

Building solutions with Dataverse for Teams

Lab 4 - Power Virtual Agents

Workshop Version: 1.3, Published: 08-2022

Table of contents

Exercise 1: Creating a basic bot	2
Task 1: Create a bot using Power Virtual Agents.....	3
Task 2: Working with conditions and variables	7
Task 3: Get bot response from Dataverse for Teams table	12
Task 4: Publish and test bot in Microsoft Teams.....	20
Optional exercise: Build complex queries	23
Task 1: Create an additional topic	24
Task 2: Query Dataverse for Teams tables using flow.....	26

Exercise 1:

Creating a basic bot



Important – Working with lab tenant

- All the labs in this course require you to use the latest version of Edge or Chrome in Incognito/InPrivate mode.
- Use Office 365 credentials retrieved from Labs on Demand (Skillable) in Lab 0.
- Always remember to replace M365XXXXXXX with your lab tenant prefix.
- If you are experiencing any problems with working in your lab tenant – please, notify your instructor as soon as possible.

Objectives:

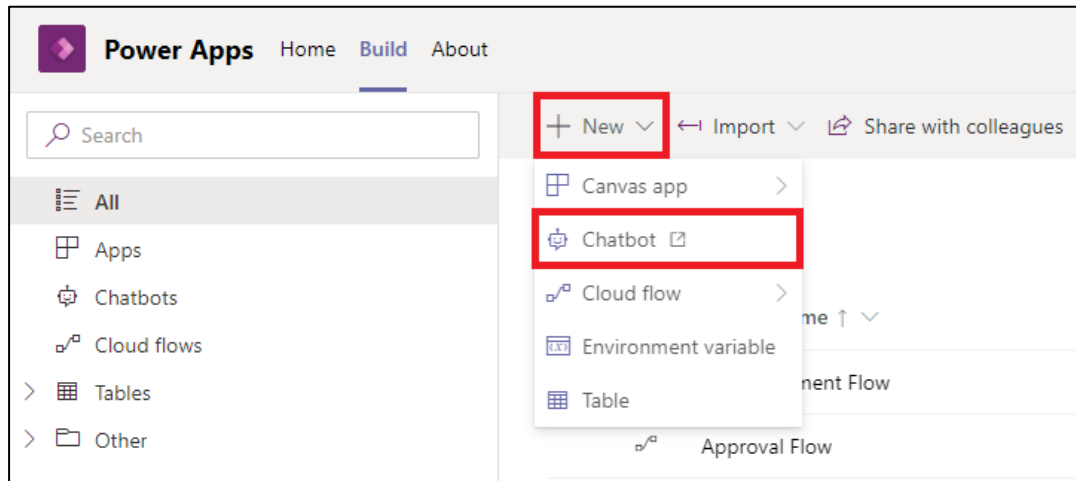
- Create a basic bot using Power Virtual Agents
- Author bot topics
- Use a flow to retrieve rows from Dataverse for Teams tables
- Test bot

Estimated time:

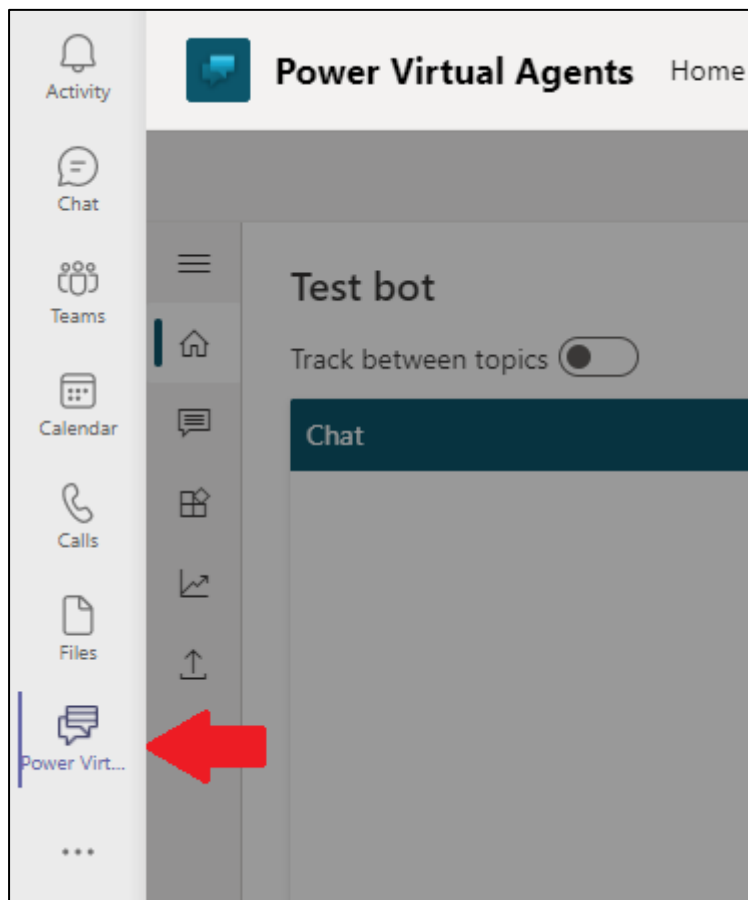
30 minutes

Task 1: Create a bot using Power Virtual Agents

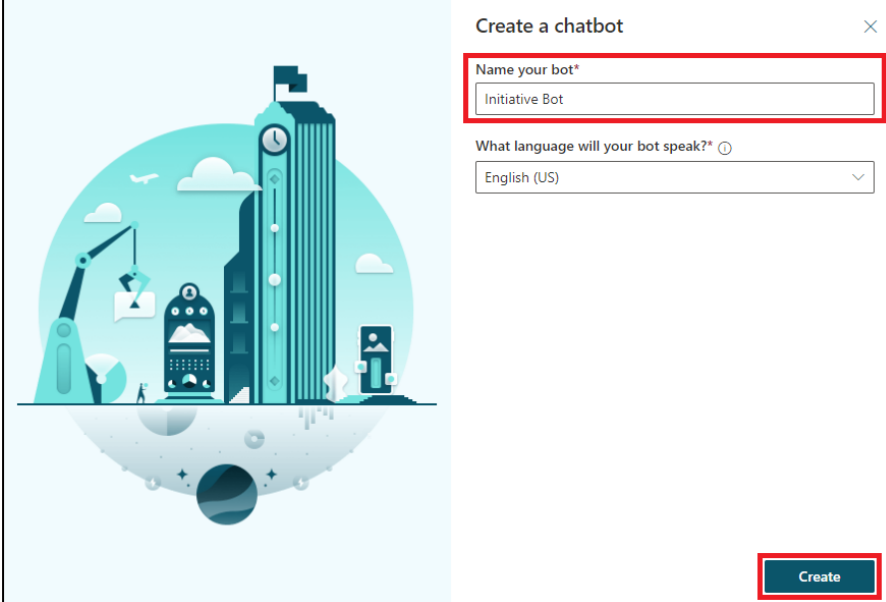
1. **Navigate to Dataverse for Teams content** of your Team
2. Click **+New** in a toolbar and select a **Chatbot** option



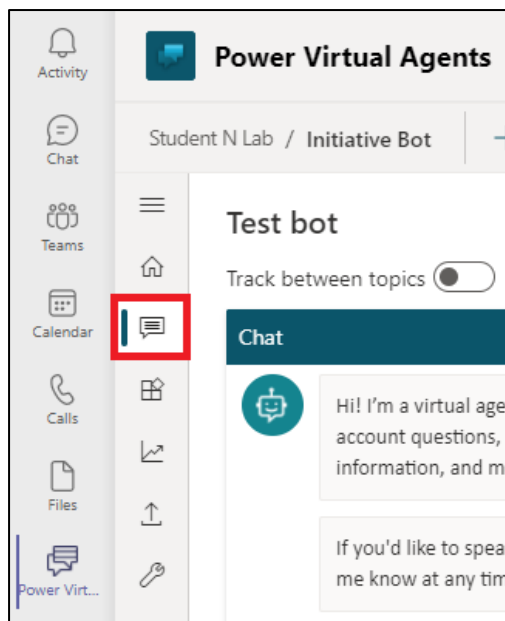
3. Pay attention, that you are redirected to **Power Virtual Agents** app to authorize the bot



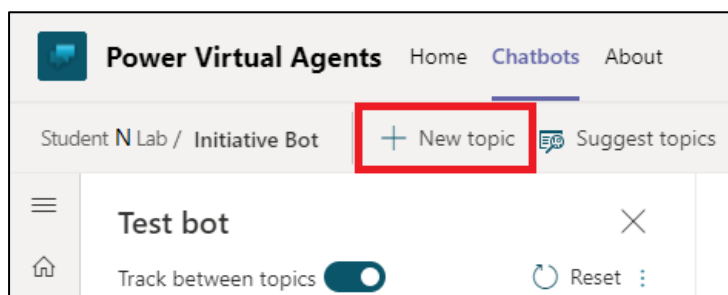
4. Define bot name as *Initiative Bot* and set a supported language, that you're using with the bot, then click **Create** button



5. In left navigation bar select **Topics** icon 

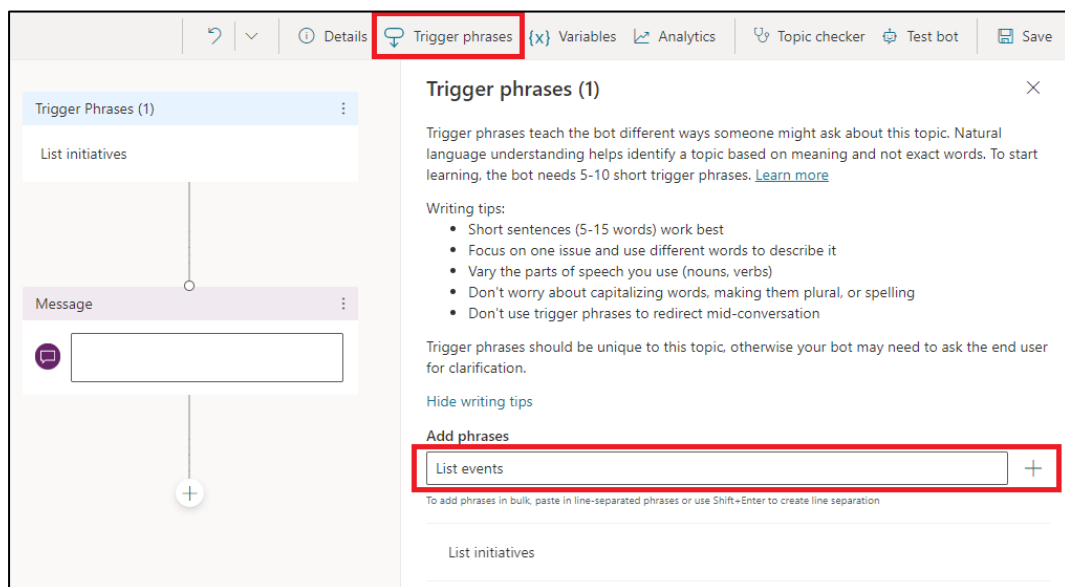


6. In toolbar click **+New topic** button



7. Set some **Trigger phrases** by typing them and clicking + button, e.g.:

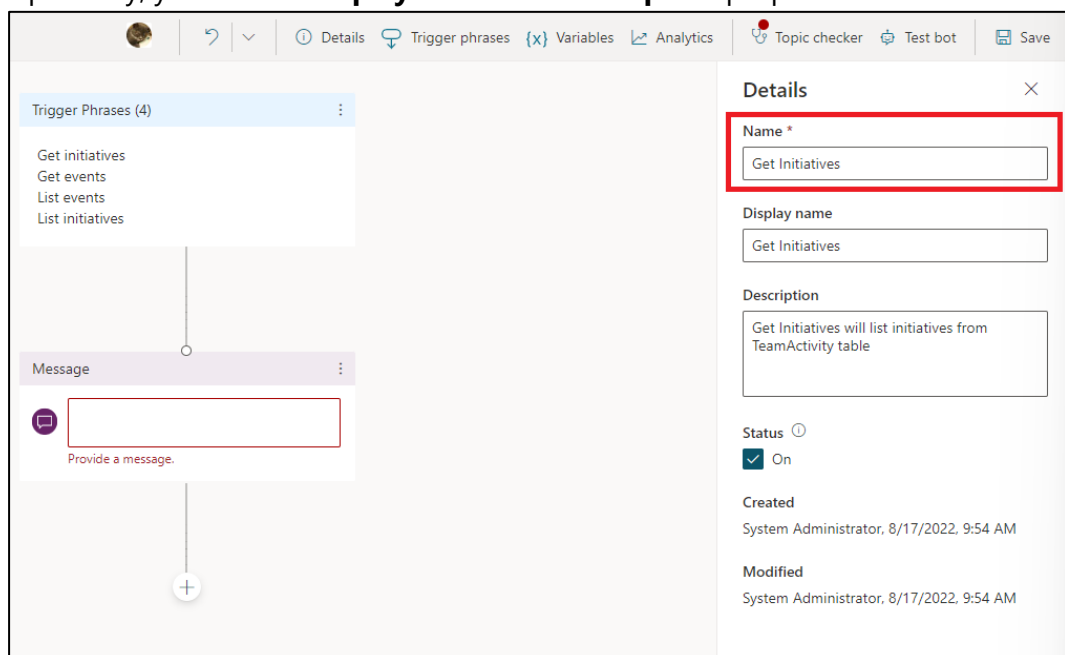
- List events
- List initiatives
- Get events
- Get initiatives



8. Click **Details** in the toolbar to open the topic details pane.

Specify topic **Name** as '*Get initiatives*'.

Optionally, you can set **Display name** and **Description** properties.

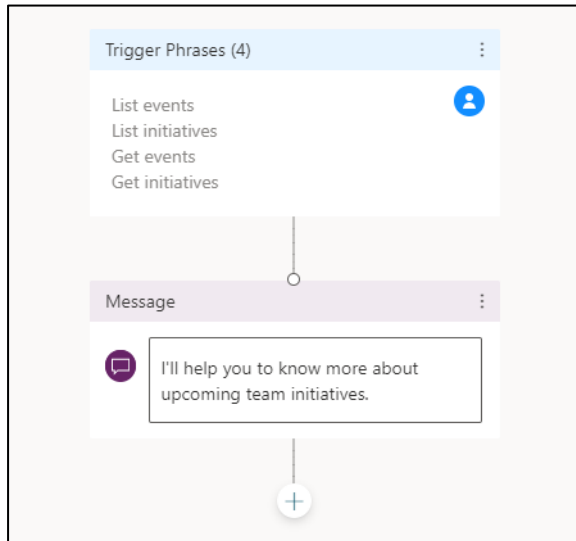


You may close pane using **X** in the top-right corner.

9. You will see a **Trigger phrases** node and an empty **Message** node on the authoring canvas.

Add any message to a **Message** node.

For example: *I'll help you to know more about upcoming team initiatives.*

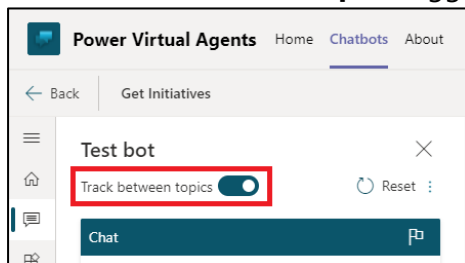


10. Using right side of a toolbar, make sure that **Topic checker** doesn't show any errors and **Save** the topic.

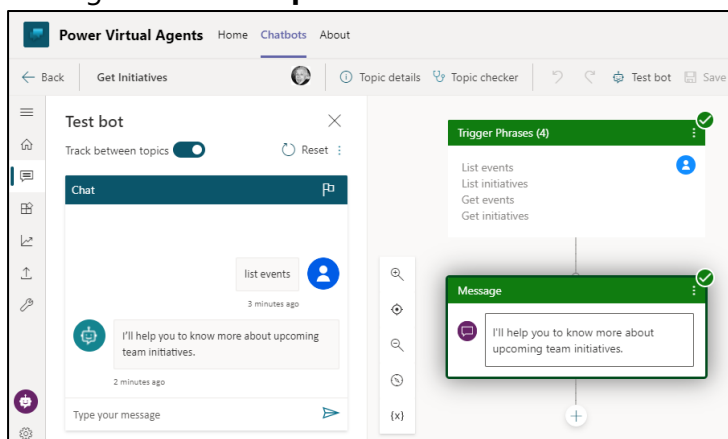


11. Let's test this topic now.

Enable **Track between topics** toggle on the top of the chat window

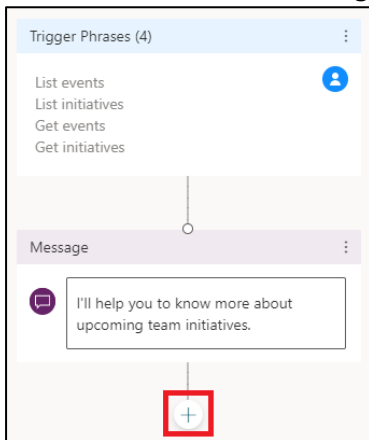


12. Type *list events* to the chat window in the left side of the screen and send the message. **Task is completed.**

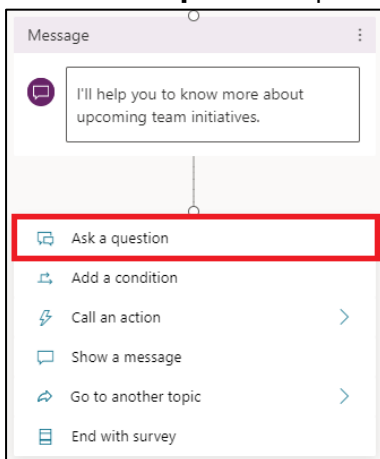


Task 2: Working with conditions and variables

1. Click + icon in the authoring canvas to add a new node



2. Select **Ask a question** option

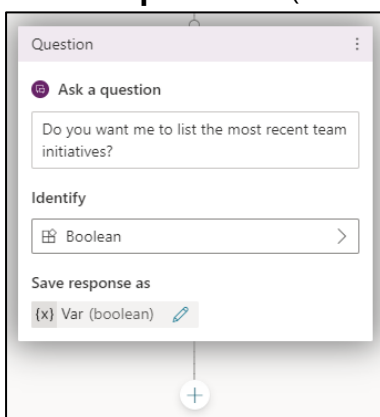


3. Set the following question properties.

Question text: Do you want me to list the most recent team initiatives?

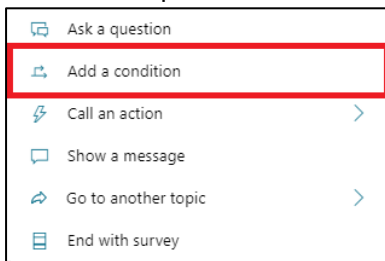
Identify: Boolean

Save a response: Var (Boolean) – leave as it is

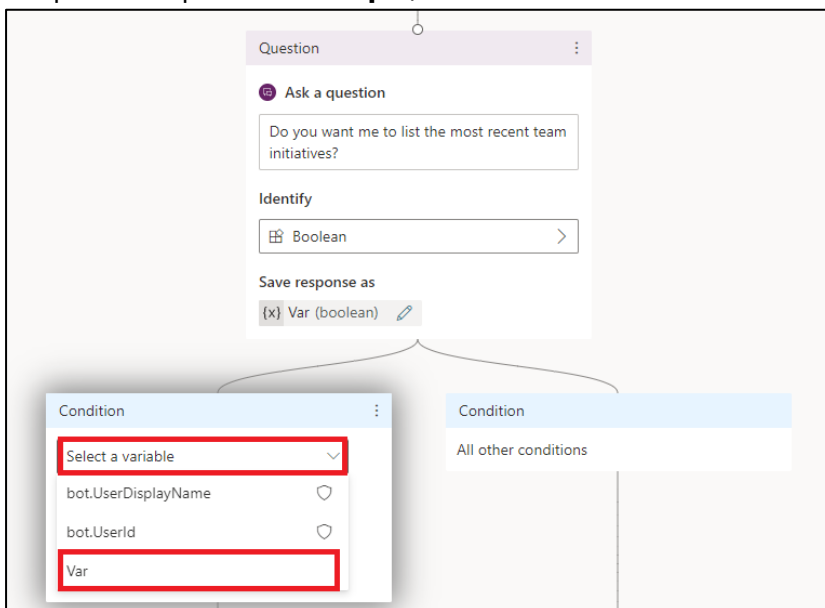


Boolean built-in entity allows chatbot to handle various positive and negative responses from the user (e.g., Yes/No, True/False)

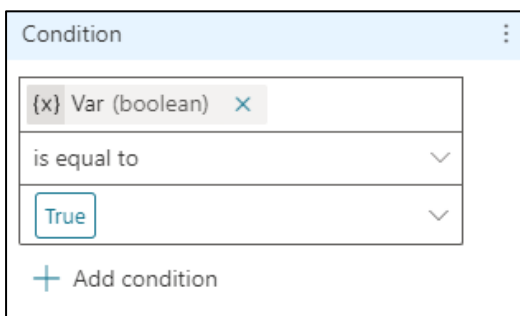
- Click **+** icon in the authoring canvas to add a new node and select an **Add a condition** option



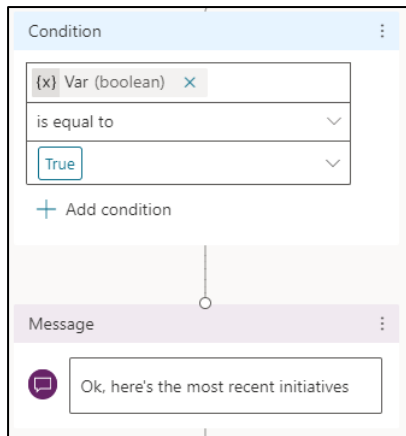
- In the **left** condition branch click **Select a variable** and specify **Var** variable (result of the previous question in **Step 3**)



- Set the other condition parameters as: **is equal to** and **True**.
The condition node should look like this:

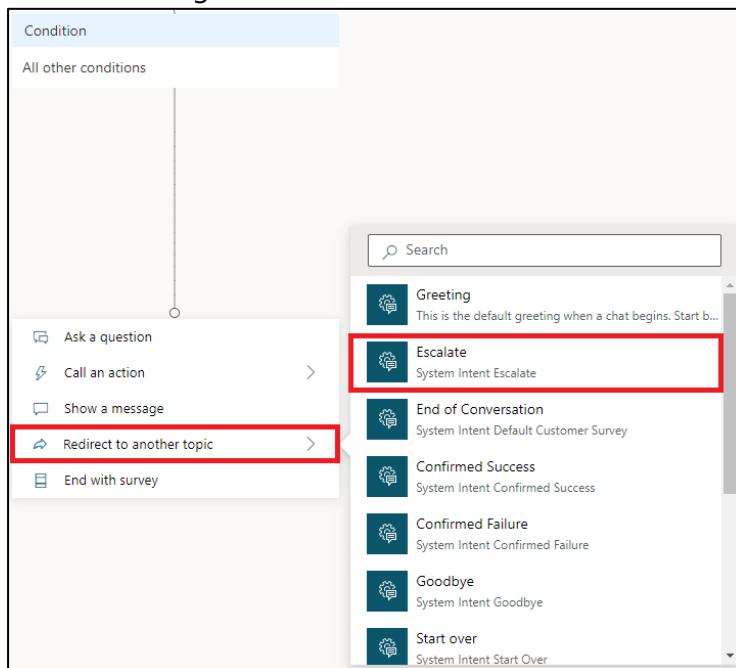


- Add a **Message** (Show a message) node to the end of the left condition branch.
Set message text as *Ok, here's the most recent initiatives.*

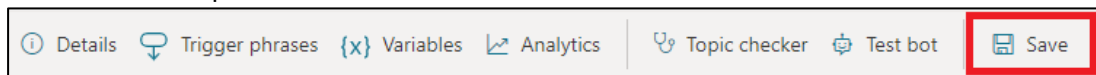


8. In the end of the **right** condition branch, add a **Redirect to another topic** and select built-in **Escalate** topic.

In this example, bot will fall back to a built-in human agent escalation topic, if your answer was negative.

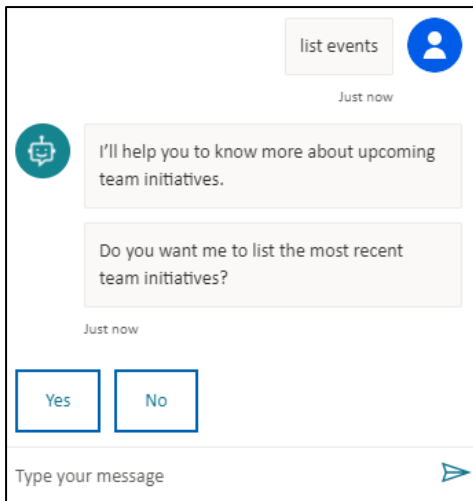


9. Using right side of a toolbar, make sure that **Topic checker** doesn't show any errors and **Save** the topic.

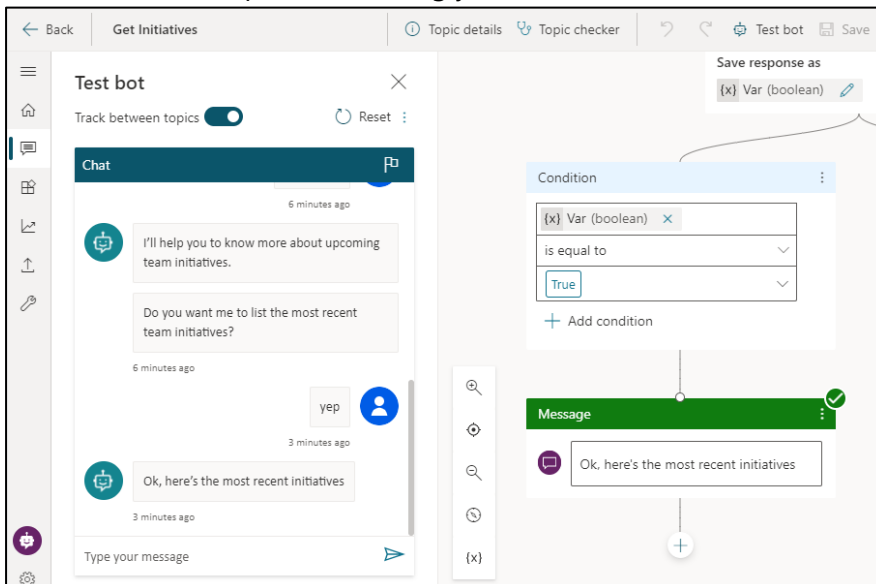


10. Let's test this topic once again.

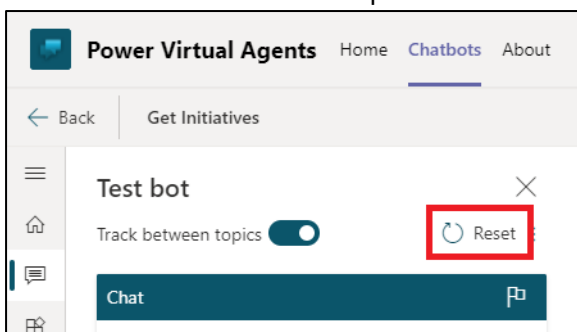
Type *list events* to the chat window in the left side of the screen and send the message.



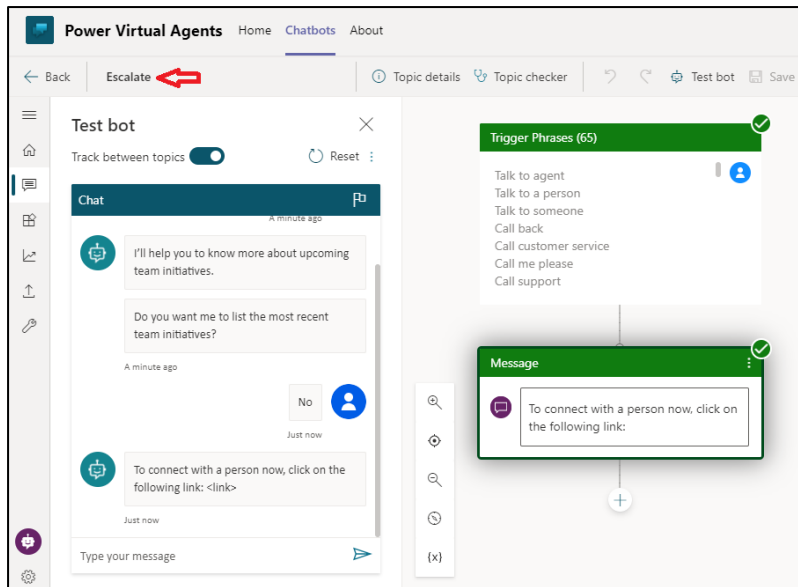
Click **Yes** button or send positive response through the chat input. The bot should respond accordingly.



11. Click **Reset** button on the top of the chat window to start over the conversation.



12. Send *list events* to the chat window once again and click **No** button or send negative response. You will notice how conversation is redirected to a built-in **Escalate** topic.

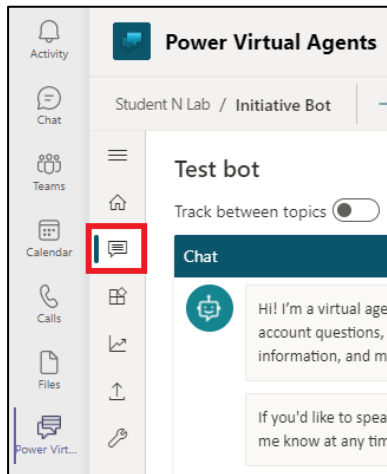


13. **Escalate** topic can be customized to redirect bot user to another Teams chat by using deep link or to another web resource (e.g., service tickets web resource).











Task is completed.

Task 3: Get bot response from Dataverse for Teams table

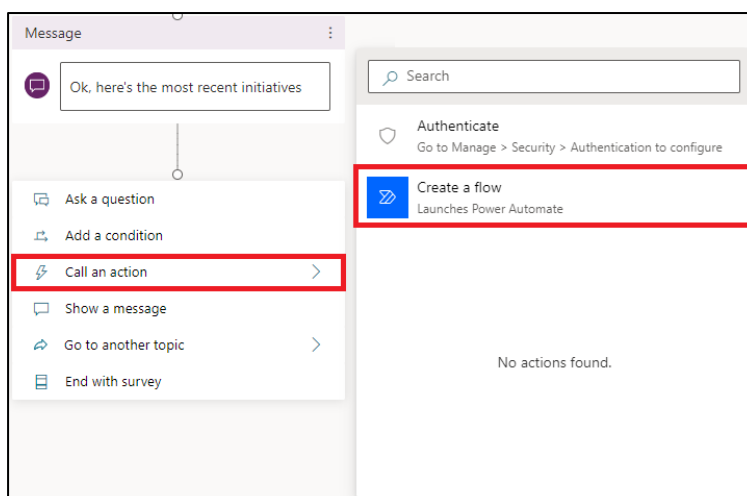
1. In left navigation bar select **Topics** icon 



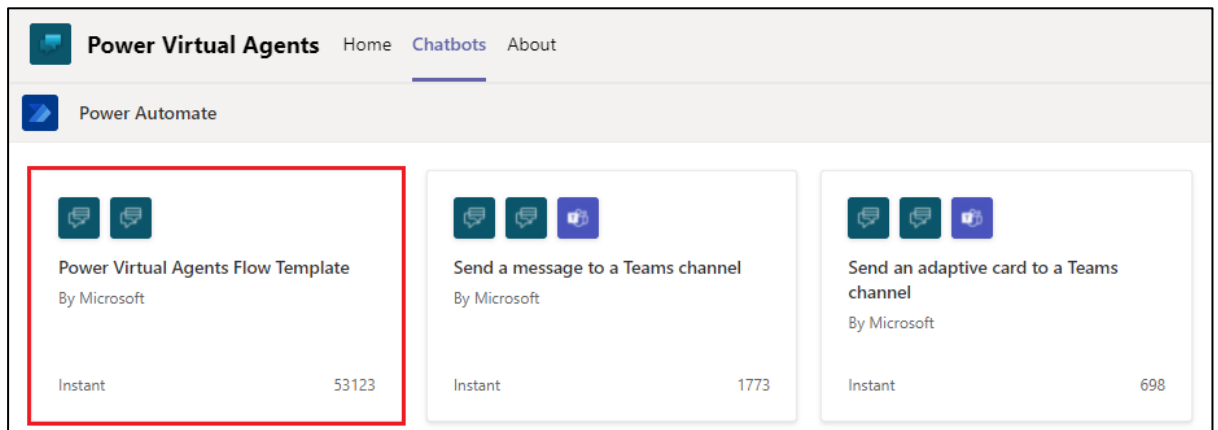
2. Select **Get initiatives** topic (from the previous tasks)

Topics ⓘ			
Existing (13)		Suggested (0)	
Type	Name	Trigger phrases	Status
	Get Initiatives	(4) Get initiatives	 On
	Lesson 1 - A simple topic	(4) When are you closed	 On
	Lesson 2 - A simple topic with a condition and v...	(5) Are there any stores aroun...	 On
	Lesson 3 - A topic with a condition, variables an...	(5) Buy items	 On
	Lesson 4 - A topic with a condition, variables an...	(5) What is the best product f...	 On

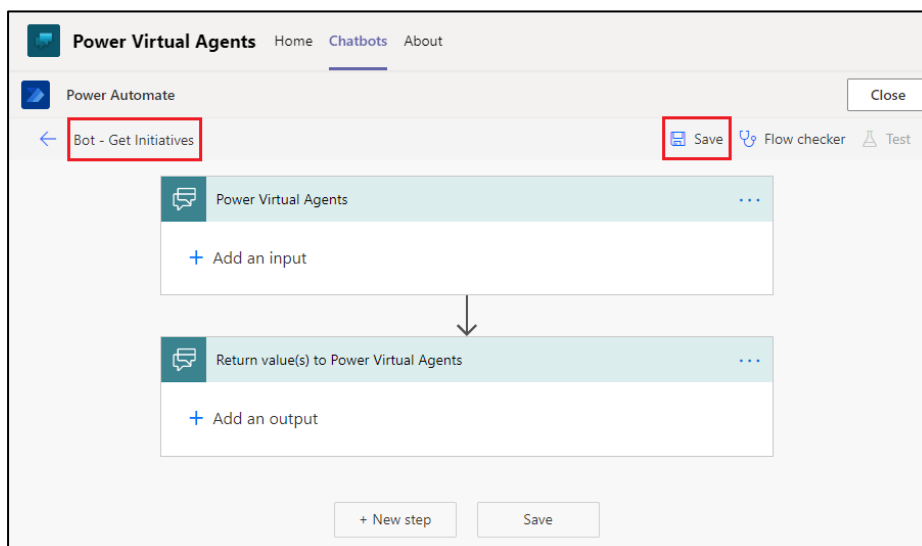
3. In the left condition branch, add **Call an action** node and select **Create a flow** option:



- Flow templates for Power Virtual Agents will be displayed.
Select the first **Power Virtual Agents Flow Template**

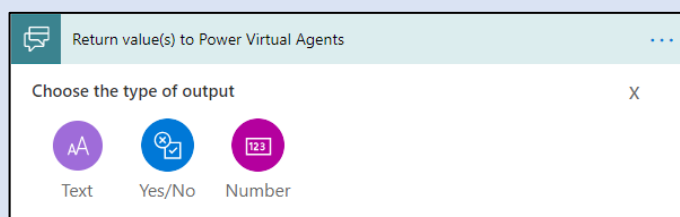


- Click on flow name in the top left corner and rename it as *Bot – Get Initiatives*, then click **Save** button



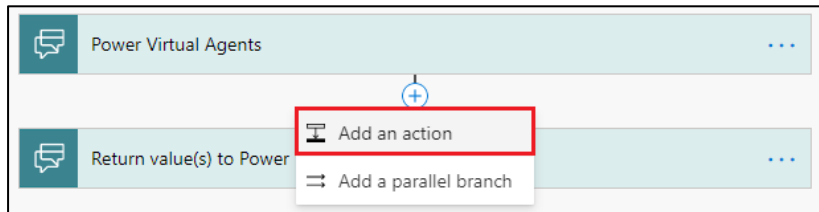
Return values to Power Virtual Agents

It's not possible to simply return a collection of objects as a bot response.
Supported outputs are **Text**, **Yes/No**, or **Number**:

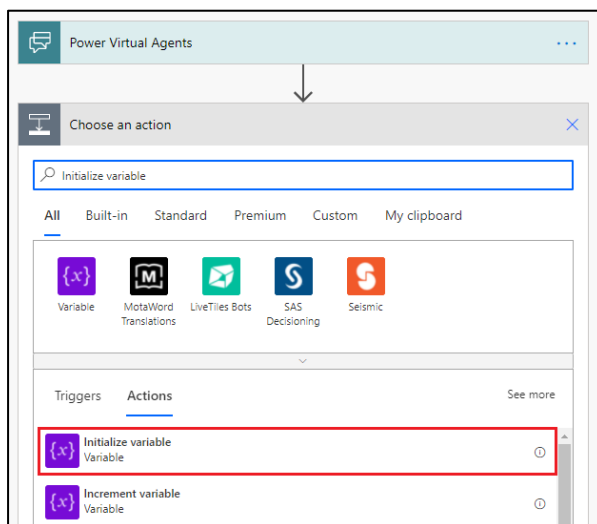


This is a reason to pre-create a *String* variable that will hold bot response text in the next step.

6. **Add an action** in between two existing blocks:



Add an **Initialize variable** action from **Variables** 'connector' by clicking on it (use Search if this connector or action is not present)



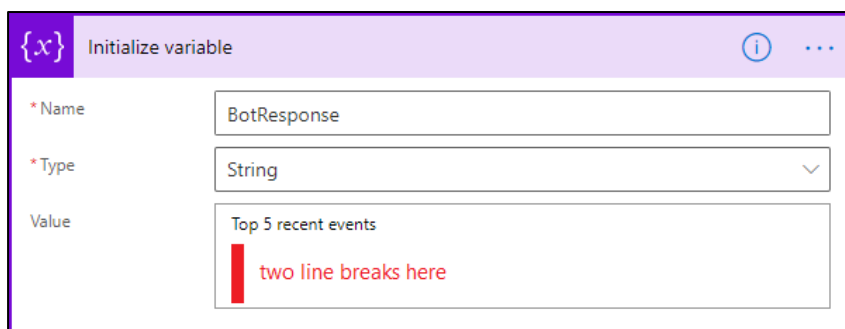
7. Set properties **Initialize variable** action as:

Name: BotResponse

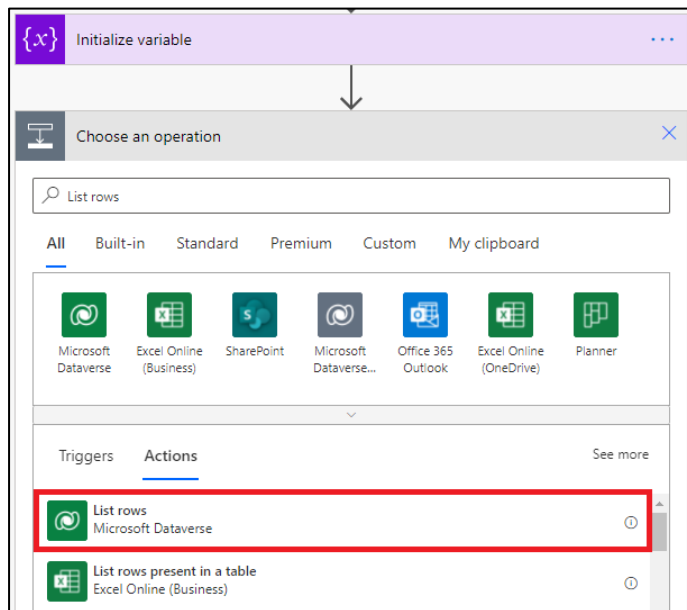
Type: String

Value: Top 5 recent events
Press **Enter** twice to add two line breaks

We will append more text to the value of this variable later.



8. Right after Initialize variable action, add a **List rows** action from **Microsoft Dataverse** connector by clicking on it
(use Search if this connector or action is not present)

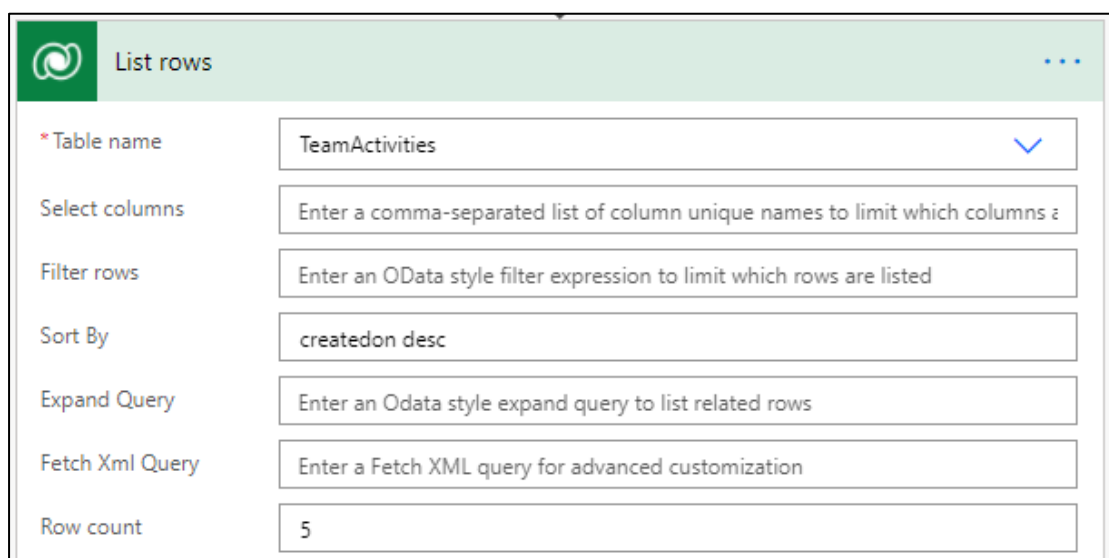


9. Set properties for **List rows** action as:

Table name: TeamActivity

Sort by: createdon desc

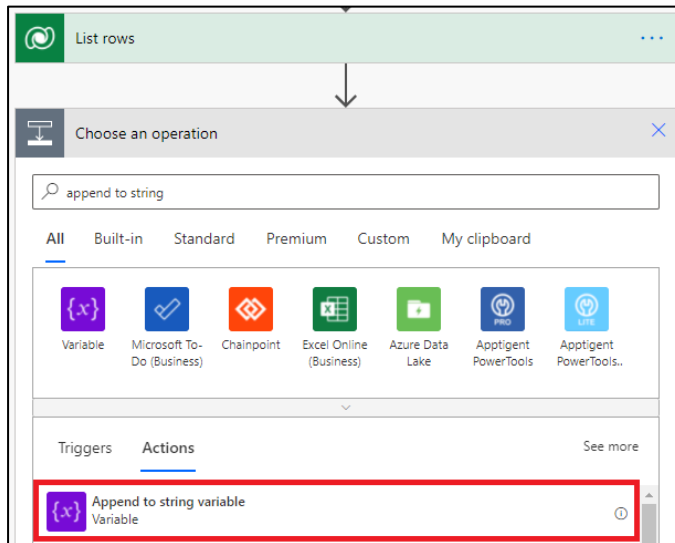
Row Count: 5



Sort By value is the part of OData query that will sort TeamActivity rows in descending order by Created On column.

Row Count in combination with Sort By will return 5 most recent rows only.

10. Right after List rows action, add an **Append to string variable** action from **Variable** 'connector' by clicking on it
(use Search if this connector or action is not present)



11. Now it's time to append **List rows** query results to **BotResponse** variable that we've defined before.



Using Markdown in bot messages

Markdown is a popular lightweight markup language with plain text formatting syntax.

Power Virtual Agent chatbots support markdown in bot messages – we will use it to format flow outputs and send hyperlinks to the user.

Headers:

This is level 1 header

This is level 2 header

Bold text:

****this is bold text****

Hyperlink syntax:

This is an [example link](<http://example.com/>)

More about Markdown (official reference): [Daring Fireball: Markdown Basics](#)

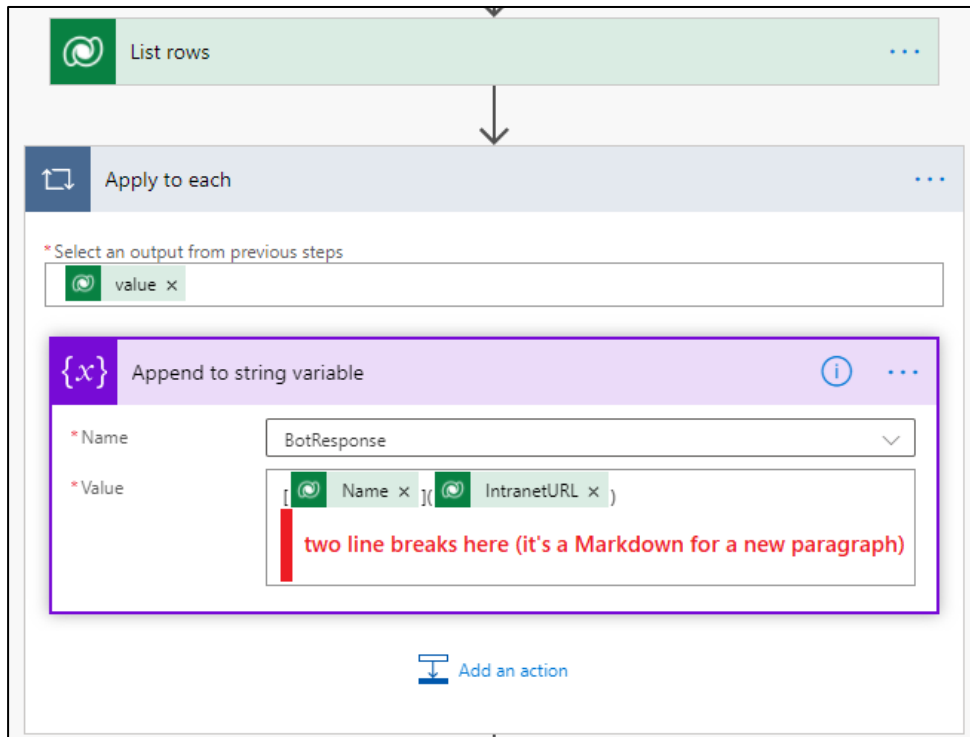
Set properties for **Append to string variable** action as:

Name: BotResponse

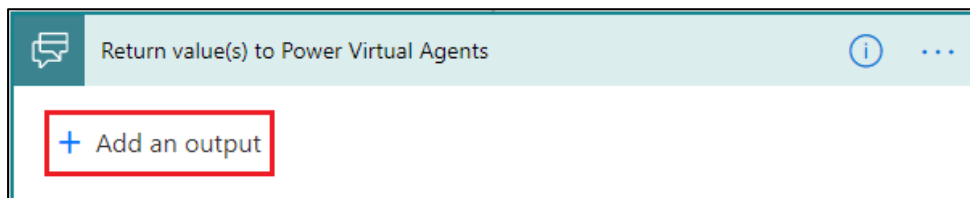
Value: [Name](IntranetURL) (property values from **List rows** action)
Press **Enter** twice to add two line breaks

Please, double-check syntax and make sure that all brackets are in place.

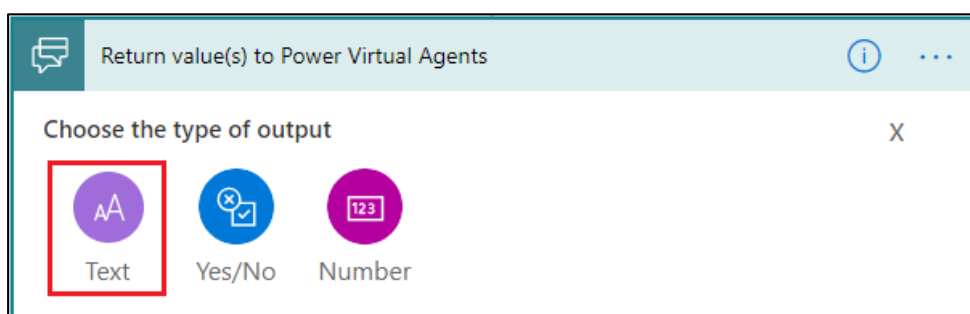
Apply to each loop will be added automatically, because List rows results may contain 0,1 or more rows.



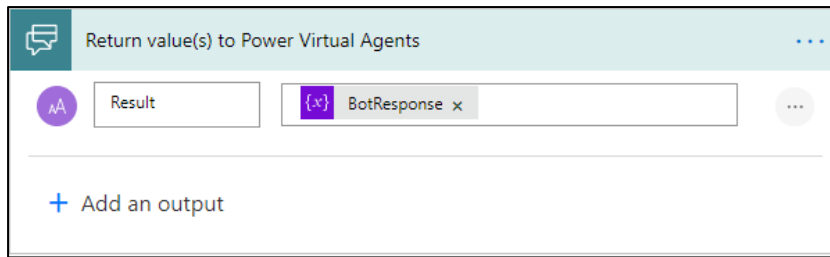
12. Expand existing **Return value(s) to Power Virtual Agents** action.
Click **+Add an output**.



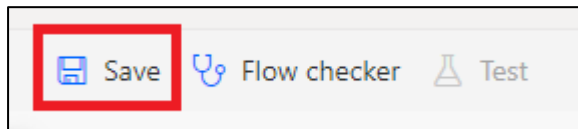
Select **Text** option by clicking on it



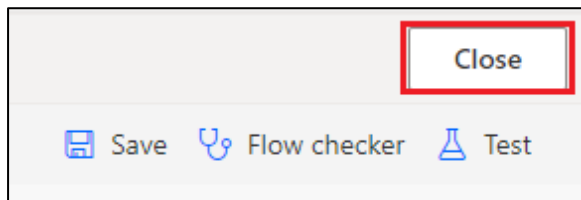
13. Set output name as **Result** and **BotResponse** variable as a value
Note: *BotResponse is a variable, not a plain text*



14. Make sure, that there are no notifications in **Flow checker** and **Save** this flow

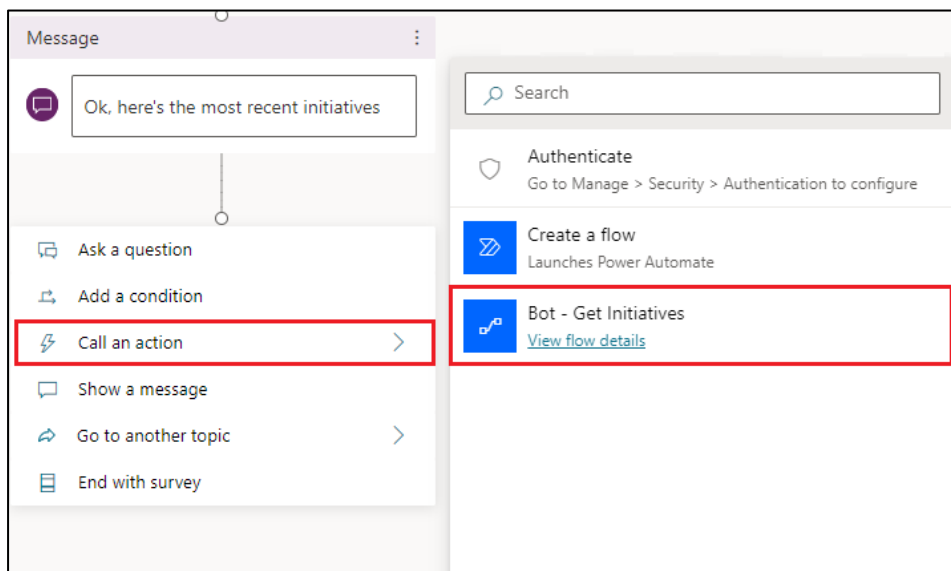


15. Click **Close** button in the top right corner after the flow is saved.

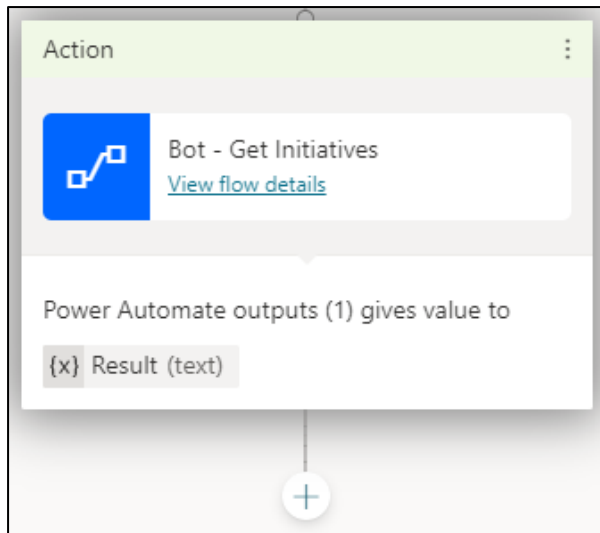


Redirect to the **Get Initiative** topic authoring canvas should happen, **if it didn't happen – repeat Steps 1,2 of this Task to navigate there.**

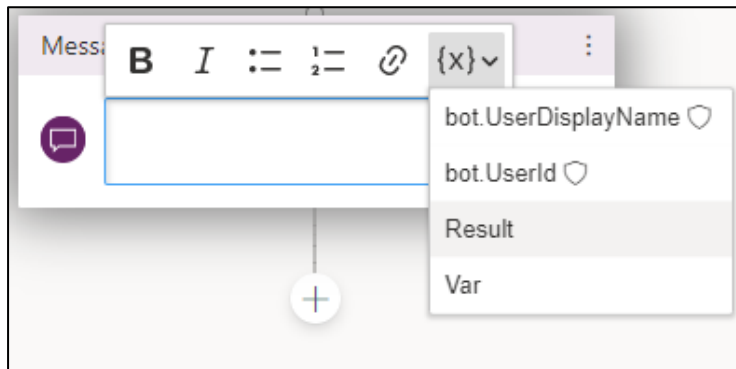
16. Once again, in the left condition branch, add **Call an action** node, select **Bot – Get Initiatives** flow:



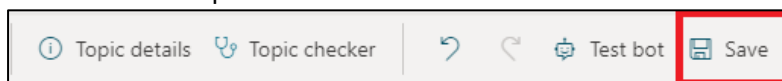
17. Action node should be displayed with the name of the flow and text variable Result as an output:



18. Add a **Message** (Show a message) node below and set **Result** variable as an output.



19. Using right side of a toolbar, make sure that **Topic checker** doesn't show any errors and **Save** the topic.

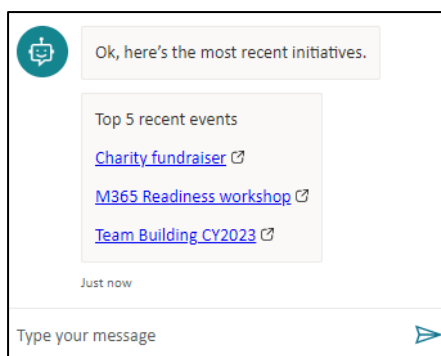


20. Let's test this topic once again.

Type *list events* to the chat window in the left side of the screen and send the message.


Click **Yes** button or send positive response through the chat input.

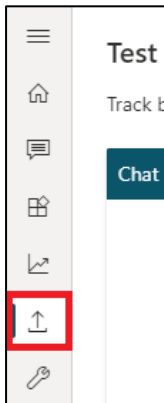
Bot should return Initiative links like this:



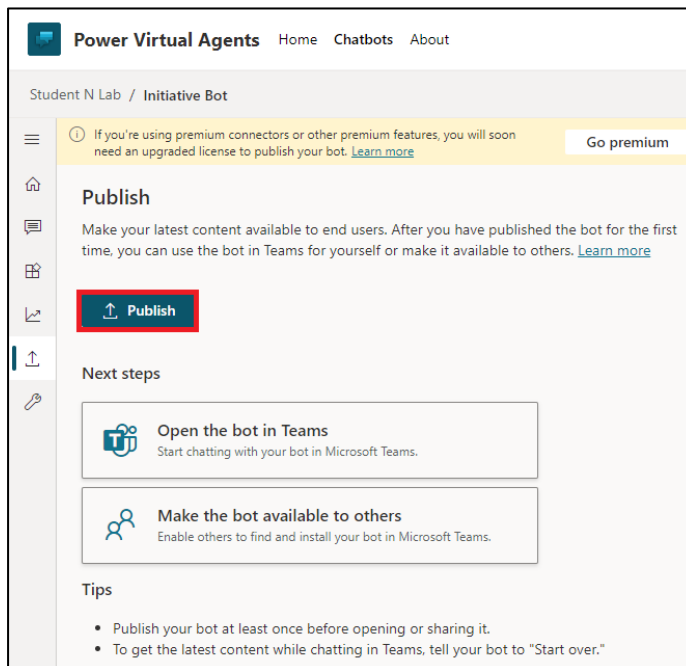
Task is completed.

Task 4: Publish and test bot in Microsoft Teams

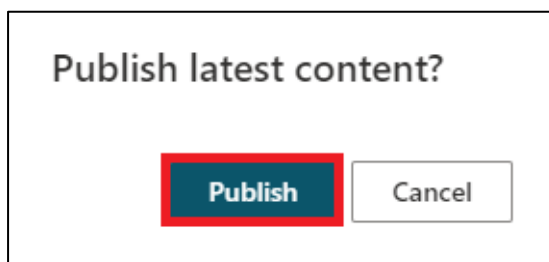
1. In the left navigation bar, click on **Publish**  icon



2. Click **Publish** button

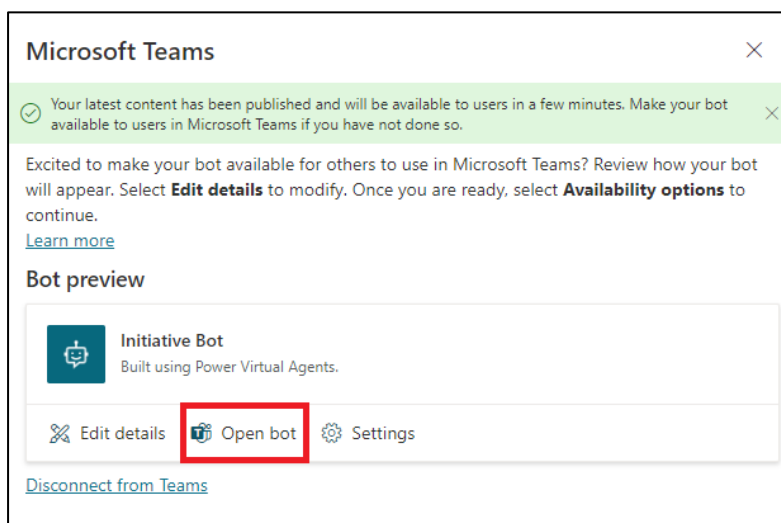


3. In **Publish latest content?** dialog popup, click **Publish** button

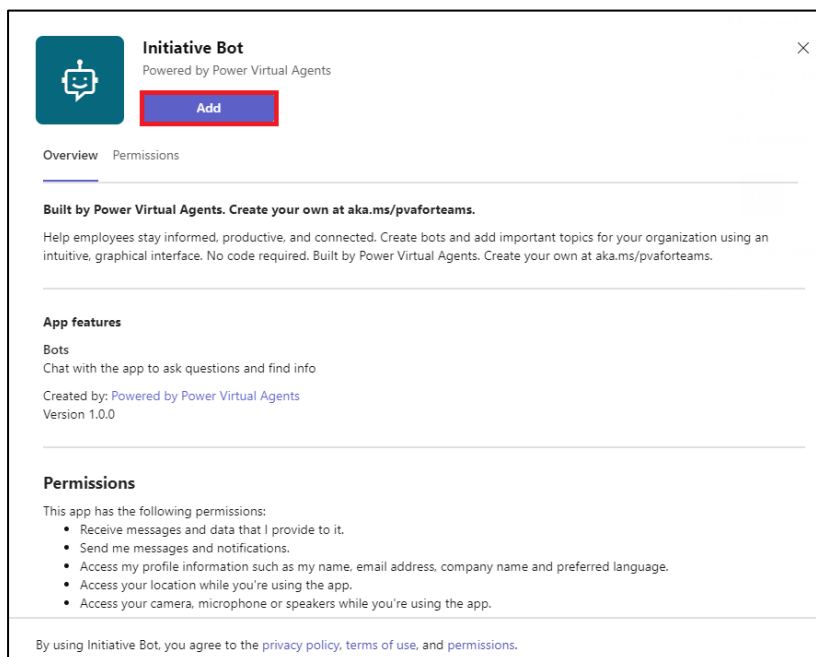


It will take a few moments to publish the bot.

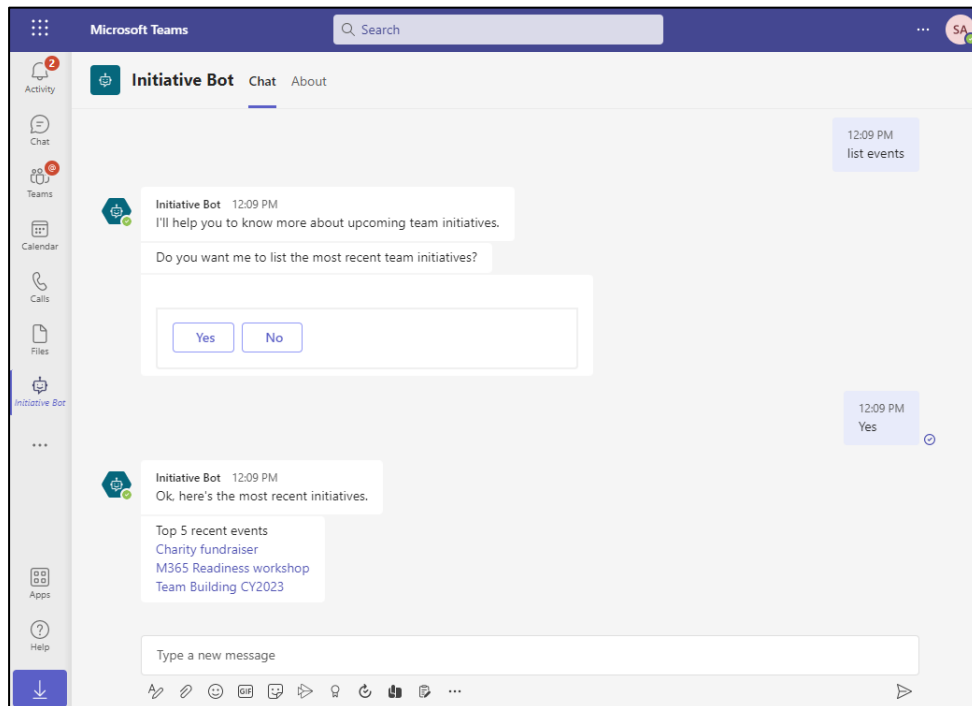
4. Click **Open bot** button to test the bot in Microsoft Teams



5. Teams App installation dialog will be opened. Click **Add** button.



6. Now you should be able to use **Initiative Bot** as a Teams app.



Task is completed.

Optional exercise:

Build complex queries



Important – Working with lab tenant

- All the labs in this course require you to use the latest version of Edge or Chrome in Incognito/InPrivate mode.
- Use Office 365 credentials retrieved from Labs on Demand (Skillable) in Lab 0.
- Always remember to replace M365XXXXXXX with your lab tenant prefix.
- If you are experiencing any problems with working in your lab tenant – please, notify your instructor as soon as possible.

Objectives:

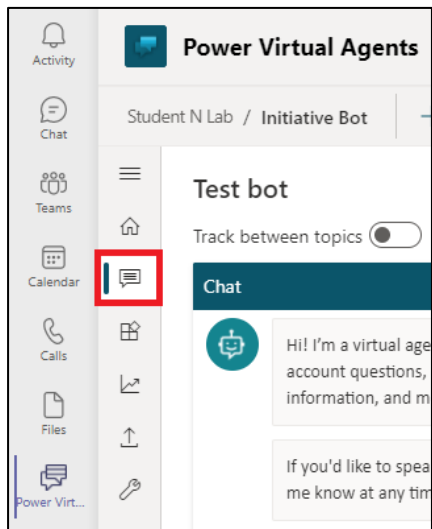
- Create an additional bot topic
- Utilize Teams authentication in bot
- Create a flow and utilize more complex queries to Dataverse for Teams tables
- Learn how to get Choices labels in flow

Estimated time:

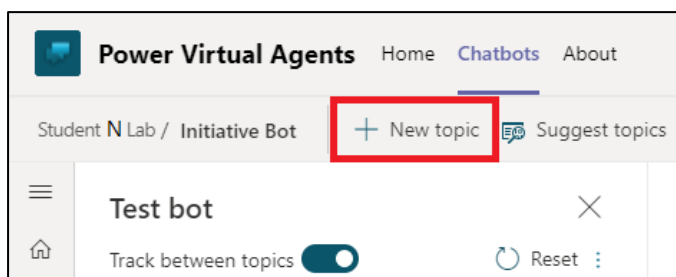
30 minutes

Task 1: Create an additional topic

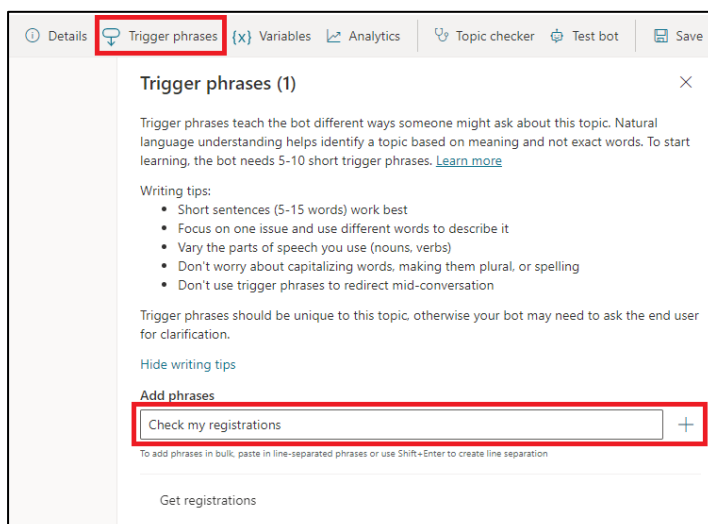
1. In left navigation bar select **Topics** icon 



2. In toolbar click **+New topic** button



3. Set some **Trigger phrases** by typing them and clicking **Add** button, e.g.:
 - Check my registrations
 - Get registrations



- Click **Details** in the toolbar and set topic **Name** and **Friendly Name** as *Check my registrations*

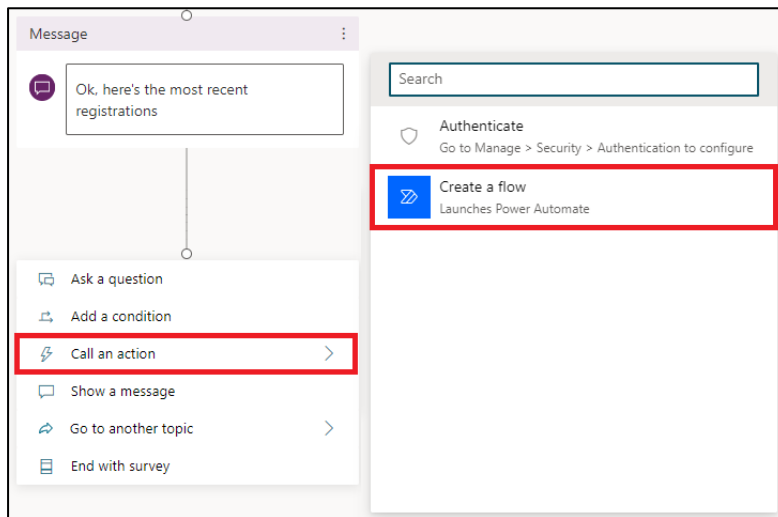
The screenshot shows a software interface with a top toolbar containing icons for 'Details', 'Trigger phrases', 'Variables', 'Analytics', 'Topic checker', 'Test bot', and 'Save'. The 'Details' icon is highlighted with a red box. Below the toolbar, a 'Details' pane is open, also with a red box around its title bar. Inside the pane, the 'Name *' field contains the text 'Check my registrations'. The 'Display name' field also contains 'Check my registrations'. The 'Description' field is empty with the placeholder text 'Enter a description'. The 'Status' is set to 'On' with a checked checkbox. The 'Created' and 'Modified' fields show 'No data'.

You may close pane using **X** in the top-right corner

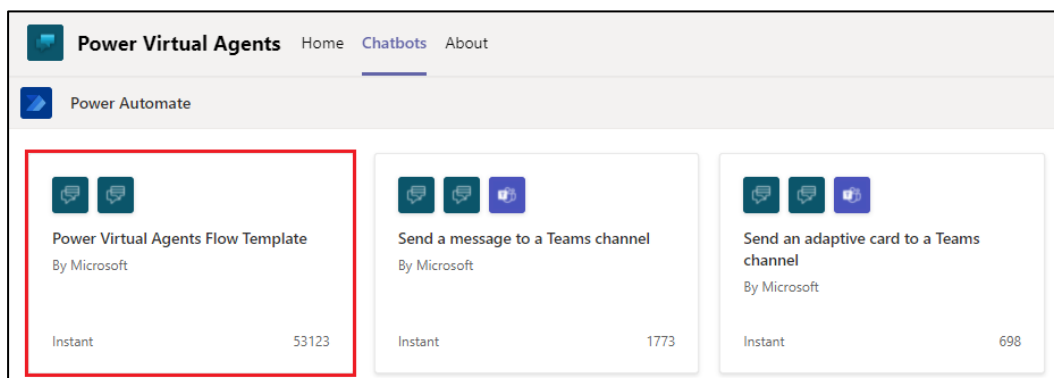
- You will see a **Trigger phrases** node and an empty **Message** node on the authoring canvas.
Add any message to a **Message** node.
For example: *Ok, here's the most recent registrations*
- Using right side of a toolbar, make sure that **Topic checker** doesn't show any errors and **Save** the topic.

Task 2: Query Dataverse for Teams tables using flow

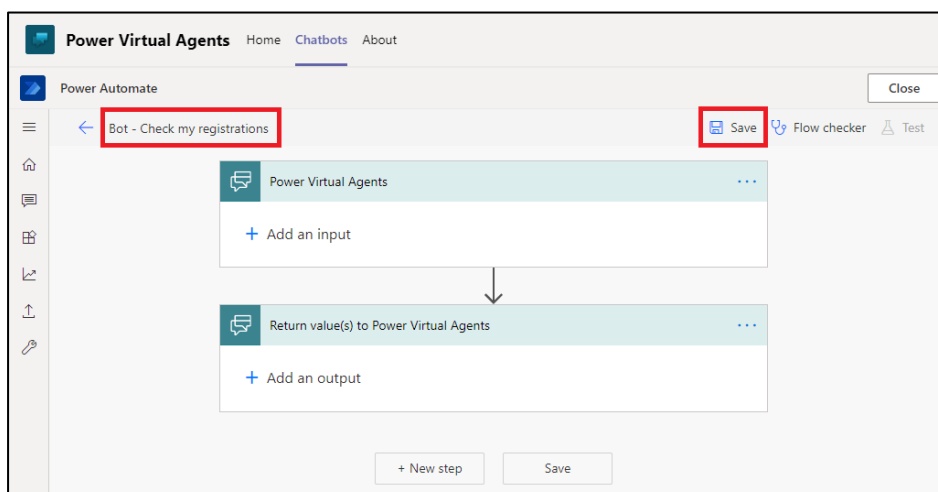
1. Add **Call an action** node and select **Create a flow** option:



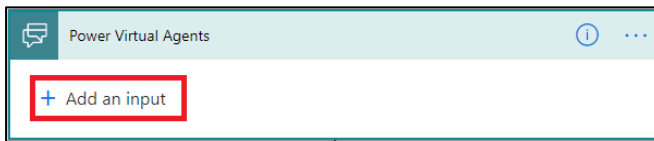
2. Flow templates for Power Virtual Agents will be displayed.
Select the first **Power Virtual Agents Flow Template**



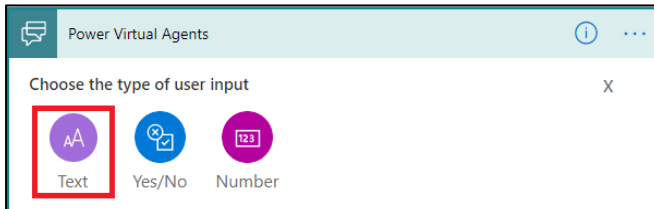
3. Click on flow name in the top left corner and rename it as *Bot – Check my registrations*, then click **Save** button



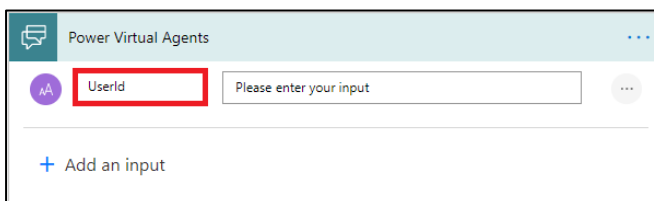
4. In **Power Virtual Agents** trigger, click **+Add an input**



Select **Text** option



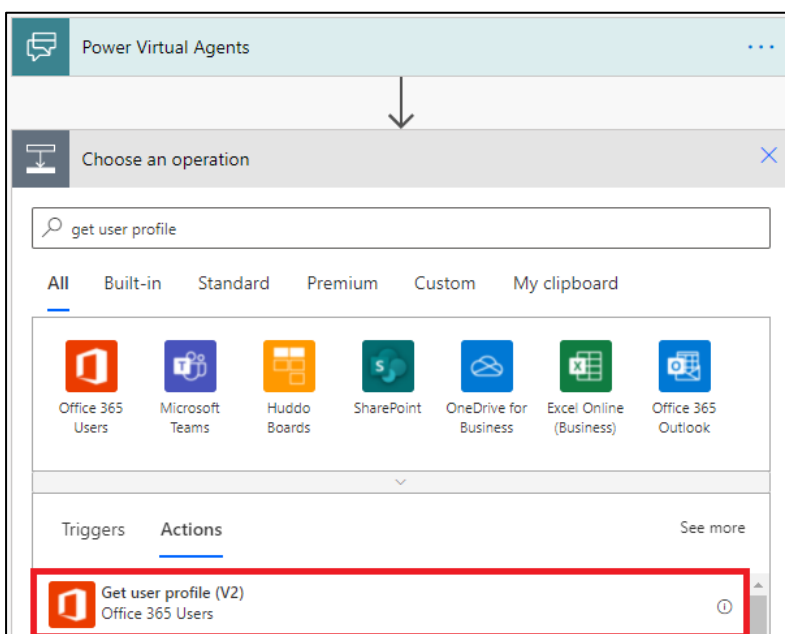
Specify input name as **UserId**



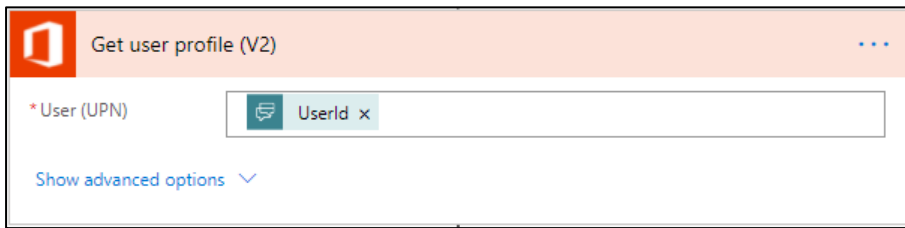
We will pass this value right from the bot.

Note: *Dataverse for Teams bots have Teams Authentication enabled by default, that's why bot will have two built-in variables with the current user's display name and AAD ObjectId.*

5. Add a **Get user profile (V2)** action from **Office 365 Users** connector

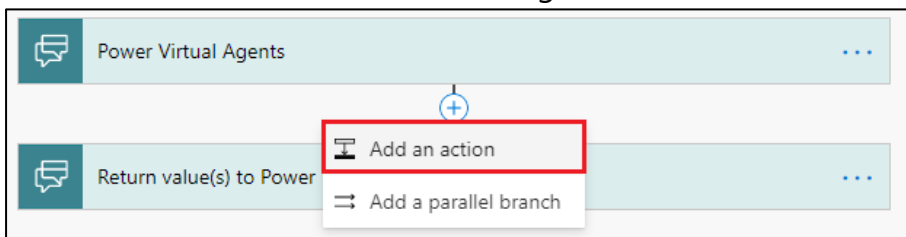


Set **User (UPN)** property as a *UserId* input parameter from trigger

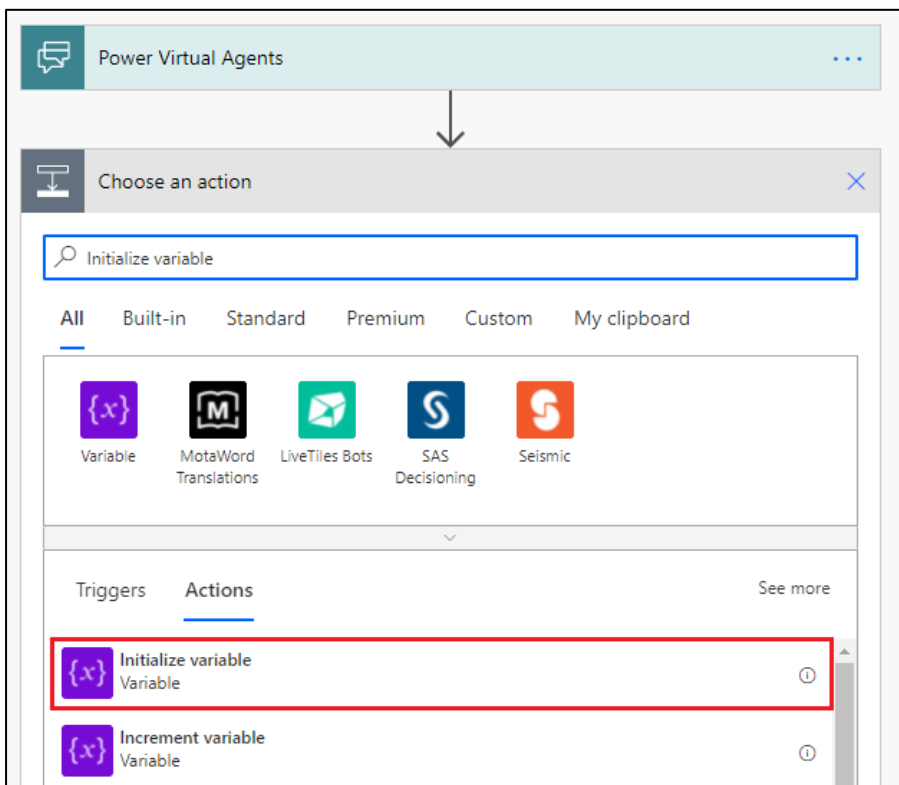


6. Let's initialize a String variable that will accumulate bot response text.

Add an action in between two existing blocks:



Add an **Initialize variable** action from **Variables** 'connector' by clicking on it (use Search if this connector or action is not present)



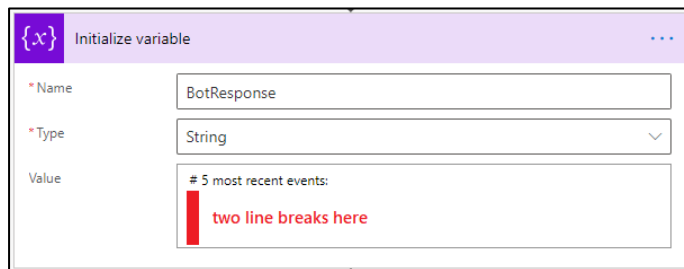
7. Set properties **Initialize variable** action as:

Name: BotResponse

Type: String

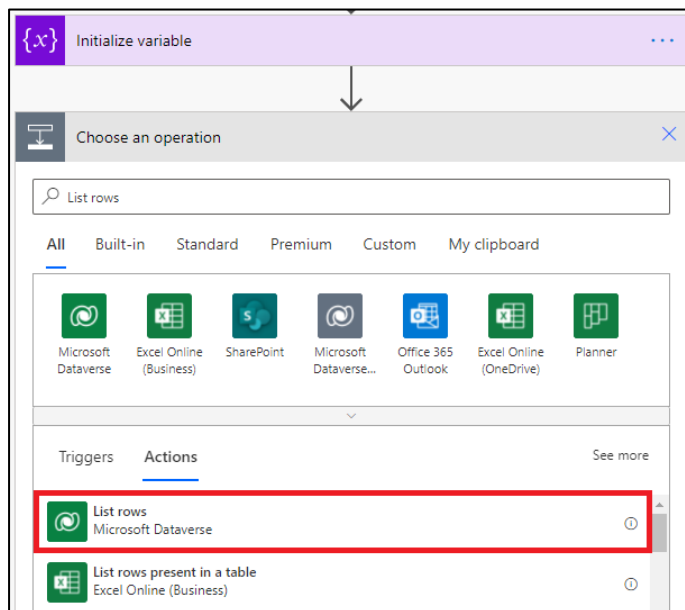
Value: # 5 most recent events:
Press **Enter** twice to add two line breaks

We will append more text to the value of this variable later.



The screenshot shows the 'Initialize variable' action configuration. The 'Name' field is set to 'BotResponse'. The 'Type' dropdown is set to 'String'. The 'Value' field contains the text '# 5 most recent events:' followed by two line breaks, indicated by red text 'two line breaks here'.

8. Add a **List rows** action from **Microsoft Dataverse** connector (use Search if this connector or action is not present)



The screenshot shows the 'Choose an operation' dialog. The search bar contains 'List rows'. The 'All' tab is selected, showing various connectors. The 'List rows' action under the 'Microsoft Dataverse' connector is highlighted with a red box.

9. We need to get 5 most recent registrations for the current user and display them as Intranet URL links with the approval status info.

Let's use some advantages of querying Dataverse for Teams tables and relationships between them.

Actually, we can retrieve all the required data in a single query.

Set properties for **List rows** action as:

Table name: Registrations

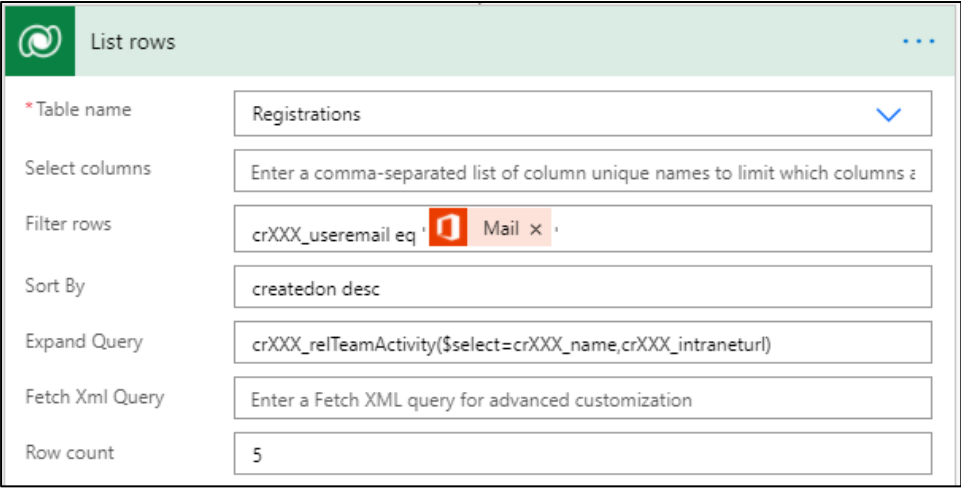
Filter rows: crXXX_useremail eq 'Mail' (from Get User Profile (V2))

(pay attention to single quotes, replace crXXX with your environment's PublisherId prefix)

Sort by: createdon desc

Expand Query: crXXX_relTeamActivity(\$select=crXXX_name,crXXX_intraneturl)
(replace with your environment's PublisherId prefix)

Row Count: 5



List rows	
* Table name	Registrations
Select columns	Enter a comma-separated list of column unique names to limit which columns are returned
Filter rows	crXXX_useremail eq 'Mail' x
Sort By	createdon desc
Expand Query	crXXX_relTeamActivity(\$select=crXXX_name,crXXX_intraneturl)
Fetch Xml Query	Enter a Fetch XML query for advanced customization
Row count	5

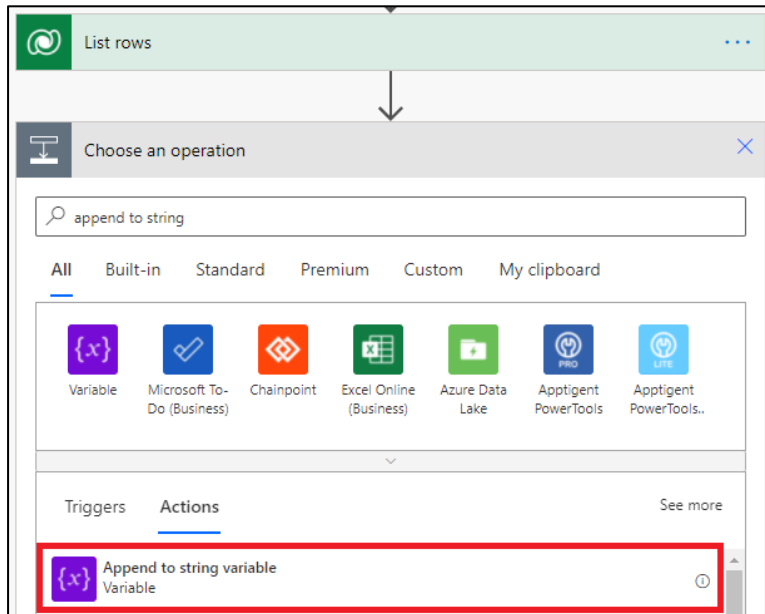
Filter rows will filter out only Registrations with email specified in flow inputs.

Sort By will sort Registrations rows in descending order by 'Created On'.

Expand Query gets the Name and IntranetURL of related TeamActivity, using 'relTeamActivity' relationship, that we created in Registrations table.

Row Count in combination with Sort By will return 5 most recent rows only.

10. Add an **Append to string variable** action from **Variable** 'connector' (use Search if this connector or action is not present)



11. Now it's time to append **List rows** query results to **BotResponse** variable that we've defined before.

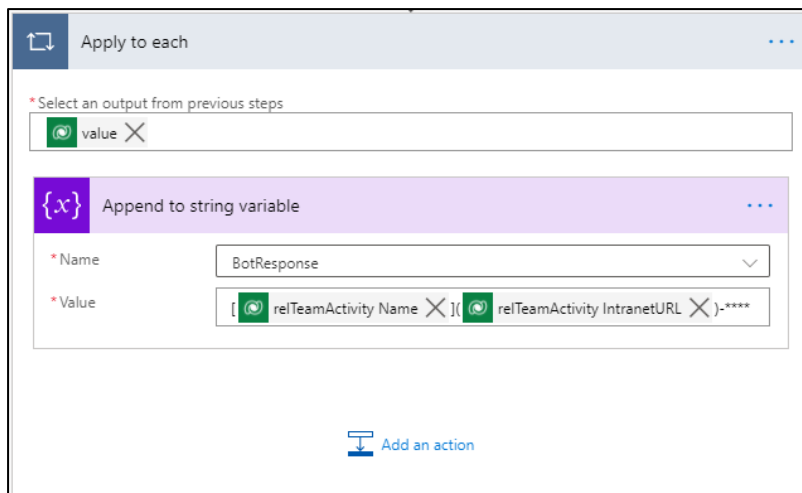
Set properties for **Append to string variable** action as:

Name: BotResponse

Value: [relTeamActivity Name](relTeamActivity IntranetURL) - ****
(property values from **List rows** action)
Press **Enter** twice to add two line breaks

Please, double-check syntax and make sure that all brackets are in place.

Apply to each loop will be added automatically, because List rows results may contain 0,1 or more rows.



12. Now we need to display **ApprovalStatus** of the registration, it's not as simple as it may look like.

ApprovalStatus property just holds enumeration (e.g., 0, 1, 2) of choices, not the labels of Choices.

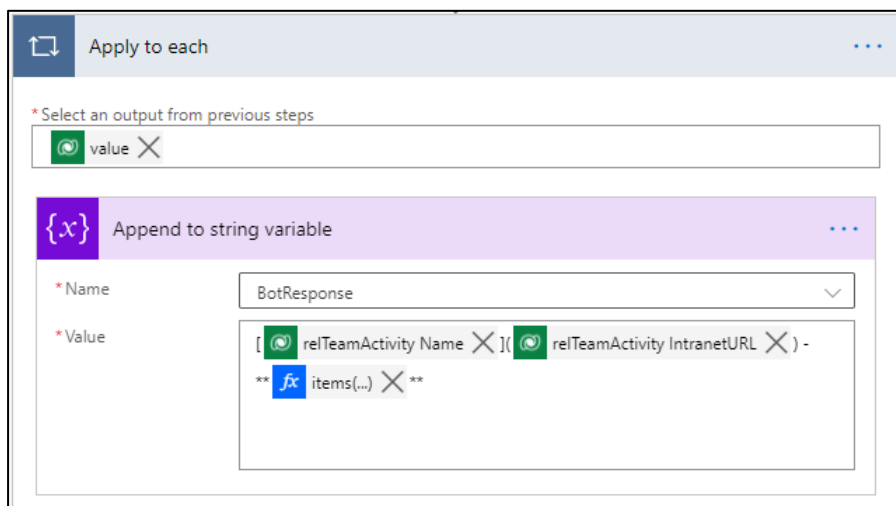
In order to display the label of Choices you need to use the expression:

```
items('Apply_to_each')?['crXXX_approvalstatus@OData.Community.Display.V1.FormattedValue']
```

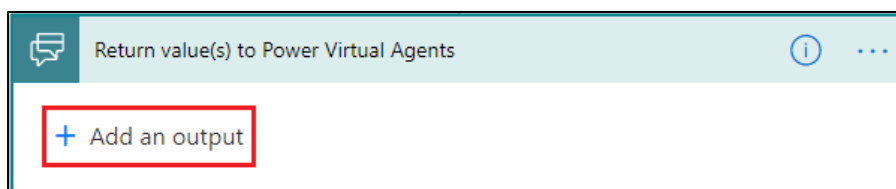
This bit `@OData.Community.Display.V1.FormattedValue` points flow at Approval Status label value.

Pay attention that you need to replace `crXXX` with your environments PublisherId prefix and make sure that loop is called Apply to each (or change this parameter in `items()`)

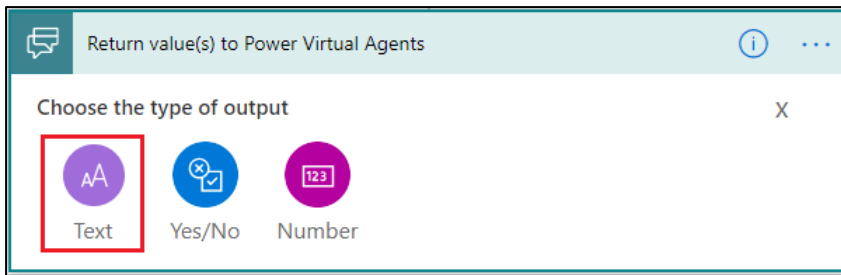
Result should look like this:



13. Expand existing **Return value(s) to Power Virtual Agents** action. Click **+Add an output**.

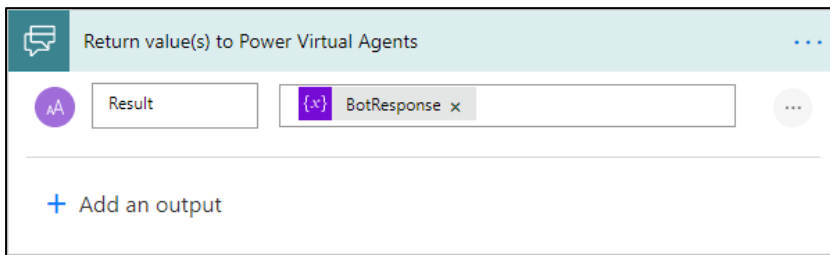


Select **Text** option by clicking on it

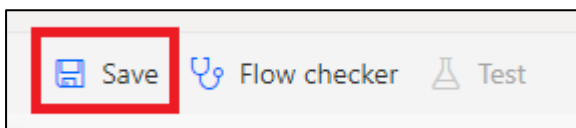


14. Set output name as **Result** and **BotResponse** variable as a value

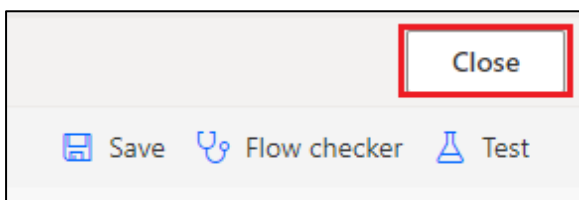
Note: *BotResponse is a variable, not a plain text*



15. Make sure, that there are no notifications in **Flow checker** and **Save** this flow

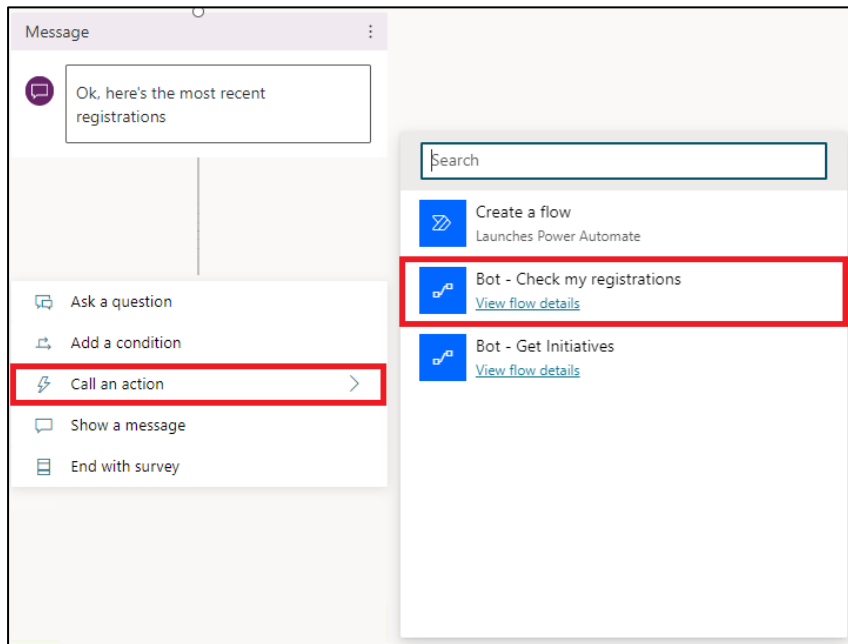


16. Click **Close** button in the top right corner after the flow is saved.



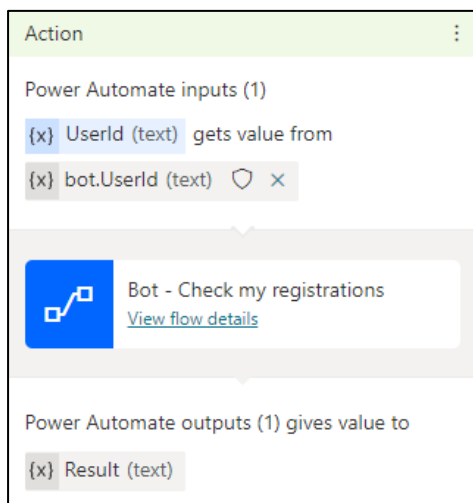
Redirect to the **Check my registrations** topic authoring canvas should happen, **if it didn't happen – if it didn't happen, then get there manually.**

17. Add **Call an action** node, select **Bot – Check my registrations** flow:

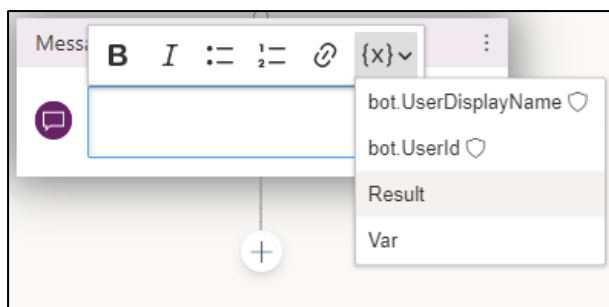


18. Action node should be displayed with the name of the flow and text variable **Result** as an output.

Set **bot.UserId** variable as an input parameter.



19. Add a **Message** node below and set **Result** variable as an output.



20. Using right side of a toolbar, make sure that **Topic checker** doesn't show any errors and **Save** the topic.

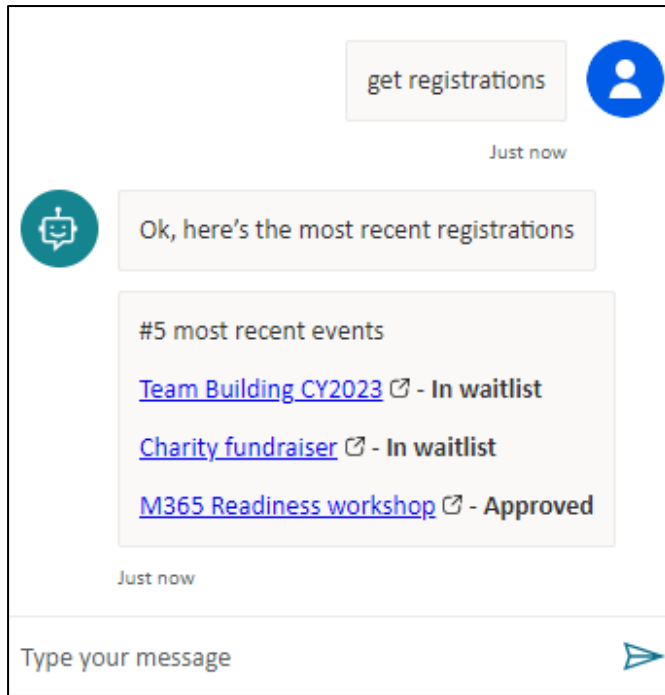


21. Let's test this topic once again.

Type *Get registrations* to the chat window in the left side of the screen and send the message.

Click **Yes** button or send positive response through the chat input.

Bot should return Initiative links with approval status like this:



Task is completed.