

**Webex Contact Center Enterprise digital
channels implementation and
troubleshooting powered by Webex
Connect
LTRCCT-2003**

Speakers: Joshua Raja / Robert W. Rogier

Learning Objectives

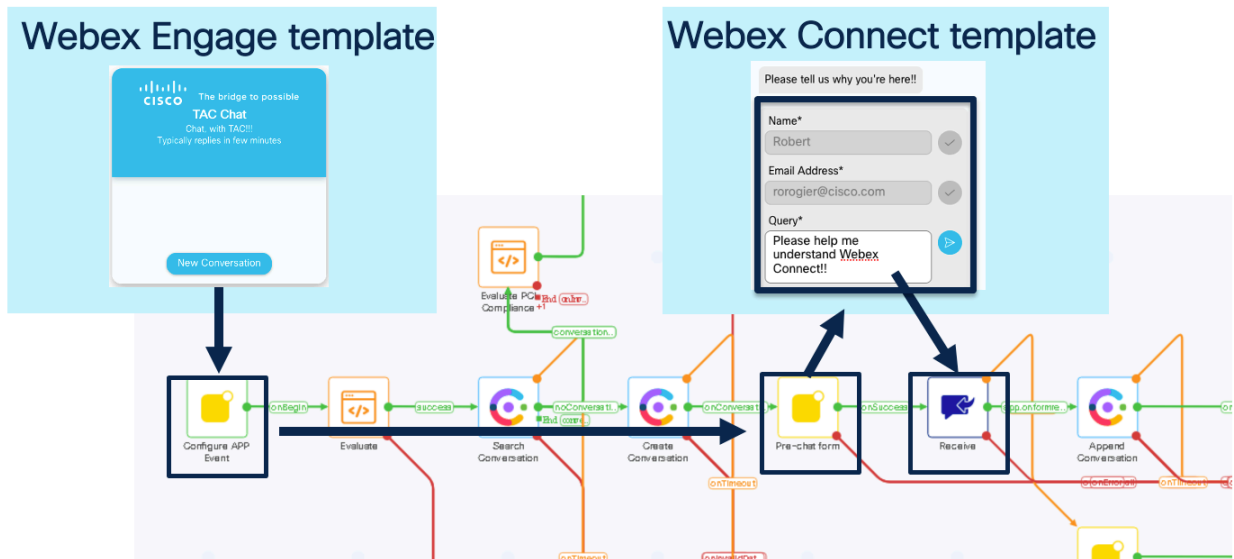
Upon completion of this lab, you will be able to:

- Understand how to create a Chat application and register it with Engage
- Know where to create questions used in Chat entry points
- Know what documentation resources are available

Scenario

This lab is designed to introduce the audience to the digital channels (Webex Connect) platform, its architecture, and its provisioning. This lab will show how to create the Live Chat application as well as the questions you will ask customers.

Live Chat Flows and Templates



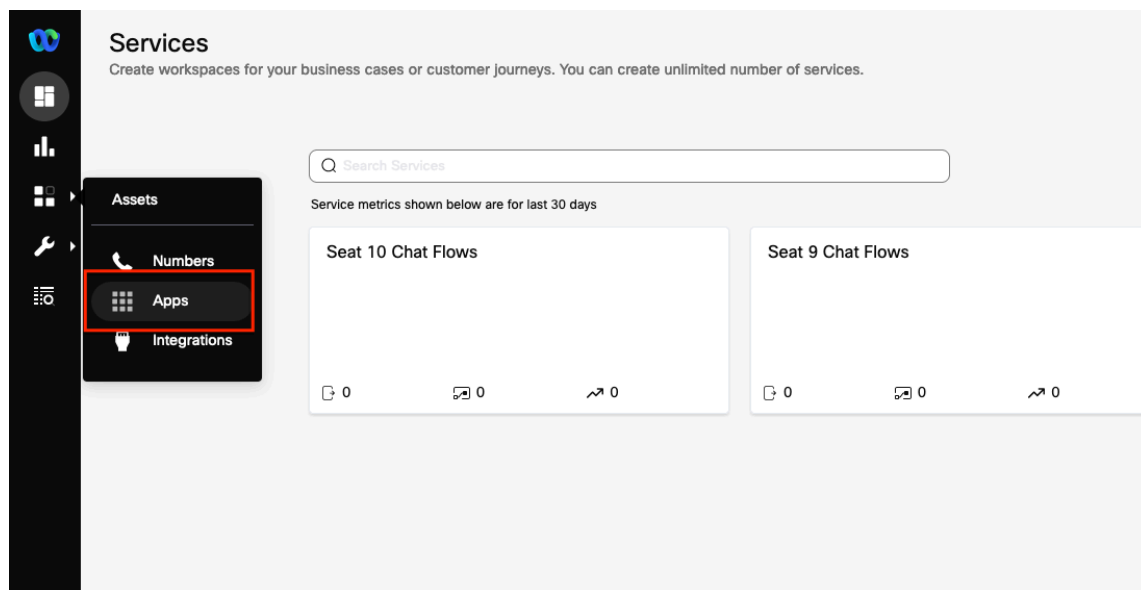
There are two templates in Webex Connect. The first is the Webex Engage template. This is the “look and feel” of your chat entry point and can have custom colors, icons, etc. The second is the Webex Connect template. This template defines the questions you ask to customers to know what they are opening the chat for so your agents can help them.

Task 1: Log in to Webex Connect Tenant

Step 1. Log in to Webex Connect

You will need the Connect URL from the Credentials document. Open a browser and enter this URL, then log in using the User and Password found in this document. You can access this site from any computer. You do not need to use the workstation to access this site. You can access this from any computer you wish.

Step 2. Navigate to Assets > Apps

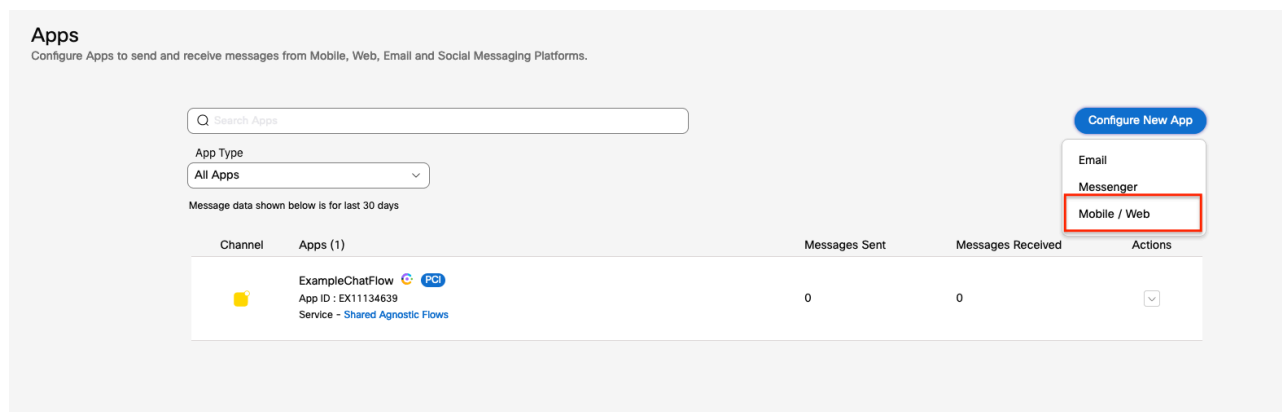


Task 2: Create the Chat App

In this task, you will create the chat application to use. There are three different types of apps you can create, Email, Messenger, and Mobile / Web. CCE currently supports Email and Mobile / Web apps. Messenger support is coming very soon.

Help Link: <https://help.imiconnect.io/docs/live-chat>

Step 1. Select Configure New App -> Mobile / Web



Step 2. Fill out the form

There are only a few details required for the Chat application. Since we are using a shared tenant, you will need to ensure that the names are unique. Before you begin, you need to check the Credentials Document for the Service that is assigned to you. This will be in the format of Seat # Chat Flows. This will be used when you register the App to Engage.

Fill out the form as follows:

1. Name: Provide the name in the format of, Seat#ChatEntryPoint.
2. Live Chat / In-App Messaging: Ensure this is selected to true.
3. Primary/Secondary Transport Protocol: Select one as MQTT and the other as Web Socket. It does not matter which is selected a primary.
4. Use Secured Port: Ensure this is selected to true.

--- Continued next page ---

< **Configure New App - Mobile & Web**
 You can use our Android, iOS, Cordova, and/or JavaScript SDKs for Push and Live Chat / In-App messaging to support these channels in your mobile and web apps. Refer SDK quick start guides for [more](#) information.

Name ⓘ
 Seat1ChatEntryPoint

Register To Webex Engage Configure Outbound Webhooks DOCS ⓘ

Access Credentials
 Please use this key and the appid to initialize the SDK. This combination is used to uniquely identify your App while communicating with Webex Connect.

Client Key
 WZUlw0b0 ⓘ

Live Chat / In-App Messaging ☒
 Live Chat / In-App Messaging provides a fast, secure and bi-directional channel of communication on mobile and web Apps. Refer help docs for [more](#) info.

Primary Transport Protocol Secondary Transport Protocol
 MQTT Web Socket

Message delivery is attempted via primary protocol and in case of failure the delivery falls back on lower level of preferences. To know more, [click here](#).

☒ Use Secured Port ⓘ
☐ Enable Payload Encryption ⓘ

> Advanced Settings

Select Save at the bottom of the page once you have filled out all the required options.

Step 3. Register the App to Engage

The app must be registered to Engage. Remember, that Engage is where the template is hosted. This is where all the look and feel items are configured. When you register the application to Engage you must choose the Service to register the App to. This cannot be changed after it is registered. In addition, you cannot delete the App from Connect as it leaves the resources in Engage. Ensure that you have carefully chosen the name and service as this is an irreversible operation.

1. Select **Register to Webex Engage** at the top of the form.

< **Manage - Mobile & Web**
 You can use our Android, iOS, Cordova, and/or JavaScript SDKs for Push and Live Chat / In-App messaging to support these channels in your mobile and web apps. Refer SDK quick start guides for [more](#) information.

Name ⓘ
 Seat1ChatEntryPoint

Register to Webex Engage Configure Outbound Webhooks DOCS ⓘ

Access Credentials
 Please use this key and the appid to initialize the SDK. This combination is used to uniquely identify your App while communicating with Webex Connect.

Client Key
 WZUlw0b0 ⓘ

Live Chat / In-App Messaging ☒
 Live Chat / In-App Messaging provides a fast, secure and bi-directional channel of communication on mobile and web Apps. Refer help docs for [more](#) info.

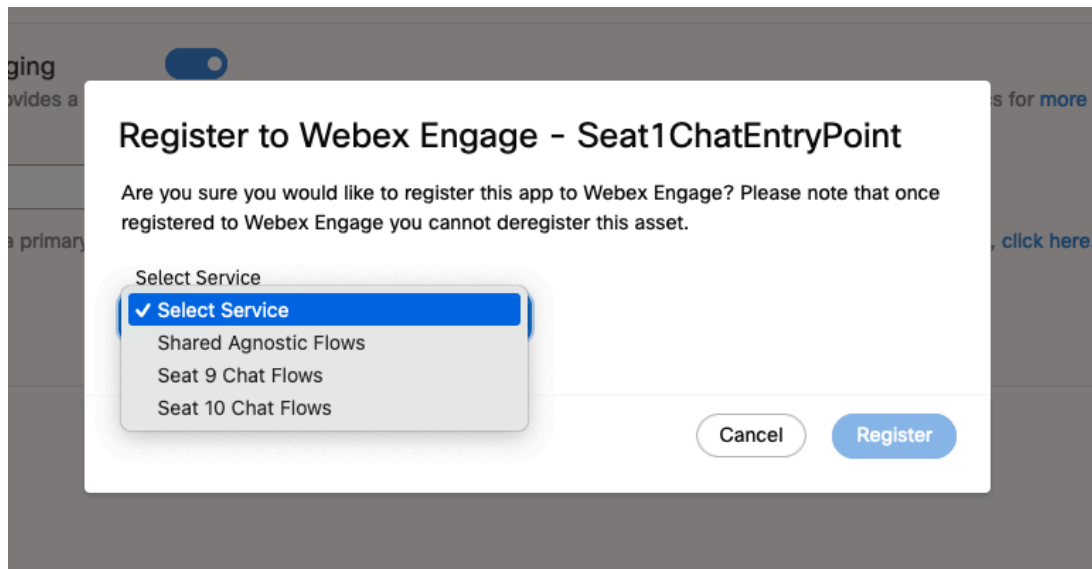
Primary Transport Protocol Secondary Transport Protocol
 MQTT Web Socket

Message delivery is attempted via primary protocol and in case of failure the delivery falls back on lower level of preferences. To know more, [click here](#).

☒ Use Secured Port ⓘ
☐ Enable Payload Encryption ⓘ

> Advanced Settings

2. In the pop-up form, select the service you want to register to. For this class, use the service in your Credentials document.

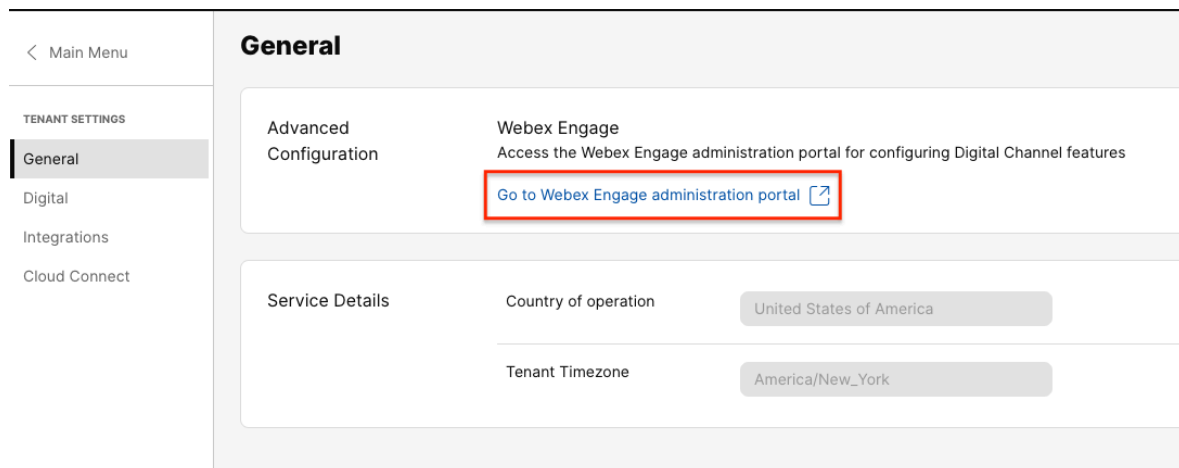


3. Select the “Register” button.

NOTE: You *must* get this correct. Once the app is registered, it is not possible to update or change the name or the service to which it is registered. Please check before you select the Register button.

Step 4. Open the Engage portal

Back in your Control Hub, ensure you are in the **General** menu under **Contact Center**, then select the link, **Go to Webex Engage administration portal**.



Step 5. Review App in Engage

Select the **Assets** link in the menu bar, then ensure that **Channel Assets** is selected. Find the asset you created above, then select the edit icon.

--- Continued next page ---

Webex Contact Center

Channel Assets

Manage channel assets that can be used to receive messages from your customers. Channel assets created in the Admin console can be shared across different teams.

All Apple Messages for Business Email Facebook Livechat SMS WhatsApp Google Messages for Business

Search

Channel	Asset Details	Asset Name	ID	Added On	Priority	Action
LiveChat	ExampleChatFlow	ExampleChatFlow	03bf1a6c-2515-4754-ac33-92fef248b58b	11 May 2024 06:46:48		

Showing 1 to 1 of 1 entries

Take a few minutes to look around. You will return to this in another lab where you will be able to make this chat interface look how you like.

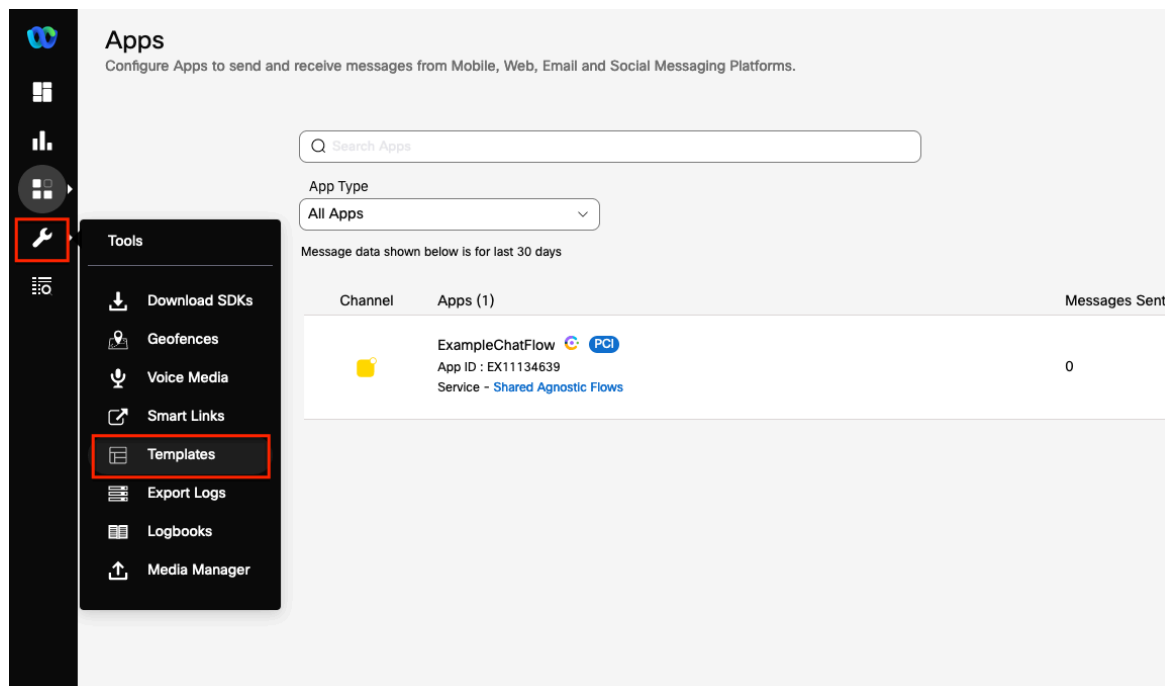
Task 3. Create Questions for your chat

Every chat entry point needs to ask the customer for some information. Depending on the business requirements, this may be as simple as an email address and phone number and may be more complex. The answers to these questions can be stored and used in the flows. This class will not show this usage but will instead focus on a basic chat.


Ensure that you are logged into your Connect tenant before you start the steps in this task.

Step 1. Select Tools > Templates

We need to go into the templates configuration section of Webex Connect.

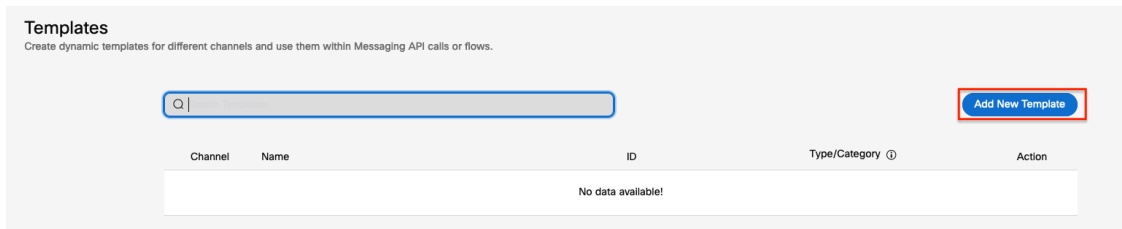


The screenshot shows the 'Apps' configuration page in Webex Connect. The left sidebar has a 'Tools' menu with 'Templates' highlighted. The main area shows a table of configured apps.

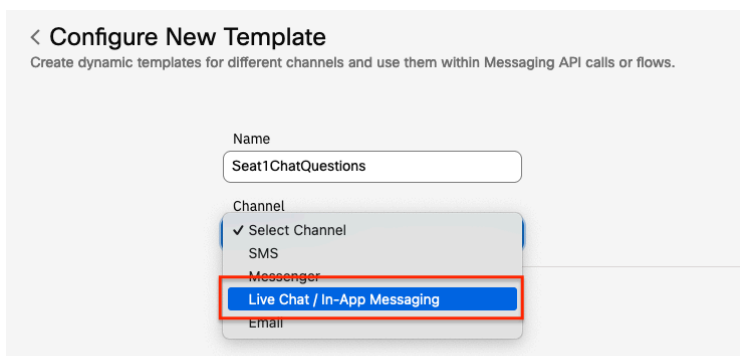
Channel	Apps (1)	Messages Sent
	<div>ExampleChatFlow </div> <div>App ID : EX11134639</div> <div>Service - Shared Agnostic Flows</div>	0

Step 2. Add initial template configuration

Select the **Add New Template** button at the top of the page.



Provide a name for the template. For this class, give the name in the format of *Seat#ChatQuestions*. Select *Live Chat / In-App Messaging* from the Channel drop-down.



Step 3. Create the questions

For this example, we will use a Form to create the questions. There are other alternatives that you can find in the help.imiconnect.io documentation.

- Select **Form** from the Message Type drop-down. Notice that you will see a preview of the form. As you add questions, you will see how they will look to the customer.

--- Continued next page ---

- Create a Title – The title is how you will greet your customers. This is the first thing that the users of this form will see. Create a title that you like.
- Add the Form Fields – You will repeat the next steps several times.
Note: the order that you create the questions is the order in which they will be asked to the customer. You must delete and recreate the questions if you find they are out of order.
 - Type – There are several pre-defined types that can do validation.
 - Name – This is how you will see the value of the field in the flows if you wish later.
 - Label – This is what is shown to the customer to prompt for information. Do not include any punctuation.
 - Mandatory Field – Select this for any field which the customer must provide

Add Field

- Select Save once you have added all the fields you wish

Message Configuration

Message Type ①
Form

Title ①
Configure your own form title here. This will be shown to user before displaying the form fields.
Welcome to Cisco Live chat! We need some initial information from you to continue.
Characters: 82/300

Form Fields
Qualify your contact's query by capturing details and use these attributes to route.
[+ Add Field](#)

Parameter	Name	Mandatory Field	Action
Email	Email	true	
Name	Name	true	
Text	CustQuestion	false	

Save

Preview

Welcome to Cisco Live chat! We need some initial information from you to continue.

What is your email

What name do you prefer we use

How can we help you today

Note: For this lab, you MUST have the following fields defined:

- Customer's Email
 - Type: Email
 - Name: Email
 - Mandatory: True
- Customer's Name
 - Type: Name
 - Name: Name
 - Mandatory: True

You may create any additional questions you wish, but these two must exist for proper routing.

You have now completed this portion of the lab.