Microsoft Power Virtual Agents in a Day

Lab 04: Improve your chatbot with variables, entities, and topic redirects

Hands-on Lab Step-by-Step

January 2022

Contents

Power Virtual Agents

This lab is subject to the Terms of Use on page 36 of this document.

Goals for this lab

After this lesson you will be able to:

- Understand the basic elements of handling dynamic content
- Build topics using variables and entities
- Switch topics using topic redirect

The time to complete this lab is [30] minutes.

Introduction to entities, variables, slots, and slot filling

You may be familiar with a party game where story has blanks and people are asked for words to fill in the blanks. For example:

"I am creating a/n ___*adjective*___ chatbot for my ___*type of transportation #1*___ and ___*type of transportation #2*___ company."

In the example above:

- "Type of transportation" is an **entity**. It's a real-world concept and contains cars, trucks, and so forth.
- Leaving a place for "type of transportation" in the sentence is what we call a **slot**. The example sentence has 3 slots; two of them are for types of transportation and one is for an adjective.
- "type of transportation #1" and "type of transportation #2" are **variables**. Later on they will be filled in with names of types of transportation.

During the game, players will provide an adjective and two types of transportation. The leader of the game will fill the slots in with those words.

"I am creating an awesome chatbot for my hot air balloon and skateboard company."

When the leader asks the players to give them words that fill certain criteria, such as "type of transportation", we call that **slot filling**. In your chatbot slot filling defines the value of the variable *for that conversation*. (Just like your party game will have different results each time you play it.)

You can imagine that if game players had started randomly calling out words, the leader would have been able to figure out that "dirt bike" was a type of transportation, "fantastic" was an adjective (and "purple" was a color and so forth) and put them in the right slots. Power Virtual Agents can do this too, using AI; we call this **proactive slot filling**.

This lab will show you how to use these features to enhance your chatbot conversations.

Exercise 1: Build a conversation using variables

Understanding and using variables in Power Virtual Agents

Variables allow you to handle dynamic content, such as product names, store locations, or even personal data input by the user, such as a user's name or location.

Variables let you save responses from your customers to help guide the conversation (such as to determine whether to provide different instructions for returns based on purchase price of the item) and also can be used directly in the conversational response from the bot ("I can help you return the {variable_ProductName}.").

By default, a variable's value can only be used in the topic where the variable is created. However, if you want the bot to use the same value across other topics, you can choose to make it a bot variable (you might know this concept from other systems as a global variable). This means that when the conversation moves to a different topic, the bot can remember and use variable values filled in previous topics in the conversation.

Every variable has a name and an entity type. We'll go deeper into entities in Exercise 2.

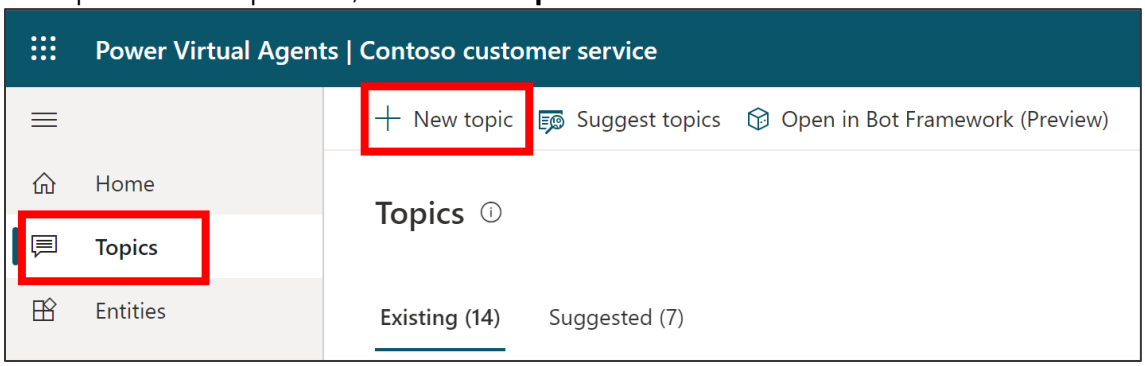
In Exercise 1, you will create a new topic to handle returns using a product price variable to route to the customer to the right resolution.

Before we start

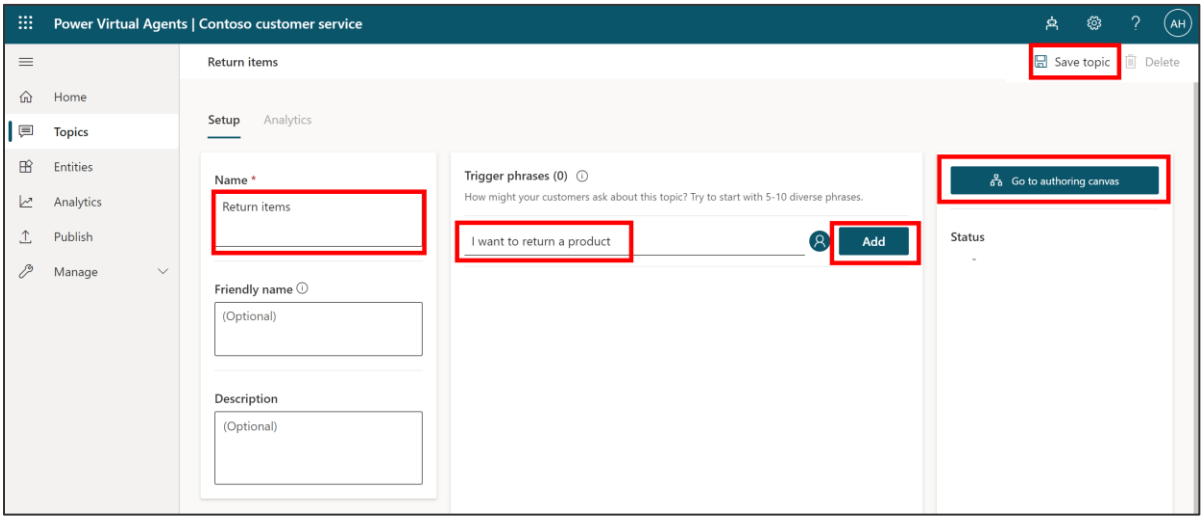
Navigate to https://powerva.microsoft.com/ and sign in with your credentials.

Task 1: Create a new topic

1. Click **Topics** in the left navigation pane.
2. At the top left of the Topics area, click **+New topic**.



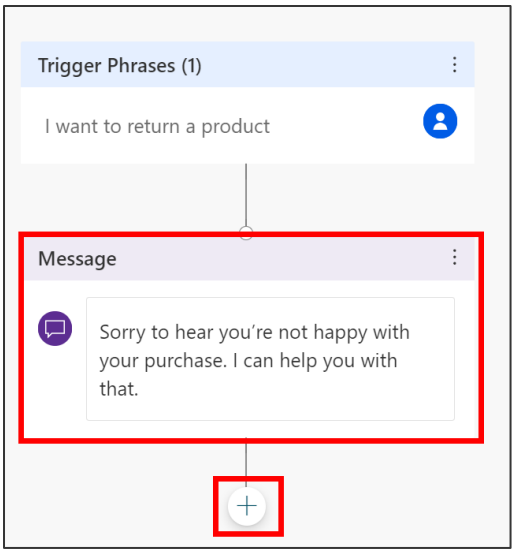
3. Enter the topic name **Return items**.
4. In the trigger phrases section, type the type the trigger phrase **I want to return a product**.
5. Click **Add**. You should add 5 to 10 trigger phrases to topics, but we'll skip ahead for purposes of this lab.
6. At the top right, click **Save topic** then **Go to authoring canvas** to open the conversation editor.



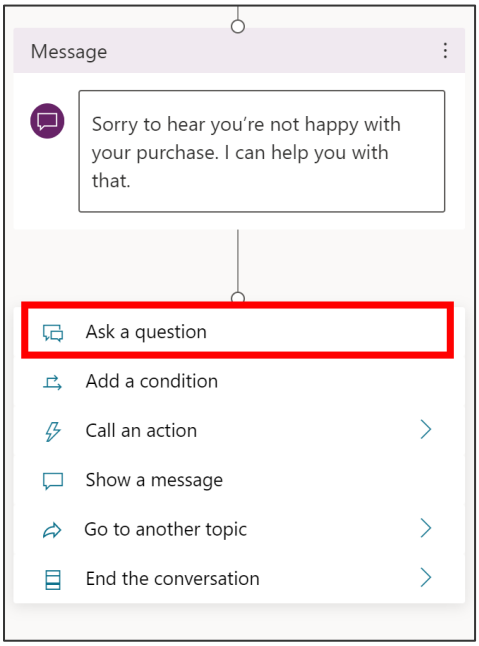
Task 2: Author the conversation, with variables

1. In the first Message node, copy and paste the following text:

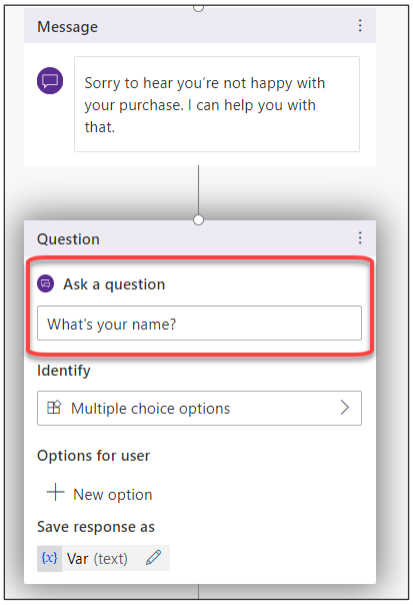
   Sorry to hear you're not happy with your purchase. I can help you with that.



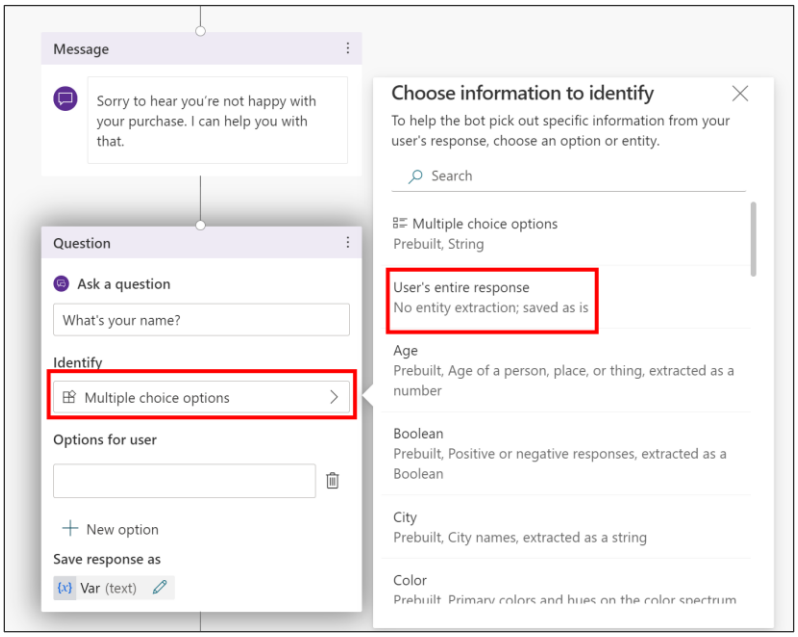
2. Under the Message node, click the **Add node** button and select **Ask a question**.



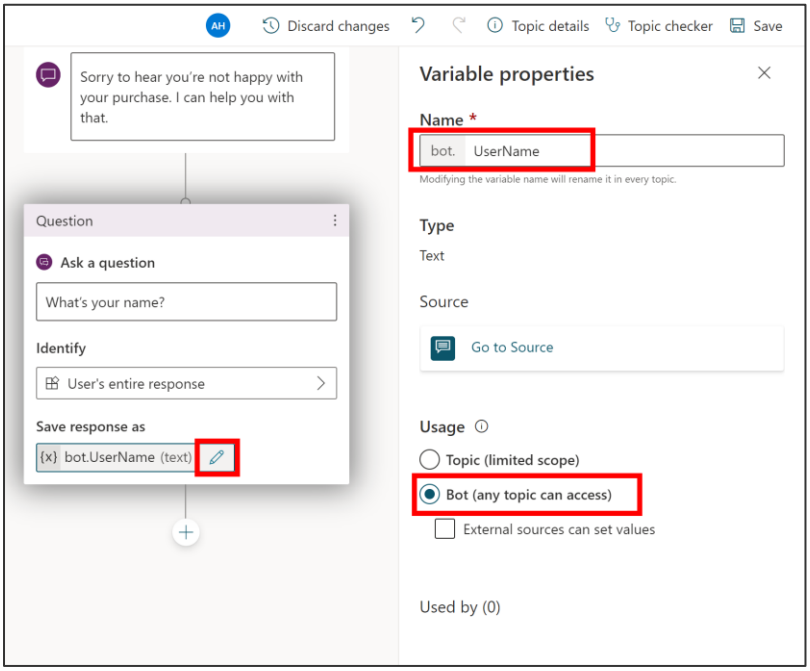
3. In the **Ask a question area** of the Question node, enter the following text: What's your name?



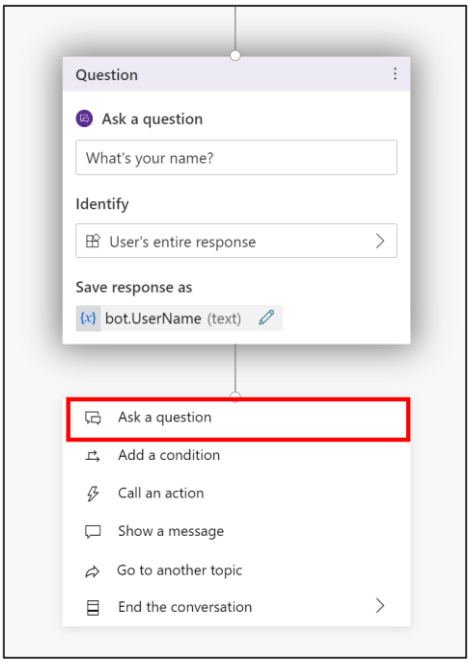
4. In the **Identify** area, click on **Multiple choice options.** This will open a search bar and menu of options. Select **User's entire response** from the popup entity list.



5. Click on the pencil icon in the **Save response as** section to set up the variable. This will open a **Variable Properties** pane on the right of the screen. Enter **UserName** as the variable name and select **Bot (any topic can access)** under the Usage section. The variable name will change to have a **bot.** prefix. This means the variable can be used anywhere in the bot, in different topics. Your variable name in the authoring canvas will also be updated to show this bot. prefix. Click on the **X** in the top right of the Variable Properties pane to close it.

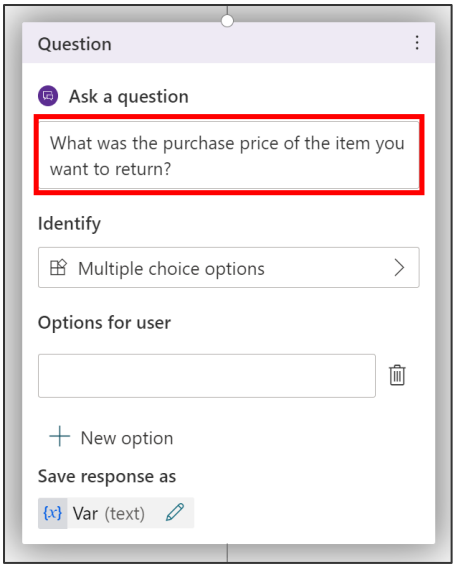


6. Under the Question node you just created, click the **Add node** button and select **Ask a question.**



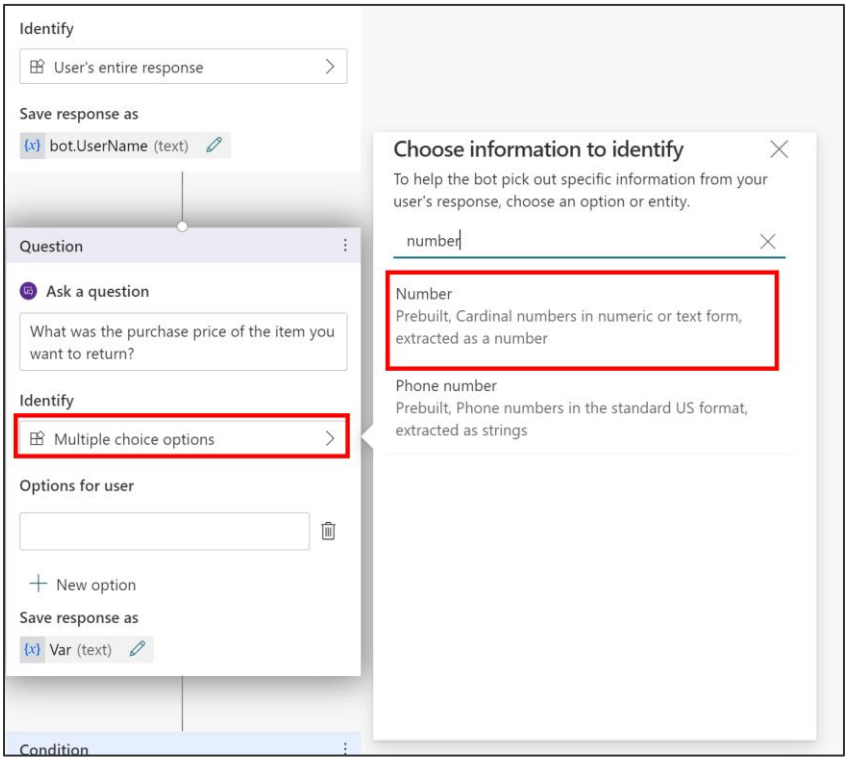
7. In the **Ask a question** area of the Question node, copy and paste the following text:

   What was the purchase price of the item you want to return?

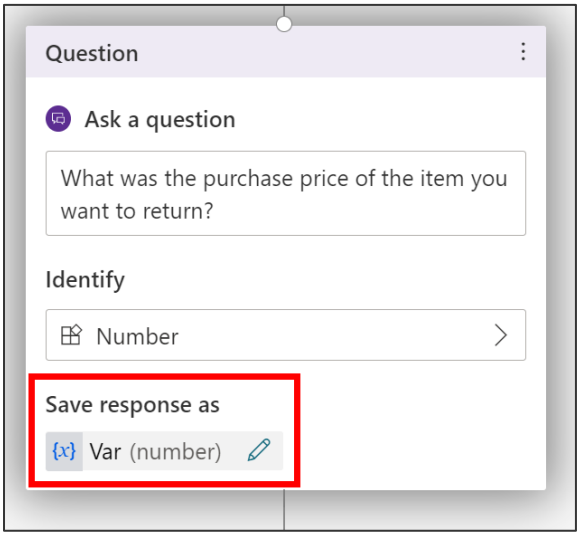


The return policy differs based on the price range. Let's add a variable to capture item purchase price.

8. In the **Identify** area, click on **Multiple choice options.** This will open a search bar and menu of options. Search and select **Number** from the popup entity list on the right. When you specify the entity, Power Virtual Agents can recognize the value whether the user types "$50" "50 bucks" "fifty" or "around $50?", and can also do math on it since it knows it is a number.

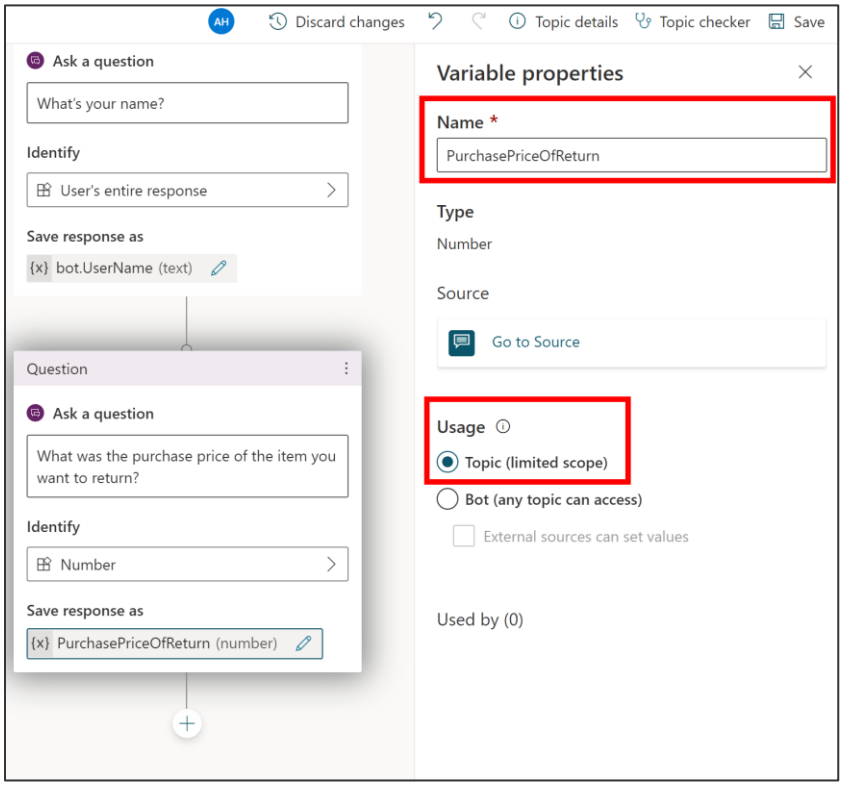


9. In the **Save response as** field, click **Var (number)**

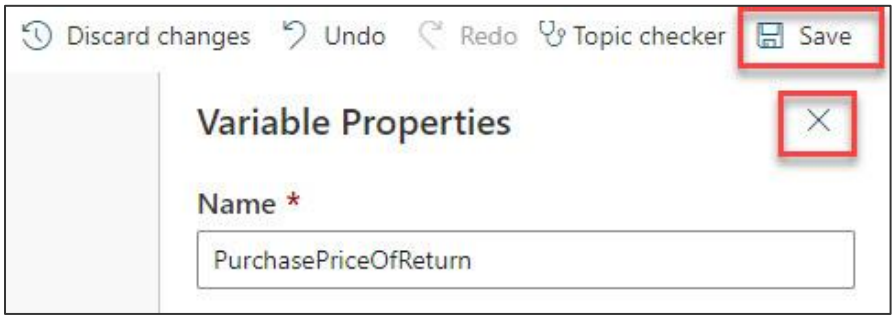


10. This will open the **Variable Properties** pane. Rename the variable from **Var** to **PurchasePriceOfReturn.** You will see the variable name automatically change on the authoring canvas as you edit the name. Naming the variable helps you remember what information is stored in it and makes it easier to retrieve again later.

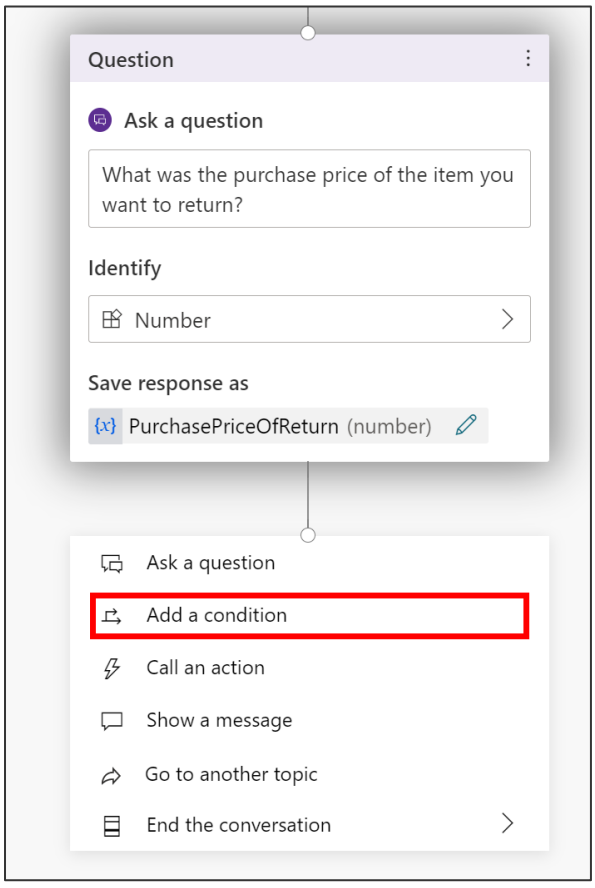
    In the **Usage** section, leave the setting as **Topic (limited scope)**.



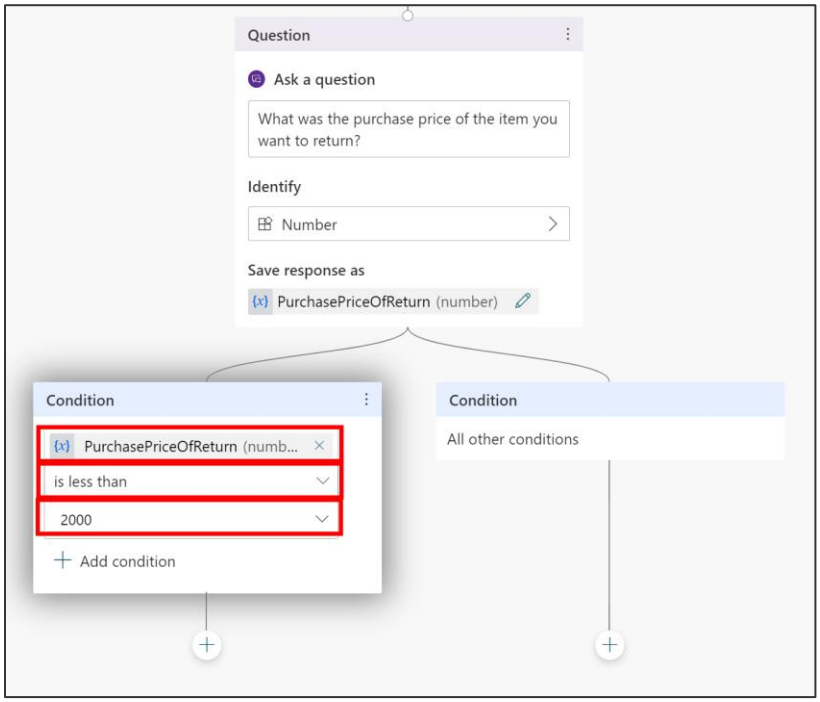
11. Click **Save** to save your changes, and then click on the X in the top right of the Variable Properties pane to close it and return to the authoring canvas.



12. Under the Question node, click the **Add node** button and select **Add a condition.**



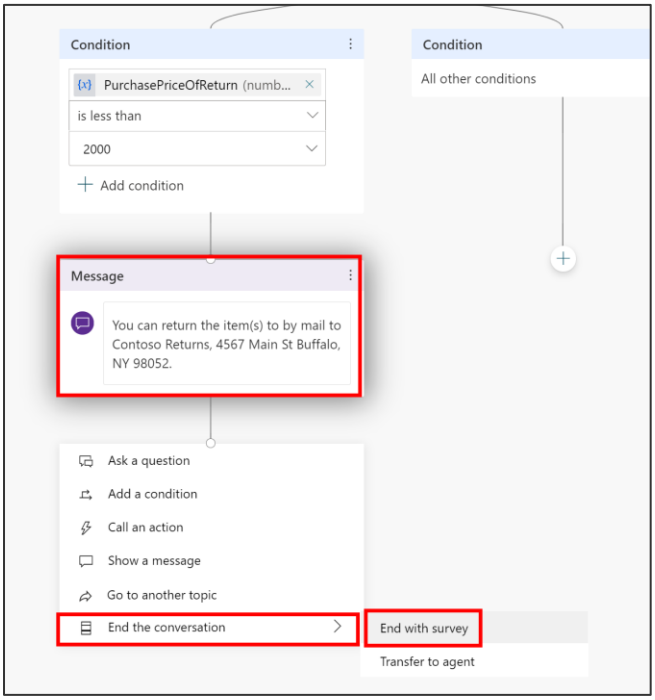
13. Branch the conversations based on conditions. Click **Select a variable** and choose `PurchasePriceOfReturn` - use the screenshot below as a guide to configure.
    - For item(s) less than $2000, customers can return the item by mail.
    - For item(s) equal to or more than $2000 (which you can handle with the "All other conditions" condition), they'll need to return it to a store.



14. For the condition when `PurchasePriceOfReturn` is less than 2000, add a node to **Show a message**:

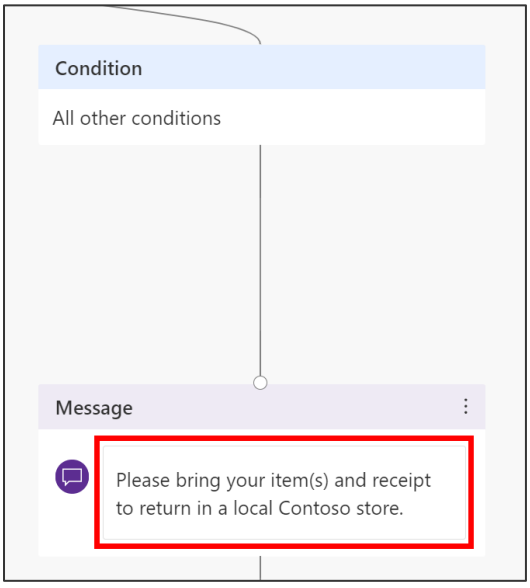
    You can return the item(s) to by mail to Contoso Returns, 4567 Main St Buffalo, NY 98052.

15. Then add another node to **End the conversation** with survey.



16. For **All other conditions** (i.e., the condition where `PurchasePriceOfReturn` is greater than or equal to 2000) – Add a node to **Show a message**:

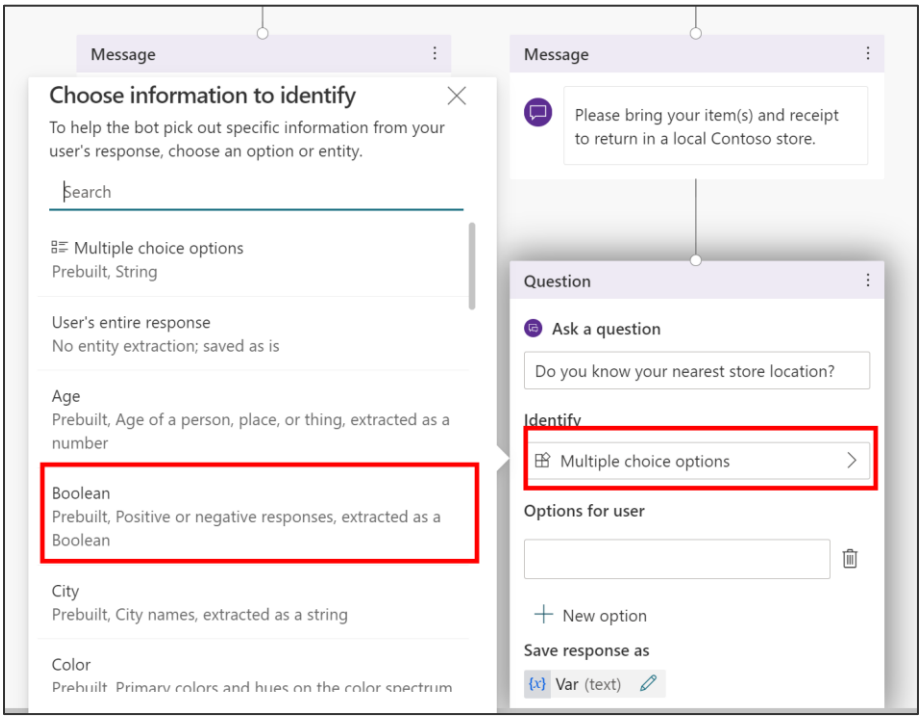
    Please bring your item(s) and receipt to return in a local Contoso store.



17. Then follow it with another node to **Ask a question**:

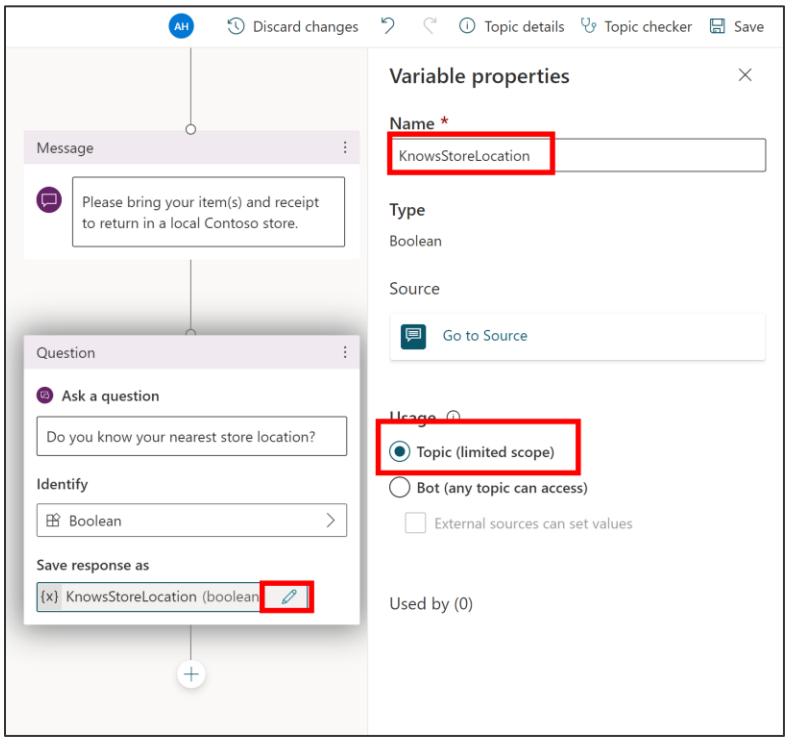
    Do you know your nearest store location?

18. In the **Identify** field, select **Boolean**. (A Boolean is a True/False question.)

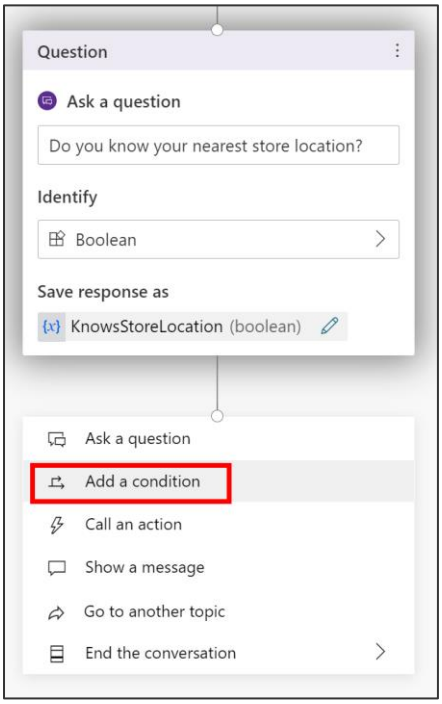


Note: Power Virtual Agents will automatically add "Yes" and "No" buttons to the chat, though you won't see them in the authoring canvas.

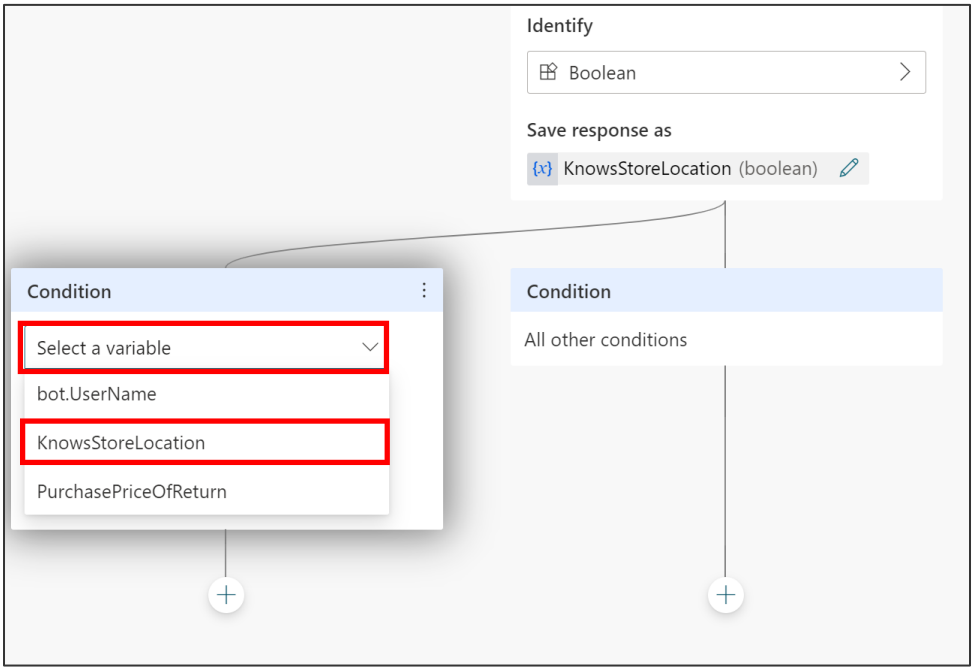
19. Name the variable **`KnowsStoreLocation`**. leaving it as a Topic variable. Click **Save** and close the Variable Properties pane. You'll see later why it can be helpful to name variables even for simple yes/no branching.



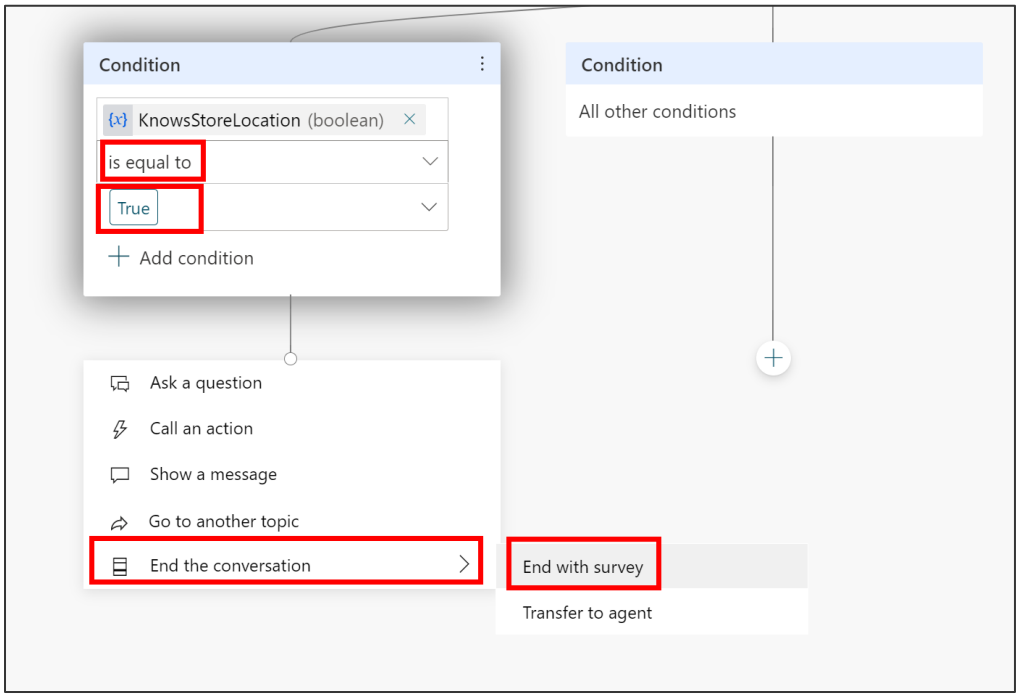
20. After the Question node, click the **Add node**  button and select **Add a condition** to configure the conversation based on user response.



21. Click **Select a variable** and choose `KnowsStoreLocation`.



22. Set a condition for if the variable is set to True (the user will see "Yes"), and follow it with an **End the conversation > End with survey** node. (See the screenshot)



In the next task, we'll handle the other ("No") condition.

Task 3: Redirect the conversation to a different topic

In the previous task, you added conditional branching and you handled what happens if the user says "Yes" (they do know their nearest store location). Now, we'll handle what the bot should do if the user says "No" (they don't know where the nearest store is).

We already have a topic about store locations (we edited it in Lab 2). So, rather than repeat that information in this topic, we will link to the other topic.

Reusing topics by using redirection has these benefits:

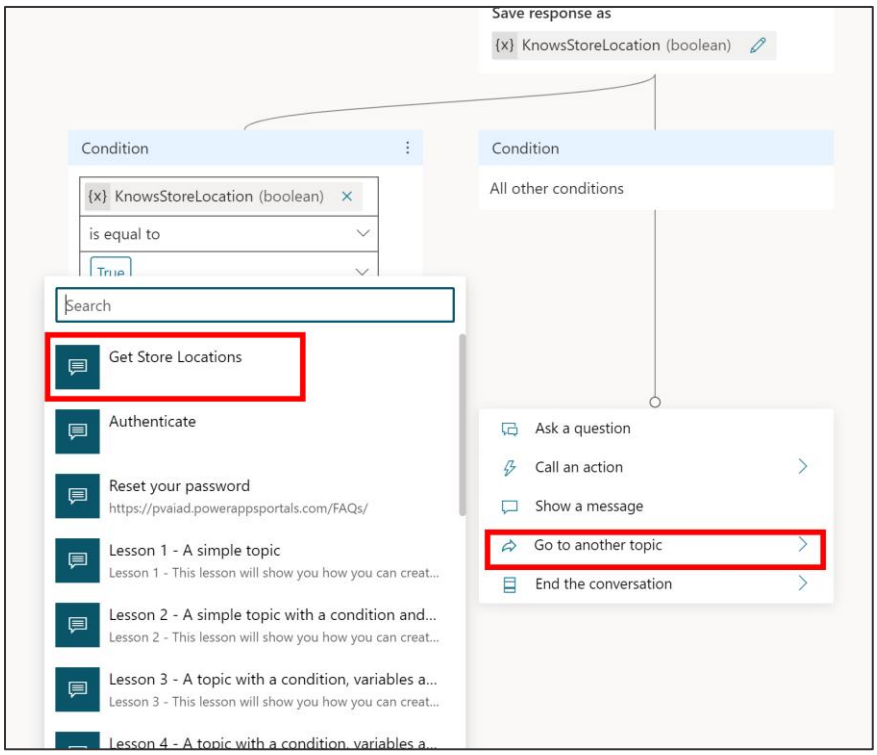
- It reduces the number of places you keep the same information in your bot, so you can update it in one place if necessary and not worry about trying to find all the different places you repeated the same information.
- Smaller topics are more efficient to load into the authoring canvas.
- And, depending on how large your company is, you may find that you divide topic authoring up among subject matter experts.

Follow these steps:

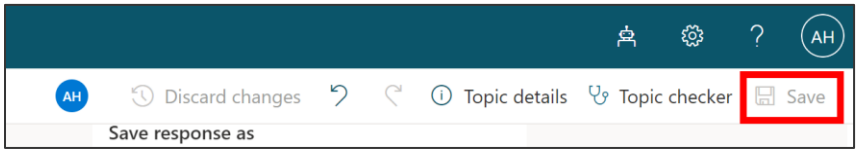
1. Under the Condition node for **All other conditions** (which will handle any response other than "Yes"), click the **Add node** button, click **Go to another topic.**

   You'll see a list of all the topics in your bot.

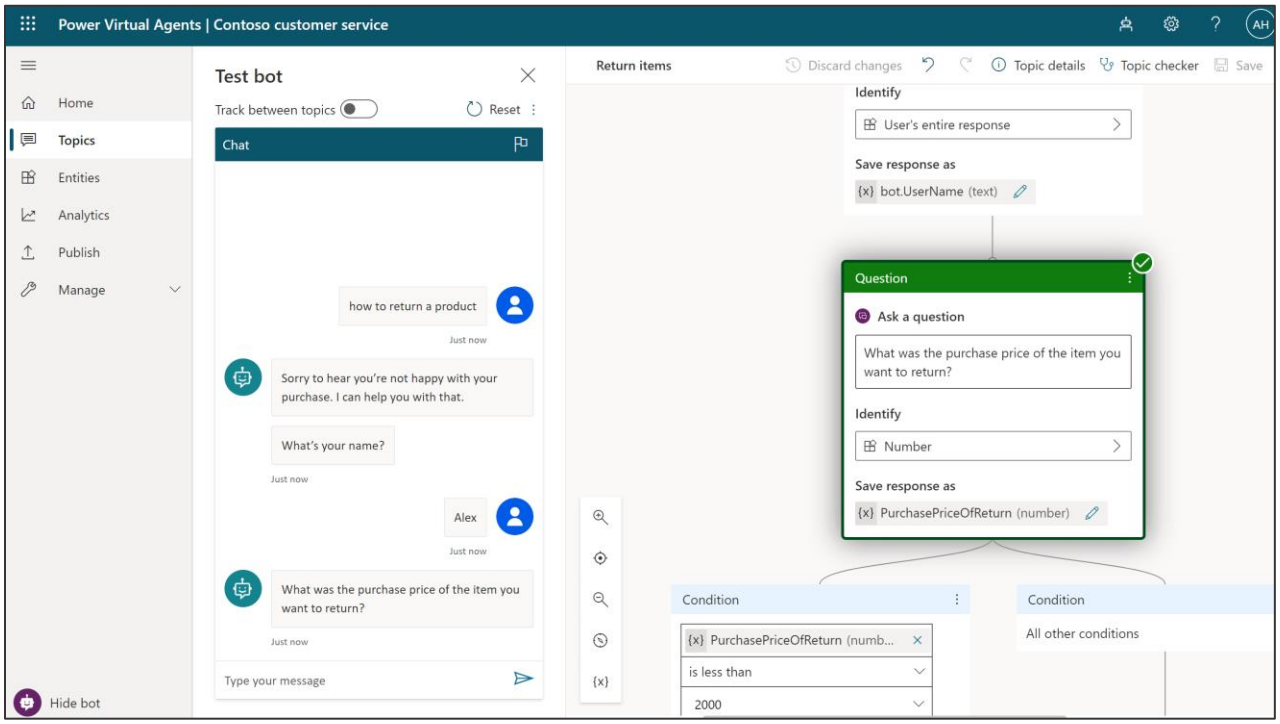
2. Click the topic we completed in Lab 02 – **Get store locations**.



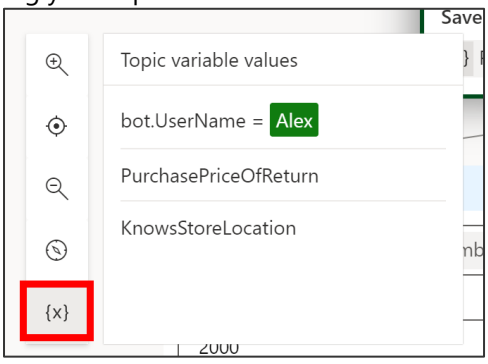
3. Click **Save.**



4. Open the **Test your bot** feature (in the lower left corner of the page) if you can't already see it. Test the topic you've just created by typing "**how to return a product**" in the test bot pane, and hit Enter.

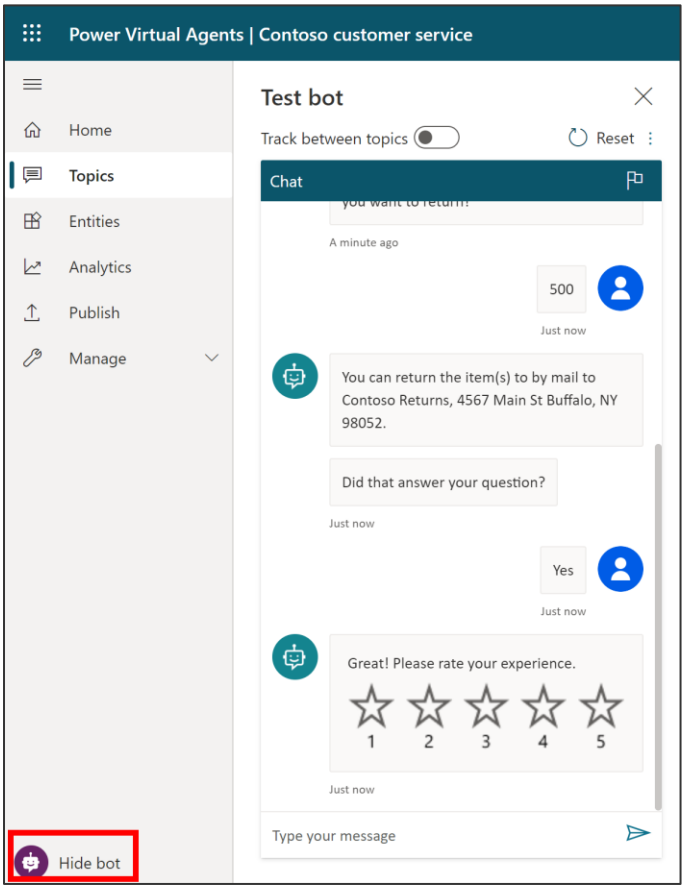


5. Click the ***{x}*** button on the utility bar to open the **Topic variable values** screen (see screenshot below) when you test the topic. Notice that as you chat with the bot, the values of the variables is set. You can see why naming your variables can help you when testing your topic!



Note: To try various branches of the conversation, click **Reset** at the top of the Test bot pane. This also resets the variables to be empty.

6. When you're done testing your bot, click **Hide bot** to free up space on the screen.



Exercise 2: Use entities and slot filling in Power Virtual Agents chatbots

Understanding and using entities in Power Virtual Agents

Natural language understanding is the ability for Power Virtual Agents to understand what the user is telling it. For example, if the user says "I tried to use my gift card but it doesn't work," the bot is able to route the user to the topic related to gift cards not working – even if that exact phrase isn't listed as a trigger phrase.

Natural language understanding also helps the bot identify entities in a user's input. An entity represents a real-world subject, such as a phone number, zip code, city, or even a person's name. Your bot can smartly recognize the relevant information from user input and save it for later use.

For example, if the user types "I want fifty purple laptops", the AI can understand that:

- "fifty" is the number "50" and is also the number of products to purchase;
- "purple" is a color and is the color of the products to purchase; and
- "laptops" is the product the person wants to purchase.

Some of these (such as numbers and colors) have already been taught to the AI for every Power Virtual Agents chatbot; others (such as the fact that "laptops" is a product or that this purple is specifically the color of a product) have to be specified by the bot author, as we will show in this lab.

There are two types of entities:

- **Pre-built entities** represent the most commonly used information, such as age, colors, numbers, and names. Power Virtual Agents bots recognize these automatically.

- **Custom entities** are entities you make. While the pre-built entities cover commonly used information types, sometimes you'll need to teach the bot's language understanding model some domain-specific knowledge. For instance, you may need to create a custom entity for your product types.

Smart match and synonyms can make your bot even smarter:

- **Smart match** provides the flexibility to let the bot match the user's input to an entity that is a near match but not perfect. Specifically, it lets the bot autocorrect misspellings and expands the matching logic semantically, such as automatically matching "softball" to "baseball". You can turn this off if you need a match to be perfect, such as if the entity contains model numbers or error codes.

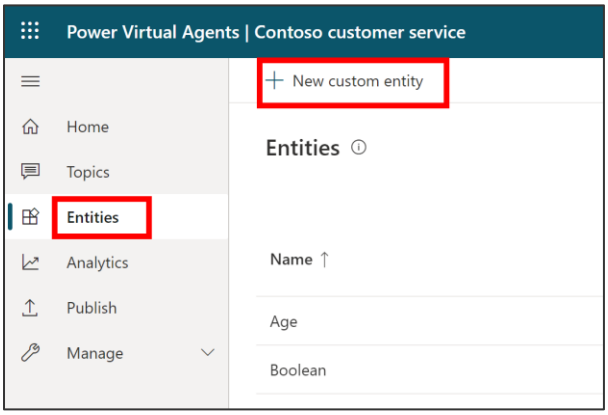
- **Synonyms** allow you to recognize that something the user typed matches an option you provided. For example, for "free shipping" you can add "complimentary shipping" as a synonym. For "expedited shipping", you can add "2 day shipping" or "overnight shipping" as synonyms. If the user types any of these, they will be matched appropriately.

Task 1: Create new custom entity

Now we are going to build a new topic on shipping options, using a custom entity.

1. In Power Virtual Agents, click the **Entities** tab on the left navigation pane.

2. Select **+New custom entity** at the top of the Entities page. Choose **Closed list** as the method.



   You'll see the entity creation window.

3. In the **Name** box, enter a name for the entity:

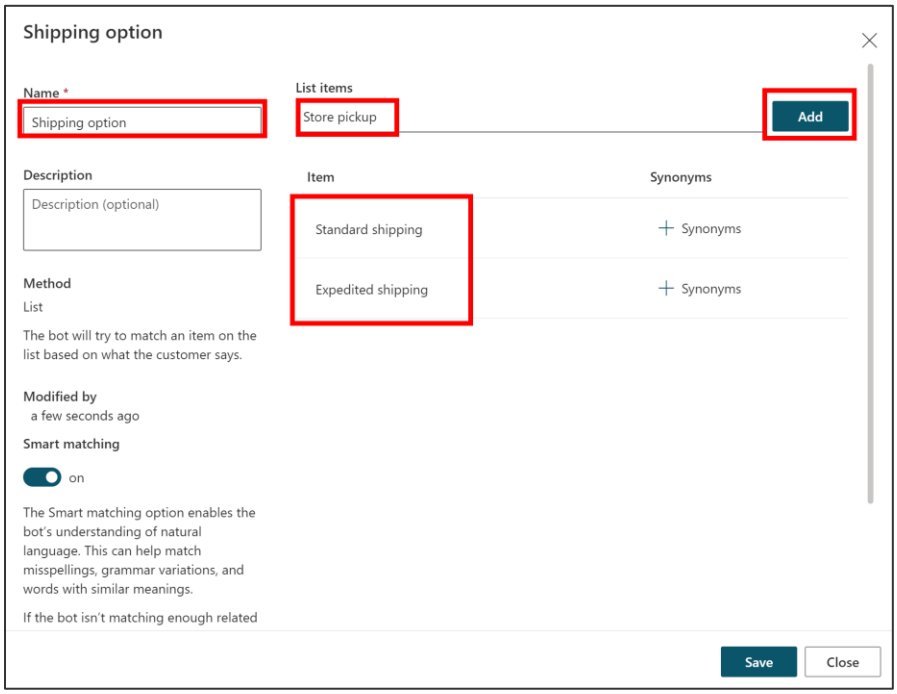
   Shipping option

4. In the **List items field** at the **Enter item** prompt, type:

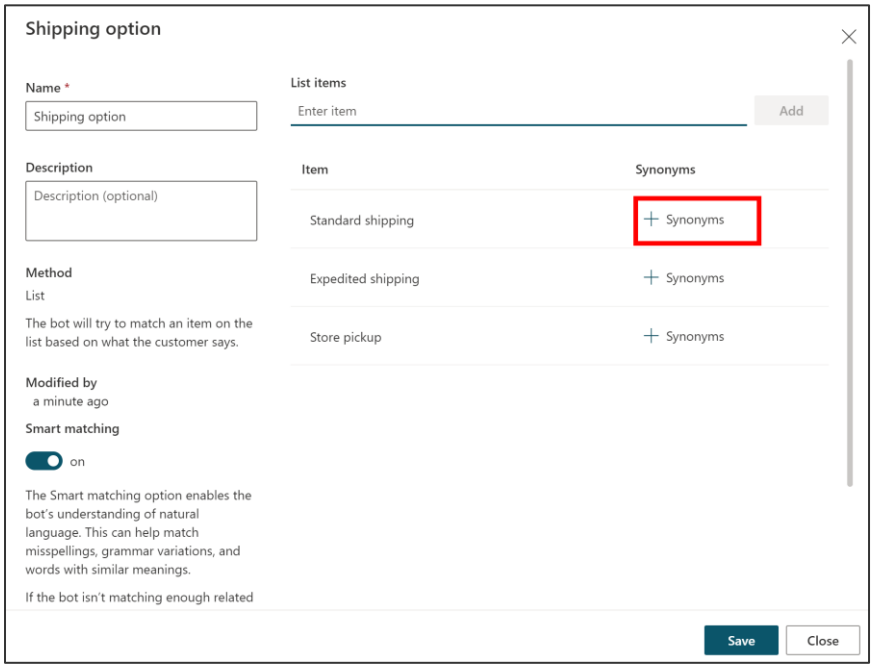
   Standard shipping

5. Click **Add**.

6. Now, add these 2 additional list items:
    - Expedited shipping
    - Store pickup

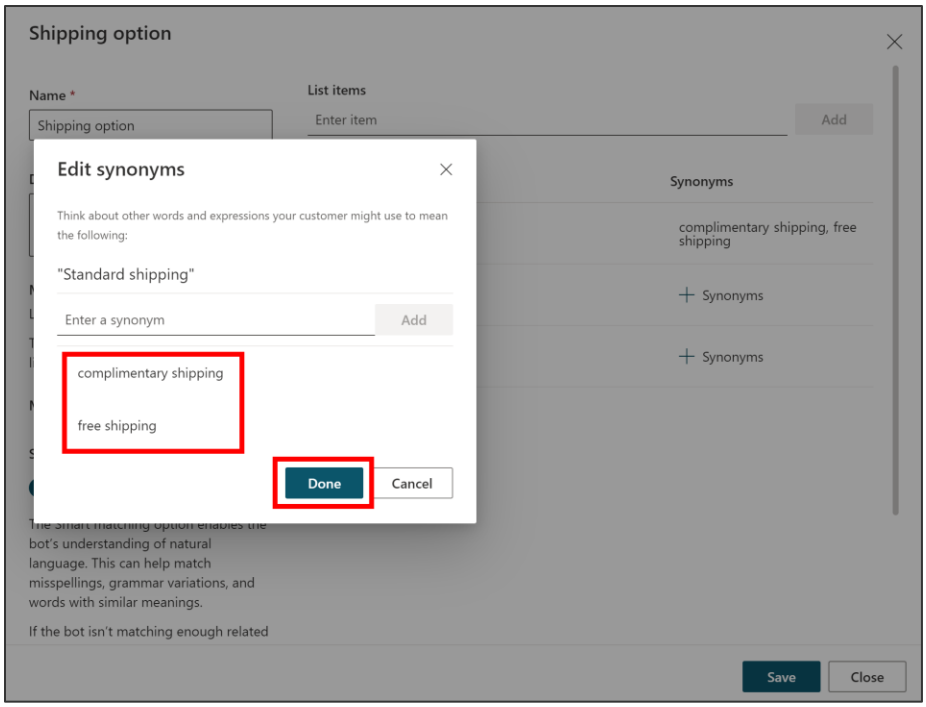


7. Next to each list item, click **+ Synonyms** to add the synonyms.

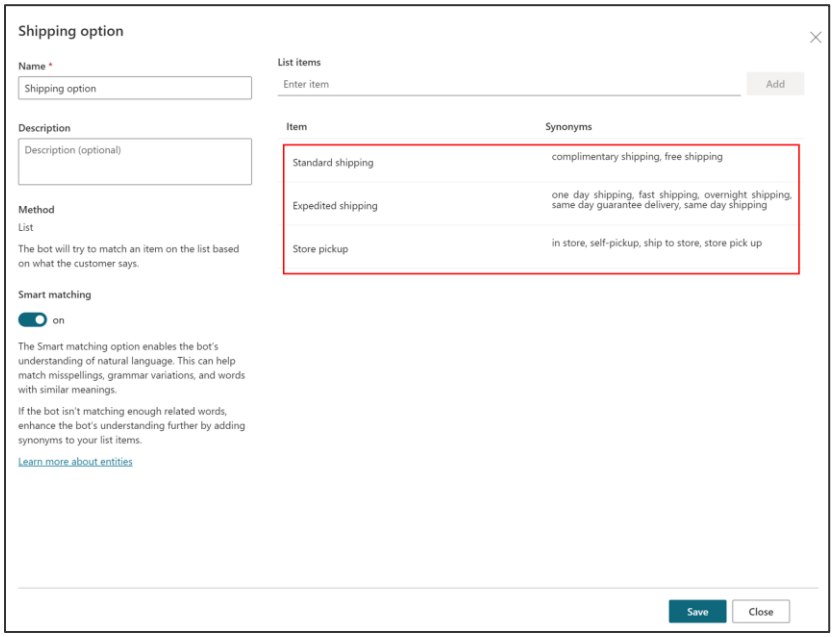


8. Add the following synonyms to "**Standard shipping**" . Enter a synonym, and click **Add.** When you finish adding all of them, click **Done.**

   Standard shipping: free shipping, complimentary shipping



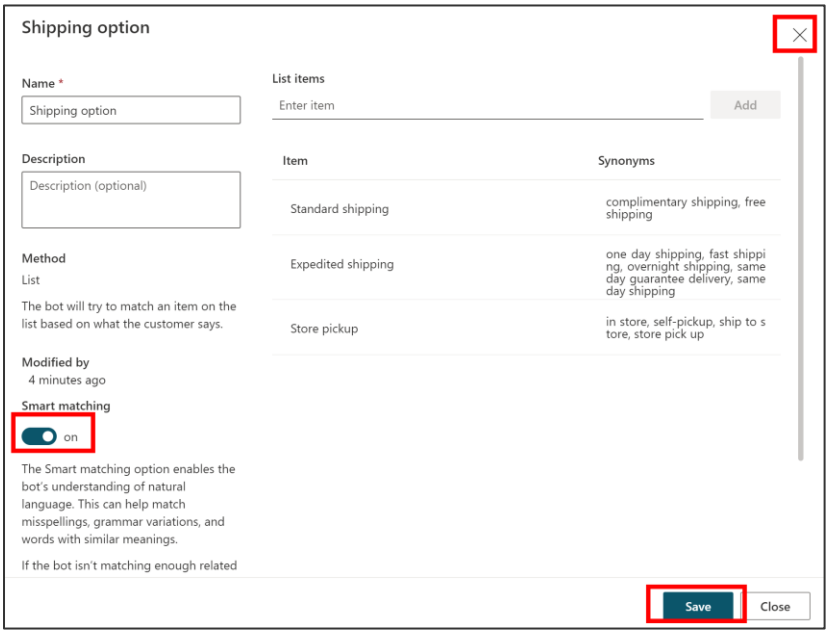
9. Repeat step 8 and add these synonyms for **Expedited shipping** and **Store pickup**.
    - Expedited shipping: same day shipping, same day guarantee delivery, overnight shipping, fast shipping, one day shipping
    - Store pickup: store pick up, ship to store, self-pickup, in store



10. Make sure the **Smart matching** toggle is set to **on**.

11. **Save** the entity.

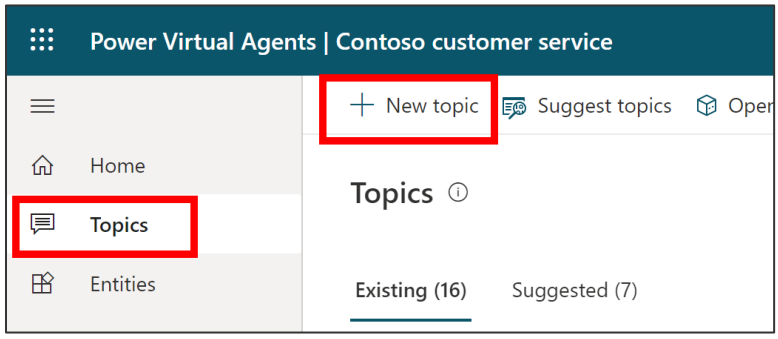
12. **Close** the window.



Task 2: Create new topic

Now, let's create the conversational topic that will use your new entity.

1. Click the **Topics** tab in the left navigation, and then click **+ New topic** at the top of the Topics page.



2. Name the topic **Shipping options**.

3. Then add 5 **Trigger phrases**:

   What are my shipping options

   Can I have free store pickup

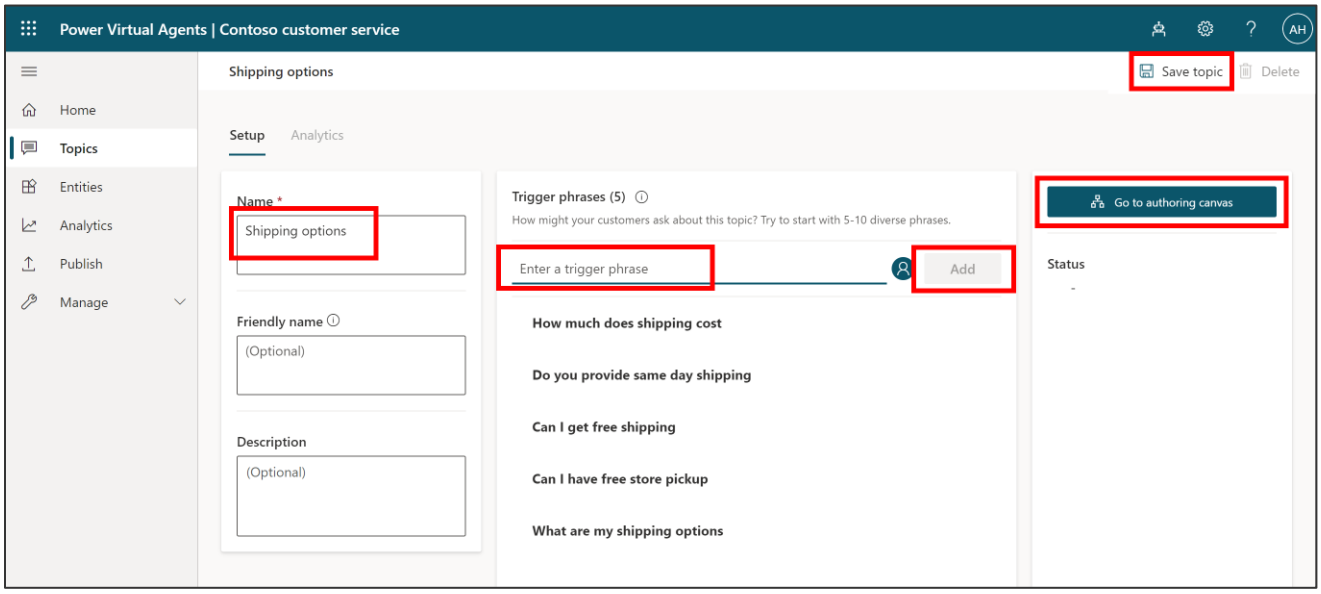
   Can I get free shipping

   Do you provide same day shipping

   How much does shipping cost

4. Click **Save topic**.

5. After the topic successfully saves, click **Go to authoring canvas**.



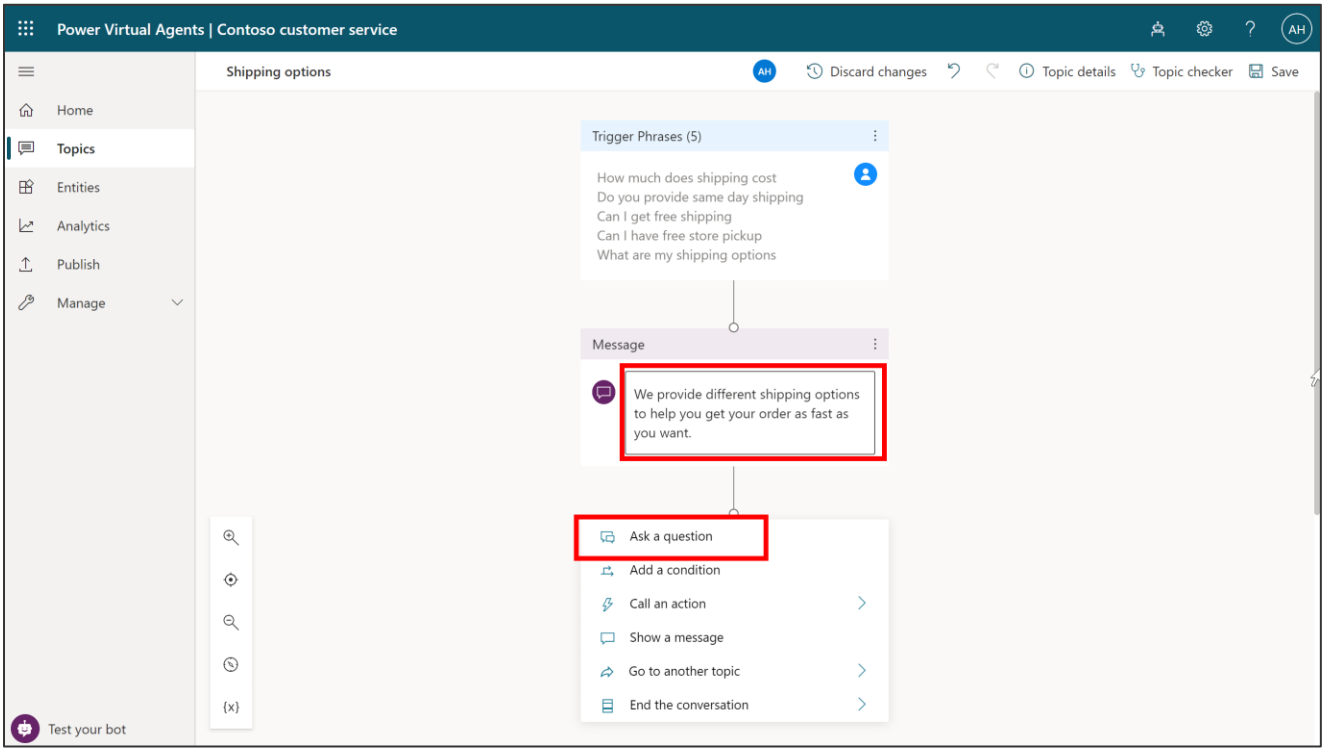
Task 3: Use the new entity in the conversation

The previous task set up the "Shipping options" topic. Now we'll design the topic and use the entity in it.

1. In the last step of the previous task, you opened the authoring canvas for the "Shipping options" topic. In the empty Message node, enter the following text:

   We provide different shipping options to help you get your order as fast as you want.

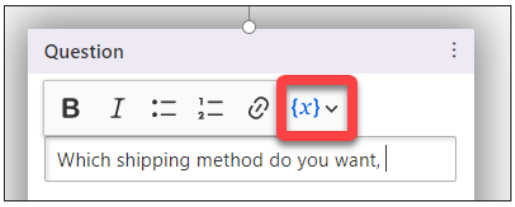
2. Under the Message node, click the **Add node** button and select **Ask a question**.



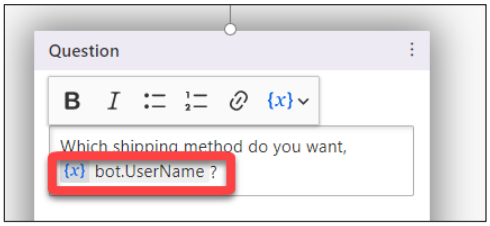
3. We can use the bot variable we set up in the previous topic to recall the person's name and personalize the chat a bit more. In the **Ask a question** area of the Question node, enter the following text:

   Which shipping method do you want,

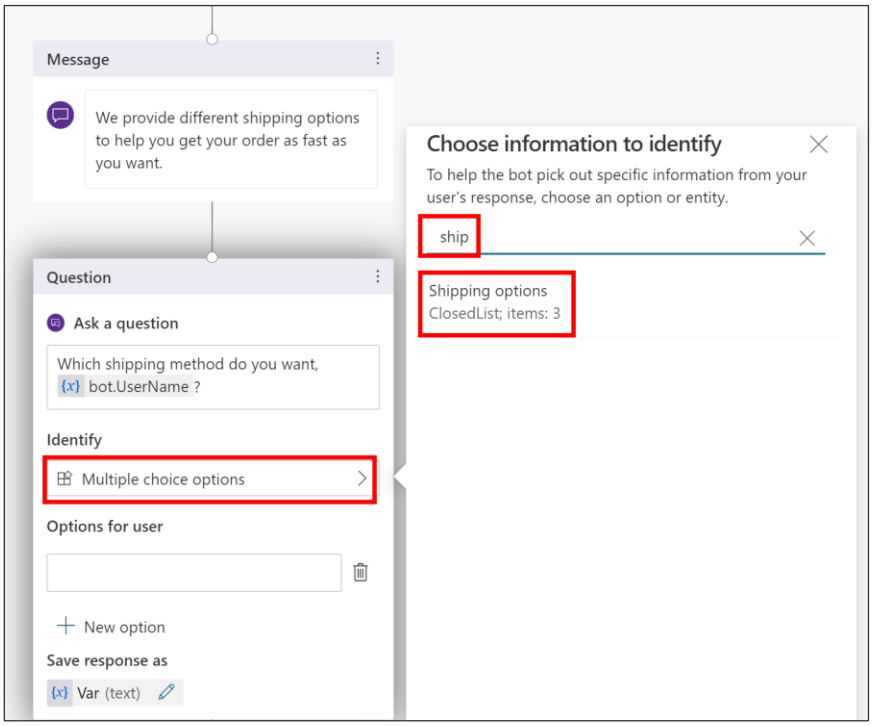
   And then click on the variable icon to add a variable into your question



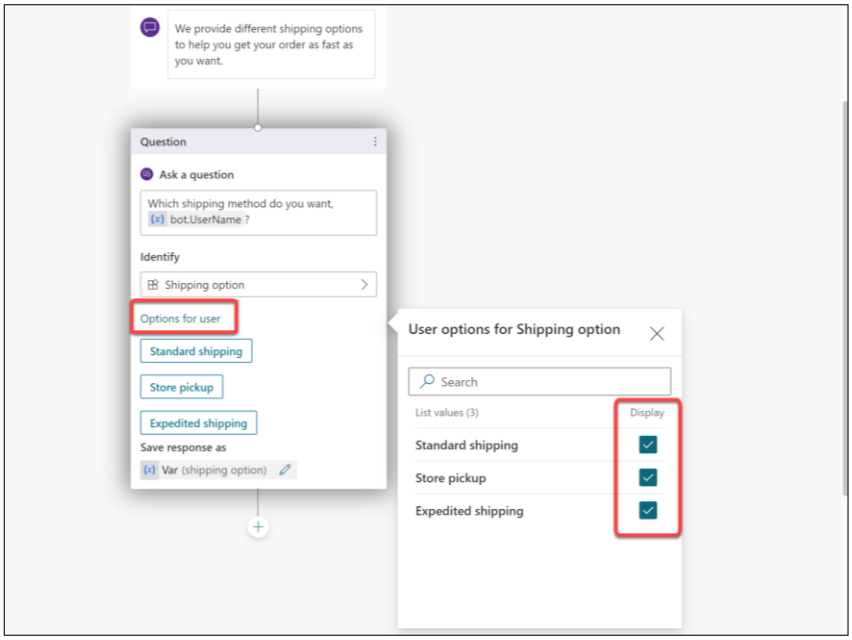
4. Select the **bot.UserName** variable you created earlier. This will remember the person's name that they entered from the chat in the previous topic about returns. Add a **?** to the end of your question to complete it.



5. Under **Identify**, click **Multiple choice options** and change the entity to **Shipping option**. (You can type in the search bar to quickly find the entity you're looking for.)

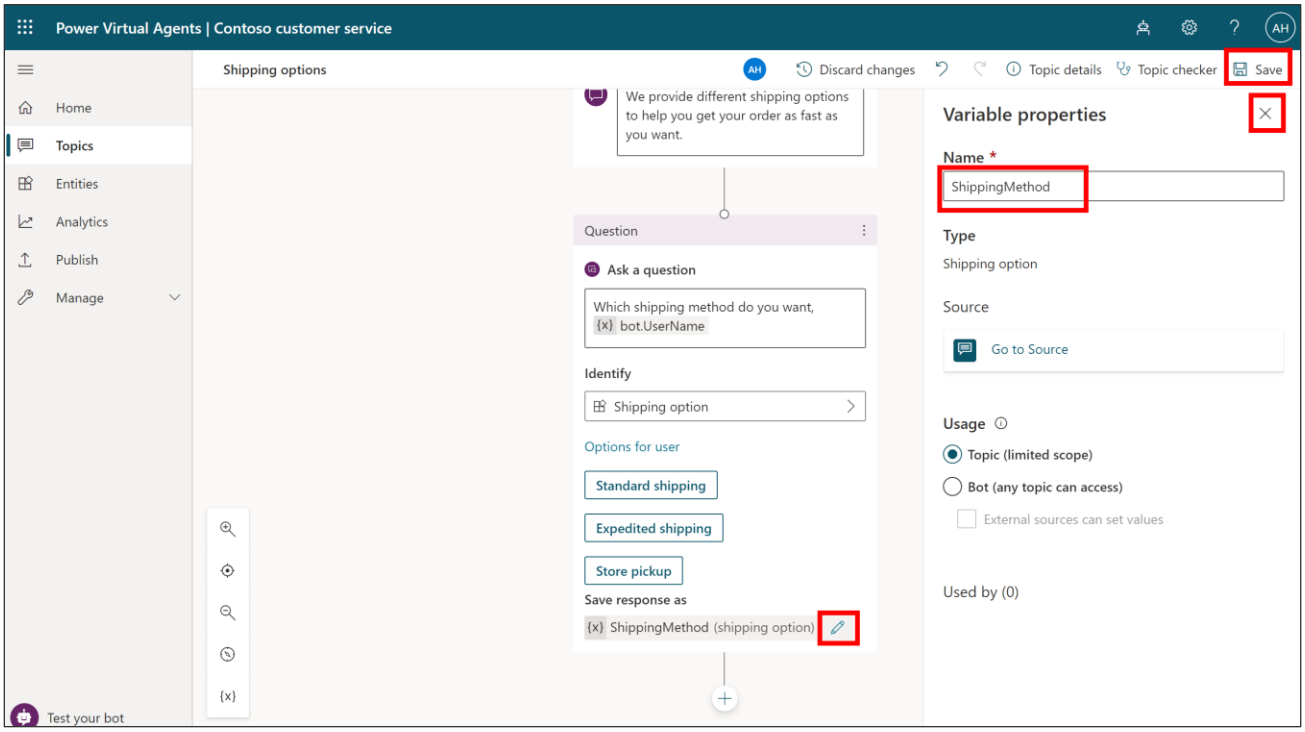


6. We want to show the user a list of their shipping options to choose from. Click **Select options for user** and then click the **Display** checkbox for each of the items.

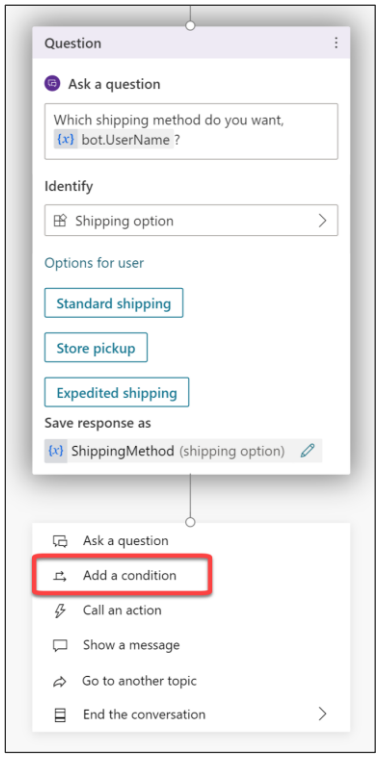


Note: The List items defined in the Entity are shown. The user can click any of those buttons in the chat window, or if they type either a list item or its synonym, the bot will respond the same way. You don't have to show response options as buttons, but for short lists it is usually the best option to ensure a smooth conversation for the user and the bot.

7. Rename the variable from **Var** to `ShippingMethod`. Leave it as a Topic (limited scope) variable. Click **Save** and close the Variable Properties window.

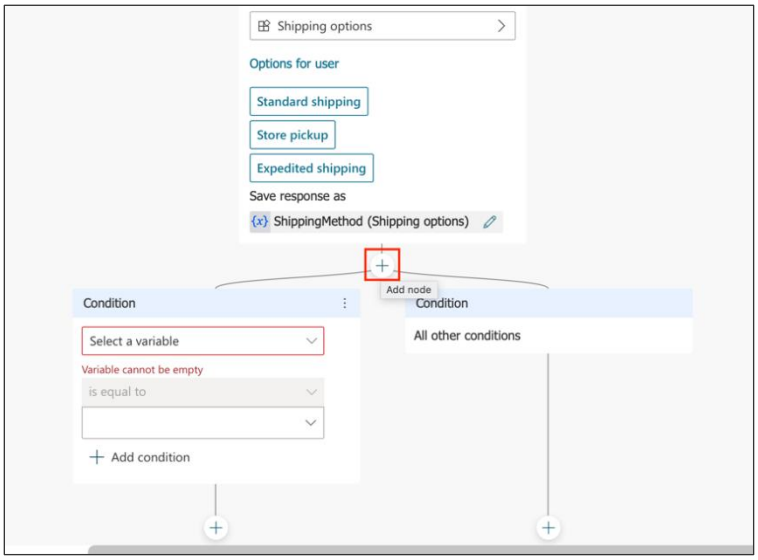


8. Under the **Question** node, click the **Add node** button and select **Add a condition.**



This will create 2 conditional branches: a **Condition** node and an **All other conditions** node.

9. Using the following screenshot as guide, add another condition by hovering under the Question node where the two condition nodes join it, clicking the **Add node** button that appears, and selecting **Add a condition**.

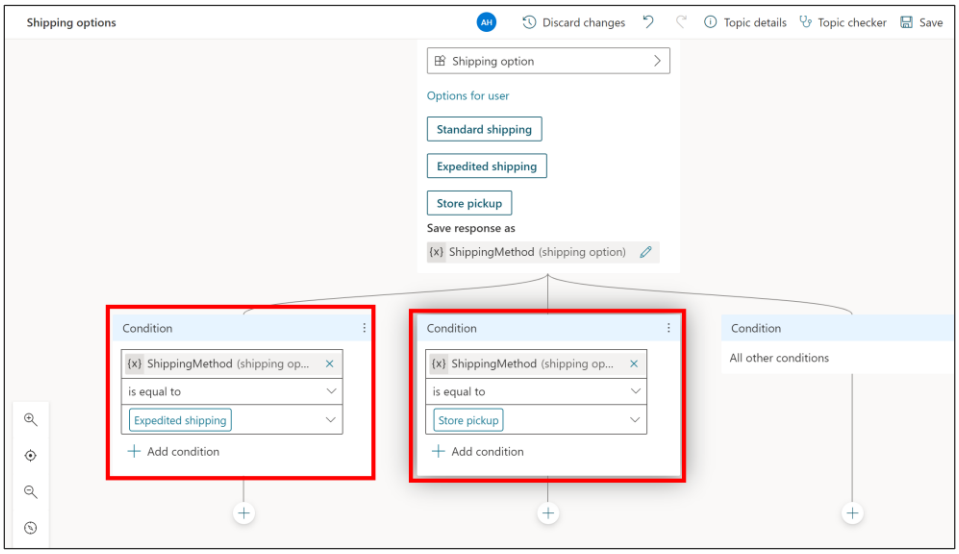


10. For the first condition (the one on the left), click **Select a variable** and choose **ShippingMethod**.

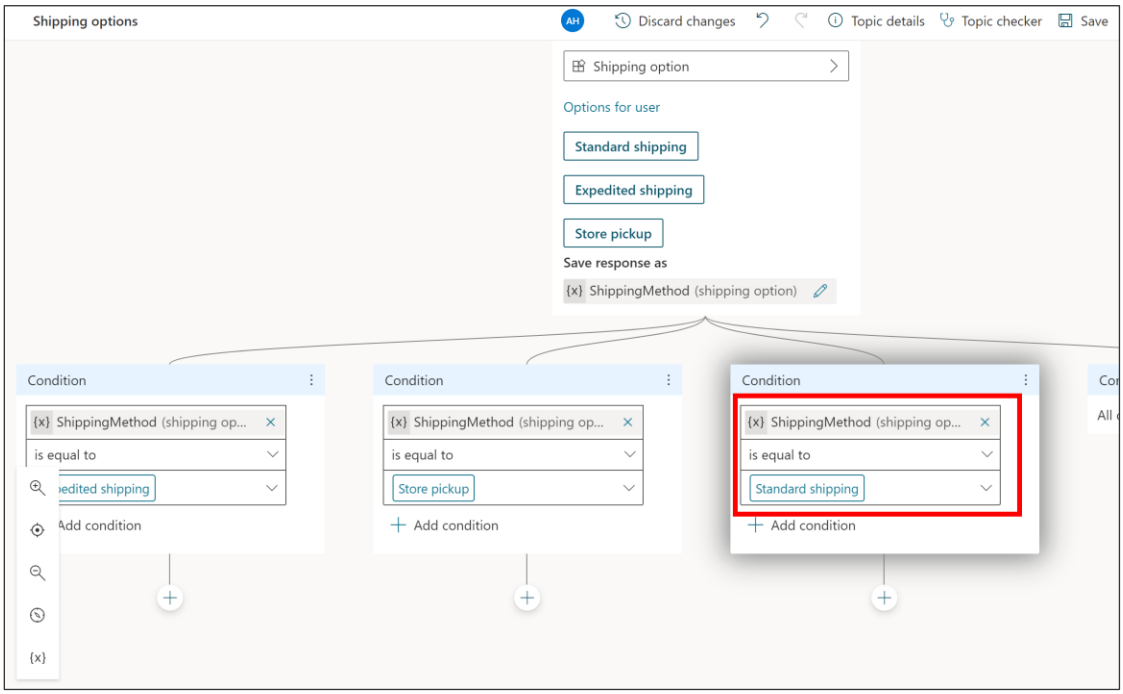
11. Leave the operator as "is equal to" (which is the only option for text entities).

12. In the bottom field of the condition, where you enter the value, click in the field and choose **Expedited shipping** from the drop-down list.

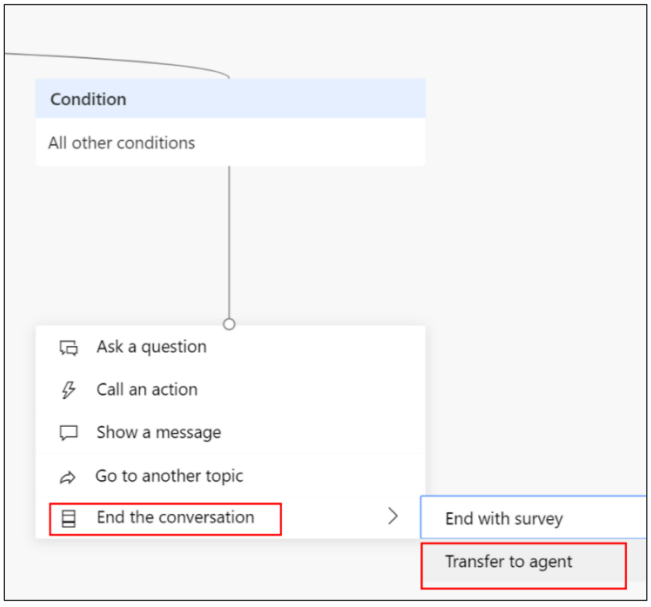
13. Configure the second condition for the **Store pickup** shipping option. (It uses the same variable (ShippingMethod).)



14. You could leave "Standard shipping" to be covered by "All other conditions" but that is risky because the user could type anything and arrive at the answer for standard shipping. So, create one more conditional branch (using the instructions in step 9) for standard shipping and set the conditions for it.



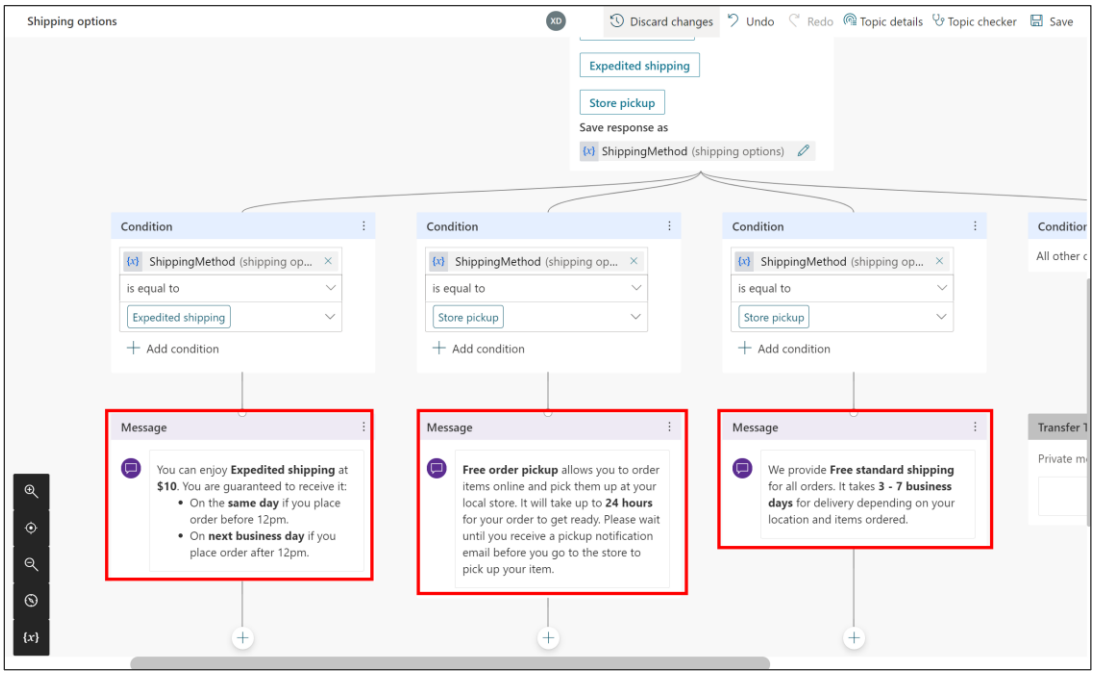
15. For the "All other conditions" node, let's assume the user needs to talk to a human agent to understand what shipping option they wanted. Under the **All other conditions** node, click **Add node** and then **End of conversation > Transfer to agent**.



16. For each remaining condition, add one **Show a message** node after the Condition node. The messages are as below:
    - Expedited shipping:
      You can enjoy **Expedited shipping** at **$10**. You are guaranteed to receive it:
        - On the **same day** if you place order before 12pm.
        - On **next business day** if you place order after 12pm.

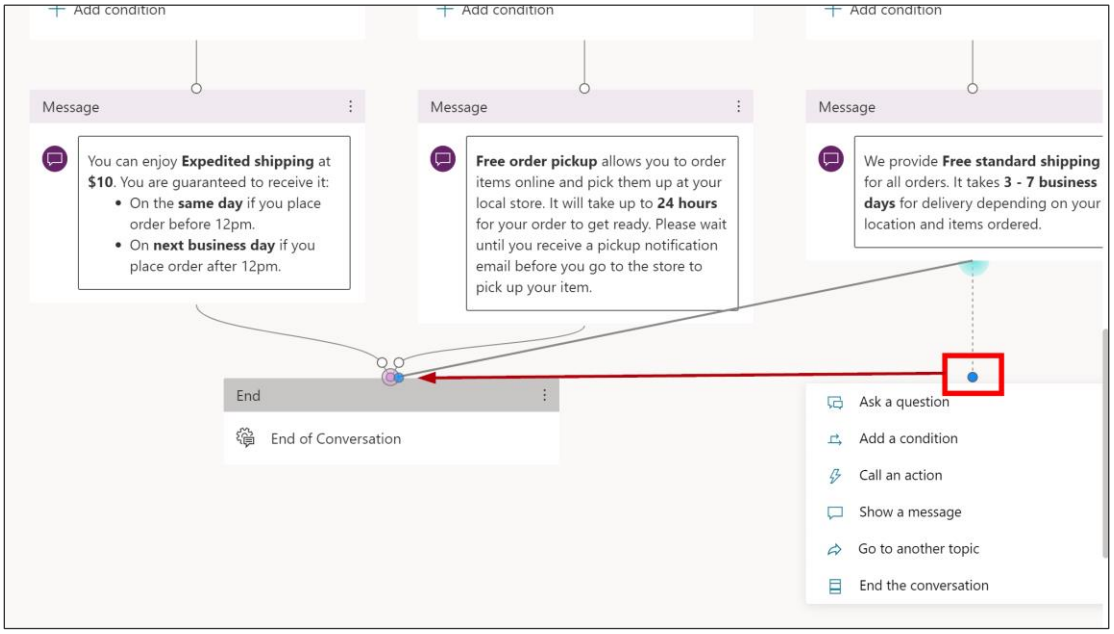
    - Store pickup:
      Free order pickup allows you to order items online and pick them up at your local store. It will take up to **24 hours** for your order to get ready. Please wait until you receive a pickup notification email before you go to the store to pick up your item.

    - Standard shipping:
      We provide **Free standard shipping** for all orders. It takes **3 - 7 business days** for delivery depending on your location and items ordered.



17. Under the Expedited Shipping Message node, add an **End the conversation** > **End with survey** node.

18. Under the other Message nodes, click **Add node** and drag to the **End of conversation** node you just added.



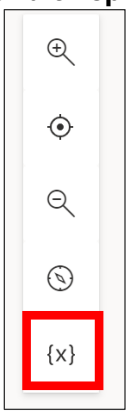
19. **Save** the topic.

Task 4: Test the conversation and see slot filling in action

As you learned in the introduction to this lab, **slot filling** defines the value of a specific variable in your chatbot conversation *for that conversation*. We will test your topic three ways and see the different ways your ShippingMethod variable is filled.

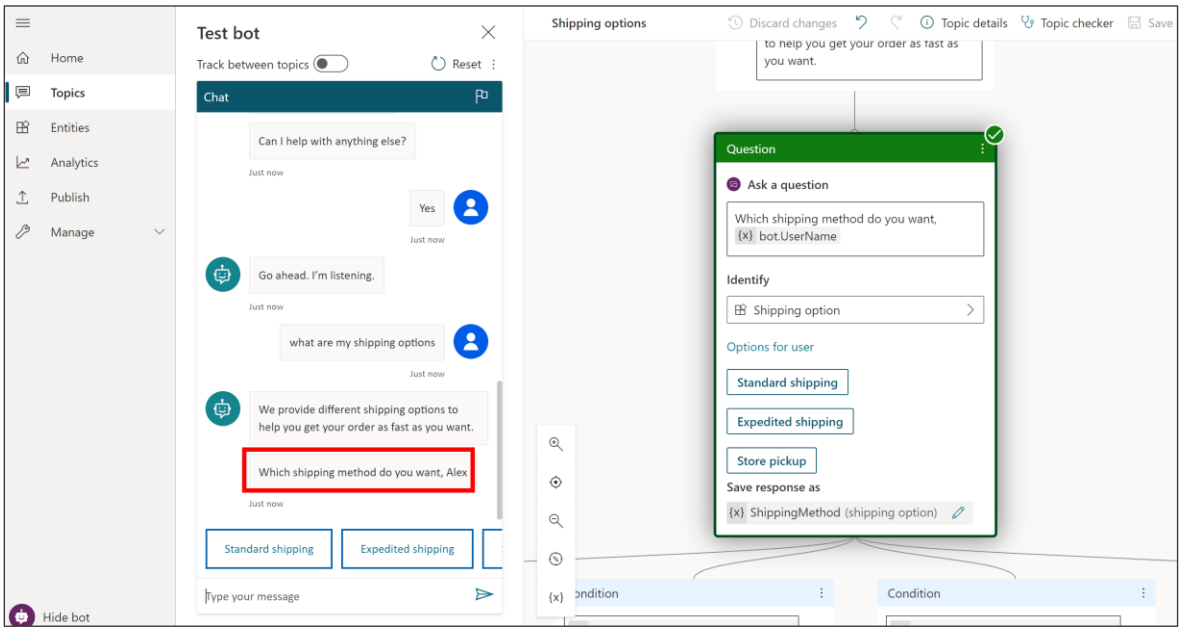
1. Open the Test bot pane (click **Test your bot** in the lower left corner of the Power Virtual Agents window).

2. Click the **{*x*}** button on the utility bar to open the **Topic variable values** screen (see screenshot below).

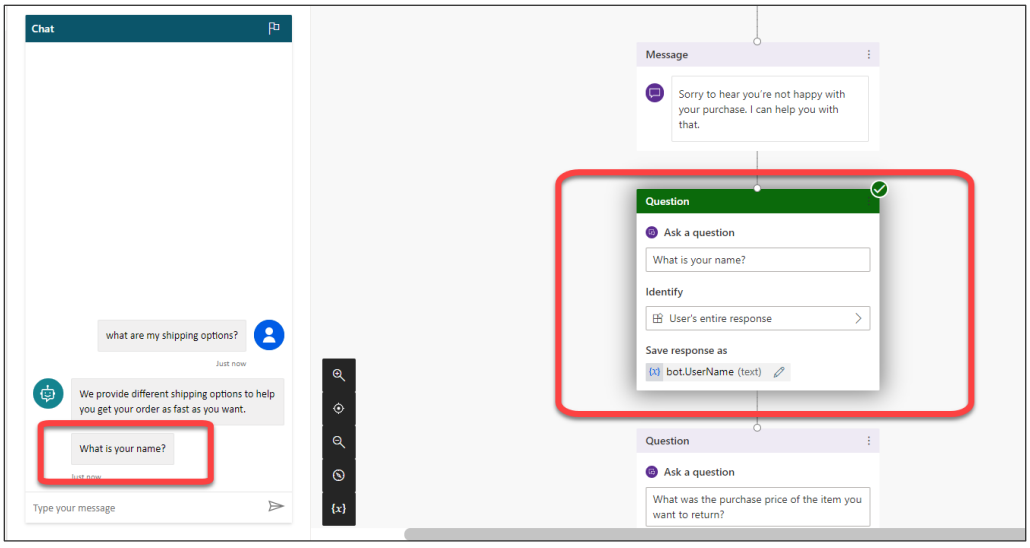


3. Let's start with the returns topic you created earlier in the lab, to simulate a conversation where the user has multiple questions, so you will see the bot variable in action. Type "**I want to return a product**" in your test bot pane. Enter your name and complete the conversation about returns with whatever options you like, right through to completing the satisfaction survey.

4. When the bot asks if it can help you with anything else, choose **Yes**. Then type "**what are my shipping options?**" to start the topic you have just created. Note that the bot remembers the user name from the previous topic, using the bot variable.

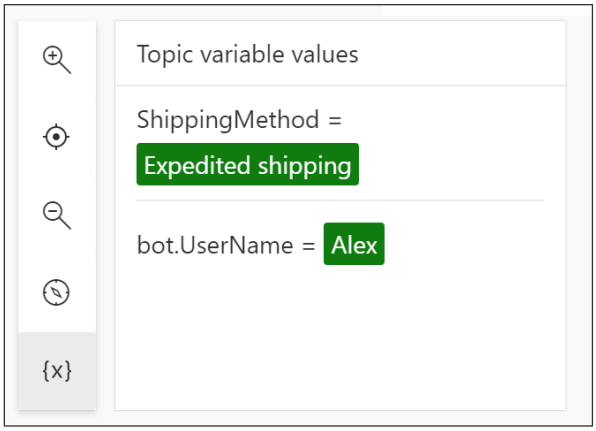


If the user had just started the conversation with the shipping topic, the bot realizes it doesn't yet have a value for this variable, so will automatically pick up the question from the topic where the bot variable is defined, and then continue on the topic that was triggered.

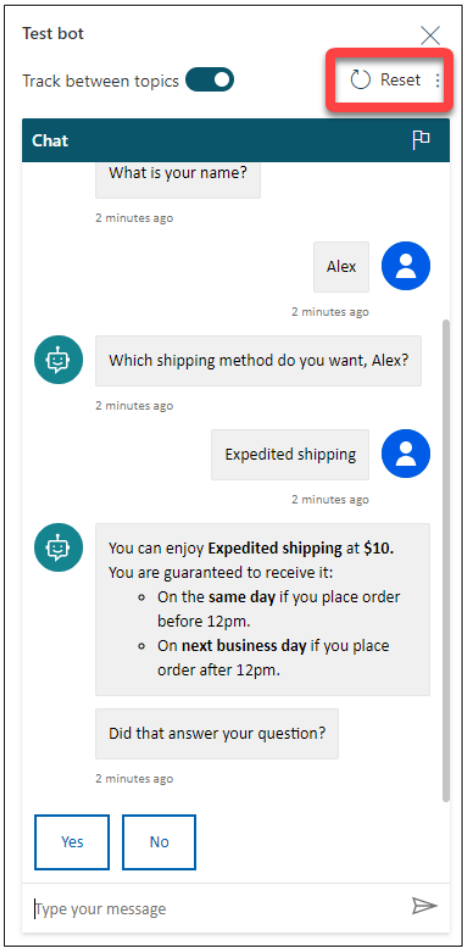


When the bot presents the shipping options, click **Expedited shipping**.

5. In the **Topic variable values** window, note that the value of the `ShippingMethod` variable is now shown. You will also see the value of the UserName bot variable.



6. Now, reset the conversation by clicking **Reset** at the top of the Test bot pane.

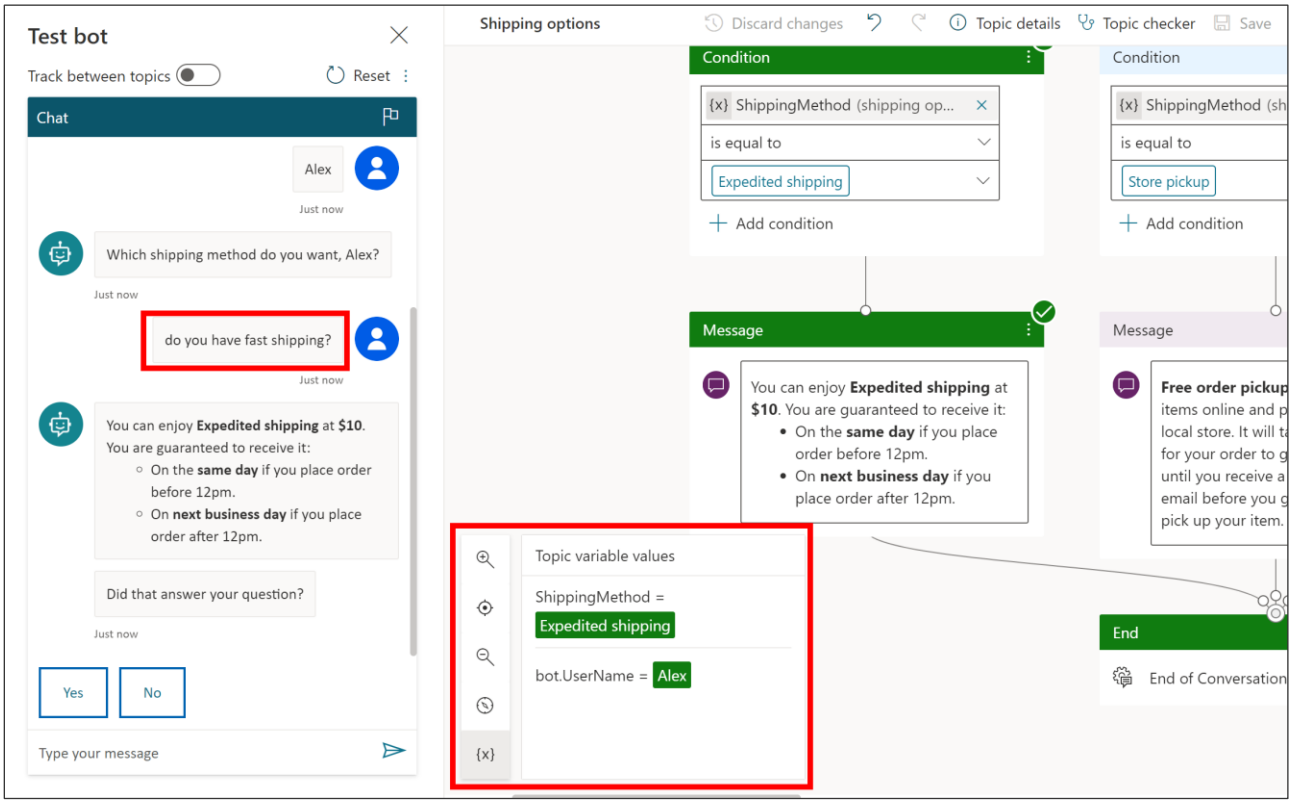


Note that the topic variable values are all blank again. This is a new conversation, so the variables have not been set (i.e., the slots have not been filled).

7. Type "**what are my shipping options?**" again.

8. This time, *instead of clicking the "Expedited shipping" button*, type one of the synonyms you created in Task 1 of this exercise (for example, "**fast shipping**").

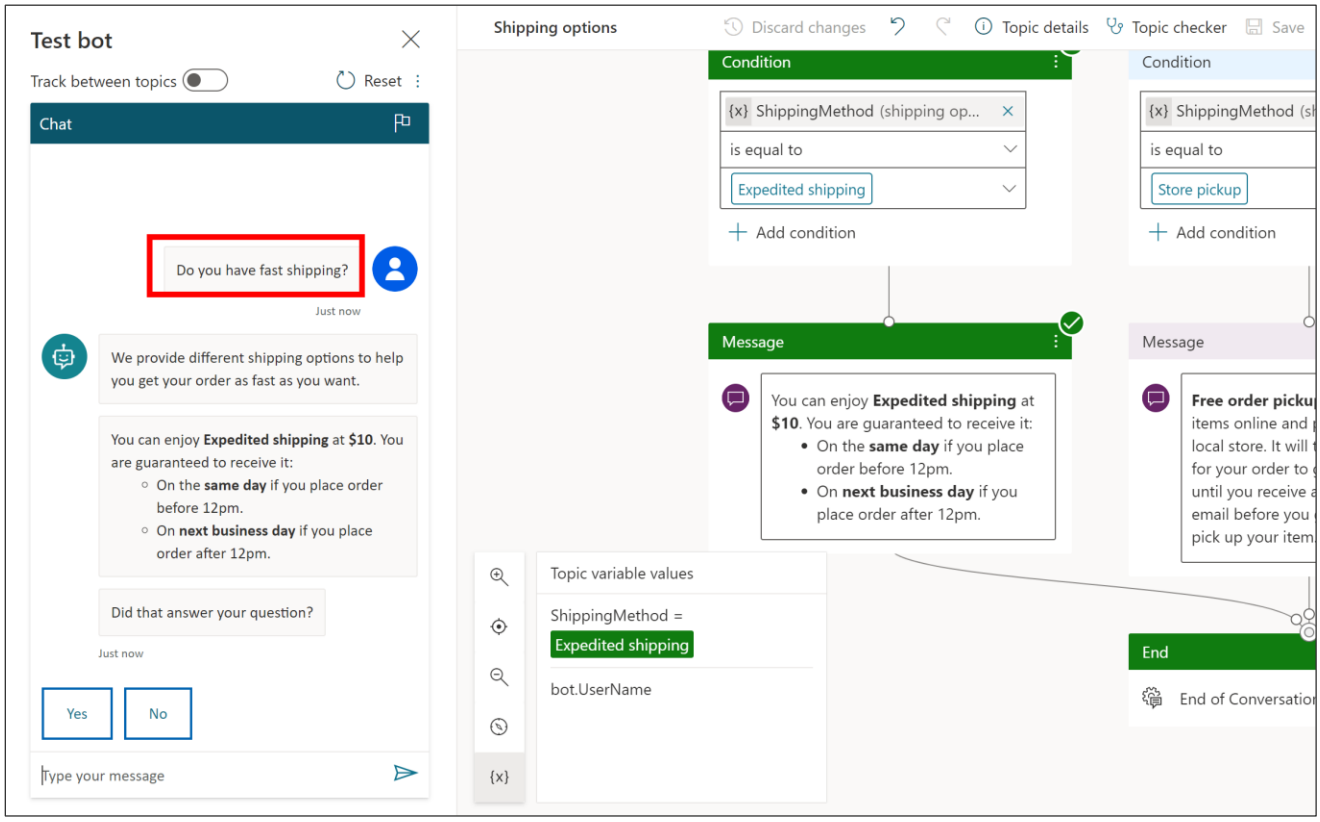
   See how entity extraction understands the **synonym**. You could even have typed "Do you have fast shipping?". (Go ahead and try it!)



   Note that the **Topic variable values** screen shows "expedited shipping"; it records the main entity as the value, not the synonym that helped it choose the right entity.

9. Let's reset the conversation one last time and see proactive slot filling in action. (Click **Reset**.)

10. Instead of asking about shipping methods, ask the bot "**Do you have fast shipping**?"



This time, instead of asking what shipping method you'd like, the bot already knows the answer and skips that question, taking you directly to the fast shipping information! This powerful feature allows your topics to more intelligently handle conversations, because it skips questions it already knows the answers to.

Lab survey

We would appreciate your feedback on Power Virtual Agents and on this hands-on-lab, such as the quality of documentation and the usefulness of the learning experience.
Please use the survey at https://aka.ms/PVAiaDSurvey to share your feedback.
You may provide feedback for each module as you complete it or at the end once you've completed all the modules.
Thank you!

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 POWER VIRTUAL AGENTS (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 POWER VIRTUAL AGENTS 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 POWER VIRTUAL AGENTS IS YOUR OWN PRODUCT OR SERVICE, SEPARATE AND APART FROM MICROSOFT POWER VIRTUAL AGENTS. 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 POWER VIRTUAL AGENTS IN CONNECTION WITH YOUR CHATBOT, INCLUDING (WITHOUT LIMITATION) ANY SUCH INJURIES TO END USERS.