS REPRESENTED IN THIS DOCUMENT MAY NOT REPRESENT FULL FEATURE CONCEPTS AND MAY

NOT WORK THE WAY A FINAL VERSION MAY WORK. MICROSOFT ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH

CONCEPTS. YOUR EXPERIENCE WITH USING SUCH CONCEPTS IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

THIS DOCUMENT, AND THE CONCEPTS AND TRAINING PROVIDED HEREIN, IS PROVIDED “AS IS”, WITHOUT WARRANTY

OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING (WITHOUT LIMITATION) THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. MICROSOFT DOES NOT

MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, THE OUTPUT THAT DERIVES FROM USE OF THIS DOCUMENT OR THE CONCEPTS, OR THE SUITABILITY OF THE CONCEPTS OR INFORMATION CONTAINED IN THIS DOCUMENT FOR ANY PURPOSE.

MICROSOFT COPILOT STUDIO (1) IS NOT INTENDED OR MADE AVAILABLE AS A MEDICAL DEVICE FOR THE

DIAGNOSIS OF DISEASE OR OTHER CONDITIONS, OR IN THE CURE, MITIGATION, TREATMENT OR PREVENTION OF

DISEASE, OR OTHERWISE TO BE USED AS A COMPONENT OF ANY CLINICAL OFFERING OR PRODUCT, AND NO LICENSE

OR RIGHT IS GRANTED TO USE MICROSOFT COPILOT STUDIO FOR SUCH PURPOSES, (2) IS NOT DESIGNED OR

INTENDED TO BE A SUBSTITUTE FOR PROFESSIONAL MEDICAL ADVICE, DIAGNOSIS, TREATMENT, OR JUDGMENT AND SHOULD NOT BE USED AS A SUBSTITUTE FOR, OR TO REPLACE, PROFESSIONAL MEDICAL ADVICE, DIAGNOSIS,

TREATMENT, OR JUDGMENT, AND (3) SHOULD NOT BE USED FOR EMERGENCIES AND DOES NOT SUPPORT EMERGENCY

CALLS. ANY CHATBOT YOU CREATE USING MICROSOFT COPILOT STUDIO IS YOUR OWN PRODUCT OR SERVICE,

SEPARATE AND APART FROM MICROSOFT COPILOT STUDIO. YOU ARE SOLELY RESPONSIBLE FOR THE DESIGN,

DEVELOPMENT, AND IMPLEMENTATION OF YOUR CHATBOT (INCLUDING INCORPORATION OF IT INTO ANY PRODUCT

OR SERVICE INTENDED FOR MEDICAL OR CLINICAL USE) AND FOR EXPLICITLY PROVIDING END USERS WITH

APPROPRIATE WARNINGS AND DISCLAIMERS PERTAINING TO USE OF YOUR CHATBOT. YOU ARE SOLELY RESPONSIBLE

FOR ANY PERSONAL INJURY OR DEATH THAT MAY OCCUR AS A RESULT OF YOUR CHATBOT OR YOUR USE OF

MICROSOFT COPILOT STUDIO IN CONNECTION WITH YOUR CHATBOT, INCLUDING (WITHOUT LIMITATION) ANY SUCH INJURIES TO END USERS.