



# Microsoft Power Virtual Agents in a Day

Integration with Bot Framework Composer

Instructor Demo

May 2021

# Contents

<b>Power Virtual Agents .....</b>	<b>1</b>
<i>Demo Prerequisites.....</i>	<i>1</i>
<i>Goals for this lab .....</i>	<i>1</i>
<i>Scenario .....</i>	<i>1</i>
<i>Prerequisite: Install Bot Framework Composer .....</i>	<i>1</i>
<i>Task 1: Explain concept of Bot Framework Composer .....</i>	<i>2</i>
<i>Task 2: Create a new topic.....</i>	<i>2</i>
<i>Task 2: Create an adaptive card using Bot Framework Composer .....</i>	<i>4</i>
<i>Task 3: Add your composer content to Power Virtual Agents bot.....</i>	<i>14</i>
<i>Task 4: Test your bot.....</i>	<i>15</i>
<i>Lab survey.....</i>	<i>16</i>
<i>Terms of Use .....</i>	<i>17</i>

# Power Virtual Agents

This lab is subject to the Terms of Use on page 17 of this document.

## Demo Prerequisites

This demo requires you to be able to download Bot Framework Composer; if you are unable to download or install software on your machine, you will not be able to do this demo.

## Goals for this lab



After this demo you will be able to:

- Extend your chatbot using the Bot Framework Composer



The time to complete this demo is **[10]** minutes.

## Scenario

In this demo you will use the integration with Bot Framework Composer to add an Adaptive Card. Adaptive Cards enable the bot to display interactive elements like images, videos form and more.

## Prerequisite: Install Bot Framework Composer

**This task should be done before you start the demo.**

Note: This demo requires the instructor to download Bot Framework Composer; if you are unable to download or install software on your machine, you will not be able to do this demo.

1. Click on one of the following links to download the Bot Framework Composer on your device:  
Windows: <https://aka.ms/bf-composer-download-win>  
Mac: <https://aka.ms/bf-composer-download-mac>  
Linux: <https://aka.ms/bf-composer-download-linux>
2. Once the download is complete, launch the Bot Framework Composer Setup and go through the installation steps.
3. Once the installation is complete, close down Bot Framework Composer. It should not be open before you start the demo.

## Task 1: Explain concept of Bot Framework Composer

1. Explain that you have created your Power Virtual Agents chatbot, and now you want to enhance or extend it with custom content, such as adaptive dialogs, language generation, regular expressions, and adaptive cards.

The integration of Power Virtual Agents with Microsoft Bot Framework Composer enables you to create custom content and add it to Power Virtual Agents. The custom content you create in Bot Framework Composer is directly embeddable and executable from inside Power Virtual Agents. It means you can extend your bot without needing additional Azure hosting, deployment or billing complexities.

2. Explain that in this demo you will use the integration with Bot Framework Composer to add an Adaptive Card. Adaptive Cards enable the bot to display interactive elements like images, videos form and more.

## Task 2: Create a new topic

1. In Power Virtual Agents, create a new topic called **Meal delivery options**.
2. Add in the following trigger phrases  
what meals can I order  
what meal options do you have  
what dishes do you deliver  
entrees available for delivery

The screenshot shows the 'Meal delivery options' topic setup page in Power Virtual Agents. The 'Name' field is highlighted with a red box and contains 'Meal delivery options'. The 'Trigger phrases' section is also highlighted with a red box and contains four phrases: 'entrees available for delivery', 'what dishes do you deliver', 'what meal options do you have', and 'what meals can i order'. A 'Go to authoring canvas' button is visible on the right.

4. Save the topic then go to the authoring canvas for the topic.
5. Delete the blank message node under the trigger phrases on the authoring canvas.

6. Add a question node underneath the trigger phrases. Ask the question **What city are you in?**. Set the Identify field to City. For Save response as, set the name of the variable to **user\_city** and usage of the variable set to **Bot**.

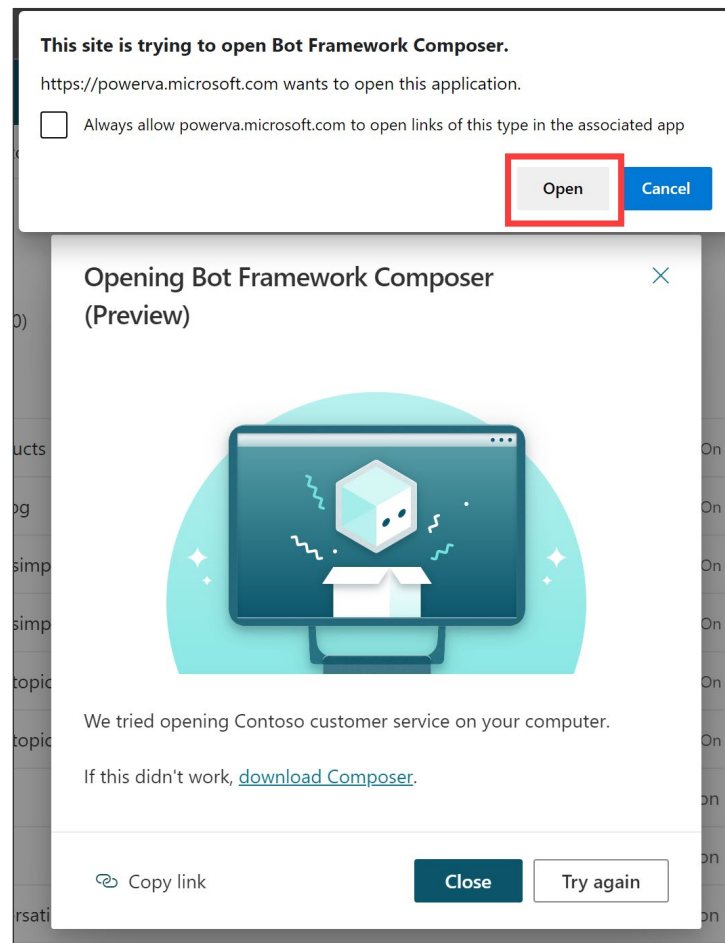
The screenshot shows the Bot Framework Composer interface for a topic named "Meal delivery options". The main canvas displays a flowchart with a "Trigger Phrases (4)" node at the top, followed by a "Question" node. The "Question" node is configured with the text "What city are you in?". Below the question, the "Identify" field is set to "City". The "Save response as" field is set to "{x} bot.user\_city (city)". The "Variable Properties" pane on the right shows the variable name "bot. user\_city", the type "City", and the source "Go to Source". Under the "Usage" section, the "Bot (any topic can access)" option is selected, and the "External sources can set values" checkbox is unchecked.

7. Explain that the variable just created will be used in the adaptive card. For the variable to be used within Bot Framework Composer, it must be set with Bot scope rather than Topic scope.
8. Save the topic, then return to the list of topics.
9. Click **Open in Bot Framework**.

The screenshot shows the "Power Virtual Agents | Contoso customer service" interface. The left sidebar contains navigation links for Home, Topics, Entities, and Analytics. The main area displays the "Topics" list, which is divided into "Existing (14)" and "Suggested (0)". The "Existing (14)" section shows a table with columns for "Type", "Name", and "Trigger phrases". The "Open in Bot Framework (Preview)" button is highlighted with a red box.

10. Now we will enhance the topic by using Bot Framework Composer. When prompted, click to open the Bot Framework Composer.

*Note: If you can't launch composer automatically from Power Virtual Agents, you can launch it manually from your browser. Click on Copy link to copy the link and paste it into the URL in your web browser.*



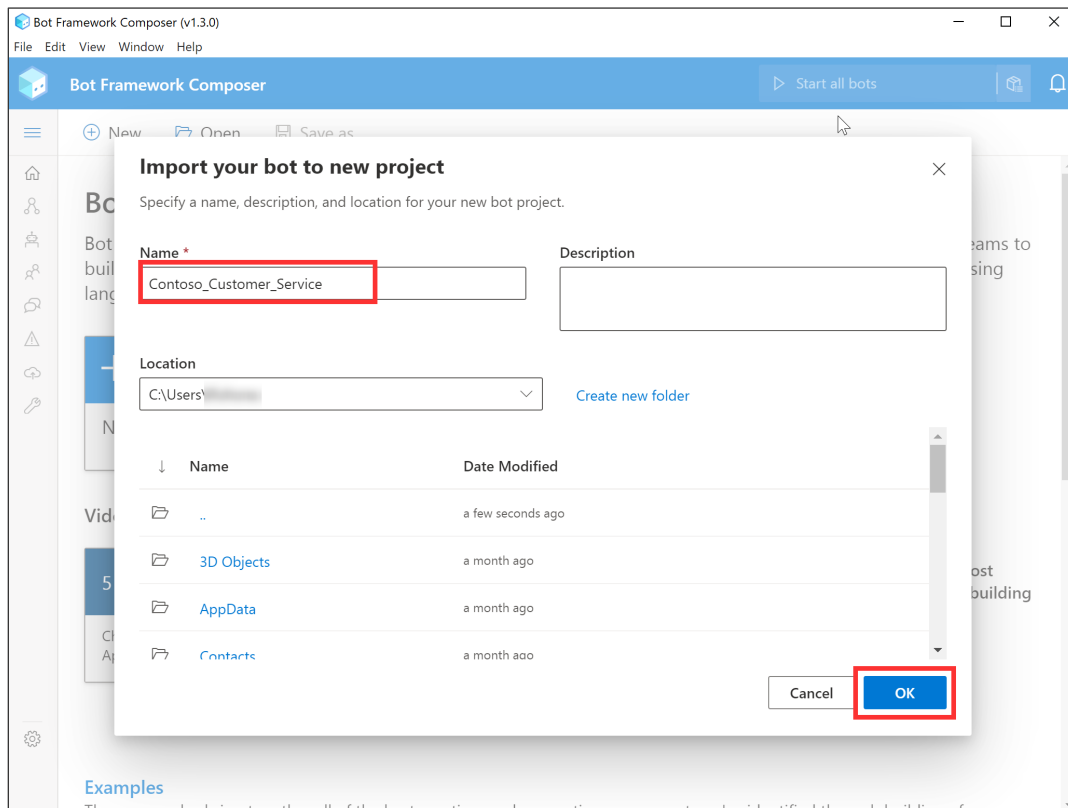
11. Explain that it is important to open Bot Framework Composer from within Power Virtual Agents if you plan to create content for Power Virtual Agents bots. This ensures that Composer has all the necessary plugins needed for integration with Power Virtual Agents.

## Task 2: Create an adaptive card using Bot Framework Composer

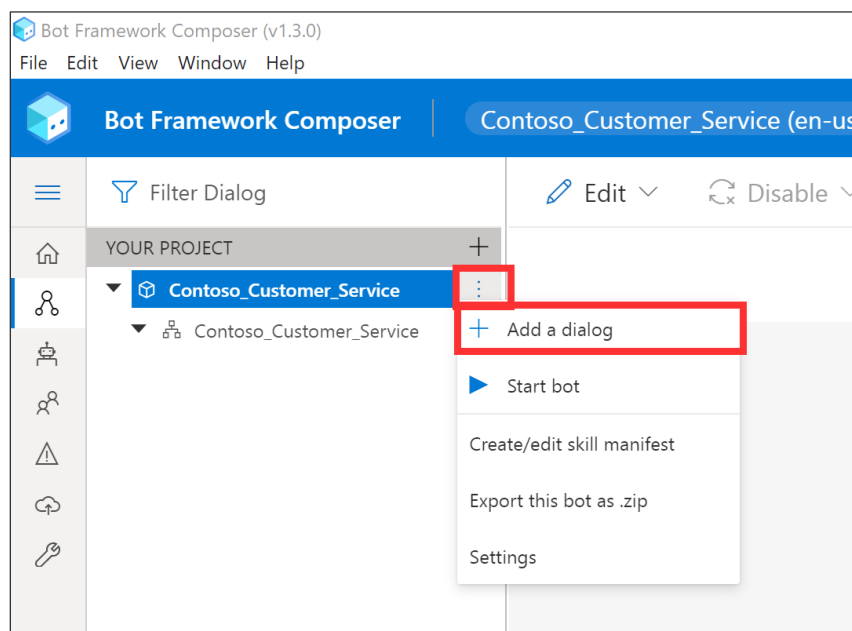
1. When Bot Framework Composer application launches, you will be prompted to sign in. Sign in with the account that contains your bot.

- Once you sign in, you will be asked to import your bot to a new project. Fill in the name field with the name of your bot, without any spaces. E.g. if your bot is called *Contoso Customer Service*, in the name field enter **Contoso\_Customer\_Service**. Then, click **OK**.

*Note: In the occasion where the following screen does not appear in Composer on launch, you will need to close the application and launch it again from Power Virtual Agents.*



- You will then see your project appear in Composer. On the left side of the screen, click on the actions where shown below. Then, click on **+ Add a dialog**.



4. Specify the name of your new dialog as **Meals**, then click **OK**.

**Create a dialog** ✕

Specify a name and description for your new dialog.

**Name \***

Meals

**Description**

Cancel OK

Select a trigger on the left

5. From the left navigation menu, click on the Bot Responses tab. With the Meals dialogue selected click Show code.

Bot Framework Composer | Contoso\_Customer\_Service (en-us)

Home Design **Bot Responses** User Input Diagnostics Publish Project Settings

**Bot Responses**

Filter Dialog

YOUR PROJECT

- Contoso\_Customer\_Ser...
- Common
- Contoso\_Customer\_Ser...
- Meals**

New template

Name	Responses	Been used
Show code		

6. Explain that this is where you will enter in the json text to create the adaptive card.



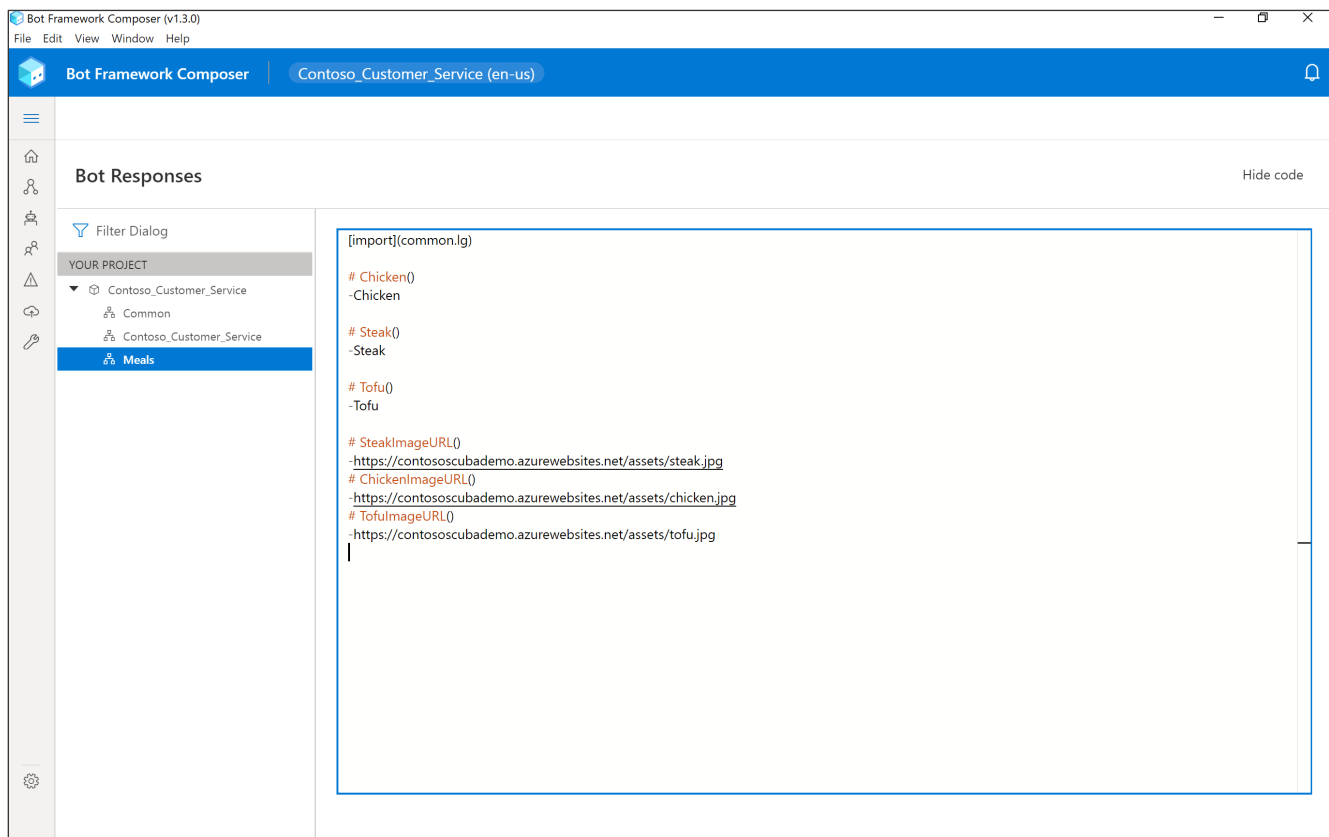
7. First, paste in the following LG (Language Generation) Template. Explain that this template sets out the names of the meals and their related image that will be displayed in the adaptive card.

```
# Chicken()
-Chicken

# Steak()
-Steak

# Tofu()
-Tofu

# SteakImageURL()
-https://contososcubademo.azurewebsites.net/assets/steak.jpg
# ChickenImageURL()
-https://contososcubademo.azurewebsites.net/assets/chicken.jpg
# TofuImageURL()
-https://contososcubademo.azurewebsites.net/assets/tofu.jpg
```



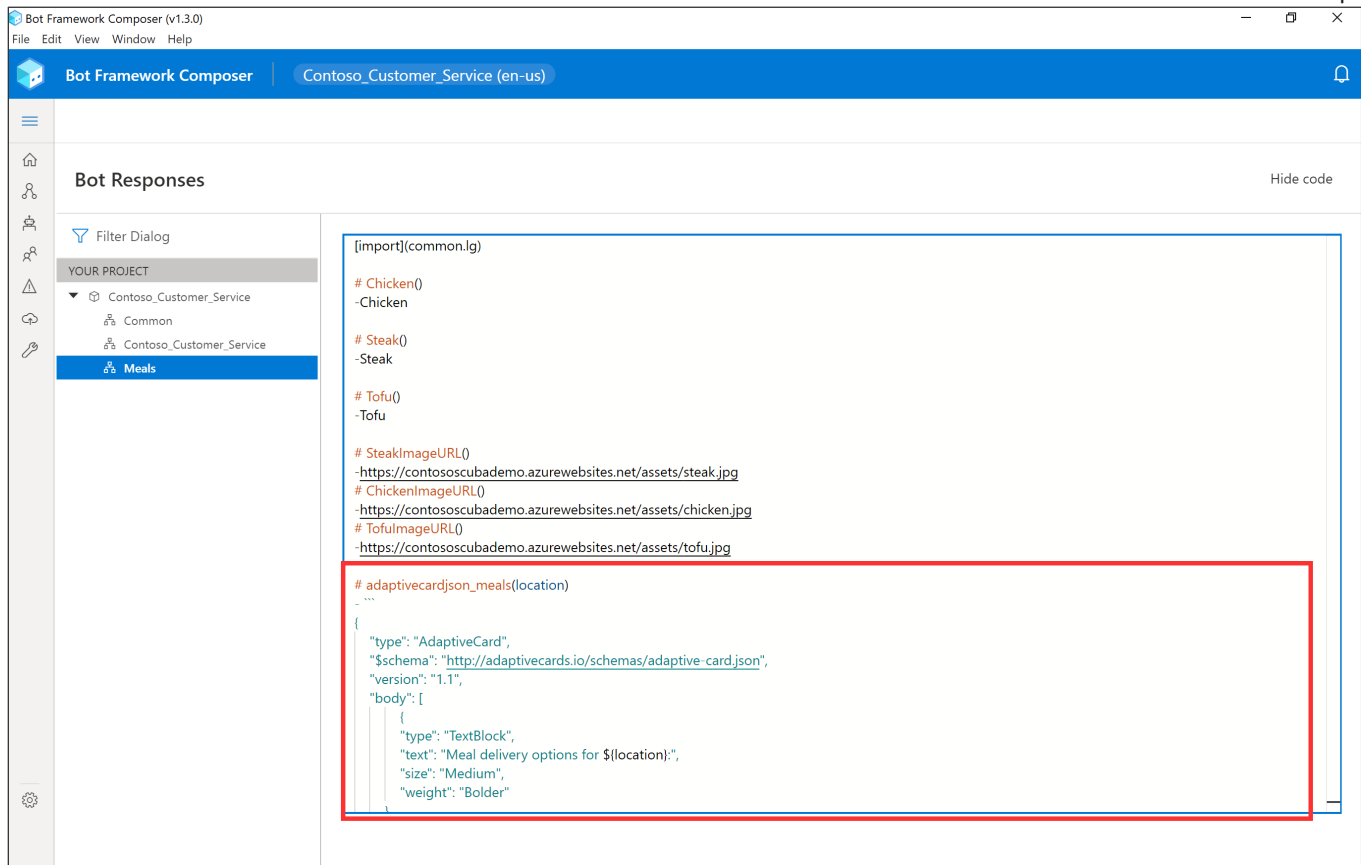
8. Next, paste in the following JSON code into the same window, underneath the code entered in step 7.

```
# adaptivecardjson_meals(location)
- ``
{
  "type": "AdaptiveCard",
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "version": "1.1",
  "body": [
    {
      "type": "TextBlock",
      "text": "Meal delivery options for ${location}:",
      "size": "Medium",
      "weight": "Bolder"
    },
    {
      "type": "ColumnSet",
      "columns": [
        {
          "type": "Column",
          "width": "stretch",
          "items": [
            {
              "type": "Image",
              "url": "${SteakImageURL()}",
              "size": "Stretch",
              "spacing": "Medium",
              "horizontalAlignment": "Center"
            }
          ]
        },
        {
          "type": "Column",
          "width": "stretch",
          "items": [
            {
              "type": "Image",
              "url": "${ChickenImageURL()}",
              "horizontalAlignment": "Center"
            }
          ]
        },
        {
          "type": "Column",
          "width": "stretch",
          "items": [
            {
              "type": "Image",
              "url": "${TofuImageURL()}",
              "horizontalAlignment": "Center"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ]
},
{
  "type": "ColumnSet",
  "columns": [
    {
      "type": "Column",
      "width": "stretch",
      "items": [
        {
          "type": "TextBlock",
          "text": "${Steak()}",
          "wrap": true,
          "horizontalAlignment": "Center"
        }
      ]
    },
    {
      "type": "Column",
      "width": "stretch",
      "items": [
        {
          "type": "TextBlock",
          "text": "${Chicken()}",
          "wrap": true,
          "horizontalAlignment": "Center"
        }
      ]
    },
    {
      "type": "Column",
      "width": "stretch",
      "items": [
        {
          "type": "TextBlock",
          "text": "${Tofu()}",
          "wrap": true,
          "horizontalAlignment": "Center"
        }
      ]
    }
  ]
}
]
}
...

```



The screenshot displays the Bot Framework Composer (v1.3.0) interface. The top bar shows the application name and the current project, 'Contoso\_Customer\_Service (en-us)'. The left sidebar contains a 'Filter Dialog' and a 'YOUR PROJECT' tree with the following structure:

- Contoso\_Customer\_Service
  - Common
  - Contoso\_Customer\_Service
  - Meals**

The main area is titled 'Bot Responses' and contains a code editor. The code defines the 'Meals' response, which includes image URLs and an adaptive card. The adaptive card is highlighted with a red border. The code is as follows:

```
[import](common.lg)

# Chicken()
-Chicken

# Steak()
-Steak

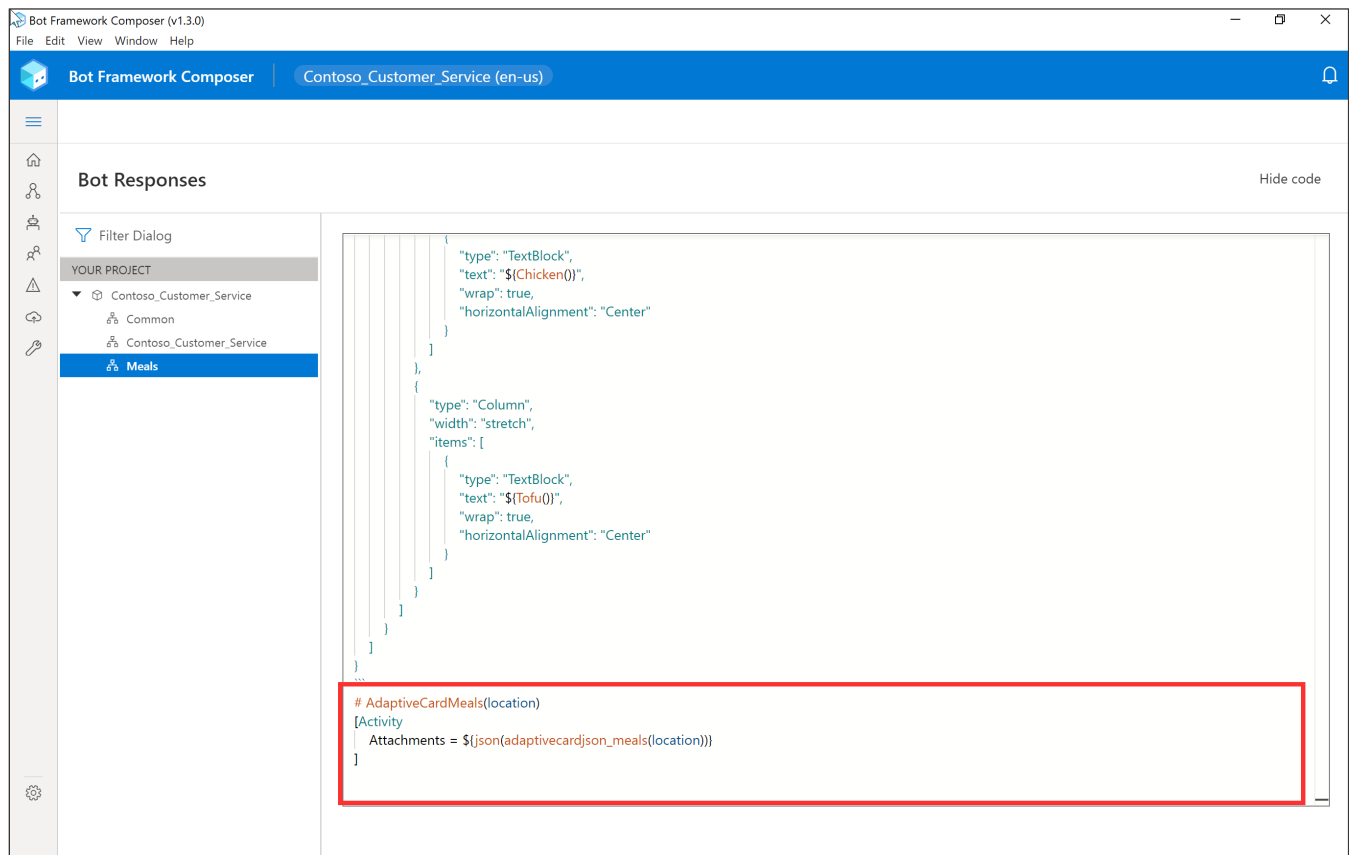
# Tofu()
-Tofu

# SteakImageUrl()
-https://contososcubademo.azurewebsites.net/assets/steak.jpg
# ChickenImageUrl()
-https://contososcubademo.azurewebsites.net/assets/chicken.jpg
# TofuImageUrl()
-https://contososcubademo.azurewebsites.net/assets/tofu.jpg

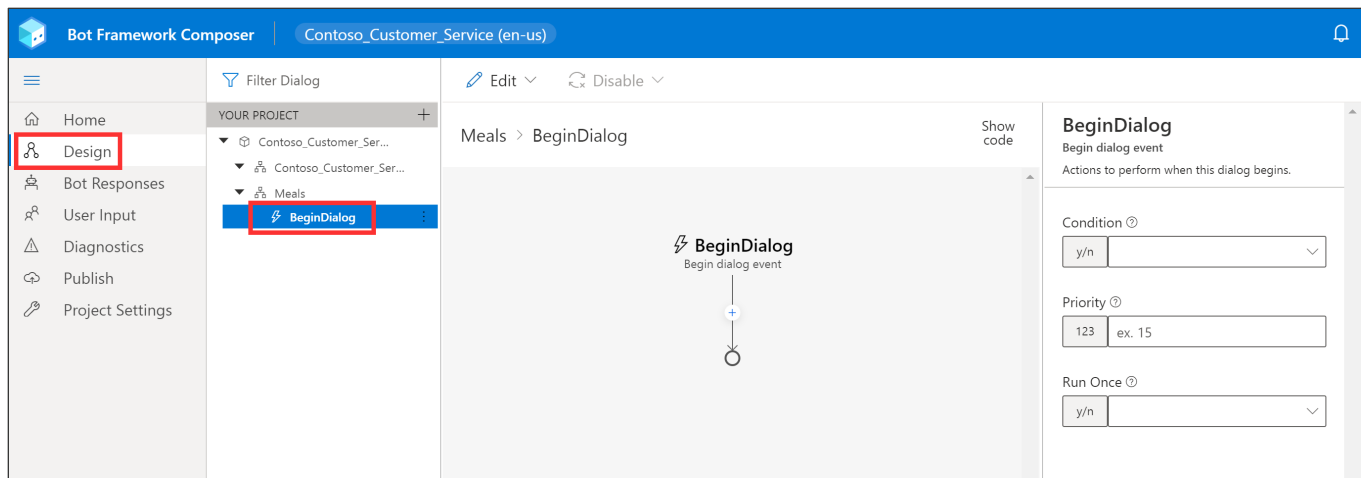
# adaptivecardjson_meals(location)
- ""
{
  "type": "AdaptiveCard",
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "version": "1.1",
  "body": [
    {
      "type": "TextBlock",
      "text": "Meal delivery options for $(location);",
      "size": "Medium",
      "weight": "Bolder"
    }
  ]
}
```

9. The last code to enter is the activity that will display this adaptive card in the Bot Framework dialog. Paste in the following code in the same window in composer, underneath the code entered in step 8.

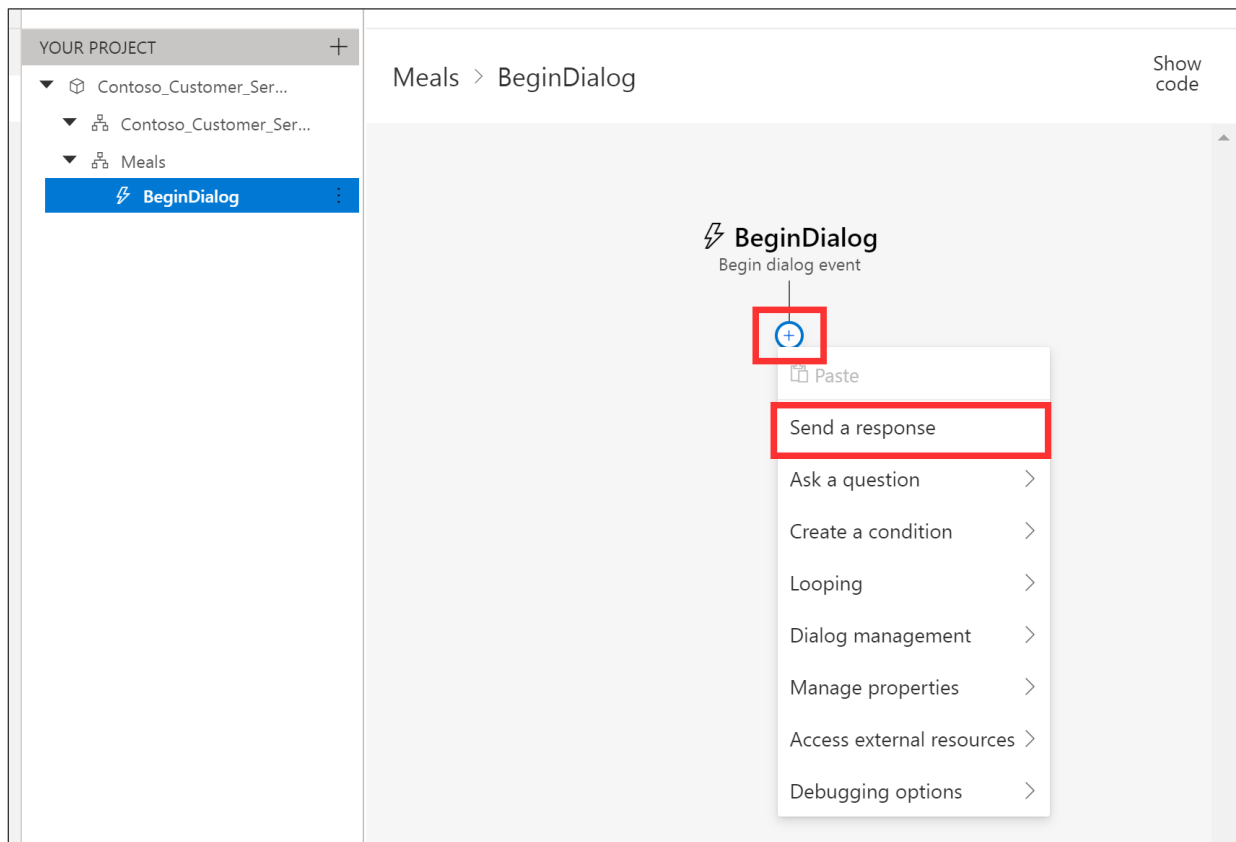
```
# AdaptiveCardMeals(location)
[Activity
  Attachments = ${json(adaptivecardjson_meals(location))}
]
```



10. From the left navigation bar, click on the **Design** tab and select **BeginDialog**.

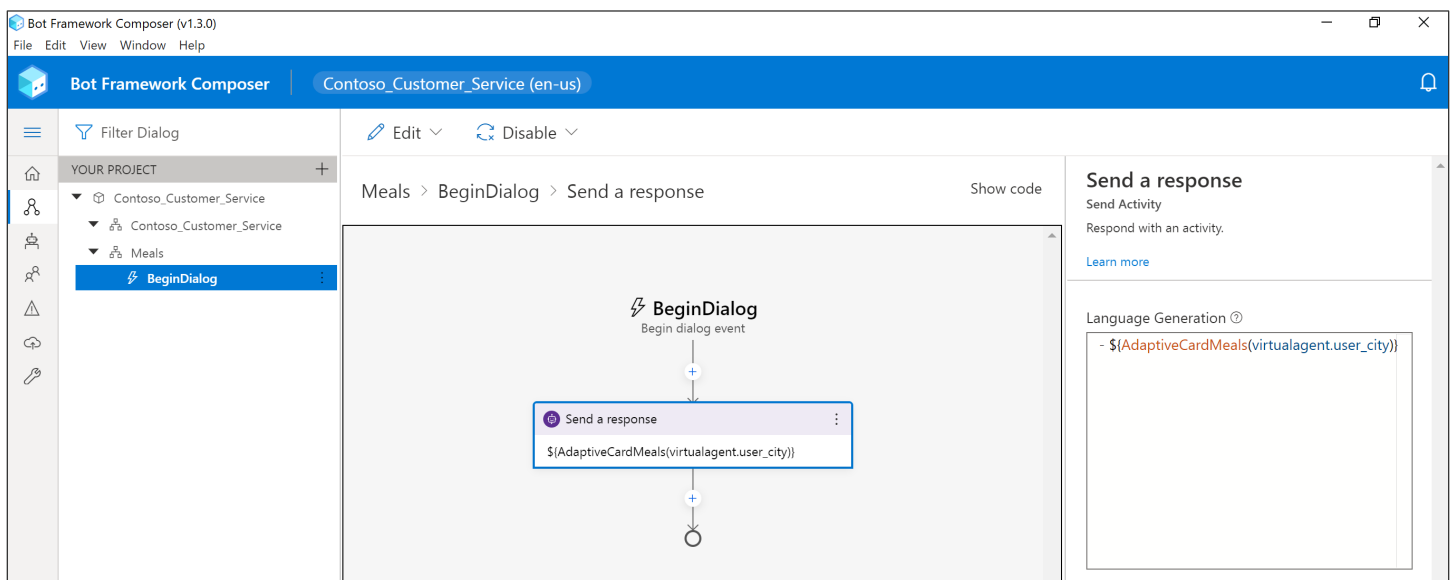


11. Click on the + button indicated below, and select **Send a response**.



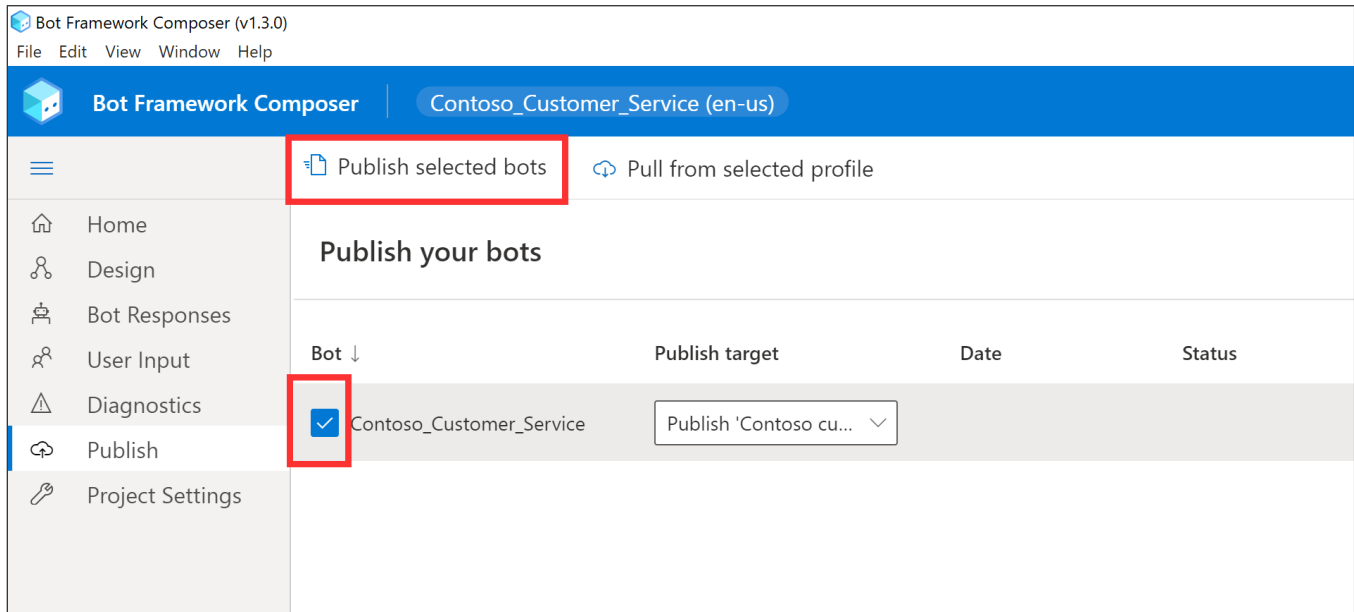
12. Explain that here, you will reference the adaptive card created earlier as well as the city indicated by the user while interacting with the bot. In the Language Generation box that appears on the right side of the screen, enter in the following JSON code. *Note: ensure there is only one – symbol at the start of the code.*

```
- ${AdaptiveCardMeals(virtualagent.user_city)}
```



13. Click **Publish** on the left navigation menu.

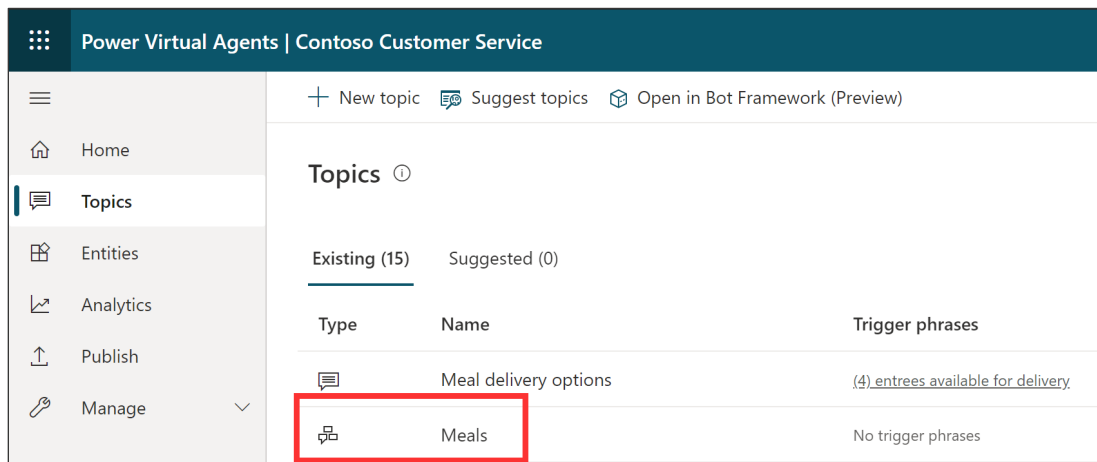
14. Select your bot, then click on Publish selected bots. Confirm that you want to publish the bot. The bot should now be in the process of publishing.



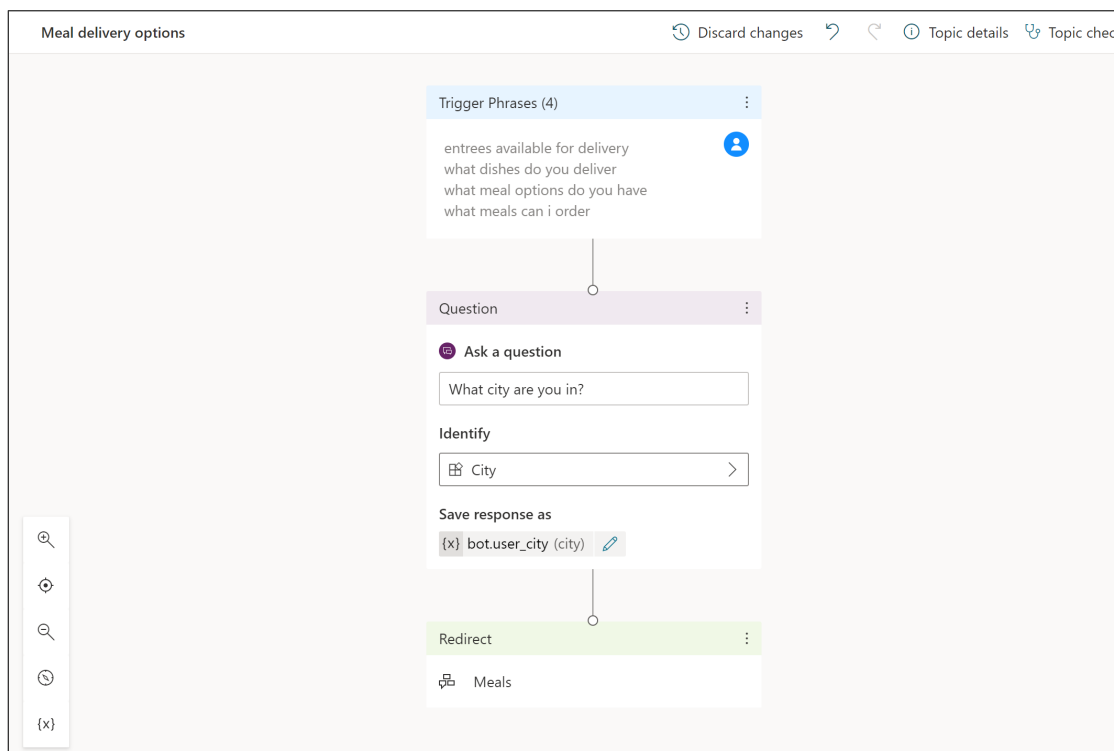
15. You will get a confirmation message pop up in Composer once the bot has been published.

## Task 3: Add your composer content to Power Virtual Agents bot

1. Go back to Power Virtual Agents.
2. Point out that you can see the adaptive card action you just published from Composer is visible in your list of topics. *Note, you may need to refresh the screen to see it.*



3. Open your topic **Meal delivery options** and open the authoring canvas.
4. Under the question/variable, add a node and select **Go to another topic**. Choose the composer topic you just created, which will now be in the list. Your Power Virtual Agents bot will call the Bot Framework dialog to display an adaptive card.

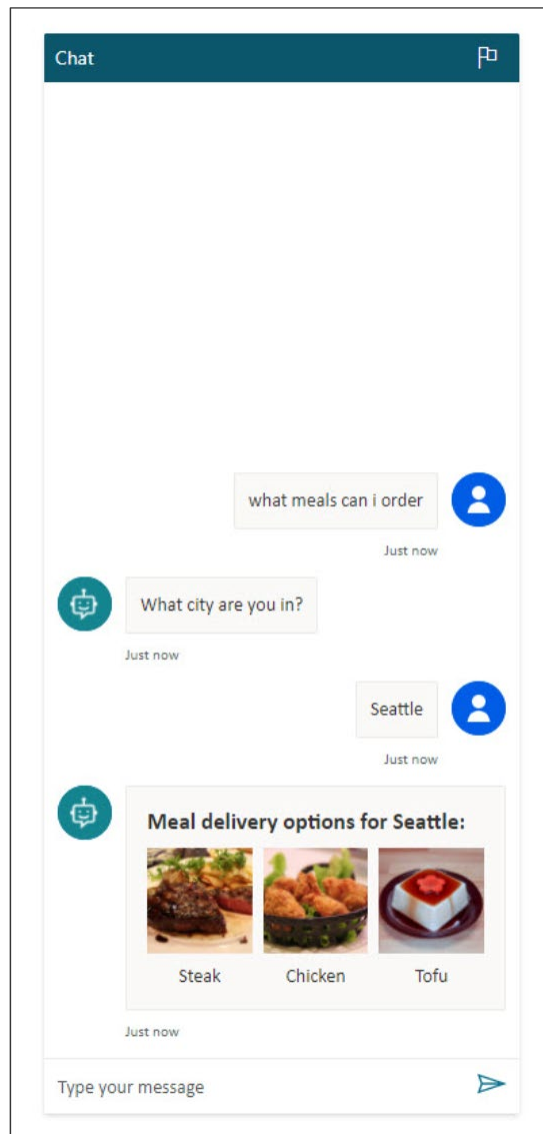


5. Save your topic.



## Task 4: Test your bot

1. Go to the test pane and switch on Track between topics.
2. Type in a trigger phrase (e.g. "what meals can I order"), and the name of a city when prompted
3. Your chatbot should display images of the meals in an adaptive card as show below.



## Lab survey

We would appreciate your feedback on Power Virtual Agents and on this demo, 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.