



Microsoft Cloud for Healthcare **in a Day**

Lab 04: Azure Health Bot

Step-by-Step Lab

April 2021

Contents

Overview.....	3
Prerequisites	3
Azure Portal.....	3
Industry Prioritized Scenarios.....	4
Recommended Resources.....	4
Goals.....	4
Exercise 1: Set up Azure Health Bot.....	5
Task 1: Install Azure Health Bot in Azure Subscription.....	5
Task 2: Update Health bot settings to enable Dynamics 365 Integration.....	9
Exercise 2: Obtain Azure AD Application Id	11
Exercise 3: Configure Customer Service Omnichannel Live chat	13
Task 1: Create Health bot User in Dynamics 365 Customer Service.....	13
Task 2: Create and Configure Omnichannel Queues	18
Task 3: Update Live Work Stream with Context Variables and Routing Rules	22
Task 4: Create Chat Widget for Health bot	24
Exercise 4: Embed Health bot in Power Apps Portal	26
Exercise 5: Extend Azure Health Bot with Custom Scenarios	29
Task 1: Navigate to Health Bot and Create Scenarios.....	30
Task 2: Test your Health Bot Escalation from Power Apps Portal to Dynamics 365 Omnichannel	51
Summary	55
Lab Survey.....	56
Terms of Use.....	57

Overview

Prerequisites

Note: If you are in an official training, the environment has been set up and provided to you. There is no further action required by you in the prerequisite section.

This is the **fourth** lab in the series covering the Microsoft Cloud for Healthcare. The assumption is you have successfully completed the previous two labs. If you have not, you must have at least completed setting up an environment.

If you have not completed the environment, please reference the following two links:

- [Set up/configure Microsoft Cloud for Healthcare](#)
- [Microsoft Cloud for Healthcare Licensing](#)

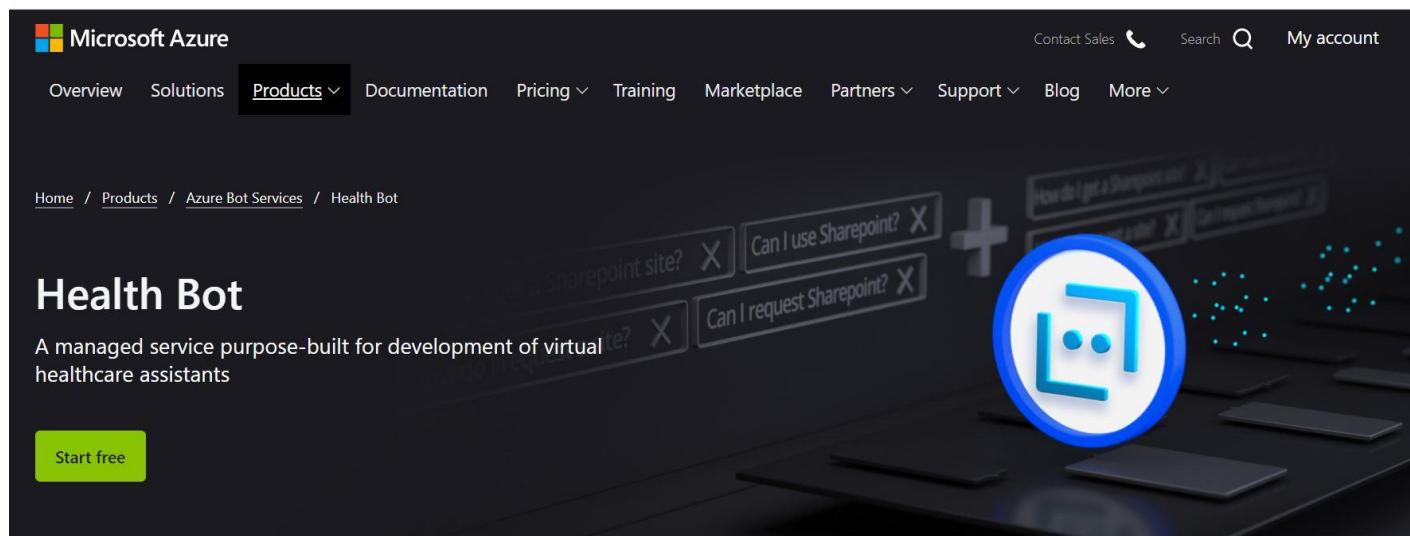
An **Azure Subscription** with **Administrator access** is also required for the Health Bot Service in this lab.

The following applications and services must be installed and accessible for this lab:

- Dynamics 365 Customer Service for Omnichannel
- Power Apps Customer Self-service Portal

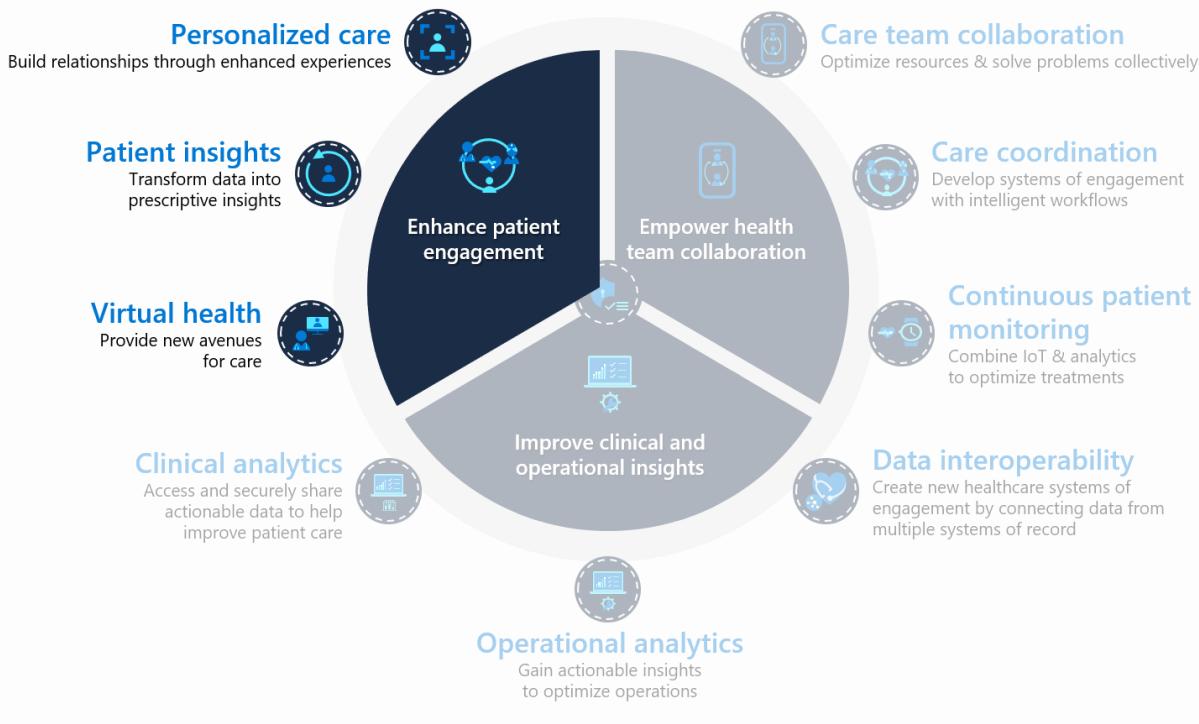
Microsoft Azure Portal

This lab focuses on the **Azure Health Bot** managed service that allows for development of virtual healthcare assistants. Azure Health Bot instances can be created in the Azure Portal in your tenant.



Industry Prioritized Scenarios

The Azure Health Bot focuses on the **Enhance patient engagement** priority scenario by creating a virtual health option to allow for new avenues of care with embedded insights.



Recommended Resources

- [Built-in Bot Scenarios](#)
- [Custom Scenarios](#)
- [Work with queues in Omnichannel for Customer Service](#)

Goals

After this lesson you will be able to:



- Install and configure Microsoft Health Bot Service
- Configure Live Chat for Dynamics 365 Omnichannel for Customer Service
- Embed Health bot in Power Apps Portal



The estimated time to complete this lab is **50** minutes.

Exercise 1: Set up Azure Health Bot

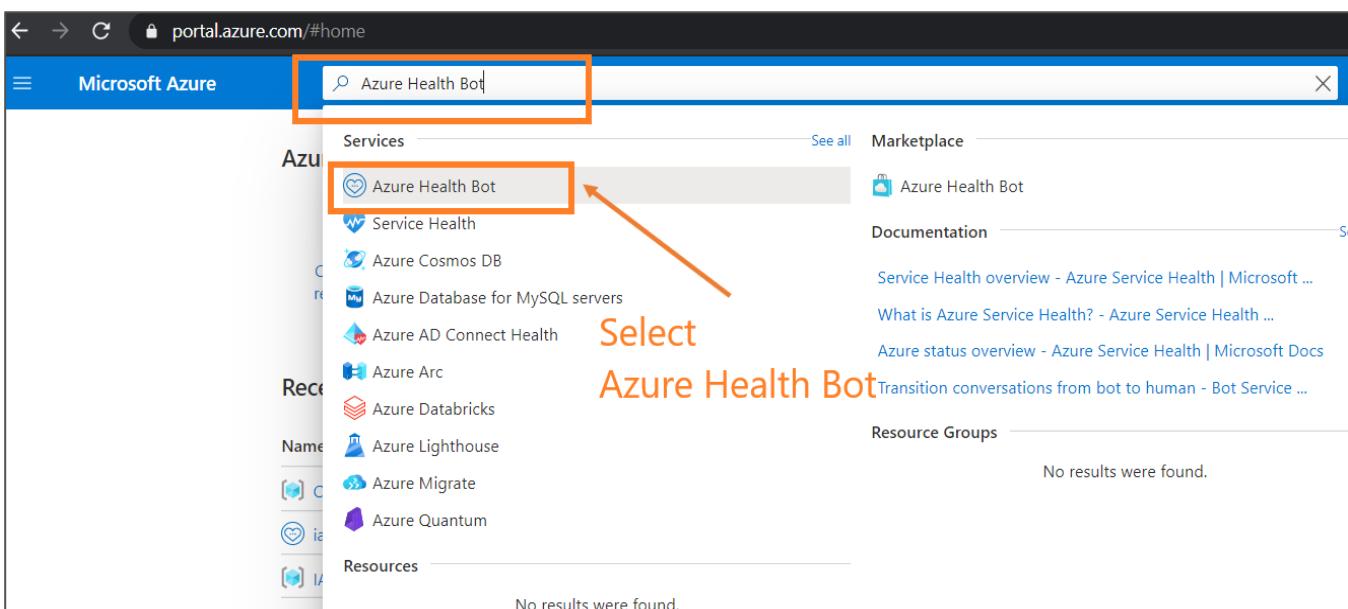
In this exercise, you will go through the following:

1. Setup health bot from Azure Portal.
2. Configure and enable the integration between Dynamics 365 Omnichannel and Health Bot.
3. Configure and enable Bot channel to obtain a Bot Id.

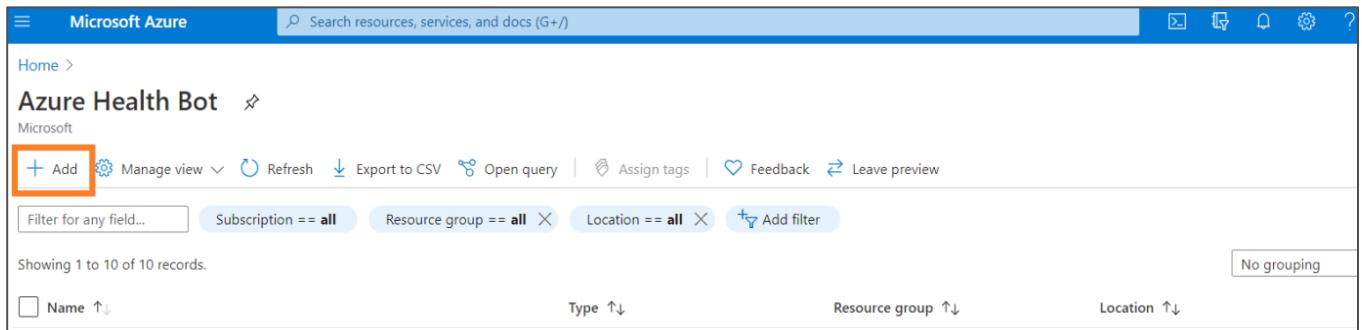
Azure Health Bot empowers developers in healthcare organizations to build and deploy AI-powered, compliant, conversational healthcare experiences at scale. It combines built-in medical database with natural language capabilities to understand clinical terminology and can be easily customized to support your organization's clinical use cases. The service ensures alignment with industry compliance requirements and is privacy protected to HIPAA standards.

Task 1: Install Azure Health Bot in Azure Subscription

1. Please open browser with **InPrivate** window for **Edge** browser/**Incognito** for **Chrome** browser
2. Go to Azure Portal (<https://portal.azure.com/>) Sign in using the Tenant email account.
3. Azure Portal >> Search Box >> Type >> **Azure Health Bot**
4. Select "**Azure Health Bot**" from the search result (refer below picture)

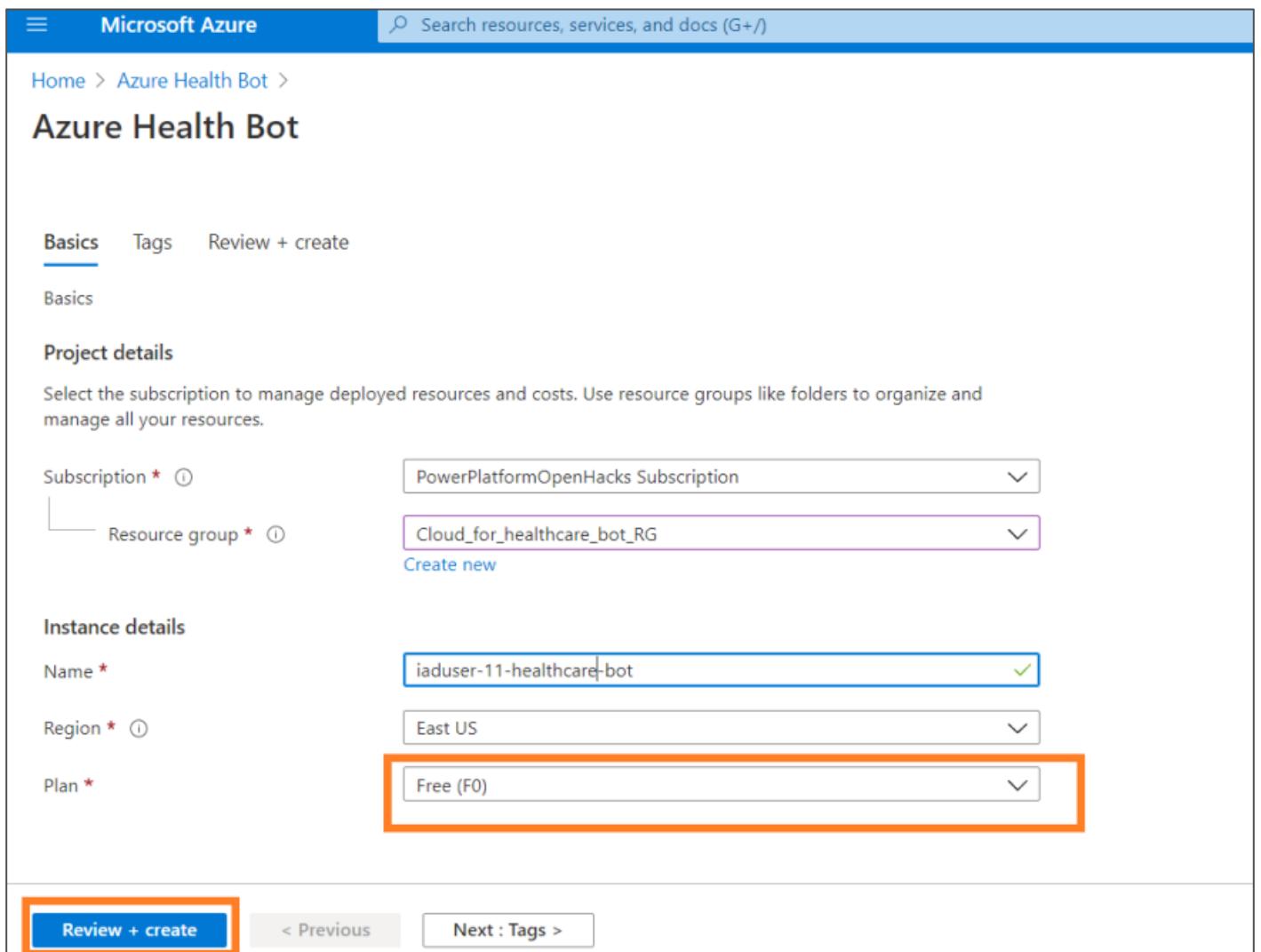


5. Create "**Azure Health Bot**": Click **Add** button to navigate to health bot creation step (refer below picture)



The screenshot shows the Microsoft Azure portal with the search bar at the top. Below it, the 'Azure Health Bot' page is displayed under the 'Microsoft' category. At the top of the page, there are several buttons: '+ Add' (highlighted with a red box), 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', 'Feedback', and 'Leave preview'. Below these buttons are filter options: 'Subscription == all', 'Resource group == all', 'Location == all', and 'Add filter'. The main content area shows a message 'Showing 1 to 10 of 10 records.' and a table with columns: Name (sorted ascending), Type (sorted ascending), Resource group (sorted ascending), and Location (sorted descending). A 'No grouping' button is also present.

6. You will be redirected to Azure Health Bot page. On the **Azure Health Bot** page, keep the plan as **Free (F0)** and click **Review + Create** to continue. (refer below picture)



The screenshot shows the 'Azure Health Bot' creation page. At the top, there are tabs: 'Basics' (highlighted with a blue underline), 'Tags', and 'Review + create'. The 'Basics' section contains fields for 'Subscription' (set to 'PowerPlatformOpenHacks Subscription') and 'Resource group' (set to 'Cloud_for_healthcare_bot_RG'). In the 'Instance details' section, the 'Name' field is set to 'iaduser-11-healthcare-bot', 'Region' is set to 'East US', and the 'Plan' dropdown is set to 'Free (F0)'. The 'Review + create' button at the bottom left is highlighted with a red box. Navigation buttons '**< Previous**' and '**Next : Tags >**' are also visible.

7. You will be redirected to Azure Health Bot – Validation page. On the **Azure Health Bot** – Validation page, will show the Subscription, Resource Group etc. when you see Create button is enabled, click “Create” button. (Refer below picture)

Note: Please wait until **Create** button is enable as it will take few seconds to run the backend process and then Create button will be enabled.

The screenshot shows the Azure Health Bot validation page. At the top, there's a green banner indicating "Validation Passed". Below it, there are tabs for "Basics", "Tags", and "Review + create", with "Review + create" being the active tab. The "TERMS" section contains a detailed legal agreement. The "Basics" section lists configuration details:

Subscription	PowerPlatformOpenHacks Subscription
Resource group	Cloud_for_healthcare_bot_RG
Name	iaduser-11-healthcare-bot
Region	East US
Plan	Free (F0)

At the bottom, there are buttons for "Create" (which is highlighted with an orange border), "< Previous", "Next", and "Download a template for automation".

8. You will be redirected to Azure Health Bot – Deployment page. On the **Azure Health Bot** – Validation page, will show the Subscription, Resource Group etc. when you see Create button is enabled, click “Create” button. (Refer below picture)

The screenshot shows the Azure Health Bot - Deployment page. At the top, there's a search bar, a delete button, a cancel button, a redeploy button, and a refresh button. A feedback link is present. The main area displays a message: "... Deployment is in progress". Below this, deployment details are shown: Deployment name: Microsoft.HealthBot-20210126212753, Subscription: PowerPlatformOpenHacks Subscription, Resource group: Cloud_for_healthcare_bot_RG. To the right, start time and correlation ID are listed. A deployment details section with a download link is also visible. A table at the bottom lists a single resource: iaduser-11-healthcare-bot, Type: Microsoft.HealthBot/healthBots, Status: Created, with an operation details link.

9. Please wait until deployment is complete for Azure Health Bot – Deployment page. On the **Azure Health Bot** – Deployment page, when you see “**Go to resource**” button is enabled, click “**Go to resource**” button. (Refer below picture)

The screenshot shows the Azure Health Bot - Deployment page after deployment is complete. The main message is "Your deployment is complete". Deployment details are identical to the previous screenshot. Below the deployment details, there are sections for "Deployment details (Download)" and "Next steps". The "Next steps" section contains a prominent blue "Go to resource" button, which is highlighted with an orange box.

10. You will be redirected to your “Azure Health Bot” – resource page. On the **Azure Health Bot** – resource page, Click **Azure Health Bot – Instance link** to navigate to Azure Health bot Instance Scenario page (Designer page) (Refer below picture)

Note: Please copy this Management Portal Link and store it for later use.

The screenshot shows the Azure Health Bot - instance page for the resource iaduser-11-healthcare-bot. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Settings, Properties, and Locks. The main pane shows deployment details: Resource group (change) : Cloud_for_healthcare_bot_RG, Location : East US, Subscription (change) : PowerPlatformOpenHacks Subscription, Subscription ID : ..., Tags (change) : Click here to add tags. To the right, there's a callout box with the text "Click below link to open healthbot instance" and a highlighted box containing the management portal URL: Plan : F0, Management portal : <https://us.healthbot.microsoft.com/account/iaduser-11-healthcare-bot-ijp...>.

Congratulations! You have successfully installed the Health Bot. Later you will be creating scenarios to design your bot to talk to customer and Live Agent.

Task 2: Update Health bot settings to enable Dynamics 365 Integration

1. **Expand** side navigation bar to view Azure Health Bot >> Configuration >> Conversation Settings.

The screenshot shows the Azure Health Bot instance interface. The left sidebar is expanded, revealing various navigation icons. A red box highlights the 'More' icon (three horizontal lines) in the sidebar. An orange arrow points from this icon to the text 'Click to expand' located above the main content area. The main content area features a large 'Welcome to your Health Bot Instance' message with a 'Click to expand' button. Below this, there are three main sections: 'Select a template scenario' (with a grid icon), 'Create a new scenario' (with a branching icon), and 'Configure built-in capabilities' (with a gear icon). Each section includes a brief description and a blue 'Manage from configuration section' link.

2. Enable Bot for **Dynamics 365 Omnichannel for Customer Service**
 - a. Navigate to **Configuration > Conversation**.
 - b. **Bridge Messages** should be enabled, Select **Human Handoff** > "**Enabled**" **Bridge Messages** under **Dynamics 365 Omnichannel**.
 - c. Click **Save**.

Note: By default, **Human Handoff** >> **Bridge Messages** would be **disabled**, please click **Enabled** option, which would allow Dynamics 365 Omnichannel to connect to Health bot.

The screenshot shows the 'Health Bot Service' interface. On the left, a navigation sidebar lists 'Scenarios', 'Language', 'Configuration' (which is selected and highlighted with an orange box), 'Integration', 'Analytics', 'Users', and 'Resources'. The main content area has tabs for 'Interactions', 'Navigation', 'Cognitive Services', and 'Human Handoff' (which is also highlighted with an orange box). Under 'Human Handoff', there's a message box stating 'Sorry, no agents are currently available.' Below it, under 'Agent Authentication (Microsoft Teams)', there are four input fields: 'Active Directory Tenant ID', 'Azure Active Directory Group Object ID', 'Application (client) ID', and 'Application (client) Secret'. Under 'Online Meetings (Microsoft Teams)', there's one input field for 'Default Meeting Organizer Object ID (optional)'. At the bottom, there's a section for 'Dynamics 365 OmniChannel' with a 'Bridge Messages' toggle switch set to 'Enabled'.

3. Enable Bot for **Microsoft Teams** Channel
 - a. Navigate to **Integration > Channels**
 - b. Enable **Microsoft Teams**
 - c. Obtain **BotId** and store it for later use

Note: This **BotId** is will be used when **Dynamics 365 Application User** getting created in the later part of the Task.

The screenshot shows the 'Channels' page under the 'Integration' section. It lists various channels: Web Chat, DirectLine, Microsoft Teams (which is highlighted with an orange box and has its toggle switch set to 'Enabled'), Twilio, Facebook, Telegram, and Alexa (preview). A callout bubble points to the Microsoft Teams row with the text 'Enable and copy BotId'. Below the table, a note says 'Other Azure Bot Framework channels can be enabled by submitting a support request. Contact support'.

Congratulations! You have successfully installed the Health Bot and enabled integrations with Omnichannel and Teams. Also, you have obtained a BotId and stored with you.

Exercise 2: Obtain Azure AD Application Id

In this exercise, you will be using an Azure AD Application Id we have already created for you in Azure called "**MCH Application Id**". Registering this Id established a trust relationship between your Dynamics 365 app and the Microsoft identity platform. Using this Id, you will later create a Dynamics 365 Application User to bridge the authentication between Azure Health bot and Power Apps.

Note: Normally, user must create this Azure AD Application In their tenant to establish the authentication between Azure Health Bot and Dynamics 365.

1. **Sign in** into [Azure Portal](#) Tenant where **Customer Service for Omnichannel** is installed and **Search for MCH Application Id**.
2. In the top Search bar, type **App Registration** and select **App Registration** from the list.

The screenshot shows the Microsoft Azure portal interface. At the top, there is a search bar with the text "App Registrations". Below the search bar, the "Services" section is displayed, featuring a list of various Azure services. The "App registrations" service is highlighted with a red box. Other services listed include Event Grid Partner Registrations, AppDynamics, App Configuration, App proxy, App Services, Function App, Application gateways, Application groups, and Application Insights. The "Resources" section at the bottom indicates "No results were found." There is also a note stating "Searching all subscriptions. Change". On the left side, there is a sidebar for "Subscriptions" with "OpenHack Subscription" selected.

3. Select +All applications >> Search Box >> Type Text >> "MCH Application Id" >> Click link "MCH Application Id".

The screenshot shows the Microsoft Azure portal's App registrations page. At the top, there's a search bar and several navigation links: New registration, Endpoints, Troubleshooting, Download, Preview features, and Got feedback?. Below the header, there are two informational cards: one about the new App registrations search preview and another about the end of support for ADAL and Azure AD Graph. The main content area has tabs for All applications and Owned applications, with All applications selected. A search bar is present. Below it, a card displays the application details for "MCH Application Id": Display name (MCH Application Id), Application (client) ID (redacted), and a small icon showing "MA". An orange arrow points from the text "Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph." to the "All applications" tab.

4. Copy Application (client) ID and store it for later use.

The screenshot shows the Microsoft Azure portal's App registration details page for "MCH Application Id". The left sidebar includes Overview, Quickstart, Integration assistant, and Manage sections with sub-options like Branding, Authentication, and Certificates & secrets. The main content area shows the application details under the Essentials section: Display name (MCH Application Id), Application (client) ID (redacted), Directory (tenant) ID (redacted), Object ID (redacted), and other properties like Supported account types, Redirect URIs, Application ID URI, and Managed application in I... (all redacted). An orange box highlights the "Application (client) ID" field. A note at the bottom states: "Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph." Learn more.

Congratulations! You have successfully obtained a "**MCH Application ID**" from Application Registration in Azure Portal.

Exercise 3: Configure Customer Service Omnichannel Live chat

In this exercise, you will be configuring Dynamics 365 omnichannel which would enable Live Chat agent to communicate with Power App Portal User via Azure Health bot.

You will be doing the following steps in this exercise.

1. Create Application User using **MCH Application Id** and **Bot ID**
2. Configure Queues for Bot and Agent User
3. Configure Context Variable and Routing rule to route the message either to Bot or Agent.

Omnichannel for Customer Service offers a suite of capabilities that extend the power of Dynamics 365 Customer Service Enterprise to enable organizations to instantly connect and engage with their customers across digital messaging channels. An additional license is required to access Omnichannel for Customer service.

Task 1: Create Health bot User in Dynamics 365 Customer Service

Assumption: Omnichannel for Customer Service is already installed to proceed further.

We would need two persona users to be configured in Dynamics 365 Customer Service Instance.

- **Health bot User** (it helps to bridge the connection between **health bot and dynamics 365 app** and allow the messages to flow from both the direction.)
- **Omnichannel Agent User** for Customer Service who receives a message from consumer/bot and respond to those conversations. (**You may want to use your account as an Omnichannel Agent User by assigning “Omnichannel Agent”**)

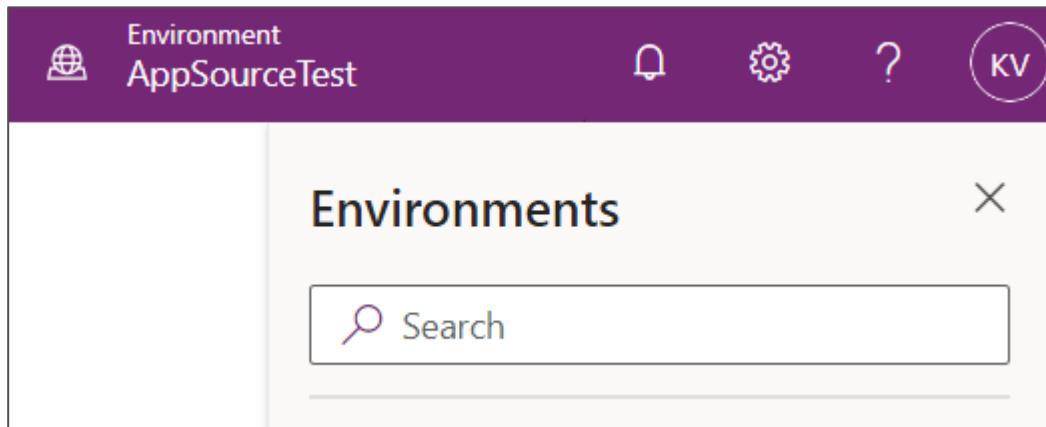
In this task we will be creating a **Bot User** which helps to connect **Azure Health Bot** with **Omnichannel live Chat**. Also, we would be using another user that would be a power platform user account assigned with Omnichannel Agent security role (You can use your logged in user account as an Omnichannel Agent account and assign “Omnichannel Agent” security role)

1. Sign into **Power Apps web studio** and **navigate to your environment**.

- a. Go to [Power Apps](#) and click **Sign in**. You may also directly navigate to [Power Apps Makers Portal](#).

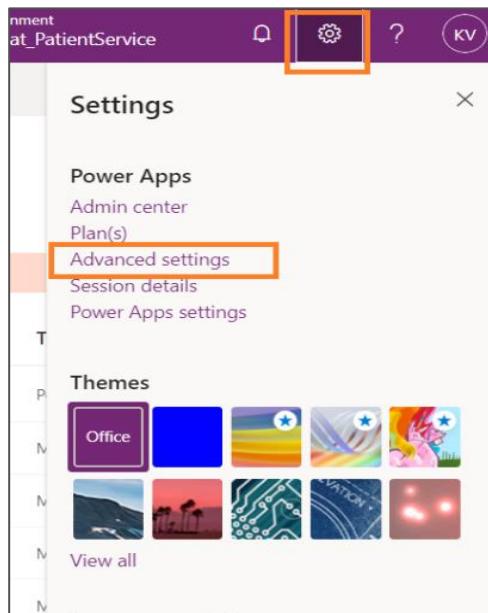


- b. Choose the environment where you installed Microsoft Cloud for Healthcare.

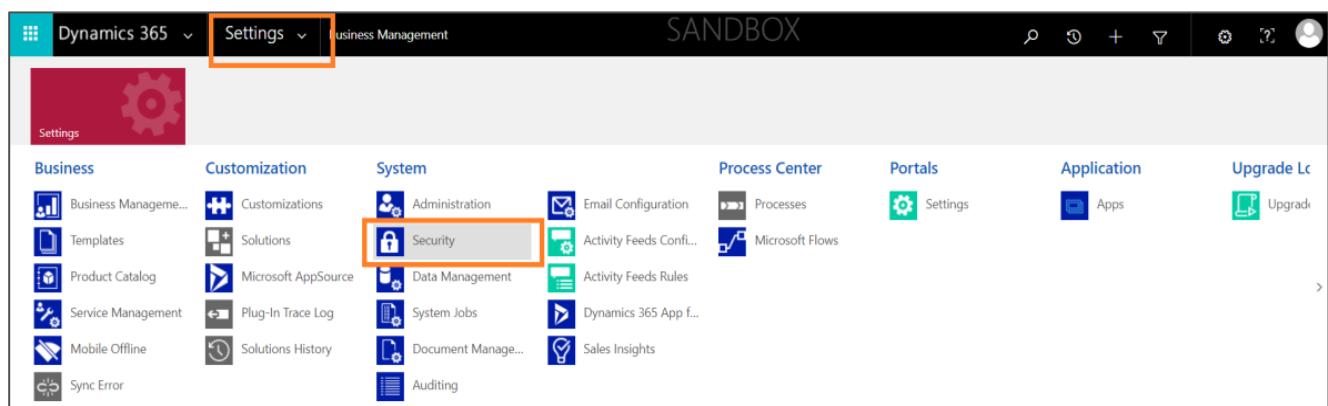


2. Create **Health bot User** in **Dynamics 365 Omnichannel for Customer Service**.

- Navigate to the [Power Apps Makers Portal](#).
- Select the gear in the upper right corner and navigate to **Settings > Advanced Settings**.



- You should be navigated to **Dynamics 365**. In the drop down, select **Settings > Security**.



d. Select **Users**.

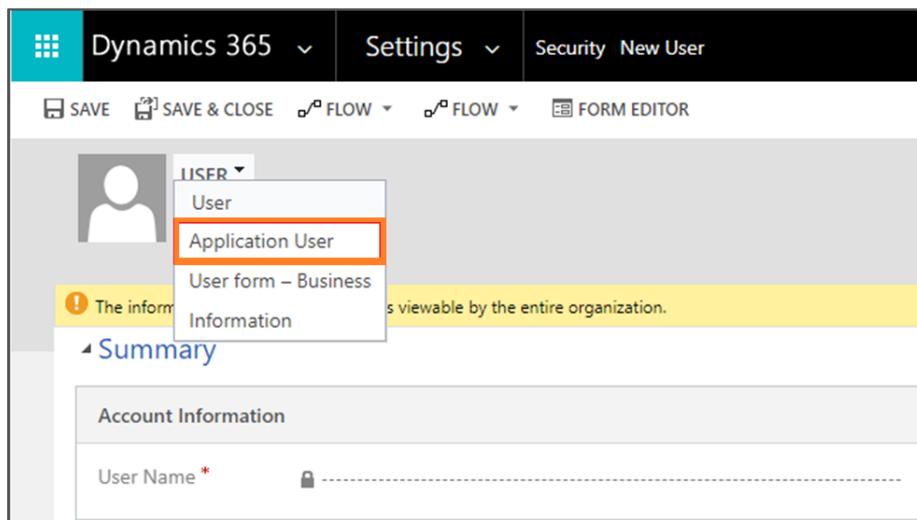
The screenshot shows the Dynamics 365 Security page under the SANDBOX environment. The top navigation bar includes 'Dynamics 365', 'Settings', and 'Security'. A yellow banner at the top right says 'Dynamics 365 for Outlook Deprecated' with a link to learn more. Below the banner, the page title is 'Security' and the sub-section is 'Which feature would you like to work with?'. There are six items listed:

- Users**: Add new users. Edit information about users and deactivate user records. Manage the teams, roles, and licenses assigned to users.
- Teams**: Add new teams and new members to existing teams. Modify the team template description.
- Security Roles**: Create new security roles. Manage and delete existing security roles for your organization.
- Business Units**: Add new business units. Edit and deactivate existing business units.
- Field Security Profiles**: Manage user and team permissions to read, create, or write information in secured fields.
- Hierarchy Security**: Configure hierarchy security, including enabling hierarchy modeling goes, and specify the entities to exclude from a hierarchy.
- Positions**: Add new Position. Modify the Position description.
- Access Team Templates**: Add new team templates. Modify the team template description.

e. In the view drop-down, select **Application Users**. Select (+) **New** button

The screenshot shows the Dynamics 365 Application Users list page. The top navigation bar includes 'Dynamics 365', 'Settings', and 'Security'. The sub-section is 'New User'. The toolbar has buttons for '+ NEW', 'EDIT', 'APPROVE EMAIL', 'REJECT EMAIL', 'PROMOTE TO ADMIN', 'MANAGE ROLES', and 'CHANGE'. The main area shows a table with one row:

	Full Name ↑	Application I...	Azure AD Obj...	Application I...
<input checked="" type="checkbox"/>	Omnichannel AppUser_18cc...	18cc9627-77...	26631d04-a8...	https://micro...

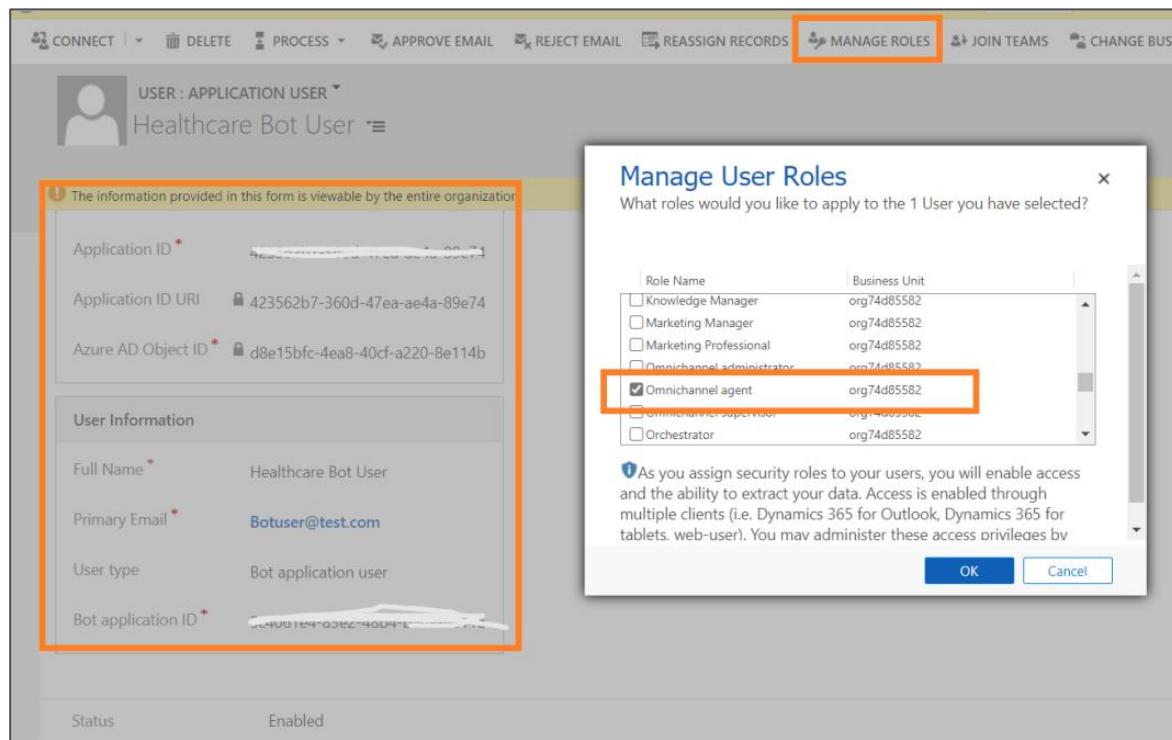


3. On the **New User** page, enter or select the following information:
 - a. **Application ID:** An application ID for any valid (non-expired) application created in Azure Active Directory (Azure AD) for the same tenant. It is not used by the bot in Omnichannel for Customer Service. You stored this value in Exercise 2. Ex: **MCH Application ID**
 - b. **User type:** Select **Bot application user**. This will show a new field to store the Bot application Id.
 - c. **Bot application ID:** Bot's ID you copied in the previous step.
 - d. Click **Save**. It will auto populate the other values in the record.

The screenshot shows the "New User" page with the following details:

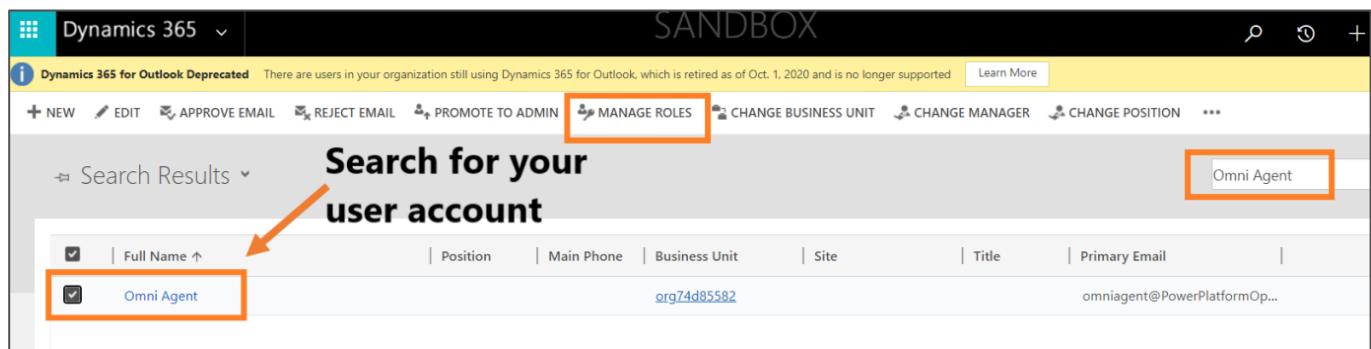
- USER : APPLICATION USER** is selected in the dropdown.
- Summary** section:
 - Account Information: User Name (locked)
 - Application ID *: 4b463652-4bc4-42ff-8ccd-
 - Application ID URI (locked)
 - Azure AD Object ID (locked)
- User Information** section:
 - Full Name * (locked)
 - Primary Email (locked)
 - User type: Bot application user
 - Bot application ID *: 5cbf53b3-3ec8-47d5-899a-

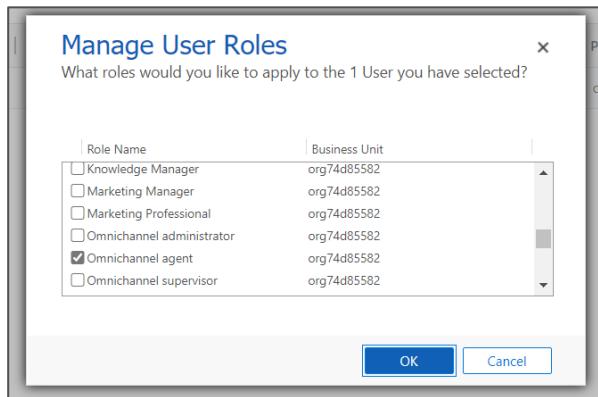
4. Select **Manage Roles** on the command bar.
5. In the **Manage User Roles** window, select **Omnichannel agent**, and then select **OK**.



6. Similarly, you need to ensure that "**Omnichannel Agent**" security role assigned to your user account. Example I have used username called **Omni Agent**

Note: If you are in an official training, use the account assigned to you. Otherwise, you can use any one of the Power Platform user account as an Omnichannel Agent user.





Task 2: Create and Configure Omnichannel Queues

1. Add a Bot User into Omnichannel Queue to distribute to incoming customer queries.

- a. Open the **Dynamics 365 Omnichannel Administration** app.

Power Apps

+ New app Edit Play Share Monitor Delete Settings

Apps

Apps Component libraries (preview)

⚠ 6 environment variables need to be updated. See environment variables

Name	Modified
Patient Access mc4hc portal	... 18 h ago
Healthcare Administration	... 14 h ago
Portal Management	... 17 h ago
FHIR Sync Agent Administration	... 1 d ago
Patient Service Center	... 1 d ago
Customer Service workspace	... 2 d ago
<input checked="" type="checkbox"/> Omnichannel Administration	... 2 d ago

2. Select **Queues** > Open **Default Messaging Queue**

Omnichannel queues

	Name	Queue type
Default entity queue		Entity
Default messaging queue		Messaging

Note: We are using **Default messaging queue** mapped with the Bot User so it will respond to incoming messages from customers without agent (human) intervention

3. Select **Add Existing User** to add the Bot user you previously created.

Default messaging queue

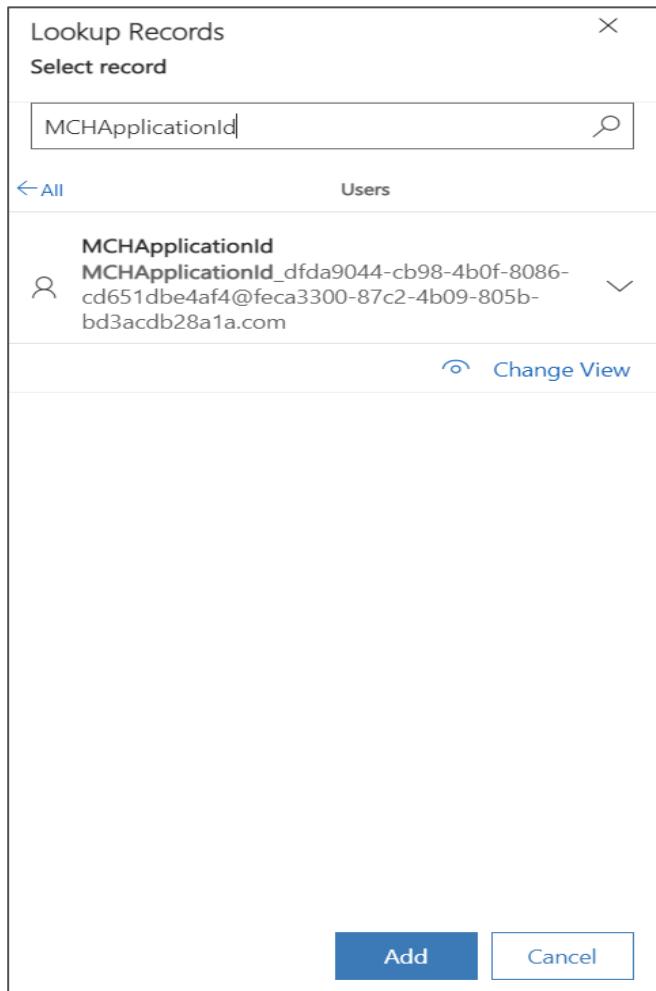
Summary Related

Full Name	Capacity	Business Unit
No data available.		

Add Existing User

4. In this “Lookup Records”: Search for your Bot User (**MCHApplicationId**) that was just created in the earlier task. It should start with **MCHApplicationId** and Select the record and click **Add**.

Note: Please ignore this step if you see MCHApplicationId already added.



The screenshot shows the 'Default messaging queue' configuration page. On the left, there's a 'SUMMARY' section with the following details:

- Name: * Default messaging queue
- Priority: * 2,147,483,647
- Queue type: * Messaging
- Owner: * SYSTEM

Below this is a note: "By using this feature, you acknowledge that this feature is in preview and you agree to the [Preview Terms](#). [Learn more](#)".

On the right, there's a 'Users (Agents)' section with a table:

Full Name	Capacity	Business Unit
Health Bot User	---	healthcareinaday

5. Go back to the **Omnichannel queues** grid. Create **New Queue** for Omnichannel Agent.

Omnichannel queues

	Name	Queue type
	Default entity queue	Entity
	Default messaging queue	Messaging

6. Name the queue “Escalate To Human” and give Priority of 1 (lower than default queue). Click Save.

New Queue

Summary

SUMMARY

Name	* Escalate To Human
Priority	* 1
Queue type	* Messaging
Owner	* Current User

By using this feature, you acknowledge that this feature is in preview and you agree to the [Preview Terms](#). [Learn more](#)

7. Select Add Existing User. Search for your user account and add that user into Escalate To Human Queue. (Ex: Added user Omni Agent)

Note: If your user account is added already, skip this step.

Escalate To Human

Summary

SUMMARY

Name	* Escalate To Human
Priority	* 1
Queue type	* Messaging

Add your user account

Users (Agents)

Full Name	Capacity	Business Unit
Omni Agent	100	org74d85582

Add Existing User

Note: The queue **Escalate To Human** is created to manage and redirect the incoming messages from a customer to a Customer Service (human) Agent when Bot responds true to “Escalate To Agent”.

Task 3: Update Live Work Stream with Context Variables and Routing Rules

In this task, we will set up basic chat routing to behave as the following:

- In this Task we will be creating two routing rules for Bot and Customer Service Agent.
-
- Bot Routing Rule : Customer initial conversation would be with chat bot, we will have When the chat bot is first opened, route to Default queue, which only contains the bot user. The condition is when the context variable **EscalateToAgent** is not present.
- Add another routing rule, when context variable **EscalateToAgent** is present and set to 1, to route to a queue that has only human users (agents) that can take control of conversation.

1. Navigate to **Work Streams** > Select the **Live chat workstream**.

The screenshot shows the 'Active Work Streams' list. On the left is a navigation sidebar with options: Queues, Users, Bots, Skills, Work Distribution Management, Work Streams (which is highlighted with an orange box), and Agent Experience. The main area lists several work streams: 'Custom messaging workstream', 'Facebook workstream', 'LINE workstream', 'Live chat workstream' (which is highlighted with an orange box), and 'Microsoft Teams workstream'. At the top of the main area are buttons for 'Show Chart', 'New', and 'Delete'.

2. In the **Live chat workstream**, select the **Context Variables** tab.

The screenshot shows the 'Live chat workstream' configuration page. At the top, there are tabs for 'Push' (selected), 'Work Distribution Mode' (set to 'Channel'), 'Live chat' (selected), 'SYSTEM' (selected), and a dropdown. Below the tabs, there are buttons for 'Work Distribution' (disabled), 'Context Variables' (selected and highlighted with an orange box), 'Skill Attachment Rules', 'Routing Rules', 'Templates', 'Smart assist', and 'Quick Replies'. The 'Context Variables' section shows a table with columns for 'Name' (sorted by 'Display Name'), 'Type', and 'Value'. A search bar at the top of this section includes filters for 'Display Name' and 'Type'. At the bottom right of this section are buttons for '+ New', 'Refresh', 'Run Report', and more options.

3. Create **New** Context Variable with the following specifications:

- a. **DisplayName: EscalateToAgent**
- b. **Name: EscalateToAgent**
- c. **Type: Number**

Quick Create: Context variable

Display Name	EscalateToAgent
Name	* EscalateToAgent
Owner	* Current User
Type	* Number

Save and Close **Cancel**

4. Click **Save and Close**.
5. Navigate back to the **Live chat workstream** record and select the **Routing Rules** tab.

Live chat workstream
Work Stream

Push
Work Distribution Mode | Live chat Channel | SYSTEM Owner

Work Distribution Context Variables Skill Attachment Rules **Routing Rules** Templates Smart assist Quick Replies

Rule items **+ Add** **Run Report** :

Name	Description	Modified On

6. Add a new **Routing Rule -- ToHealthcareBot**
 - a. **Name:** ToHealthcareBot
 - b. **Queue:** Default messaging queue
 - c. **No Conditions.**
 - d. Select **Save & Close**.

New Rule Item

General

General Information	Condition
Name * ToHealthcareBot	AND OR Ungroup
Owner *	+ Condition
Work stream * Live chat workstream	
Queue * Default messaging ...	
Description ---	

7. Add another new Routing Rule – ToAgent

- Name:** ToAgent
- Queue:** EscalateToHuman
- Add Condition:** Context Variable “EscalateToAgent = 1”
- Save & Close**

New Rule Item

General

General Information

Name	ToAgent
Owner	O [User]
Work stream	Live chat workstream
Queue	Escalate To Human
Description	Rule Agent Routing

Condition

AND OR Ungroup Delete Reset

Context variable EscalateToAgent Equals 1

Condition

AND	OR	Ungroup	Delete	Reset
<input type="checkbox"/> Context variable	EscalateToAgent	Equals	1	

8. On the Live chat workstream, you can see the two **Routing Rules** we created for **Bot (ToHealthcareBot)** and **Agent (ToAgent)**.

Live chat workstream

Work Stream

Push
Work Distribution Mode

Work Distribution Context Variables Skill Attachment Rules **Routing Rules** Templates Smart assist Quick Replies

Rule items

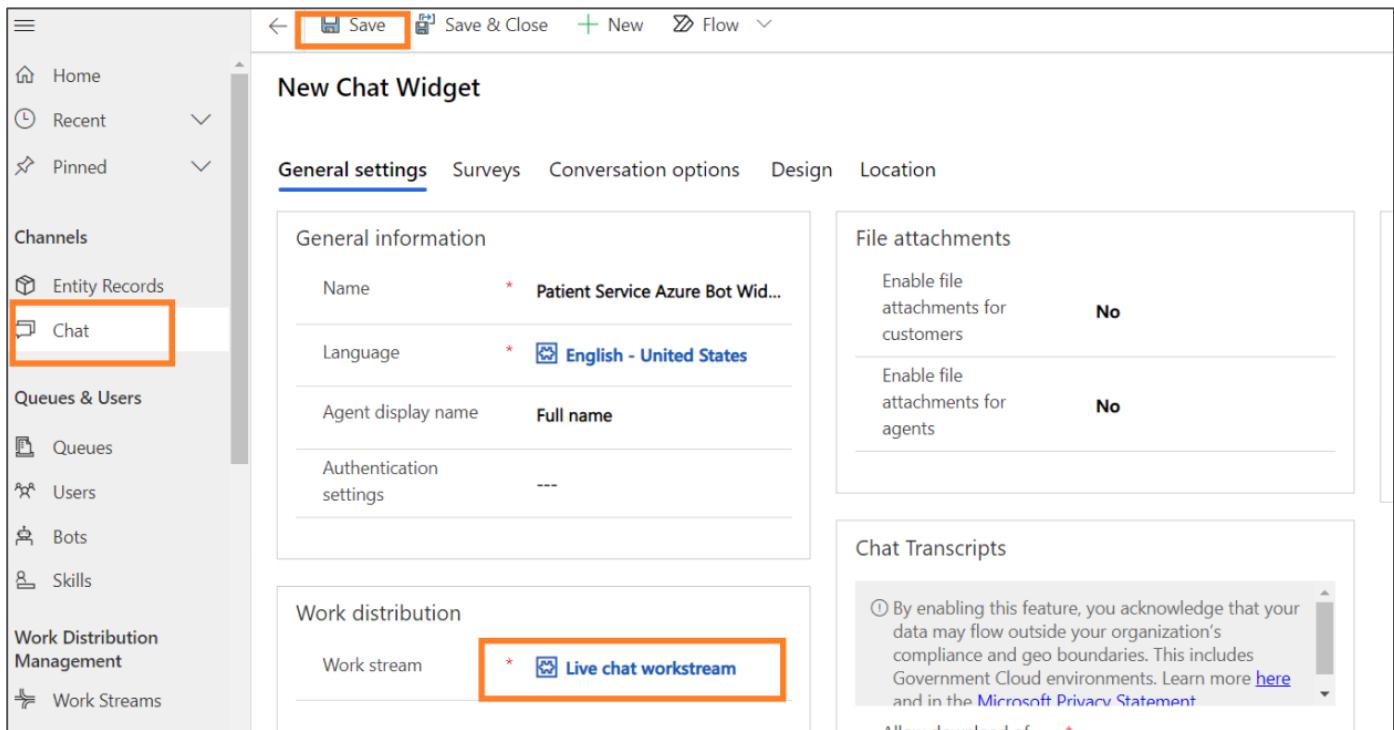
Name	Description	Modified On
ToHealthcareBot	Rule for Bot Routing	11/12/2020 8:10 PM
ToAgent	Rule Agent Routing	11/12/2020 8:09 PM

Task 4: Create Chat Widget for Health bot

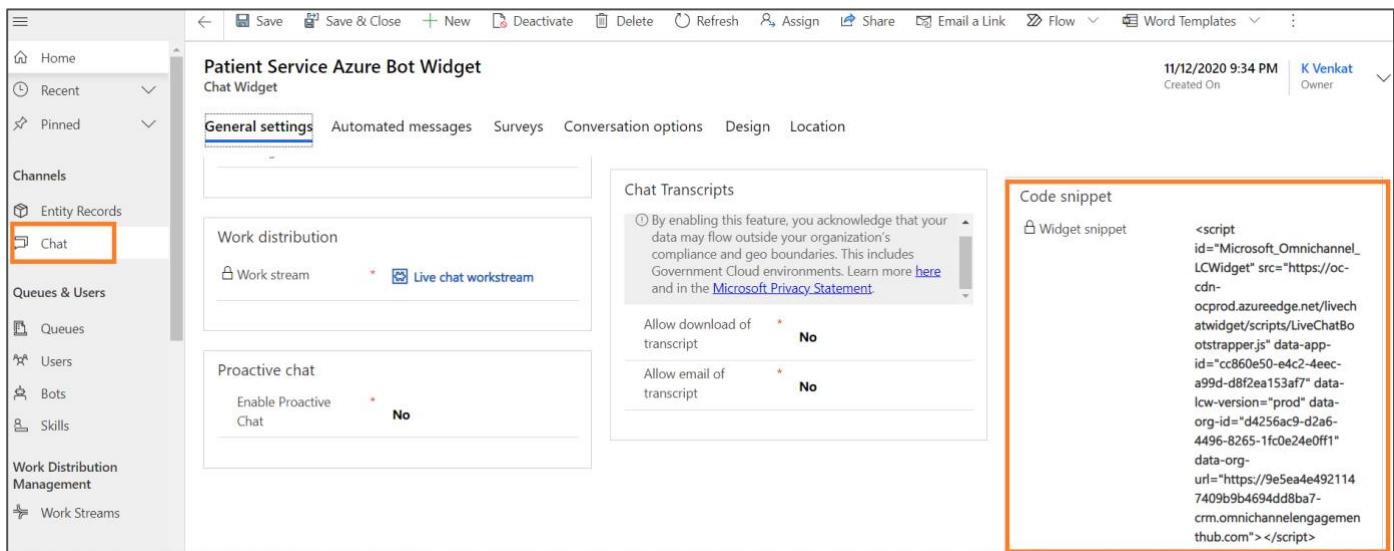
1. Navigate to **Chat** > **New Chat Widget**.

2. Give the **Chat Widget** a **Name** (ex: Patient Portal Chat Widget).

3. Click **Save**.



4. After you saved the record, it generates a **Widget Code Snippet** (Copy the code snippet and store it for later use!)



Congratulations! You have successfully configured Customer Service Omnichannel Live chat by creating the necessary Users, Queues, Work Streams, Context Variables, Routing Rules, and Chat Widget. These all work together and allow patients to chat with a virtual health bot with the option to escalate up to a human agent if needed.

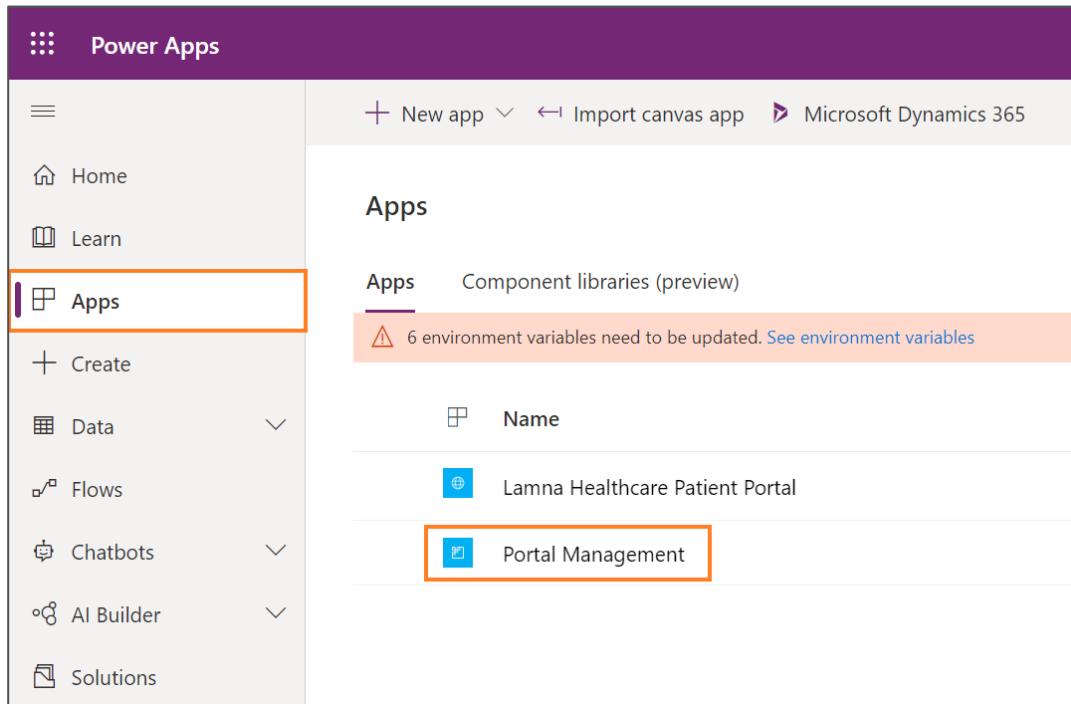
Exercise 4: Embed Health Bot in Power Apps Portal

In this exercise, you will be embedding the **Omnichannel Chat Widget** into the Power Apps Customer self-service portal using Portal Management configuration. We have already created Lamna Healthcare Company Portal using **Customer self-service portal** template before deploying Microsoft Cloud for Healthcare. Now we will configure the chat widget to show on the customer website.

- **Customer self-service portal:** A customer self-service portal enables customers to access self-service knowledge, support resources, view the progress of their cases, and provide feedback.
- **Portal Management** app to help you get started with the advanced portal configuration. In this walk-through, you will learn how to configure Chat widget in **Portal Management** app.

Note: Power Apps **Customer Self-service** portal must be installed as a prerequisite for this task. If you are in an official training, this has already been set up for you.

1. Login to **PowerApps** web portal (<https://make.powerapps.com>). Select your environment where you've installed Microsoft Cloud for Healthcare.
2. Go to **Apps** > Select and open **Portal Management** app



3. Select **Content Snippets** in the sitemap (left navigation panel)
4. In **Active Content Snippets**, type “**Chat Widget**” in the **Search** box.
5. It retrieves “Chat Widget Code” records (refer below picture)

6. Click to Open "**Chat Widget Code**" record related to **Customer Self-service**.

Name	Website	Type
Chat Widget Code	Customer Self-Service	HTML
Chat Widget Code	Healthcare Patient Portal	HTML

7. In the **Chat Widget Code** record, Select **Value (HTML)** >> **HTML Tab** and then past the **Chat Widget Code snippet** that you [copied and stored](#) (see screenshot below)

8. Click **Save & Close**.

```

1 <script id="Microsoft_Omnichannel_LCWidget"
2 src="https://oc-cdn-ocprod.azureedge.net/livechatwidget/scripts/LiveChatBootstrapper.js"
3 data-app-id="0e50-e4c2-4eec-a99d-d8f2ea153af7" data-lcw-version="prod" data-org-id="d4256aa6-4496-8265-1fc0e24e0ff1"
4 data-org-url="https://21147409b9b4694dd8ba7-crm.omnichannelengagementhub.com"></script>

```

9. Now click to Open the **Chat Widget Code** associated with the **Healthcare Patient Portal** website.

Active Content Snippets ▾				
✓ Name ↑	Website ▾	Content Snippet Language ▾	Type ▾	Value ▾
Chat Widget Code	Customer Self-Service	---	HTML	---
Chat Widget Code	Healthcare Patient Portal	---	HTML	<script id="Microsoft_Omnichannel_LCWidget..."

10. In the **Chat Widget Code** record, paste in **Value (HTML)** the same **Chat Widget Code snippet** that you copied and stored in [exercise 3, task 4](#) and added to the customer self-service chat widget code.

11. Click **Save & Close**.

The screenshot shows the Dynamics 365 Portal Management interface. On the left, the navigation pane includes 'Executive Utilities', 'Settings', 'Content' (with 'Content Snippets' selected), 'Security', and 'Portals'. The main area displays a record for 'Chat Widget Code' under 'Content Snippets'. The 'General' tab is selected. The record details are as follows:

- Name:** Chat Widget Code
- Website:** Healthcare Patient Portal
- Display Name:** Chat Widget Code
- Type:** HTML
- Content Snippet Language:** ---

The 'Value (HTML)' field contains the following code, which is highlighted with a red box:

```

<script id="Microsoft_Omnichannel_LCWidget" src="https://oc-cdn-ocprod.azureedge.net/livechatwidget/scripts/LiveChatBootstrapper.js" data-app-id="12345678-9abc-def0-4321-a1b2-c3d4e5f6g7h8"></script>

```

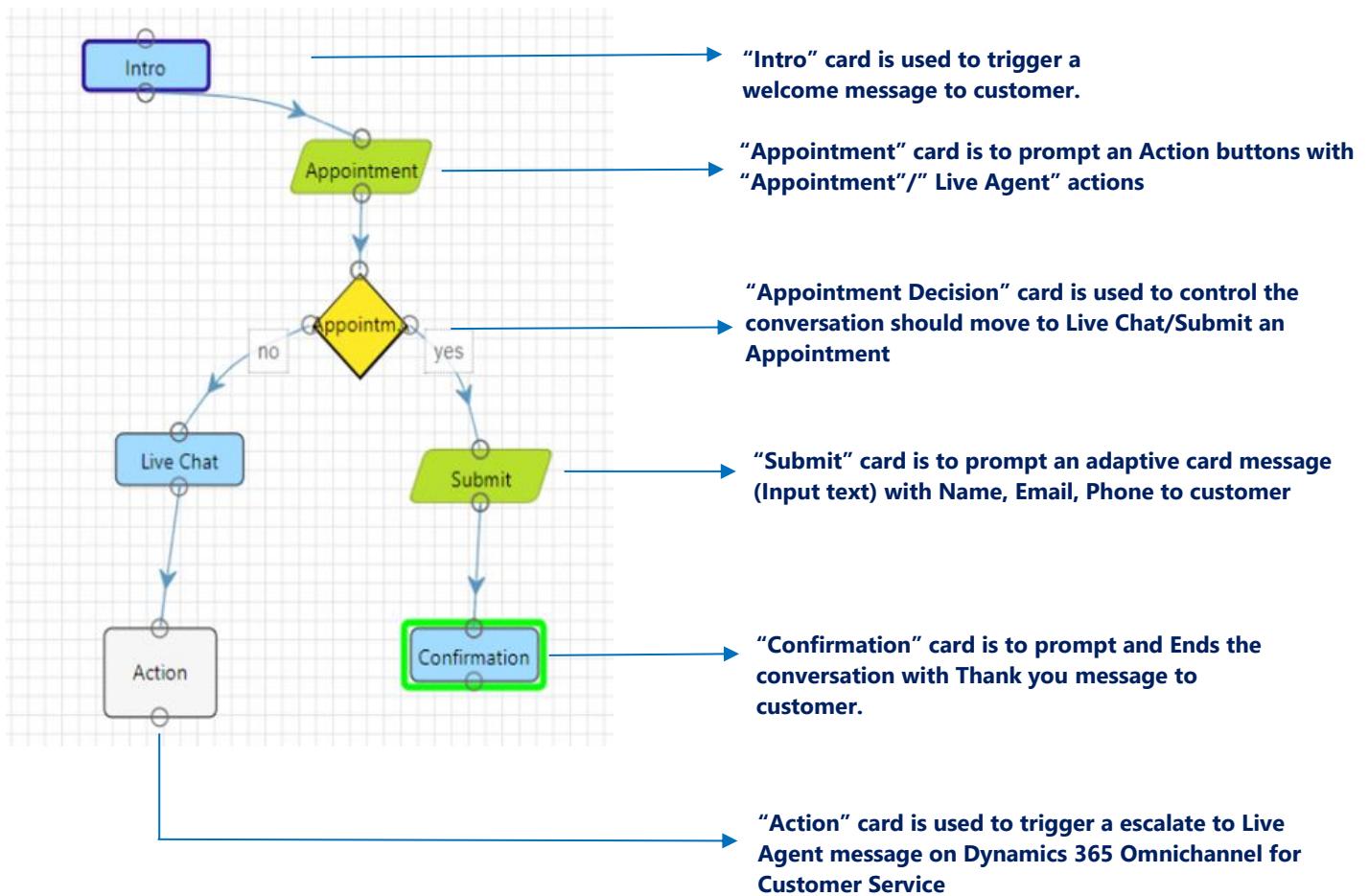
Congratulations! In this exercise you have successfully updated the chat widget in the Power App Portal Content Snippets. With this configuration, Bot will be visible on the Power Apps portal (both the customer self-service portal e and the healthcare patient portal).

Exercise 5: Extend Azure Health Bot with Custom Scenarios

Dynamics 365 Omnichannel integration allows the patient to interact with **Azure Health Bot** using the Dynamics 365 chat widget to access the medical knowledge and your custom scenarios. It also, allows the escalation of a bot conversation to a live agent to continue the interaction. When escalating a conversation, Dynamics passes along the conversation history and the context to the agent.

In this exercise, you will be do the following.

1. You will be designing the below Health Bot Scenario **MCH_PatientService**



2. You will be designing another Health Bot Scenario called "**MCH_Patient Welcome Scenario**". This scenario just holds the Intro scenario which would invoke the **MCH_PatientService** scenario.
3. Making "MCH_PatientServiceWelcome" scenario as a startup scenario by updating the setting **Automatic Welcome Scenario**

Task 1: Navigate to Health Bot and Create Scenarios.

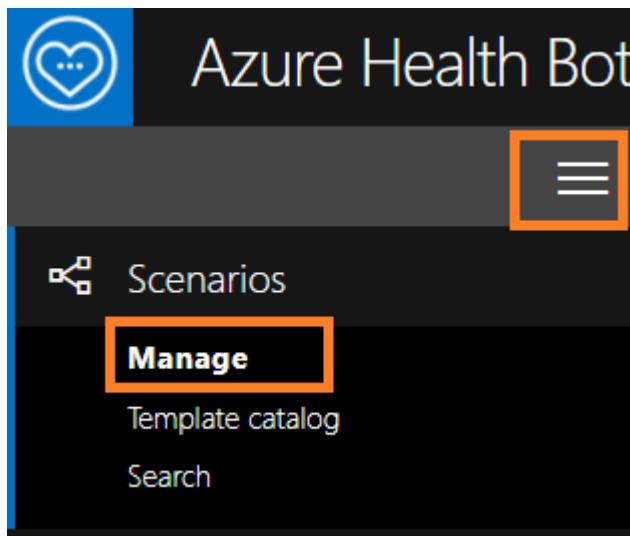
In this task, you will be create two bot scenarios “**MCH_PatientService**” and “**MCH_Patient Service Welcome Scenario**”.

Note: You have copied the Management Portal link in the **Exercise-1, Task- 1.10**

1. If you are not seeing the below **scenario page**, use the copied Azure Bot Management Portal link in [Exercise-1, Task- 1.10](#) and Open the link with **InPrivate** window for **Edge** browser/**Incognito** for **Chrome** browser

2. Let us create a **New** health bot scenario **MCH_PatientService**

1. Click to Expand the Side navigation bar (Navigate to **Scenario > Manage**)



2. Navigate to Scenario >> Manage >> Click (+) New button on the top ribbon.

A screenshot of the "Scenario Management" page in the Azure Health Bot Service. The top navigation bar has a blue icon with a heart and the text "Health Bot Service". The left sidebar has the following options:

- Scenarios (highlighted with an orange box)
- Manage (selected)
- Template catalog
- Search

The top ribbon has buttons for "+ New", "Refresh", "Import", "Export", and "Delete". The main content area is titled "Scenario Management" and contains the following text: "Create and manage custom scenarios for your bot instance. [Learn more](#)". A table lists the scenarios:

Active	Name	Scenario ID	Description
<input type="checkbox"/>	FlowConnection	FlowConnection	
<input type="checkbox"/>	MCH_Patient Service...	MCH_PatientServiceWe...	Patient Service Welcome Scenario
<input type="checkbox"/>	MCH_PatientService	MCH_PatientService	MCH_PatientService

3. Provide the health bot Name and Scenario Id

a. Name: **MCH_PatientService**

b. Scenario Id: **MCH_PatientService**

New Scenario

Name* ⓘ
MCH_PatientService

Description ⓘ

Scenario ID* ⓘ
MCH_PatientService

Returning Message ⓘ

Interrupting scenario ⓘ

Breaking scenario ⓘ

Create **Cancel**

4. Now that you have created the **MCH_PatientService** bot scenario, let us go and design the bot conversation. Select the Scenario **Name** in **Scenarios > Manage** to edit.

Scenario Management

Create and manage custom scenarios for your bot instance. [Learn more](#)

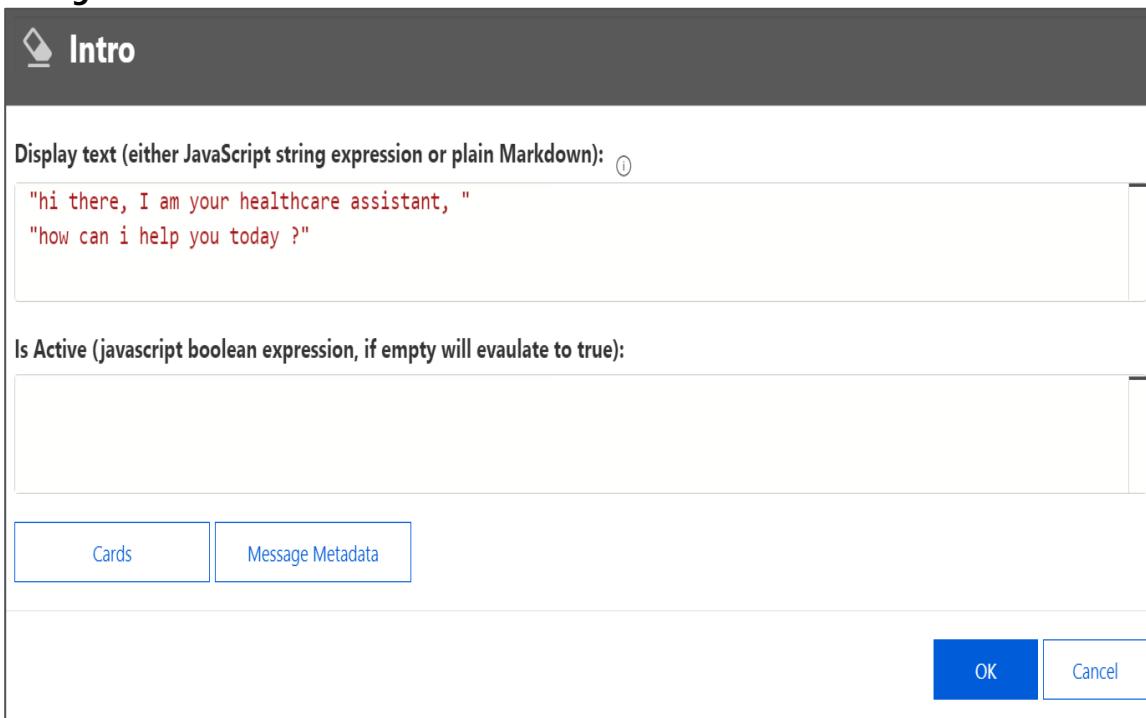
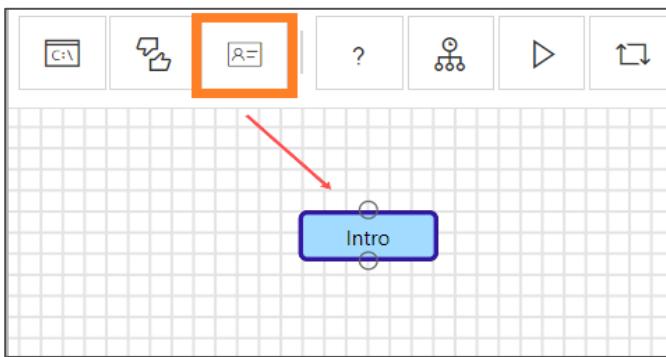
<input type="checkbox"/> Active	Name ⓘ	Scenario ID	Description	Last Modified	Actions
<input type="checkbox"/> <input checked="" type="checkbox"/>	MCH_PatientSer...	MCH_PatientServ...		Today at 23:10:27	

Scenario : Open [MCH_PatientService](#) scenario editor**1 . Intro - Add Bot Introduction statement**

Mouse select and hold on “**Statement**” icon >> Drag onto canvas editor.

**a. Enter Display Text:**

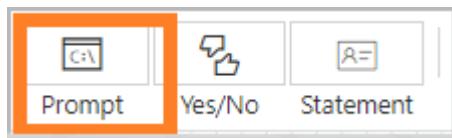
"Hi there, I'm your Healthcare Assistant."
"How can I help you today?"

b. Change Title to "Intro"**c. Click OK.**

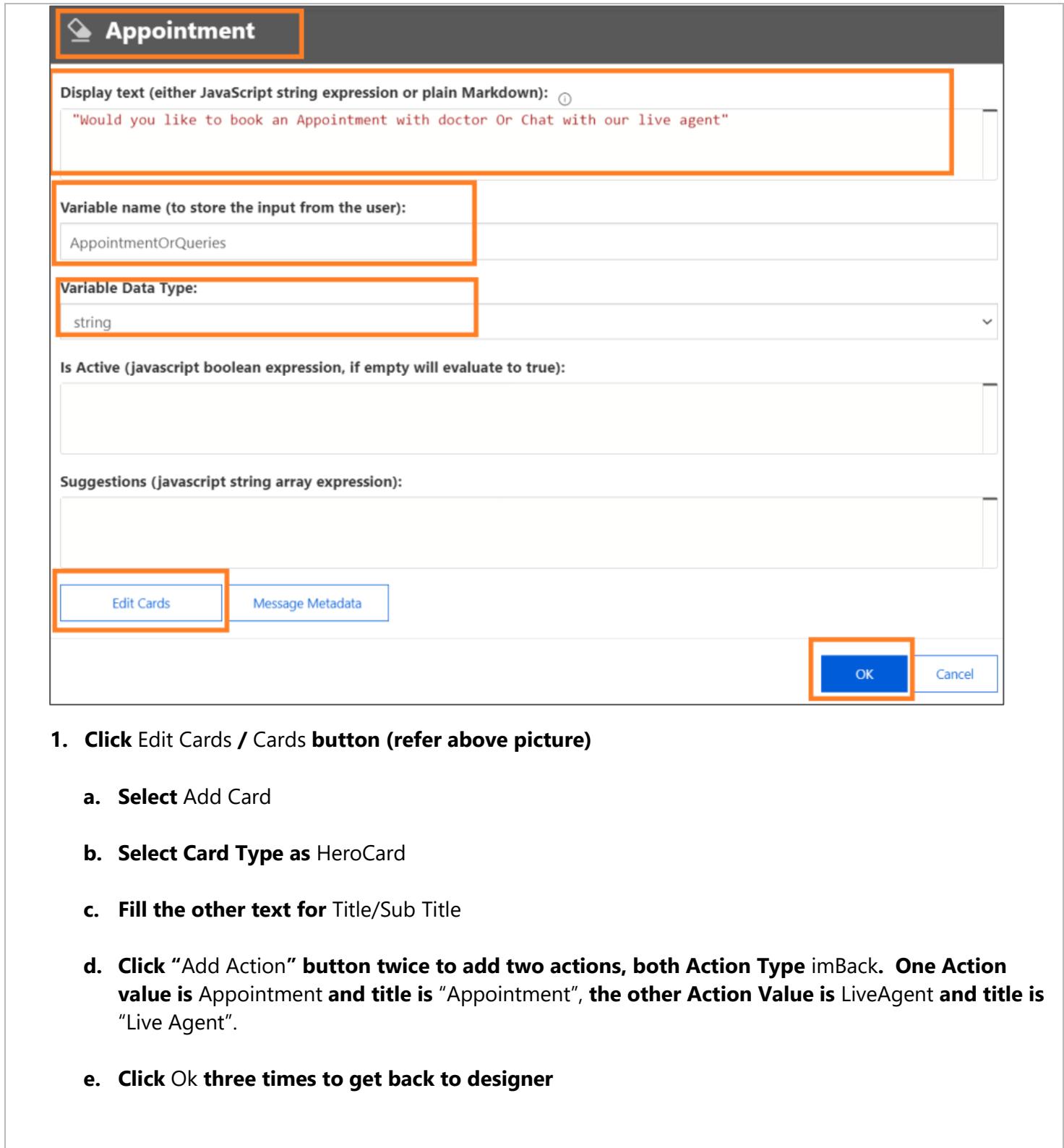
2. Add Book an Appointment or Talk to Live Agent statement

Use Case: This section prompts two buttons **Appointment** and **Live Agent**. When user click any one of the buttons it will set the appropriate text to the variable **AppointmentOrQueries**

Select “**Prompt**” icon and drag down onto canvas.



- a. Enter Display Text: **“Would you like to book an appointment with the doctor or chat with a live agent?”**
- b. Add Variable name as **AppointmentOrQueries**
- c. Select data type as **string**
- d. Rename title to **Appointment**



1. Click Edit Cards / Cards button (refer above picture)

- Select Add Card**
- Select Card Type as HeroCard**
- Fill the other text for Title/Sub Title**
- Click “Add Action” button twice to add two actions, both Action Type imBack. One Action value is Appointment and title is “Appointment”, the other Action Value is LiveAgent and title is “Live Agent”.**
- Click Ok three times to get back to designer**

Card

Card Type: HeroCard

Image Url:

Title: Appointment Or Chat with live agent

Sub Title:

Actions

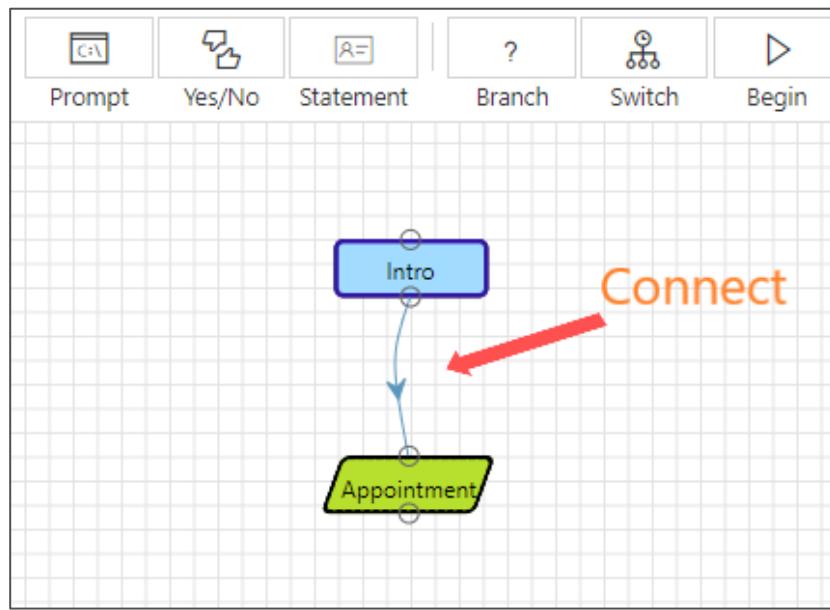
Action type	Action value	Action title
imBack	Appointment	"Appointment"
imBack	LiveAgent	"Live Agent"

Add Action

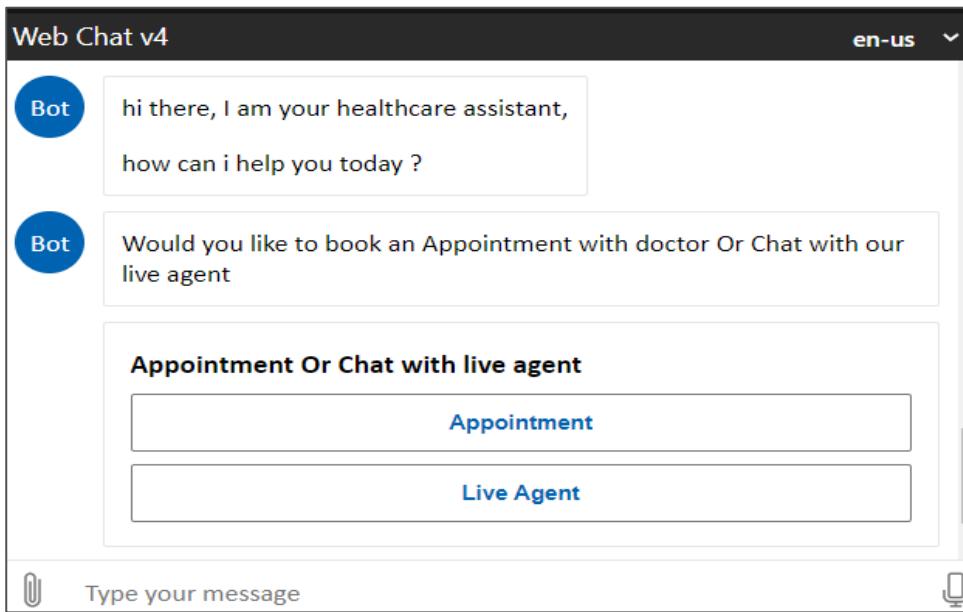
OK **Cancel**

2. Connect Intro and Appointment boxes.

Automatically arrow will appear when you try to connect Intro and Appointment boxes using ellipse pointer



You will be able to see the below conversation **output** when you run your bot.

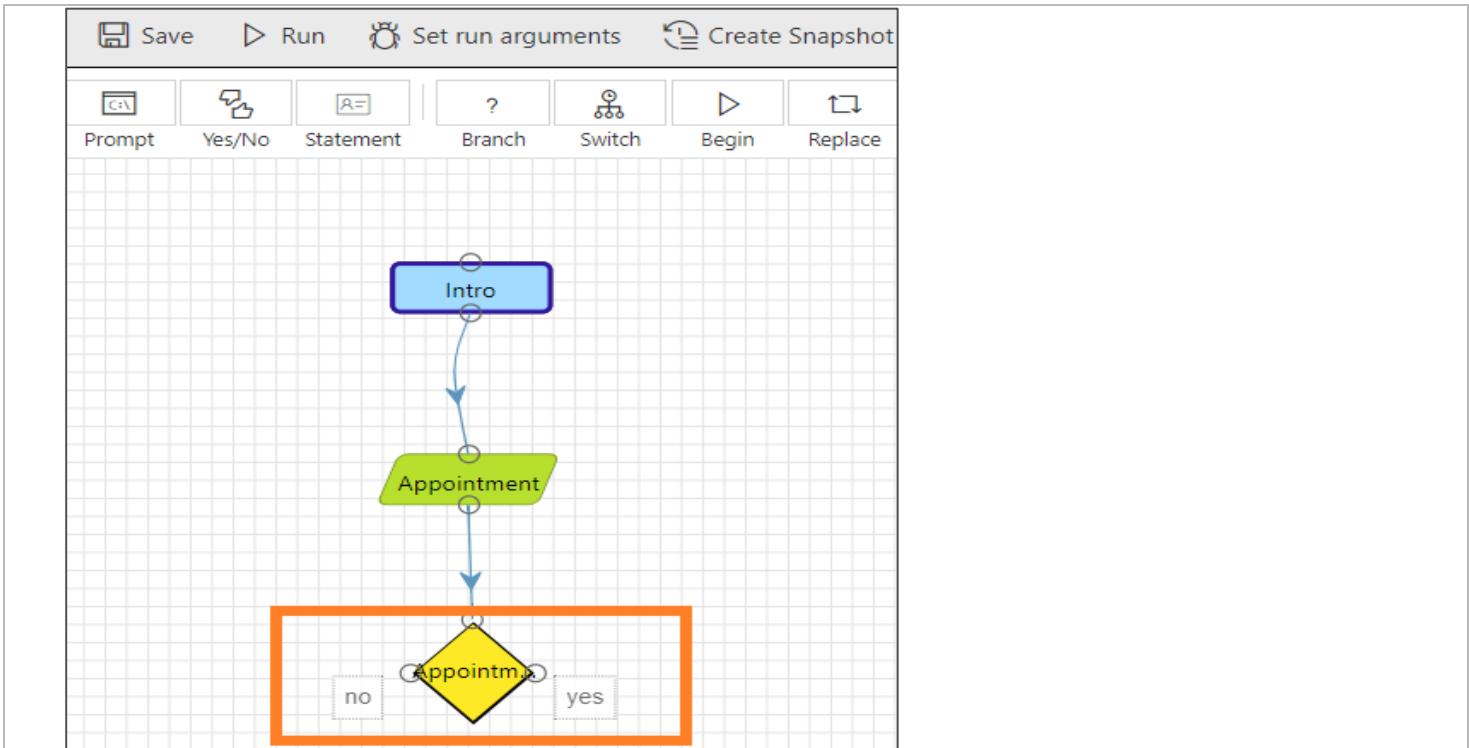


3. Add decision Branch to capture variable **AppointmentOrQueries** and check whether the user selected value is “**Appointment**.”

+ Select Branch



Use Case: This section checks whether the user has clicked **Appointment** or **Live Agent** with the help of **AppointmentOrQueries**. It will redirect the message accordingly.



a. Add this below statement

```
scenario.AppointmentOrQueries === "Appointment"
```

⌚ Appointment/Agent ?

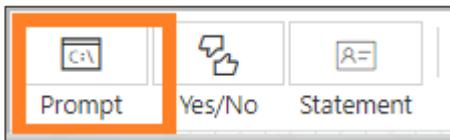
javascript boolean expression:

```
scenario.AppointmentOrQueries === "Appointment"
```

OK Cancel

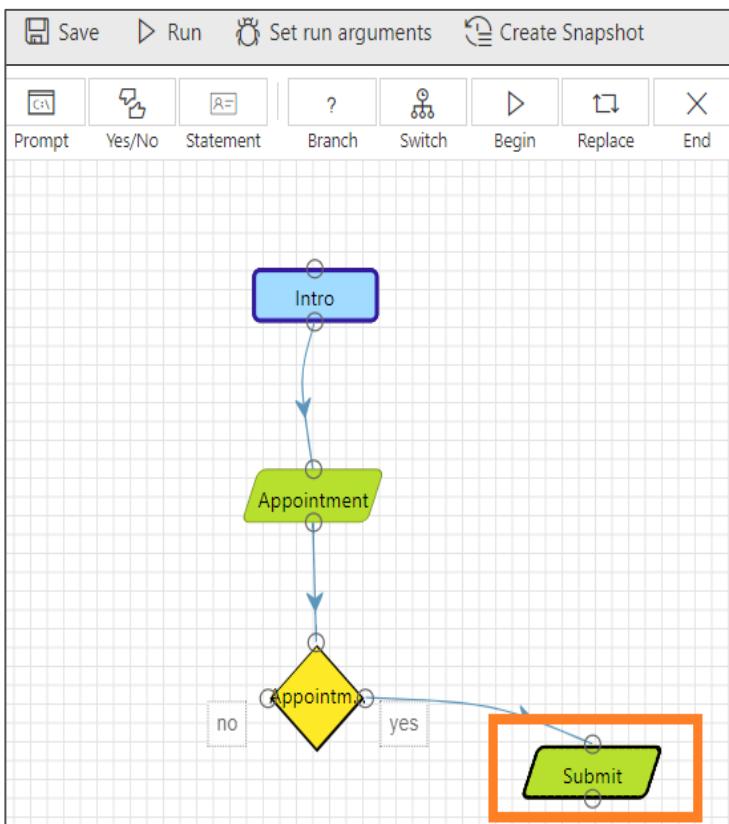
4. Add “**Prompt**” element to display the Form data (using Adaptive Card) to capture Patient **name**, **email**, and **phone** to create an appointment.

+ Select Prompt



Use Case: This section checks Submit the or **Live Agent** with the help of **AppointmentOrQueries**. It will redirect the message accordingly.

- Add “Submit” button



- Display Text: **Submit**
- Change Title to **Submit** as well
- Variable name: **formData**
- Variable Data Type: **Object**

Submit

Display text (either JavaScript string expression or plain Markdown):

Variable name (to store the input from the user):

Variable Data Type:

Is Active (javascript boolean expression, if empty will evaluate to true):

Maximum number of retries:

Cards **Message Metadata**

OK **Cancel**

1. Click Cards button > Add Card > Adaptive Card

Cards

Add Card

Layout:

OK **Cancel**

Refer the below attached file AdaptiveCardForAppointment.txt **and copy the json content and paste it in the card section.**



AdaptiveCardForApp
ointment.txt

Card

Card Type: AdaptiveCard

Design your own Adaptive Card

```
1 {
2   "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
3   "type": "AdaptiveCard",
4   "version": "1.0",
5   "body": [
6     {
7       "type": "ColumnSet",
8       "columns": [
9         {
10          "type": "Column",
11          "width": 2,
12          "items": [
13            {
14              "type": "TextBlock",
15              "text": "Tell us about yourself",
16              "weight": "bold",
17              "size": "medium"
18            },
19            {
20              "type": "TextBlock",
21              "text": "We just need a few more details to get you booked your Appointment!",
22              "isSubtle": true,
23              "wrap": true
24            },
25            {
26              "type": "TextBlock",
27            }
28          ]
29        }
30      ]
31    }
32  ]
33}
```

OK Cancel

2. Click Ok three times to get back to designer

You will be able to see the below conversation output when you run your bot.

Web Chat v4

en-us Just now

Bot Submit

Tell us about yourself

We just need a few more details to get you booked your Appointment!

Don't worry, we'll never share or sell your information.

Your name

Last, First

Your email

youremail@example.com

Phone Number

XXX.XXX.XXXX

Submit

Type your message



5. Add “Statement” element to display message.

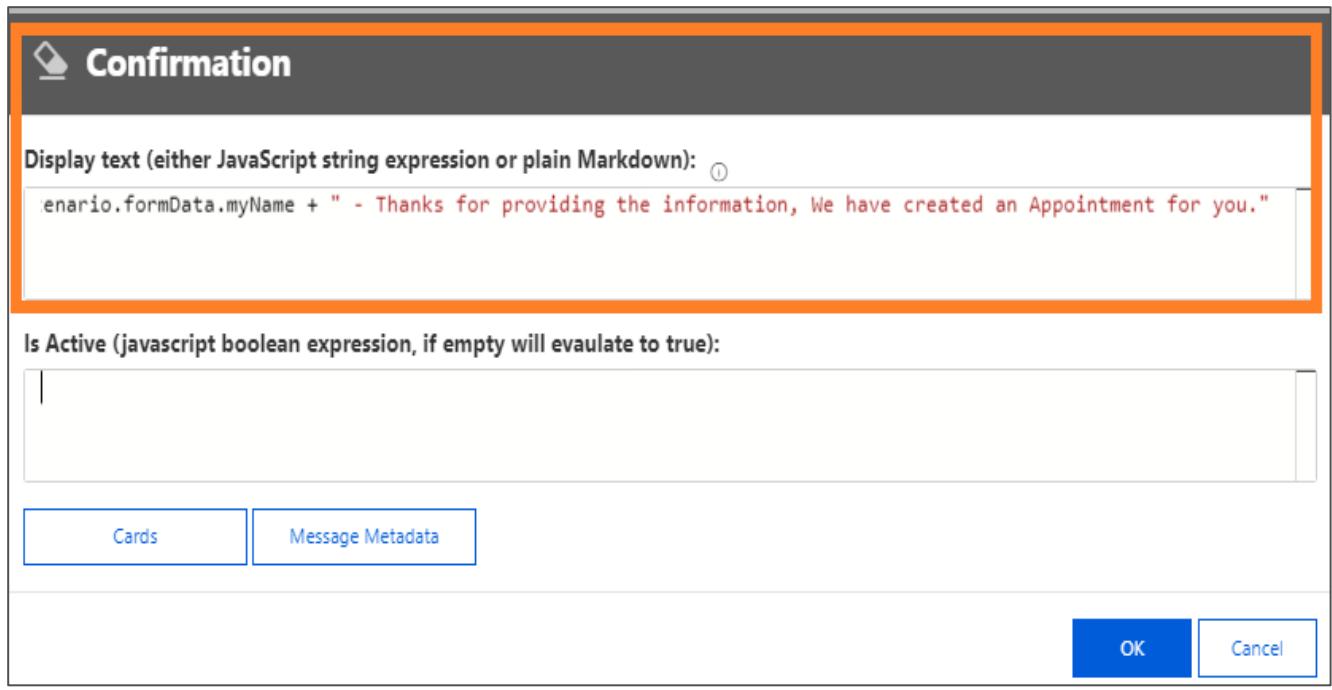
+ Select Statement

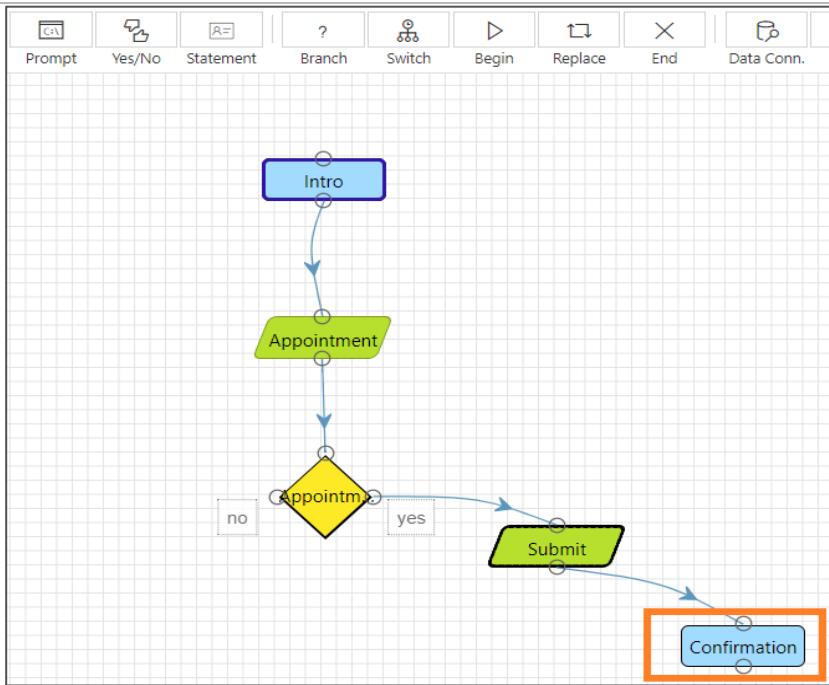


a. Add “Confirmation” button to display the below message to user.

b. Copy the below text and paste it in the display textbox.

```
scenario.formData.myName + " - Thanks for providing the information, we have created an  
Appointment for you."
```





Web Chat v4

en-us

Casey
Your email
Casey@contoso.com
Phone Number
324234234

Submit

Bot Casey - Thanks for providing the information, We have created an Appointment for you.
Just now

Type your message

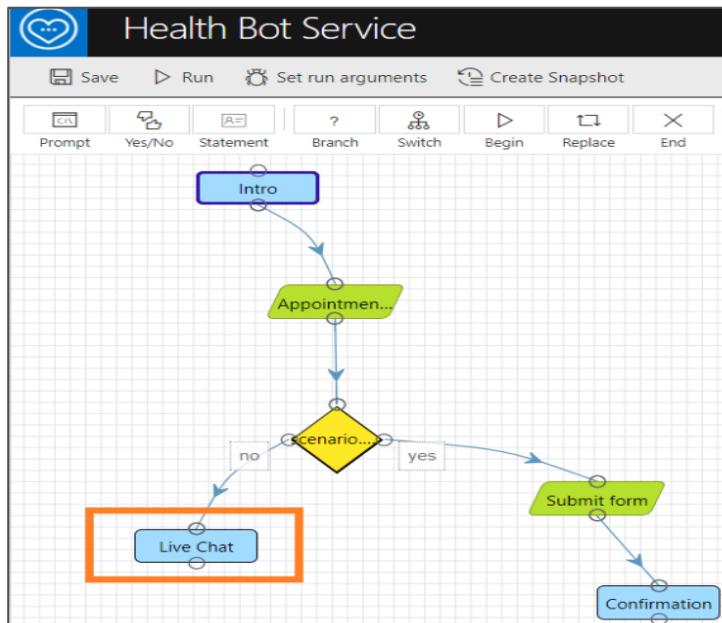
This screenshot shows a Microsoft Web Chat interface. On the left, there are input fields for "Casey", "Your email" (with value "Casey@contoso.com"), "Phone Number" (with value "324234234"), and a "Submit" button. On the right, a bot message is displayed: "Casey - Thanks for providing the information, We have created an Appointment for you." with a timestamp "Just now". At the bottom, there is a text input field with placeholder "Type your message".

6. Add “Statement” element to display the message and invoke Live Agent Action

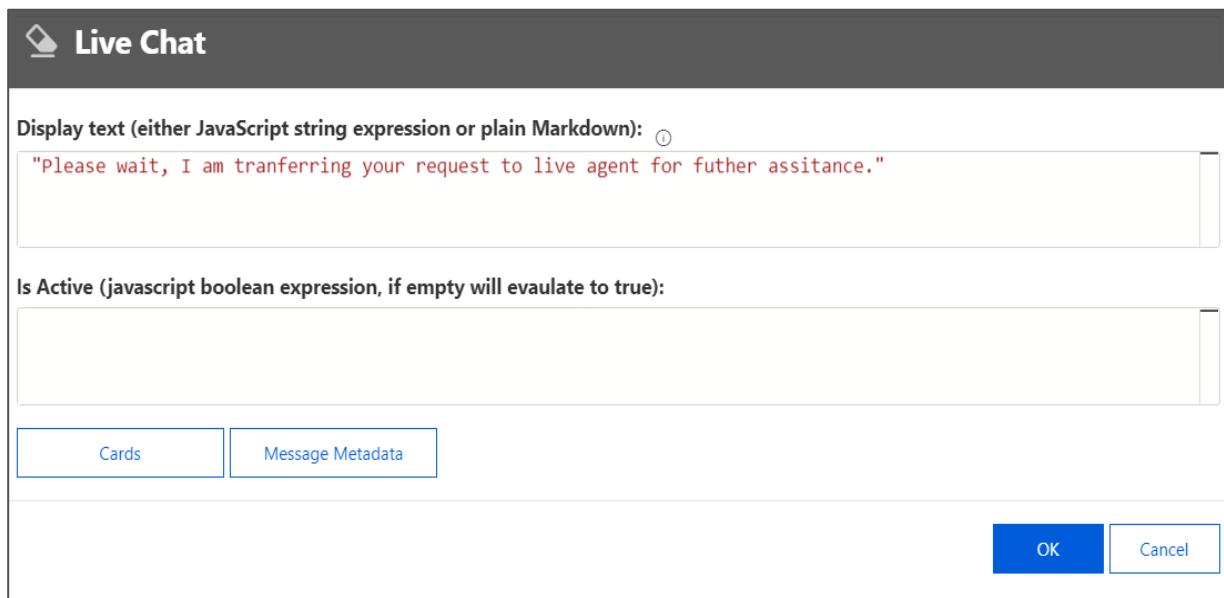
+ Select Statement



- a. Add “Live Chat” button to display the below message

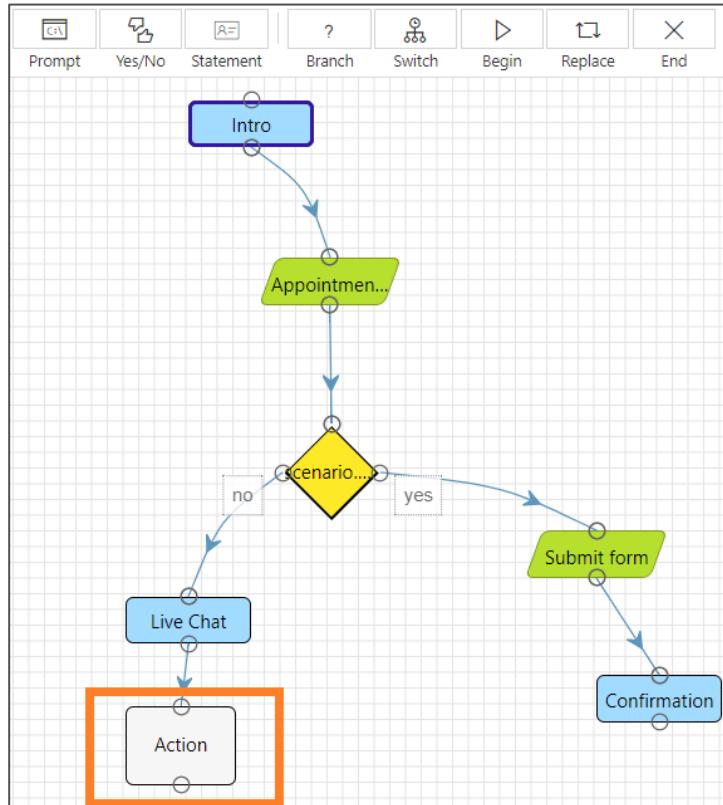


- b. Enter Display Text: “Please wait, I am transferring your request to a live agent for further assistance.”



- c. Click OK to return to the designer page.

7. Add “Action” element to invoke Omni Channel Live Agent Chat

+ Select Action element**a. Add Action button to trigger a message to Omnichannel Live Agent****b. Please add the following code in the action, which will trigger the Live agent chat:**

```
session.sendChannelData('Escalating...', {  
    "tags": JSON.stringify({type: "Escalate", context: {"EscalateToAgent": 1}})  
});
```

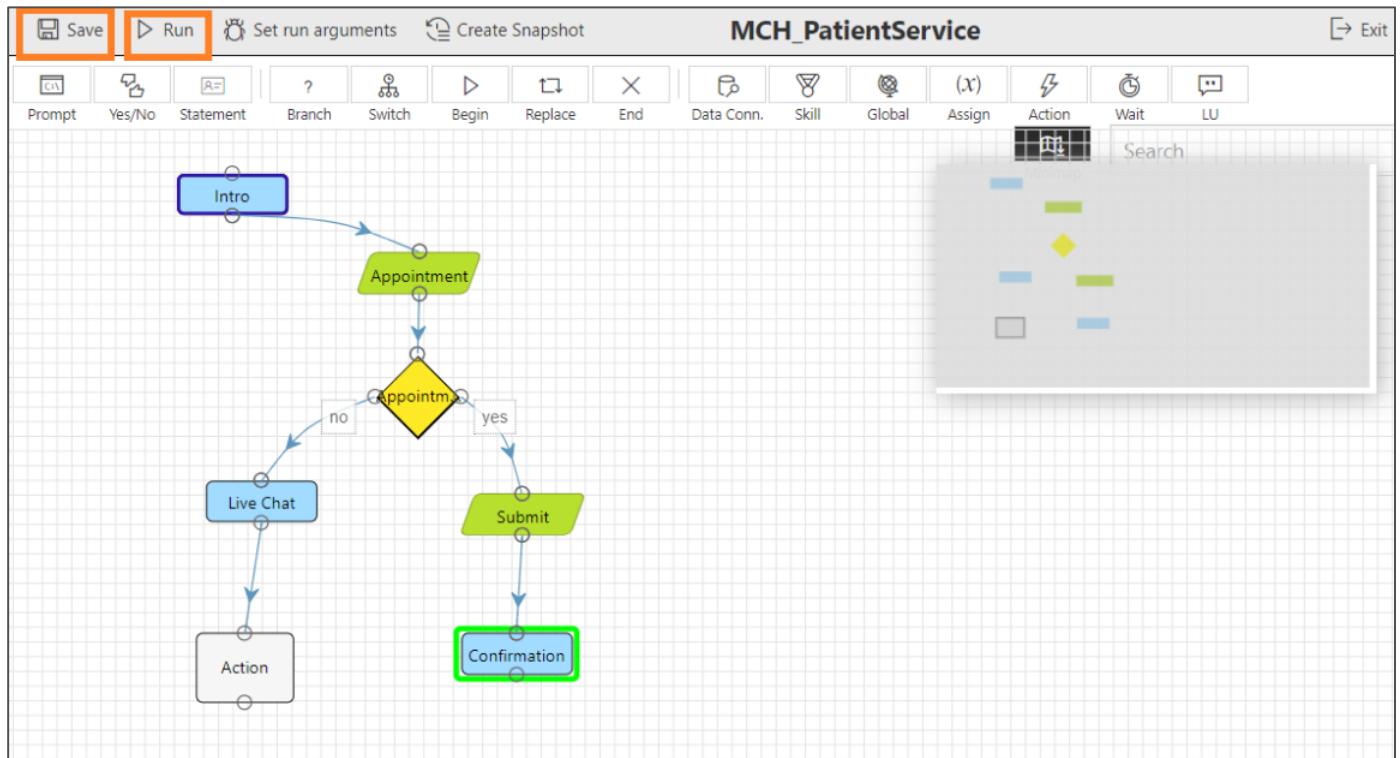
The screenshot shows the 'Action' dialog box. At the top, there's a title bar with the word 'Action'. Below it is a text input area labeled 'Action javascript Expression: ?'. Inside the input area, there is a code snippet:

```
1 session.sendChannelData('Escalating...', {  
2 |   "tags": JSON.stringify({type: "Escalate", context: {"EscalateToAgent": 1}})  
3 });|
```

At the bottom right of the dialog box are two buttons: 'OK' and 'Cancel'.

- c. Click Ok to return to the designer page.

3. This is how your final layout looks like for your Bot scenario. **Save and Run (Test) your Bot.**



4. Create another bot scenario "**MCH_Patient Service Welcome Scenario**" to invoke an **MCH_PatientService** scenario.

Active	Name	Scenario ID	Description
<input type="checkbox"/>	FlowConnection	FlowConnection	
<input checked="" type="checkbox"/>	MCH_Patient Service...	MCH_PatientServiceWelcomeScenario	Patient Service Welcome Scenario
<input type="checkbox"/>	MCH_PatientService	MCH_PatientService	

New Scenario

Name* ⓘ
MCH_Patient Service Welcome Scenario

Description ⓘ

Scenario ID* ⓘ
MCH_PatientServiceWelcomeScenario

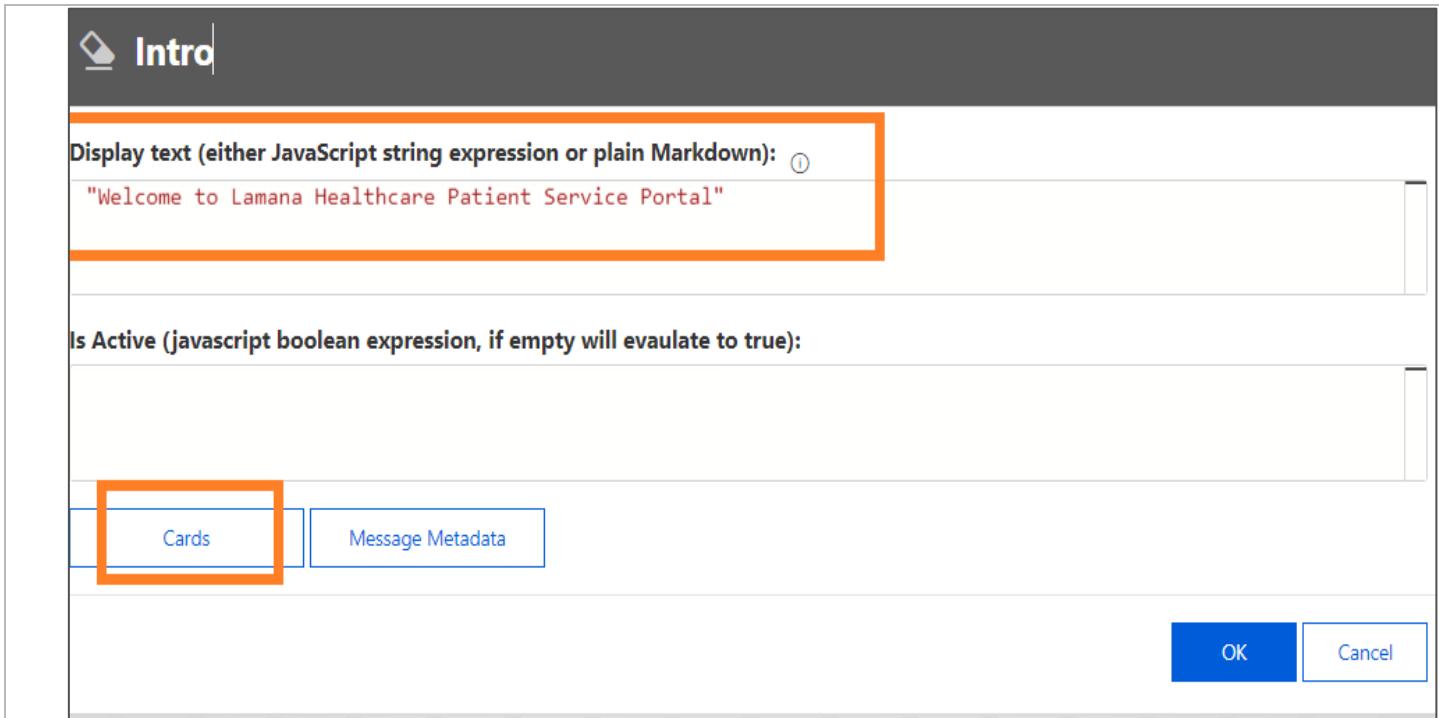
Returning Message ⓘ

Interrupting scenario ⓘ

Breaking scenario ⓘ

Create **Cancel**

1. Add “Statement” element to invoke “MCH_PatientService”**+ Select Statement**



- a. Enter display text as "Welcome to Lamna Healthcare Patient Service Portal!"
- b. Click Cards > Add Card



- c. Choose HeroCard
- d. Title: "Welcome to Lamna Healthcare Patient Service Portal"

Card

Card Type: HeroCard

Image Url: 

Title: "Welcome to Lamana Healthcare Patient Service Portal"

Sub Title:

Actions

Action type	Action value	Action title
imBack	"begin MCH_PatientService"	"Lamna Healthcare Support"

Add Action 

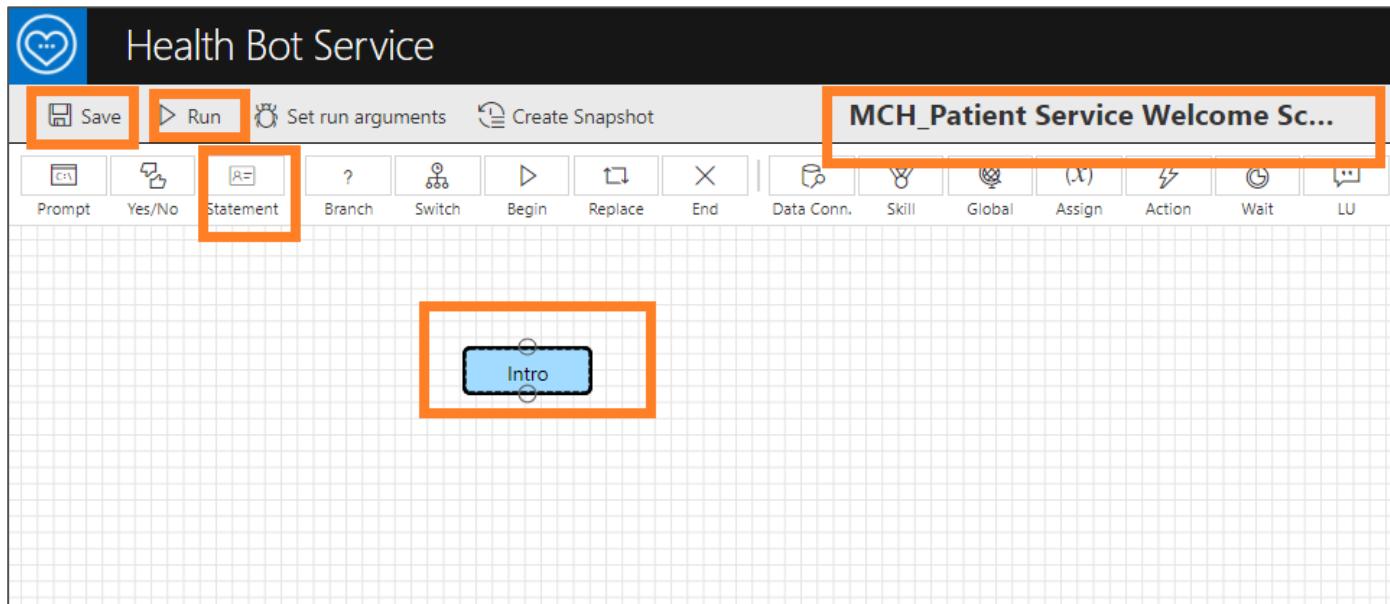
OK **Cancel**

6. Add Action with following details:

Action Type : imBack

Action Value : "begin MCH_PatientService"

Action Title : "Lamna Healthcare Support"



5. **Save** and **Run** to test your bot scenario **MCH_Patient Service Welcome Scenario** in the Web Chat



6. Configure scenario "**MCH_Patient Service Welcome Scenario**" as an "Automatic Welcome Scenario" so that bot will always trigger this scenario when user starts the conversation with **health bot** from Portal.

Configuration >> Conversation >> Interactions >> Update "Automatic Welcome Scenario" with "MCH_PatientServiceWelcome"

Congratulations! You have successfully created two bot scenarios designed with appointment adaptive cards and trigger Omnichannel Live Agent flow.

Task 2: Test your Health Bot Escalation from Power Apps Portal to Dynamics 365 Omnichannel

1. Login to Power Apps web portal and click to Open **Lamna Healthcare Patient Portal**.

Name	Modified	Owner
Lamna Healthcare Patient Portal	2 wk ago	SYSTEM
Patient Service Center	23 h ago	K Venkat
Customer Service Hub	1 wk ago	SYSTEM
Healthcare Administration	2 wk ago	K Venkat
Portal Management	2 wk ago	K Venkat

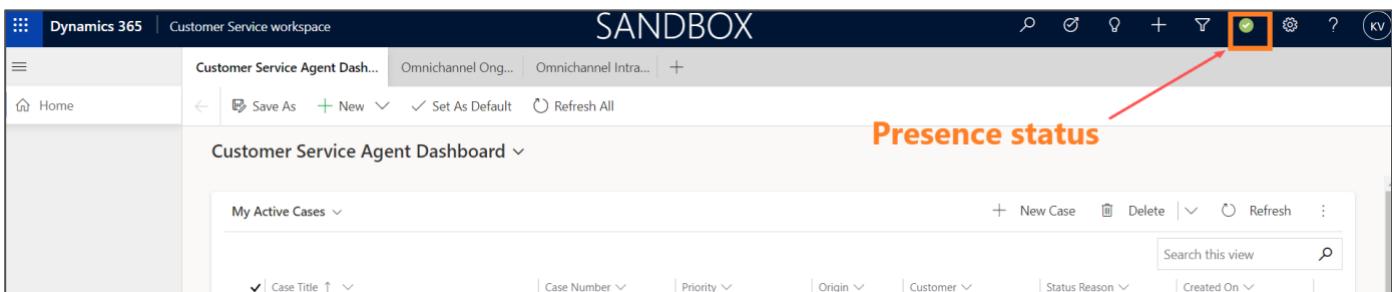
2. You should be able to see Health bot integrated with Customer Self-service portal (right side bottom of the page). When you click chat widget, bot will trigger a welcome scenario message (**MCH_Patient Service Welcome Scenario**).

The screenshot shows the Contoso Customer Self-Service portal. At the top, there is a navigation bar with links for Home, Knowledge Base, Forums, My Support, and Sign in. Below the header, the text "CONTOSO CUSTOMER SELF-SERVICE" is displayed over a background image of a medical office. A search bar with "All" and "Search" buttons is present. In the center, there is a section titled "Most Popular" with three categories: "Most Popular Articles", "Most Recent Articles", and "Top Rated Articles". At the bottom right, there is a "Let's Chat" button with a speech icon and the text "We're Online", which is highlighted with an orange rectangle.

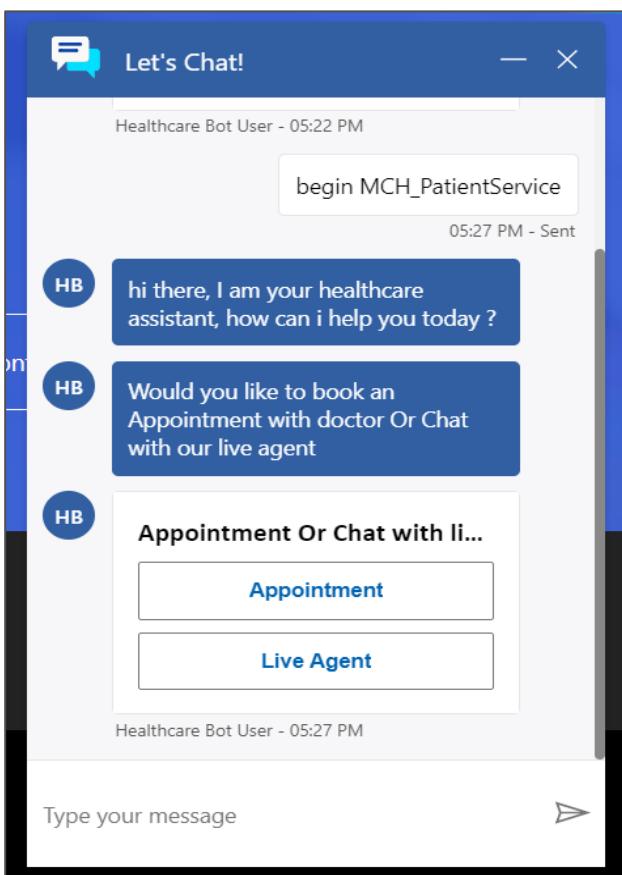
3. Open Dynamics 365 Omnichannel for Customer or Customer Service Workspace

The screenshot shows the Microsoft Power Apps portal. The left sidebar has navigation options: Home, Learn, Apps (selected), Create, Data, Flows, Chatbots, AI Builder, and Solutions. The main area is titled "Apps" and shows a list of apps under the "Apps" tab. The list includes: Lamna Healthcare Patient Portal (modified 2 wk ago), Patient Service Center (modified 23 h ago), Customer Service Hub (modified 1 wk ago), Healthcare Administration (modified 2 wk ago), Portal Management (modified 2 wk ago), FHIR Sync Agent Administration (modified 3 wk ago), and Customer Service workspace (modified 3 wk ago). The "Customer Service workspace" item is highlighted with an orange rectangle.

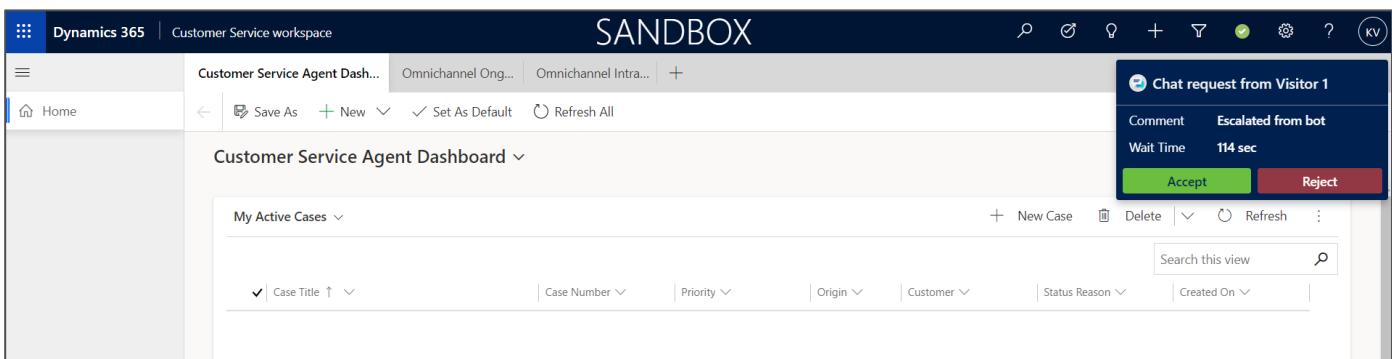
Note: Omnichannel for Customer Chat Widget will work only when you see the presence status is enabled (by default status would be **Active** with **green light** refer below picture to check the **presence status**)



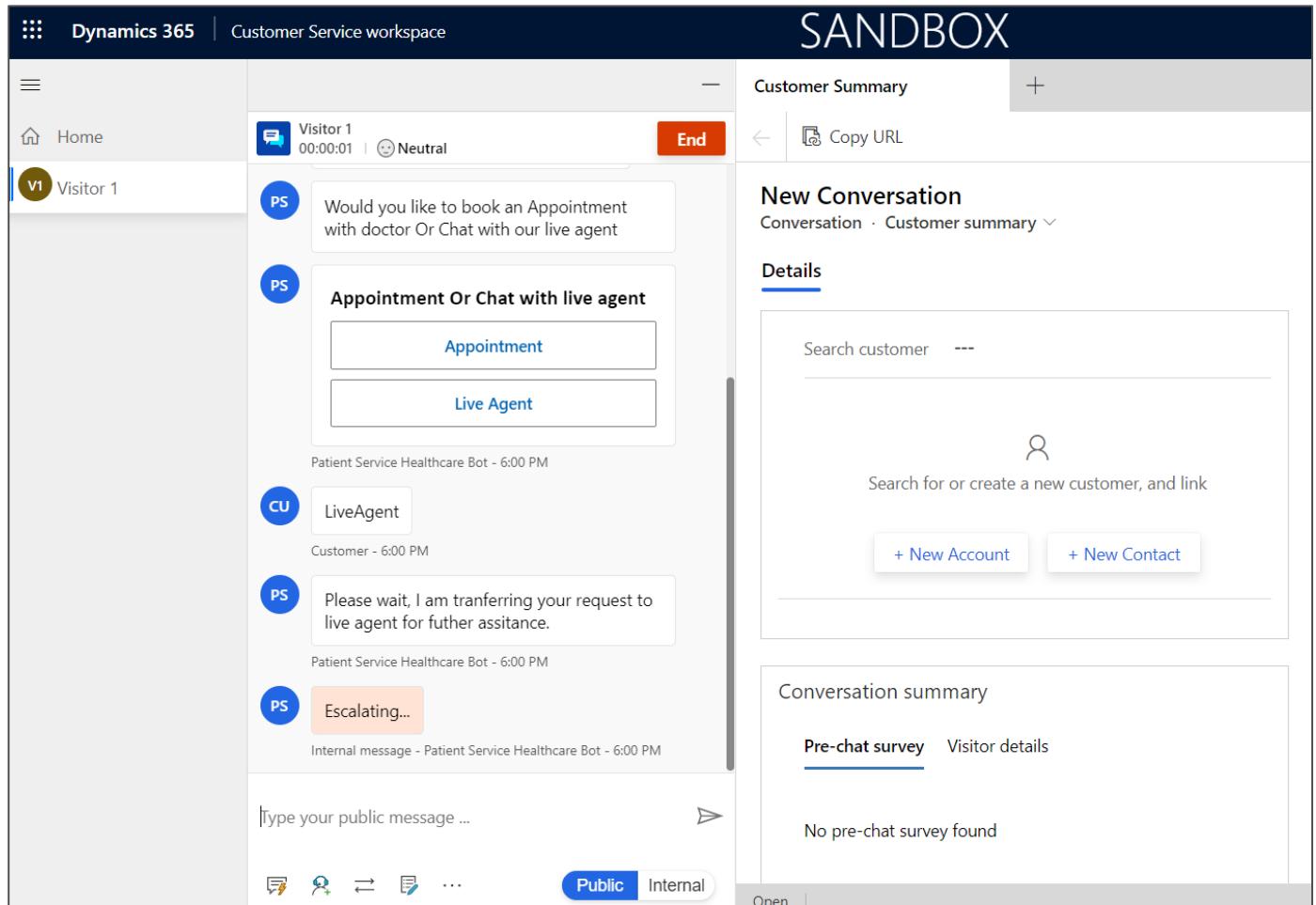
4. In the Health Bot Click “Live Agent” button to chat with “Omnichannel Live Agent”



5. “Omnichannel for Customer Service **Live Agent** receives an incoming notification with “Accept”/“Reject”, Click **Accept** to connect and chat with Consumer (In this case chat with **Patient**)



6. As soon as Live Chat Agent accepts the incoming chat notification, Omnichannel for Customer Service has opened a **Live Chat Widget** and Agent would be able to see the entire bot conversation with user and continue the chat conversation with user for further assistance.



Congratulations! You have successfully Tested end to end Patient to Bot conversation and Patient to Live Agent conversation from Power Apps Customer Service Portal.

Summary

In this lab, you have learned about how to setup Microsoft **Azure Health bot**, **Dynamics 365 Customer Service for Omnichannel**, **PowerApps Customer self-service portal**. Also, you have learned about how to integrate all these apps together to build an end-to-end scenario.

1. Install and setup Microsoft Health bot

- We have seen how to install Azure Health bot from Azure Market Place and create a new health bot instance.
- We have looked at the steps to obtain a **Bot Id** and update health bot settings to enable the "Bridge Connection" setting to integrate health bot with Dynamics 365 Omnichannel for Customer Service
- Created two health bot scenarios "**MCH_PatientService**" and "**MCH_Patient Service Welcome Scenario**" and integrated them.

2. Dynamics 365 for Customer Service configuration for Health bot

- Created Azure AD App
- Created Dynamics 365 Application User using with Azure AD app and added required security roles.
- Created Dynamics 365 Omnichannel Context Variables, Routing rules, Queues for Bot and Agent.

3. Power Apps Customer self-service portal Configuration and embed Health bot

- Created Customer self-service portal (**Lamna Healthcare Patient Service Portal**)
- Configured Omnichannel for Live Agent chat widget script into **Portal Management** app

Lab Survey

We would appreciate your feedback on Microsoft Cloud for Healthcare in a Day and this hands-on-lab, such as the quality of documentation and the usefulness of the learning experience.

Please use the survey at <https://aka.ms/MCHIADSurvey> to share your feedback.

You may provide feedback for each module as you complete it or at the end once you have completed all the modules. Thank you!

Terms of Use

© 2021 Microsoft. All rights reserved.

By using this hands-on lab, you agree to the following terms:

The technology/functionality described in this hands-on lab is provided by Microsoft Corporation in a "sandbox" testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the hands-on lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. Without written permission, you may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this hands-on lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION WITHOUT WRITTEN PERMISSION IS EXPRESSLY PROHIBITED.

THIS HANDS-ON LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCITONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

FEEDBACK If you give feedback about the technology features, functionality and/or concepts described in this hands-on lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies, and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER This lab contains only a portion of new features and enhancements in Microsoft Power BI. Some of the features might change in future releases of the product.