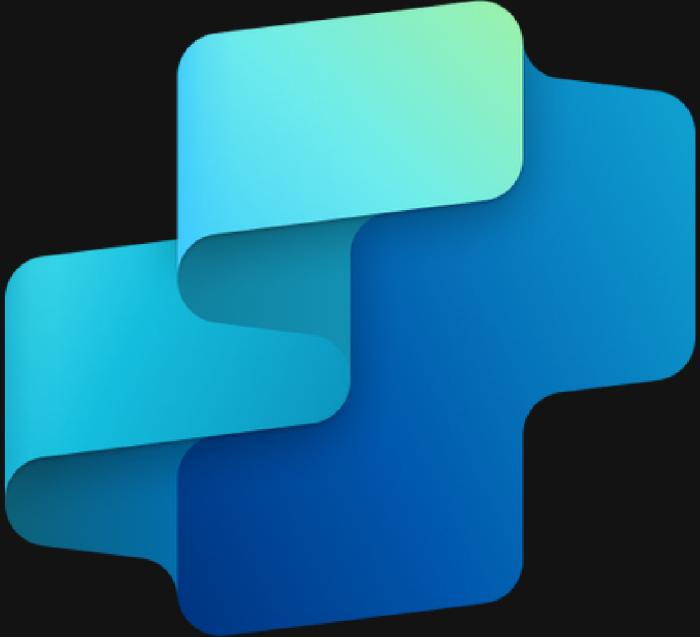


- 13 -

Customizing the
Ask a Question node for
Smarter Copilot Interactions



MONTHLY MASTERY

FEATURE-A-DAY

with Copilot Studio



Handcrafted Insights by
Katerina Chernevskaya



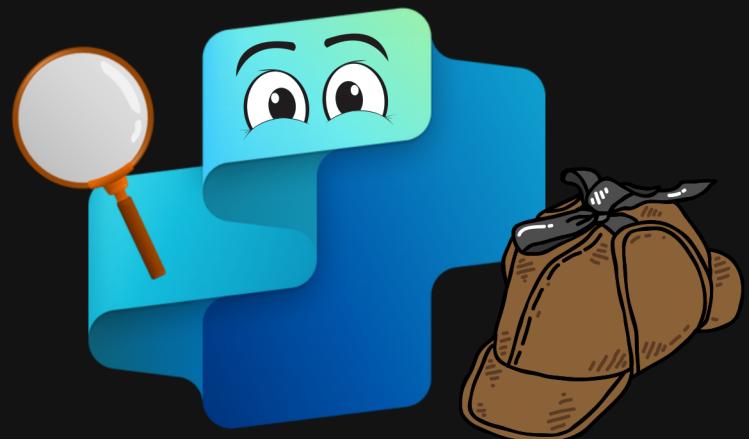
Ask a question

Introduction

The Sherlock Holmes of Copilot Studio, dedicated to unraveling the mysteries of user inputs.

This little digital detective helps you **gather** crucial clues (ahem, user responses) in the most engaging ways.

Think of it as your AI sidekick, eager to **ask** the right questions, **collect** the answers, and even **handle** the curveballs users might throw its way.



Handcrafted Insights by
Katerina Chernevskaya



Ask a question

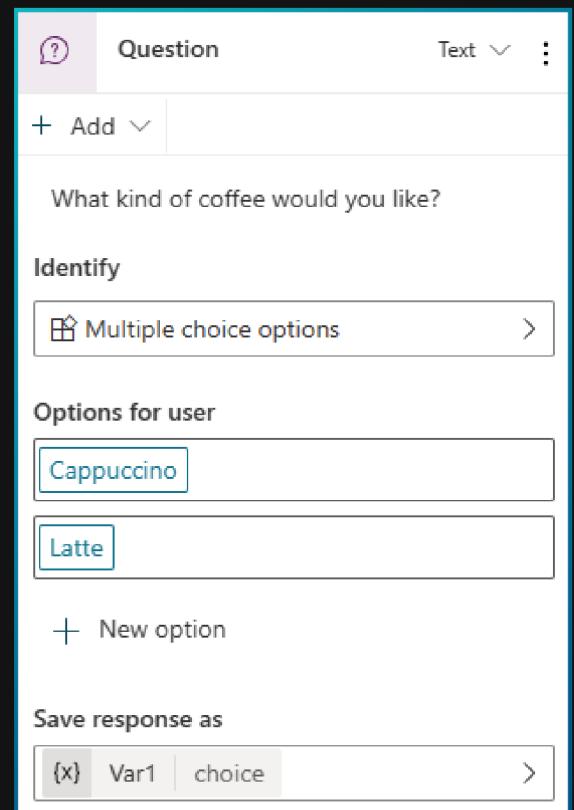
Collect user input

To gather user responses, let's revisit the [Ask a question](#) node, which we previously explored on [Day 4](#) (find a refresher here - <https://t.ly/lAS73>, page 8).

Adding this node starts with [Multiple choice options](#), turning your questions into clickable buttons for users.

The responses they choose are saved in a variable, like a digital keepsake.

You're free to rename this variable for clarity, but its type is automatically set based on the options you provide, ensuring that each user's choice is accurately captured.



Handcrafted Insights by
Katerina Chernevskaya



Ask a question

Identify options

This feature lets you choose response types for your questions.

You can set **multiple-choice** for straightforward, predefined answers, or opt for **open-ended** responses for detailed feedback.

This flexibility ensures you can tailor interactions to gather the specific information you need.

Let's explore them!

Multiple choice options

Options from a list variable

User's entire response

Entity



Handcrafted Insights by
Katerina Chernevskaya



Ask a question Options from a list variable

This feature allows you to create **dynamic**, interactive experiences by presenting users with a **set of options** derived from a predefined list variable.

The screenshot shows the Miro platform interface. On the left, a card titled "Set variable value" is open. It contains fields for "Set variable" (CoffeeMenu table), "To value" (fx Table({ id: 1, name: "Cappu...") with a plus sign to add more items), "Question" (What kind of coffee would you like?), "Identify" (Options from a list variable), "List variable" (CoffeeMenu table), and "Save response as" (UserChoice record). On the right, a panel titled "Variables test" shows the variable "UserChoice" defined as a record with a single item. The item is a table row with columns: 1, DisplayName: "Latte", id: 2, and name: "Latte small".



Handcrafted Insights by
Katerina Chernevskaya



Ask a question Options from a list variable

Prerequisite:

A **Table type variable** that serves as a container for your potential options or answers. It can be pre-populated with a static list of values or dynamically updated from external data sources or user inputs.

The screenshot shows the 'Set variable value' step in the Power Automate designer. A blue arrow points from the 'To value' field to the 'Variable value' pane on the right. The 'To value' field contains the formula `Table({ id: 1, name: "Cappu..." })`. The 'Variable value' pane displays the contents of the 'UserChoice' record, which is a table with one item:

```
1 ↴ {
2   "DisplayName": "Latte",
3   "id": 2,
4   "name": "Latte small"
5 }
```



Handcrafted Insights by
Katerina Chernevskaya



Ask a question Options from a list variable

Key Requirement:

Ensure your table schema includes a `DisplayName` field.
Copilot utilizes this field to display options as clickable
action buttons for the user.

```
1 Table(  
2 {  
3   id: 1,  
4   name: "Cappuccino small",  
5   DisplayName: "Cappuccino"  
6 },  
7 {  
8   id: 2,  
9   name: "Latte small",  
10  DisplayName: "Latte"  
11 }  
12 )
```

The screenshot shows the Microsoft Copilot interface for configuring a list variable. On the left, a code editor displays a JSON-like table schema with two items. The second item's `DisplayName` field is highlighted with a red box. A blue arrow points from this highlighted field to the "Set variable value" dialog on the right. This dialog shows the variable `CoffeeMenu` set to a table with the same two items. The "To value" field contains a function call `Table({ id: 1, name: "Cappuccino small", DisplayName: "Cappuccino" }, { id: 2, name: "Latte small", DisplayName: "Latte" })`. Below this, the "Question" field contains the text "What kind of coffee would you like?". The "Identify" field has "Options from a list variable" selected. The "List variable" field is set to `CoffeeMenu`, and the "Save response as" field is set to `UserChoice`. On the far right, a preview window titled "Variables test" shows the variable `UserChoice` as a record containing the two menu items with their `DisplayName` fields explicitly defined.



Handcrafted Insights by
Katerina Chernevskaya



Ask a question Options from a list variable

Capturing User Choices:

When a user selects an option, their choice is stored as a **record** within the variable. This record keeping allows for a seamless and interactive decision-making process within your conversation flow.

The screenshot shows the Microsoft Bot Framework Composer interface. On the left, the 'Set variable value' card is open, allowing the configuration of a variable. The 'Set variable' section is set to 'CoffeeMenu' table, and the 'To value' section contains the expression 'Table({ id: 1, name: "Cappu..." })'. Below this, the 'Question' section asks 'What kind of coffee would you like?'. The 'Identify' section is set to 'Options from a list variable'. The 'List variable' section is also set to 'CoffeeMenu' table. The 'Save response as' section is set to 'UserChoice' record. On the right, the 'Variables test' pane displays the current value of the 'UserChoice' variable, which is a record containing a single item. This item has an ID of 2 and a display name of 'Latte'. A blue arrow points from the 'UserChoice' variable in the 'Variables test' pane to the corresponding entry in the list variable table.

```
1 ↵ {  
2   "DisplayName": "Latte",  
3   "id": 2,  
4   "name": "Latte small"  
5 }
```



Handcrafted Insights by
Katerina Chernevskaya



Ask a question

Options from a list variable

The Options from a list variable feature in Microsoft Copilot Studio offers a **dynamic** and **flexible alternative** to the more rigid **Multiple choice options**.

The screenshot shows the Microsoft Copilot Studio interface. On the left, the 'Set variable value' dialog is open, displaying a configuration for a 'UserChoice' record. It includes fields for 'Set variable' (CoffeeMenu table), 'To value' (Table({ id: 1, name: "Cappu..."}), 'Question' (What kind of coffee would you like?), 'Identify' (Options from a list variable), 'List variable' (CoffeeMenu table), and 'Save response as' (UserChoice record). On the right, the 'Variables test' panel shows the resulting JSON object:

```
1 ↴ {
2   "DisplayName": "Latte",
3   "id": 2,
4   "name": "Latte small"
5 }
```



Handcrafted Insights by
Katerina Chernevskaya

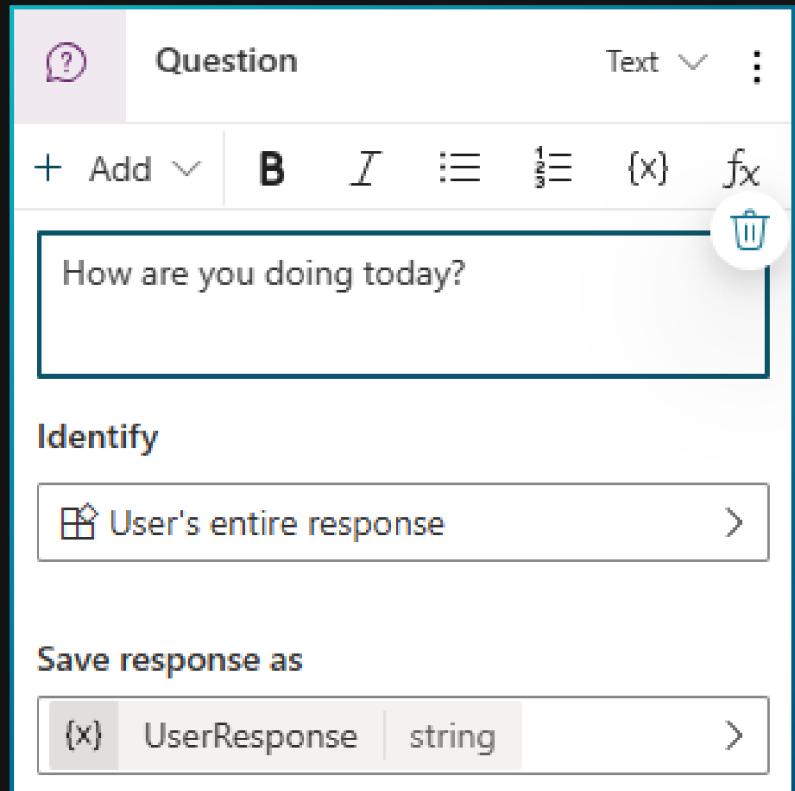


Ask a question

User's entire response

It is an invaluable tool for capturing the **full** breadth of a user's input.

When you select this option, the system is configured to record the **complete text** of the user's response, rather than just a specific part or a selection from predefined choices.



Handcrafted Insights by
Katerina Chernevskaya

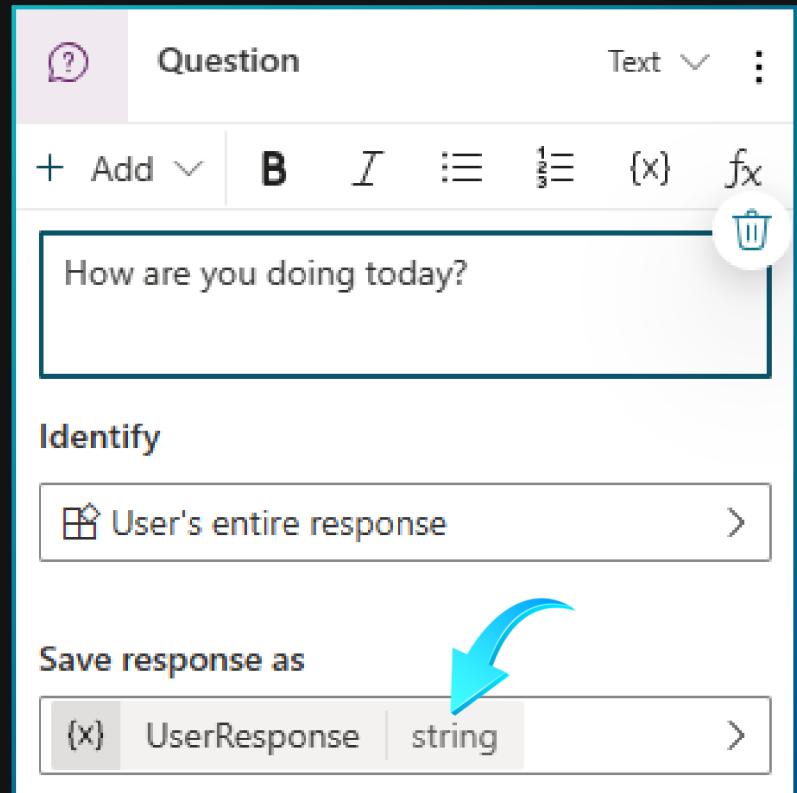


Ask a question

User's entire response

Capturing User Input:

The entire response from the user is stored in a variable as a **string**. This means you can access and use the full text later in the conversation for analysis, decision-making, or follow-up questions.



Handcrafted Insights by
Katerina Chernevskaya



Ask a question

User's entire response

Applications:

This feature is particularly useful when you need **in-depth** feedback, opinions, or **explanations** that cannot be confined to simple choices or single entities.

It's ideal for situations where you want to understand the user's perspective in **their own words** or need **comprehensive** information.

Question

Text

+ Add

B I $\frac{1}{3}$ {x} fx

How are you doing today?

Identify

User's entire response

Save response as

{x} UserResponse | string



Handcrafted Insights by
Katerina Chernevskaya

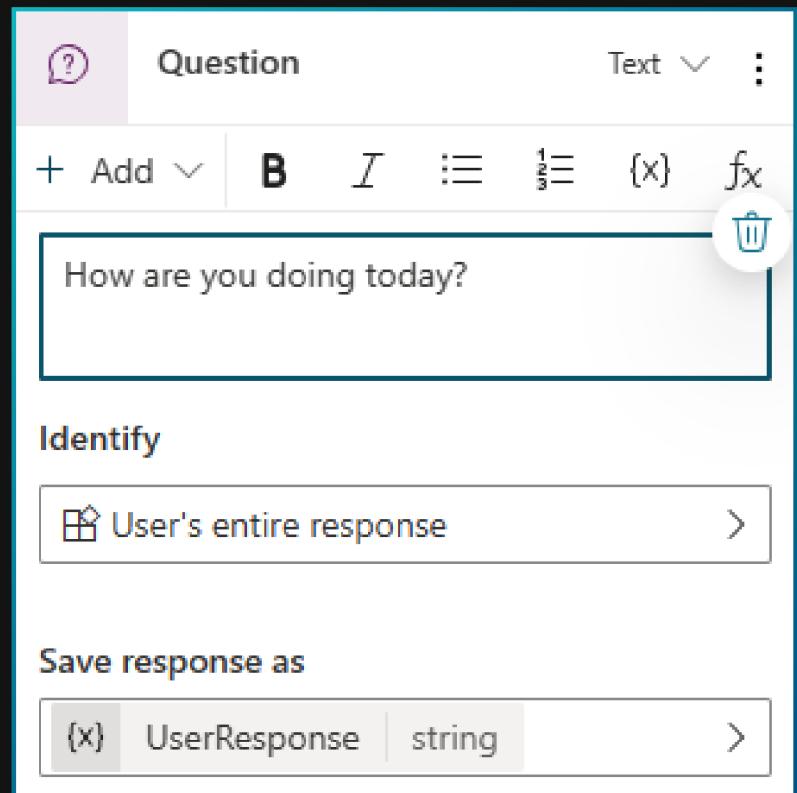


Ask a question

User's entire response

By leveraging this feature, you can gain deeper insights into user thoughts and feedback, making your Copilot interactions richer and more informative.

This approach is key to creating a more nuanced and responsive AI conversation experience.



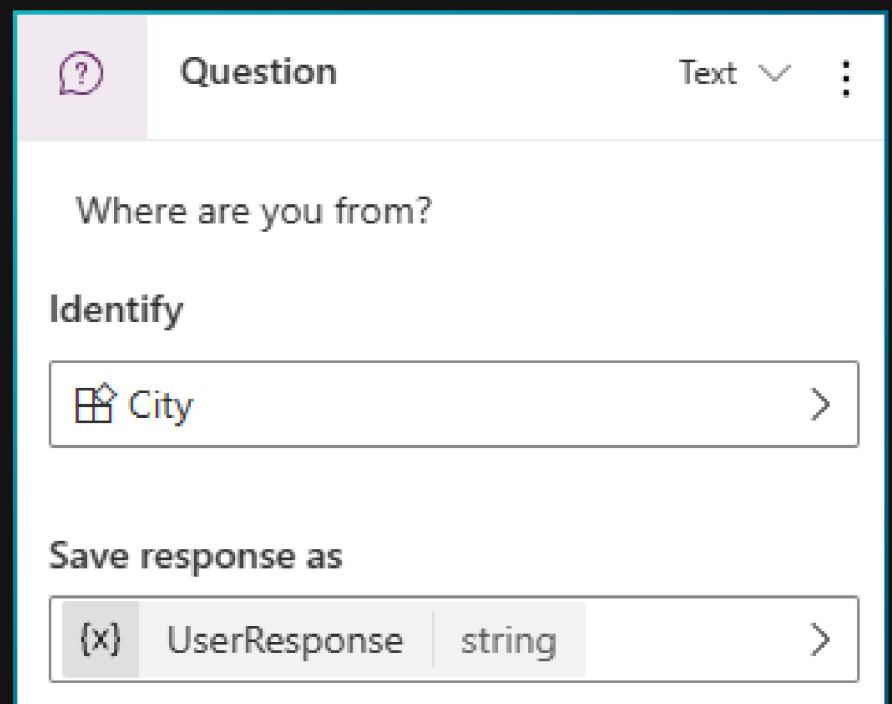
Handcrafted Insights by
Katerina Chernevskaya



Ask a question Entity

It is a powerful feature that allows you to target **specific** types of information in user responses.

When you use this option, you're instructing the AI to look for and extract **particular entities** - like names, dates, locations, or other custom categories - from what the user says.



Handcrafted Insights by
Katerina Chernevskaya



Ask a question Entity

Specific Data Extraction:

This option is all about **precision**. It enables the AI to pinpoint and extract key pieces of information from a user's response based on the entity type you define.

City		
Description	City names, extracted as a string	
Method	Prebuilt	
Usage		
User input	Entity	Saved value
The business headquarters is in Seattle	Seattle	Seattle
I love NYC	NYC	New York City
I'm going to Vancouver B.C.	Vancouver B.C.	Vancouver B.C.

Question Text :

Where are you from?

Identify

City >

Save response as

{x} UserResponse string >



Handcrafted Insights by
Katerina Chernevskaya



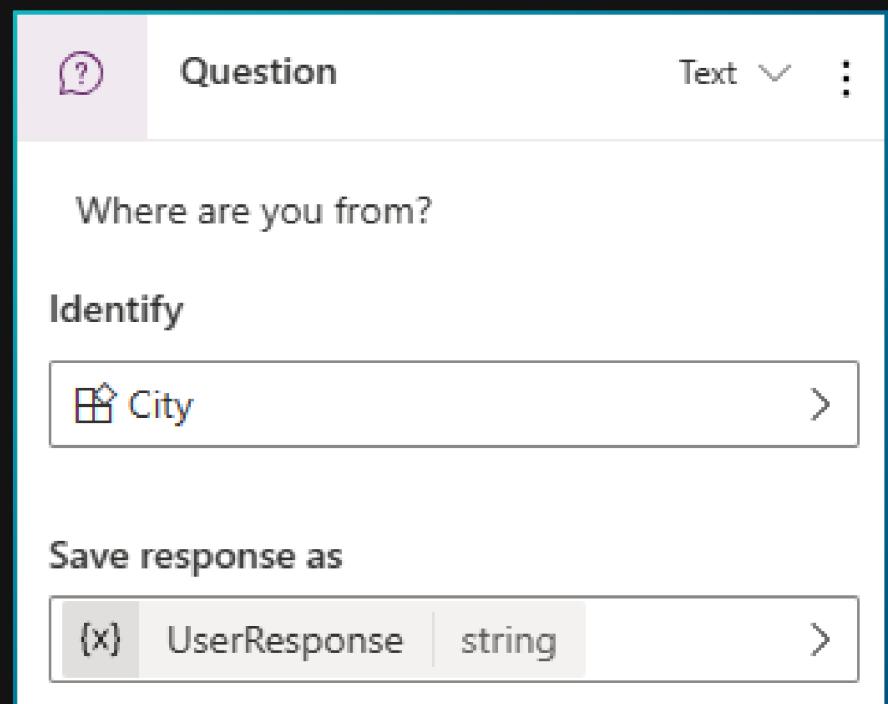
Ask a question Entity

Prebuilt Entities:

Copilot Studio offers various **prebuilt** entities for **common** data like names and dates.

Be mindful that if these entities don't recognize a rare value, the system might re-prompt the user.

Adjusting the Copilot's response settings can help ensure smoother interactions.



Handcrafted Insights by
Katerina Chernevskaya



Ask a question Entity

Custom Entities:

In addition to these prebuilt options, you also have the flexibility to create **custom** entities that are specifically tailored to your **unique** requirements.

This opens up a world of possibilities for personalized interactions.

We're excited to delve deeper into how to craft custom entities in one of our upcoming series, expanding the horizon of what you can achieve with Copilot Studio.



The screenshot shows the Copilot Studio interface with a configuration card for a custom entity. The card has a purple header with a question mark icon and the word "Question". To the right are "Text" and a more options menu. The main body of the card contains the question "Where are you from?", followed by an "Identify" section with a "City" field, and a "Save response as" section with a dropdown set to "{x} UserResponse string".



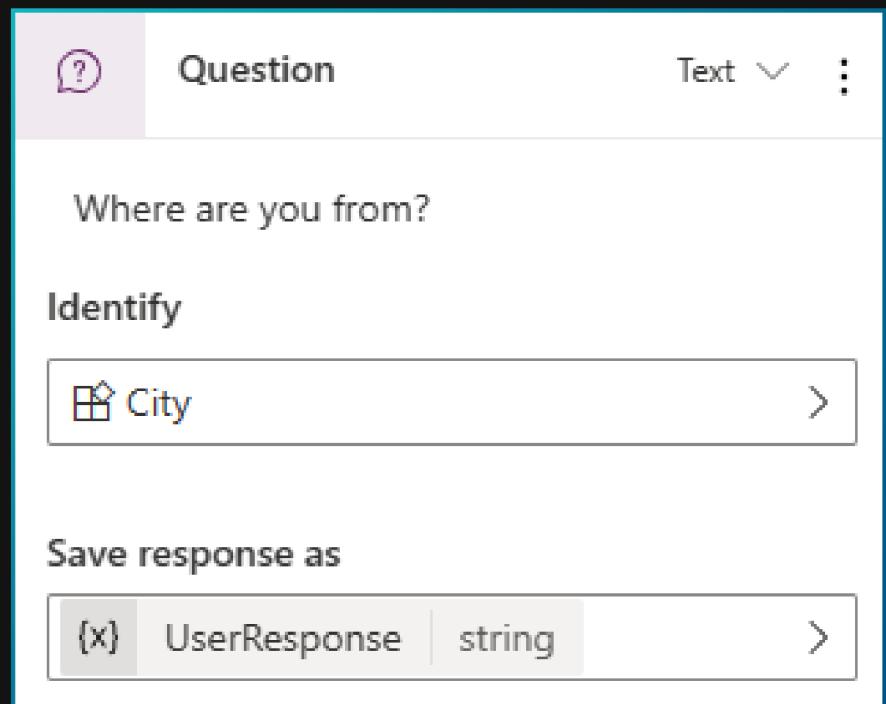
Handcrafted Insights by
Katerina Chernevskaya



Ask a question Entity

The Entity option is a cornerstone for creating intelligent, context-sensitive interactions in Copilot Studio, enabling the AI to understand and act on specific information within user responses.

This feature is essential for anyone looking to build sophisticated and responsive AI-driven conversational experiences.



Handcrafted Insights by
Katerina Chernevskaya

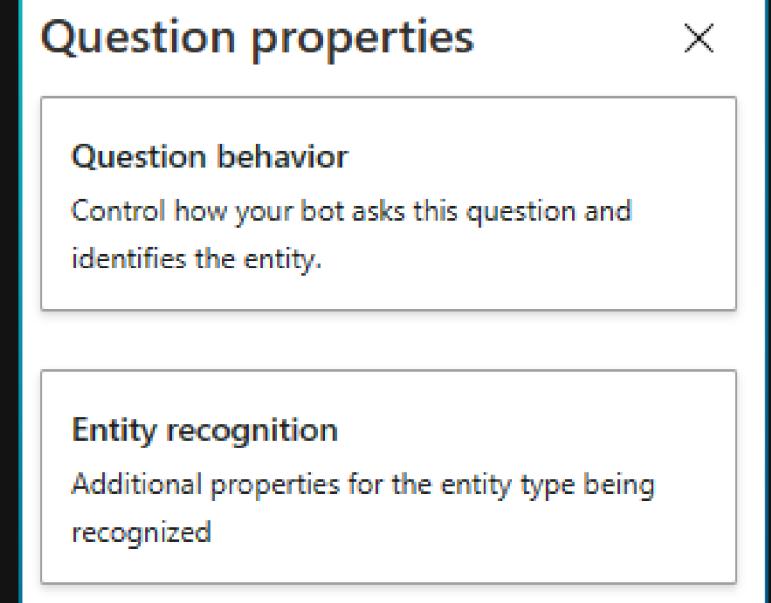


Ask a question

Additional properties

To summarize, the [Ask a Question](#) node offers a variety of ways to gather user input, from multiple-choice queries to open-ended questions.

But that's just the beginning. This node also comes packed with [additional properties](#) that let you customize its behavior to suit your needs, like settings for [skipping questions](#), [reprompting users](#), and more. In our upcoming sessions, we'll dive deeper into these features, unlocking the full potential of your copilot's conversational abilities.

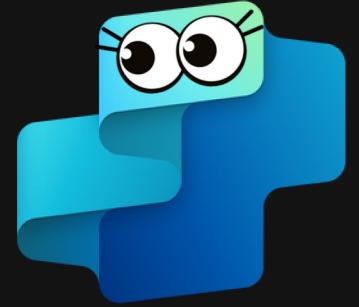


Stay tuned for a more detailed exploration!



Handcrafted Insights by
Katerina Chernevskaya





Today's Task: Exploring the 'Ask a Question' Node

1. Create an 'Ask a Question' Node

Begin by adding an 'Ask a Question' node to your conversation flow. This is your starting point for interactive queries.

2. Switch to 'User's Entire Response'

Change the response type to capture the 'User's Entire Response' and test your topic.

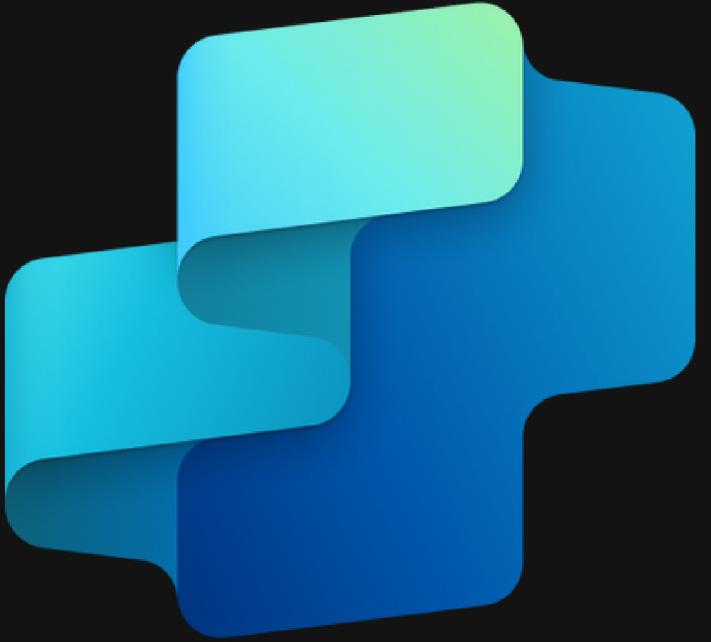
3. Experiment with 'Options from a List Variable'

Modify the node to use the 'Options from a List Variable' and test again.



Handcrafted Insights by
Katerina Chernevskaya





MONTHLY MASTERY

FEATURE-A-DAY

with Copilot Studio



Handcrafted Insights by
Katerina Chernevskaya

Follow for more!