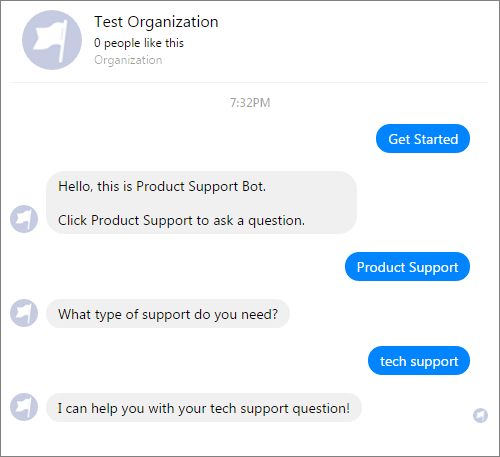
**Get Started with Flow XO**

Flow XO offers numerous features to build, host, and manage chatbots for social platforms. This tutorial will help you with some of the basics of Flow XO. You’ll learn how to ask your user a question, then respond depending on the answer. You’ll also learn how to test your chatbot in Facebook Messenger.

**If you don’t already have a free Flow XO account,**[**get yours now**](https://flowxo.com/app/signup)**.**

Let’s say you offer a prepaid phone service. Your customers usually have questions about billing or tech support. You can create a chatbot in Flow XO that will direct your customers to the type of product support they need.



**Create a Bot for Facebook Messenger**

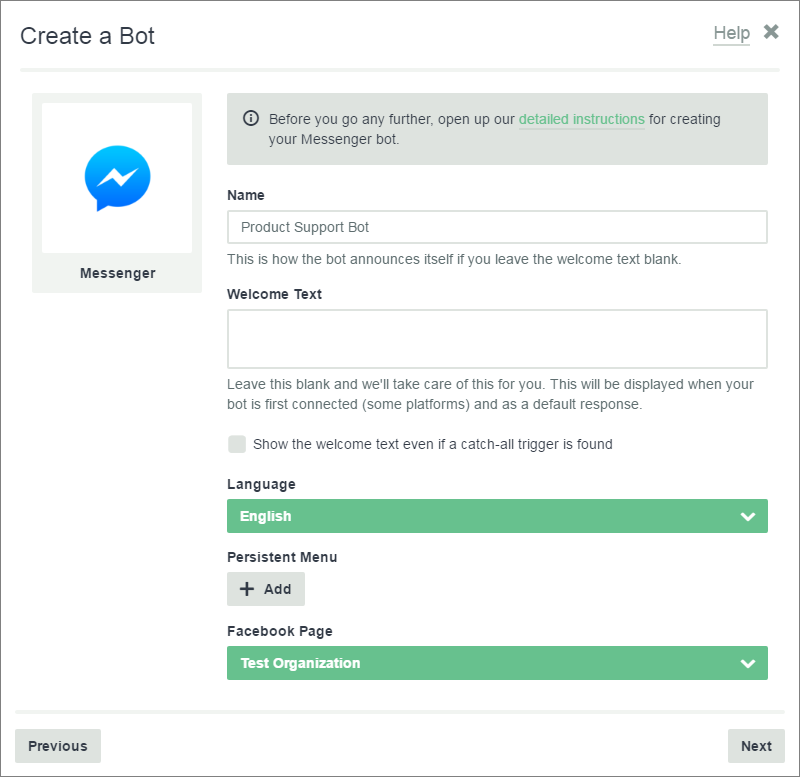
First, you’ll need to create a bot and add it to Facebook Messenger. In Flow XO, a bot is what users interact with. Later, you’ll create a flow that tells your bot how to respond to user input.

Before you create a bot:

1. Sign in to Facebook. If you don’t have an account, create one.
2. Create a Facebook Page. This Page is where users can access your chatbot.

Next, create a bot in Flow XO.

1. Sign into Flow XO. Open the editor to the **Bots** page.
2. Click **+ New Bot**, and select **Messenger** in the Choose a Platform window.
3. Give your bot a descriptive **Name**.
4. Leave the **Welcome Text**blank for now.
5. Click **Login to Facebook**, and let Flow XO view your Pages.
6. The Create a Bot window should now have a **Facebook Page** menu. Select your page in the drop-down list.
7. The window should look something like this. Click **Next**to continue.



Now, test your bot. Click **Message Us** in the Test & Distribute window. This will take you to Facebook Messenger where you can try your new chatbot.

At this point, you’ll see a greeting, and nothing else. To interact with users, your bot needs a flow.

**Create a New Flow**

In Flow XO, a flow is a message or action that your chatbot responds to. In this tutorial, you’ll create a flow that asks the user a question and responds depending on their answer.

You can create your own flow or install the shared [Product Support](https://flowxo.com/share/5nj6edr6) flow.

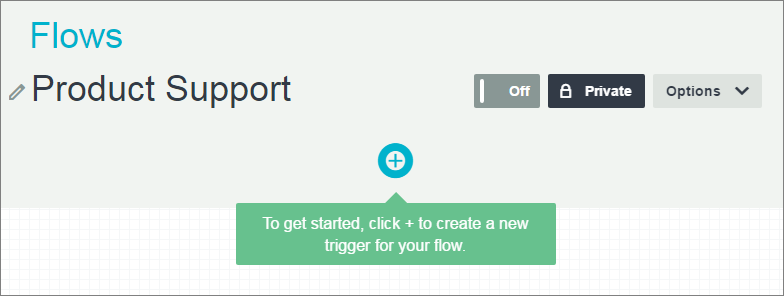
To set up a new flow:

1. Open the Flow XO editor to the **Flows** page.
2. Click **+ New Flow**.
3. Select **Blank Flow** in the Templates window.
4. Give your flow a descriptive title.

**Set Up a Trigger**

Every flow starts with a trigger. A trigger tells the chatbot what user input to listen for. Your flow will begin when the user’s input matches the trigger.

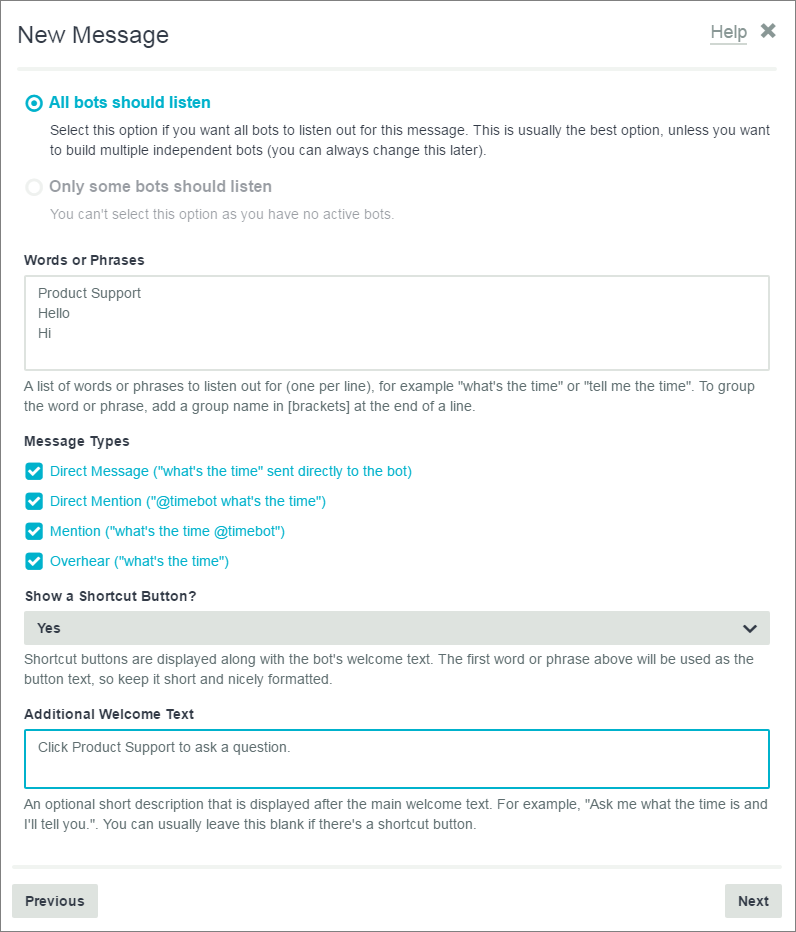
For example, a user could type, “I have a question about my bill”. If this matches the trigger, your bot will start the flow. Or, you could give the user a button to press that will trigger the flow.



To set up a trigger for your flow:

1. Click **+** to create a trigger.
2. Select **Bot** in the Choose a Service window.
3. Select **New Message** as your Trigger.

The New Message window displays options for your trigger. Here, you can tell your bot when to start the flow.



* **Which bots should listen?:**Select which active bots should listen for the trigger. For now, select **All bots should listen**.
* **Words or Phrases:** Enter some text that a user can type to trigger a flow. Add one word or phrase per line.
* **Message Types:** This tells your bot which messages to listen to.
* **Show a Shortcut Button?:** Create an optional button that will trigger the flow. For this tutorial, select **Yes**. The first line under **Words or Phrases**will be your button text.
* **Additional Welcome Text:**An optional field. Type some text to help users get started.

When you’re done with the New Message form:

1. Click **Next** at the bottom of the New Message window.
2. Click **Next**again in the Filter window. There’s no need to add a filter yet.
3. Give your trigger a descriptive **Name**.
4. Click **Save** to finish the trigger. You can edit it later if you need to.

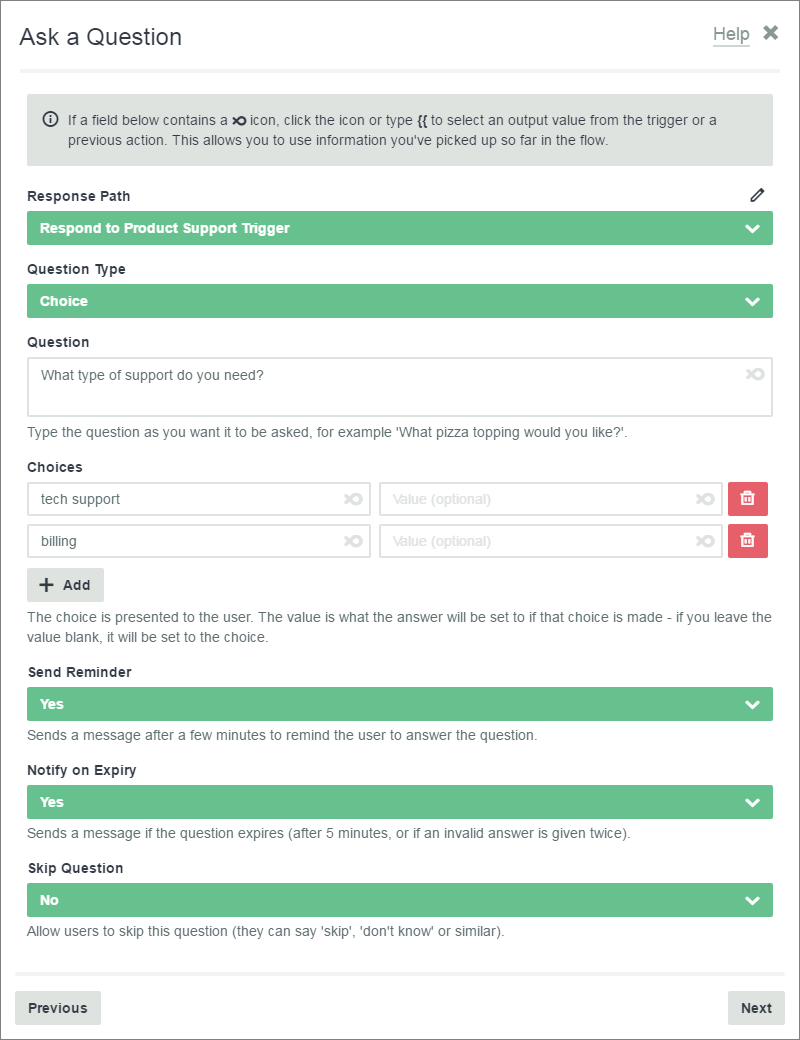
***Note:****It’s a good idea give every action in your flow a clear name so that you find it later. The name you choose will display on the Flows page. It will also help you when you need to refer back to it from other actions.*

**Ask a Question**

Now that you have initiated a flow, your bot can ask a question.

1. On the Flow page, click **+** to add a task.
2. Click **Bot** in the Choose a Service window.
3. For your action, select **Ask a Question**.

The Ask a Question window has several options:



* **Response Path:** Select your trigger. If you didn’t name your trigger, it will display as **New Message** in the drop-down.
* **Question Type:** The type of question you’re asking. For this tutorial, select **Choice** to give the user a list of options.
* **Question:**Ask the user a question.
* **Choices:** Enter choices that the user can select. Click **+ Add** to add more choices. Leave the Value field empty.
* **Set Reminder:** Set this to **Yes**.
* **Notify on Expiry:** Set this to **Yes**.
* **Skip Question:** Set this to **No**.

When you’re done with the Ask a Question form:

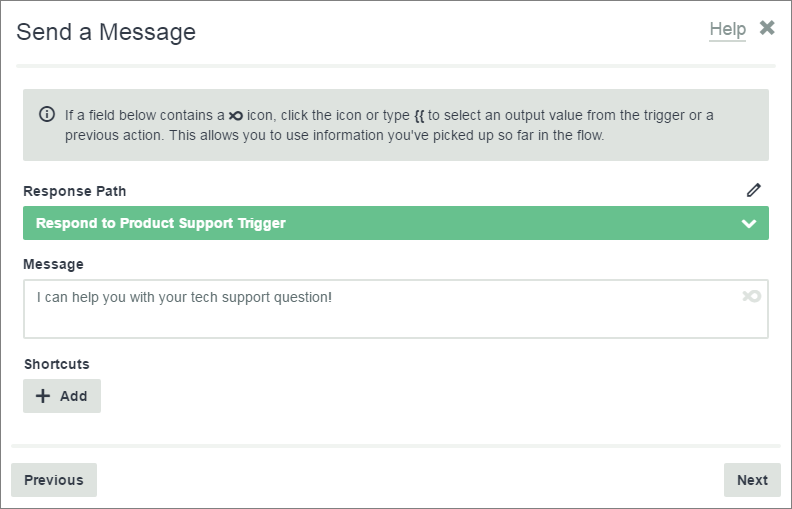
1. Click **Next**at the bottom of the Ask a Question window.
2. Click **Next**again in the Filter window.
3. Give your question a descriptive **Name**.
4. Click **Save**to finish the question.

**Respond to an Answer**

Next, respond to the user’s choice. Once again, set up your flow:

1. On the Flow page, click **+** to add a task.
2. Click **Bot** in the Choose a Service window.
3. For your action, select **Send a Message**.

Complete the Send a Message window and click **Next**.

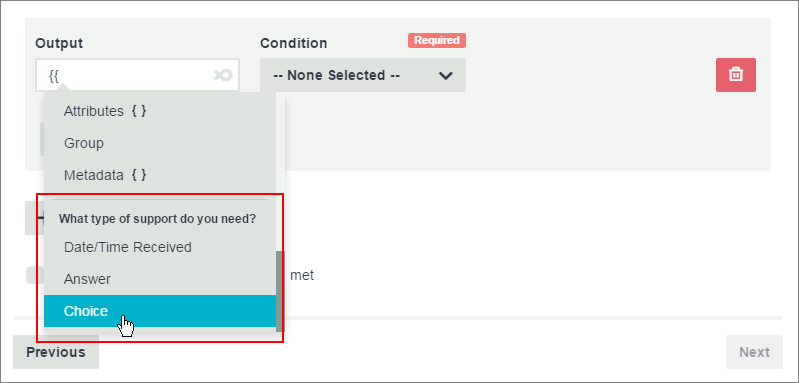


**Add a Filter**

Next, set up a Filter to tell your bot which choice will activate the response. In this case, we want “tech support” to activate the response, “I can help you with your tech support question!”

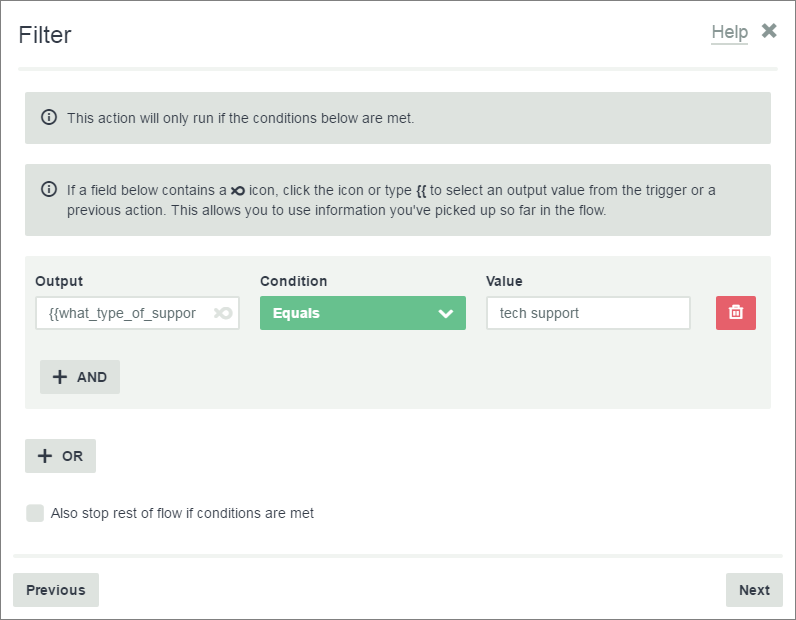
Click **+ Add a Filter** in the Filter window, and enter your conditions.

Tell your bot which message to look in and what conditions to check for:



1. In the **Output**field, type **{{** for a drop-down list. Find your question, and select **Choice**below it.This inserts the path to your question in the Output field.
2. Set **Condition** to **Equals**.
3. Set the **Value** to “tech support”.

This response will only display when the user selects “tech support”.



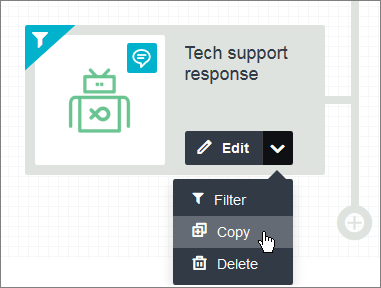
Click **Next** when your filter is done. This response will display in your flow.

**Copy a Response**

Now you need a similar response to the second option, in this case, “billing”. Instead of creating an entirely new message, you can copy the previous response and edit it. This is a quick way to build flows with several similar actions.

To copy the previous response:

1. In your flow, find the response to “tech support” that you just created.
2. Click the drop-down menu next to Edit, and select **Copy**. This will copy the entire message, including the filter.



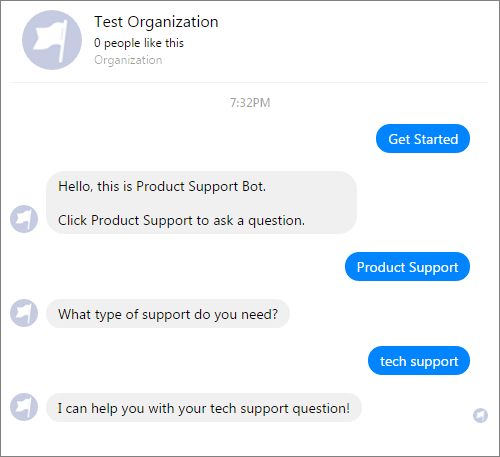
Now you can edit the flow. This second response should correspond to the second choice, “billing”. So, edit the Message and Filter accordingly.

You now have a functional flow. Set your new flow to **On**.

**Test Your Finished Bot**

Now that your flow is done, you can interact with it in Facebook Messenger. To do this:

1. Go to the **Bots**page in the Flow XO editor.
2. Click the bot you created earlier.
3. If you need to, click **Login to Facebook**, and select your Facebook Page. Click **Next**.
4. Click **Message Us** to test your bot. You should be able to interact with it.



Your basic chatbot is done. You can use more features in Flow XO to build on your bot. It can ask text questions, respond to variable answers, and even send the chat to a human. You can also put your flow on multiple platforms to answer user questions wherever they look for you.

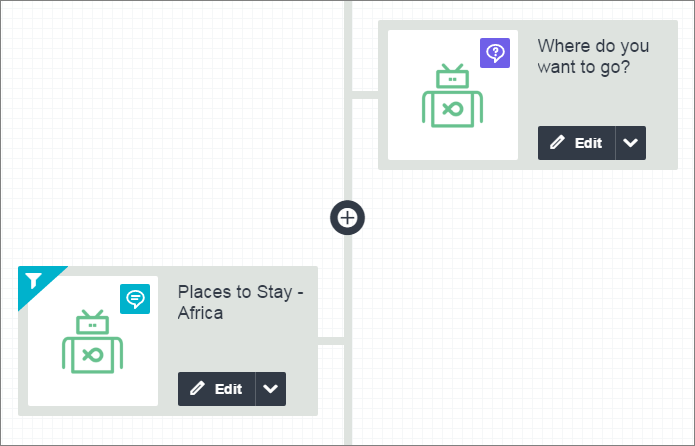
**Using Filters in Flow XO**

In Flow XO, filters are a great way to customize how your chatbot responds to user input. In this tutorial, you’ll learn what filters are and some practical ways to use them.

**If you don’t already have a free Flow XO account,**[**get yours now**](https://flowxo.com/app/signup)**.**

With filters, your chatbot can account for changeable information. You can respond to input with a tailored action.

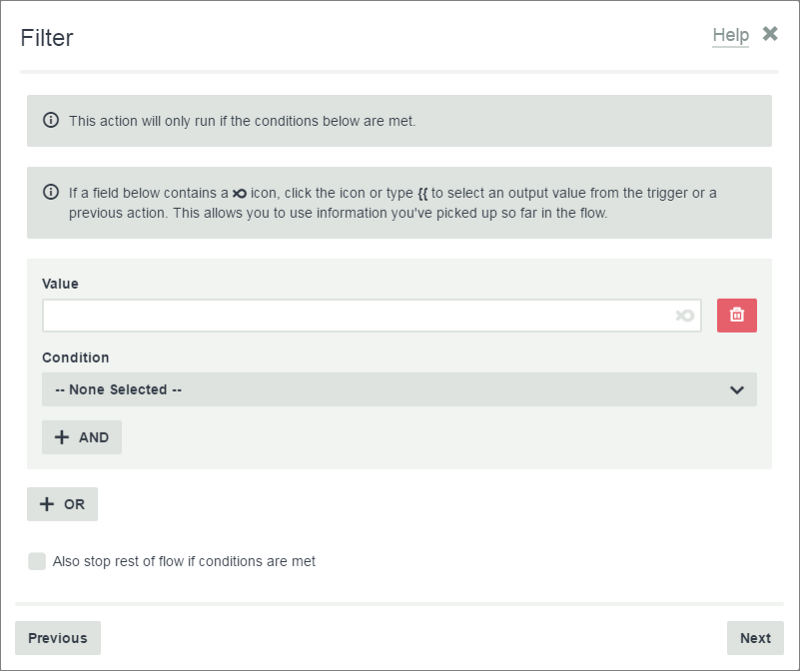
For instance, you might want to make a chatbot that helps users plan a trip. Your chatbot needs to gather information about where and how the user wants to travel. Then it will make recommendations based on this information.



**Filters**

You might want your trip planning chatbot to recommend a place to stay based on where the user is traveling. You can do this with filters:

1. In your flow, Ask a Question to find out where the user wants to go.
2. Below that, click **+**, and select **Bot** for your service. Then select **Send a Message**for your action**.**
3. In your message, respond to a choice the user might make. For instance, if the user wants to go to Africa, list recommendations for places to stay in Africa. Click **Next**to continue.
4. Click **+Filter,**and set up a filter to make sure your chatbot will only go to this response if the user answers “Africa”. The empty Filter looks like this:

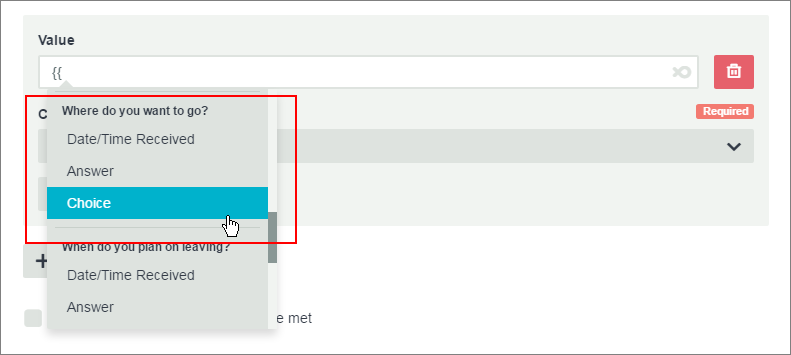


Your filter should look back to your original question, and check for the choice “Africa”. To tell your chatbot to check the answer to your question, the next step is to add a **Value**.

**Filter Values**

In the filter window, the **Value** is where you tell your chatbot where to look and what to look for. In this case, you want your chatbot to look in the question you just asked your user (“Where do you want to go?”). To set this up:

1. In the **Value** box, type **{{**, and find your question in the drop-down.
2. Select the type of user input to check for. In this case, we gave the user a list of choices, so check for a **Choice**.



Your filter should look something like this:



Now you need to tell your chatbot which choice to look for. To do this, you need to set a **Condition**.

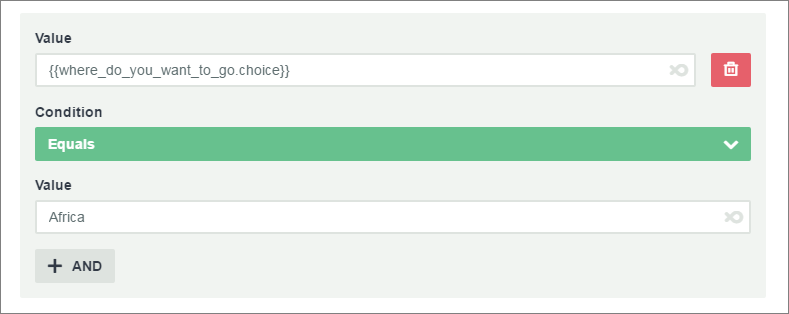
**Conditions**

Within a filter, a **Condition** tells your chatbot what to check for when it looks in a Value. Your chatbot will do a filtered action when the Value aligns with the Condition.

You’ve already told your chatbot to look for the answer to “Where do you want to go?” Now you want your response to display only if the user answers “Africa”.

In other words, if “Where do you want to go?” **Equals** “Africa” your chatbot should use this response. So, set your Condition to **Equals**.

The Condition **Equals** requires a second Value**.**Here is where you tell your chatbot what choice to look for, in this case “Africa”. Your filter will look like this:



This response will only display if the user answers your question with “Africa”. Click **Next,**then **Save** to finish the response.

**Filters with Multiple Conditions**

To make your responses even more useful, you can gather several pieces of data from the user. You can then use filters with multiple conditions to respond with what the user is looking for.

At this point, you’ve asked your user where they want to go. Now you want to make travel recommendations based on where they are coming from. To set this up:

1. In your flow, Ask a Question to find out where the user is traveling from.
2. Send a Message based on a possible answer from the user. For instance, the user might be traveling from Asia to Africa. When you’ve added your message, click **Next**.
3. Click **+ Filter**.

Now you need to create a filter that will only display this response if the user is traveling from Asia to Africa. Because you want to match the output from two questions, you need to use the **And**condition in your filter.

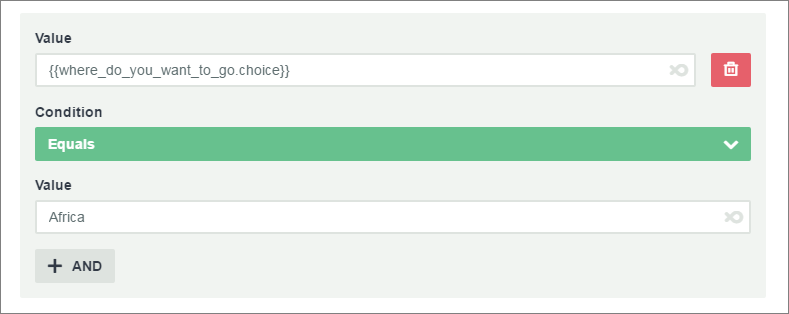
**Using the And Condition**

The **And**condition tells your bot to match everything inside a group. In the Filters window, a group is a set of values and conditions surrounded by a gray box.

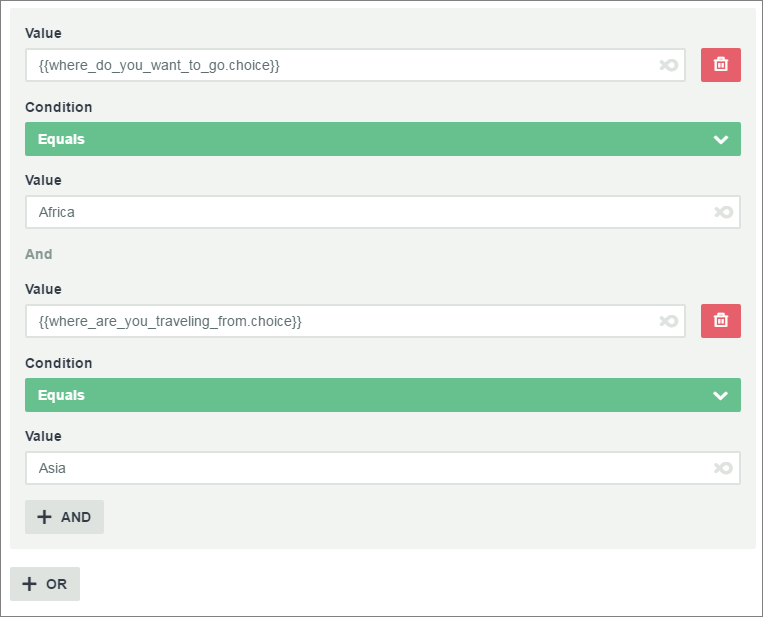
In this case, your filter should match *both*of these choices:

* Where do you want to go? = Africa, **And**
* Where are you traveling from? = Asia

Set the first **Value**and **Condition.**This will look the same as the filter you made earlier:



Now, click **+AND**. This will display an empty Value and Condition**.**Tell your bot to look in the question “Where are you traveling from?” and check if it equals “Asia”:



You can use add more And conditions if you need to.

***Note:****When using****And****, avoid using two Values that can’t be true at the same time. For instance, the answer to “Where are you traveling from?” can’t be “Asia” and “Australia” at the same time.*

**Using the Or Condition**

Use the **Or**condition when you want to respond in the same way to different user input.

For example, you might want to recommend the same mode of transportation if the user is going to Africa from Asia, or to Asia from Africa. In other words:

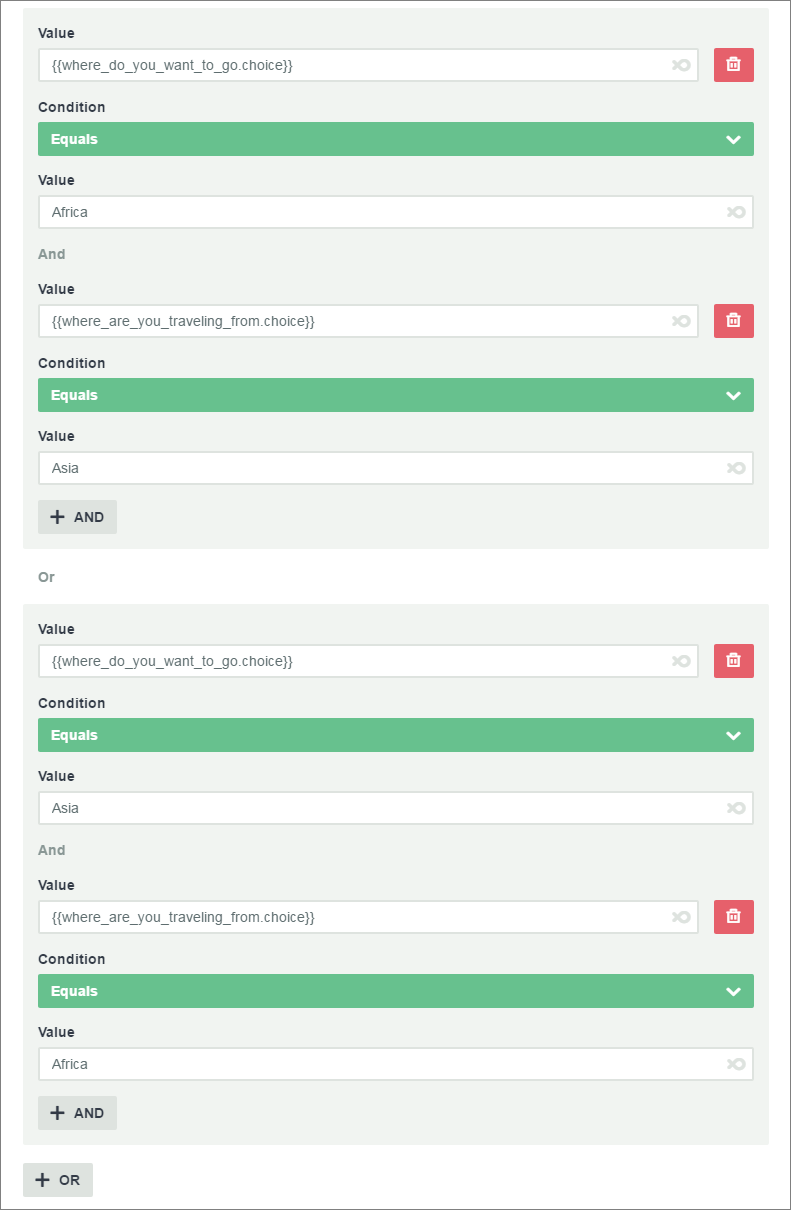
* Where do you want to go? = Africa, **And**
* Where are you traveling from? = Asia

**Or**

* Where do you want to go? = Asia, **And**
* Where are you traveling from? = Africa

Instead of creating a new message, you can reuse the existing message by adding an **Or** condition. Click **+ OR**, then add your values.

The whole filter should look like this. Note that each **Or**condition is in a separate gray box:



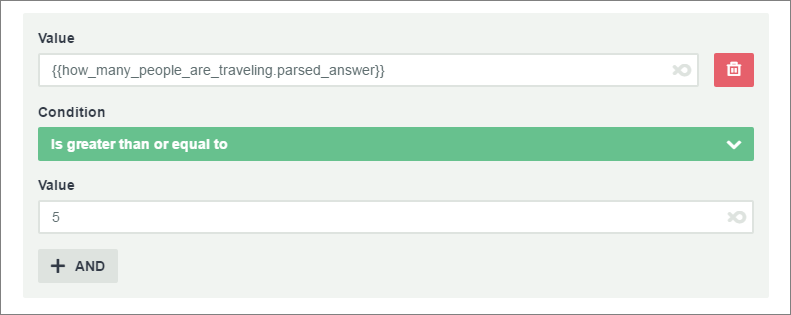
**Filtering Numbers**

Filters can account for nearly any kind of user input. It’s often useful to gather numbers from users. You can set you filter to check a numerical value.

For example, you might have a special rate for groups of 5 or more people. Find out how many people the user is traveling with, and offer the large group rate if it applies. To do this:

1. Ask a Question to find out how many people are traveling. Set **Validation**to **Integer**. This will tell your chatbot to expect a number from the user.
2. Create a response to offer the large group rate. Set your filter to check for a number greater than or equal to 5.

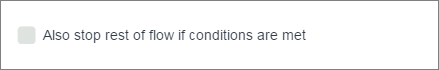
Your filter will look something like this:



If the user is traveling in a group of 5 or more people, your chatbot will offer the large group rate. If the group is smaller, the chatbot will move on to the next action.

**Stop the Rest of the Flow**

You may have noticed this checkbox at the bottom of the Filters page:



Check this box to tell your chatbot to stop the flow after it does the current action. This is useful if the action will meet the user’s needs, but your flow continues for users with other needs.

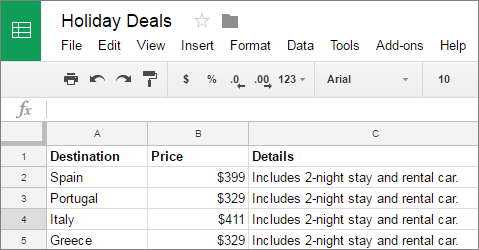
For instance, a user might indicate that they are not ready to make travel reservations. You can end the flow there, while continuing it for users who are ready.

**Use a Filter on a Trigger**

In the above examples, you placed each filter on an **action**within the flow. Your chatbot will only do the action if conditions are met.

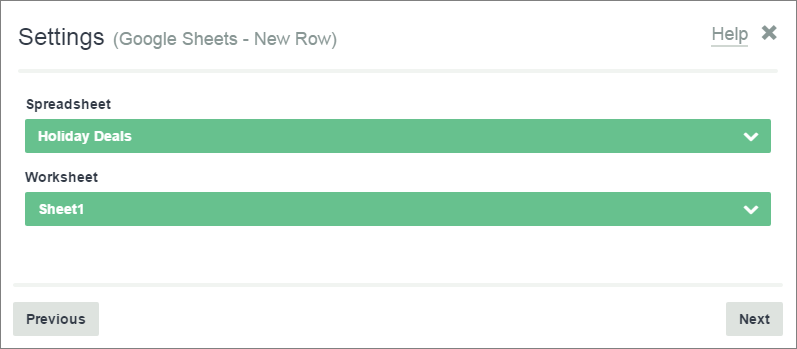
You can also place a filter on a **trigger**. The difference is that your chatbot will only start the *flow*if conditions are met. Also, instead of checking the output from previous actions, you can only filter output from the trigger itself.

You might have a Google Sheet with deals for certain destinations. You can set up a new flow that only triggers if your user sets the destination to “Spain”, for instance.



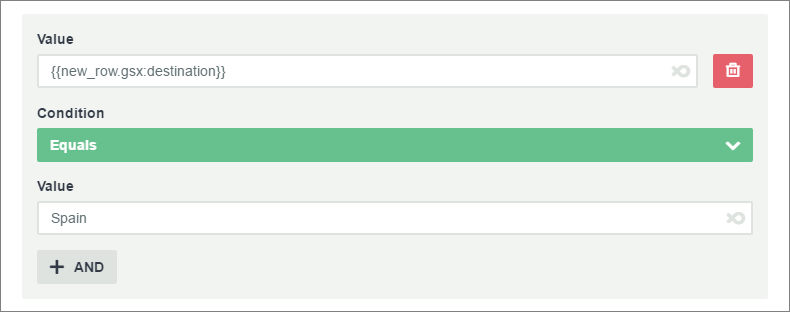
Follow these steps:

1. Set up your Google Sheet of holiday deals.
2. Go to the **Flows**page and click **+ New Flow**.
3. Set up a flow with a Google Sheets **New Row**trigger.



Now set up your filter:

1. On the Filterspage, click **+ Add a Filter**.
2. Make a filter that looks for the row “Spain” in your Google Sheet.



The whole flow will only trigger if the “destination” field in the Google Sheet contains “Spain”.

**Conclusion**

In this tutorial, you’ve learned how to use filters to look back to an action and check for specific values. You also learned how to use And/Or conditions, account for variable numbers, and use a filter on a trigger.

Flow XO offers even more ways to customize your responses. You can gather and filter data like date & time, contact information, age, and budget. Then you can set up your chatbot to book an entire trip!

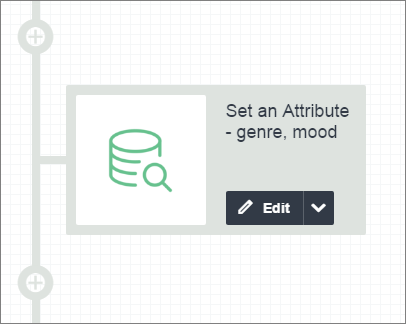
# Attributes in Flow XO

Attributes are a powerful feature in Flow XO. You can use attributes to store information, then access it later. Attributes are useful if you want your chatbot to do things like remember who the user is, complete an order, tabulate a response, or just give a personal greeting.

**If you don’t already have a free Flow XO account,**[**get yours now**](https://flowxo.com/app/signup)**.**

Attributes gather data, either actively, from direct user input, or passively, like counting how many time a user has accessed a flow. Your chatbot can use the same attributes across multiple flows.

For example, you might have a chatbot that recommends a song based on the user’s preferred genre and mood. When the user comes back for more, your chatbot will have this information on hand, and can make more recommendations. Or, the user can change their preferences if they want to hear something different.



### Attribute Basics

Each attribute in your flow has a name and a value. One way to use attributes is to set the name, and later ask the user for a value. Your chatbot will store this as a **name: value** pair. For example:

* **genre:**heavy metal
* **mood:**happy

In the above example, the attribute names in your flow are **genre** and **mood**. The user-input values are **heavy metal**, and **happy**.

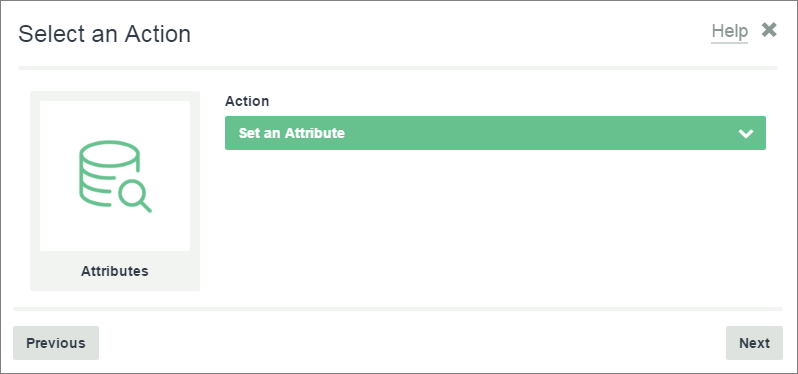
You can use this data in your flow to greet the user, then recommend a song.

### Set an Attribute

Use **Set an Attribute** to establish an attribute name and assign a value. The value is the output from an action in your flow. This can be either active or passive data from the user.

Here’s how to set this up in Flow XO:

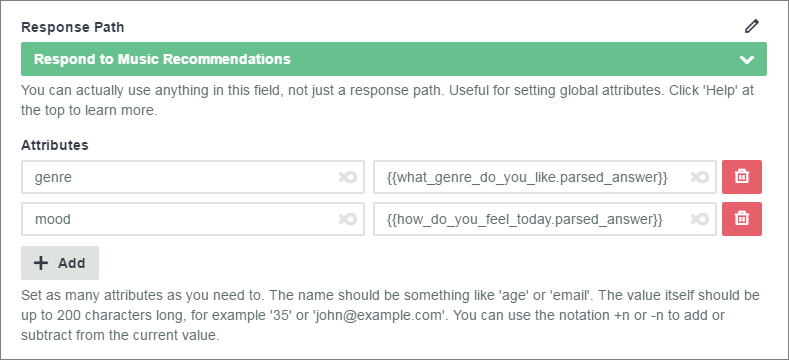
1. In your flow, ask for the user’s preferred genre, and mood. For each of these actions, select **Bot → Ask a Question**.
2. Now store each piece of data in an attribute. To do this, add a new action in your flow, and select **Attributes**in the **Choose a Service** window.
3. Select **Set an Attribute**for your action and click **Next**.



Now add your attributes in the **Settings** window.

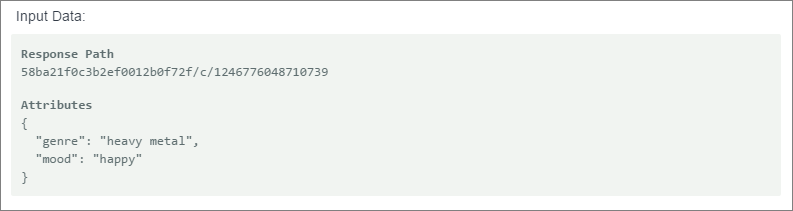
1. Click **+ Add**to add a new attribute.
2. In the **Name** field, enter the name you want to give to the attribute, like “**genre**”.
3. In the **Value** field, enter the output that you want to use as the attribute value. For example, the value for **genre** should be the answer to your question, “What genre do you like?” So, the Value field might look like “{{what\_genre\_do\_you\_like.parsed\_answer}}”.
4. Follow the same steps to set the attribute for **mood**.

The **Attributes**list should look something like this:



Click **Next**to continue.

When the user runs your chatbot, it will log the attribute values at this point in the flow. Your chatbot logs input data that is sent from the user to the bot. The log for attributes looks like this:



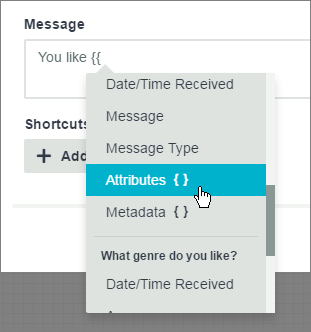
After you set an attribute, you can retrieve the information to use it in your flow.

### Use an Attribute in a Flow

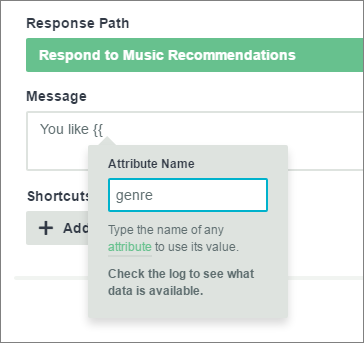
Once you have set an attribute, you can use it anywhere in your flow. You chatbot will use the attribute values logged in the input data .

At this point, we have stored the genre and mood. If you want use these values in a message, you can set up an action like this:

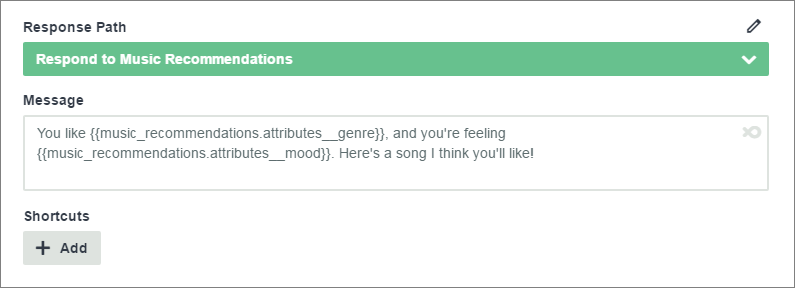
1. In your flow, add a **Bot**to **Send a Message**.
2. Start typing your message.
3. In the place you want to display the genre, type **{{**. Then select **Attributes { }** in the drop-down.



Type the exact attribute name that you want to retrieve and press **Enter.**In this case, type “**genre**”, to mach the attribute that you created earlier:



Follow the same process to retrieve the **mood**attribute. The completed message will look something like this:



Your chatbot will replace “{{music\_recommendations.attributes\_\_genre}}” with the value stored in the **genre**attribute. It will do the same for **mood**.

So, if your user enters “heavy metal” and “happy”, your chatbot will send the message:

“You like heavy metal and you’re feeling happy. Here’s a song I think you’ll like!”

### Update an Attribute

To update an attribute with a new value, use **Set Attribute** again.

A user may want to change the genre or mood the next time they visit your chatbot. If so, you can **Set an Attribute**again to update those values. The new value will overwrite the previous value.

* Make sure to match the name of the attribute you are overwriting. For instance, to overwrite the value for **mood**, you must use the same attribute name, **mood**.
* To completely reset an attribute, leave the Value empty.

### Using Filters with Attributes

Another good way to use attributes is to filter an action in your flow based on an attribute value.

For example, you can have your chatbot send one message if the user is happy, but a different message if the user is sad. You can do this with a filter that checks the **mood**attribute you set earlier.

First, create a message that your chatbot will only send if the user’s **mood**is **happy**.

1. Add an action in your flow. Use a **Bot** to **Send a Message**.
2. Type your message for a happy user and click **Next**.
3. On the Filters page, click **+ Add a Filter**.
4. Set the **Value**to your **mood**attribute.
5. Set the **Condition** to **Equals**.
6. Set the second **Value** to “happy”. You filter will look something like this:



This action will only run if the **mood**attribute has the value **happy**. To send a different message to a sad user, follow the same steps, but check for the value **sad**instead.

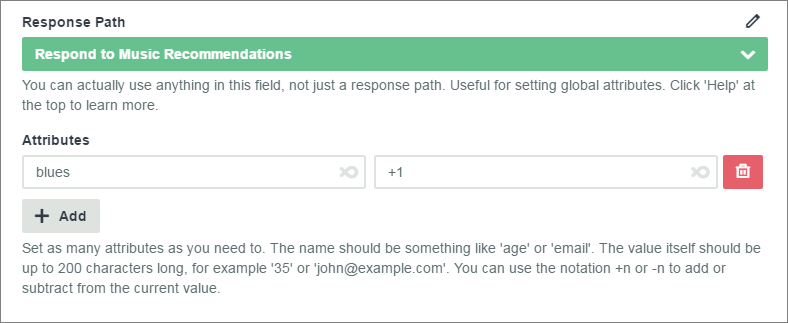
### Increment an Attribute

You can increment the value of numeric attributes with +n or -n. This is useful for tracking patterns, like how many times a user has visited your chatbot or given a certain response.

Say the user also likes blues music. If your user has played one blues song, you’ll want to give them a different song the next time they pick the genre.

One way to change the song is by tracking how many times the user triggers your blues flow. Here’s how you can do this with attributes:

1. Set up a new flow for the blues genre.
2. Immediately after the trigger, add a new action and **Set an Attribute.**
3. Give the attribute a logical **Name**, like **blues.**
4. Set the **Value** to **+1.**Each time the user triggers this flow, the value for the **blues**attribute will increment by 1. By default, the starting value is 0.



Now that you’ve set up your incremental attribute, you need to tell your chatbot to log the value. To do this, use **Get an Attribute**.

#### Get an Attribute

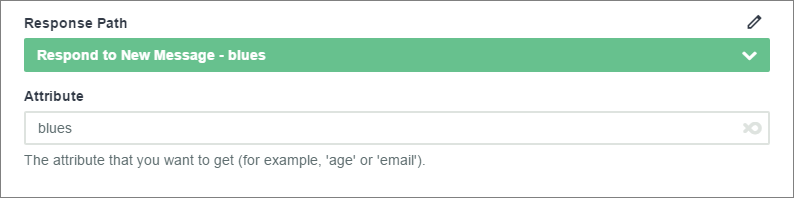
Use **Get an Attribute** after you set an incremental attribute. This feature logs the new attribute value each time the counter goes up. It prevents your counter from resetting to 0.

***Note:****It’s usually unnecessary to use****Get an Attribute****because attributes are included in the data you receive when a message comes in to your bot.*

*You may still need to Get an Attribute if the attribute value may have changed during the flow (like when you increment a value).*

To get an attribute:

1. Add a new action in your flow.
2. Select **Attributes → Get an Attribute**, and click **Next**.
3. Type the exact attribute name that you want to log. In this case, **blues**:



Each time the user runs this flow, the value for **blues** will increment by one. Your chatbot will log each instance. So, if the user triggers this flow three times, the Value in your log will be “3”:



You can use this data in several ways. For example, you can search your external database, cycle songs listed in Google Sheets, or you can trigger new actions in your flow.

### Conclusion

In Flow XO, attributes are an effective way to store and retrieve information. In this tutorial, you learned what attributes are, how to use them in your flow, and how attribute values persist across flows.

There are endless ways to use attributes in Flow XO. With attributes, users can change their responses so that you can provide service that meets changing requirements. You can track the frequency of a response and adjust output accordingly. You can also store attribute data outside your flow, like in Google Sheets.

# Using Google Sheets in Flow XO

In Flow XO, your bot can read and edit spreadsheet data in Google Sheets. In this tutorial, you’ll get an overview of Google Sheets. You’ll also learn how to use a bot to search for information in a spreadsheet, then update it if necessary.

**If you don’t already have a free Flow XO account,**[**get yours now**](https://flowxo.com/app/signup)**.**

For example, say you own a bakery, and periodically email your customers with news and deals. In Flow XO, Google Sheets is a good way to collect contact information. When a customer interacts with your bot for the first time, you can get their email address. Later, you might want to verify and update the email if necessary.

### About Google Sheets

Google Sheets is a free spreadsheet tool that you can access here: <https://drive.google.com/>. If you don’t have a Google account, it’s free to sign up for one.

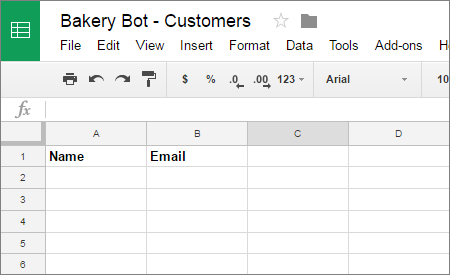
When you have your account, go to Google Drive and create a new spreadsheet. Click **New → Google Sheets**.

Later, you’ll authorize your Google Sheets account in Flow XO. Your bot can then access spreadsheets that you create in Google Sheets.

### Google Sheets Setup

It’s important to set up any spreadsheets you want to use before you create your flow. Your bot will use the header row and other content as a reference point when it interacts with your spreadsheet.

In this example, you want to gather customer contact information in Google Sheets. Create a new spreadsheet, then give it a clear title and a header row:

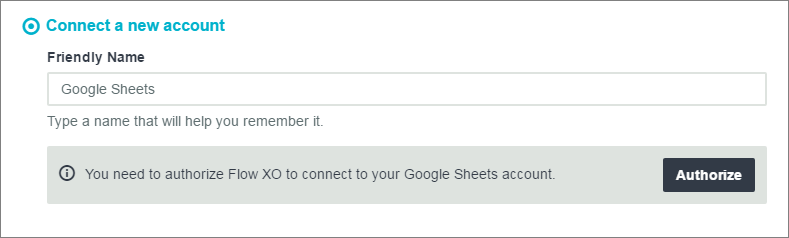


The header row is the first row of the spreadsheet. It specifies the data that will go in each column. In this case, we want to collect each customer’s **Name**and **Email**.

#### Authorize Google Sheets

When you use Google Sheets in your flow for the first time, you’ll need to authorize your account. To do this:

1. Sign in to your Google account.
2. In your flow, add an action and select **Google Sheets → Search Rows.**Then click **Next**.
3. In the **Authorize the Service** window, select **Connect a new account**. Then click **Authorize**.
4. When the Google Accounts window opens, choose the account you want to authorize.



The next time you use Google Sheets in a flow, Flow XO will connect to your authorized account.

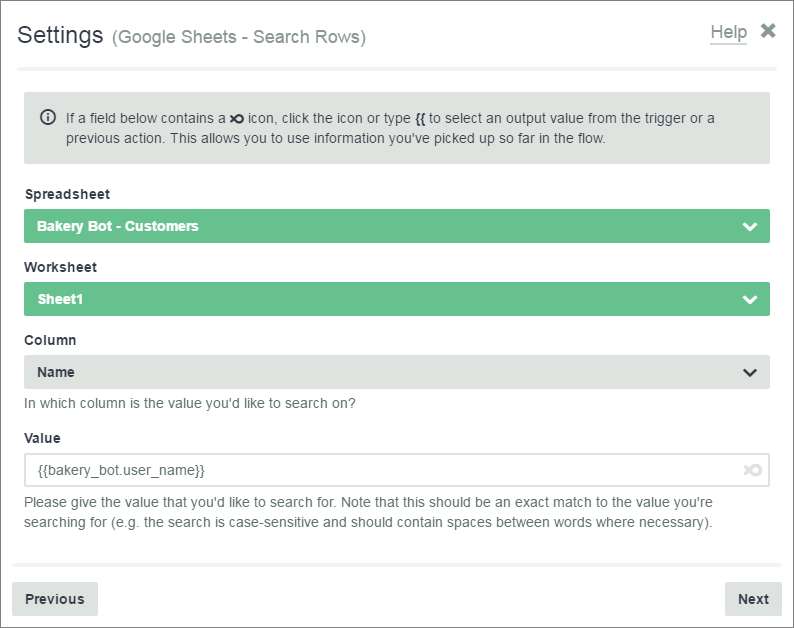
### Search Rows

After you set up your spreadsheet, you can interact with it in your flow. One useful interaction is to **Search Rows** for data.

In this tutorial, you’ve created a spreadsheet to collect customer names and email addresses for your bakery mailing list. Before you add a customer to the spreadsheet, it’s a good idea to check if they’re already on the list. This will help you avoid redundant data.

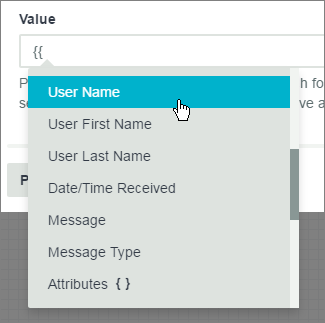
If you followed the steps above, under **Authorize Google Sheets**, you should have the start of a **Search Rows** action.

From the **Authorize the Service**window, click **Next**, then complete the **Settings**window. This window tells your bot what spreadsheet to search in and what value to search for:



* **Spreadsheet:**Select the spreadsheet you want your bot to search.
* **Worksheet:**Select the worksheet you want to use. If your spreadsheet has several worksheets, they will all be listed here. In this case, there’s only one worksheet, so select **Sheet1**.
* **Column:**This field takes content from the header row in your spreadsheet. Select the header for the column that you want to search. For this example, tell your bot to search the **Name** column to see if the customer name is already on the list.
* **Value:**Enter the value you’re searching for. This can be a plain text value, or you can use an output from your flow. Here, the value {{bakery\_bot.user\_name}} tells the bot to search for text that matches the user name.

***Note:****Your bot will capture the user name for you on most platforms. To access it, type****{{****, then select****User Name:***



Click **Next**to finish and save your action. The next step is to create another action that depends on whether your bot finds a match.

#### Filtering Search Results

Once you have your search results, you can use a filter to tell your bot what to do next.

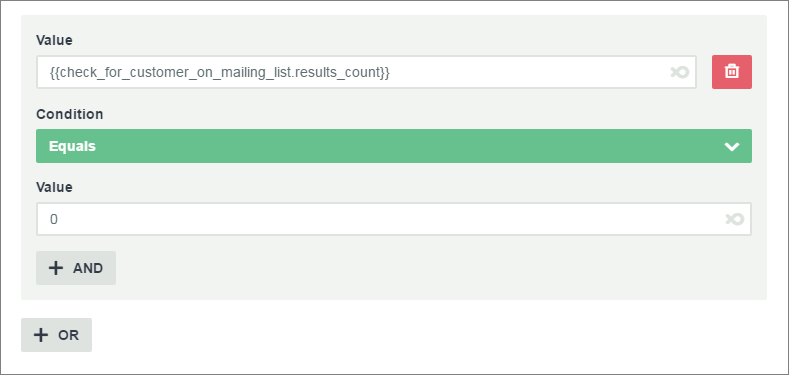
For your mailing list, you might want to do two different actions, depending on search results:

* If the user is not on the the mailing list, ask for an email address.
* If the user is on the mailing list, verify that the email address on record is correct.

First, set up an action for users who are not on the mailing list.

1. In your flow, add an action. Select **Bot → Ask a Question**.
2. Type a Question to ask for the user’s email address. Make sure to use the **Email Address** validation. Then click **Next**.
3. In the Filter window, create a filter to check if there are matching results from your **Search Rows**action.

For the filter, you want your bot to check the results count. Then, you only want it to do the action if there are no results. Your filter should look something like this:



This action will only run if there are no matching results from your **Search Rows** action.

For the second action, you’ll create a similar filter. But this time, you’ll only want it to run if there is a match. In other words, if the results count is **1**, then do the second action. We’ll cover the second action later in this tutorial.

### Add a Row

In Flow XO, your bot can add new data to Google Sheets. You can do this with the **Add a Row**action.

At this point, you’ve asked the user for their email address. Now you want to take that data and add it to your spreadsheet. To set this up:

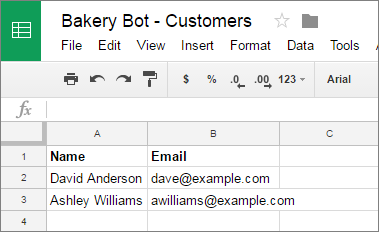
1. In your flow, add an action and select **Google Sheets →** **Add a Row.** Then click **Next**.
2. At this point, you’ve authorized your Google account, so select **Choose a connected account**. Click **Next.**
3. In the **Settings**window, tell your bot what to add to the spreadsheet.

Your Settings window should look like this. The **Name**and **Email**fields correspond to the columns in your spreadsheet:



* **Spreadsheet:**Select the spreadsheet you are adding a row to.
* **Worksheet:**Select the worksheet to edit.
* **Name:**The value in this field will go in the **Name**column of your spreadsheet. In this example, the bot will grab the user name.
* **Email:**The value in this field will go in the **Email** column of your spreadsheet. Here we’ve told the bot to take the output from our last question, “What is your email address?” and add it to the spreadsheet.

When a new user enters an email address, your bot will add it to the spreadsheet:



### Update a Row

Another useful feature of Flow XO is to update a row in your spreadsheet. If the user changes information, you can tell your bot to update it in Google Sheets.

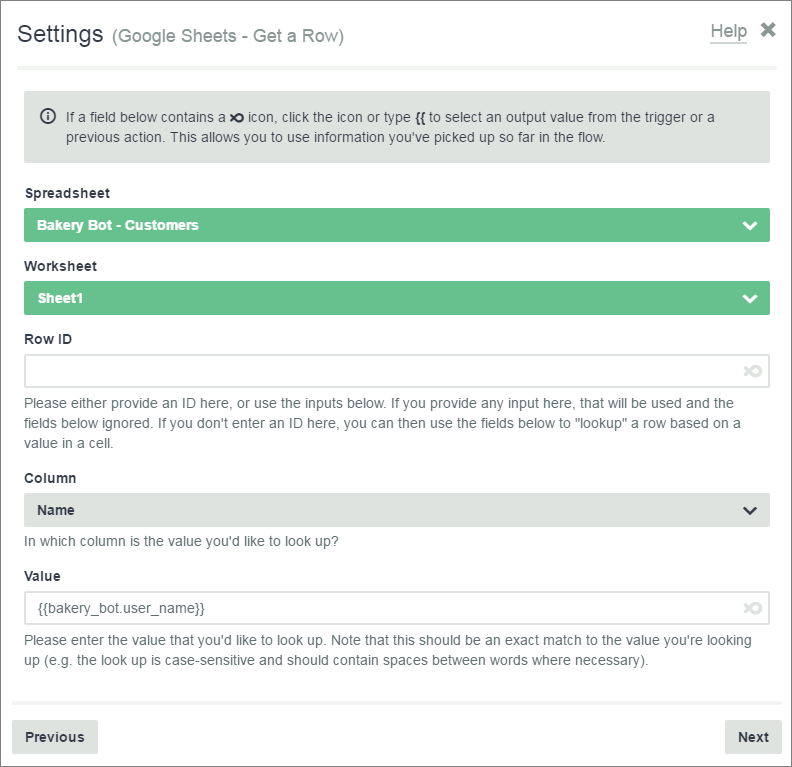
At this point, you’ve handled new users who aren’t on the mailing list yet. For returning users, you might want to verify that their email is correct.

To do this, there are a few steps you need to follow in sequence.

#### 1. Get a Row

Use **Get a Row**to tell your bot what spreadsheet row to read. Later you will use the output from this action to tell your bot what to update. To get a row:

1. In your flow, add a new action. Chose **Google Sheets → Get a Row**, and click **Next.**
2. Connect to your authorized account, and click **Next**.
3. In the **Settings**window, tell your bot what row to get.



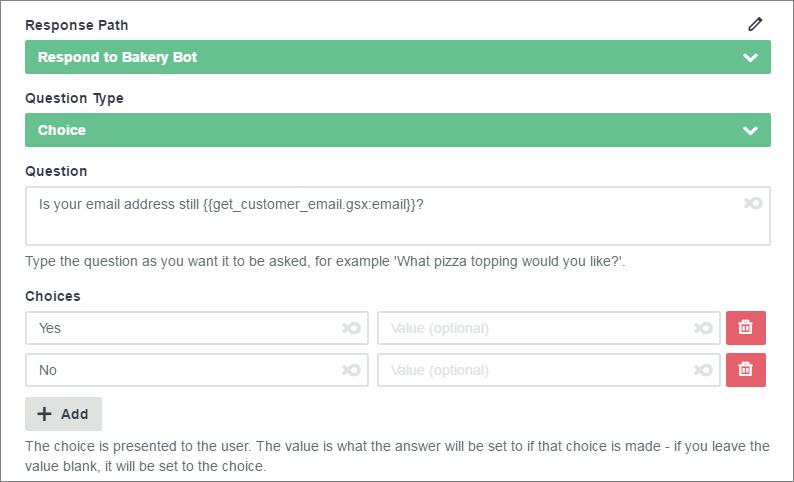
* **Spreadsheet:**Select the spreadsheet you want your bot to look in.
* **Worksheet:**Select the worksheeet that contains the row you want to update.
* **Row ID:**You can use this field if you want your bot to get an absolute row (Row 3, for example). But, in this case, you want to your bot to get a row that will vary depending on who your bot is chatting with. So, leave this field blank. Use the other fields below it.
* **Column:**Select the column you want your bot to look in.
* **Value:**Enter the value you’re searching for. In this case, the bot will search for text that matches the user name.

You bot will log all data in the row that matches the user name. In this example, it will log the user name and email address. You can now use and update this data in your flow.

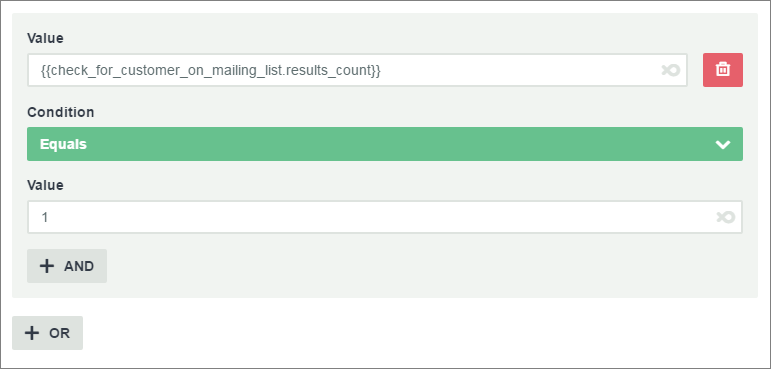
#### 2. Ask a Question

The next step is to ask the user if the email address you have on record is correct.

1. In your flow, add an action. Select **Bot → Ask a Question**, and click **Next.**
2. In the **Question** field, ask the user to verify the email address on record. Use the output from the **Get a Row**action, and tell your bot to show the data that is in the **Email** column:



Now, add a filter so that this action will only run if the **Search Rows**action you set up earlier returns a match:

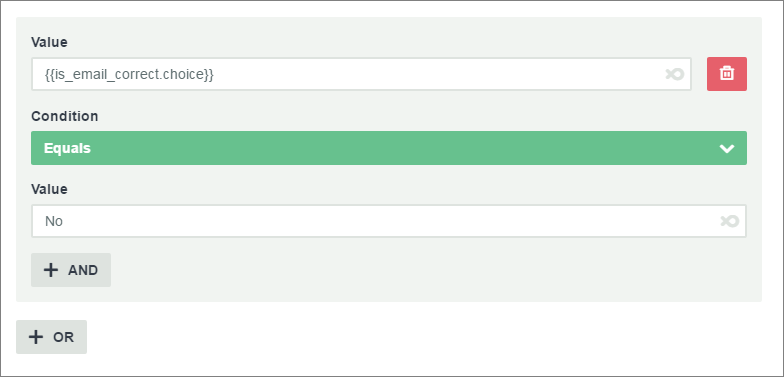


The user can signal whether the email address you have is correct.

#### 3. Ask for the Correct Email

If the user signals that the email address is incorrect, you need to ask for the correct email:

1. In your flow, add an action. Select **Bot → Ask a Question**, and click **Next.**
2. Ask your user to enter the correct email address. Remember to set **Validation**to **Email Address**. Click **Next**.
3. Create a filter that will only run if the user answers “No” to the previous question. Then click **Next**and save the action:

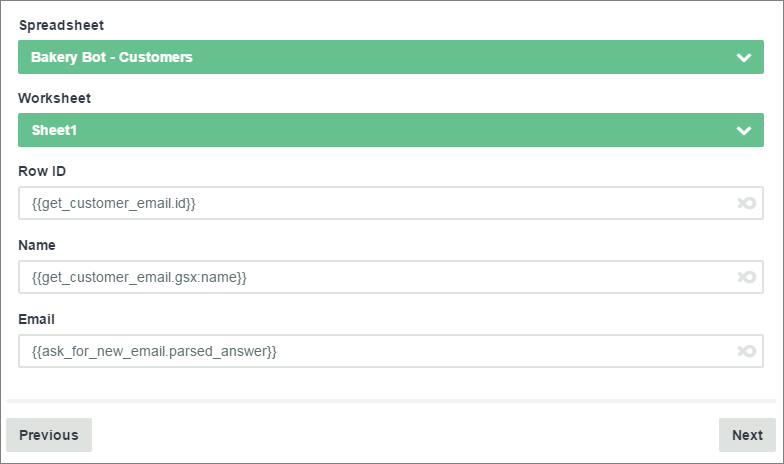


Now you have the correct email address. You’ll use the output from this action in the last step.

#### 4. Update a Row

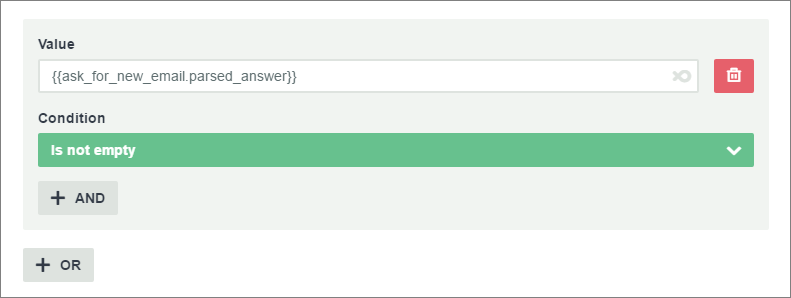
Finally, your bot can update a row with the new email address.

1. In your flow, add an action. Select **Google Sheets → Update a Row.**
2. In the Settings window, tell your bot to look the row that you logged with the **Get a Row** action. Then specify what data to put in each column:



* **Spreadsheet and Worksheet:**Select the same spreadsheet and worksheet you used for the **Get a Row** action.
* **Row ID:**Tell your bot to use the same Row ID as the row that you logged with the **Get a Row**action.
* **Name:**Keep the original value in the **Name**column. Use the value that you logged with the **Get a Row**action.
* **Email:**Update the **Email**column with the output from your last question. This will update the spreadsheet with the new email address.

Add a filter to this action as well. It should only run if the user answers your request for a new email address:



Now your bot can verify and update an email address.

### Conclusion

In this tutorial, you learned how to use Flow XO with Google Sheets. We covered the basics of reading and editing a spreadsheet, as well as using spreadsheet data in your flow. You can use Google Sheets with even more Flow XO features, like cards, attributes, and flow triggers.

# Using Google Sheets in Flow XO, Part 2

In [Part 1](https://medium.com/flowxo/using-google-sheets-in-flow-xo-4535d904514f) of this tutorial, we covered how to use Flow XO to search, add, and update information in Google Sheets. This tutorial will show you how you can use Google Sheets with even more features in Flow XO.

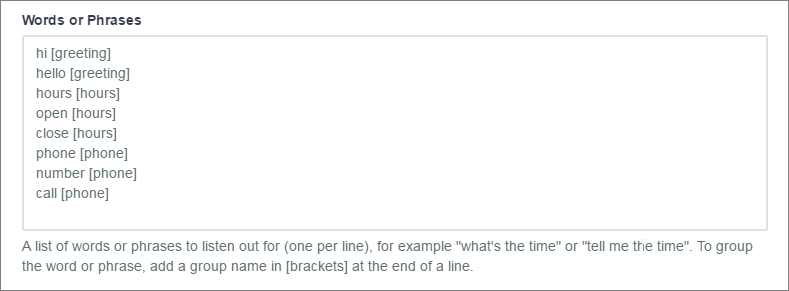
**If you don’t already have a free Flow XO account,**[**get yours now**](https://flowxo.com/app/signup)**.**

In the last tutorial, you created a mailing list bot for a bakery. You set up a flow to collect and verify contact information from your customers. Now you can build on your bot. In this tutorial, you’ll learn a time-saving way to use Google Sheets when you respond to grouped words or phrases. You’ll also learn how to use Google Sheets to make a card set that will be quick to update later.

### Respond to Grouped Words or Phrases

In Flow XO, you can use **groups**to categorize words or phrases you might expect from users. You can then account for variable wording with a single response.

For example, this **New Message** trigger accounts for different ways users might send a greeting, request hours, or ask for a phone number. The bracketed word next to each message is the group:



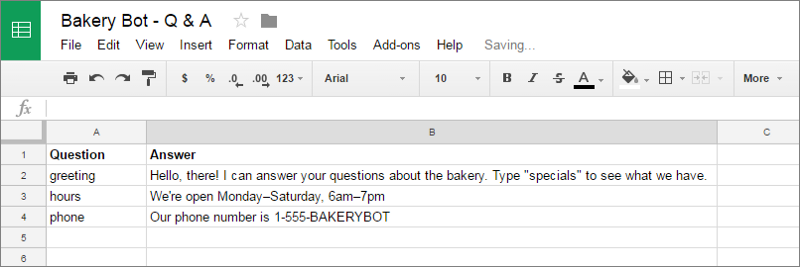
With Google Sheets, you can set up a spreadsheet that contains a response to each group. This can save you time when you set up your flow. Instead of creating a new action for every response, you can create just two actions: one to search your spreadsheet and one to send the response.

#### Set Up Responses in Google Sheets

Remember, always set up your spreadsheet before you use it in your flow. Flow XO needs an existing spreadsheet with a header row as a reference point.

Follow these steps to create a spreadsheet that contains a response to each group in your trigger.

1. In Google Sheets, create a new spreadsheet and give it a clear name.
2. Give your spreadsheet a header row that labels the contents of each column. In this case, label the columns **Question**and**Answer**.
3. In the **Question**column, list the exact group names that you’ll use in the trigger for your flow.
4. In the **Answer** column, type your response to each group. Your spreadsheet should look like this:



All of your responses are now in one place. Later, if you decide to add more groups and responses, you only need to update the trigger and the spreadsheet. Because of what you’ll do in the next steps, any new responses won’t require new actions in your flow.

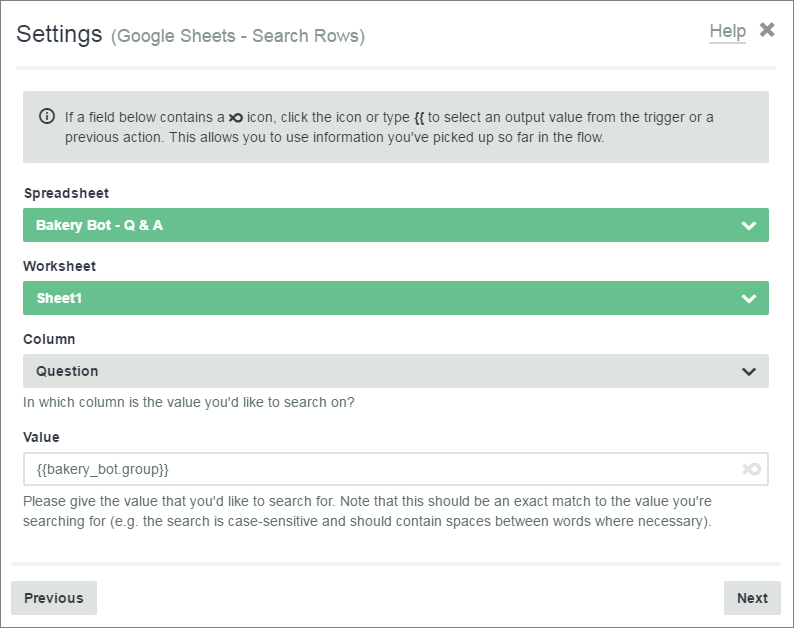
#### Search Rows for a Response

By now, you should have a trigger with grouped words or phrases and a spreadsheet with corresponding questions and answers. Now you’ll want your bot to respond to the user with an answer from your spreadsheet.

First, you need your bot to search your spreadsheet for the group that corresponds to the user’s input. For instance, “hello” is in the [greeting] group. If the user says “hello”, your bot should search for “greeting” in your spreadsheet. Here’s how you can do this:

1. In your flow, add an action. Select **Google Sheets → Search Rows**, then click **Next**.
2. Your Google Sheets account should already be authorized (we covered this in the [last tutorial](https://medium.com/flowxo/using-google-sheets-in-flow-xo-4535d904514f)). Click **Next**.
3. In the **Settings**window, select the **Spreadsheet**and **Worksheet**you want your bot to search.
4. Select the **Column** you want your bot to search. Remember that the options in this menu match the header row of your spreadsheet. Choose the column that contains your group names, in this case, the **Question**column.
5. In the **Value**field, tell your bot what to look for. You want the bot to take the group from the trigger output and find it in your spreadsheet. In the Value field, type **{{**, find your trigger in the drop-down, and select **Group**below it.

Your Settings window should look something like this:



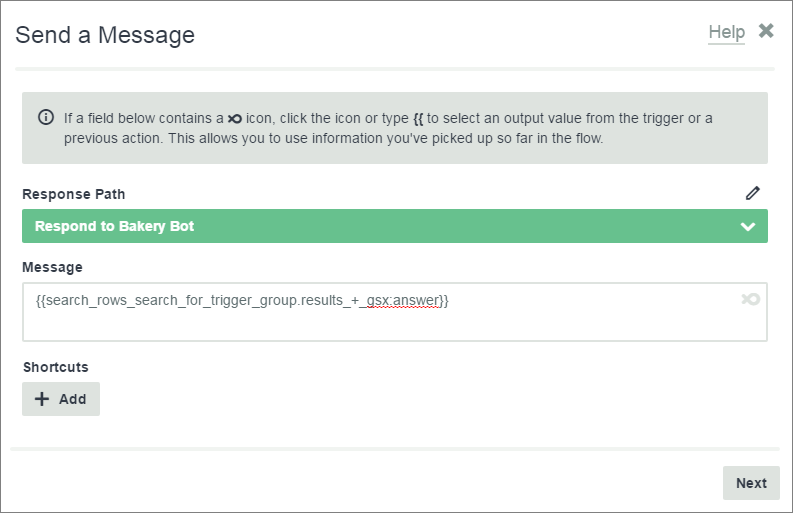
Click **Next**,then finish and save the action. When this action runs, your bot will search the **Question** column of your spreadsheet and find the group that matches the trigger output.

#### Send a Message

Next, your bot should display the **Answer**on your spreadsheet that corresponds to the **Question**it found in the last step**.**To do this:

1. In your flow, add an action. Select **Bot → Send a Message**,and click **Next**.
2. Instead of typing a message in the **Message**box, tell your bot to look back at the output from your **Search Rows** action. Then have it display the contents of the corresponding **Answer**column.

Here’s what the Send a Message window should look like:



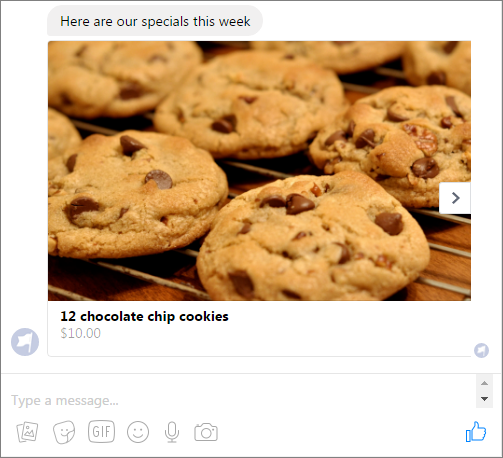
Your bot will send a response from your spreadsheet that depends on the trigger output. So, if the user sends a greeting, your bot will reply with the text in the **greeting** row, **Answer**column.

With this setup, your bot can respond to any input from the user with just two actions. As long as there is a group in the trigger and matching group in your spreadsheet, your bot will always have the right response.

### Send Cards with Google Sheets

In Flow XO, you can also use Google Sheets to send a card set to users. With cards, your bot can send links and captioned images. If you want to change card content regularly, it can be helpful to keep data in a spreadsheet rather than editing your flow directly.

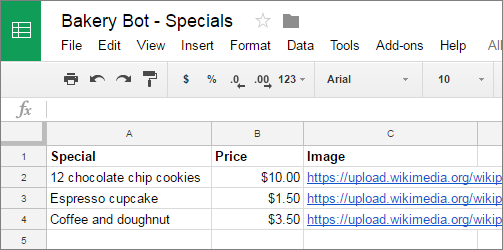
For example, you might want your bakery bot to display weekly specials as a card set. You can set up your flow once, and change the specials each week in Google Sheets.



#### Set Up Cards in Google Sheets

The first step is to set up a spreadsheet for cards. You’ll input the same data you would in a normal card set. But, instead of putting it directly in your flow, you’ll put it in a spreadsheet. To do this:

1. Create a new spreadsheet in Google Sheets and give it a clear name.
2. Add a header row to your spreadsheet. In this example, we’ll use the headings, **Special**, **Price**, and **Image**.
3. Add content to your spreadsheet. Describe each Special and add a Price. In the Image column, paste the public URL to the image you want to display:

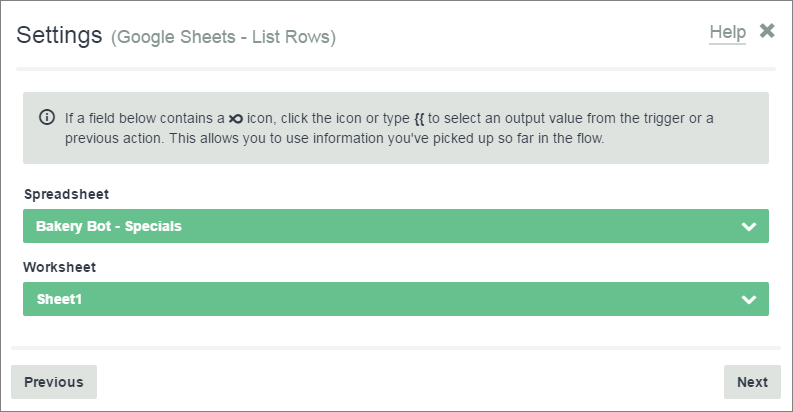


The next step is to log this data so you can use it in your flow.

#### List Rows

To log your spreadsheet data, use the **List Rows**action in your flow. This action will read and log the entire spreadsheet so that you can use all of the data in later actions. Here’s how you can set this up:

1. In your flow, add an action. Select **Google Sheets → List Rows**, then click **Next**.
2. In the Settings window, select the **Spreadsheet**and **Worksheet**you want your bot to log. Because this action reads the entire spreadsheet, there’s no need to specify any rows:



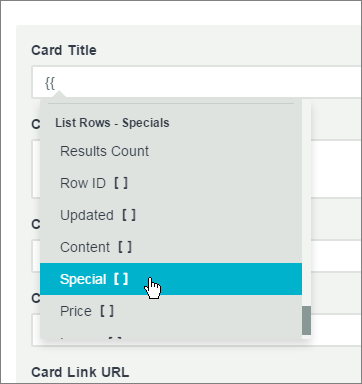
When this action runs, your bot will read all of the data in the spreadsheet. You’ll use this data in the next action.

#### Send a Card Set

Now you can set up a card set. You’ll use the output from the **List Rows** action in the fields for each card. To do this:

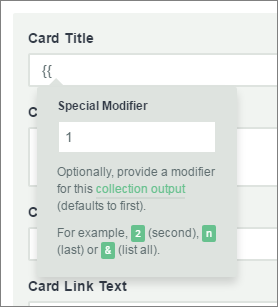
1. In your flow, add an action. Select **Bot → Send a Card Set**, and click **Next**.
2. For the first card, use the data from the first row of your column. You’ll use **collection outputs**to signal the row you want to use.

In the **Card Title**field, type **{{.**Then, in the drop-down menu, find the output from your **List Rows**action. Select the column that you want to take the card title from, in this case **Special**:



As soon as you select a column, the **Special Modifier**window will pop up. Here is where **collection outputs**come in. In this case, your spreadsheet is the collection.

* For the first card, you want your bot to look in the first row of the spreadsheet. The first row is item 1 of the collection.
* So, type **1**in the Special Modifier window. This will tell your bot to only use the data from the first row:

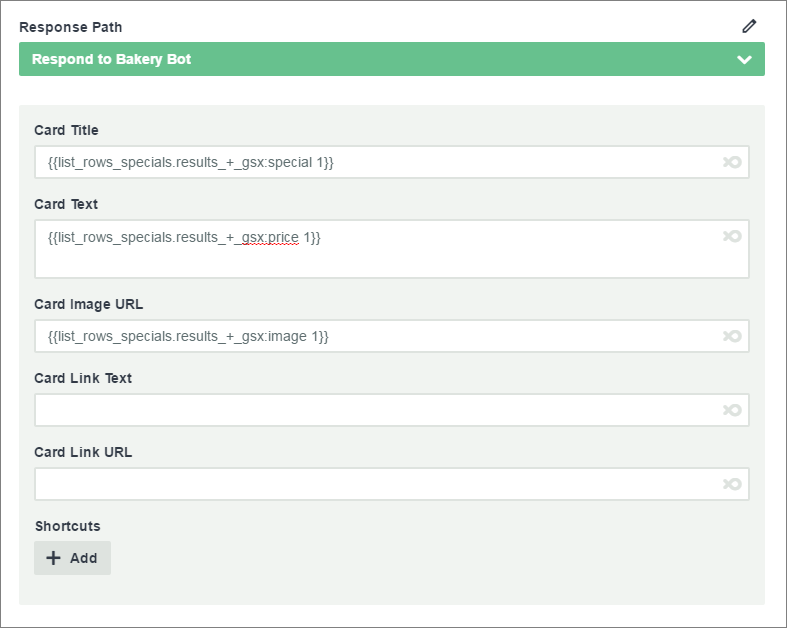


***Note:****If you leave the Special Modifier blank, it will default to 1. You can either type “1” or leave a blank for the first row. You must specify a collection output for all other rows.*

When you press **Enter**, your first Card Titlewill look like this:



Complete the rest of the fields in the same way. Use your **Price**column in the **Card Text**field and your **Image**column in the **Card Image URL**field:



To add the next card, click **+ Add Card**. For the second card, follow the same steps but change the collection output to **2**. For the third card, change the collection output to **3**.

Now your bot can read the data on your spreadsheet and send it to the user as a card set.

### Conclusion

In Flow XO, Google Sheets can help you save time when you create a flow and when you update it later. In addition to collecting data, you can use Google Sheets to respond to users and even send images.