

LTRCCT-2813

Mastering Demo-Driven Customer Journeys with Webex Customer Experience

Seb Huber - Solutions Engineer
Christian Wolf - Solutions Engineer
Jan Jäkel - Solutions Engineer

Table of Contents

Disclaimer.....	1
Scenario.....	1
Module 0 – Preparation.....	2
0.1 User Accounts.....	2
0.2 PSTN Numbers.....	3
0.3 Webex Connect.....	4
0.4 Webex Engage.....	4
0.5 Journey Data Service.....	5
0.6 Idle/Wrap-Up Codes.....	6
0.7 Airtable.....	7
0.8 Glitch.....	10
0.9 Website creation.....	11
Module 1 – Voice Flow Creation.....	15
1.1 Wrap Up Codes	16
1.2 Business Hours	17
1.3 Multimedia Profile	19
1.4 Desktop Layout.....	20
1.5 Teams.....	22
1.6 Queues.....	23
1.7 Flows.....	26
1.8 Channels (Entry Point).....	30
1.9 Desktop Profile	34
1.10 User Profile.....	39
1.11 Agent Configuration	41
1.12 Testing the Agent Desktop	44
1.13 Addon: Caller Identification.....	55
Module 2 – Web Chat	63
2.1 Webex Connect Web Chat Channel Configuration.....	64
2.2 Control Hub Configuration	68
2.3 Webex Connect: AI Agent Studio	76
2.4 Webex Connect: Web Chat Workflow	81
2.5 Testing the Web Chat Demo	101
2.6 Addon: Ongoing bot curation	107
2.7 Addon: Flow enhancements (verification loop and gen AI summaries).....	112
Module 3 – Digital Channels.....	118
3.1 Webex Connect: WhatsApp Workflow.....	119
3.2 Testing the WhatsApp demo	137
3.3 Addon: Flow enhancements (XXXXX)	140

Disclaimer

This training document is to familiarize with Webex Customer Experience. Although the lab design and configuration examples could be used as a reference, it's not designed for any real, production-environment deployment, and thus not all recommended features are used, or enabled optimally. For the design related questions please contact your representative at Cisco, or a Cisco partner.

Scenario

In this lab activity, you will step into the role of a **Solutions Engineer at Better Bikes Inc.**, a leading (fictitious) company in the cycling industry. Your mission is to create a seamless and efficient **customer journey** using **Cisco Webex Customer Experience** solutions, helping your company deliver outstanding support and engagement.

Your journey begins with a well-structured **voice flow**, powered by **Webex Contact Center (WxCC)**, ensuring that customers can easily connect with your support team. However, modern customer expectations go beyond traditional call centers. To enhance accessibility and convenience, you will implement a **Web Chat widget** on your company's website, allowing customers to ask questions and receive assistance in real time.

As digital transformation continues to shape customer interactions, offering **multichannel support** becomes essential. You will take the next step by integrating **WhatsApp**, making it even easier for customers to engage with your company on the channels they use daily. This omnichannel approach not only improves response times but also creates a more personalized and efficient customer experience.

By the end of this lab, you will have a comprehensive understanding of the various **components, tools, and applications** involved in building a modern customer support system. While there is no universal, one-size-fits-all blueprint, this experience will equip you with the knowledge and insights to explore **customized solutions** that best fit the needs of your company and/or customers.

Module 0 – Preparation

To ensure that the exercises run smoothly, we have already configured some elements in advance, we detail these from Module 0.1 to 0.8, please note that 0.9 requires you to follow some steps to create an account and order for the lab.

Please note that throughout the documentation we will use the term 'PodXX_...' as an identifier, in which the 'XX' indicates where you need to replace the X's with the number of the pod you are issued with.

0.1 User Accounts

For each pod, we have configured 3 types of users "PodXX_User", "PodXX_Agent", and "PodXX_Super".

"C1sc0Live25!" has been set as password for every account.

"PodXX_User" will be used for all configurations and administrative tasks.

"PodXX_Agent" is used for Contact Center Agent Desktop login.

"PodXX_Super" is the Supervisor account of your Contact Center.

These users have different permissions in Control Hub and Contact Center. The following permission and functions have been assigned:

	PodXX_User	PodXX_Agent	PodXX_Super
Control Hub Admin	X		
Webex Contact Center Admin	X		
Premium Contact Center Agent		X	
Contact Center Supervisor			X
Webex Connect Admin	X		
Webex Engage Admin	X		
webmail.webex-lab.com	X		
glitch.com	X		

0.2 PSTN Numbers

To enable inbound dialing via PSTN, we have provided public phone numbers for your PodXX_User and PodXX Service Queue. The numbers provided for your pod can be found in the following table:

Pod Number	PodXX_User	PodXX Service Queue
01	+442046200494	+442046200496
02	+442046200501	+442046200506
03	+442046200507	+442046200517
04	+442046200523	+442046200524
05	+442046200526	+442046200527
06	+442046200528	+442046200530
07	+442046200532	+442046200533
08	+442046200534	+442046200538
09	+442046200549	+442046200629
10	+442046200852	+442046200941
11	+442046200943	+442046200947
12	+442046200949	+442046200951
13	+442046200952	+442046200954
14	+442046200955	+442046200956
15	+442046200961	+442046200974
16	+442046200976	+442046200977
17	+442046200978	+442046200980
18	+442046200981	+442046200982
19	+442046200986	+442046200989
20	+442046200991	+442046200994

0.3 Webex Connect

Webex Connect, part of Cisco's Webex suite, is a cloud-based platform for enterprise communication and collaboration. It integrates various channels—SMS, email, social media, and chat apps—into one interface, streamlining customer interactions.

The platform supports automated workflows, chatbots, and omnichannel strategies, making it ideal for improving customer service and engagement. With analytics and reporting tools, businesses can enhance communication processes and deliver seamless, consistent customer experiences.

For your reference:

0.4 Webex Engage

Webex Engage, part of Cisco's Webex suite, is a cloud-based platform for enhancing customer engagement through multichannel communication. It supports interactions via voice, video, chat, email, and social media.

Designed for customer service teams, it offers features like automated workflows, chatbots, and CRM integration to deliver personalized, efficient experiences and improve response times. With a unified interface and analytics tools, Webex Engage helps businesses maintain consistent communication, strengthen customer relationships, and optimize engagement strategies.

0.5 Journey Data Service

Customer Journey Data Service ([CJDS](#)) is an API-first service that enables organizations to:

Listen: Integrate with any data source or third-party applications to listen to disparate data sources.

Identify: Create a dynamic customer profile capturing propensity drivers, such as a customer's preferred mode of communication or preferred language.

Analyze: Apply different aggregation techniques to all customer data collected.

Act: Use the data and insights within CJDS to dynamically change the flow within Webex Contact Center Flow Control and personalize the customer experience at a granular level.

These insights are visible to customer-facing teams in real time through Agent Desktop.

Customer Journey Data service (CJDS) is a next-generation customer journey management service empowering organizations to go from Data to Insights to Action. CJDS enables businesses to capture customer journeys across any channels/application, identify insights and take real-time actions to provide an excellent customer experience.

CJDS needs a Webex Contact Center OAuth Token to use the API. In this lab we have established a token management. The Documentation can be found on [GitHub](#).

Also, you can find an example of how to inject CJDS [here](#).

0.6 Idle/Wrap-Up Codes

In preparation for presence synchronization, different codes have already been created and assigned to the presence states.

Idle/Wrap-up Codes					
	Search by name	Active	8 Idle/Wrap-up Codes	Create Idle/Wrap-up Code	
Idle/Wrap-up code	Description	Idle/Wrap-up code type	Default	Status	Last modified
away		Idle Code	No	● Active	December 02, 20...
DND		Idle Code	No	● Active	December 02, 20...
in a Calendar Mee...		Idle Code	No	● Active	December 02, 20...
in a external Call		Idle Code	No	● Active	December 02, 20...
in an Meeting		Idle Code	No	● Active	December 02, 20...
in presentation		Idle Code	No	● Active	December 02, 20...
Meeting	Default idle code	Idle Code	Yes	● Active	October 30, 2024...
Wrap Up	Default wrap-up c...	Wrap-up Code	Yes	● Active	December 04, 20...

Under Desktop in Contact Center main menu, these codes have been assigned to the appropriate Presence States.

Webex App
You can control access to these features for your agents inside Desktop profiles.

Display user details

State synchronization

Map Webex Availability States with Webex Contact Center's Idle codes. For more information on mapping [Click here](#).

Webex Availability States	Webex Contact Center Idle c...
Do not disturb, Quiet hours	<input type="button" value="DND"/>
In a calendar meeting	<input type="button" value="in a Calendar Meeting"/>
In a meeting	<input type="button" value="in an Meeting"/>
On a call	<input type="button" value="in a external Call"/>
Out of office	<input type="button" value="away"/>
Presenting	<input type="button" value="in presentation"/>

0.7 Airtable

The screenshot shows the Airtable homepage with a central banner for "Digital operations for the AI era". Below the banner is a sub-headline: "Create modern business apps to manage and automate critical processes." At the top, there's a navigation bar with the Airtable logo, "Platform > Solutions > Resources > Enterprise Pricing", and buttons for "Contact Sales", "Sign up for free", and "Sign in". On the right side of the banner, there's a sidebar titled "Re Zelos" with sections for "Launch", "Integrations", "Reporting", and "Appviews". The main area displays a "Launch - Executive dashboard" with several charts and data points. One chart shows "Total units: 51,250" and "Total sales: \$4,612,500". Another chart shows "Zelos sales: \$1,153,125" and "Partner sales: \$3,459,375". A callout box highlights the "Trail Summit 500" launch, stating: "How is the Trail Summit 500 launch performing across markets? This launch is exceeding expectations. We hit our year-one sales target globally. As we move into week two, customer acquisition is slowing slightly in the EMEA region. Consider implementing promotional discounts to offset new dips in the following months." Below the dashboard, there are "Sign up for free" and "Contact Sales" buttons.

Airtable is a collaboration platform that combines the simplicity and user-friendliness of a spreadsheet with the powerful functionality of a database. Its main purpose is to enable individuals and teams to organize, track, and manage various types of information and projects in a customizable, visually intuitive format. Users can create and share databases that support a range of views (like grid, calendar, gallery, and kanban), facilitating seamless collaboration and integration with other tools and services, making it suitable for a variety of tasks such as project management, content planning, and customer relationship management.

Add that the tool is a cloud-based, open-access solution, makes this an ideal tool for using in lab scenarios, such as our own, and so we have integrated Airtable with our solutions in this lab for the purpose of imitating a database. We have integrated several CRUD (create, read, update & delete) functions into our lab, using Webex Connect's Custom Nodes.

Below, you can see a screenshot of the list of functions, some of which you will use in this lab.



Manage Custom Node
Use custom node to integrate with your REST/SOAP API service. These are available as nodes in workflow.

Request Details

POST	Account: Create	Request Name Account: Create	Connection Timeout (Ms) 15000
PATCH	Order: Update r...	Request Timeout (Ms) 15000	Resource URL https://api.airtable.com/v0/appZlsm4WdS7DZUEF/accounts
PATCH	Order: Update r...	Type Post	Parse Variables
DELETE	Order: Delete a ...		
GET	Account: Retrieve		
POST	Order: Create sl...		
PATCH	Account: Updat...		
DELETE	Account: Delete		
GET	Order: Retrieve		
PATCH	Order: Update d...		
GET	Retrieve Queue		
PATCH	Account: Updat...		

Authorization

Type No Auth

Security Configuration

Throttling

On enabling, an additional outcome throttleLimitReached* will be available with this custom node.

Rate Limit Limit total number of request per second.
Concurrency Limit

Here's an example of an API call to create a 'Record' in the 'Account' table.

Lt Airtable API for "LTRCCT-2813 - Mastering Demo-Driven Customer Journeys with Webex Customer Experience" ↴

INTRODUCTION

METADATA

RATE LIMITS

AUTHENTICATION

ACCOUNTS TABLE

- Fields
- List records
- Retrieve a record
- Create records**
- Update/Upsert records
- Delete records

ORDERS TABLE

ERRORS

Create accounts records

To create new records, issue a **POST** request to the `@accounts` endpoint. Note that table names and table ids can be used interchangeably. Using table ids means table name changes do not require modifications to your API request.

Your request body should include an array of up to 10 record objects. Each of these objects should have one key whose value is an inner object containing your record's cell values, keyed by either field name or field id.

Returns an array of record objects created if the call succeeded, including record IDs which will uniquely identify the records within [LTRCCT-2813 - Mastering Demo-Driven Customer Journeys with Webex Customer Experience](#).

The Airtable API will perform best-effort automatic data conversion from string values if the `typecast` parameter is passed in ([click to show example](#)). Automatic conversion is disabled by default to ensure data integrity, but it may be helpful for integrating with 3rd party data sources.

You can also include a single record object at the top level. [Click here to hide the example](#).

curl **JavaScript**

```
EXAMPLE REQUEST
curl -X POST https://api.airtable.com/v0/appZlsm4WdS7DZUEF/accounts \
-H "Authorization: Bearer YOUR_SECRET_API_TOKEN" \
-H "Content-Type: application/json" \
--data '{
  "fields": {
    "customerEmail": "pod00_user@webex-lab.com",
    "mainTel": "████████████████████████████████████████",
    "secondaryTel": "████████████████████████████████████████",
    "ccTel": "████████████████████████████████████████",
    "customerName": "Pod00 User"
  }
}'
```

EXAMPLE RESPONSE

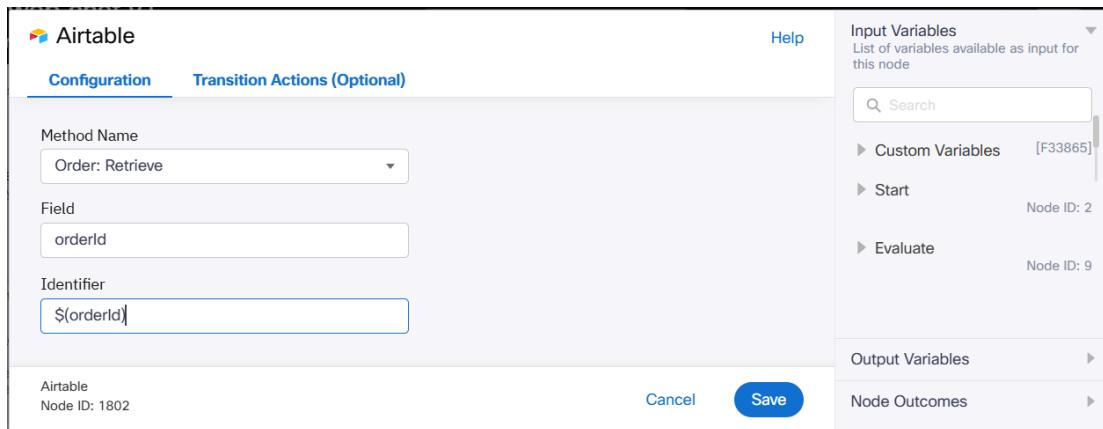
```
{
  "id": "recX2hdV2WwWm0I",
  "createdAt": "2025-01-31T12:59:49.000Z",
  "fields": {
    "customerEmail": "Pod00 User",
    "mainTel": "████████████████████████████████████████",
    "secondaryTel": "████████████████████████████████████████",
    "ccTel": "████████████████████████████████████████",
    "customerName": "Pod00 User"
  }
}
```

An example of what the data looks like in the database.

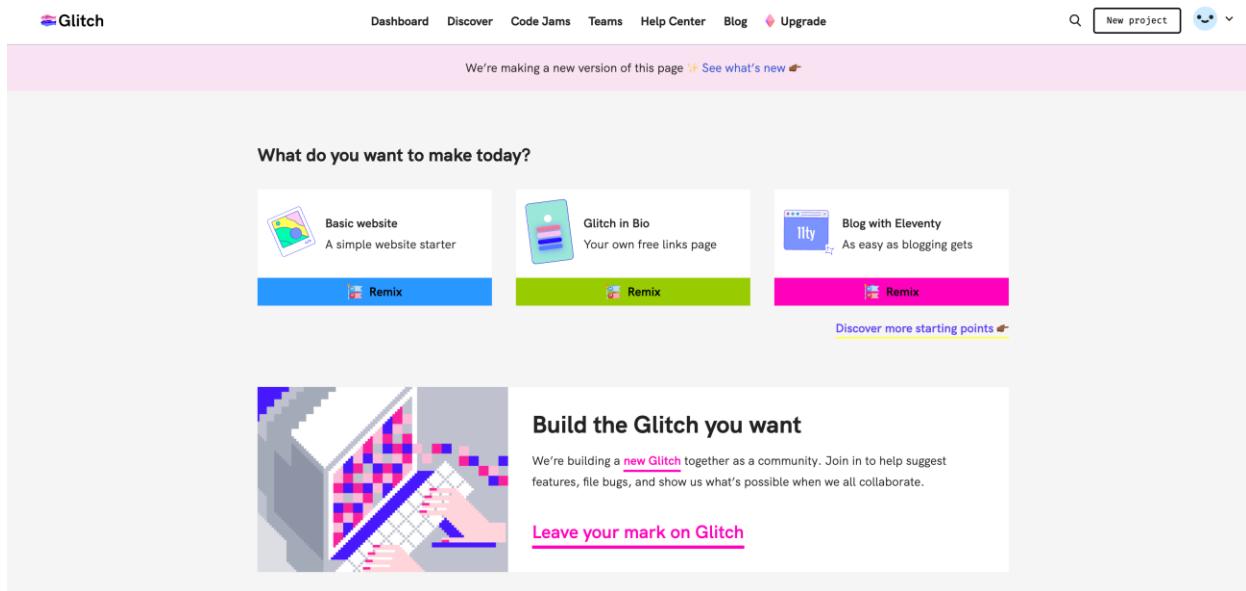
LTRCCT-2813 - Mastering Demo-Driven Customer Jou... ↴

	customerEmail	mainTel	secondaryTel	ccTel	customerName	contacted	socialQueue
1	pod00_user@webex-lab.com	██	██	██	Pod00 User		
2	pod01_user@webex-lab.com	██	██	██	Pod01 User		
3	jake1@cisco.com		491/09912	██	Ian Jake1		05aeef73-3f6b-4fe3-b7e2-cdc00a0ffab4
4	sebh@cisco.com	██	██	██	Seb Huber		05aeef73-3f6b-4fe3-b7e2-cdc00a0ffab4
5	██	██	██	██	Seb Huber		05aeef73-3f6b-4fe3-b7e2-cdc00a0ffab4
6	██	██	██	██	Seb Huber		05aeef73-3f6b-4fe3-b7e2-cdc00a0ffab4

An example of what an Airtable, custom node in Webex Connect looks like. Custom nodes are configured, simplifying the process of API calls you use regularly.



0.8 Glitch



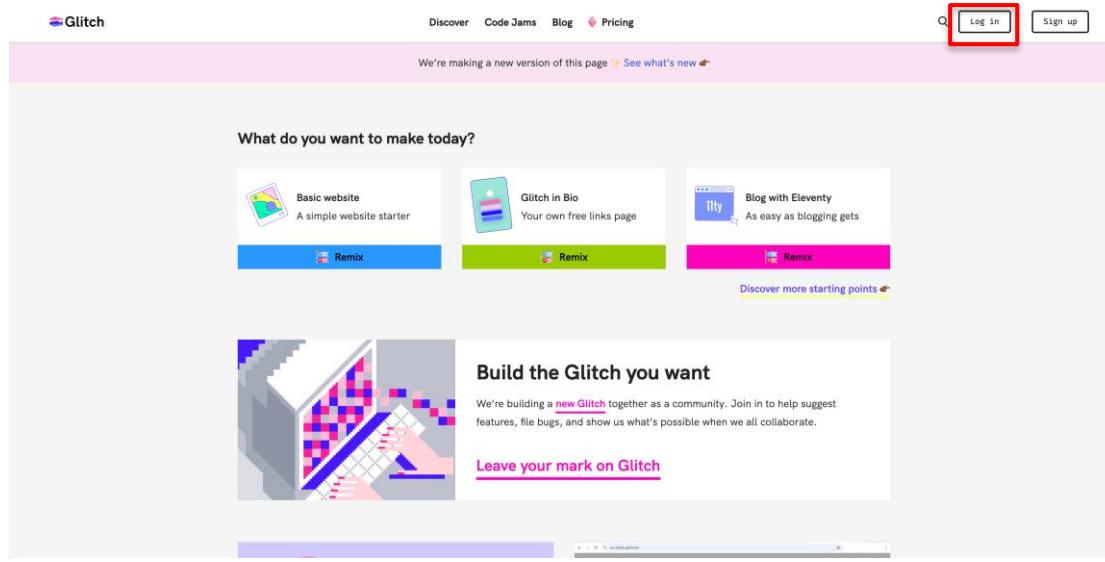
Glitch.com is a platform designed for creating, sharing, and remixing web applications. It offers an intuitive, collaborative environment where developers can build full-stack apps using JavaScript, Node.js, and other web technologies. One of its standout features is the ability to instantly see changes live as you code, making it ideal for experimentation and learning.

Glitch emphasizes community and collaboration, allowing users to "remix" projects, which essentially means creating a copy of someone else's project that you can modify and customize. This fosters a learning environment where users can explore and build upon existing work.

Moreover, Glitch provides hosting for projects, meaning that users can deploy their apps directly from the platform without needing external hosting services. This makes it particularly appealing for both beginners looking to get started with coding and experienced developers wanting to quickly prototype ideas.

0.9 Website creation

Open your browser and go to <https://glitch.com/> and click Log-In on the top right.



Select Password and enter your PodXX_User credentials, solve the CAPTCHA and click on Sign In.

The diagram illustrates the sign-in process. Step 1 shows the 'Sign in to Glitch' page with various social login options and a 'Password' input field highlighted with a red box. Step 2 shows the 'Sign in With Password' page where the 'PodXX_user@webex-lab.com' email and 'Password' field are filled in, and the 'I'm not a robot' checkbox is checked. Red numbers 1 through 4 correspond to these steps.

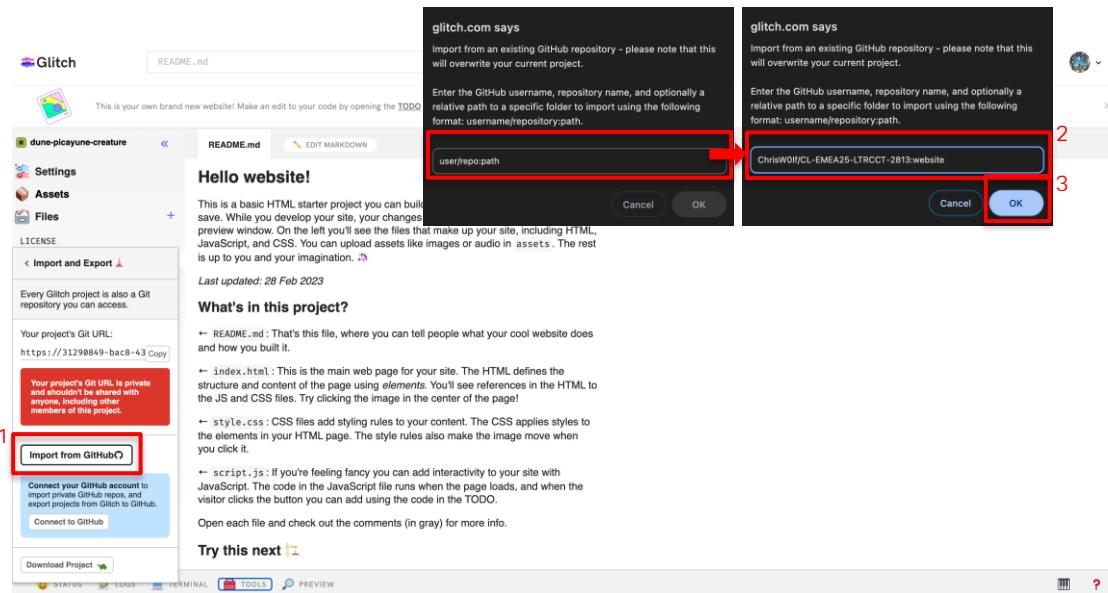
After successfully being logged in, create a new glitch project with clicking on New Project on the top right and select glitch-hello-website.

The screenshot shows the Glitch interface. At the top, there's a navigation bar with links like Dashboard, Discover, Code Jams, Teams, Help Center, Blog, and Upgrade. Below the navigation is a pink banner with the text "We're making a new version of this page! See what's new". The main area is titled "Manage your projects" and shows a summary of active hours (1,000). On the right, there's a sidebar titled "Starter apps" with various project templates listed. One template, "glitch-hello-website", is highlighted with a red box and labeled "2". The URL "https://glitch.com/edit/@remix/glitch-hello-website" is visible at the bottom left of the main content area.

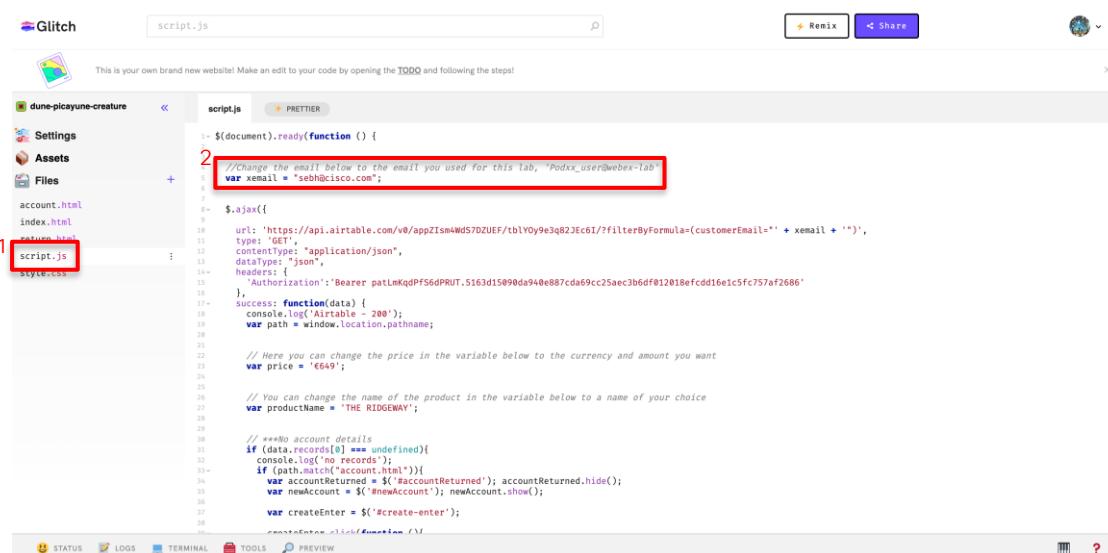
When your new project has been loaded, we need to import Better Bikes Inc's website into our glitch project. Therefore, on the bottom menu bar click on TOOLS and select Import / Export.

This screenshot shows the Glitch project editor for a project named "dune-picayune-creature". The left sidebar lists files like README.md, LICENSE, TODO.md, index.html, script.js, and style.css. The main content area displays the "Hello website!" template. At the bottom, there's a menu bar with buttons for STATUS, LOGS, TERMINAL, TOOLS (which is highlighted with a red box labeled "1"), and PREVIEW. To the right of the menu bar, there's another set of buttons for REMIX, SHARE, and a close icon. The "Assets" section in the sidebar also has an "Import / Export" button, which is highlighted with a red box labeled "2".

In the popup window on the left use Import from Github and enter ChrisW0lf/CL-EMEA25-LTRCCT-2813:website in the user/repo:path field. Finish the import with clicking on OK.



To finish the preparation, we need to update one variable. Open the script.js file and change the value of xemail in line 5 to the email address of your PodXX_User.



Finally, click in the bottom menu on PREVIEW and Preview in a new window.

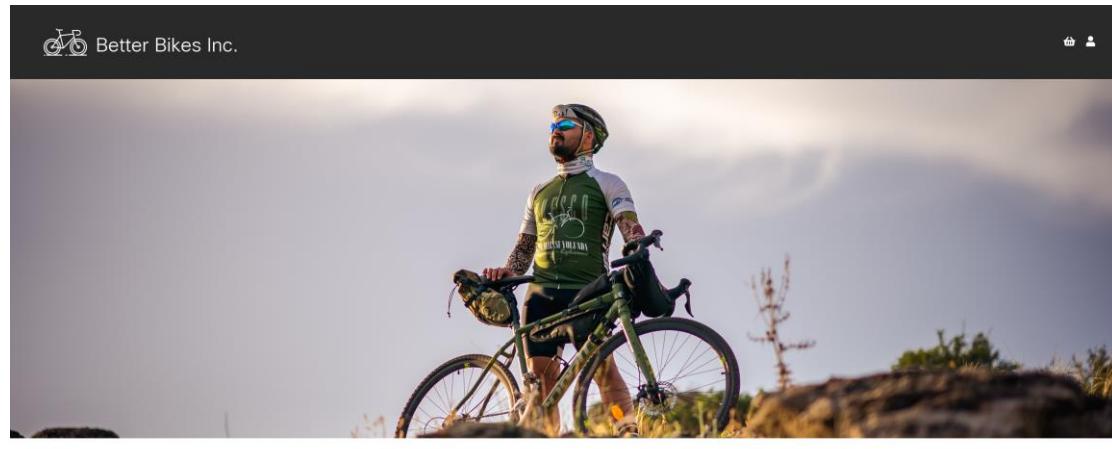
This is your own brand new website! Make an edit to your code by opening the [TODO](#) and following the steps!

```

dune-picayune-creature < script.js > PRETTIER
Settings Assets Files
account.html index.html return.html script.js style.css
This is your own brand new website! Make an edit to your code by opening the TODO and following the steps!
$(document).ready(function () {
    //Change the email below to the email you used for this lab, 'Podxx_user@webex-lab.com'
    var xemail = "chrismwol@webex-lab.com";
    $.ajax({
        url: 'https://api.airtable.com/v0/appZIsm4Wds7DZUEF/tbLYOy9e3q82JEc6I/?filterByFormula=(customerEmail=' + xemail + ')',
        type: 'GET',
        contentType: "application/json",
        dataType: "json",
        headers: {
            'Authorization': 'Bearer patLMKqdPfS6dPRUT.S163d15090da940e887cda69cc25aec3b6df012018efcd16e1c5fc757af2686'
        },
        success: function(data) {
            console.log('Airtable - 200');
            var path = window.location.pathname;
            // Here you can change the price in the variable below to the currency and amount you want
            var price = '€649';
            // You can change the name of the product in the variable below to a name of your choice
            var productName = 'THE RIDGEWAY';
            // See if account details
            if (data.records[0] == undefined){
                console.log('no records');
                if (path.match("account.html")){
                    var accountReturned = $(`#${'accountReturned'}`);
                    accountReturned.hide();
                    var newAccount = $(`#${'newAccount'}`);
                    newAccount.show();
                }
            }
            Open preview page
            Create-enter
        }
    });
}

```

2 1



Confirm your order



Congratulations!

You have created your website and finished Module 0 – Preparations.
Please keep the website open as we need it later in Module 1.

Module 1 – Voice Flow Creation

The configuration of this module is done through Webex Control Hub. Open your web browsers and type <https://admin.webex.com> in the address bar. To log in to Control Hub, please use your PodXX_User account and password.

Click on Contact Center in Services on the left side to access the contact center configuration menu.

MANAGEMENT

- Users
- Groups
- Locations
- Workspaces
- Devices
- Apps
- Account
- Security
- Organization Settings

SERVICES

- Updates & Migrations
- Messaging
- Calling
- Vidcast
- Contact Center >
- Connected UC
- Hybrid

Overview

Welcome to Webex! Let's get started.

Here's your starting guide, which is customized for your organization.



Start using Webex

Deploy Messaging and Meeting via foundational setup and user management tasks.

[View guide](#)



1.1 Wrap Up Codes

Idle or wrap up codes are two types of Auxiliary codes. Agents select Idle or Wrap-Up codes in Webex Contact Center Agent Desktop to indicate their unavailability or status of the customer contacts. Idle codes typically indicate why an agent is not available to take customer contacts, such as during a lunch break or meeting. Wrap-up codes indicate the result of the customer contacts, for example, the agent escalated the contact or sold a service.

To create new codes, please go to Idle/Wrap-Up Codes on the left side in the sidebar.

Idle/Wrap-up Codes					
Search by name		Active	8 Idle/Wrap-up Codes		
idle/Wrap-up code	Description	Idle/Wrap-up code type	Default	Status	Last modified
away	Idle Code	No	● Active	December 02, 20...	⋮
DND	Idle Code	No	● Active	December 02, 20...	⋮
in a Calendar Me...	Idle Code	No	● Active	December 02, 20...	⋮
in a external Call	Idle Code	No	● Active	December 02, 20...	⋮
in an Meeting	Idle Code	No	● Active	December 02, 20...	⋮
in presentation	Idle Code	No	● Active	December 02, 20...	⋮
Meeting	Default idle code	Idle Code	Yes	● Active	October 30, 2024...
Wrap Up	Default wrap-up ...	Wrap-up Code	Yes	● Active	December 04, 20...

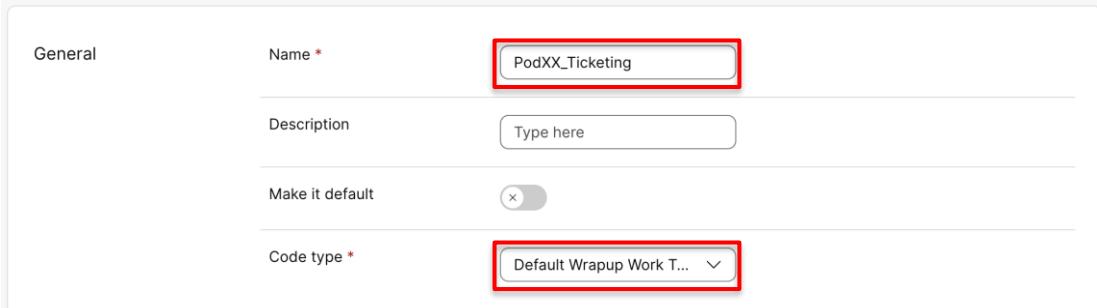
Use the Create Idle/Wrap-Up Code button to create the following new codes:

Name: PodXX_Pause
Code Type: Default Idle Work Type

Click Create.

Go back to Idle-Code overview and create another code.

Name: PodXX_Ticketing
 Code Type: Default Wrapup Work Type



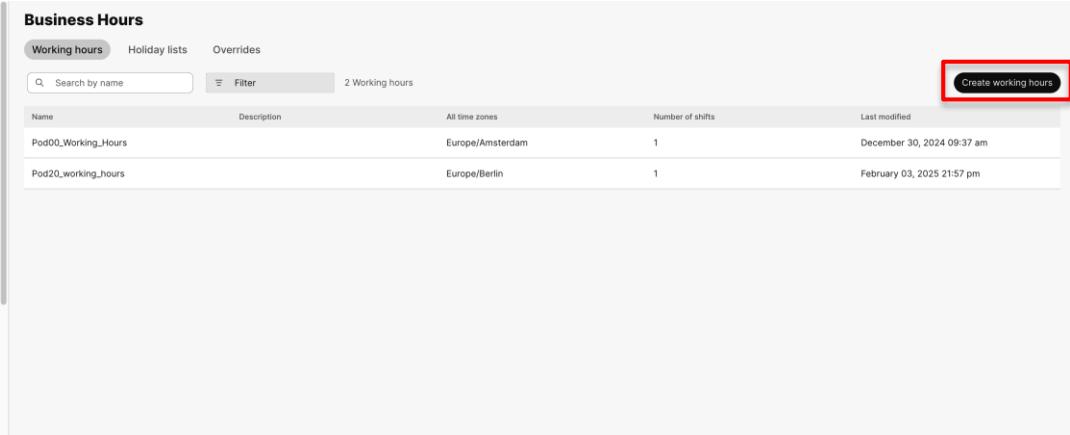
The screenshot shows a 'Create New' dialog box. It has two main sections: 'General' and 'Advanced'. In the 'General' section, there are three fields: 'Name *' which contains 'PodXX_Ticketing' and is highlighted with a red box; 'Description' which contains 'Type here'; and 'Make it default' which has a toggle switch. Below these is a 'Code type *' dropdown menu which is currently set to 'Default Wrapup Work T...' and is also highlighted with a red box.

Click Create.

1.2 Business Hours

The Working Hours and Holidays created here can be used directly in the Call Flow Designer. Please configure the working hours for your Pod.

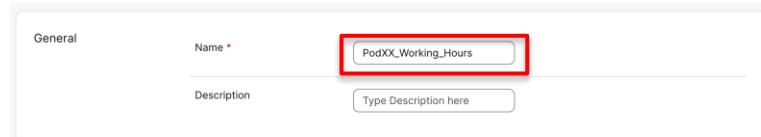
Navigate to Business Hours in the sidebar on the left and click Create Working Hours



Name	Description	All time zones	Number of shifts	Last modified
Pod00_Working_Hours		Europe/Amsterdam	1	December 30, 2024 09:37 am
Pod20_working_hours		Europe/Berlin	1	February 03, 2025 21:57 pm

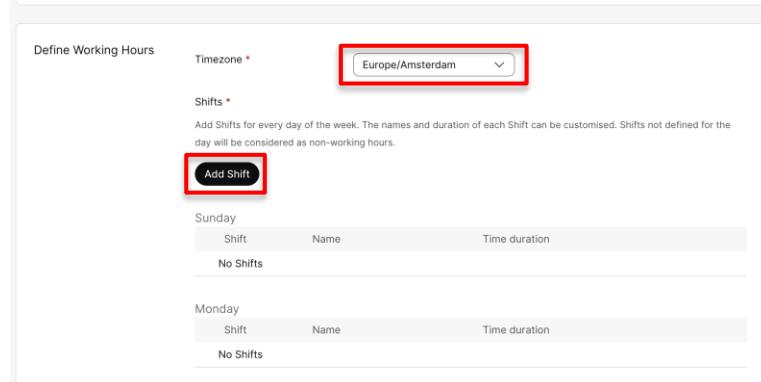
Name: PodXX_Working_Hours
 Timezone: Europe/Amsterdam

Click Add Shift



General

Name *	<input type="text" value="PodXX_Working_Hours"/>
Description	<input type="text" value="Type Description here"/>



Define Working Hours

Timezone *	<input type="text" value="Europe/Amsterdam"/>
------------	---

Shifts *

Add Shifts for every day of the week. The names and duration of each Shift can be customised. Shifts not defined for the day will be considered as non-working hours.

Add Shift

Sunday	Shift	Name	Time duration
No Shifts			
Monday	Shift	Name	Time duration
No Shifts			

Add Shift

Add Shift

Name * PodXX_Shift

Select all that apply:

- Sunday
- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday

The start time selected will begin from the first millisecond. The end time selected will be until the last millisecond.

Time duration to

Save Save

Click Save and Create.

1.3 Multimedia Profile

If your administrator enables multimedia for your enterprise, you can assign each agent with a multimedia profile. Each profile specifies the number of each type of media, such as email, chat, or telephony, that the agent can handle simultaneously.

Navigate to Multimedia Profile in the sidebar on the left.

Multimedia profile	Description	Voice	Chat	Email	Social	Status	Last modified
Default_Mult...	Default Mult...	1	5	5	5	Active	October 30,...
Default_Tele...	Default tele...	1	0	0	0	Active	October 30,...

Click Create Multimedia Profile.

Name: PodXX_Multimedia Profile
 Channels: as of the screenshot below

Click Create.

1.4 Desktop Layout

The Desktop Layout feature allows you to configure the Webex Contact Center Desktop as per your business requirements. You can customize elements such as logo, title, and widgets. You can create a desktop layout and assign it to a team. This layout generates the agent experience on the desktop for all agents who sign in as part of that team.

Navigate to Desktop Layout in the sidebar on the left.

First, we need the Template for our Lab. Open the Desktop Layout Template and download the template.

The screenshot shows two pages of the Cisco Webex Contact Center interface:

- Desktop Layouts Page:** A list of desktop layouts. The 'Template' entry is highlighted with a red box. The table columns are: Desktop layout, Description, Status, and Last modified. The 'Template' entry has a description of 'Template', status 'Active', and was last modified on December 30, 2024 at 11:14 AM.
- Template Details Page:** A detailed view of the 'Template' desktop layout. It includes fields for Name (set to 'Template'), Description (empty), and Teams (empty). It also includes a JSON file section where a file named 'Default Desktop Layout LTRCCT-2813.json' is uploaded. The file status is 'File is ready for import'. There is a 'Replace file' button.

Go back to Desktop Layout List.

Click Create Desktop Layout.

Name: PodXX/Desktop Layout

JSON File: replace the template with the file you have downloaded

Layout Details

Name *

Description

Teams

0 Teams

JSON File *

Please use the default desktop layout to customize your desktop.

[Download default desktop layout](#)

File is ready for import

Default Desktop Layout.json uploaded

This is an unmodified Desktop Layout. Therefore, new layout features appear on the Desktop upon reload or when the user signs into the Desktop next time

Click Create.

1.5 Teams

A team is a group of people who support a specific group of functions. For example, supporting the “Gold” customers or managing billing, and so on. A team consists of agents and is associated with a specific site.

Navigate to Teams in the sidebar on the left and click Create Team.

Team	Description	Site	Type	DN	Multimedia prof	Skill profile	Desktop layout	Ranked queue	Status	Last modified	
Team-1		Site-1	CAPAC...	0			Global ...	No ⓘ	● Active	October...	ⓘ

Name: PodXX_Team
Parent Site: LTRCCT-2813
Team Type: Agent based
Skill Profile: None
Multimedia Profile: PodXX_Multimedia Profile
Desktop Layout: PodXX/Desktop Layout

General	
Name *	<input type="text" value="Pod00_Team"/>
Description	<input type="text" value="Type here"/>
Parent Site *	<input type="text" value="LTRCCT-2813"/>
Team settings	
Team type *	<input type="radio"/> Capacity Based <input checked="" type="radio"/> Agent Based
Skill profile	<input type="text" value="None"/>
Multimedia profile	<input type="text" value="Pod00_Multimedia Profile"/>
Desktop layout	<input type="text" value="Pod00/Desktop Layout"/>
Agents	<input type="text" value="Type here"/>
0 Agents Clear All	

Click Create and Done.

1.6 Queues

An inbound queue is where the customer contact waits before the system assigns the customer to an agent or DN.

The inbound queue that represents a third-party Automatic Call Distributor (ACD) is known as a proxy queue. Calls that are sent to an external ACD are distributed to agents by the external ACD. Webex Contact Center uses proxy queues to track the call activity that occurs on the external ACD queues.

Navigate to Queues in the sidebar on the left and click Create Queue.

Queue	Description	Queue type	Channel type	Status	Last modified
Outdial Queue-1	Outdial Queue cre...	Outdial Queue	Telephony	● Active	October 30, 2024...
Queue-1	Queue created by...	Inbound Queue	Telephony	● Active	October 30, 2024...

Name: PodXX_Incoming Queue
 Queue Type: Inbound Queue
 Channel Type: Telephony

General

Name *	<input type="text" value="PodXX_Incoming Queue"/>
Description	<input type="text" value="Type here"/>
Queue Type *	<input type="text" value="Inbound Queue"/>
Channel Type *	<input type="text" value="Telephony"/>

You can't change Queue Type once the queue is created

You can't change Channel Type once the queue is created

Click Create Group.

Contact Routing Settings

Outbound Campaign ①

Outbound Campaign is only applicable for Outdial Queues.

Queue Routing Type *

▼

You can't change Queue Routing Type once the queue is created

Call Distribution *

You can add one or more teams to this Call through a Call Distribution Group. You can add multiple groups to distribute a call to more teams. Call distribution is independent of other queues login defined in Flows.

Select your Team and Save.

Group 1

Priority

▼

Switch to this group after ②

Seconds

Search by name

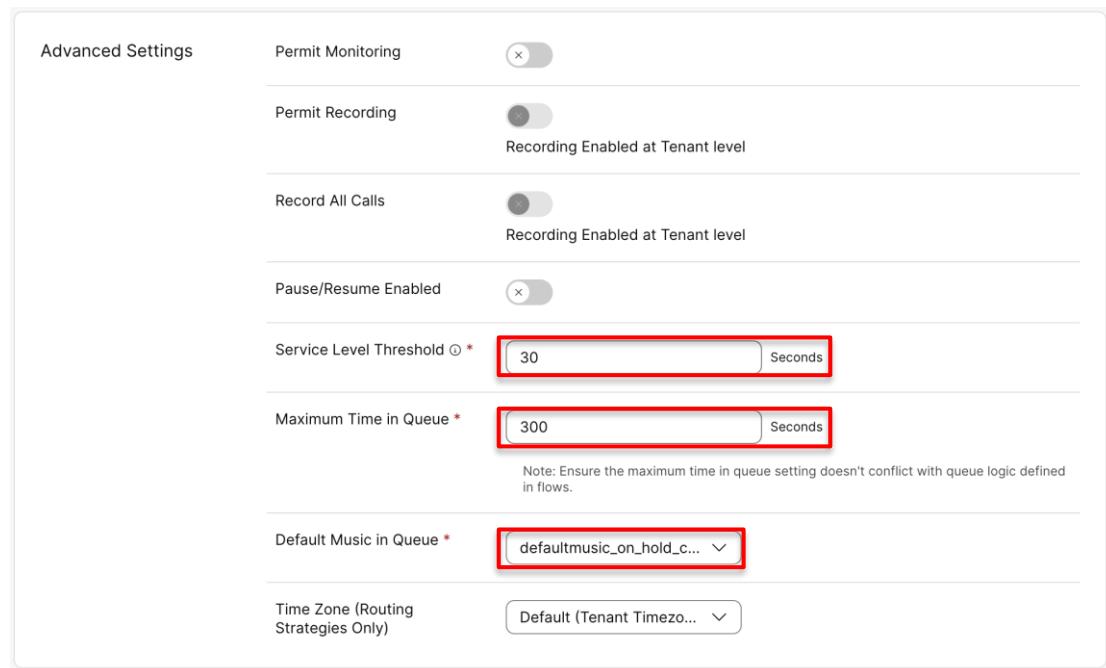
2 Records

Team	Site	Team type
<input type="checkbox"/> PodXX_Team	Site-1	AGENT
<input type="checkbox"/> Team-1	Site-1	CAPACITY

Under Advanced Settings configure the following:

Service Level Threshold: 30 seconds
 Maximum Time in Queue: 300 seconds
 Default Music in Queue: first entry

Then click Create.



Advanced Settings	Permit Monitoring	<input checked="" type="checkbox"/>
Permit Recording	<input checked="" type="checkbox"/>	Recording Enabled at Tenant level
Record All Calls	<input checked="" type="checkbox"/>	Recording Enabled at Tenant level
Pause/Resume Enabled	<input checked="" type="checkbox"/>	
Service Level Threshold ⓘ *	<input type="text" value="30"/> Seconds	
Maximum Time in Queue *	<input type="text" value="300"/> Seconds	<small>Note: Ensure the maximum time in queue setting doesn't conflict with queue logic defined in flows.</small>
Default Music in Queue *	<input type="text" value="defaultmusic_on_hold_c..."/>	
Time Zone (Routing Strategies Only)	Default (Tenant Timezo... ▾)	

Click Create.

1.7 Flows

Navigate to Flows in the sidebar on the left.

The screenshot shows the Cisco Contact Center software interface. On the left, there is a sidebar with the following sections:

- Main Menu
- Contact Center**
 - Overview
 - CUSTOMER EXPERIENCE
 - Channels
 - Queues
 - Business Hours
 - Audio Prompts
 - Flows** (highlighted with a red box)
 - Surveys

The main area is titled "Flows" and contains the following details:

- Search bar: Search by name
- Filter: All
- Status: 1 Flows
- Manage Flows button
- Table header: Flow, Description, Status, Last modified
- Table data: Flow_PodXX, To learn how to configure, s..., Draft, December 05, 2024 14:26 PM

To get a flow template export the Flow Flow_PodXX through the 3 dots menu.

The screenshot shows the same interface as above, but with a context menu open over the row for "Flow_PodXX". The menu options are:

- Copy
- Export
- Delete

Next, you need to rename the flow name. For that, please open the exported file in an Editor and change the Flow Name to Flow_PodXX.

```
{
  "orgId": "2414081e-175b-478b-b759-4efcf7b683b",
  "version": 2,
  "id": "674d70885b521b7b2700d227",
  "flowType": "FLOW",
  "name": "Flow_Pod_01",
  "description": "To learn how to configure, see supporting documentation here - https://help.webex.com/en-us/article/n5595zd/Webex-Contact-Center-Setup-and-Administration-Guide#Cisco_Generic_Topic.dita_e338e055-64b0-4973-bd52-8a5581dc0e",
  "comment": "",
  "variables": [],
  "process": {
    "activities": [
      "baf09352-fa55-42f9-h121-8b02cf634c7e": {
        "id": "baf09352-fa55-42f9-h121-8b02cf634c7e",
        "name": "WelcomePrompt",
        "group": "action",
        "properties": {
          "text": "Hello, welcome to our contact center!",
          "language": "en-US"
        }
      }
    ]
  }
}
```

Save the file.

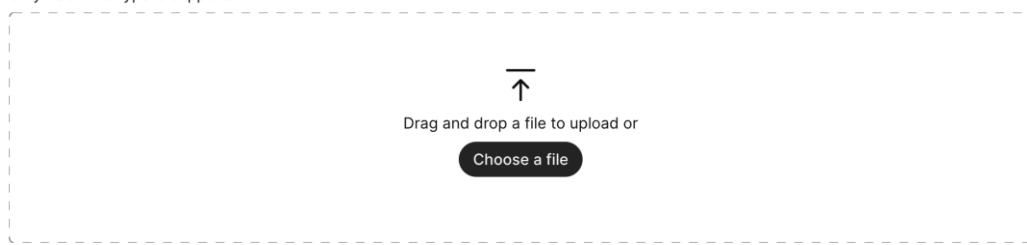
Now the new file must be uploaded. For that, import the new Flow File through Manage Flows and Import Flows.

The screenshot shows the Cisco Flows interface. At the top, there's a search bar labeled 'Search by name' and a filter button 'All'. Below that is a table with one row, '1 Flows'. The table columns are 'Flow', 'Description', 'Status', and 'Last modified'. The single entry is 'Flow_PodXX' with a description 'To learn how to configure, s...', status 'Draft', and last modified 'December 05, 2024 14:26 PM'. To the right of the table is a 'Manage Flows' dropdown menu with options: 'Import Flows' (which is highlighted with a red box), 'Create Flows', and 'Delete'.

Choose your File and click Import.

Upload The File

Only JSON file type is supported.



Now your Flow should be listed. If not, please contact us.

The screenshot shows the Cisco Flows interface. The 'Manage Flows' dropdown menu is open, and the 'Import Flows' option is highlighted with a red box. The rest of the interface is identical to the previous screenshot, showing a table with two flows: 'Flow_PodXX' and 'Flow_Pod_01'.

Open the new flow you've just added with the highlighted Button

This screenshot is very similar to the previous ones, showing the Cisco Flows interface with the 'Manage Flows' dropdown open and the 'Import Flows' option highlighted. The main table below shows two flows: 'Flow_PodXX' and 'Flow_Pod_01'.

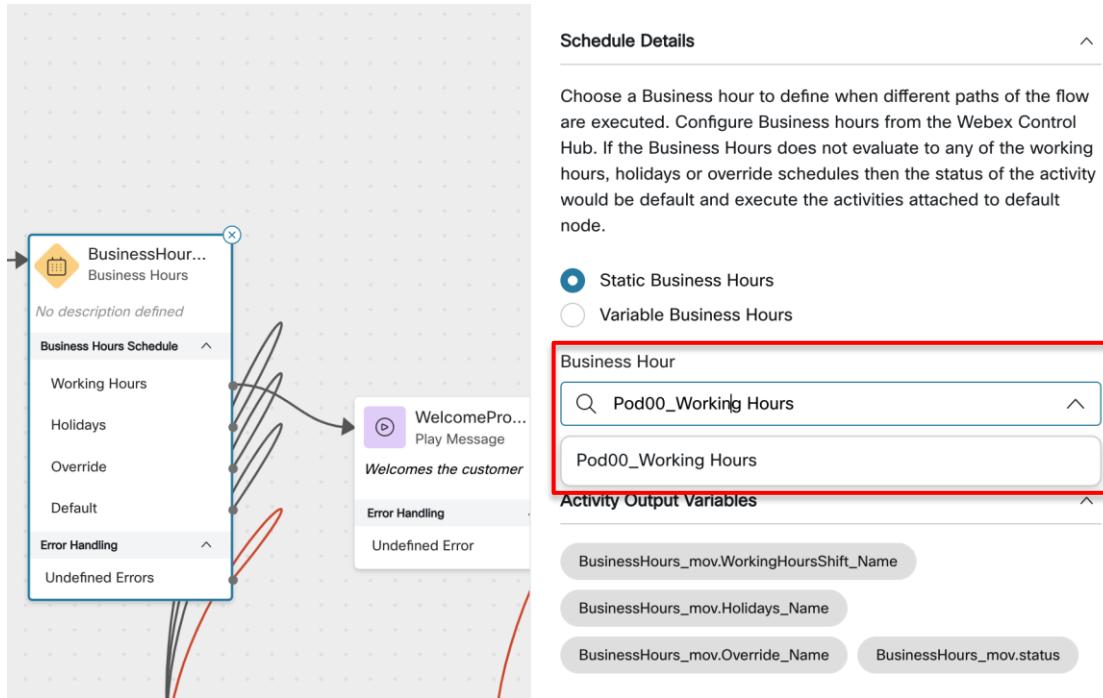
Toggle the Edit mode on.

Cisco Webex
Flow Designer

Flow_Pod00 ▾ Draft



Click on Business Hours and choose your Pod Working Hours.



Click on the grey background and zoom out.

Click on Queue and select your Pod Queue.

The screenshot shows a flow diagram and its configuration settings. The flow starts with a 'Queue' activity labeled 'Queue Contact' which 'Queues the call to agents'. This is followed by an 'Error Handling' section and a 'Failure' section. The flow ends at an 'EndFlow_p88' activity. A red arrow points from the 'Queue' activity to the configuration window. The configuration window is titled 'Contact Handling' and contains the following sections:

- Contact Handling**: A note about queue selection based on flow variables.
- Static Queue** (selected): A radio button option.
- Variable Queues**: An unselected radio button option.
- Queue**: A dropdown menu set to 'Pod00_Incoming Queue', highlighted with a red border.
- Set Contact Priority**: A toggle switch that is off. A note explains it prioritizes contacts over others in the queue.
- Check Agent Availability**: A toggle switch that is off. A note explains it excludes teams with no available agents.

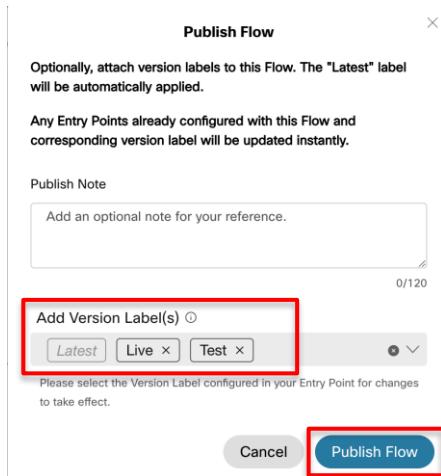
Enable the Validation – when it's green click on Publish Flow.

The validation details dialog box shows the following information:

- Validation Details**
- Flow Errors:0** (green button)
- Flow has no errors and can be published**
- Recommendations: 1** (orange button)
- 1. Add descriptions for activities**
- Dismiss Recommendations**

At the bottom of the dialog, there are three buttons: 'Flow Errors: 0' (green), 'Autosave' (blue), 'Validation: On' (blue), and a large blue 'Publish Flow' button, which is highlighted with a red border.

Open the Version Label Drop Down, select Live and Test and then Publish the Flow.



After that you can close the browser tab.

1.8 Channels (Entry Point)

The inbound entry point is the initial landing place for a customer contact in the Webex Contact Center system. For the customer calls, you can associate one or more toll-free or dial numbers with a given entry point. The system performs IVR call treatment for a call while the call is in the entry point.

You can assign a flow to an entry point. Moreover, if you want to implement the business hours to define the working and nonworking hours for your contact center, you can choose a flow that has the business hours activity configured and assign it to the entry point.

Navigate to Channels in the sidebar on the left and click Create Channel.

The screenshot shows the webex Control Hub interface. On the left, there's a sidebar with 'Main Menu', 'Contact Center' (Overview, Channels, Queues, Business Hours), and 'CUSTOMER EXPERIENCE' (Channels). The 'Channels' link is highlighted with a red box. The main area is titled 'Channels' with a search bar and a filter for 'Active'. It shows three channels: 'Entry Point-1', 'Outdial Entry Poin...', and 'Outdial Transfer t...'. A 'Create Channel' button is at the top right, also highlighted with a red box.

In the first step we are creating a channel for inbound calls and associate this to our flow and our service queue Number.

Name: PodXX_Incoming Call Queue
 Channel Type: Inbound Telephony

The form has fields for 'Entry Point' (Name: * PodXX_Incoming Call Queue, Description: Type here), 'Channel Type' (Inbound Telephony selected), and other optional fields like 'Service Level Threshold', 'Timezone', 'Routing Flow', 'Version Label', and 'Music on Hold'.

Configure the following Entry Point Settings:

Service Level Threshold: 30 Seconds
 Timezone: Europe/Amsterdam
 Routing Flow: Your pods Flow
 Version Label: Live
 Music on Hold: Choose the first entry

Entry Point Settings	Service Level Threshold ⓘ *	<input type="text" value="30"/> Seconds
	Timezone (Business Hours only) *	<input type="text" value="Europe/Amsterdam"/>
	Routing Flow	<input type="text" value="Flow_Pod00"/>
	Version Label	<input type="text" value="Live"/>
	Music on Hold	<input type="text" value="defaultmusic_on_hold.w..."/>

For your reference:

You have seen that you can choose different Version Labels. This Feature is the Version Management on Flows. So, with that you can create a dedicated Test Channel for this Flow. This is so you can test all new configurations, and then when the tests are successfully completed, you can publish the flow, making it Live.

Now add a PSTN Number to this Entry Point. Under Support Number click Add.

Choose LTRCCT-2813 as Webex Calling Location.

Support Number

Your customers will use these numbers to reach you up in Webex Calling services.

Number	Webex Calling location
1	<input type="button" value="Select"/> <input type="button" value="Search"/> Cisco Live EMEA... LTRCCT-2813
<input type="button" value="Add"/>	

For Support Number choose the correlating Number that is assigned to your Pod!

Pod Number	PodXX Service Queue
01	+442046200496
02	+442046200506
03	+442046200517
04	+442046200524
05	+442046200527
06	+442046200530
07	+442046200533
08	+442046200538
09	+442046200629
10	+442046200941
11	+442046200947
12	+442046200951
13	+442046200954
14	+442046200956
15	+442046200974
16	+442046200977
17	+442046200980
18	+442046200982
19	+442046200989
20	+442046200994

Leave the PSTN Region as Default and Accept under Actions.

Support Number

Your customers will use these numbers to reach your Contact Center. You can pick more than one number from the list of numbers available and already set up in Webex Calling services.

Number	Webex Calling location	Support Number	PSTN Region	Actions
1	LTRCCT-2813	+442046200941	Default	<input checked="" type="checkbox"/> <input type="button" value="X"/>

Add

Afterwards, click on Create.

1.9 Desktop Profile

Navigate to Desktop Profiles in the sidebar on the left and click Create Desktop Profile.

Desktop profile	Description	Parent type	Parent name	Status	Last modified
Agent-Profile	Agent profile	Tenant	LTRCCT-2813	● Active	October 30, 2024 1...
Agent-Profile (Auto ...	Agent profile Autowr...	Tenant	LTRCCT-2813	● Active	October 30, 2024 1...
PodXX/Desktop Prof...		Tenant	LTRCCT-2813	● Active	December 30, 2024 ...

Define the Name & Description as PodXX/Desktop Profile and click Next.

Configure the following Idle/Wrap-up Code settings:

Wrap-up codes:	Auto wrap-up with timeout of
Time in seconds:	15 seconds
Wrap-up codes:	Specific
Select wrap-up codes:	Wrap Up (Default), PodXX_Ticketing
Idle Codes:	Specific
Selected Idle codes:	PodXX_Pause, Meeting (Default)

Idle/Wrap-up Code

Wrap-up codes

- Manual wrap-up
- Auto wrap-up with timeout of

Time in seconds (*) seconds

Auto wrap-up extension

Agent available after outdial

Wrap-up codes

- All
- Specific

Select wrap-up codes *

2 Wrap-up codes

Idle codes

- All
- Specific

Select idle codes *

2 Idle codes

Click on Next.

Configure the following Collaboration settings:

Entry Point/Queue transfer targets: Specific

Select Entry Points/Queues: PodXX_Incoming Call Queue

Buddy Teams: Specific

Select Teams: PodXX_Team

General Create Idle/Wrap-up ... **Collaboration** Dial Plans Voice Channel options Agent Statistics Desktop Timeout

Collaboration

Entry Point/Queue transfer targets

All
 None
 Specific

Select Entry Points/Queues *

Pod00_Incoming Call Queue

Type here

1 Entry Points/Queues

Consult to queue

Buddy Teams ⓘ

All
 None
 Specific

Select Teams *

Pod00_Team

Type here

1 Teams

Webex App

Configure Webex features for your tenant.

Display user details ⓘ

State synchronization ⓘ For mappings refer Tenant Settings > [Desktop](#).

Click on Next.

Leave the Dial plans and Voice channel options configuration as it is by clicking Next.

The screenshot shows two configuration panels. The top panel is titled 'Dial Plans' and contains three sections: 'Outdial' (disabled), 'Address Book' (set to 'Select'), and 'Dial Plan' (disabled). The bottom panel is titled 'Voice Channel options' and includes a note that 'One Voice option must always be checked.' It lists three checked options: 'Agent DN', 'Extension', and 'Desktop'. Below this, under 'Validation for Agent DN', there are three radio button options: 'Unrestricted (Allow any value)' (selected), 'Provisioned DN (Restrict login DN to provisioned agent DN)', and 'Validate using Dial Plans (Select from list)'.

Configure the following Agent statistics settings:

Agent statistics:	Enabled
Queue statistics:	Specific
Selected Queues:	PodXX_Incoming Queue
Logged-in team statistics:	Enabled
Team statistics:	Specific
Selected Teams:	PodXX_Team

The screenshot shows a series of tabs at the top: General, Create Idle/Wrap-up ..., Collaboration, Dial Plans, Voice Channel options, Agent Statistics, and Desktop Timeout. The Agent Statistics tab is active. Below the tabs, there are two main sections: Agent Statistics and Desktop Timeout.

Agent Statistics:

- Agent statistics:** A toggle switch is turned on (blue with a checkmark) and highlighted with a red box.
- Queue statistics:**
 - Select Queues ***: A dropdown menu shows "Pod00_Incoming Queue" selected, highlighted with a red box.
 - Logged-in team statistics:** A toggle switch is turned on (blue with a checkmark) and highlighted with a red box.

Desktop Timeout:

- Desktop inactivity timeout:**
 - Default value** (radio button selected, highlighted with a red box)
 - Custom value** (radio button unselected)

Click Next.

Leave the Desktop timeout on Default value.

The screenshot shows the Desktop Timeout section of the configuration interface. It includes a "Desktop Timeout" label and a "Desktop inactivity timeout" section. In the "Desktop inactivity timeout" section, the "Default value" radio button is selected and highlighted with a red box.

Click Create.

1.10 User Profile

User profiles determine the features accessible to a user in Cisco Contact Center.

Navigate to User Profiles in the sidebar on the left.

User profile	Description	Profile type	Status	Last modified
Administrator Only Profile	Profile for Administrator ...	ADMINISTRATOR_ONLY	● Active	October 30, 2024 16:32 ...
Administrator Profile	Administrator profile	ADMINISTRATOR	● Active	October 30, 2024 16:32 ...
Pod XX Premium Agent U...	Premium agent user profile	PREMIUM_AGENT	● Active	December 30, 2024 09:1...
Premium Agent User Pro...	Premium agent user profile	PREMIUM_AGENT	● Active	October 30, 2024 16:32 ...
Standard Agent User Pro...	Standard agent user prof...	STANDARD_AGENT	● Active	October 30, 2024 16:32 ...
Supervisor Profile	Supervisor Profile	SUPERVISOR	● Active	October 30, 2024 16:32 ...

Now click on Copy from Premium Agent User Profile

User profile	Description	Profile type	Status	Last modified
Administrator Only Profile	Profile for Administrator withoutou...	ADMINISTRATOR_ONLY	● Active	October 30, 2024 16:32 PM
Administrator Profile	Administrator profile	ADMINISTRATOR	● Active	October 30, 2024 16:32 PM
Premium Agent User Profile	Premium agent user profile	PREMIUM_AGENT	● Active	October 30, 2024 16:32 PM
Standard Agent User Profile	Standard agent user profile	STANDARD_AGENT	● Active	October 30, 2024 16:32 PM
Supervisor Profile	Supervisor Profile	SUPERVISOR	● Active	October 30, 2024 16:32 PM

Rename the Name to your Pod IDs number and remove the _copy in the end.

General	Name *	Pod XX Premium Agent User Profile
	Description	Premium agent user profile
	Profile type *	Premium Agent

You can't change Profile type once the User Profile is created.

Click on Next.

Leave the Desktop experience as it is by clicking on Next.

Desktop Experience	Agent Desktop	Edit ▼		
<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Multimedia</td> <td style="width: 30%;">Edit ▼</td> </tr> </table>			Multimedia	Edit ▼
Multimedia	Edit ▼			
Select the features you want to provide to your users. Click here to learn more.				
<input checked="" type="checkbox"/> Basic Digital <input checked="" type="checkbox"/> Social Channel				

Configure the following Access Rights settings:

Entry points: PodXX_Incoming Call Queue
 Queues: PodXX_Incoming Queue
 Sites: LTRCCT-2813
 Teams: PodXX_Team

Access Rights	
Entry Points *	<div style="border: 1px solid #ccc; padding: 2px; width: 100%; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; padding: 2px; width: 100%; height: 20px; background-color: #f0f0f0;"></div>
1 Entry Points	<input type="button" value="Clear All"/>
Queues *	<div style="border: 1px solid #ccc; padding: 2px; width: 100%; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; padding: 2px; width: 100%; height: 20px; background-color: #f0f0f0;"></div>
1 Queues	<input type="button" value="Clear All"/>
Sites *	<div style="border: 1px solid #ccc; padding: 2px; width: 100%; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; padding: 2px; width: 100%; height: 20px; background-color: #f0f0f0;"></div>
1 Sites	<input type="button" value="Clear All"/>
Teams *	<div style="border: 1px solid #ccc; padding: 2px; width: 100%; height: 20px; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; padding: 2px; width: 100%; height: 20px; background-color: #f0f0f0;"></div>
1 Teams	<input type="button" value="Clear All"/>

Click on Create.

Note: All Supervisor Agents are using the same Supervisor Profile.

1.11 Agent Configuration

Every user who has been assigned a license on Control Hub is automatically synced to the contact center. However, this user still needs to be enabled and configured. In preparation, 3 users were created. For license assignment, please refer to the Preparations section (Module 0.1).

Navigate to Contact Center Users in the sidebar on the left and search for your Pod.

Select the entry with your Pod ID in First name and Agent in the Last name.

Contact Center Users												
<input type="text" value="pod00"/>		Active		3 Contact Center Users								
Manage settings for existing users below. To create or delete users, go to the Control Hub Users												
First name	Last name	Email	User profile	Contact center en	Supervisor team	Site	Team	Desktop profile	Multimedia profile	Skill profile	Last modified	Status
Pod00	Agent	pod00_ag...	Premium ...	No	-	-	-	-	-	-	December... ● Active	
Pod00	User	pod00_us...	Administr...	No	-	-	-	-	-	-	December... ● Active	
Pod00	Super	pod00_su...	Superviso...	No	-	-	-	-	-	-	December... ● Active	

Enable the user for Contact Center.

Pod00 Agent

ID: 5a9804b6-f394-4fcf-9990-7d43183d595a • Last Modified: February 04, 2025 10:23 AM

General	First name	Pod00
	Last name	Agent
	Email	pod00_agent@webex-lab.com
	User Profile *	Premium Agent User Profile ▾
	Contact Center *	<input checked="" type="checkbox"/> X
	Referenced by	You can access following link to see which other entities are referenced.
		Reference list

Configure the following Agent Settings:

Site: LTRCCT-2813
 Teams: PodXX_Team
 Desktop-Profile: PodXX/Desktop Profile
 Multimedia profile: PodXX/Multimedia Profile
 Skill Profile: None

Agent Settings	Site *	LTRCCT-2813
	Teams	(Pod00_Team x)
		1 Teams Clear All
	Desktop Profile *	Pod00/Desktop Profile
	Multimedia Profile	Pod00/Multimedia Profile
	Skill Profile	None
	Default DN	
	External ID	

Click Save and go back to user list by clicking Contact Center Users on the top left.

Search again for your Pod and select the entry with your Pod ID in First name and Super in the Last name and enable this user for Contact Center as well.

General	First name	Pod00
	Last name	Super
	Email	pod00_super@webex-lab.com
	User Profile *	Supervisor Profile
	Contact Center *	<input checked="" type="checkbox"/>
	Referenced by	There are no references available. Reference list

Configure the following Supervisor and Agent Settings:

Primary team: PodXX_Team
 Site: LTRCCT-2813
 Teams: PodXX_Team
 Desktop-Profile: PodXX/Desktop Profile
 Multimedia profile: PodXX_Multimedia Profile
 Skill Profile: None

Supervisor Settings	Primary team	<input type="text" value="Pod00_Team"/>
Agent Settings	Site	<input type="text" value="LTRCCT-2813"/>
	Teams	<input type="text" value="Pod00_Team"/>
		<small>1 Teams Clear All</small>
	Desktop Profile	<input type="text" value="Pod00/Desktop Profile"/>
	Multimedia Profile	<input type="text" value="Pod00_Multimedia Profile"/>
	Skill Profile	<input type="text" value="None"/>
	Default DN	<input type="text"/>
	External ID	<input type="text"/>

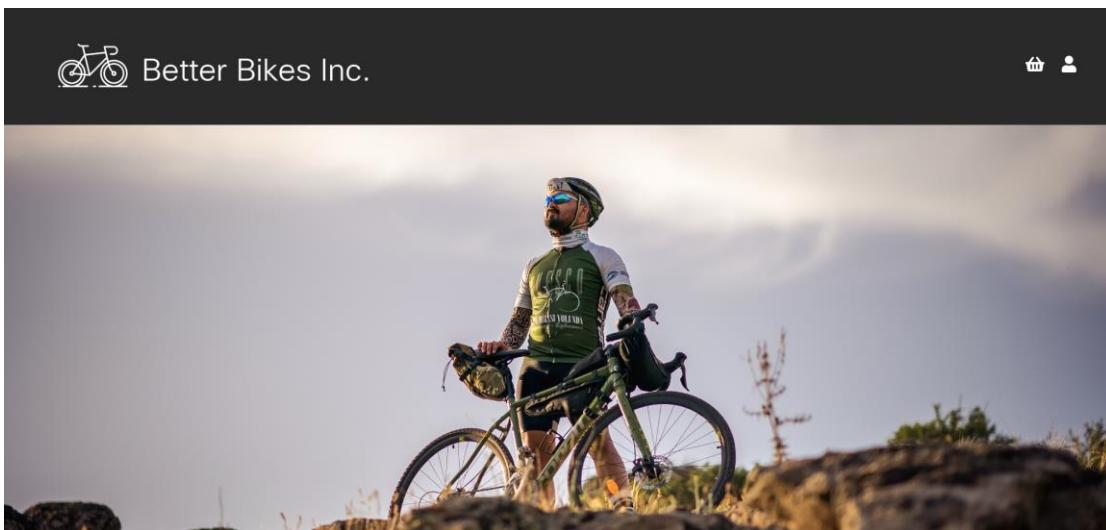
Click Save and you have done all voice configurations.

Let's test now

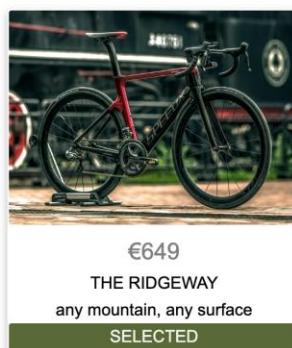
1.12 Testing the Agent Desktop

To start testing, we first need to create an account on the website created in Module 0.9.

For that please open your Glitch Website URL and click Create Account.



Confirm your order



Before you place your order, please [create an account](#).

[Create Account](#)

Please enter the following information to create an account:

- First & Last name: type in your (or any other) First and Last name
Lab Email Address: your PodXX_User Email Address
Phone number: your PodXX_User Phone Number
2nd Phone number: can be any other Phone number (your mobile phone number for example – this will be needed when testing WhatsApp in Module 3)
Customer Service Number: your PodXX Service Queue Number

Manage your account

To manage accounts please interact below.

Please enter details to create an account for your lab. You will need your lab details for some of the fields below.

Enter

Pod Number	PodXX_User	PodXX Service Queue
01	+442046200494	+442046200496
02	+442046200501	+442046200506
03	+442046200507	+442046200517
04	+442046200523	+442046200524
05	+442046200526	+442046200527
06	+442046200528	+442046200530
07	+442046200532	+442046200533
08	+442046200534	+442046200538
09	+442046200549	+442046200629
10	+442046200852	+442046200941
11	+442046200943	+442046200947
12	+442046200949	+442046200951
13	+442046200952	+442046200954
14	+442046200955	+442046200956
15	+442046200961	+442046200974
16	+442046200976	+442046200977
17	+442046200978	+442046200980
18	+442046200981	+442046200982
19	+442046200986	+442046200989
20	+442046200991	+442046200994

Note: The first field must not contain any numbers. You don't need a leading '+' for the phone numbers.

Manage your account

To manage accounts please interact below.

Please enter details to create an account for your lab. You will need your lab details for some of the fields below.

When the fields are filled, please click Enter.

The web page will confirm and your account is now created.

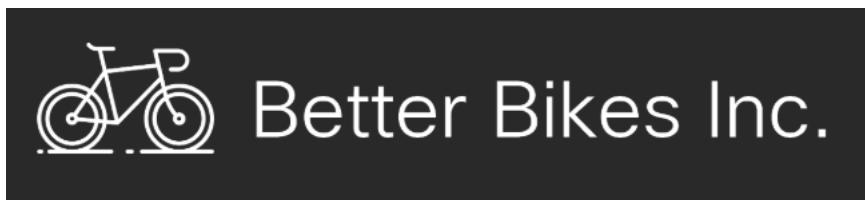
Thank you

Account created.

Thank you **Pod00 User**, your account is now created.



Click on the logo on the top left corner.



Now you can confirm your order by clicking on Enter.

Confirm your order

€649
THE RIDGEWAY
any mountain, any surface
SELECTED

To receive a delivery confirmation for your order, **please tap enter**.



The web page will confirm that an order has been placed. Your order will be created, and you'll receive a confirmation email in your PodXX_User@webex-lab.com inbox.

Thank you

Order ID - **better67100**

You should receive an email shortly with confirmation for your order, ID:
better67100.

Please check your junk folder.



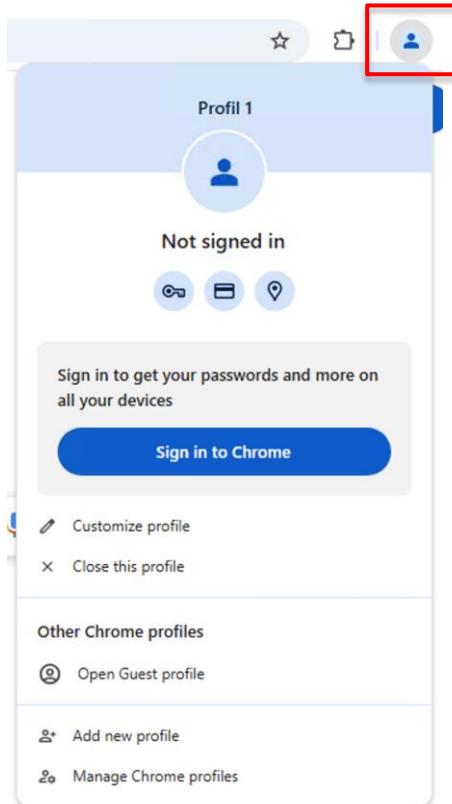
To read the mail, please login to <https://webmail.webex-lab.com/> with your PodXX User account.

Now we can login to the Agent Desktop.

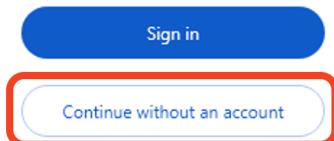
For that we need two different Chrome Profiles.

Please create two new Chrome Profiles and name Agent and Supervisor.

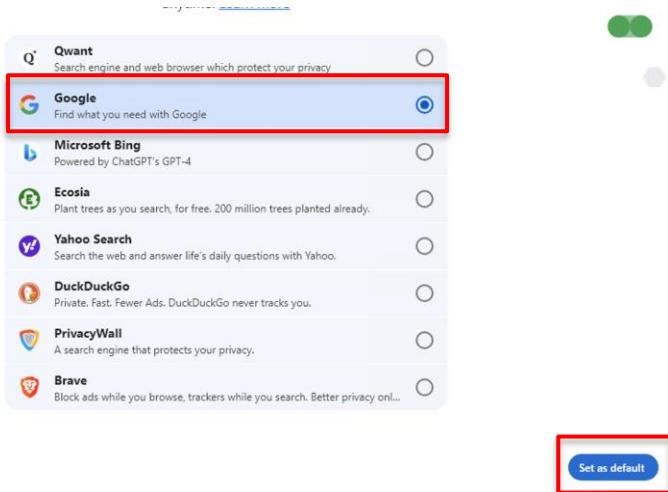
Open the Chrome browser and click on people Icon in the upper right corner.



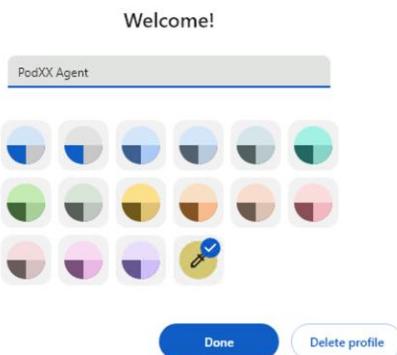
Click on Add new Profile and select Continue without an account



Choose your search machine and set it to default.



Use PodXX Agent for the profile name.



Repeat the above steps to create a profile also for Supervisor.

Open Chrome with the Agent Profile.

Open the following URL <https://desktop.wxcc-eu2.cisco.com/>

Login with your Agent Username podXX_agent@webex-lab.com and with the Password: **C1sc0Live25!**

After the login you will see the following Window.

Station Credentials

Select your telephony option i

Dial Number Extension Desktop

International Dialing Format i

+1 Enter Dial Number

Team

Pod00_Team

Remember My Credentials

Now you can choose how you are reachable:

Dial Number: can be any Number worldwide
Extension: is an internal Webex Calling Extension
Desktop: is the WebRTC Calling Connection

Please choose Desktop and Submit.

Select your telephony option i

Dial Number Extension Desktop

Desktop allows to receive inbound calls and make outdial calls through the internet.

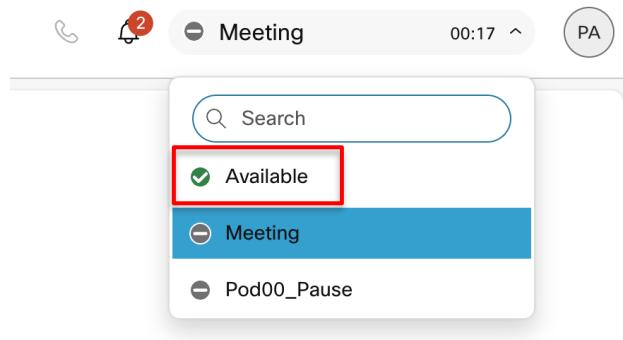
Team

Pod00_Team

Remember My Credentials

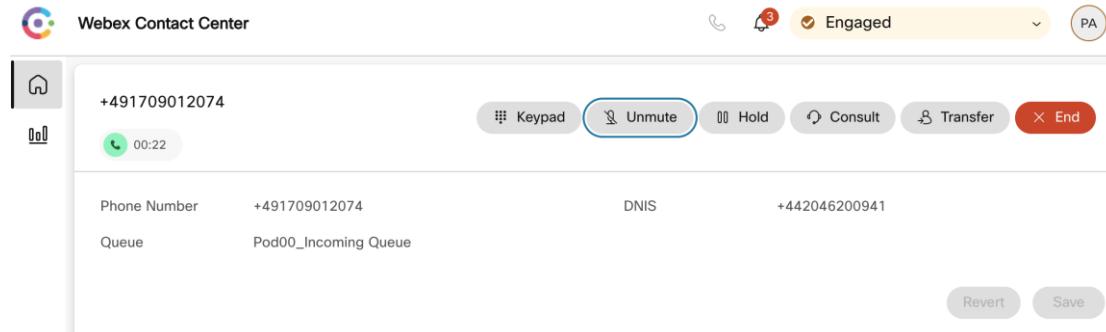
On the Status Dropdown you can see your configured Wrap-up and Idle Codes.

Please choose Available.



When you logged in in Webex App with the User Account you can dial the last 3 digits of your CC Number. When you are using your mobile phone, you can dial the E164 number like +442046200XXX.

When you call in you will hear a greeting, and your call is visible in your Agent Desktop. Please answer the call.



When you hang up you can choose your predefined WrapUp Code.

Open <https://desktop.wxcc-eu2.cisco.com/> in a second Chrome window with profile Supervisor and login with the podXX_super@webex-lab.com and with the Password: **C1sc0Live25!**

Station Credentials

Role: Supervisor and Agent

Select your telephony option: Desktop Dial Number Extension

Desktop allows to receive inbound calls and make outdial calls through the internet.

Team: Pod00_Team

Remember My Credentials

Cancel Submit

Now you see that you have more capabilities on Supervisor Desktop.

Choose Team Performance Details on the left side. Click on the three dots in the line of the Agent and change the the status to Available.

Agent Name	Agent State	Agent State Duration	Phone Number	Site	Team	Contact Status	Time In Contact Status	Interaction Duration	Actions
Pod00 Agent	Meeting	00:01:02	Desktop	LTRCCT...	Pod00_Team	-			
Pod00 Super	Meeting	00:01:11	Desktop	LTRICCT...	Pod00_Team	-			

Change Agent State

- Available
- Meeting
- Pod00_Pause

The Agent will get a notification that the status was changed from supervisor.



Congratulations, you have finished the standard lab elements for Module 1. There is an additional module element, but we recommend completing these after you have finished the standard lab in Module 3.

Skip to Module 2 [HERE](#)

1.13 Addon: Caller Identification

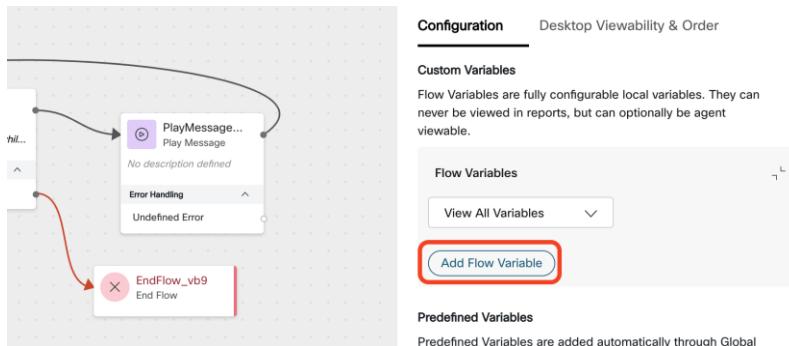
Please note, this is additional content.

This task can be skipped, but you will learn how to implement database connections for receiving more information about the caller. We recommend completing Addon modules once you have reached the end of your ‘standard’ lab if you have leftover time.

To do this, please open your current flow in the Flow Editor and set it in edit Mode.



And 2 new Variables. For that click in the grey Background and add the first Variable.



Fill out the Fields.

Add Flow Variable

Name
customername

Description

Variable Type
String

Variable Type cannot be edited once the variable is created.

Default Value

Variable value cannot contain backslashes or double quotes. Alphanumeric, spaces and other characters are allowed.

Contains Sensitive Information ⓘ

Make Agent Viewable ⓘ

Desktop Label
Customer Name

This label appears on Desktop in place of the Variable Name. Ensure that it is useful to agents.

Agent Editable ⓘ

Add a second one with these fields.

Add Flow Variable

Name
customermail

Description

Variable Type
String

Variable Type cannot be edited once the variable is created.

Default Value

Variable value cannot contain backslashes or double quotes. Alphanumeric, spaces and other characters are allowed.

Contains Sensitive Information ⓘ

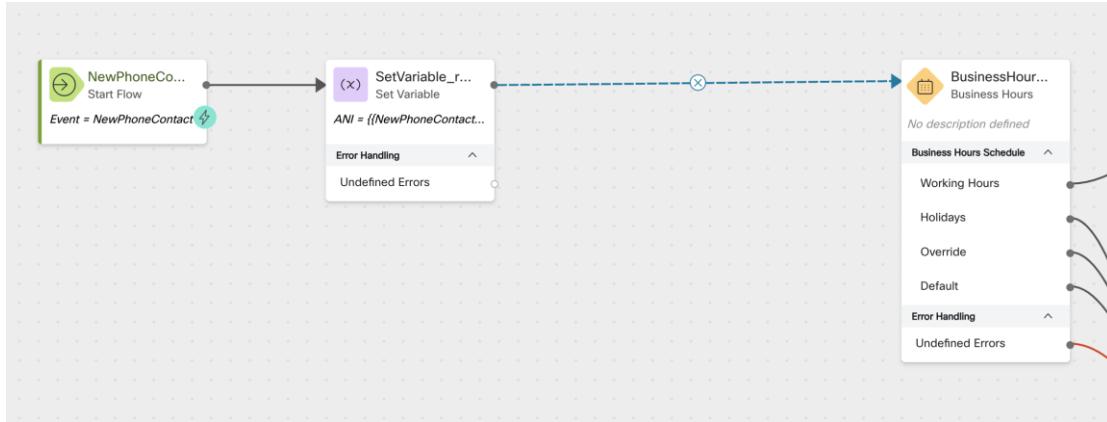
Make Agent Viewable ⓘ

Desktop Label
Customer Mail

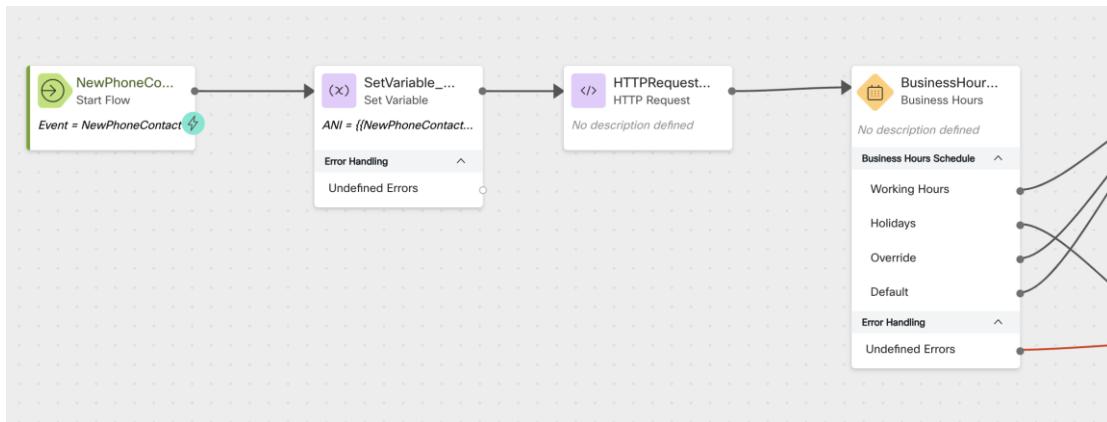
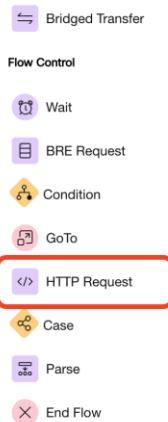
This label appears on Desktop in place of the Variable Name. Ensure that it is useful to agents.

Agent Editable ⓘ

Now move the SetVariable Node to the left and remove the Connection.



Add from the Library a HTTP Request Node in the middle and connect all three nodes.



Click on the HTTP Request Node and fill out the following fields.

Use Authenticated Endpoint: disabled

Request URL:

`https://api.airtable.com/v0/appZlsm4WdS7DZUEF/tblYOy9e3q82JEc6I/?filterByFormula=OR(mainTel = '{{ANI}}', secondaryTel = '{{ANI}}')`

Method: GET

HTTP Request Headers:

Key: Authorization

Value: Bearer

`patLmKqdPfS6dPRUT.5163d15090da940e887cda69cc25aec3b6df012018efcdd16e1c5fc757af2686`

Content Type: Application/JSON

Use Authenticated Endpoint

Request URL

https://api.airtable.com/v0/appZlsm4WdS7DZUEF/tblYOy9e3q82JEc6I/?filterByFormula=OR(mainTel = '{{ANI}}', secondaryTel = '{{ANI}}')

Method

GET

Query Parameters

Key	Value

Add New

HTTP Request Headers

Key	Value
Authorization	Bearer patLmKqdPfS6dPRUT.5163d15090da940e887cda69cc25aec3b6df012018efcdd16e1c5fc757af2686

Add New

Content Type

Application/JSON

Now we must parse the Response and fill our variables

Set Content Type to JSON and add to new Output Variable:

Output Variable: customername

Path Expression: \$.records[0].fields.customerName

Output Variable: customermail

Path Expression: \$.records[0].fields.customerEmail

Parse Settings

Content Type: JSON

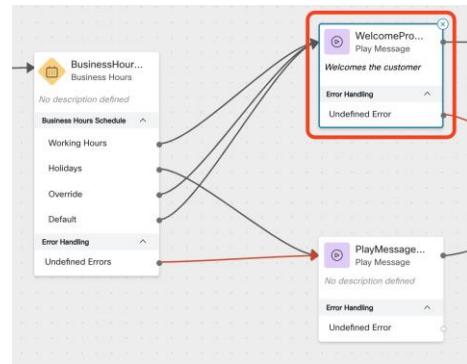
Output Variable: customername
Path Expression: \$.records[0].fields.customerName

Output Variable: customermail
Path Expression: \$.records[0].fields.customerEmail

Add New

After that click on the WelcomePrompt Node and change the Text-to-Speech Message to

Hello {{customername}}. Welcome to Webex Contact Center!



Prompt ^

Enable Text-to-Speech
Add the ability to read dynamic messages. These messages can contain variables and be used in a sequence with audio files. If typing variables, use this syntax: {{ variable }}. You can also use SSML to construct the message. If using SSML, insert it inside the <speak></speak> tags.

Connector ▾
Cisco Cloud Text-to-Speech

Add one or more audio files or text-to-speech messages to play in a sequence. ⓘ

1 Text-to-Speech Message

Hello {{customername}} Welcome to Webex Contact Center!

Add Audio File Add Audio Variable
Add Text-to-Speech Message

Now save the Flow and Publish.

Set your Agent to Available and call in again. Now you will hear your new greeting and see 2 new Variable is Agent Desktop.

+491709012074

Keypad Mute Hold Consult Transfer End

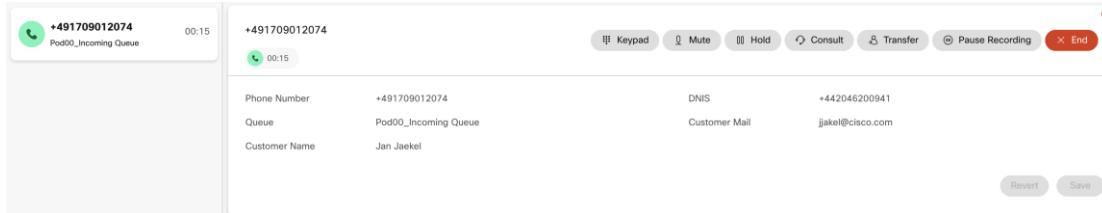
00:12

Phone Number	+491709012074	DNIS	+442046200941
Queue	Pod00_Incoming Queue	Customer Mail	jjakel@cisco.com
Customer Name	Jan Jaekel		

Revert Save

!! The solution is also on Flow List, named with Flow_PodXX_V2.

Make a new Call. Now you will hear a greeting with your personal name, and you will see it in Agent Desktop.



All Events in CJDS are preconfigured in the Flows. For the identity that calls in you see every interaction.

1. System has sent an email confirmation
2. Calls arrive in near real-time
3. System is getting all of the relevant information from your orders

Set your agent status to meeting and call in again. Wait 30 seconds before you set the Status to available. Let's see what's happen.

For example: depending on the wait time we can set individual settings.

The screenshot shows a "Customer Journey Widget" interface. At the top, there are tabs for "Customer Journey", "Contact History", and "Screen Pop". Below the tabs, a search bar says "Lookup Identity +491709012074". A "Customer Information" section shows a profile picture of Jan Jaekel and lists "Contacts within last 10 days" and "None". To the right, there are two expandable sections: "Orders for Jan Jaekel" (status: Order received, price: €649, product name: THE RIDGEWAY, contact: +491709012074, delivery date: Thu Jan 30 2025) and "Voice Queue : Pod00_Incoming Queue" (waited in queue: 32 s, team name: Pod00_Team, destination: +42046200941, contact: +491709012074, agent name: Pod00 Agent).

Congratulations, Module 1 has been finished!

Module 2 – Web Chat

For this lab we will configure a Web Chat asset for our ‘return.html’ page we built on glitch earlier. This lab will teach you how to:

- create an asset, assign the relevant settings in WxCC
- create a customer journey workflow in Webex Connect with deterministic routing
- allow visitors of your Better Bikes website to engage with Natural Language Processing and generative AI based bots, browse through some products using rich-media based, dynamic carousels
- utilize rich, HTML email templates
- escalate chats to WxCC
- enable granular-level debugging
- report to external systems

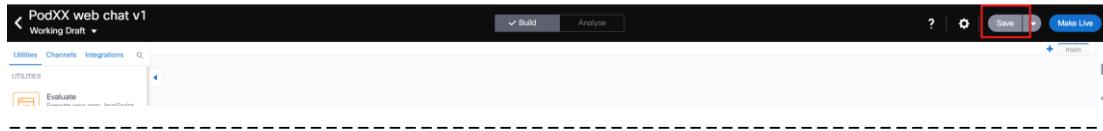
In the Addon tasks we will enhance this customer journey further by adding an email verification loop and supplementing our agents in WxCC with a generative AI based summary of the conversation.

Before we get started with this lab we will need to grab some pre-configured items from the Module 2 folder in GitHub. Head to

<https://github.com/ChrisW0lf/CL-EMEA25-LTRCCT-2813/tree/main/flows/WebexConnect/Module2> and download the following files:

- 33865_webchat-v1.workflow
- 33864_webchat-v2.workflow
- taskbotcorpus_10492.json

Additional note: Whilst you configure your flows in Webex Connect for Module 2 (and 3) we recommend you use the Save flow function semi-regularly, so progress is maintained in case you need to refresh or open the flow page.



Webex Connect Web Chat Channel Configuration

Head back to Webex Control Hub (<https://admin.webex.com>), then using the left-hand menu navigate to Contact Centre.

You will land on the WxCC overview, using quick links in the bottom right hand of the screen, click on the Webex Connect link.

This will open the Webex Connect platform in a new tab.

Using the top-left-hand menu navigate to Assets, then Apps.

Services
Create workspaces for your business cases or customer journeys. You can create unlimited number of services.

Service metrics shown below are for last 30 days

PodXX

0 0 0

This is where we manage our rich-capable channels, such as RBM and WhatsApp, which will we use later in module 3.

Click Configure New App, and then select Mobile / Web.

Apps
Configure Apps to send and receive messages from Mobile, Web, Email and Social Messaging Platforms.

Search Apps

Configure New App

App Type: All Apps

Message data shown below is for last 30 days

Channel	Apps (28)	Messages Sent	Messages Re...	Actions
PodXX	PodXX_WebChat	0	0	
	App ID : PO31090227			
	Service - PodXX			

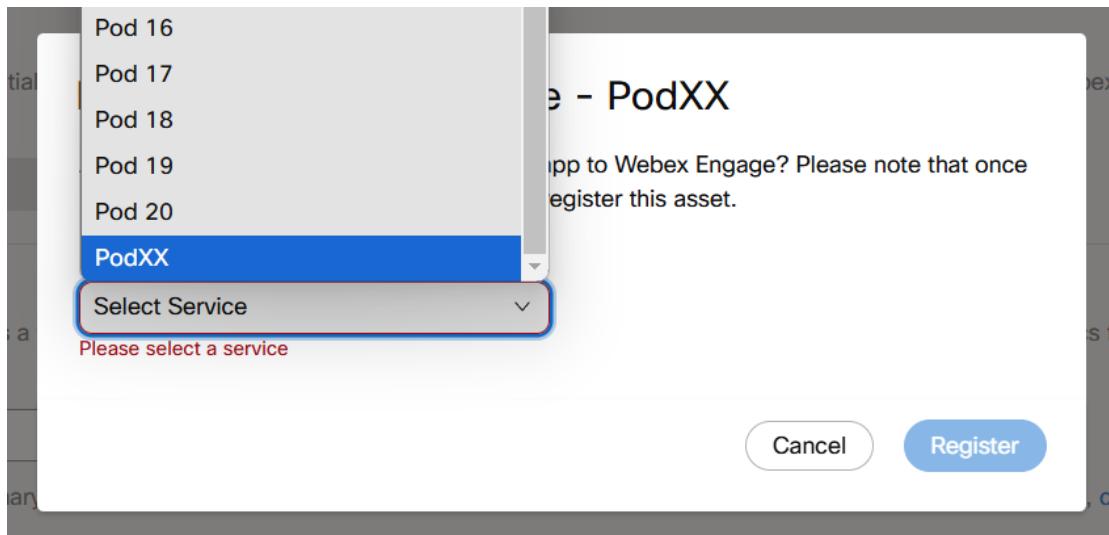
Here we will configure your web chat channel asset. Fill in the following fields:

Name:	PodXX_WebChat
Live Chat / In-App Messaging:	Enabled
Primary Transport Protocol:	MQTT
Secondary Transport Protocol:	Web Socket
Use Secured Port:	Enabled

Then click Save.

The asset is created, and now we need to link it to Webex Engage. Click the Register to Webex Engage button.

Select your service and click Register.



You've now finished configuring your app on Webex Connect. Let's head back to Webex Control Hub (<https://admin.webex.com>), then to Contact Center again from the left-hand menu.

2.1 Control Hub Configuration

In the left-hand menu now navigate to Web Chat Assets.

Contact centre overview

Current cycle agent licence usage
Billing cycle: n/a

No licence data
Please contact the partner for more licence information.

View daily details | Learn more about license consumption

What's new

Multimedia profiles
Create new and manage existing multimedia profiles.

Sites

Helpful resources

- What's new in Webex Contact Centre?
- Agent desktop user guide
- Supervisor desktop user guide
- Analyser desktop user guide
- Flow designer guide
- Google CCAI Guide

Quick Links

- Contact Centre Suite
 - Desktop
 - Analyser
 - Create new flow
 - Webex Contact Centre Management Portal

Select the Web chat asset that you configured in Webex Connect.

Web Chat assets

Create Web Chat assets in Webex Connect, then configure them here.

Asset name	Added on
BetterBikesWeb	January 16, 2025 13:25 pm
PodXX_WebChat	January 31, 2025 09:02 am

Now we will configure the asset and widget settings, then finish with an installation. Let's configure the Asset settings first:

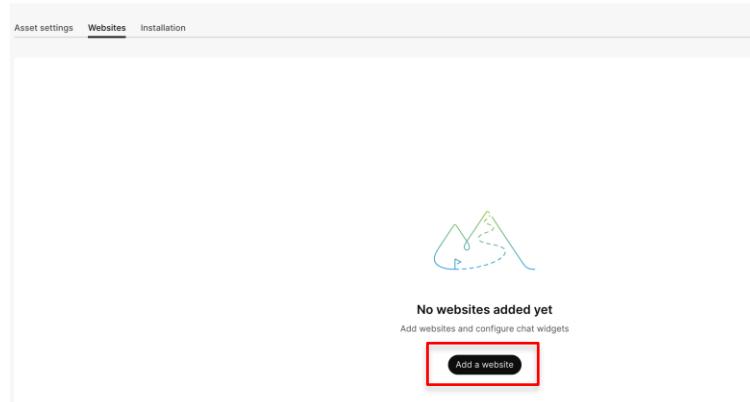
Enable the following toggles:

- Show chat conversation history to user on re-login
- Allow customer to request for chat transcript from Web Chat widget
- Clear threads for user's widget
- Play a chime sound when user receives a new message

Then hit Save.

Once the success notification has popped up, now head to Websites.

Then click Add a website.



Fill in the details:

Display Name:	Better Bikes
Byline text:	Write something for a strapline or similar
Button text:	Chat, start or similar
First message:	Hi there 🤝
Domain:	Your Glitch Website URL without the http:// in the beginning, e.g. 'first-second-third.glitch.me'

Edit website

General Appearance Availability Banned visitors

General

Basic details

Chat widget language * English (US)

Display Name * Better Bikes

Byline text Bikes, Better

Button text Chat

First message Hi there 🤝

Domain

Domain * itrccct-2813-betterbikes.glitch.me

Then hit **Next**.

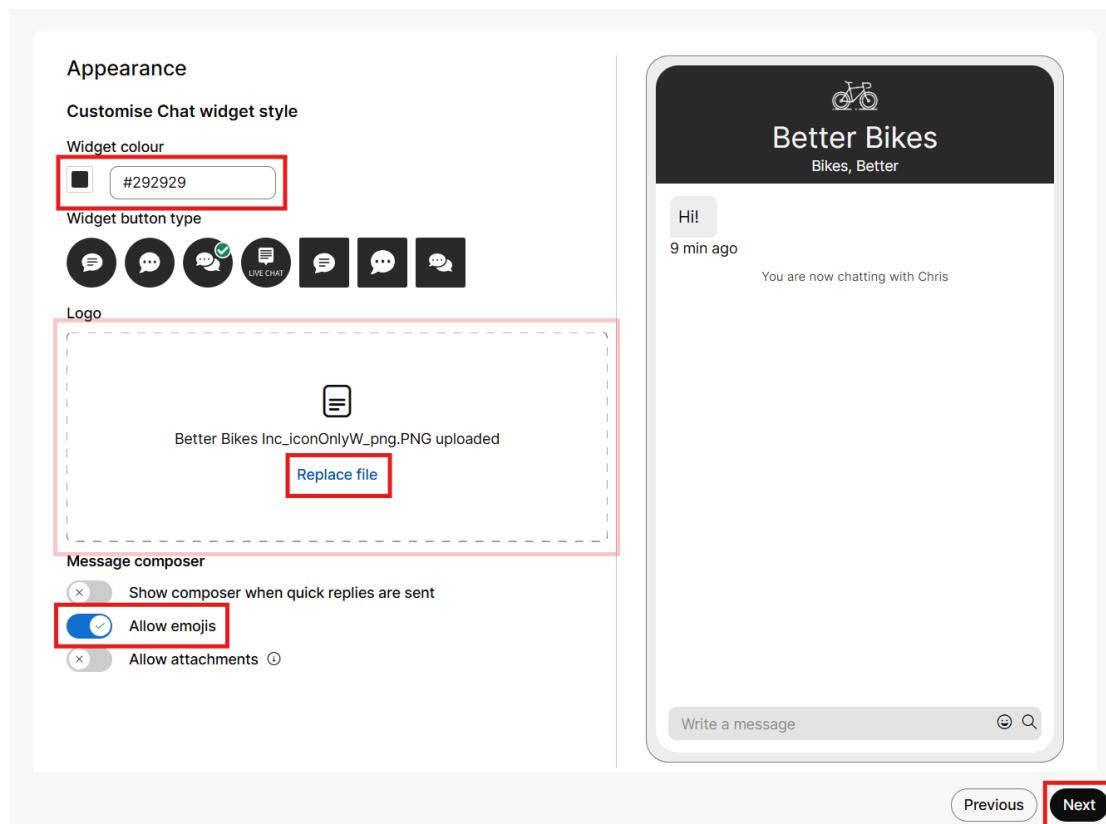
Head to https://awsprodemailassets.s3.amazonaws.com/0ae1dbc4-e5aa-46c3-8aed-55995aabf5b7/BetterBikesInc_iconOnlyW.png_11402228692944843.png, right-click on the image and save the image to your downloads file.

Now go back to the Appearance page so we can finish configuring your web chat asset.

Set Widget colour to #292929.

You can either click Replace file and upload the image you just saved or simply drag the file from your local storage into the upload box.

Then click Next.



Then click Save and next, and then Done. Your web chat asset is created and ready for installation on your glitch website.



Amsterdam | February 9-14, 2025

GO BEYOND

Head to the Installation tab and then copy the code.

Contact Centre

Overview

CUSTOMER EXPERIENCE

Channels

Queues

Business Hours

Audio prompts

Flows

Surveys

DIGITAL SETTINGS

Web Chat Assets

USER MANAGEMENT

Sites

Skill definitions

Skill profiles

Teams

User profiles

Contact centre users

Web Chat Assets

BetterBikesWeb

ID: 25EE86B6-EE82-49C0-8BF3-D6AB905CAB3A

Asset settings Websites Installation

Installation script

Place this script above the '</body>' tag in your HTML DOM

```
<div id="divicw" data-bind="E325485D-DF10-4F9-84D5-7009FBC2BE86" data-org="" data-guid="25ee86b6-ee82-49c0-8bf3-d6ab905cab3a"></div><script>var l=(t:function(t){var e="https://attachuk.iml.chat/widget/t/lmichatini.js";try{var o=new XMLHttpRequest();o.onreadystatechange=function(){if(this.readyState==4){var t=document.createElement("script");e.innerHTML=t.responseText;parentNode.insertBefore(e,t.nextSibling)}},o.open("GET",e),true);o.send();}catch(s){console.error(s)}};function(t){t.insertAdjacentHTML("afterend","<iframe id="ts_al_frm" frameborder="0" style="overflow:hidden;height: 208px;width: 394px;position: fixed;display: block;right: 48px;bottom: 12px;z-index: 9999;display:none;"></iframe>");var e=document.getElementById("ts_al_frm");var o=e.contentWindow||e.contentDocument.document||e.contentDocument};o.document.open();o.document.write("<!doctype html><html><meta charset="utf-8"><title>Untitled Document</title><style>body{font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;color: #99a0b0;font-size: 14px;}.popover_content{background-color: #fffbe;padding: 1.5rem; border-radius: 5px; width: 300px; box-shadow: 0 2px 5px 0 rgba(0, 0, 0, 0.26);position: relative;}.popover_message{font-weight: 600;color: #56627c;font-size: 16px;pull-left:float:left; clear:both;}.hdr-txt{width: 218px; margin-top: 3px; para-txt a{text-decoration: none;color: #005cd;}.close-btn{position: absolute;right:15px; top:15px;}.close-btn a{text-decoration: none;font-weight: 400; color: #56627c; font-size: 16px;}</style>
```

Copy

Let's head back to your Glitch Project page and open the return.html file. Paste the copied code where indicated, to replace the existing code on line 64.



Now after we have configured the asset we need to assign the web chat asset a queue and a team must be created so that we can elevate chats to the Contact Center. To do this, head back to Contact Center and click on Channels on the left.

We are going to use the PodXX incoming Chat channel, click on the copy icon at the right of the line.

Channel	Description	Channel type	Number of support numbers	Status	Last modified
Entry Point-1	Entry point created...	Inbound telephony	0	● Active	October 30, 2024 1...
Outdial Entry Point-1	Outdial Entry Point ...	Outbound telephony	0	● Active	October 30, 2024 1...
Outdial Transfer to ...	System generated ...	Outbound telephony	0	● Active	October 30, 2024 1...
Pod00 incoming C...		Chat	0	● Active	January 24, 2025 1...
Pod00 incoming S...		Social channel	0	● Active	January 30, 2025 1...
Pod00_Incoming C...		Inbound telephony	1	● Active	December 30, 202...
Pod20_incoming_c...		Inbound telephony	1	● Active	February 03, 2025 ...
PodXX incoming C...	PodXX incoming C...	Chat	0	● Active	January 31, 2025 1...

Change the fields with PodXX to your relevant assigned Pod number, select your asset from the Asset Name dropdown and click Create.

Create channel

Entry point	Name * <input type="text" value="PodXX incoming Chat"/> Description <input type="text" value="PodXX incoming Chat"/> Channel type * <input type="text" value="Chat"/> Asset name * <input type="text" value="PodXX"/> <div style="border: 1px solid #ccc; padding: 5px; width: 150px;"> <input type="text" value="Q"/> BetterBikesWeb PodXX <small>Europe/Amsterdam</small> </div>
Entry point settings	Service level threshold <input type="text" value="0"/> Timezone (Business hours only) * <input type="text" value="Europe/Amsterdam"/>

Create

Now we need a Queue where the Channel is assigned, click in the left sidebar on Queues.

We are going to use the PodXX incoming Chat queue, click on the copy icon at the right of the line.

Queue	Description	Queue type	Channel type	Status	Last modified
Outdial Queue-1	Outdial Queue cre...	Outdial queue	Telephony	Active	October 30, 2024 ...
Pod00 Incoming C...		Inbound queue	Chat	Active	January 24, 2025 ...
Pod00 Incoming S...		Inbound queue	Social channel	Active	January 30, 2025 ...
Pod00_Incoming Q...		Inbound queue	Telephony	Active	December 30, 202...
Pod20_Incoming_q...		Inbound queue	Telephony	Active	February 03, 2025...
PodXX Incoming C...		Inbound queue	Chat	Active	February 05, 2025 ...

Change Name with PodXX to your relevant assigned Pod number and remove the '_copy'.

Create queue

General

Name *

PodXX Incoming Chat

In Contact Routing Settings, under Group details, click on the Edit Pen for existing Group 1.

Priority	Group Name	Switching Time	Actions
1	Group 1	NA	

Remove the assignment for PodXX_Team then choose your PodXX_Team and Save.

Group 1

Priority	1
Switch to this group after ⓘ	0 Seconds

Search by name 4 Records

Team	Site	Team type
<input checked="" type="checkbox"/> PodXX_Team	LTRCCT-2813	AGENT
<input type="checkbox"/> Pod20_team	LTRCCT-2813	AGENT
<input type="checkbox"/> Team-1	LTRCCT-2813	CAPACITY

[Cancel](#) [Save](#)

Then click Create.

Your web chat is ready for using in a Webex Connect workflow.

Head back to Overview in the left-hand menu and then let's head back to Webex Connect.

≡ webex Control Hub

Search

< Main menu

Contact Centre

- Overview** (highlighted with a red box)
- CUSTOMER EXPERIENCE

Contact centre overview

Sites

Create new and manage the existing site. Associate your sites with multimedia profiles.

Teams

Create new and manage the existing team. Associate your teams with sites.

Skill profiles

Webex Contact Centre

Topic analytics

Digital channels

Webex Connect ⓘ (highlighted with a red box)

2.2 Webex Connect: AI Agent Studio

For this part of the lab, we will use a pre-configured, scripted, task-oriented bot. We will detail parts of the AI Agent Studio, however if you would like to understand how to configure AI agents in more detail, you can use the online guides [here](#).

Let's get started - in the bottom left-hand menu, hover on App Tray and select AI Agent Studio – we're going to configure a Scripted Task Bot for your lab.

The screenshot shows the Cisco AI Agent Studio interface. On the left, there's a sidebar with various icons. The 'App Tray' icon is highlighted with a red box. Below it, the 'AI Agent Studio' button is also highlighted with a red box. The main area displays 'Global Default Flows' and 'PodXX'. There's a search bar at the top and a 'Create New Service' button. The 'Service metrics shown below are for last 30 days' section includes data for 'App Tray' (12, 37.27), 'PodXX' (0, 0, 0), and 'Pod 19'.

Now let's import the agent we downloaded from GitHub. Click on Create agent.

The screenshot shows the 'AI agents' interface. At the top right, there's a blue button labeled '+ Create agent' which is highlighted with a red box. Below it, there are two cards: 'Test Pod 00' and 'PodXX Knowledge'.

Then Start from scratch and Next.

The screenshot shows the 'Create an AI agent' wizard. It has a red box around the 'Start from scratch' option, which says 'Build a new AI agent from the ground up.' At the bottom right, there's a blue 'Next' button which is highlighted with a red box.

Then Scripted, Perform actions and Next, this will create a task-oriented bot that is scripted by us, trained by our inputs.

Create an AI agent
Select the agent type and primary function

Define agent Set up profile

What type of agent are you building?

- Autonomous**
Uses generative AI to create dynamic responses.
- Scripted**
Uses natural language processing to follow your set logic and responses.

What's your agent's main function?

- Answer questions**
Delivers answers using information from a knowledge base.
- Perform actions**
Automates tasks in response to customer requests.

What to expect from your agent selection

- Customer experience:
Precise and reliable conversations that stay focused on your provided topics. Best suited for specific questions where accuracy is important.
- Setup:
Train your agent using intents, entities, and responses.

[Learn more about how AI agent works](#)

[← Previous](#) [→ Next](#)

Now we'll name bot, your 'PodXX_Task-Bot' (remember to replace the XX with your relevant pod number) and click Create.

Create an AI agent
Select the agent type and primary function

Define agent Set up profile

Profile

Agent name *

System ID *

[← Previous](#) [Create](#)

This takes us to the Script tab, on the Settings page by default. Here is where we upload the JSON file containing the bot corpus that you downloaded from GitHub earlier. You will notice that the 'Intents' listed will grow and the 'Utterances' for certain intents will increase in number too.

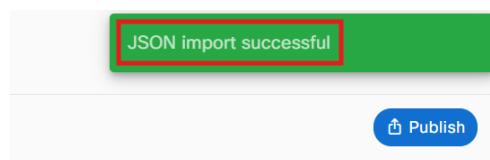
Click on the kebab menu and then Import from JSON.

The screenshot shows the 'AI agent configurations' section of the Webex AI Agent Studio. It displays a table with columns for Intent name, Utterances, Response name, Last update, and Actions. In the Actions column, there is a red box around the 'Import from JSON' button. Other buttons visible include '+ Create intent' and 'Export to JSON'.

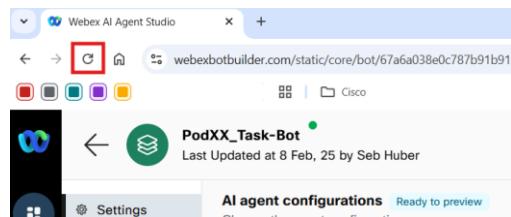
Toggle the Overwrite current intents and entities on, then click Browse and select the JSON file we downloaded from GitHub (taskbotcorpus_10492.json). Now click Import.



You will get a JSON Import Successful message pop up in the top right of your screen.



Give the page a quick refresh.



You'll also see a new intent, Purchase product, let's take a quick look at this Intent, click on the intent (anywhere in the row).

Intent name	Utterances	Response name	Last update	Actions
Goodbye	17	Goodbye	8 Feb 25, 12:11 AM	Edit Delete
Purchase product	18	purchase	8 Feb 25, 12:11 AM	Edit Delete

Here you see how the intent is configured:

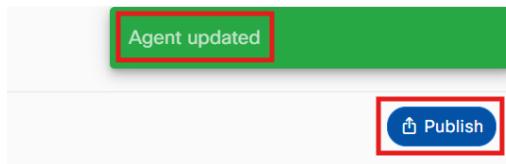
- **Intent and utterances:** these are different ways in which a user of our bot may ask or mean something, so we are training the bot to recognise the terms we expect a user to say when wanting to match this intent. A cool feature here is that we can use generative AI (gen AI) to provide us with more utterances, to help train our bot.
- **Slots:** This lists the entities we are trying to retrieve from the conversation. For our lab we are attempting to understand the product choice a user makes. If we needed more details we list entities we are aiming to attain in chronological order. Linked to each entity we have a template key, which is a template designed to ask the user to provide us with an answer that matches an entity we are looking for. Another example use case, would be for a travel booking bot, our bot recognises the user is trying to get travel details for holidays, an intent. The entities we would need to attain then are the destination/s they wish to travel to and where they are travelling from, what dates and how many people are travelling, all entities.
- **Response:** The dropdown, filled with '**purchase**' in our demo, is the template that we send to the user once we have finished fulfilling the users intent.

Let's head back to configuring our bot, click Cancel to exit the intent page.

Now we need to train our bot and make it live. Click Save changes.

The screenshot shows the 'AI agent configurations' section of a bot named 'PodXX_Task-Bot'. The 'Intents' tab is selected. A prominent blue 'Save changes' button is located in the top right corner of the main content area, which is highlighted with a red box.

You will see a success message pop up and you can now Publish. Click Publish.



Give your version a name, such as '1.0' then Publish.

This screenshot displays a modal dialog titled 'Publish changes'. It contains a text input field labeled 'Version name *' with the value '1.0' entered. At the bottom of the dialog are two buttons: 'Cancel' and a blue 'Publish' button, which is highlighted with a red box.

You will see another success message and the Save Changes button will be grayed out meaning your bot is now updated and trained – ready for use later in our lab.

Note: There will be an optional task later in this module to curate the bot's understanding and responses. Normally when working with clients we aim to deploy the bot in phases with rounds of testing to ensure the bot is well trained and ready for a production environment.

2.3 Webex Connect: Web Chat Workflow

Let's head back to Webex Connect, switch tabs in your browser.



You should be on the default landing page, Services. You can either scroll down to locate your service, named with **your** corresponding lab Pod number (e.g. PodXX), or you can search in the search bar for your pod by typing in your Pod name and hitting Enter on your PC keyboard, then click on the container with your Pod ID.

Head to Flows.

Click Create Flow.

Name your new flow 'PodXX Web Chat v1' (remember to change XX to your pod number), select your method as Upload a flow, and then click Choose File to upload your file.

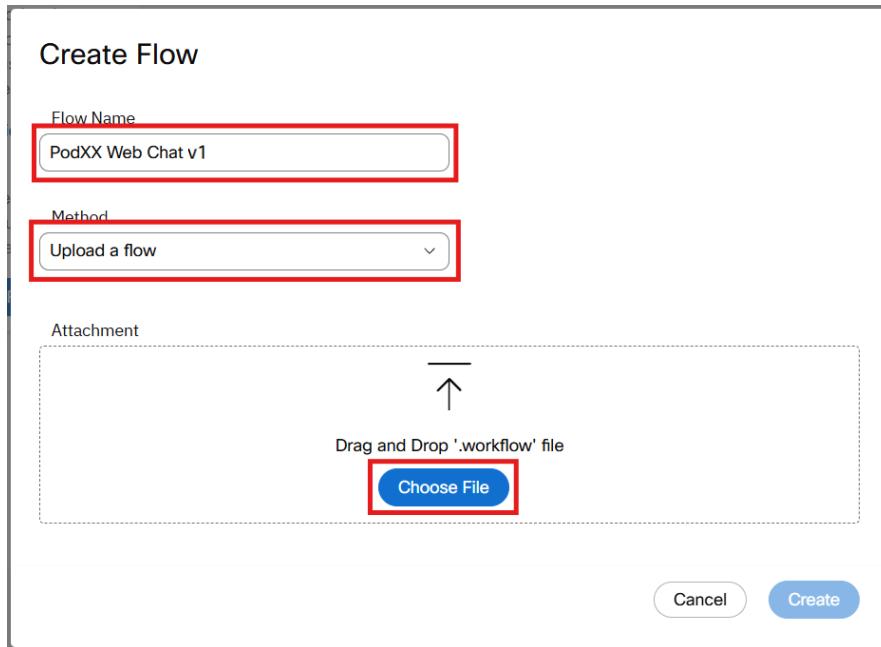
Create Flow

Flow Name

Method

Attachment

Drag and Drop '.workflow' file



We're going to use the 33865_webchat-v1.workflow file we downloaded from GitHub earlier. This contains a pre-configured flow where a lot of automated logic is already built, but we need to complete the work below before our demo can run. Select this file and then click Create.

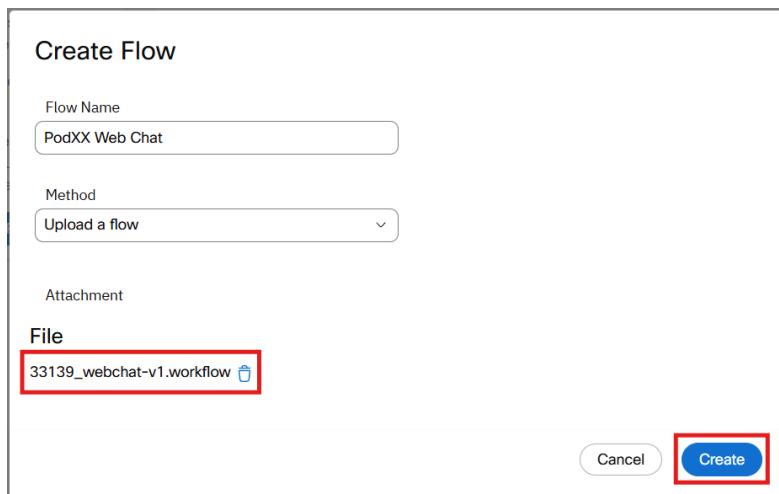
Create Flow

Flow Name

Method

Attachment

File



The system will take a moment to load and then your flow canvas will load with our Start node (A node is a ‘step’ or ‘function’ within our flow).

These nodes can handle ‘triggers’ from several entry points, such as a webhook, an inbound email, an in-app profile registration or in our circumstance, an inbound web chat message. You’ll also notice Conditions are enabled, this is where we can determine logic to tell Webex Connect when you want this flow to be used or not, such as for an inbound keyword, or only trigger when an inbound API call contains a variable that doesn’t equal a certain value and so on.

Configure APP Event

Configuration **Transition Actions (Optional)** **Data Stream (Optional)**

Configure the event and conditions to trigger this flow.

APP Incoming message

Conditions

Flow will invoke only when these conditions are met

inappmessaging.message equals typing_indicator

AND

OR

NOTE - MOBILE/WEB app can be selected before a flow is taken live

Start Node ID: 2

Cancel **Save**

Input Variables

List of variables available as input for this node

Search

Custom Variables [F33886]

Output Variables

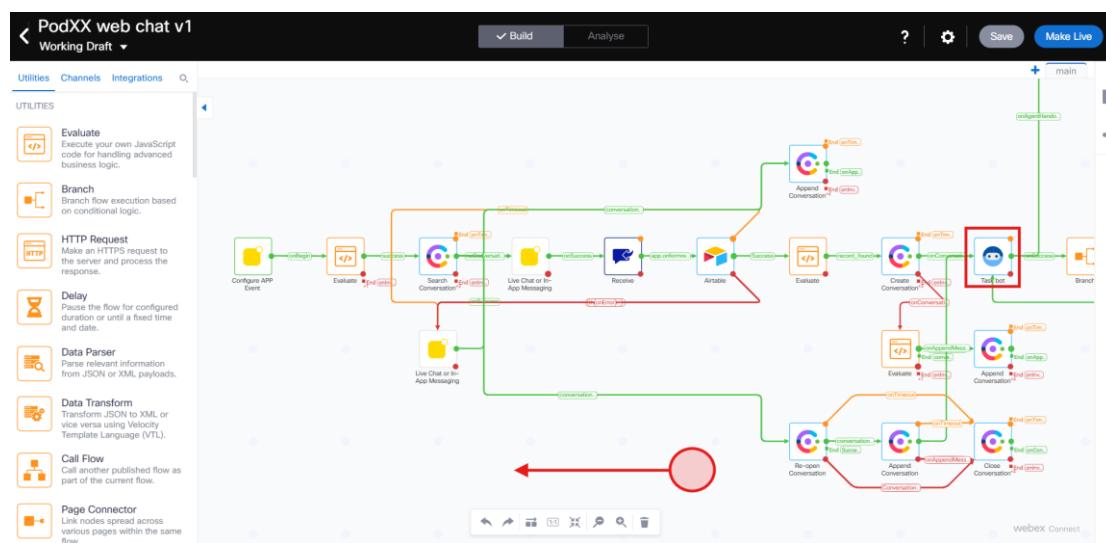
Node Outcomes

Let's click Save.

Once the Start node is saved and closed, you will see the flow canvas, relatively similar to a canvas like Miro or Visio. We will explain the full flow and the purpose of the steps a little later in the module.

Zoom out a little bit, you can do this by scrolling on your mouse, pinching your mousepad, or using the ‘-’ magnifying glass. Then move your flow to the left, you can do this by holding left click and dragging your cursor to the left.

We are looking for the Task Bot node, once located (see below for reference), double click on the node to open it.

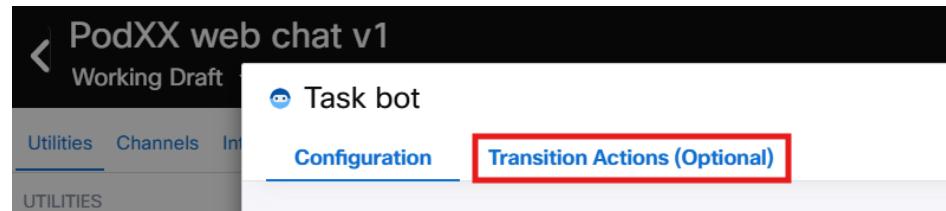


Note: Before we get into the configuration of nodes, it's good for you to understand that some nodes provide standard Output Variables by default, and we can also set Custom Variables from a node (we can also do this from the settings of the flow too – but we'll see that later in the guide). Custom Nodes are useful because we can set them from multiple locations and use them later in the flow.

All nodes other than the Start node can use input variables from other nodes that come earlier in the flow.

Finally, each of the nodes have Node Outcomes, such as a success or error path. For most of our demos, even pretty robust demos, we will mainly configure the success paths only, and some nodes will have multiple success paths. Some nodes will also require us to configure the timeout and error paths for flows to function correctly.

Let's quickly take note of what we are doing in the Transition Actions for this Task Bot node. Click on Transition Actions in the node top menu.



You will see that we are setting a variable, 'botresponse' with a value of \$(n1723.FullResponse). Let's break this variable down quick:

botresponse is a custom variable we are declaring. You can add variables using the right-hand Custom Variable dropdown in the Input Variables or on transition actions just by clicking Add and following the steps to declare a new variable. We will use this variable later in this module, and it is used in several steps throughout this flow.

The n1723 denotes the node ID for when this variable is set, you can see the Node ID at the bottom this node (bottom left-hand corner) has the same ID, meaning that this node is what is generating the value for that variable.

FullResponse is the variable identifier. To see the other variables that this node provides we can click on Output Variables in the right-hand menu.

You'll also note that the Time is set to On-leave, meaning that the variable is declared after this node has executed and we have interacted with our Task Bot.

The screenshot shows the 'Transition Actions (Optional)' configuration screen for a 'Task bot' node. The 'Action 1' section is displayed, with the 'Time' set to 'On-leave', 'Action' set to 'Set variable', 'Variable' set to 'botresponse', and 'Value' set to '\$(n1723.FullResponse)'. A red box highlights the 'Value' field. To the right, a sidebar titled 'Task Bot' shows the 'Output Variables' section, which includes 'TextResponse', 'Datastore', 'TemplateKey', 'TransactionId', 'SessionId', 'ConsumerId', and 'PotentialIntents'. The 'TextResponse' section is expanded, and 'FullResponse' is highlighted with a red box. A red arrow points from the 'Value' field in the main configuration area to the 'FullResponse' entry in the sidebar. The bottom of the screen shows 'Task bot v1.5' and 'Node ID: 1723'.

Now let's will setup this node to use the Bot we set up in 2.3. Click on Configuration in the top menu.

The screenshot shows the 'PodXX web chat v1' interface with a 'Working Draft' title. The top navigation bar has tabs for 'Utilities', 'Channels', and 'Int...'. Below this is a sub-menu with 'UTILITIES' selected. The main content area has two tabs: 'Configuration' (which is highlighted with a red box) and 'Transition Actions (Optional)'. A sub-instruction below says 'Configure node on-enter / on-leave operations'.

First, select your bot from the BOT dropdown.

Then in the MESSAGE field remove the character and using the right hand menu click on the Custom Variables dropdown*.

Select the 'messageforbot' variable, this will populate your field. Hit your space and then the backspace keys so that the field registers the character input.

This screenshot shows the configuration for a 'Task bot' named 'PodXX_Task-Bot'. The 'MESSAGE' field contains '\$(messageforbot)' and is highlighted with a red box. To the right, a 'Custom Variables' dropdown menu is open, also highlighted with a red box, showing various variables like 'conversationId', 'err_msg_toomanyrequests', etc., with 'messageforbot' selected. Other sections visible include 'CHANNEL' (set to 'Apple Messages for Business'), 'ABC USER ID' (set to 'a'), and 'Customer Parameters (Optional)' and 'Message Parameters (Optional)' sections.

Now click on the CHANNEL dropdown and select Live Chat / In-App Messaging. Remove the character from the UNIQUE ID field and in the right-hand menu, collapse the Custom Variables dropdown by clicking on the menu title or the dropdown icon.

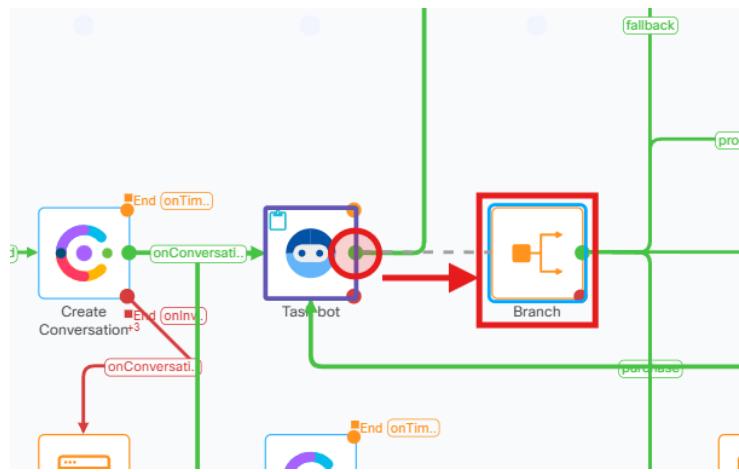
Now click on Start in the right hand Input Variables menu, and select the inappmessaging.userId variable, and again use the space and then the backspace keys.

The screenshot shows the configuration of a 'Task bot' named 'Process Message'. The 'CHANNEL' dropdown is set to 'Live Chat / In-App Messaging' and the 'UNIQUE ID' field contains the expression \${n2.inappmessaging.userId}. On the right, the 'Input Variables' sidebar shows a list of variables under the 'Start' section, with 'inappmessaging.userId' being highlighted. The 'Save' button at the bottom right is also highlighted.

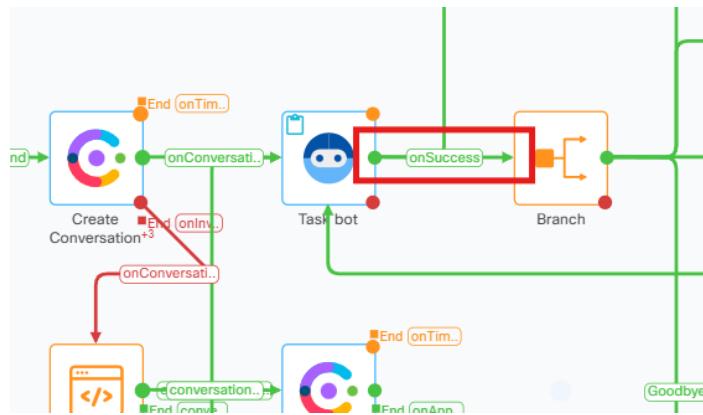
Now you can click Save.

Now the node is configured let's connect it to the Branch node to the right.

Click and hold on the green dot on the right of the node, now drag your cursor to the Branch node.



You'll then see a green onSuccess line that shows that when the Task bot node has executed successfully the next step in the flow will be the Branch node.

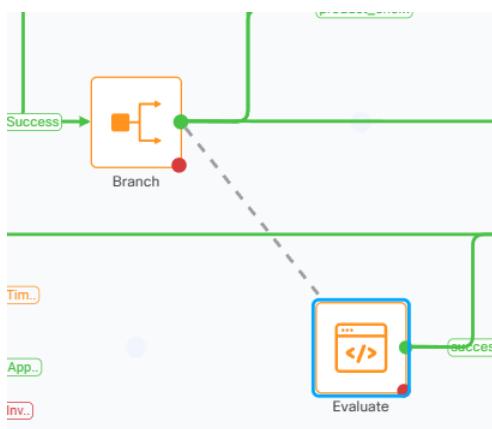


Now let's open the Branch node.

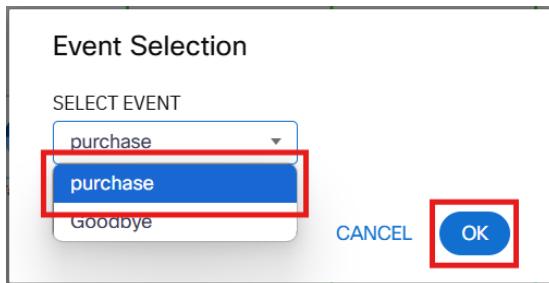
You'll notice there are 4 branches, fallback, product_choice, purchase and Goodbye, and each branch has conditional logic, e.g. the Goodbye branch is only used when \$(n1723.TemplateKey) equals 'Goodbye'. To leave the node without making any changes click on Cancel.

The screenshot shows the 'Branch' configuration screen. It lists four branches: 'fallback', 'product_choice', 'purchase', and 'Goodbye'. The 'Goodbye' branch is currently selected. Below it, there is a conditional logic section with fields for 'Variable' (\$n1723.TemplateKey), 'Condition' (Equals), and 'Value' (Goodbye). There are also 'AND' and 'OR' buttons. A red box highlights this conditional logic section. At the bottom of the main panel, there is a note: 'None of the above'. On the right side, there is a sidebar titled 'Input Variables' which lists various variables and their descriptions. At the bottom of the screen, there are 'Test', 'Cancel' (button), and 'Save' buttons.

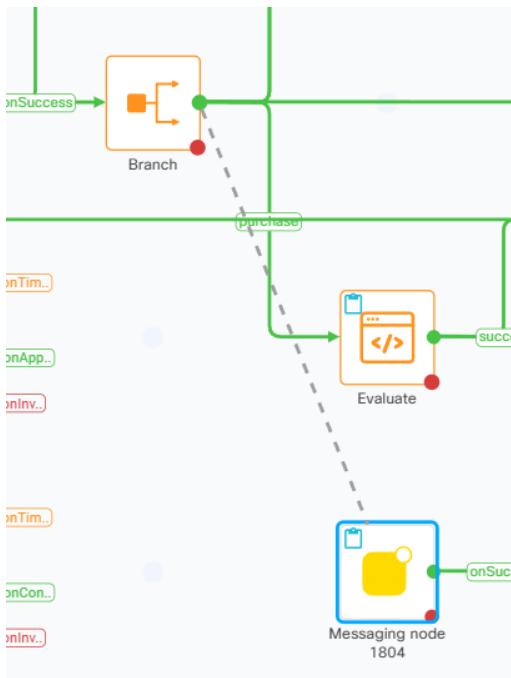
There are two branches that we want to use that are currently not connected to a node. Click on the green dot for the Branch node and drag it to the Evaluate node, which is just below and to the right.



Because there are several branches available, we get an Event Selection. In the Select Event dropdown ensure that the purchase option is selected and click OK.



We're going to connect the last branch out of the Branch node to the Messaging node 1804 located below the Evaluate node.



The Branch node is now configured with all branches utilized. This is a great node to highlight the low-code / no-code capabilities of Webex Connect, with simple and understandable conditional logic that doesn't require you to understand a coding language. We will however use the Evaluate node later, in Module 3, which allows us to use JavaScript.

To the right of the Evaluate node is an Airtable node that has no Success path configured. Connect this to the Email node to its right, then open the Email node.

Scroll down until you see the Message Source section, and under Template click on the X to the right of the 'ABC' input.

The screenshot shows the 'Email' configuration page. In the 'Template' input field, the value 'ABC' is entered, and a red box highlights the 'X' button to its right. To the right of the main form, there is a sidebar titled 'Input Variables' which lists several items with their corresponding Node IDs:

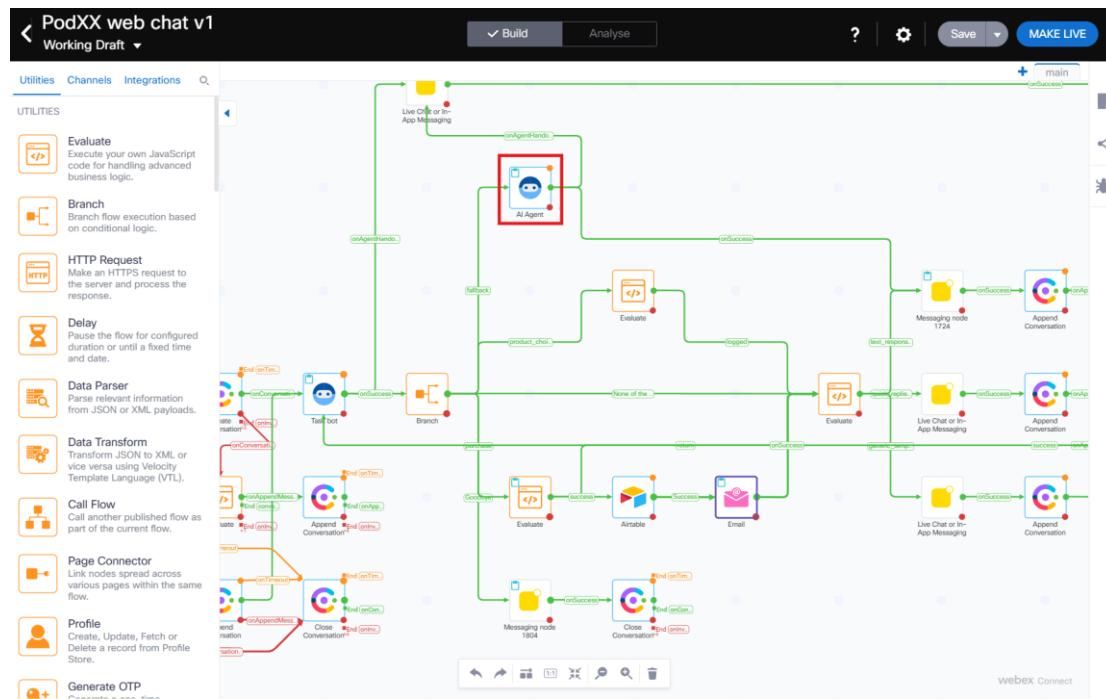
- Custom Variables [F33865]
- Start Node ID: 2
- Evaluate Node ID: 9
- Search Conversation Node ID: 613
- Create Conversation Node ID: 665
- Re-Open Conversat... Node ID: 676
- Append Conversation

Click the dropdown and select the BetterBikesDeliveryTemplate_additionalProduct template, then click Save.

The screenshot shows a dropdown menu for selecting a template. The option 'BetterBikesDeliveryTemplate_additionalProduct' is highlighted with a red box. At the bottom right of the screen, the 'Save' button is also highlighted with a red box.

We will now receive templated HTML based emails when we 'place an order' in our demo.

In the next step we'll make use of the botresponse custom variable we used earlier, locate and open the AI Agent node (see screenshot below for reference)



You'll notice that this AGENT TYPE is a gen AI powered, Autonomous. This bot will provide responses for interactions that the task bot does not understand, essentially what the task bot is not trained to handle, otherwise known as the task bot's Default fallback intent.

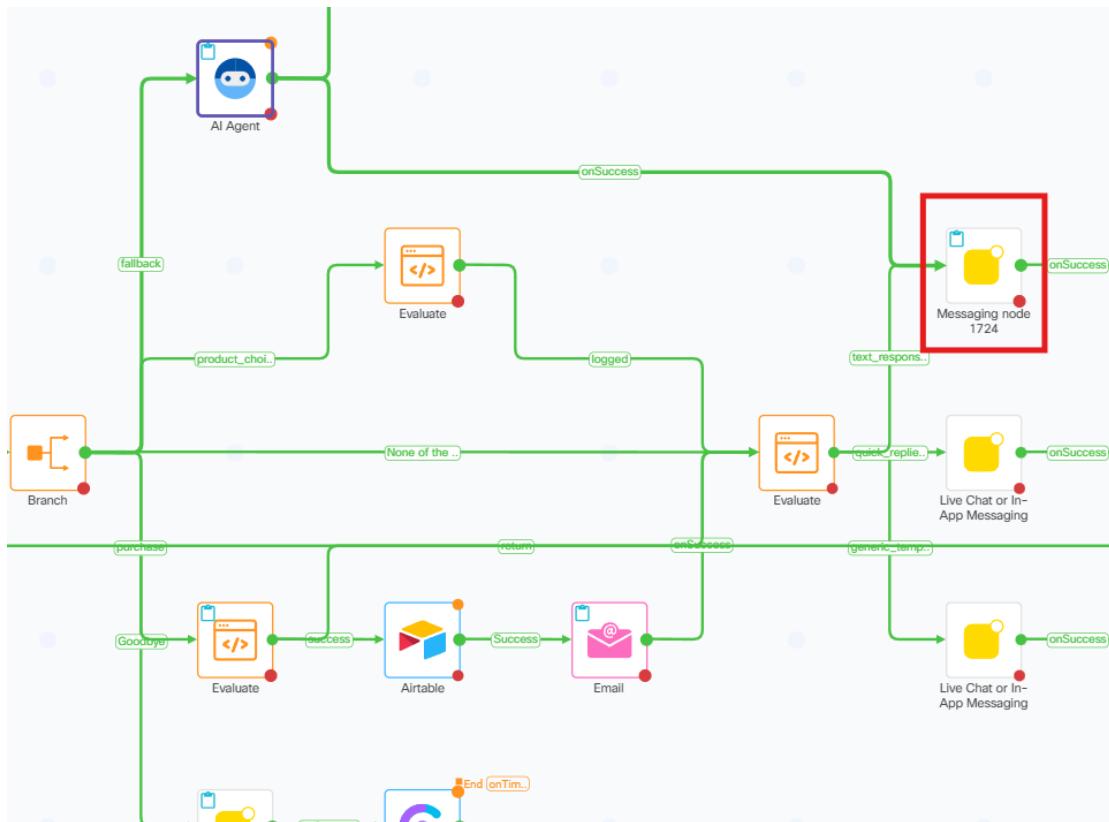
Let's head to Transition Actions.

We need to set the botresponse to use the Autonomous bot's response. In the Variable dropdown select botresponse and set the Value to TextResponse from this node's Output Variables.

The screenshot shows the AI Agent configuration interface. Under 'Action 1', there is a 'Set variable' action. The 'Variable' dropdown is set to 'botresponse' and the 'Value' dropdown is set to '\$(n1768.TextResponse)'. Both dropdowns are highlighted with red boxes. The 'Save' button at the bottom right is also highlighted with a red box. The right sidebar shows 'Output Variables' with 'TextResponse' selected, also highlighted with a red box.

To finish, click Save

Locate and open Messaging node 1724 (see screenshot below for reference).



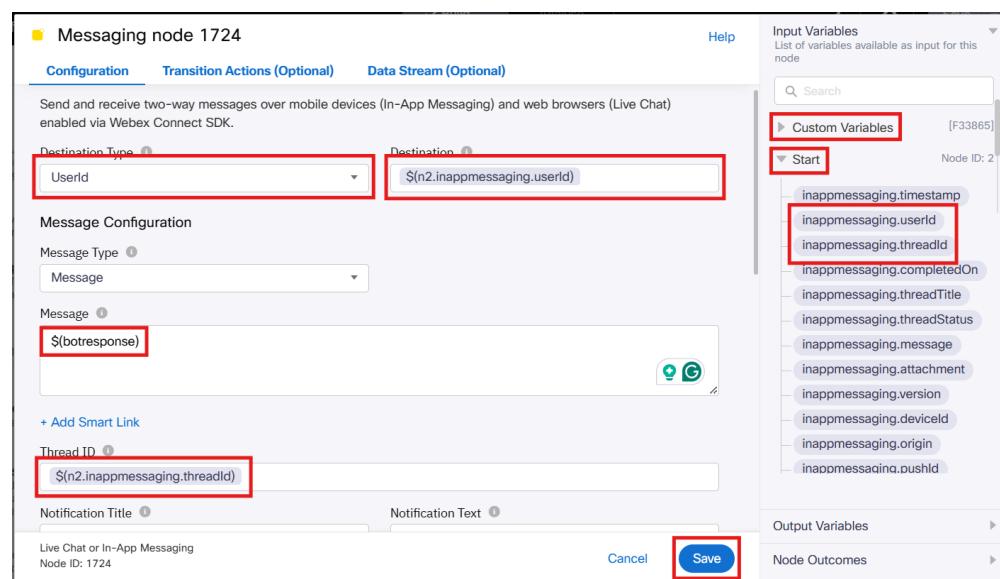
We'll configure this message node to ensure that if a text response from either the task or autonomous bot is needed than we can dynamically handle the message content.

First set the Destination Type dropdown to UserId.

Change the Destination to the inappmessaging.userId from the Start node under Input Variables.

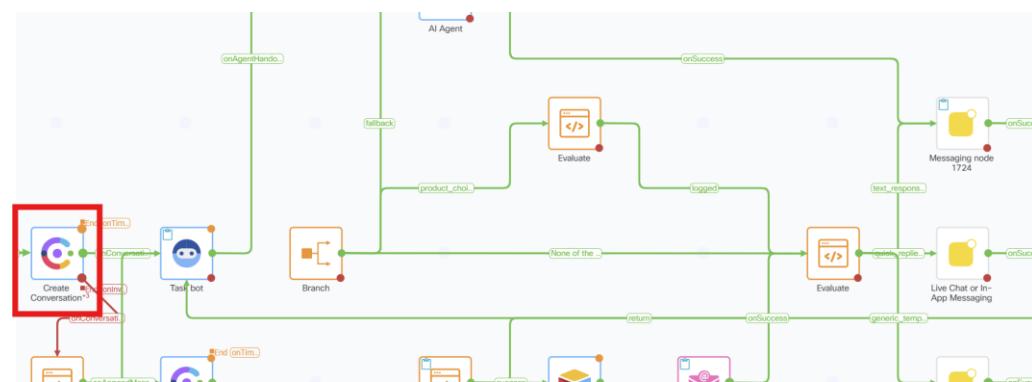
Then remove the 'a' from the Message field and select the botresponse from the Custom Variables, use the right-side menu again.

Then we need to replace the 'a' in the Thread ID field with the inappmessaging.threadId variable, using the Start node variables again.



Now click Save.

The final two nodes we need to update will escalate our conversation to an agent when needed. Zoom out, locate and open the Create Conversation node (see screenshot below for reference).



For this node we need to update the LIVECHAT Website Domain to the URL for our Glitch web page, removing the https:// at the start.

Create Conversation

Configuration **Transition Actions (Optional)**

Method Name

Create Conversation

Node Authentication

Auth

Conversation Details

CHANNEL

Livechat

LIVECHAT APP ID

`$(n2.inappmessaging.appId)`

LIVECHAT Website Domain

puzzling-broken-carpet.glitch.me

Now click Save to exit the node.

The last node we need to configure is the Queue Task node which is found on the right-side of this flow (see screenshot below for reference). Locate and open the node.



Click on the Queue Name dropdown and select the queue you configured in 2.2.

Queue Task

Configuration **Transition Actions (Optional)**

Method Name: Queue task

Node Authentication: Auth

Task ID: \${flid}

Conversation ID: \${conversationId}

Media Type: **Search**

- Pod00 Incoming Chat
- PodXX Incoming Chat
- PodXX Incoming Chat** (selected)

MEDIA CHANNEL: Livechat

Queue routing type: Longest available agent

Queue Task: v1.1

Node ID: 736

Test Cancel **Save**

Input Variables
List of variables available as input for this node

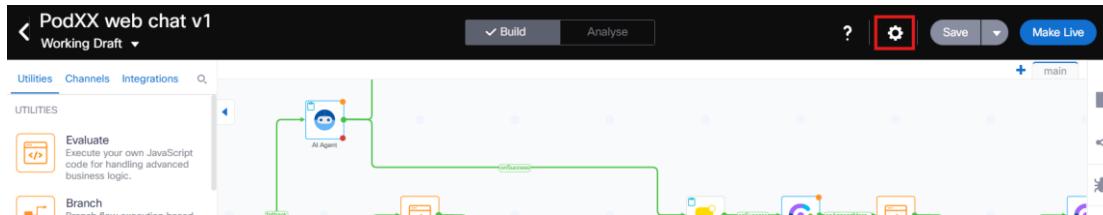
- Custom Variables [F33865]
- Start Node ID: 2
- Evaluate Node ID: 9
- Search Conversation Node ID: 613
- Create Conversation Node ID: 665
- Re-Open Conversation Node ID: 676
- Append Conversation Node ID: 698

Output Variables

Node Outcomes

Click Save.

Now you've finished configuring all the steps for the flow, click on the settings cog in the top right of the UI.



Unfortunately, a commonly essential step when building demos is having to debug!

Toggle Descriptive logs so that they are Enabled and set Enabled for to 1440 minutes (this is the maximum time frame we can enable Descriptive Logs for at a time, however you can always re-enable this should you need to).

Flow Settings

General Custom Logs Flow Outcomes Custom Variables

Flow Name: PodXX web chat v1

Description (Optional): Enter Description here

Advanced settings (optional)
Set flow behaviour when running multiple instances of flows in parallel.

Correlation ID: \$(corrid)

Descriptive logs It can take up to 2 minutes to take effect
Activate to capture information required for developers when trying to debug a problem, helpful when testing your service. Descriptive logs capture complete transaction details including sequence of all activities within flows, send/receive message payload, HTTP request & response entities which may include sensitive PII data of your customers. Descriptive logs captures data for latest live flow version.

Enabled for: **1440** Mins or 1000 Transaction Note: Descriptive logs for debugging will be auto-disabled in maximum 1440 minutes

Flow id: 33865

Cancel **Save**

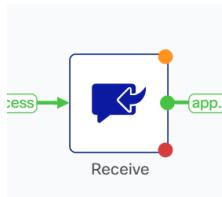
Input Variables
List of variables available as input for this node

- Custom Variables [F33865]
- Start Node ID: 2
- Evaluate Node ID: 9
- Evaluate Node ID: 51
- Search Conversation Node ID: 613
- Append Conversation Node ID: 624
- Create Conversation Node ID: 665
- Re-Open Conversation Node ID: 676
- Append Conversation Node ID: 698
- Close Conversation Node ID: 699
- Queue Task Node ID: 736
- Append Conversation Node ID: 738
- Close Conversation Node ID: 767
- Create Task Node ID: 1181

Click Save.

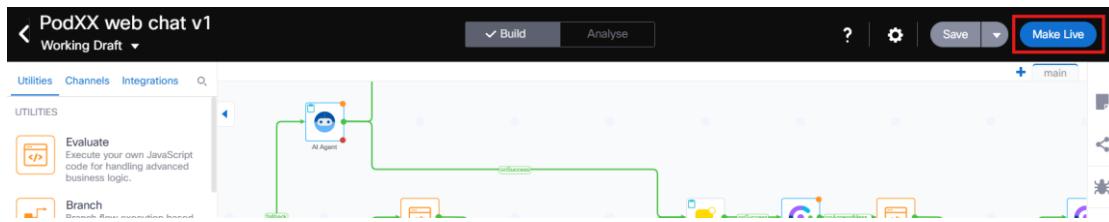
Note: You are now almost ready to publish the flow
When publishing any flows on Webex Connect that you have imported, **you will need to open and save all receive nodes**, so that the flow recognizes the nodes as configured before we can make the flow live.

To do this locate a Receive node in the flowbuilder UI (see example below)



Double click on these to open the node, and then click Save to close the node again.

OK, let's publish the flow. Click Make Live in the top right of your UI.



Now let's select our PodXX_WebChat for the Mobile & Web App asset, then PodXXMailEntry for the Email asset (remember to replace the XX for your own pod number), and click Make Live.

PodXX web chat v1 - Make Live Configuration - Version 1 Help

Assets Configuration
Select apps and define values for custom variables or dynamic numbers that you have created in the flow. Apps will only need to be selected if you have used a send or a receive node and selected one of the app based channels such as Mobile/Web apps, Messenger, Email etc. Dynamic numbers will only be needed when you have one or more SMS receive node with a dynamic number selected.

App Selection ⓘ

Type	Application
	PodXX_WebChat
Type	Application
	PodXXMailEntry

Comments (Optional) ⓘ

Cancel Make Live

Congratulations - you've built your first digital channels workflow! 🎉

Let's recap what you have configured:

- You enabled a Task Bot to drive logic and provide flexible, NLP based logic, which when coupled with the Branch node can handle several user intents with a variety of flow outcomes. A powerful combination of using AI and deterministic logic.
- You connected the Airtable database with our Email template to ensure a personalized Email, making the most of rich and dynamic, HTML-based emails.
- You then enhanced the AI capabilities of the flow, ensuring the generative AI based Autonomous agent was able to send messages using the same steps (nodes) as the task bot, reducing engineering overhead, streamlining processes and most importantly greatly improve the AI based capabilities of this bot powered flow.

Once the flow is Live you're ready to test your workflow.



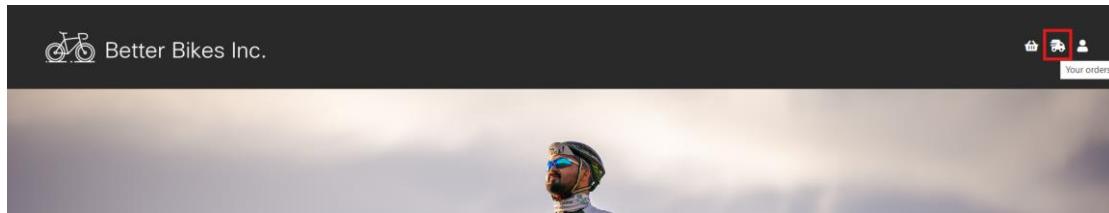
Note: You may need to refresh your browser if Live doesn't show after a minute.

2.4 Testing the Web Chat Demo

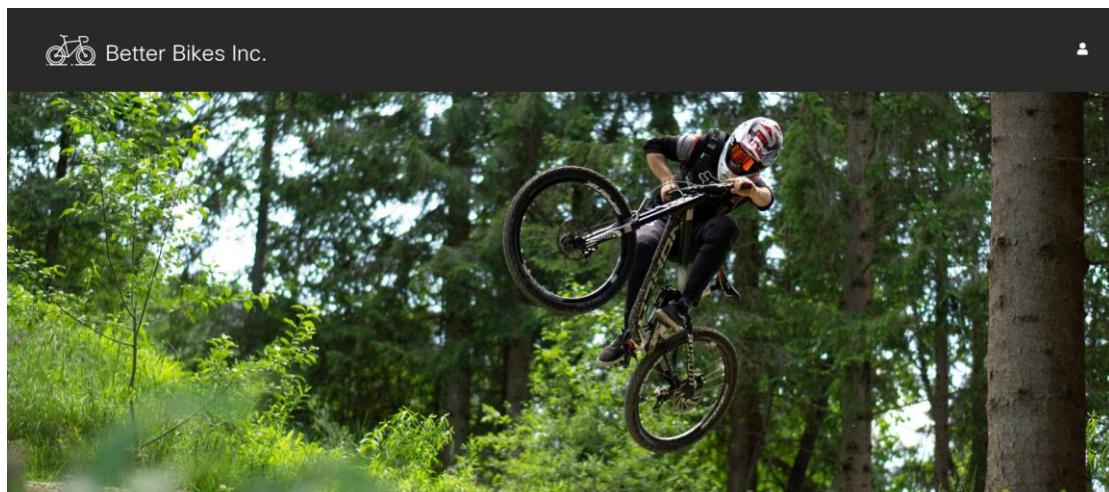
To test we'll use the Webex Contact Centre UI like we did in the last Module (1.12), you will likely need to log in again, this is to test our bot is configured correctly and able to escalate if needed.

Make sure your agent is set to Available.

We also need to head back to our glitch website. Click on the delivery truck icon in the top-right of the website, navigating to the 'return.html' page we configured earlier in 2.2, with the web chat widget.



Now click on the floating web chat widget in the bottom right of the web page.

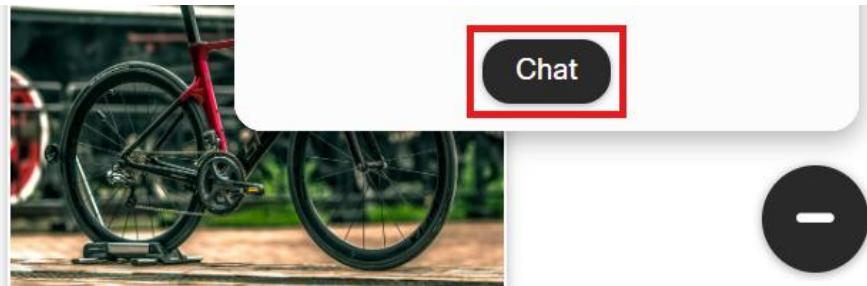


Your orders

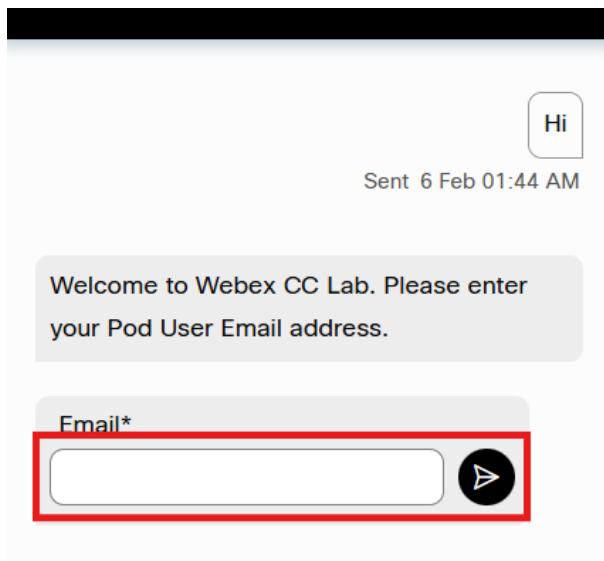
Welcome back Seb Huber



Click 'Chat'.



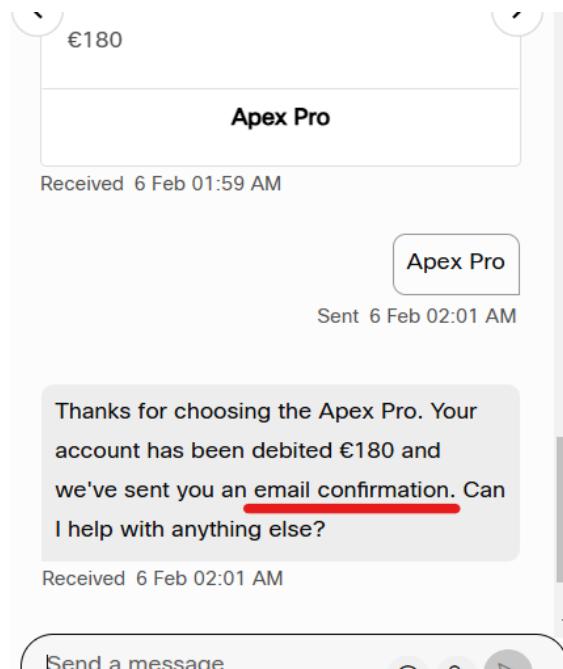
Now enter your email (your pod email address) and hit enter.



Now you're verified you can ask the Bot questions about Better Bikes, the additional products, helmet, the founder, etc

The intent the bot is looking for to progress your demo further first is to place an order. We recommend you ask the bot something similar to "I'd like make an order".

This will present you with a carousel of products that you can look through and select. When you select a product, it will confirm and send you an email.



Switch over to the <https://webmail.webex-lab.com/> tab and check your inbox – you should have a new order with the product you selected.

Order better6264 Confirmation

From LTRCCT2813 on 2025-02-06 02:01

Replies Headers Plain text

Better Bikes Inc.

Your delivery details for:
Better Bikes Inc.

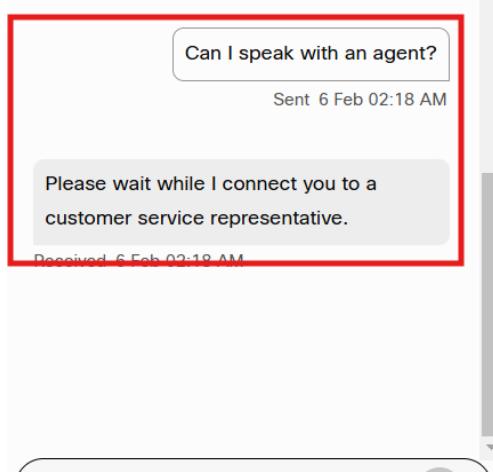
Hi Seb Huber,

Order - **better6264**, is being processed. Your delivery of Apex Pro™ is due Sat Feb 08 2025.

If you have questions related to your delivery please use one of the contact options below, choose your channel and press the button, or scan the relevant QR code.

Apex Pro™:

Head back to your web chat, the next intent you want to match is the agent handover, we recommend you ask something like “I’d like an agent please”. A successful handover will look something like this:

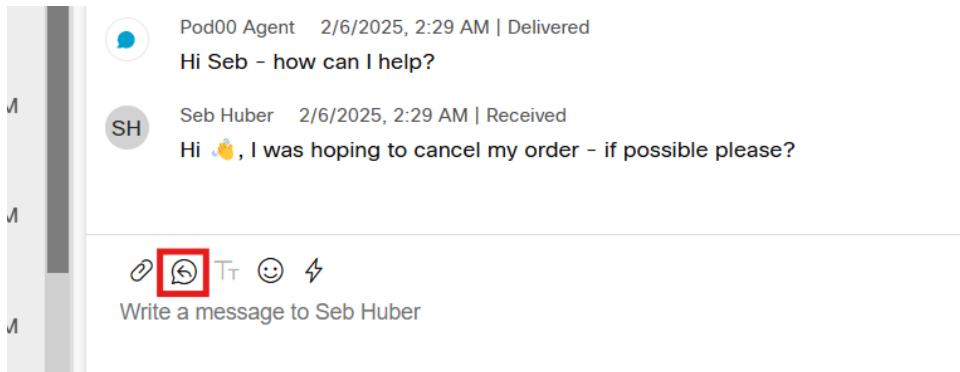


Head to the WxCC UI, where if you are Available, you should now have a chat waiting to be accepted. Click Accept, and the chat will load in your UI.



Once it's loaded you can message back and forth.

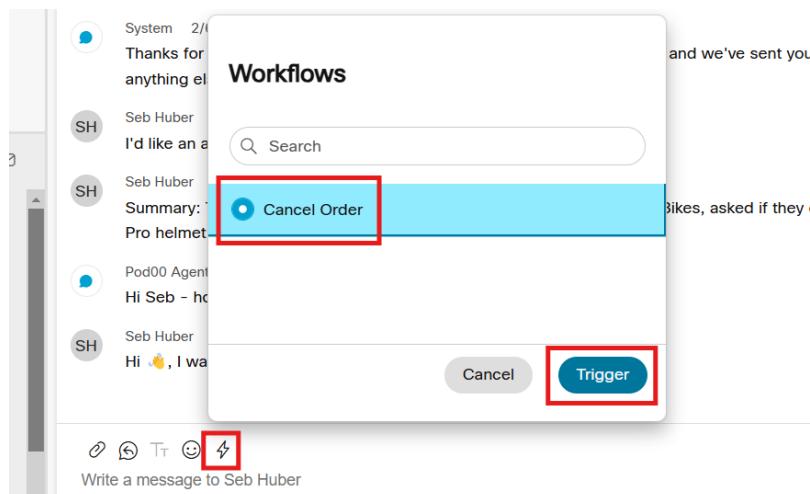
To use a template, click on the reply icon (see screenshot below for reference)



You can also cancel order using the Trigger Workflow. Click on the lightning icon (see below screenshot for reference)



Then select Cancel Order and Trigger. This is a pre-configured rule that uses the 'Website Order Cancelation' workflow, found in the Global Default Flows on Webex Connect.



If you now check your email inbox, you will also see that you have a cancellation email and that will delete an order record in the Airtable database.

When you're done with your chat, remember to End the chat and submit your wrap-up.

Congratulations – you've finished all the standard elements in this Module! 🎉

There are two additional module elements, but we recommend completing these after you have finished the standard lab in Module 3.

Skip to Module 3 [HERE](#)

2.5 Addon: Ongoing bot curation

Please note, this is additional content.

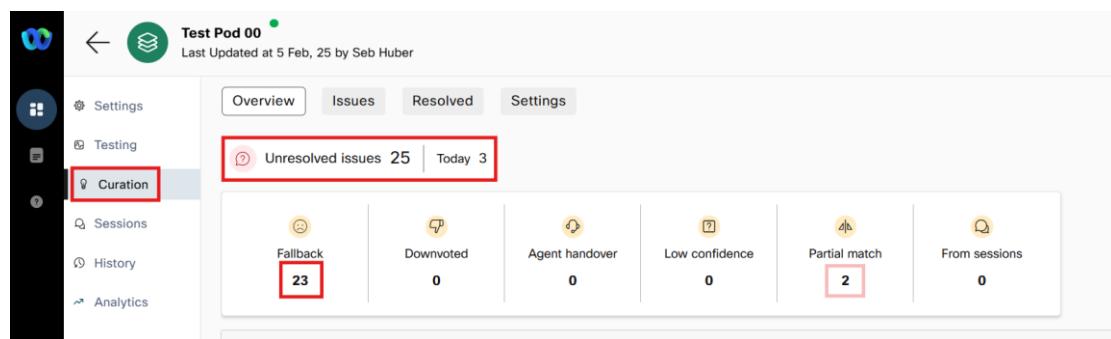
This task can be skipped, but you will learn how to implement database connections for receiving more information about the caller. We recommend completing Addon modules once you have reached the end of your ‘standard’ lab if you have leftover time.

Before you curate your bot, run through the Web Chat demo a few times, try asking the same questions in varying methods. By interacting with our bot, time and time again, we are generating the data for our bot to be fine-tuned with, so hopefully the bot will struggle to understand some of the questions you ask.

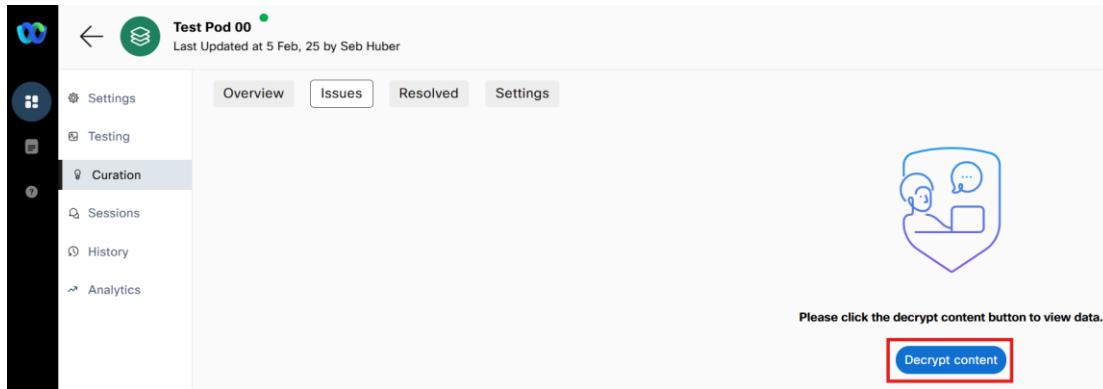
Once that’s done let’s head back to our bot, in Webex Connect (hover on the App Tray, bottom left-hand menu, and click on AI Agent Studio) or navigate to the relevant tab if you have still this open.

From the dashboard select your Pod corresponding bot that you configured in 2.3.

In the left-side menu head to Curation. We land on the Overview page, if we have successfully challenged the bot enough, then we should see some numbers against Fallback and possibly even Partial match. Let’s click on the Fallback segment.



Now Decrypt Content.



Now we have a list of Issues ready for us to resolve and help train our bot. Let's go through them one-by-one and either Link to an existing intent or Ignore them. Our advise is that if you are unsure if a message matches your intent, in other words, it was too different from the intent you think it might match, then you are better to be cautious and choose to ignore.

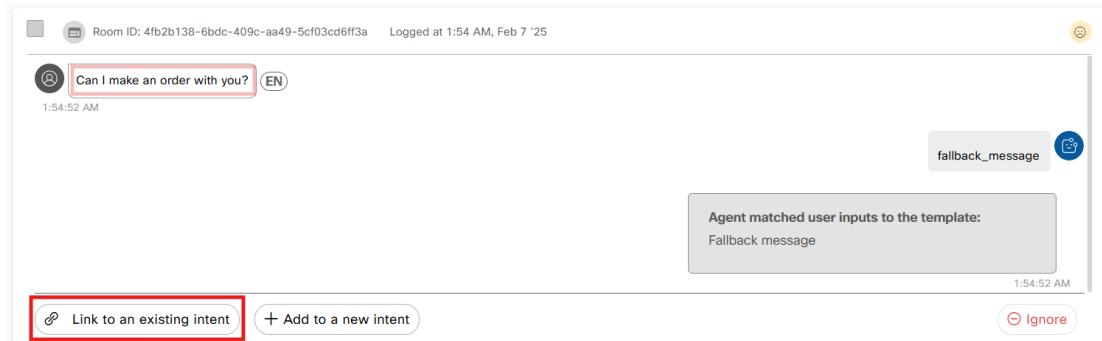
You can also select multiple Issues at once and choose either action to link with existing intents or ignore them. For this lab, let's review them one-by-one (feel free to explore handling multiple issues after).

Link to an existing intent

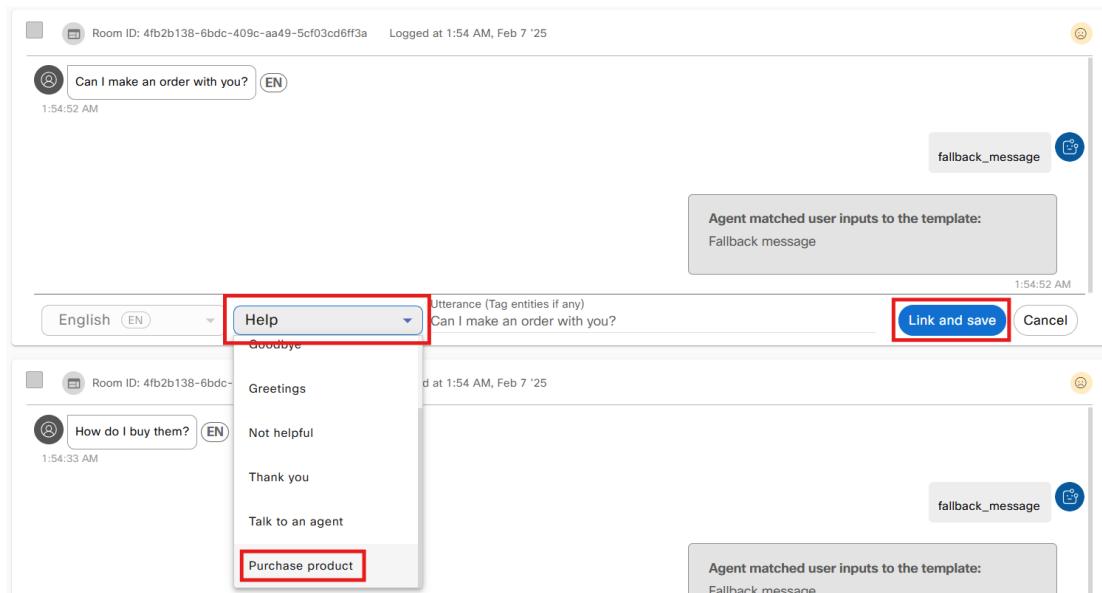
To link the message with existing intents, review the messages on the left and determine does this seem to match with an existing intent. Either the intent we added (to make a purchase) or a default intent, such as greetings or the help intent (which we have set to the fallback template key).

To do this see our example below.

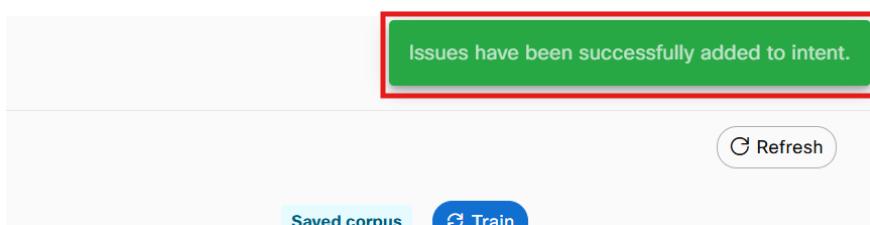
This message, “Can I make an order with you?” looks like it should match our Purchase product intent so we’ll click Link to an existing intent.



Now we'll choose the intent Purchase product to link this too and save.



You'll see a success message pop up.

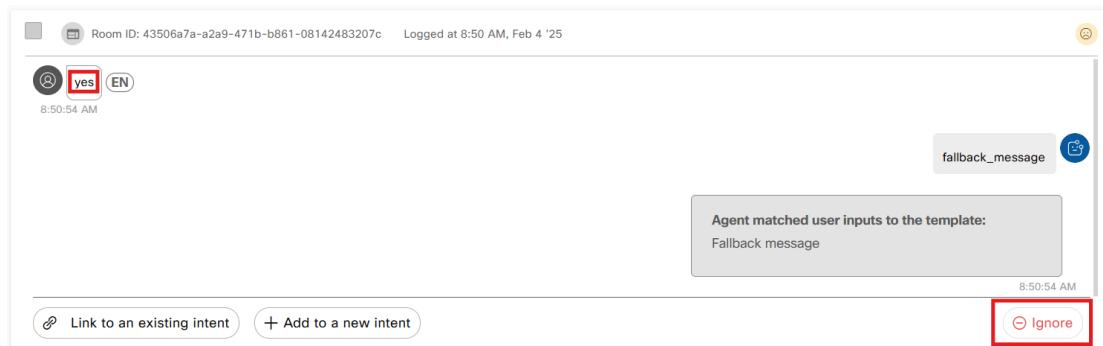


We'll also now be able to see the linked message in the resolved tab, let's first ignore an issue and then head to Resolved.

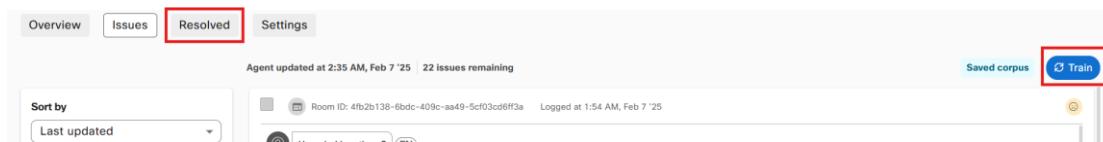
Note: If we noticed that an order contained an entity (we have products listed as entities in our example), than we can tag them here, however it is also worth noting that our demos are not configured to handle this, but can be incorporated into your own future demo designs and is a great highlight of how we can use AI to help steer our customer experience journeys.

Ignore

Find an article you think doesn't need matching and let's click to ignore it. See our example below, containing the message, "yes".



You will receive a different success message pop up. Now that we have started to resolve a few issues we click Train, add any comment (these are handy for colleagues working in teams or reflective work) and then head to Resolved to see the issues we've solved so far.



Here we can see the resolved issues, including metrics, a filtered view and details of what action has been taken for each message.

The screenshot shows a user interface for managing resolved issues. At the top, there are four metrics: 'Total resolved issues 3' (resolved today 3), 'Intents updated 2', 'Intents added 0', and 'Ignored queries 1'. Below these are sections for 'Sort by' (set to 'Last updated') and 'Filter by' (with a red box around 'Hide ignored queries'). The main area displays two messages. The first message, from Room ID 43506a7a-a2x9-471b-b861-08142483207c, was logged at 8:50 AM on Feb 4, 2025. It shows a user input 'yes' and an AI response 'fallback_message'. A note says 'Agent matched user inputs to the template: Fallback message'. The second message, from Room ID 40e20072-700c-4f63-be24-c84118b5fe6c, was logged at 2:27 AM on Feb 6, 2025. It shows a user input 'Can I order a helmet?' and an AI response 'Can I order a helmet?'. Both messages were resolved at 2:41 AM on Feb 7, 2025, by Seb Huber.

2.6 Addon: Flow enhancements (verification loop and gen AI summaries)

Please note, this is additional content.

This task can be skipped, but you will learn how to implement database connections for receiving more information about the caller. We recommend completing Addon modules once you have reached the end of your ‘standard’ lab if you have leftover time.

For this flow let’s head back to your Service on Webex Connect.

First, we will need to disable the flow you built in 2.4 (PodXX web chat v1). Simply toggle the State so the flow is disabled.

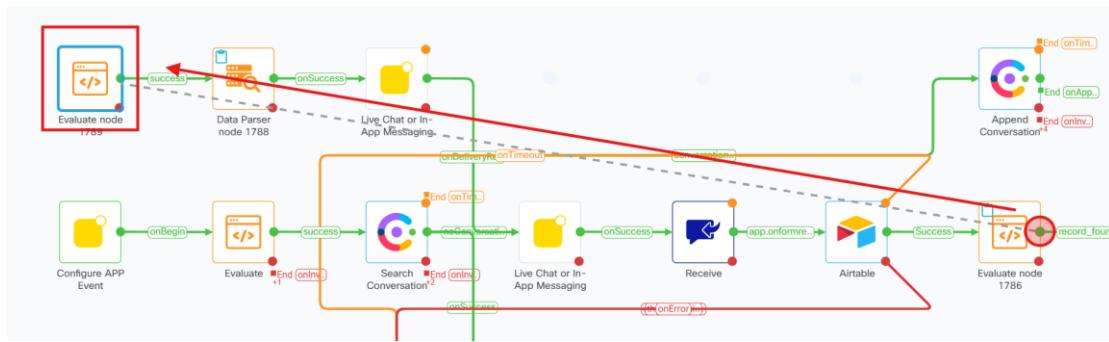


We’re going to Create a new flow using the Upload method again, and this time we will use the 33864_webchat-v2.workflow file we saved from GitHub earlier. Name your flow ‘PodXX Web Chat v2’ (remember to change XX to your pod number), and create your flow.

Zoom out and you will notice that we have added a few nodes extra from our v1 flow. The extra nodes already configured will add CJDS to our flow, so when the chat hands over we will get better insights to the customer’s journey.

We’re going to add email verification for the form when you first initiated the web chat, and we’ll add an AI generated summary of the chat for WxCC.

Zoom in again and focus on the first part of the flow. The first thing we will do is connect the no_record path for Evaluate node 1786 to Evaluate node 1789.



This will loop round and ask the customer to try again if we have no found records for the email address configured (to test this we will run the demo again, replying to the form with an email address we haven't added to the database).

Now let's open Data Parser node 1788 to the right of Evaluate node 1789, the node we just connected to. You will see that a sample JSON payload has already been saved and parsed.

In the SELECT OUTPUT VARIABLES section search for and select the Initiated_from_URL variable.

Then click Import.

PROVIDE SAMPLE INPUT

```

1  {
2    "hasprechatform": "0",
3    "Webpage": "https://attachuk.imi.chat/widget/widgetload",
4    "initiatedon": "",
5    "useragent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)",
6    "Website": "puzzling-broken-carpet.glitch.me",
7    "browserfingerprint": "a3e9695c-75f3-487d-8fe4-d5d34198c1d",
8    "customprofileparams": "ADAOjBbUvj0zbBmzh0QuvcgBdpZbYvEc",
9    "website_id": "933",
10   "Initiated_from_URL": "https://puzzling-broken-carpet.g
11   "proactive_id": 0,
12   "Browser language": "en-GB"
13 }

```

Parsed data output

SELECT OUTPUT VARIABLES

Search

\$._useragent
Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36

\$.Website
puzzling-broken-carpet.glitch.me

\$.Initiated_from_URL
https://puzzling-broken-carpet.glitch.me/

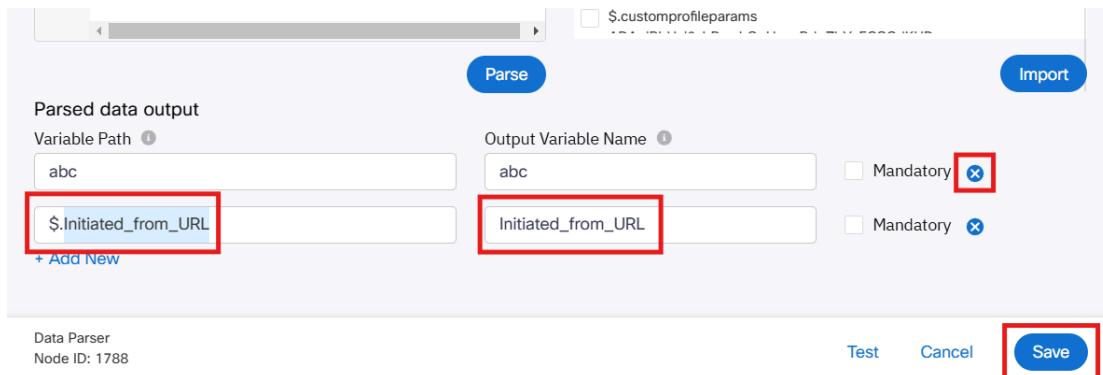
\$.browserfingerprint
a3e9695c-75f3-487d-8fe4-d5d341980c1d

\$.customprofileparams

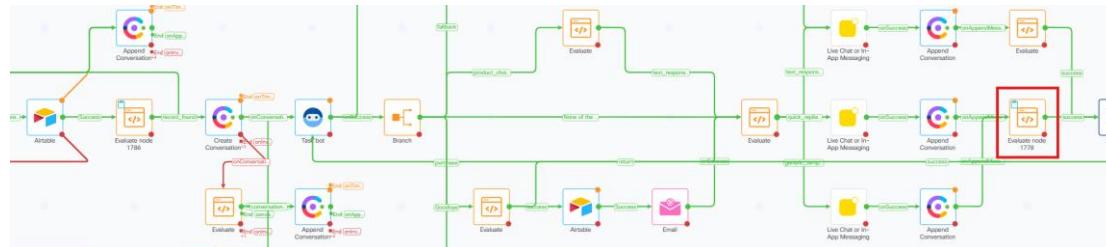
Parse

Import

This will populate a Variable Path below, that we just need to provide an Output Variable Name for, for ease, double click on the highlighted part of the Variable Path (see screenshot below), copy and paste that into the Output Variable Name field. Finally let's delete the 'abc' variable above click Save to exit the node.



That's the email input verification loop done, now let's head to Evaluate node 1778 where we will build a transcript that we can later summarize.

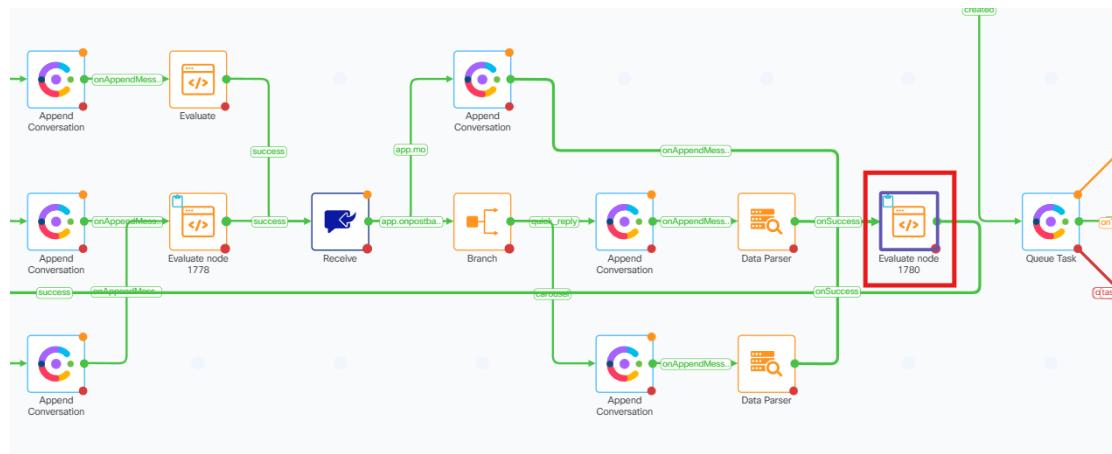


Copy and paste this line of JavaScript before the '1':

```
chat = chat +"Agent: " + text + " ";
```

```
1 chat = chat +"Agent: " + text + " ";
2 1;
```

Now let's save our node. This provides our transcript with context from the bot. Now open Evaluate node 1780.

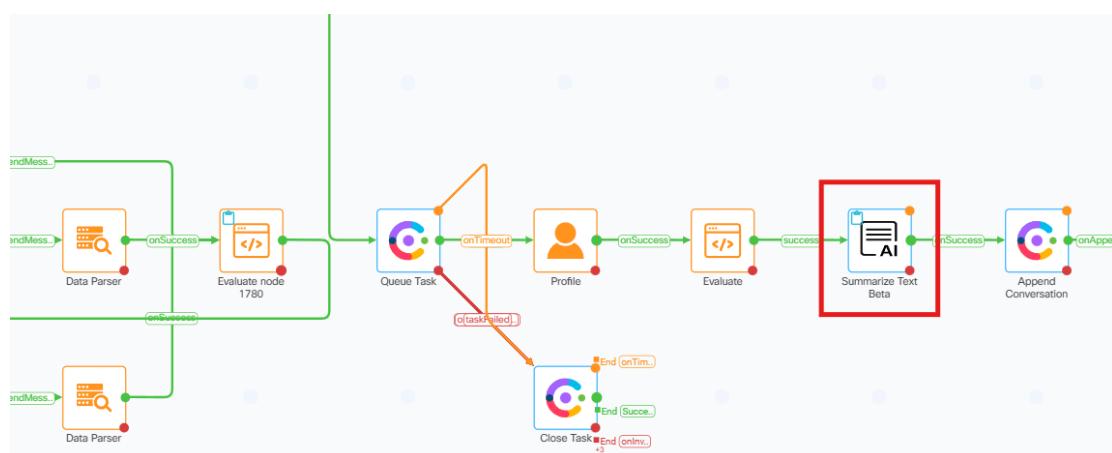


Copy and paste this line of JavaScript before the '1;':

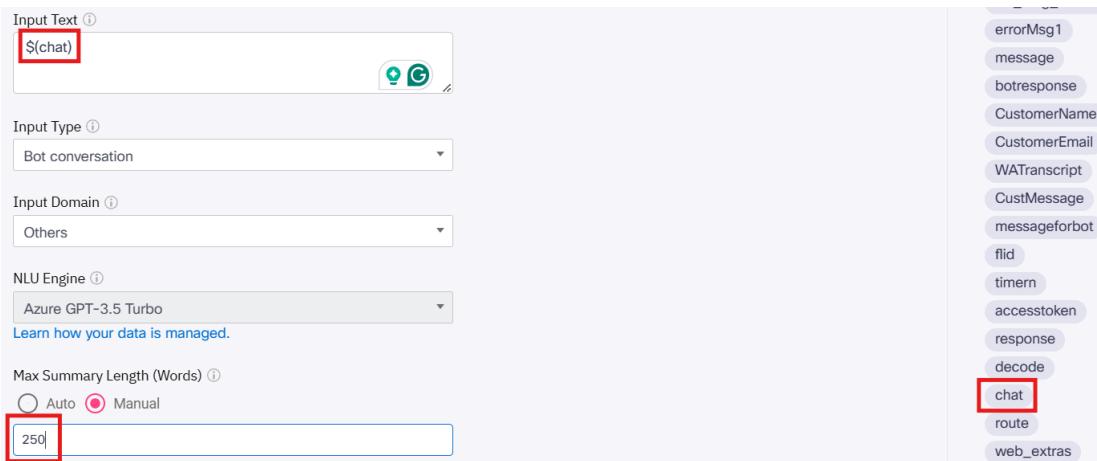
```
chat = chat +" Customer: " +messageforbot + " ";
```

```
1 chat = chat +" Customer: " +messageforbot + " ";
2 1;
```

Now let's save our node. This provides our transcript with context from the user (the customer), meaning we now have context from both the bot and user, time to summarize. Locate and open the Summarize Text Beta node.

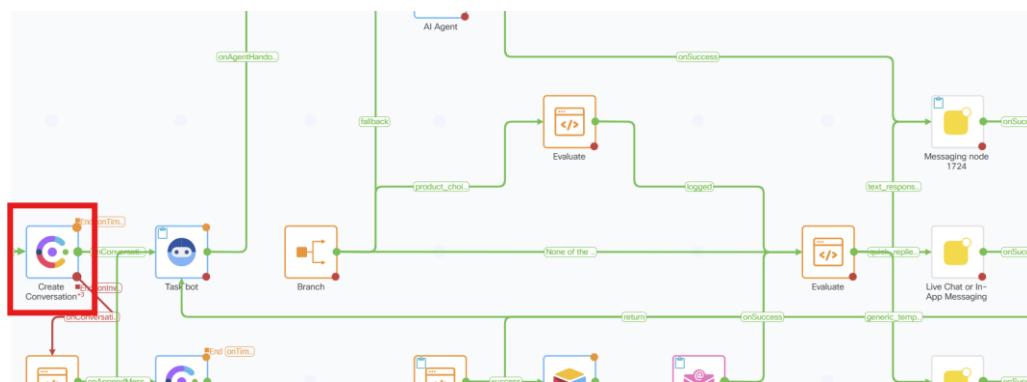


Change 'abc' in the Input Text field to the chat variable from Custom Variables. Change the Max Summary Length field to 250. This will set a limit to how long the summary can be. Click Save.



Now your transcript is also ready.

Like we did in Module 2.3 we need to update the LIVECHAT Website Domain so that we can escalate our conversations to an agent. Zoom out, locate and open the Create Conversation node (see screenshot below for reference).



For this node we need to update the LIVECHAT Website Domain to the URL for our Glitch web page again, removing the https:// at the start.

The screenshot shows the 'Conversation Details' section of a software interface. It includes fields for 'CHANNEL' (set to 'Livechat'), 'LIVECHAT APP ID' (\$n2.inappmessaging.appld), and 'LIVECHAT Website Domain' (puzzling-broken-carpet.glitch.me). The 'LIVECHAT Website Domain' field is highlighted with a red rectangular box.

Now click Save to exit the node.

Let's publish the flow, click Make Live and set the asset to the same you used in 2.4. Once the flow is Live you're ready to test your workflow.

Note: You may need to refresh your browser if Live doesn't show after a minute.

Please note, for testing:

Run through your web chat demo using an incorrect email input, a successful test will look like this:

The screenshot shows a web-based chat application. An input field labeled 'Email*' contains the text 'dopBB@xebew-bal.moc', which is highlighted with a red rectangular box. Below the input field, an error message is displayed: 'There are no records for: dopBB@xebew-bal.moc. If there was a typing error, try again or create an account here: <https://puzzling-broken-carpet.glitch.me/account.html>'. This message is also highlighted with a red rectangular box.

Now input your correct email address and after an interaction with the bot, ask for an agent

Module 3 – Digital Channels

For this lab we will configure a WhatsApp flow designed to allow users to rearrange their delivery in a self-serve, automated fashion. This lab will teach you how:

- to use JavaScript value generation as a functional demo asset
- create restricted, sandbox environments for an individual or team
- create engaging, rich messages with convenient tap-functionality
- create dynamic API calls and parse the responses
- use custom integrations, simplified for workflow usage
- create reporting outcomes for varying success paths
- use rich WhatsApp templates

In the Addon tasks we will enhance this customer journey further by adding the option to escalate conversations to agents in WxCC with additional, deterministic based insights.

Before we get started with this lab we will need to grab some pre-configured items from the Module 3 folder in GitHub. Head to

<https://github.com/ChrisW0lf/CL-EMEA25-LTRCCT-2813/tree/main/flows/WebexConnect/Module3> and download the following files:

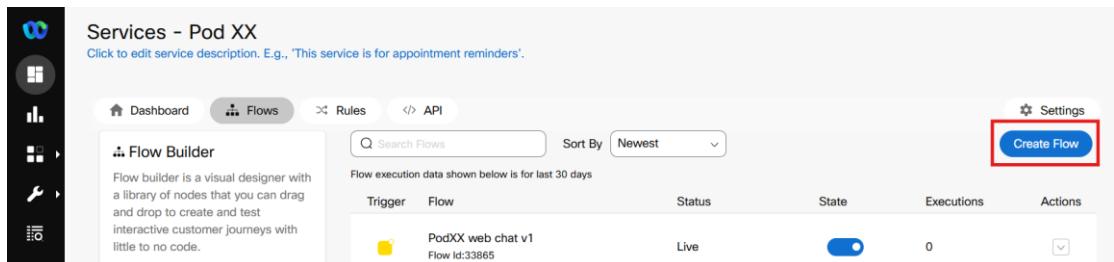
- 33894_whatsapp-v1.workflow
- 33949_whatsapp-v2.workflow
- deliverydate_module3.js

3.1 Webex Connect: WhatsApp Workflow

Head back up to your Service in Webex Connect. You can navigate back to a service using the back icon in the top left of the UI or follow the steps from the start of Module 2 if you need to log back in (remember to return to the service allocated with your Pod number if needed).



Let's create a new flow, click Create.



Enter your Flow Name, Pod XX WhatsApp v1 (remember to replace the XX with your relevant pod number), choose the Upload a flow Method, then upload your file (drag your file into the box or click Choose File again), and click Create.

Create Flow

Flow Name
Pod XX WhatsApp v1

Method
Upload a flow

Attachment
File
33894_whatsapp-v1.workflow

Cancel Create

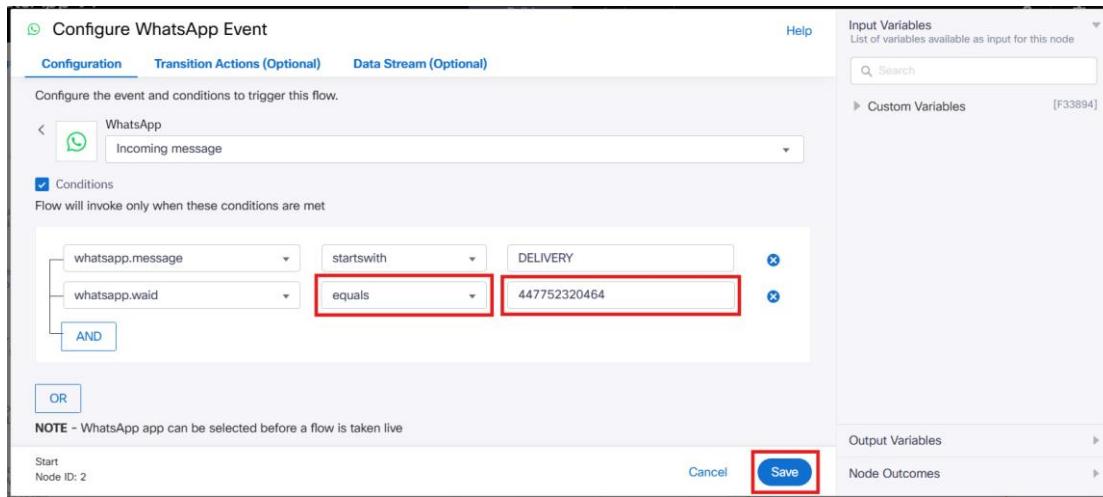
Because this WhatsApp asset is shared between all pods, we are going to add a condition to the Start node. Click AND under Conditions.

The screenshot shows the 'Configure WhatsApp Event' interface. Under the 'Configuration' tab, there is a 'WhatsApp' section with a dropdown set to 'Incoming message'. Below it, a 'Conditions' section is expanded, showing a logic builder with three fields: 'whatsapp.message', 'startswith', and 'DELIVERY'. The 'startswith' field has a dropdown menu open, and the 'AND' operator is highlighted with a red box. A note at the bottom left says 'NOTE - WhatsApp app can be selected before a flow is taken live'. On the right side, there are sections for 'Input Variables', 'Output Variables', and 'Node Outcomes', each with a 'Custom Variables' sub-section. At the bottom, there are 'Start Node ID: 2' and 'Save' buttons.

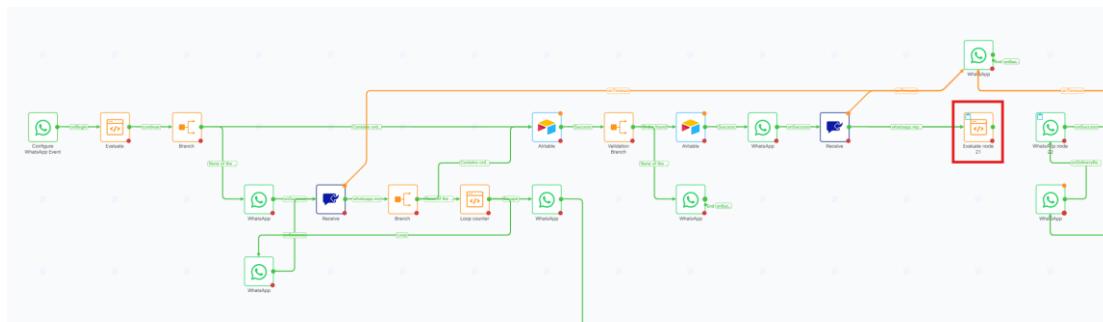
For the Choose Field dropdown select the whatsapp.waid variable

This screenshot shows the same 'Configure WhatsApp Event' interface as above, but with a different focus. The 'Choose Field' dropdown in the conditions section is expanded, showing a list of WhatsApp variables. The 'whatsapp.waid' variable is highlighted with a red box and selected. The rest of the interface elements are visible but less prominent.

Then choose the equals logical operator and enter your mobile number (DO NOT add '+' at start, and make sure there are no gaps). This will ensure that only you can interact with this workflow, and so we don't have multiple of the lab flows all being triggered at the same time. Click Save.



Locate the first node to edit, let's head to and open Evaluate node 21.



We now need to use the JavaScript from GitHub, copy and paste the code from [HERE](#)

```
CL-EMEA25-LTRCCT-2813 / flows / WebexConnect / Module3 / deliverydate_module3.js
```

```
Code Blame 9 lines (9 loc) · 253 Bytes ⌂ Code 55% faster with GitHub Copilot
```

```
1 var dt = new Date();
2 dt.setHours(dt.getHours() + 2);
```

Now in the Evaluate node, paste this code in above the '1,' and save.

Evaluate node 21

Configuration Transition Actions (Optional)

Describe the code you wish to write in English and click "Generate" to keep doing this multiple times by iterating on the prompt. Check get the most useful output.

Generate Code using AI 

Enter JavaScript to perform logic and return values as output. C

```

1 var d1 = new Date();
2 d1.setDate(d1.getDate() + 4);
3 var date1 = d1.toDateString();
4 var d2 = new Date();
5 d2.setDate(d2.getDate() + 5);
6 var date2 = d2.toDateString();
7 var d3 = new Date();
8 d3.setDate(d3.getDate() + 6);
9 var date3 = d3.toDateString();
10
11 1;

```

This code creates 3 days for us to use in the options selections (dates A, B & C), which you will see later in the module.

Connect the success path of the Evaluate node to WhatsApp node 22, (to the right) and then open that node.

Click Configure List items.

WhatsApp node 22

Help

Configuration Transition Actions (Optional) Data Stream (Optional)

Message Configuration

HEADER TEXT (Optional) 

Change your Delivery

MESSAGE BODY 

Please choose from a date below by selecting A, B or C
from the Options.
If you need more options, choose MORE.
or HELP if you want to chat with an agent.

FOOTER TEXT (Optional) 

Use the 'Options' button below to choose an option

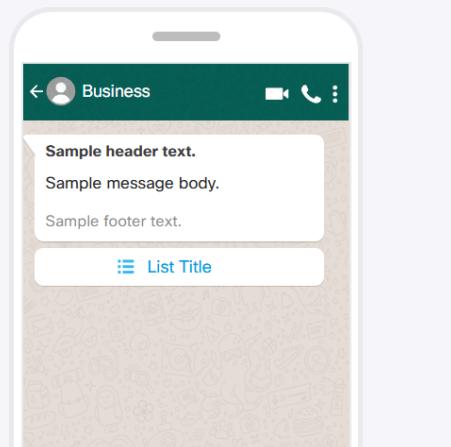
LIST TITLE 

Options

[Configure List items !\[\]\(ff733065bcc331f0ae0d842619e85c21_img.jpg\)](#)

You have successfully added the list items. click above to update the list.

Message Preview



Click the right chevron icon.

Configure List Items

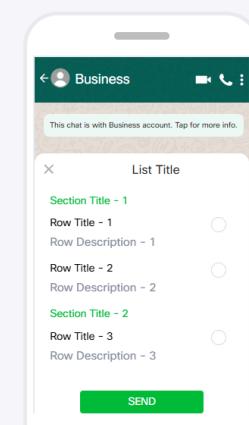
A list message can include maximum of 10 rows counted under all sections, with each section having atleast one row.

List Section

Section Title: Dates

Row Title: 1. A - \$(date1)	Identifier: a
Description: <input type="text"/>	
Row Title: 2. B - \$(date2)	Identifier: b
Description: <input type="text"/>	
Row Title:	Identifier: <input type="text"/>

Message Preview



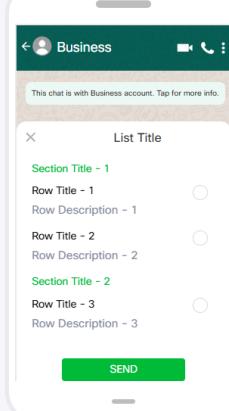
Click + Add Row, and we'll add a HELP option to our list.

Row Title: Help
Identifier: help

List Section

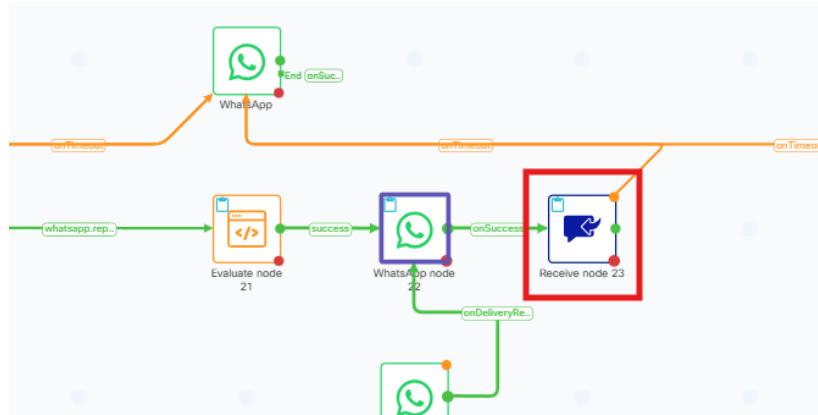
Section Title: Other options

Row Title: 1. MORE	Identifier: more
Description: <input type="text"/>	
Row Title: 2. HELP	Identifier: help
Description: <input type="text"/>	
+ Add Row	



Now click Save and Save again.

Let's configure the Receive node 23 node, locate and open the node.



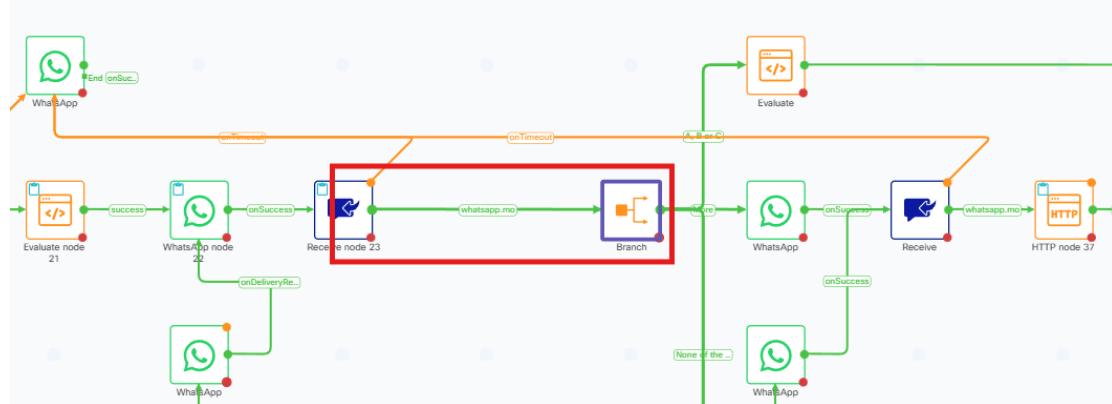
We already have an event listener for an Incoming message, but now let's add a listener for a list message response. Click + Add Another WhatsApp Event.

From the Select Event Type dropdown choose the List Message event listener.

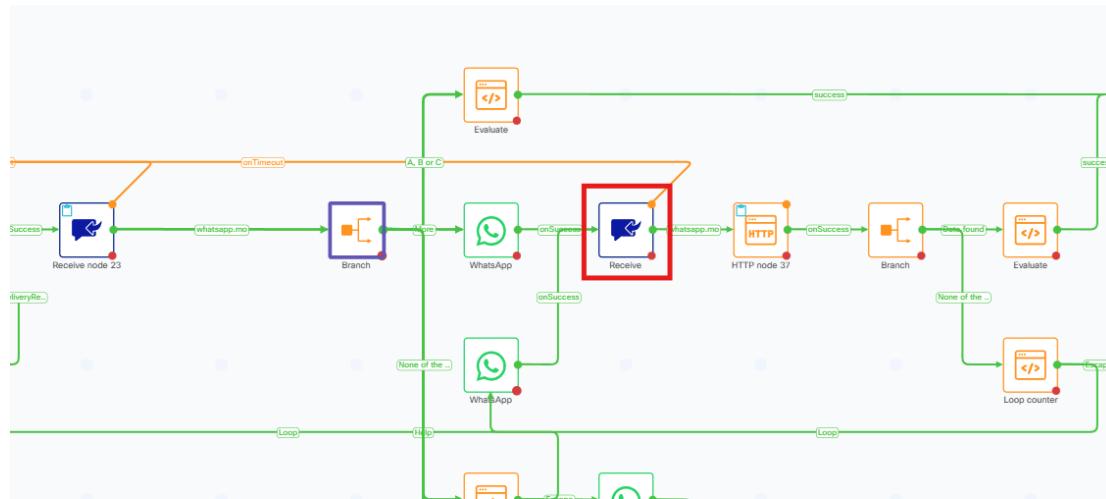
For your reference in Transition actions we are unifying either type of response into a single custom variable we can use named apptChoice in the On-leave transition action (see screenshot below for reference).

The screenshot shows the configuration interface for a 'Receive node 23'. The 'Transition Actions (Optional)' tab is selected. Under 'Action 1', the 'Time' dropdown is set to 'On-leave'. The 'Action' dropdown is set to 'Set variable'. The 'Variable' dropdown is set to 'apptChoice', and the 'Value' dropdown contains the expression '\$(n23.receive.message) \$(n23.whatsapp.li)'. The 'Input Variables' and 'Output Variables' sections on the right show various variables related to the receive operation.

Save the node, then connect the TWO success paths to the branch node to the right.



The first half of the flow is now complete, with validation and date options, now we'll add some Natural Language Processing (NLP) to the flow. Locate and open Receive node 36.



This node expects an open text-based response and we will use this message as a variable to call upon an NLP function in our API call (we didn't have this tenant enabled with NLP functions, but we will make use of an existing NLP environment). In Transition Actions we can see our variable `dateMessage` is set. There's nothing to change, this is for reference only, now let's click Cancel and head to the next Node.

Receive node 36

Help

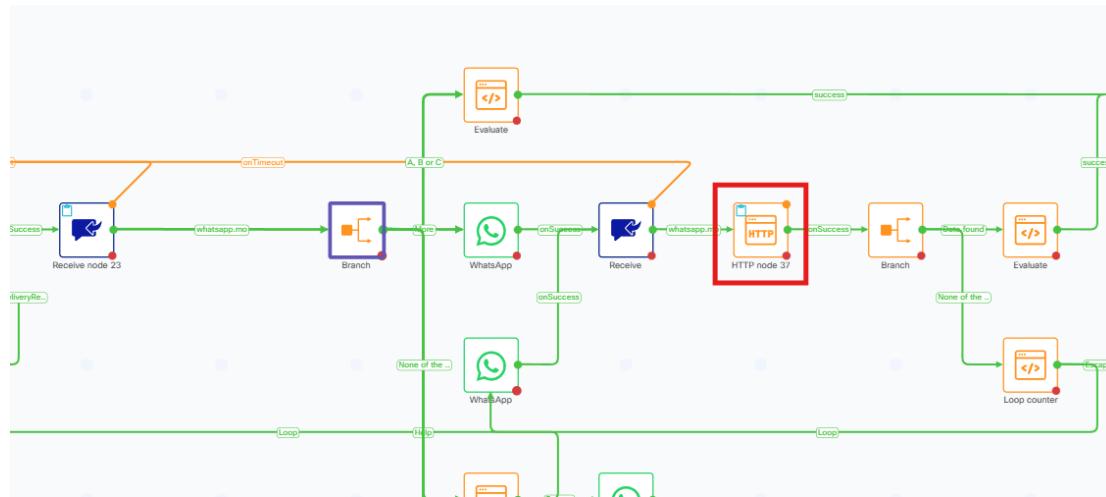
Configuration Transition Actions (Optional) Data Stream (Optional)

Configure node on-enter / on-leave operations

Action 1

Time	Action
On-leave	Set variable
Variable	Value
<code>dateMessage</code>	<code>\$(n36.receive.message)</code>

Open HTTP node 37, the next node along.



This is a flexible, open-ended node allowing for performing requests to external API services. This particular API call was an early version NLP function we developed to understand date and time. First let's make the Body field a bit bigger so we can see the complete JSON body. Then let's replace the 'ABC' value for 'msg' to our dateMessage variable from the Custom Variables.

HTTP node 37

Help

Input Variables
List of variables available at this node

Search

Configuration Transition Actions (Optional)

Method: POST Endpoint URL: https://imibot.ai/api/v1/webhook/intelligent/

Header: bot-access-token Value: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MTAz

Content-Type: application/json

+ Add Another Header

Body:

```
{  
  "msg": "$dateMessage"  
  "msg_type": "text"  
}
```

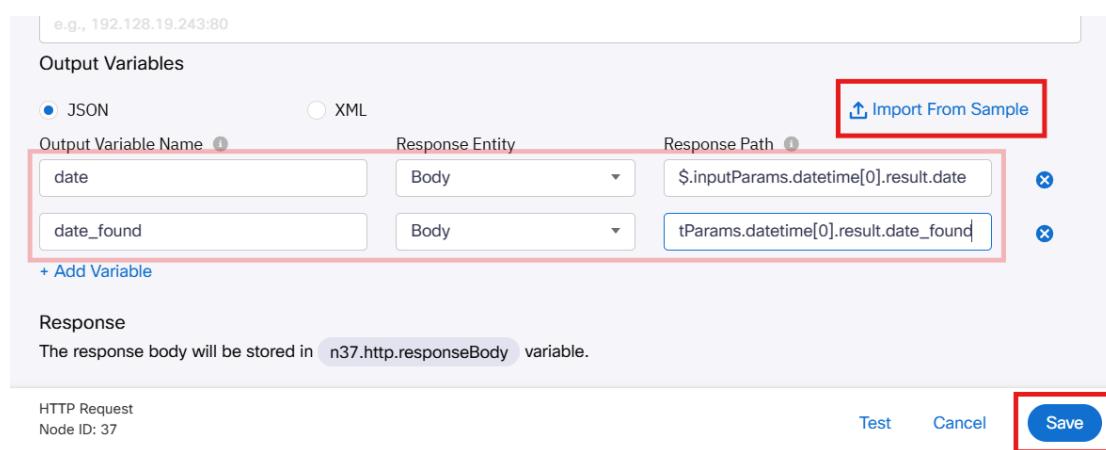
Connection Timeout: 1 ms Request Timeout: 1 ms

Custom Variables

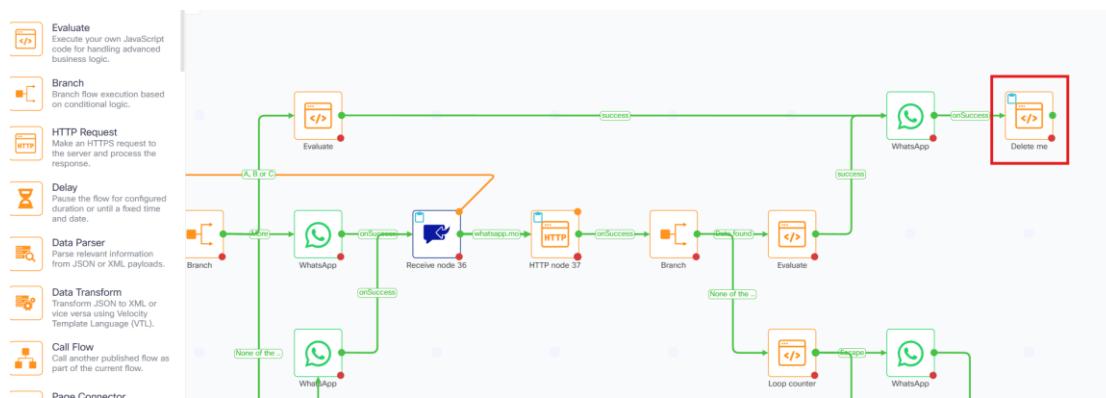
- message
- orderId
- status
- deliveryDate
- customerName
- wa_id
- date1
- date2
- date3
- apptChoice
- dayChoice
- airtable_id
- dateMessage
- nlpDate

Below, you'll see some Output Variables are configured. For your understanding, let's see how we got the two variables. Click on Import From Sample.

Here we see a sample JSON body response load for this API call, we parsed the relevant variables and then gave them a local flow variable name (date & date_found). Go ahead and close the sample view, then click Save to exit the node.

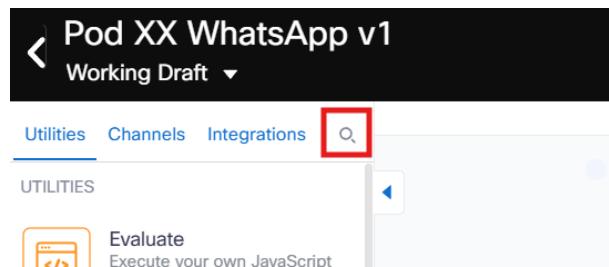


Let's locate the evaluate node named Delete me.

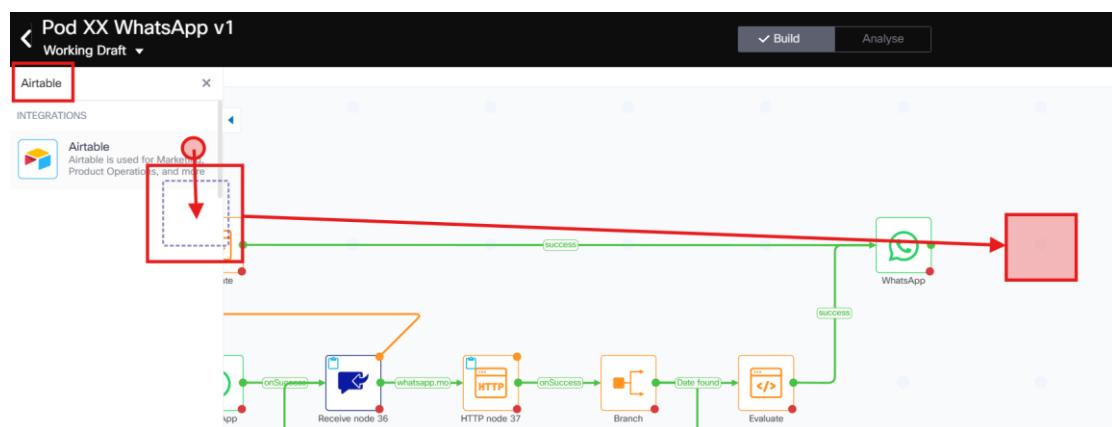


Click on the node ONCE – we just want to select the node, now hit Delete on your keyboard or click the trash bin icon at the bottom of your UI.

Let's replace this node with the Airtable, custom node. Click on the magnifying glass icon at the top of the left-side node menu.



Now type 'Airtable', click and hold on the Airtable node, dragging the node to where we just deleted the evaluate node, on the flow palette.



Connect the success path from the WhatsApp node to the left, to our new Airtable node, then let's open the node.



Click Select Method Name and from the dropdown select the Order: Update delivery date method. For your reference, the other methods listed are all API calls we have integrated to our custom node for reuse across this demo. Now let's fill the fields needed to update our delivery date in the database (both are found in our Custom Variables):

Id: airtable_id
 DeliveryDate: dayChoice

Airtable Configuration

Method Name: Order: Update delivery date

Id: \$(airtable_id)

DeliveryDate: \$(dayChoice)

Input Variables

- message
- orderId
- status
- deliveryDate
- customerName
- wa_id
- date1
- date2
- date3
- apptChoice
- dayChoice**
- airtable_id**
- dateMessage

Now Save and exit the node.

Before we close the success path of the Airtable node we need to create a Custom Variable, and custom Flow Outcome. Click on the settings cog, top right of the UI, and after you enable Descriptive Logs for 1440 minutes (like we did in Module 2), head to the Custom Variables tab, and scroll down to the bottom of the list until you locate the Add New Variable, go ahead and click on this.

Pod XX WhatsApp v1

Working Draft

Airtable

Flow Settings

Custom Variables

+ Add New Variable

Input Variables

- Custom Variables [F33894]
- Start Node ID: 2
- Evaluate Node ID: 3
- Branch Node ID: 4
- WhatsApp Node ID: 6
- Receive Node ID: 7
- Branch Node ID: 8
- Evaluate Node ID: 9
- WhatsApp Node ID: 10
- WhatsApp Node ID: 11
- Airtable Node ID: 12
- Branch Node ID: 13
- Airtable Node ID: 14
- WhatsApp Node ID: 15
- WhatsApp Node ID: 16

This will add a new blank variable, let's fill the fields:

Variable Name (left field): myName

Default Value (right field): Type your name (first and last)



Now head to the Flow Outcomes tab. Click on the dropdown menu next to the last OUTCOME (STATUS CODE 1007), to expand the view.

Outcome	Status
Custom, 1004, No matching order	Disabled
Custom, 1005, Flow loop escaped	Disabled
Custom, 1006, Interaction timed out	Disabled
Custom, 1007, ABC	Enabled

Notification Settings

METHOD: POST URL: https://sebhuber.requestcatcher.com/

Headers

PARAMETER	VALUE
Content-Type	application/json

This flow outcome has been enabled with an option to notify a 3-rd party reporting tool (our lab webhook – the requestcatcher URL), feel free to open that URL in a new tab. Your lab proctors will also receive a notification when this flow outcome has been reached.

Let's change the DESCRIPTION field from 'ABC' to Success path complete and click Save.

Click, hold and drag the Airtable node's success path into empty space.

Then choose Success as the Node Event from the dropdown, and for Flow Result, let's select the new custom Flow Outcome we just made.

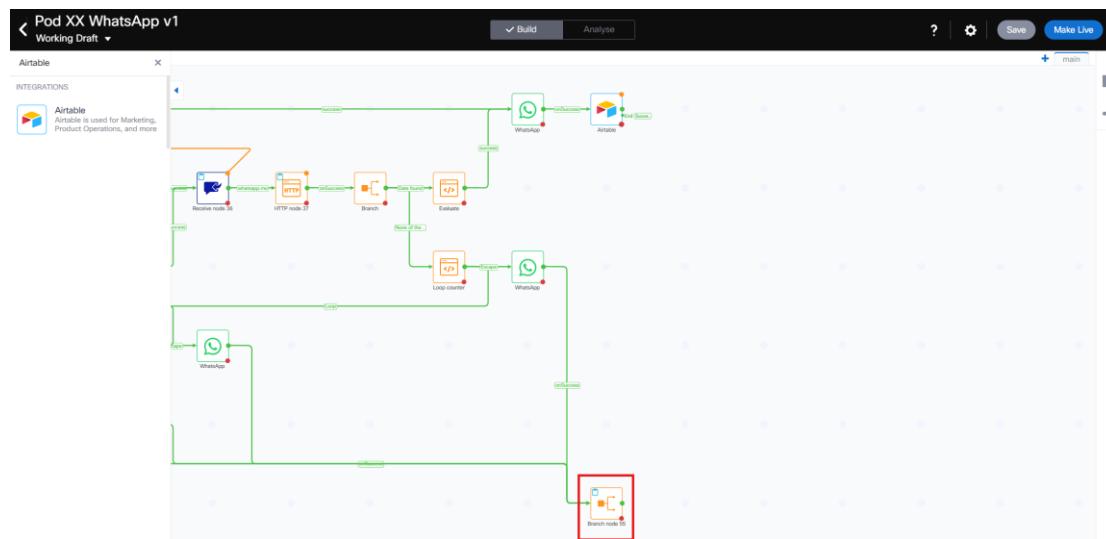
End

Configuration Transition Actions (Optional)

Map node events and flow results to end the flow

Node Event <input type="button" value="Success"/>	Flow Result <input type="button" value="select"/> select 101 - Successfully completed flow [Success] 102 - Flow completed with an error [Error] 103 - Flow ended due to an interruption [Incomplete] 1004 - No matching order [Custom] 1005 - Flow loop escaped [Custom] 1006 - Interaction timed out [Custom] 1007 - Success path complete [Custom]
---	---

Let's configure our final piece of this flow, we want to match the flow outcomes of Branch node 55.



Click and drag the success path for the branch node until our paths are set. Match the following branches with the following outcomes:

id_fail: 1004

Node Event	Flow Result
id_fail	1004 - No matching order [Custom]

unrecognised_response: 1005

Node Event	Flow Result
unrecognised_response	1005 - Flow loop escaped [Custom]

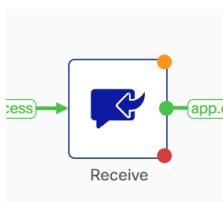
help 1007

Node Event	Flow Result
help	1007 - Success path complete [Custom]

Note: You are now almost ready to publish the flow

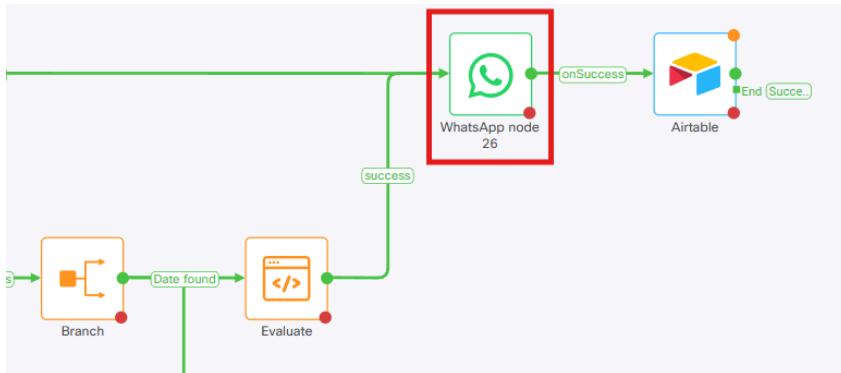
When publishing any flows on Webex Connect that you have imported, **you will need to open and save all receive nodes**, so that the flow recognizes the nodes as configured before we can make the flow live.

To do this locate a Receive node in the flowbuilder UI (see example below)

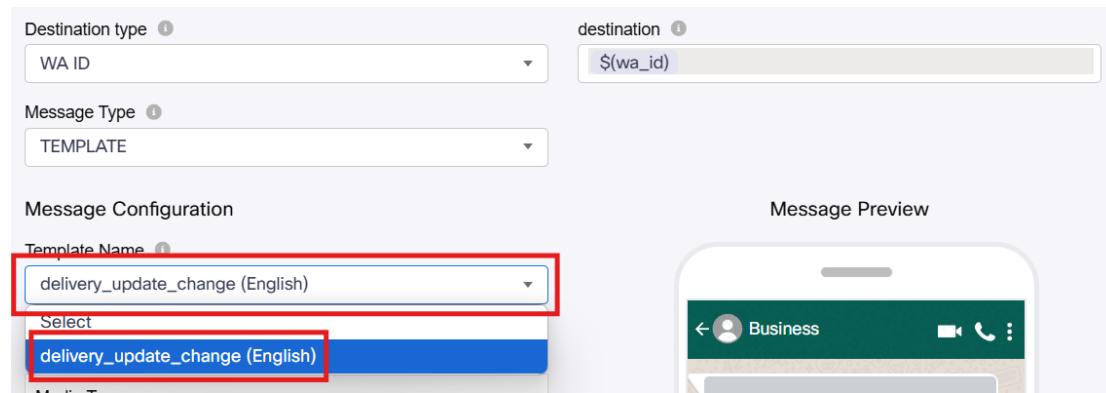


Double click on these to open the node, and then click Save to close the node again.

Because our flow contains a WhatsApp node with a template, **we must also configure this node**. Locate and open the WhatsApp node that connects to our Airtable node at the end of the flow (see screenshot below for reference).



Now select the template listed in the Template Name dropdown.



You will need to scroll down and fill the variables with the following values:

Image: https://awsprodemailassets.s3.amazonaws.com/0ae1dbc4-e5aa-46c3-8aed-55995aabf5b7/calendar_spaced_157009324497803.png

variable1: Select Text as Parameter Type, then enter Value as: \${customerName}

variable2: Select Text as Parameter Type, then enter Value as: \${dayChoice}

Then click Save.

OK, let's publish the flow. Click Make Live in the top right of your UI, then select our Webex CX LTRCCT2813 WhatsApp asset and click Make Live.

Let's recap what you have configured in this module:

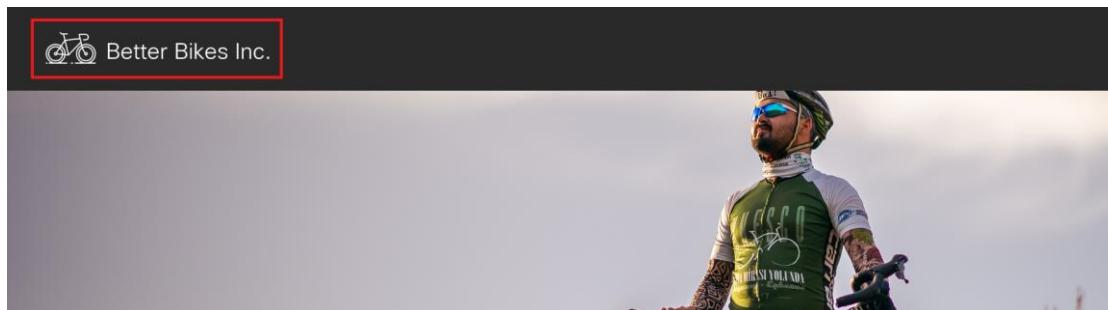
- You created a deterministic-logic based customer journey, enhanced with AI Based NLP
- You used Airtable to verify the messenger, and then updated delivery dates using our custom integrations node for a simplified API call build process.
- You then enhanced the reporting capabilities with custom outcomes and variables.

Once the flow is Live you're ready to test your workflow.

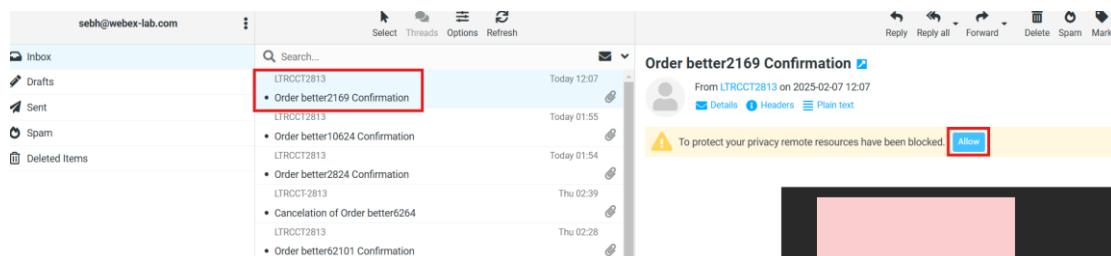
Note: You may need to refresh your browser if Live doesn't show after a minute.

3.2 Testing the WhatsApp demo

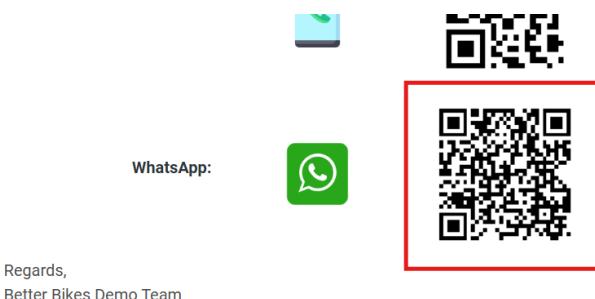
To test the demo, let's head back to the glitch website you created in 0.9 and head back to the home page (click on the logo in the top left corner).



Now let's click Enter, so we have a new order in our email inbox. Head over to your email inbox, select the new order and click Allow to let the images load.

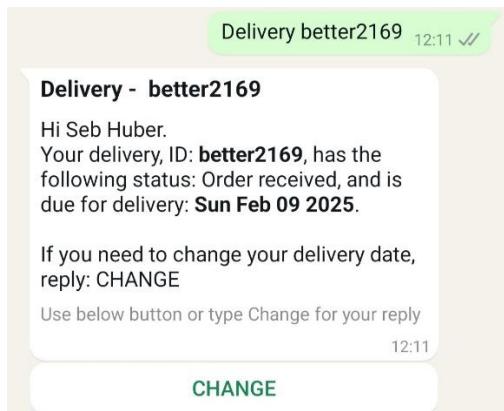


Scroll down in your email so you can see the WhatsApp QR code and scan it with your camera or QR code reader on your mobile phone (If you do not have WhatsApp on your phone or have issues with your camera, please ask for help from a lab proctor).



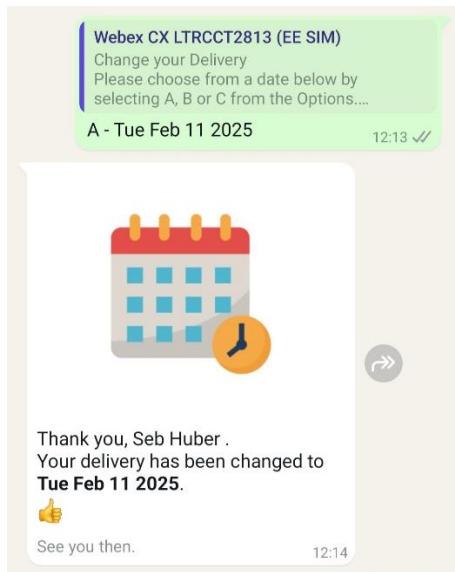
Want to learn more? Refer to our [product guide](#)

Run through the journey by following the prompts in the chat, 'CHANGE'



Then run through the A, B and C, MORE and HELP options.

For the A, B and C, and MORE options, if you run through the instructions in the WhatsApp chat should both end with a Calendar update that looks like the below screenshot.



For the HELP option you will receive a placeholder WhatsApp message that reads:

"No problem. We're transferring you to our help team now."

In the Addon task (3.3) we will elevate this option to WxCC for an agent to pickup.

We will also receive an API call to our webhook listener that will look like the below screenshot for all the different journey options.

request catcher

POST /

2025-02-07T12:17:46+00:00
34.250.50.84

POST /

2025-02-07T12:16:15+00:00
52.17.23.194

```
POST / HTTP/1.1
Host: sebhuber.requestcatcher.com
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-Length: 43
Content-Type: application/json
User-Agent:

Pod user - Seb Huber has completed Module 3
```

Congratulations – you've finished all the standard modules in this lab! 🎉

3.3 Addon: Flow enhancements (Elevate to WxCC)

Please note, this is additional content.

This task can be skipped, but you will learn how to implement database connections for receiving more information about the caller. We recommend completing Addon modules once you have reached the end of your ‘standard’ lab if you have leftover time.

Before we get started on the work in Webex Connect, we need a Queue where the Channel will be assigned. Head back to Control Hub > Contact Center > Queues in the left sidebar.

We are going to use the PodXX incoming Social queue, click on the copy icon at the right of the line.

Queue	Description	Queue type	Channel type	Status	Last modified
Outdial Queue-1	Outdial Queue created by syst...	Outdial queue	Telephony	Active	October 30, 2024 15:32 pm
PodXX Incoming Chat		Inbound queue	Chat	Active	February 07, 2025 10:56 am
PodXX Incoming Social		Inbound queue	Social channel	Active	February 07, 2025 23:28 pm
Queue-1	Queue created by system	Inbound queue	Telephony	Active	October 30, 2024 15:32 pm

Change Name with PodXX to your relevant assigned Pod number and remove the ‘_copy’.

General	Name *	<input type="text" value="PodXX Incoming Social"/>
	Description	<input type="text" value="Type here"/>
	Queue type *	<input type="text" value="Inbound queue"/>

In Contact Routing Settings, under Group details, click on the Edit Pen for the existing Group 1.

Contact Routing Settings

Outbound Campaign ⓘ Outbound Campaign is only applicable for Outdial Queues.
Switch

Queue Routing Type * Longest Available Agent
You can't change Queue Routing Type once the queue is created

Chat Distribution * You can add one or more teams to this Chat through a Call Distribution Group. You can add multiple groups to distribute a call to more teams. Call distribution is independent of other queues login defined in Flows.

Create Group

Group Details

Priority	Group Name	Switching Time	Actions
1	Group 1	NA	

Remove the assignment for PodXX_Team then choose your PodXX_Team and Save.

Group 1

Priority 1

Switch to this group after ⓘ 0 Seconds

Search by name 4 Records

Team	Site	Team type
<input checked="" type="checkbox"/> PodXX_Team	LTRCCT-2813	AGENT
<input type="checkbox"/> Pod20_team	LTRCCT-2813	AGENT
<input type="checkbox"/> Team-1	LTRCCT-2813	CAPACITY

Cancel **Save**

Then click Create.

Now let's head back to your Service on Webex Connect.

First, we will need to disable the flow you built in 3.1 (Pod XX WhatsApp v1). Simply toggle the State so the flow is disabled.



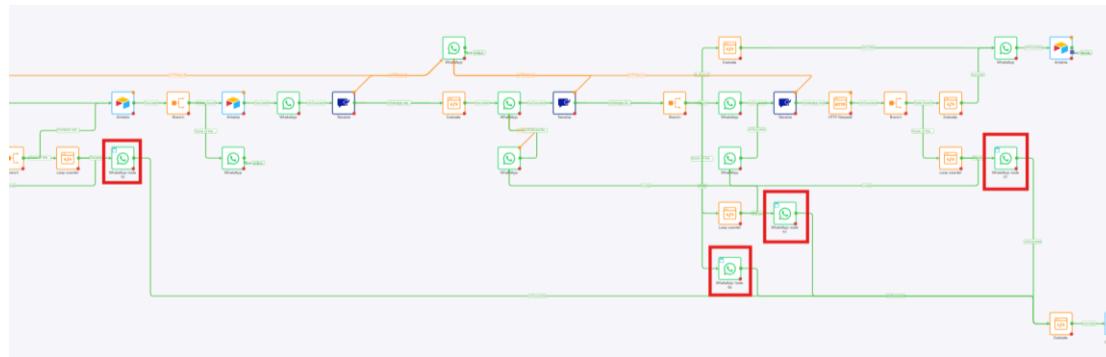
We're going to Create a new flow using the Upload method again, and this time we will use the 33949_whatsapp-v2.workflow file we saved from GitHub earlier.

Name your flow 'Pod XX WhatsApp v2' (remember to change XX to your pod number), and create your flow.

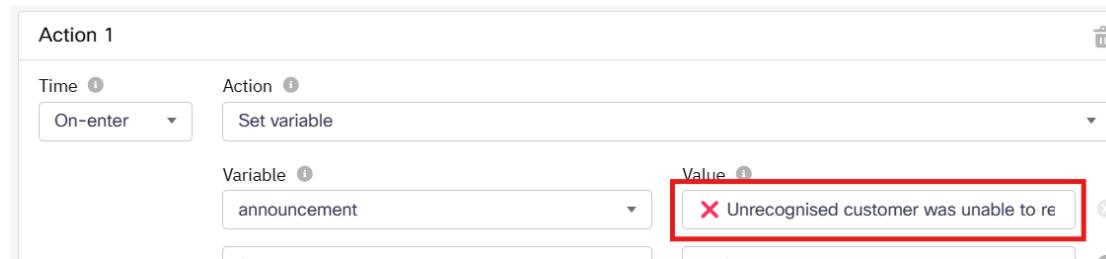
Zoom out and navigate to the end of the new flow. You will notice that we have added a few nodes extra from our v1 flow. The extra nodes already configured will provide us with the ability to elevate our WhatsApp conversations to an agent on WxCCC.

We're going to review the dynamic announcements that different routes provide and append some deterministic information for the agent to understand the context of why the customer has been escalated.

First let's review one of the WhatsApp nodes that lead to our WxCCC escalation route, you can open on any of these (node 10, 47, 48 or 51).



Now head to Transition Actions and you will see the announcement set.

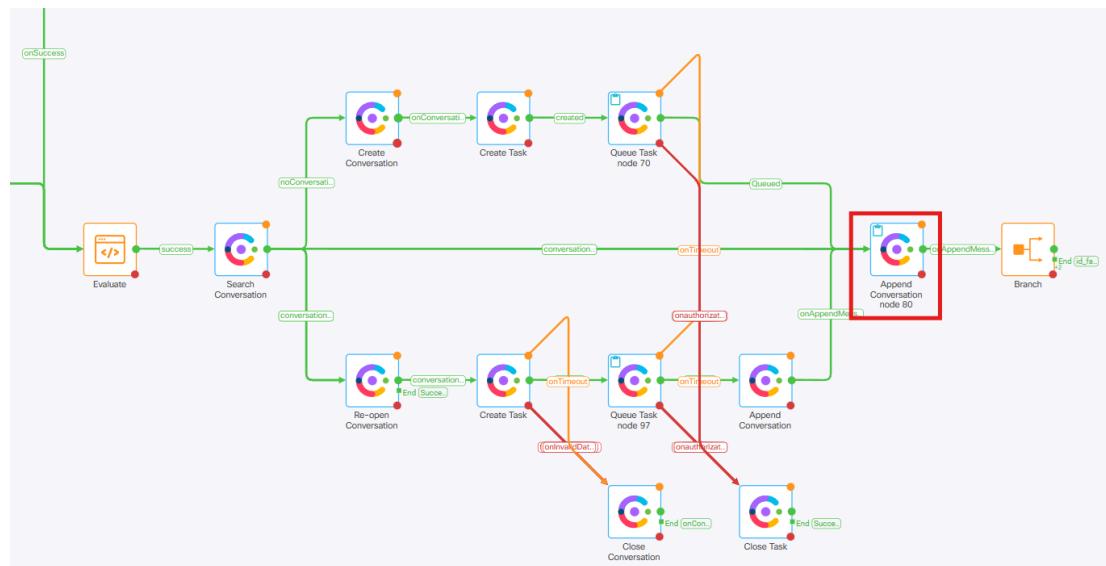


Click on the field, copy the text and paste it here <https://onlinenotepad.org/notepad> so you can get a better view of the text context.

Head back to the flowbuilder, close that node and open one of the others highlighted above and repeat the steps to compare the different announcements. See an example below of contrasting announcements.

X Unrecognised customer was unable to retrieve their order ID.
I Customer: **\$(customerName)** would like some help regarding their order. Current date is \$(deliveryDate) Order ID - \$(orderId)

Now let's head to the escalation part of our flow. Locate and open Append Conversation node 80



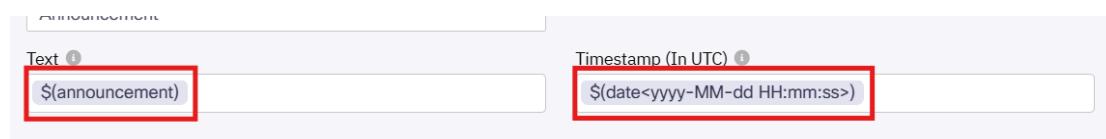
For the Text field we're going to use the announcement variable we just observed from the Transition Actions on the WhatsApp nodes. Delete the 'abc' and replace with announcement from our Custom Variables. You will need to tap your space bar and Backspace keys for this characters to register with the field.

We're now going to use a system timestamp variable. In the Timestamp field delete the 'abc' and paste in this variable:

`$(date<yyyy-MM-dd HH:mm:ss>)`

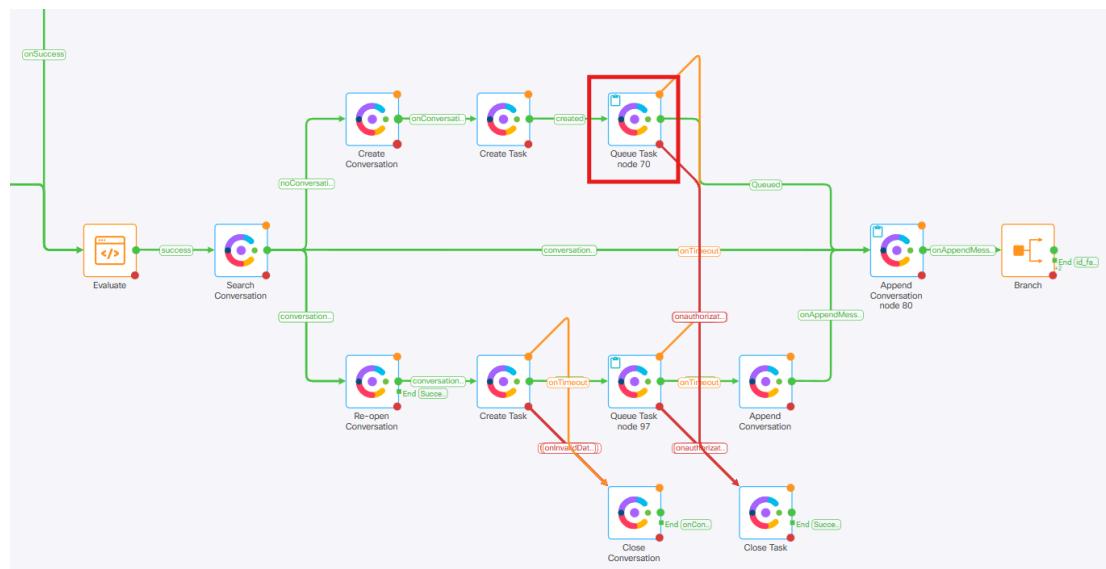
Tap your space and backspace keys ensuring there are no spaces after the variable.

When you click somewhere off the field, the variables should like this if configured properly:

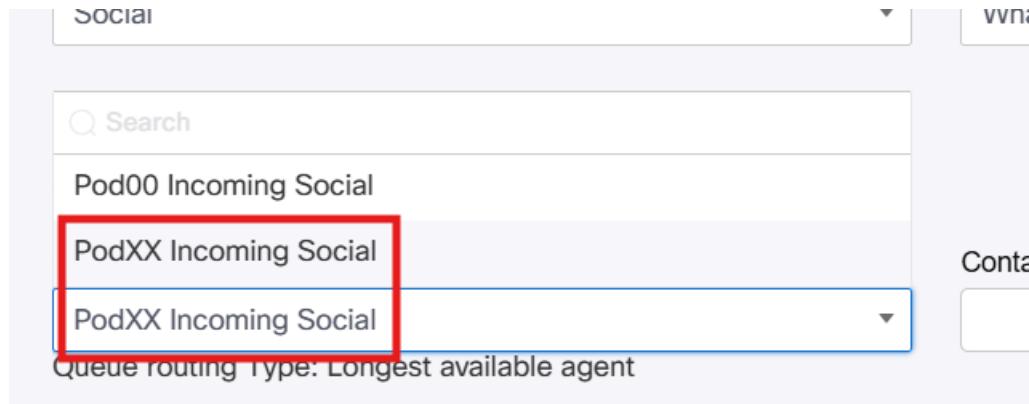


Click Save to exit the node.

Now let's open Queue Task node 70

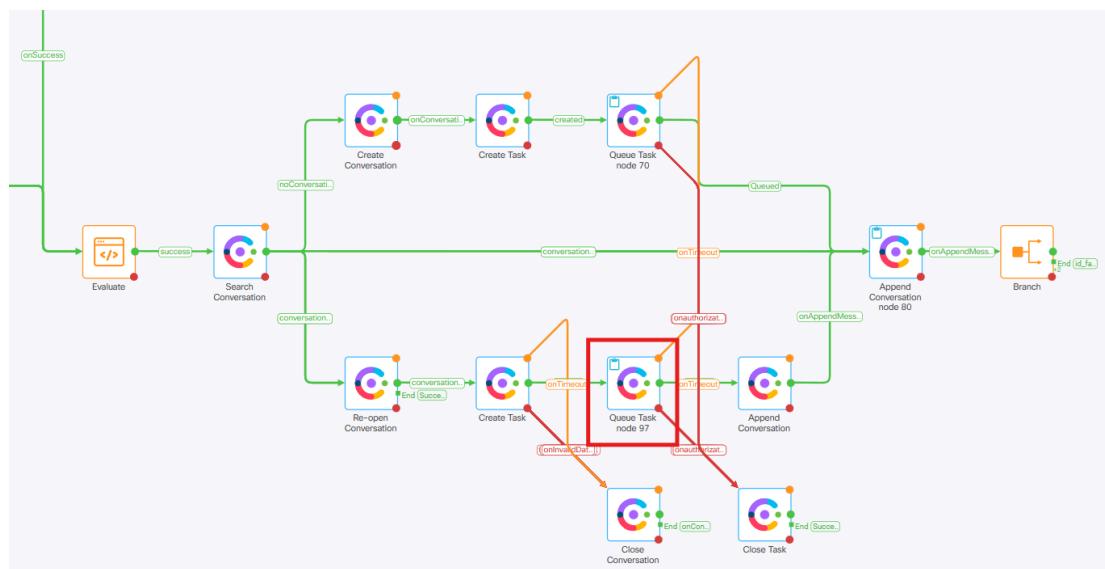


For the Queue Name dropdown we need to select the queue we created at the start of this task, named 'PodXX Incoming Social' (remember you have replaced the XX with your relevant pod number)



And click Save.

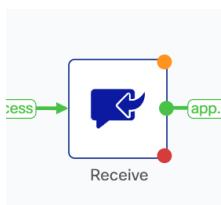
Now let's repeat this process for Queue Task node 97



Note: You are now almost ready to publish the flow

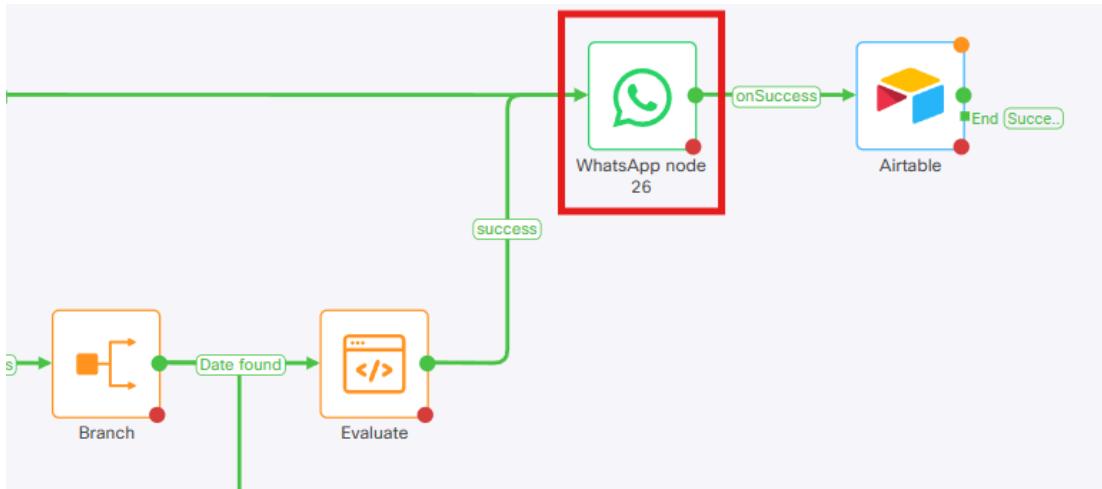
When publishing any flows on Webex Connect that you have imported, **you will need to open and save all receive nodes**, so that the flow recognizes the nodes as configured before we can make the flow live.

To do this locate a Receive node in the flowbuilder UI (see example below)

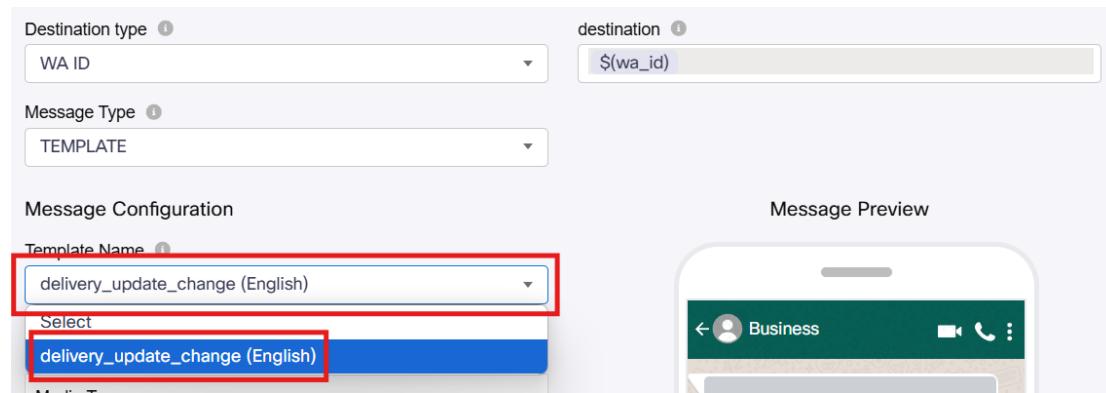


Double click on these to open the node, and then click Save to close the node again.

Because our flow contains a WhatsApp node with a template, **we must also configure this node**. Locate and open the WhatsApp node that connects to our Airtable node at the end of the flow (see screenshot below for reference).



Now select the template listed in the Template Name dropdown.



You will need to scroll down and fill the variables with the following values:

Image: https://awsprodemailassets.s3.amazonaws.com/0ae1dbc4-e5aa-46c3-8aed-55995aabf5b7/calendar_spaced_157009324497803.png

variable1: Select Text as Parameter Type, then enter Value as: \$(customerName)

variable2: Select Text as Parameter Type, then enter Value as: \$(dayChoice)

Then click Save.

OK, let's publish the flow. Click Make Live in the top right of your UI, then select our Webex CX LTRCCT2813 WhatsApp asset and click Make Live.

Once the flow is Live you're ready to test your workflow.

Note: You may need to refresh your browser if Live doesn't show after a minute.

Testing:

To test, run the same steps you completed for testing v1, but look to trigger the HELP route, as this will escalate to WxCC. You can handle chats as you did in the prior Module, 2.5 & 2.7.

You can also try exiting on the loop escapes. There are 3 looping parts of the flow.

1st is the initial message we send to trigger the message. For your Order ID, remove the ‘better’ part of the ID so we trigger the looping experience.

2nd is the main options menu. Try replying with messages the branch logic isn’t anticipating.

3rd is on the MORE branch. The NLP function is anticipating a date, so try responding without any day/date context.

As demonstration builders, you may think to enhance the experience by ‘catching’ the unrecognized intents with AI agents. You could also use AI to categorize only, with no responses being sent to the messenger, the types of unrecognized responses, and then enhance your journey based on data observed using the AI agents.

Congratulations – you’ve finished ALL modules in this lab! 🎉