slido

Join at slido.com
#1740593

ⓘ Start presenting to display the joining instructions on this slide.

From Features to Solutions

Standing on the shoulders of giants

Kevin Simpson
Technical Leader

slido

What is a solution?

ⓘ Start presenting to display the poll results on this slide.

Greater than the sum of their parts!

- Using flow logic to create a specific experience
- Using APIs to extend the functionality of a call flow
  - Reporting
  - Global Variables
  - Business Hours
- Using Webex Connect to extend the functions of a call flow
- Utilizing the whole suite of products to reach your desired outcome
  - Using Webex Calling to route calls to non-contact center team members
  - Adding Journey Data Services
- Connecting external services to enhance the customer experience
  - Data dips
  - Outbound Campaigns
  - Contact Center AI

It’s not Magic!

Defining the Problem

- What are you trying to solve?
  - Proper isolation is the key to success.

- What is the success criteria?
  - How will you know that you were successful?

- What are your obstacles?
  - Obstacles equals actions!

Cherry Picking Calls

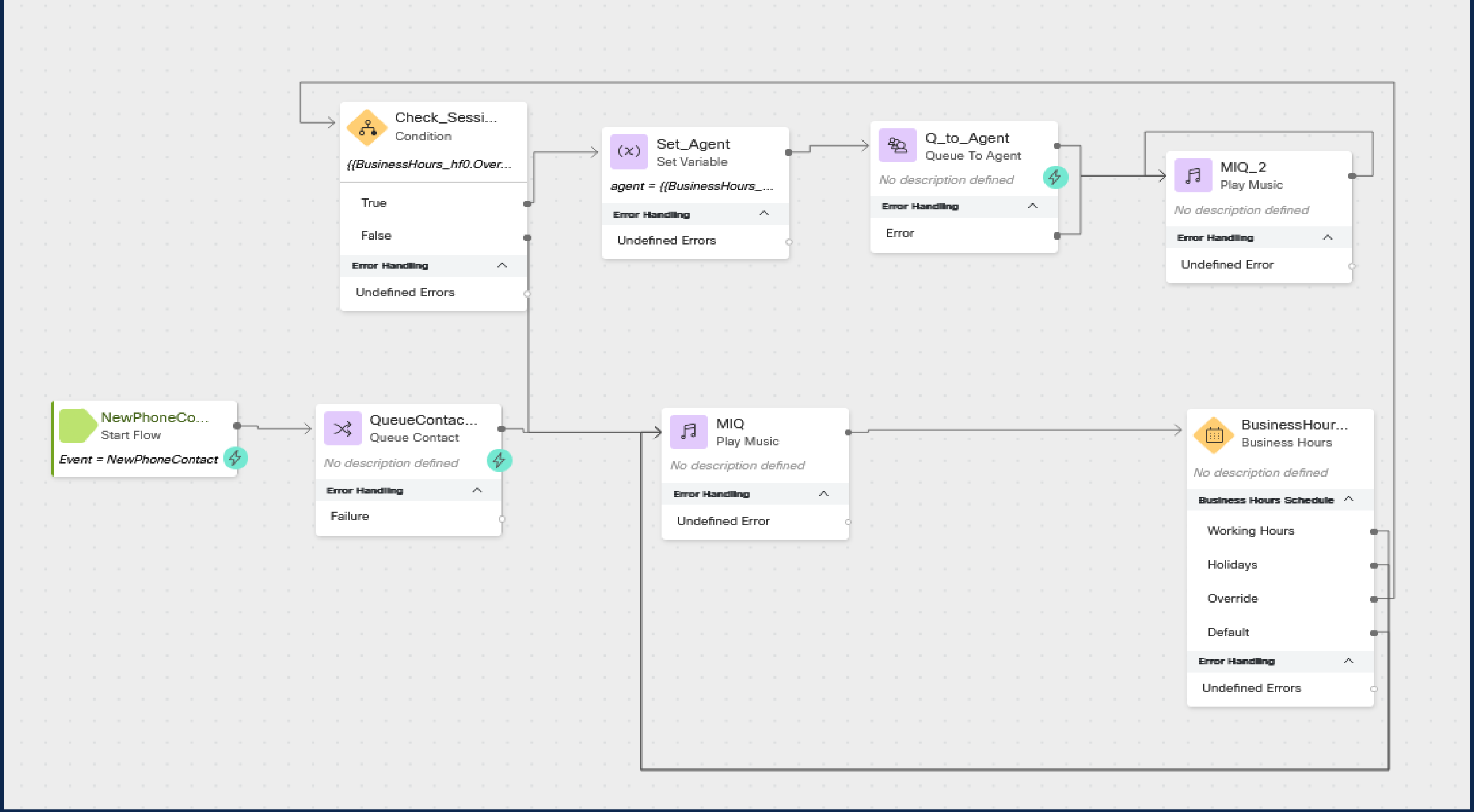
Cherry Picking Calls

- Use Case:
  - To move a call which is in queue to a specific agent
- Features Used
  - Business Hours
  - Queue to agent
  - Analyzer
- APIs used in demo
  - Overrides

Cherry Picking Calls

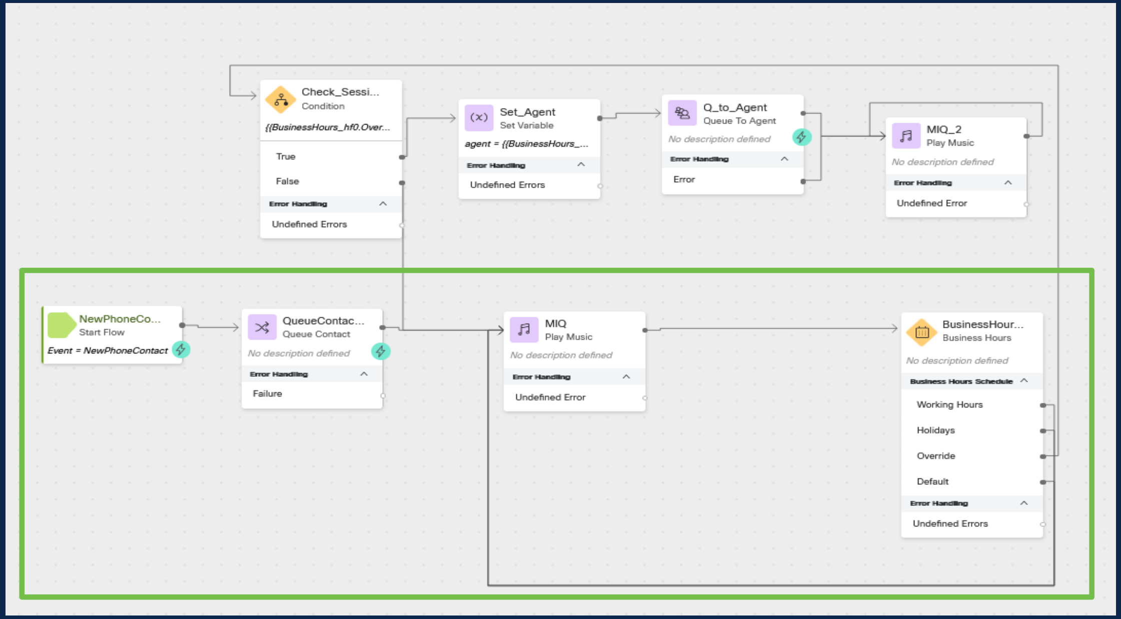
- Configure Business Hours
- Create and link an Override
  - Note the Override ID
- Create an analyzer report
  - Needs to include the SessionID
  - Should include the ANI
- Create the flow

Cherry Picking Calls



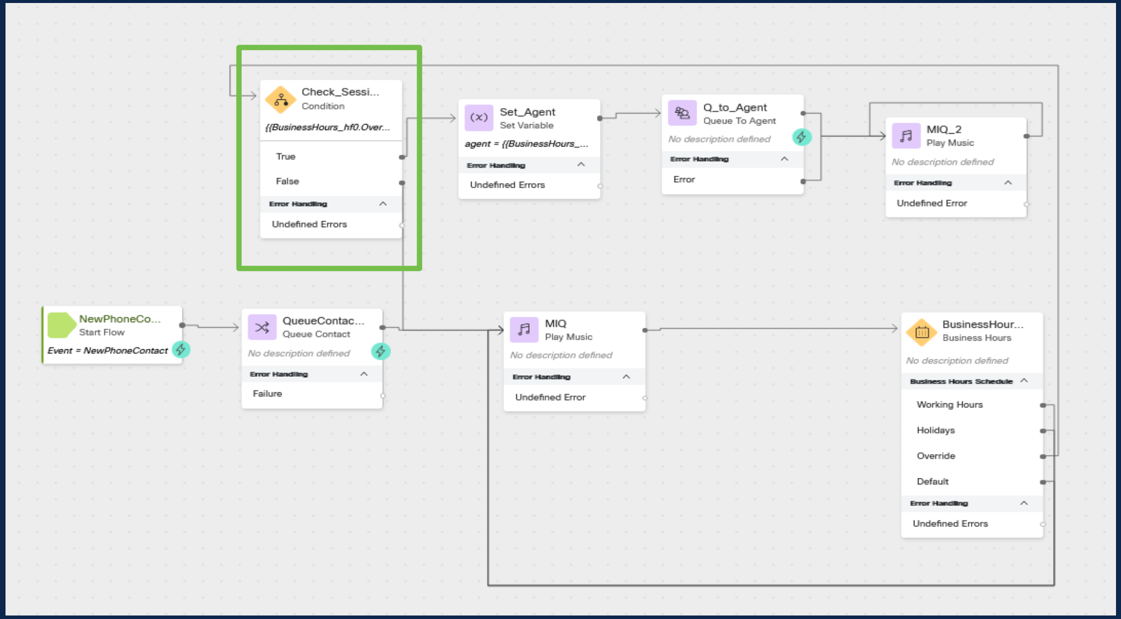
Cherry Picking Calls

- Normal Flow
  - Call gets queued
  - Music in Queue played for caller
- Check Business Hours
  - If an Override is not active loop queue music



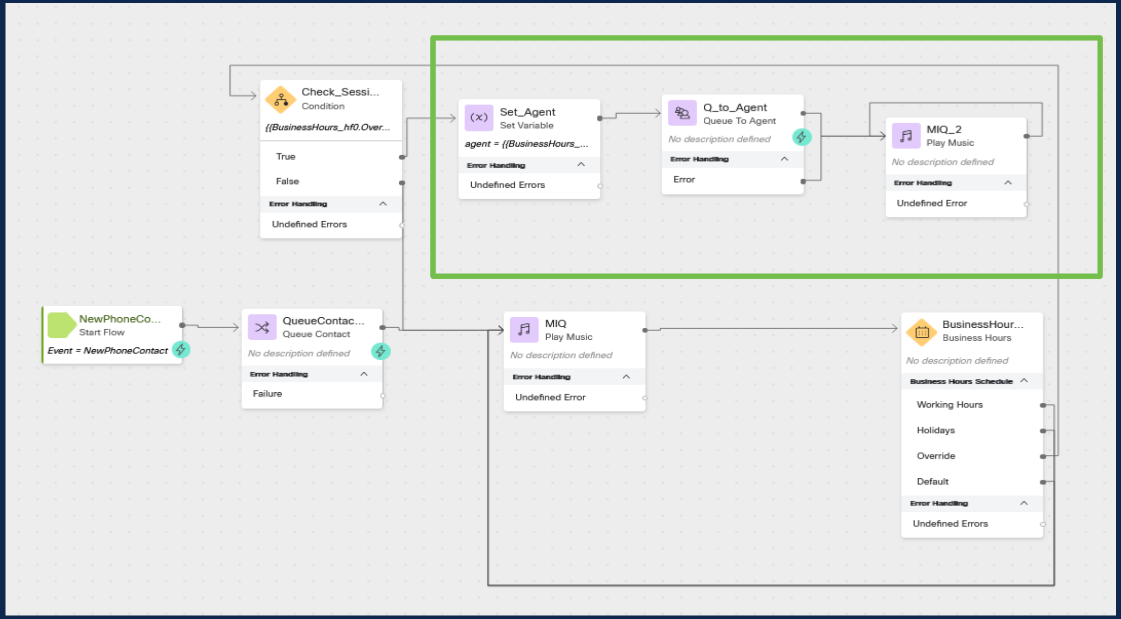
Cherry Picking Calls

- Check Business Hours
  - An Override is active
    - The Override name does not include the current SessionID
      - Loop queue music



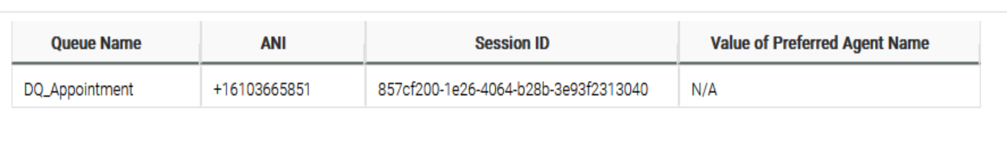
Cherry Picking Calls

- Check Business Hours
  - An Override is active
    - The Override name does include the current SessionID
    - (<SessionID>__<AgentLogin>)
      - Parse the agent information and queue to that agent

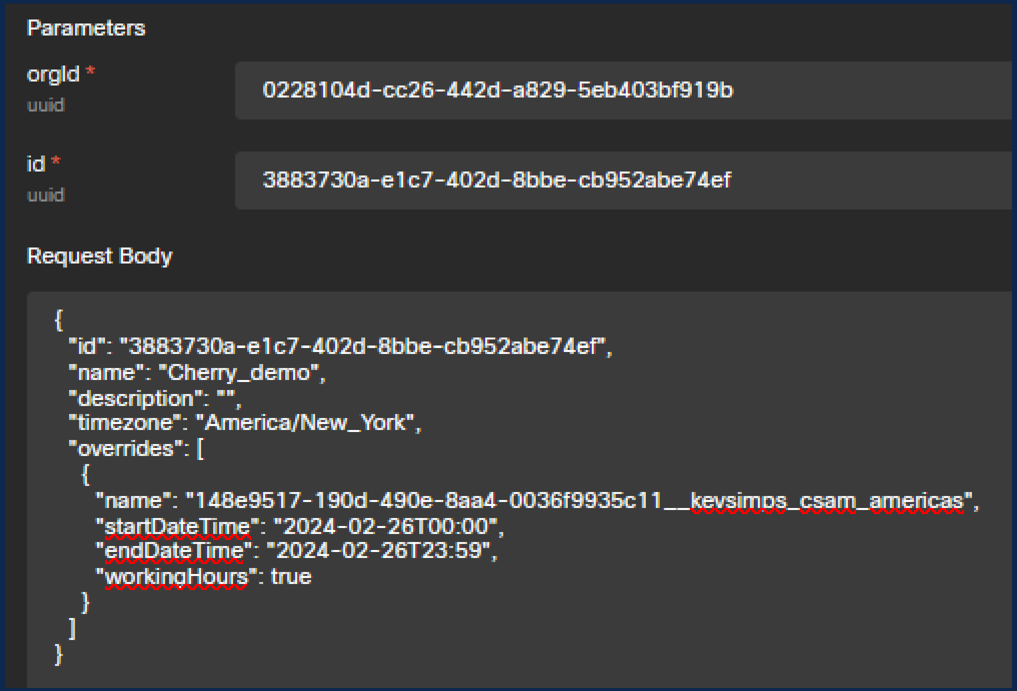


Cherry Picking Calls

- Copy the Session ID from the analyzer report
- Paste the Session ID as the first part of the name field followed by the double underscore delimitator
- Paste the agent login name after the double underscore delimitator
- The call will be queued to the agent the next time the music in queue loops



Queue Name	ANI	Session ID	Value of Preferred Agent Name
DQ_Appointment	+16103665851	857cf200-1e26-4064-b28b-3e93f2313040	N/A



Parameters

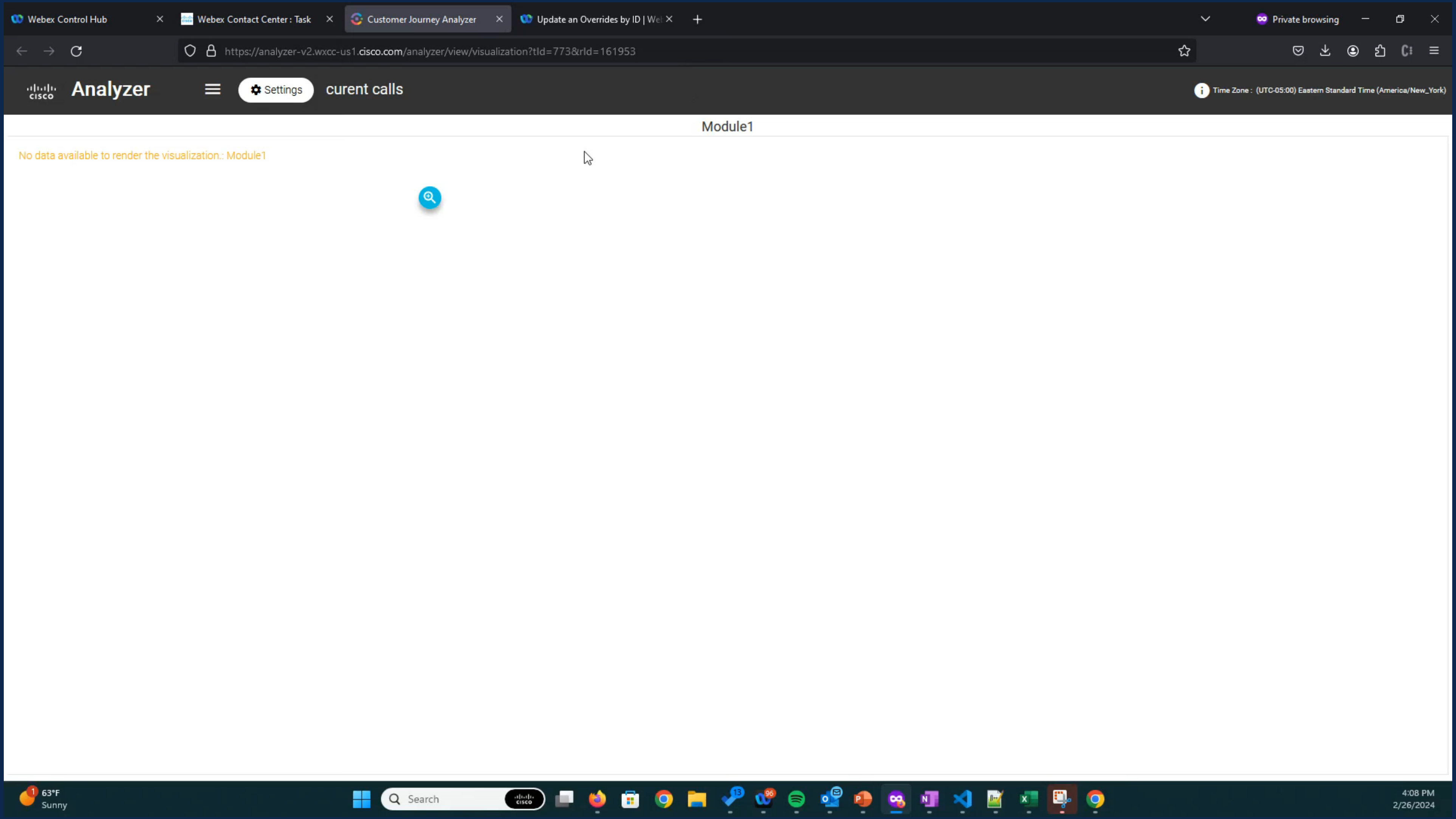
orgId *
uuid
0228104d-cc26-442d-a829-5eb403bf919b

id *
uuid
3883730a-e1c7-402d-8bbe-cb952abe74ef

Request Body

```
{
  "id": "3883730a-e1c7-402d-8bbe-cb952abe74ef",
  "name": "Cherry_demo",
  "description": "",
  "timezone": "America/New_York",
  "overrides": [
    {
      "name": "148e9517-190d-490e-8aa4-0036f9935c11__kevsimps_csam_americas",
      "startDateTime": "2024-02-26T00:00",
      "endDateTime": "2024-02-26T23:59",
      "workingHours": true
    }
  ]
}
```

Cherry picking Calls



Webex Control Hub
Webex Contact Center : Task
Customer Journey Analyzer
Update an Overrides by ID | We
Private browsing
https://analyzer-v2.wxcc-us1.cisco.com/analyzer/view/visualization?tId=773&rId=161953
Analyzer
Settings
curent calls
Time Zone : (UTC-05:00) Eastern Standard Time (America/New_York)
Module1

No data available to render the visualization.: Module1

63°F
Sunny
Search
4:08 PM
2/26/2024

How could this be improved?

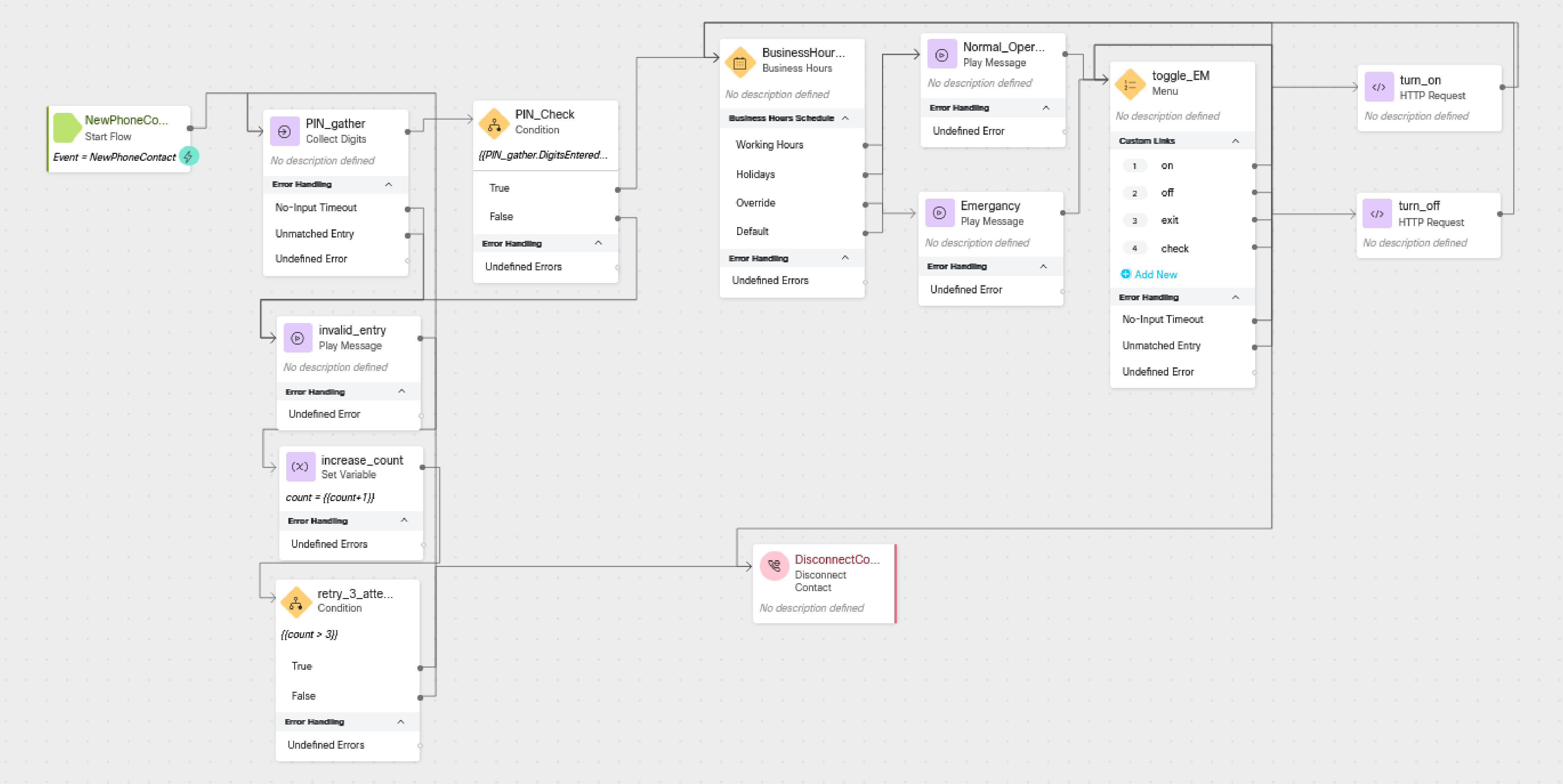
- Place cherry picking in a subflow
- Create a UI with buttons for assignment
- Use an external data source and do a data dip
- Add logic to turn off the override once the call is delivered

Update emergency close via IVR

Update emergency close via IVR

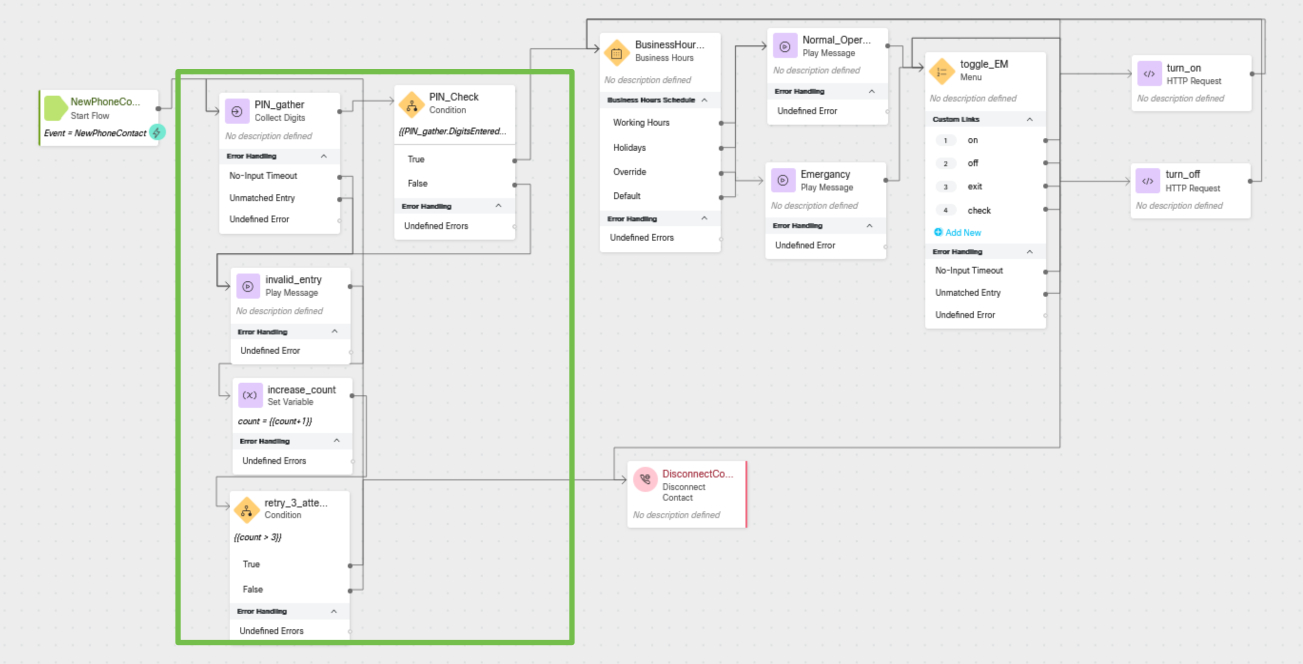
- Use Case:
  - In the event of an emergency, supervisors need to update the welcome message and/or update callers in the queue with a message.
- Features Used
  - Business Hours
  - Calling APIs from a flow
- APIs used in demo
  - Overrides

Update emergency close via IVR



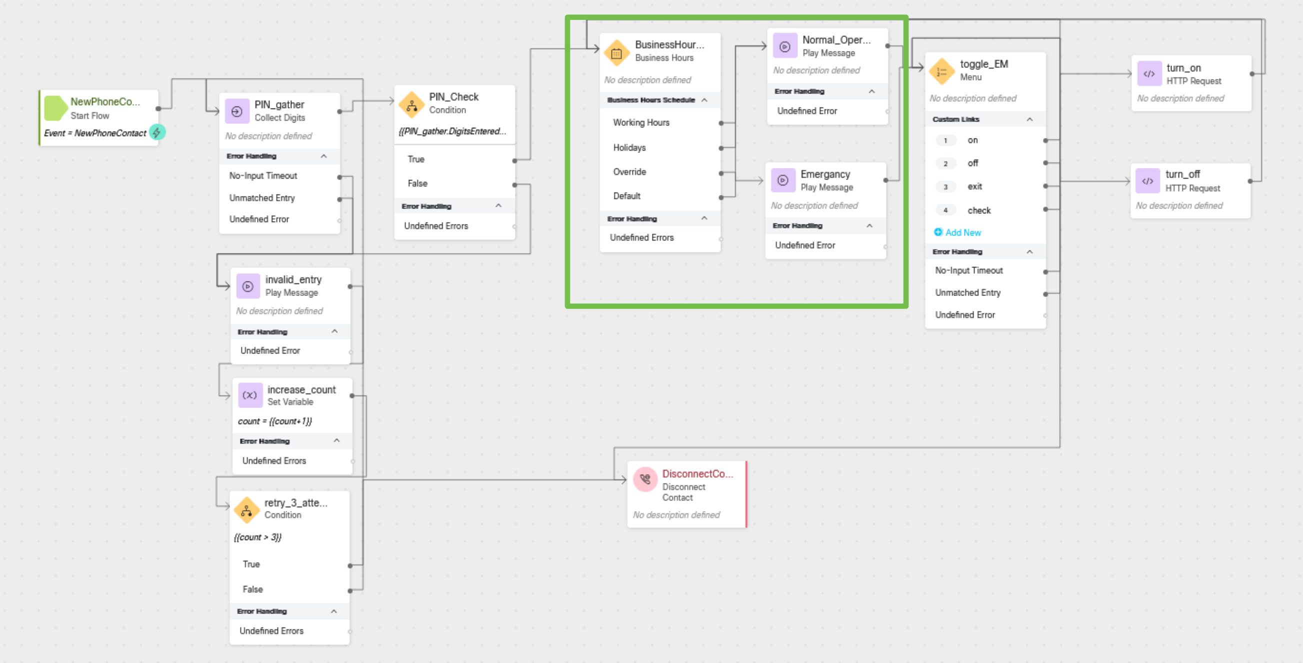
Update emergency close via IVR

- Check for auth pin



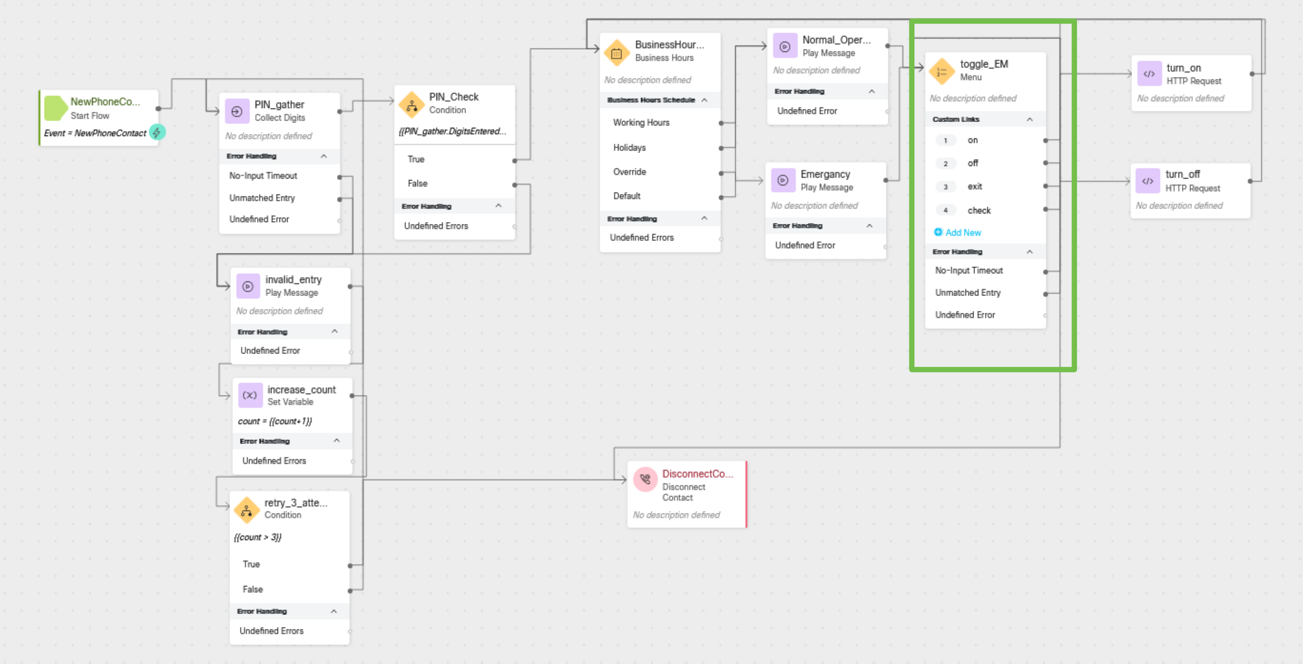
Update emergency close via IVR

- Check for auth pin
- Check Current Status



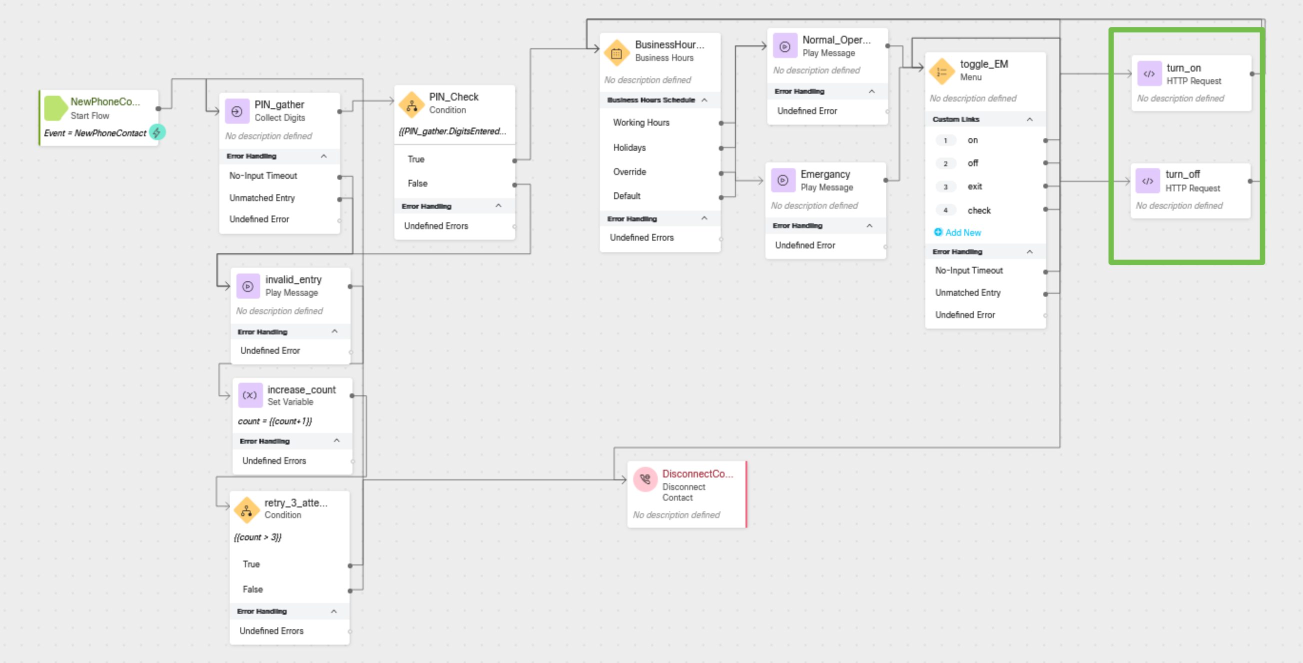
Update emergency close via IVR

- Check for auth pin
- Check Current Status
- Offer Menu options
  - Turn on emergency mode
  - Turn off emergency mode
  - Hang up
  - Check Status Again



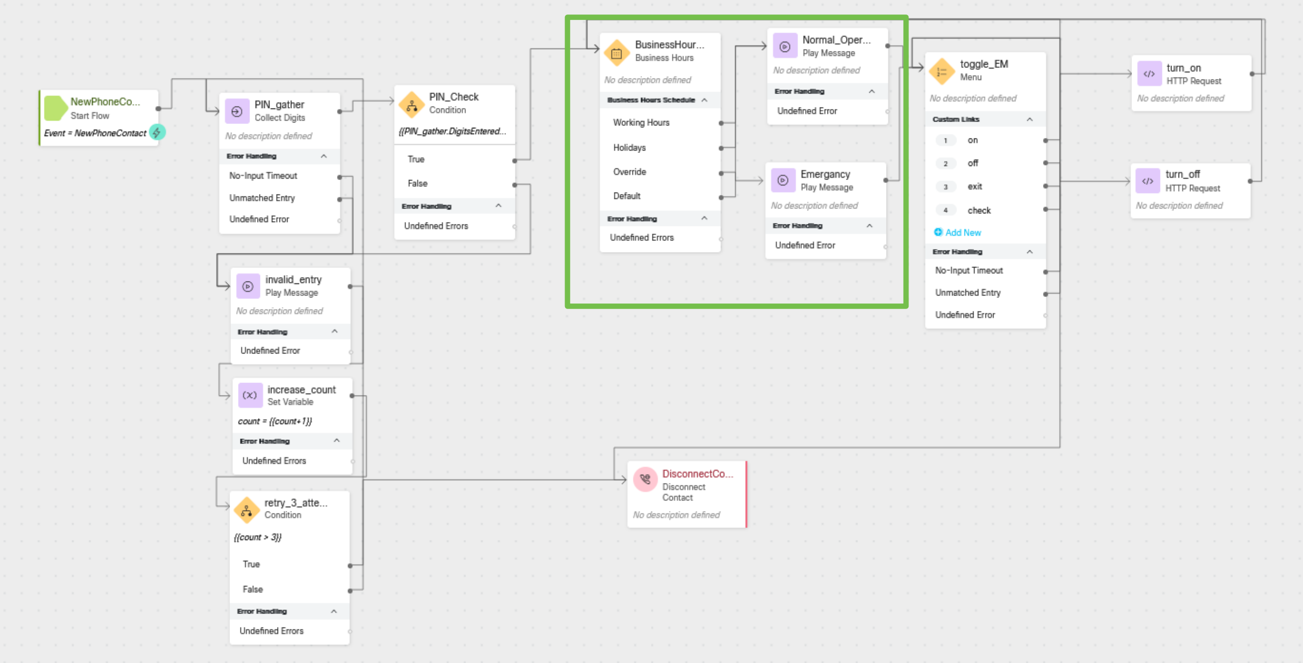
Update emergency close via IVR

- Check for auth pin
- Check Current Status
- Offer Menu options
  - Turn on emergency mode
  - Turn off emergency mode
  - Hang up
  - Check Status Again
- Turn on or off emergency mode

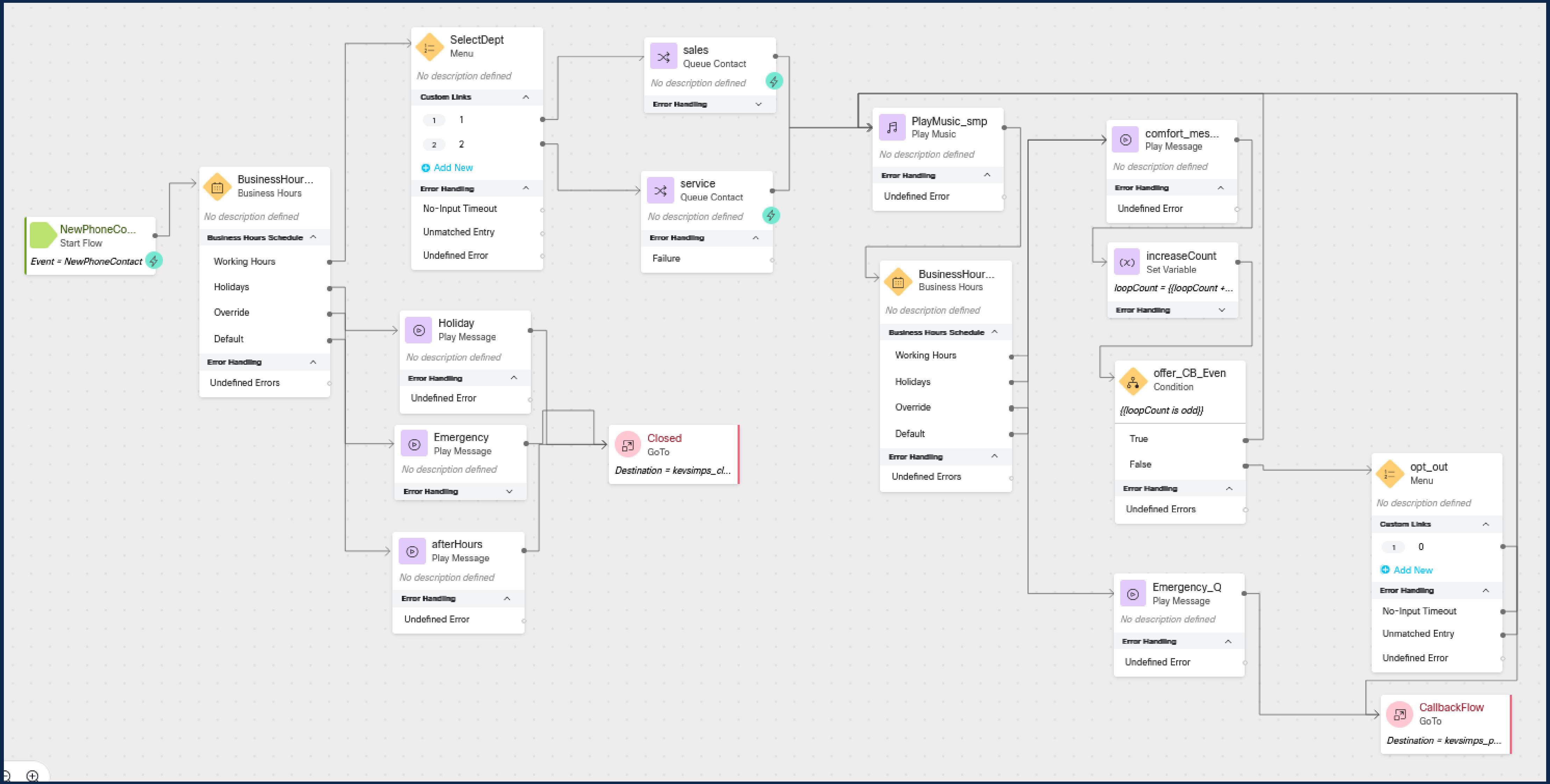


Update emergency close via IVR

- Check for auth pin
- Check Current Status
- Offer Menu options
    - Turn on emergency mode
    - Turn off emergency mode
    - Hang up
    - Check Status Again
- Turn on or off emergency mode
- Confirm status

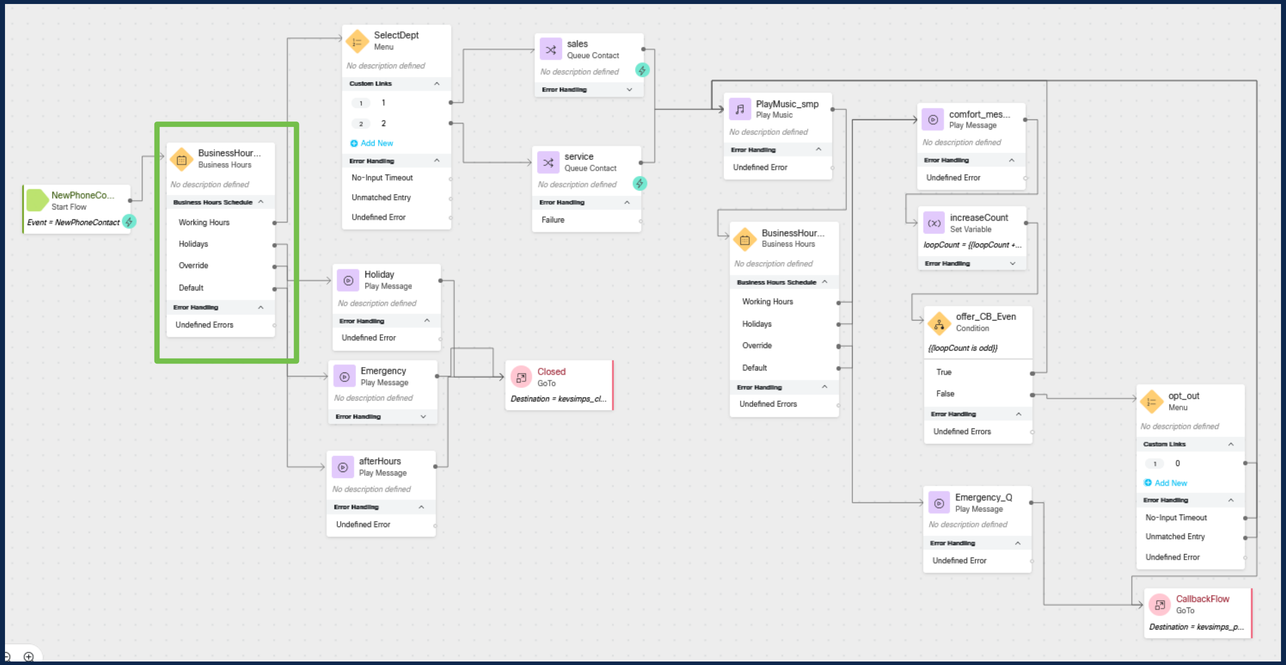


Update emergency close via IVR



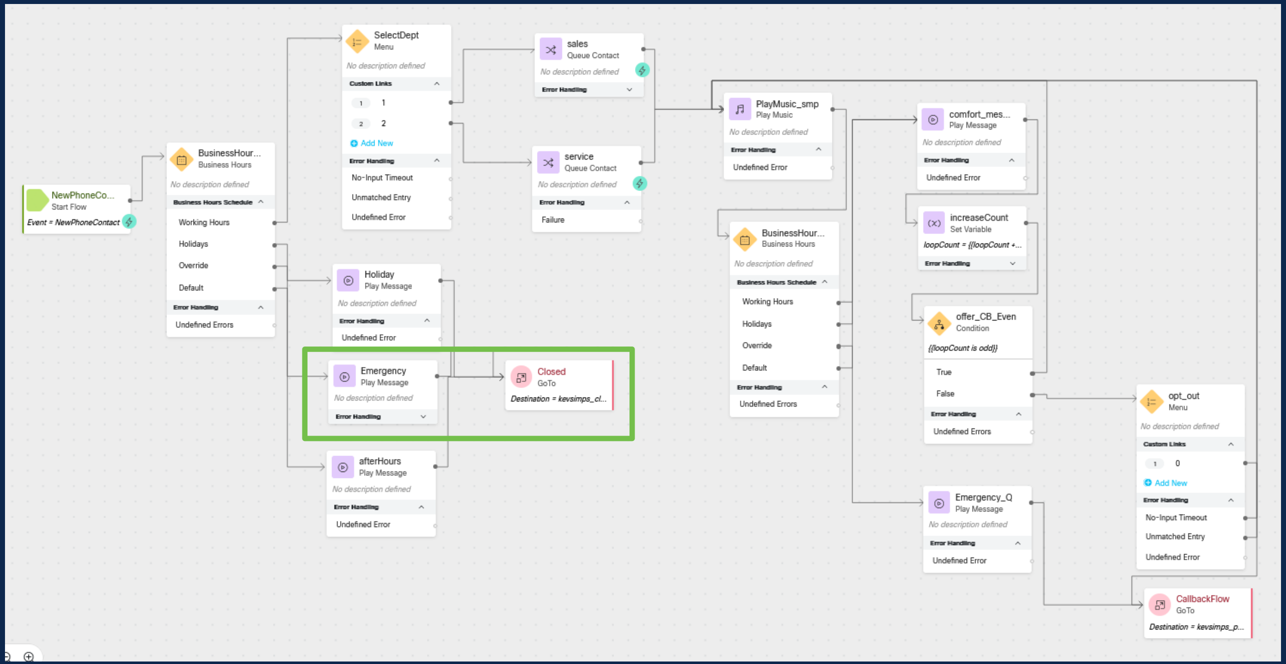
Update emergency close via IVR

- Emergency changes will take effect on new calls



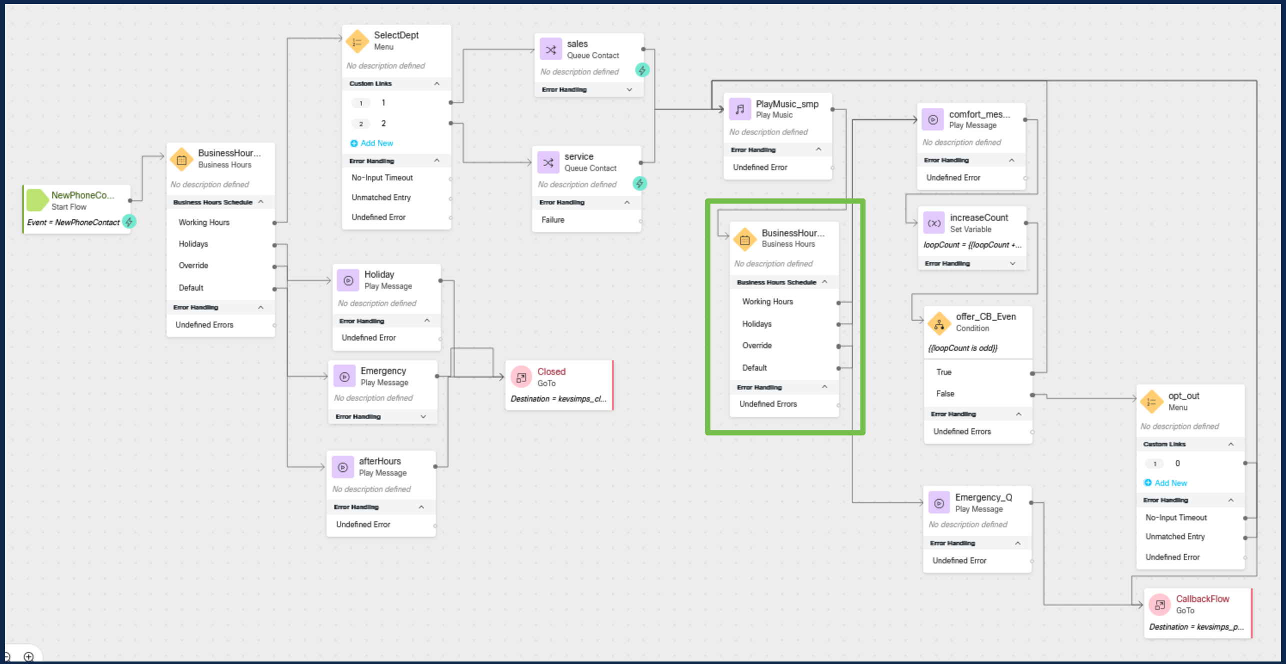
Update emergency close via IVR

- Emergency changes will take effect on new calls
  - Plays a different message
  - Follows the closed flow



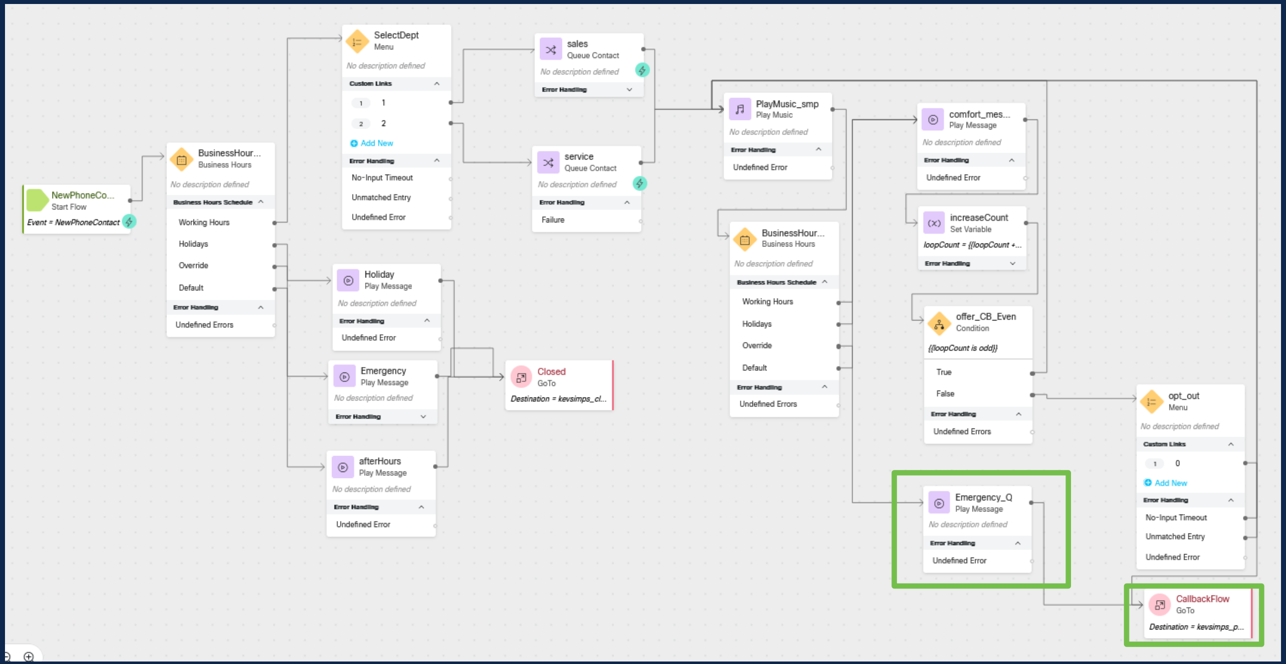
Update emergency close via IVR

- Emergency changes will take effect on new calls
  - Plays a different message
  - Follows the closed flow
- Emergency changes will take effect on calls that are in the queue



Update emergency close via IVR

- Emergency changes will take effect on new calls
  - Plays a different message
  - Follows the closed flow
- Emergency changes will take effect on calls that are in the queue
  - Plays a special message
  - Forces calls to the callback flow



How could this be improved?

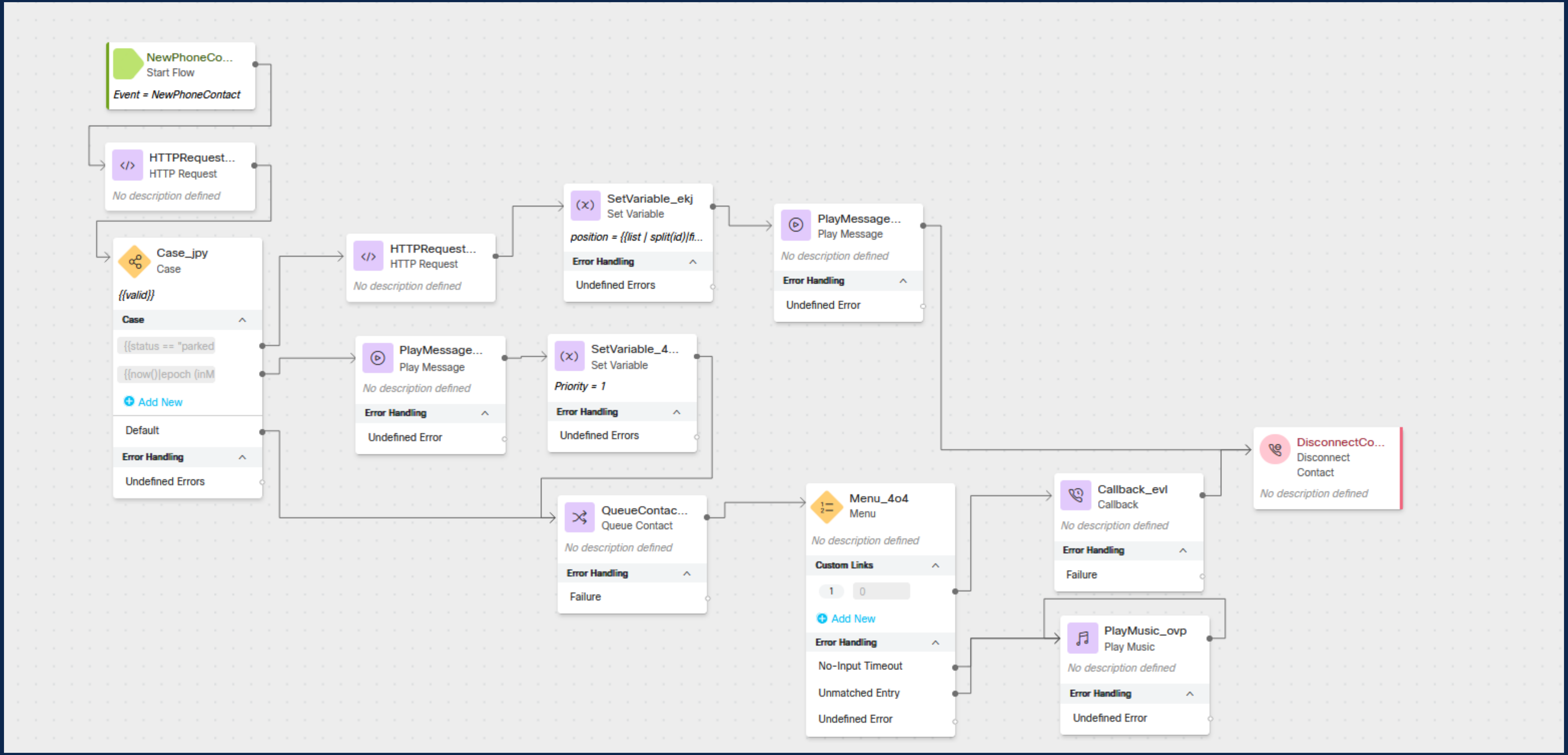
- Place emergency treatment for new calls in a subflow
- Integrate with another system for in call authorization instead of just a pin
- Add logging so you can report on who made the change and when it was made
- Send Webex message or SMS when changes are made

Catching and handling a potential duplicate callback

Catching and handling a potential duplicate callback

- Use Case:
  - Query new inbound call ANI to see if there is a callback pending and if there was a recent missed callback attempt/recently ended call (10 minutes)
  - If there is a Pending callback
    - The caller will be given their position in queue and informed that they will receive a callback when it is their turn.
  - If there was a recent missed callback attempt/recently ended call (10 minutes)
    - The caller will be moved to a P1 in the queue as it is still their turn.
  - If neither is true, the call will be placed in the queue treatment.
- Features Used
  - Calling APIs from a flow
- APIs used in demo
  - Search API

Catching and handling a potential duplicate callback



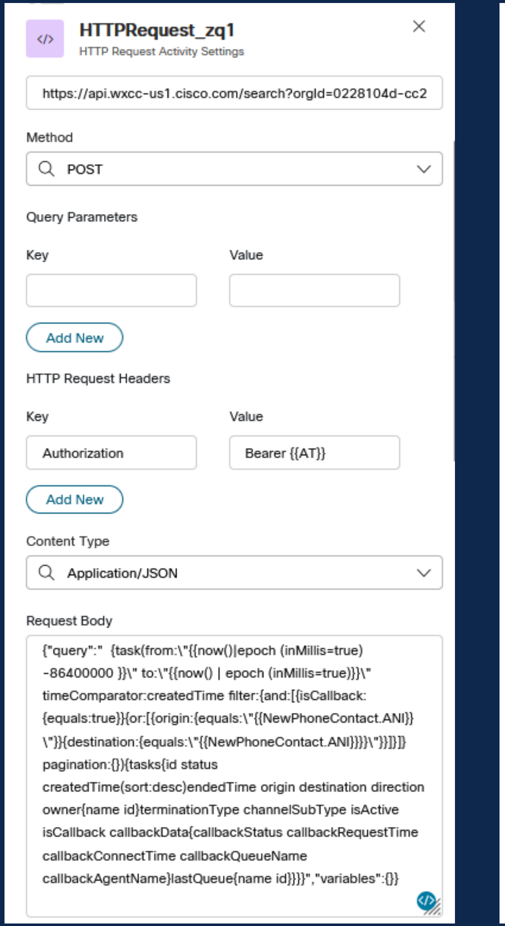
Event = NewPhoneContact

{{valid}}

position = {{list | split(id)|fi...

Priority = 1

Catching and handling a potential duplicate callback



HTTPRequest_zq1

HTTP Request Activity Settings

https://api.wxcc-us1.cisco.com/search?orgId=0228104d-cc2

Method: POST

Query Parameters

Key	Value

Add New

HTTP Request Headers

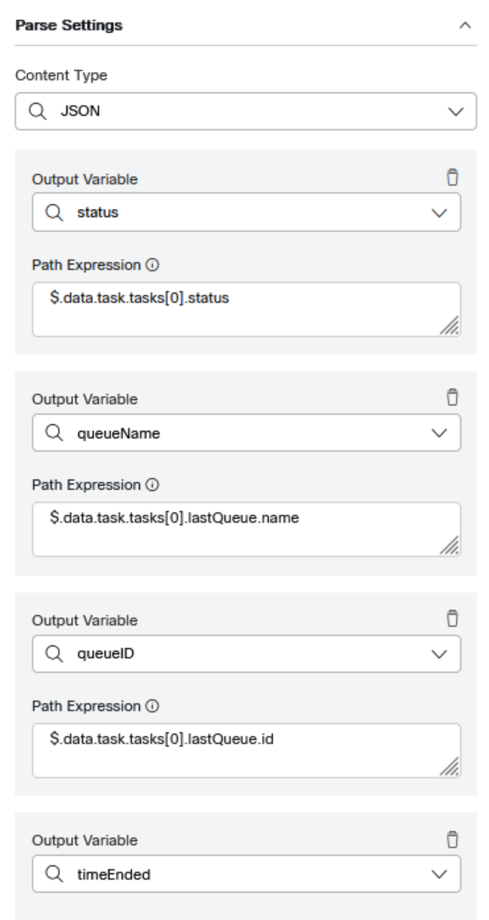
Key	Value
Authorization	Bearer {{AT}}

Add New

Content Type: Application/JSON

Request Body

```
{"query":"  {task(from:\"{{now()|epoch (inMillis=true)
-86400000 }}\" to:\"{{now() | epoch (inMillis=true)}}\"
timeComparator:createdTime filter:{and:[{isCallback:
{equals:true}}{or:[{origin:{equals:\"{{NewPhoneContact.ANI}}
\"}}{destination:{equals:\"{{NewPhoneContact.ANI}}}}\"}}]}]}
pagination:{}){tasks{id status
createdTime(sort:desc)endedTime origin destination direction
owner{name id}terminationType channelSubType isActive
isCallback callbackData{callbackStatus callbackRequestTime
callbackConnectTime callbackQueueName
callbackAgentName}lastQueue{name id}}}}","variables":{}}
```



Parse Settings

Content Type: JSON

Output Variable: status

Path Expression: $.data.task.tasks[0].status

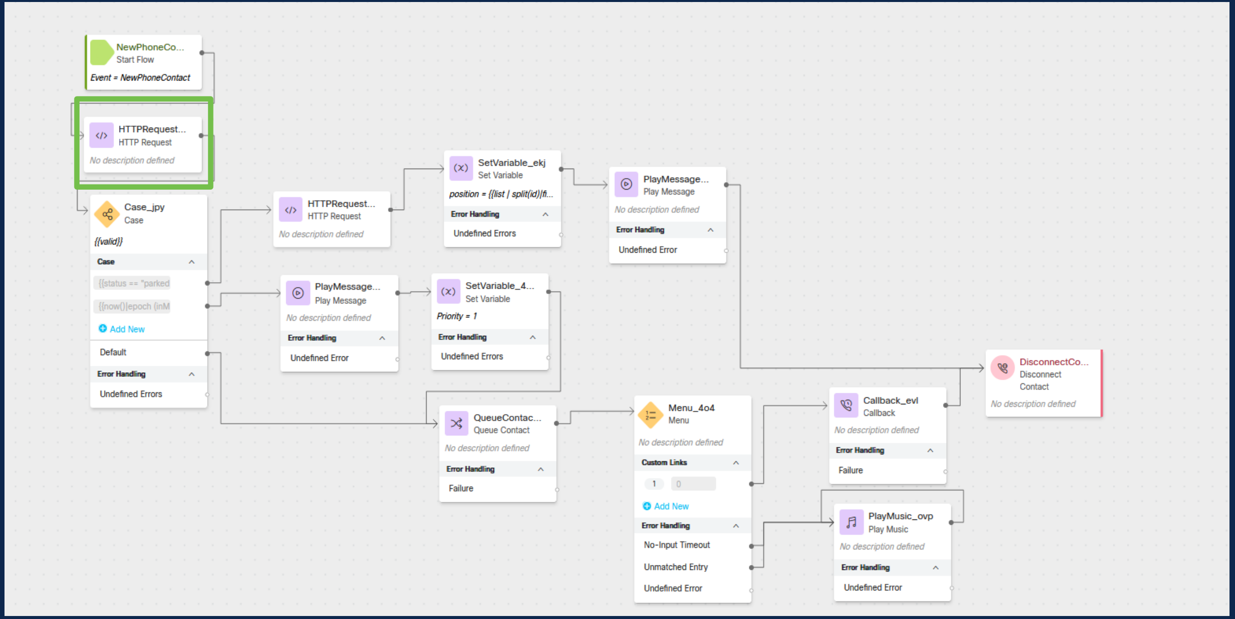
Output Variable: queueName

Path Expression: $.data.task.tasks[0].lastQueue.name

Output Variable: queueID

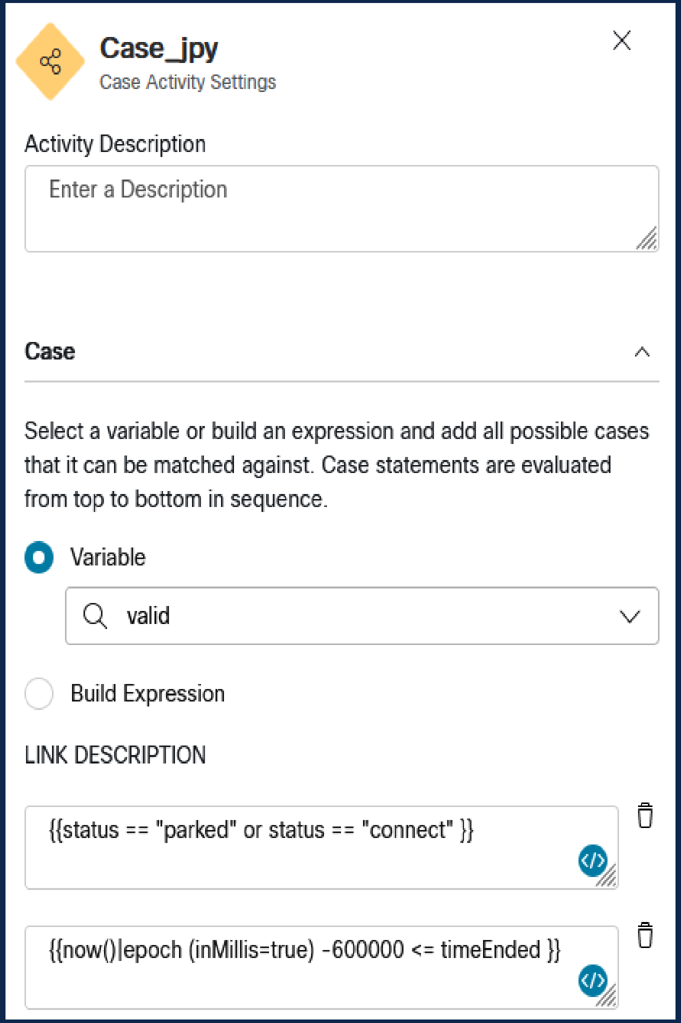
Path Expression: $.data.task.tasks[0].lastQueue.id

Output Variable: timeEnded



- Query Search API filtering on isCallback and ANI for both inbound and outbound calls
- Returning
  - id
  - Status
  - endedTime

Catching and handling a potential duplicate callback



Case_jpy

Case Activity Settings

Activity Description

Enter a Description

Case

Select a variable or build an expression and add all possible cases that it can be matched against. Case statements are evaluated from top to bottom in sequence.

● Variable

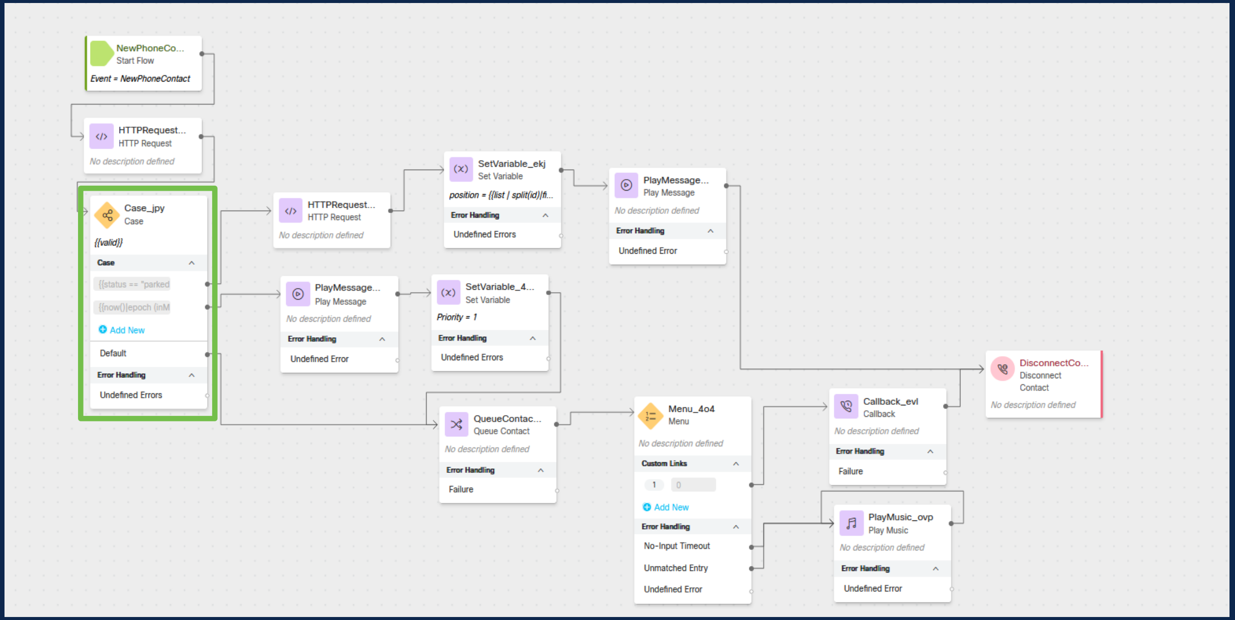
valid

○ Build Expression

LINK DESCRIPTION

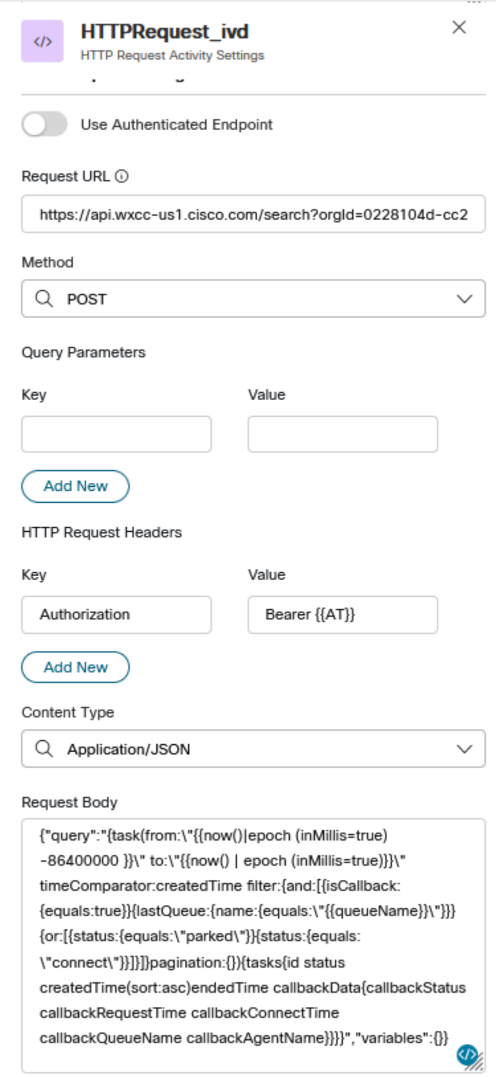
{{status == "parked" or status == "connect" }}

{{now()|epoch (inMillis=true) -600000 <= timeEnded }}



- If the Status is parked or connect (in case of RONA)
  - Query Search API for all calls in queue

Catching and handling a potential duplicate callback



HTTPRequest_ivd

HTTP Request Activity Settings

Use Authenticated Endpoint

Request URL: https://api.wxcc-us1.cisco.com/search?orgId=0228104d-cc2

Method: POST

Query Parameters

Key	Value

Add New

HTTP Request Headers

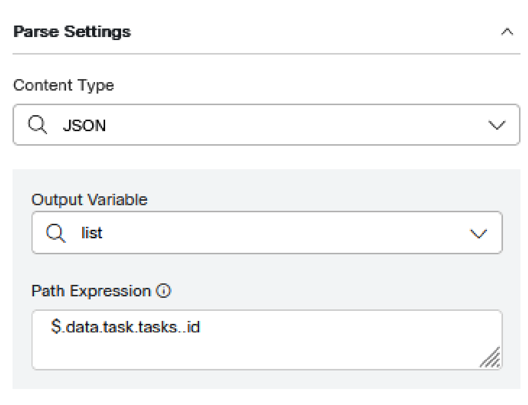
Key	Value
Authorization	Bearer {{AT}}

Add New

Content Type: Application/JSON

Request Body

```
{"query":"{task(from:\"{{now()|epoch (inMillis=true)
-86400000 }}\" to:\"{{now() | epoch (inMillis=true)}}\"
timeComparator:createdTime filter:{and:[{isCallback:
{equals:true}}{lastQueue:{name:{equals:\"{{queueName}}\"}}}
{or:[{status:{equals:\"parked\"}}{status:{equals:
\"connect\"}}]}]}pagination:{}){tasks{id status
createdTime(sort:asc)endedTime callbackData{callbackStatus
callbackRequestTime callbackConnectTime
callbackQueueName callbackAgentName}}}}","variables":{}}
```

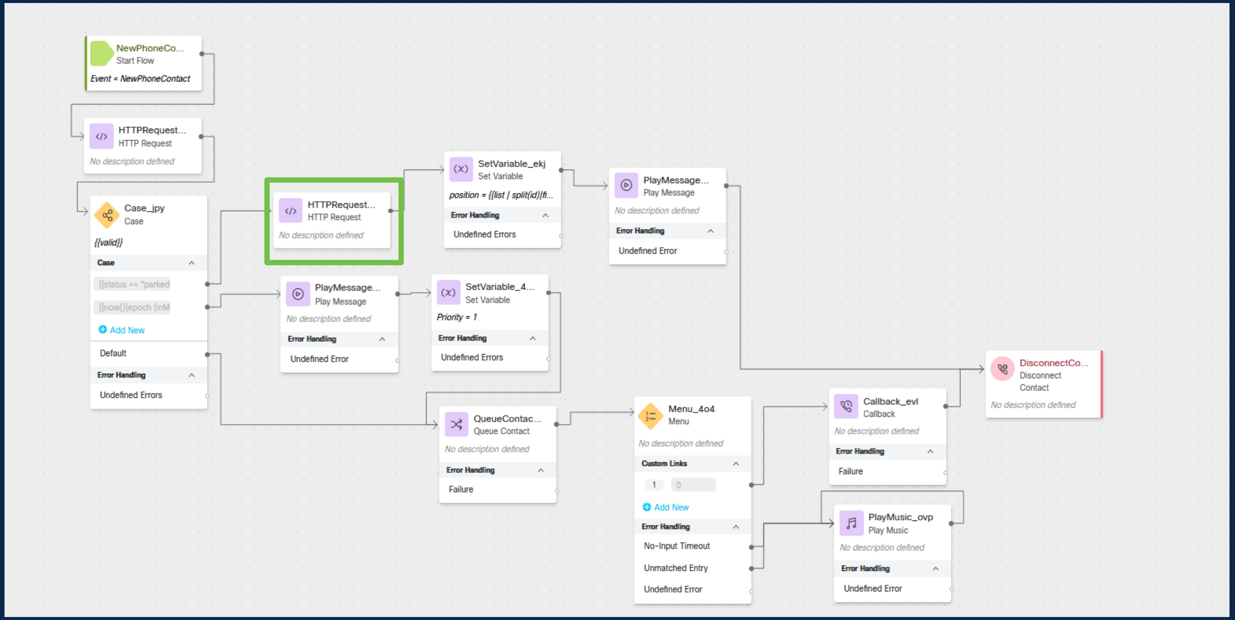


Parse Settings

Content Type: JSON

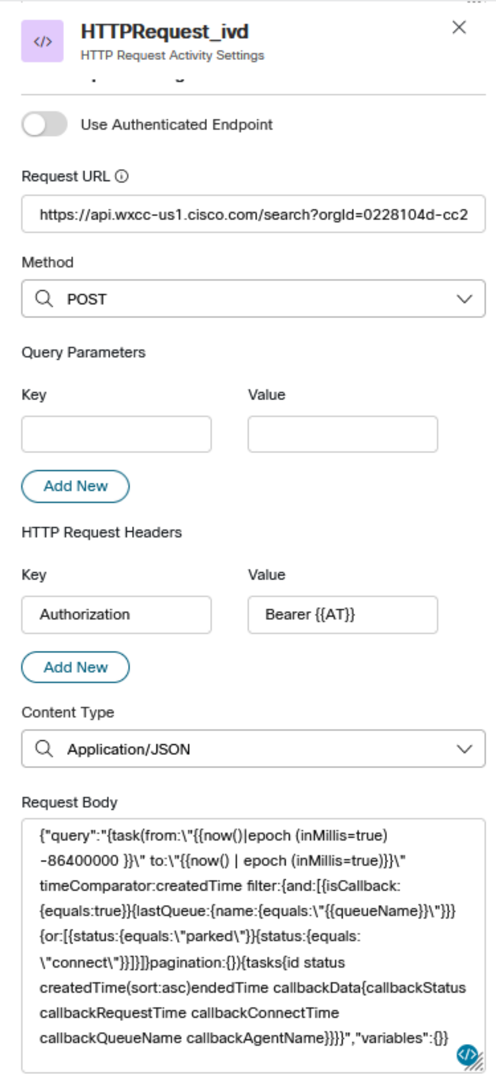
Output Variable: list

Path Expression: $.data.task.tasks..id



- If the Status is parked or connect (in case of RONA)
  - Query Search API for all calls in queue
  - Return a list of tasks

Catching and handling a potential duplicate callback



HTTPRequest_ivd

HTTP Request Activity Settings

Use Authenticated Endpoint

Request URL ⓘ

https://api.wxcc-us1.cisco.com/search?orgId=0228104d-cc2

Method

POST

Query Parameters

Key	Value

Add New

HTTP Request Headers

Key	Value
Authorization	Bearer {{AT}}

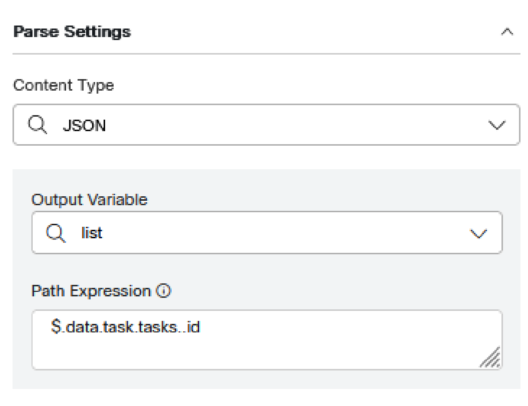
Add New

Content Type

Application/JSON

Request Body

```
{"query":"{task(from:\"{{now()|epoch (inMillis=true)
-86400000 }}\" to:\"{{now() | epoch (inMillis=true)}}\"
timeComparator:createdTime filter:{and:[{isCallback:
{equals:true}}{lastQueue:{name:{equals:\"{{queueName}}\"}}}
{or:[{status:{equals:\"parked\"}}{status:{equals:
\"connect\"}}]}]}pagination:{}){tasks{id status
createdTime(sort:asc)endedTime callbackData{callbackStatus
callbackRequestTime callbackConnectTime
callbackQueueName callbackAgentName}}}}","variables":{}}
```



Parse Settings

Content Type

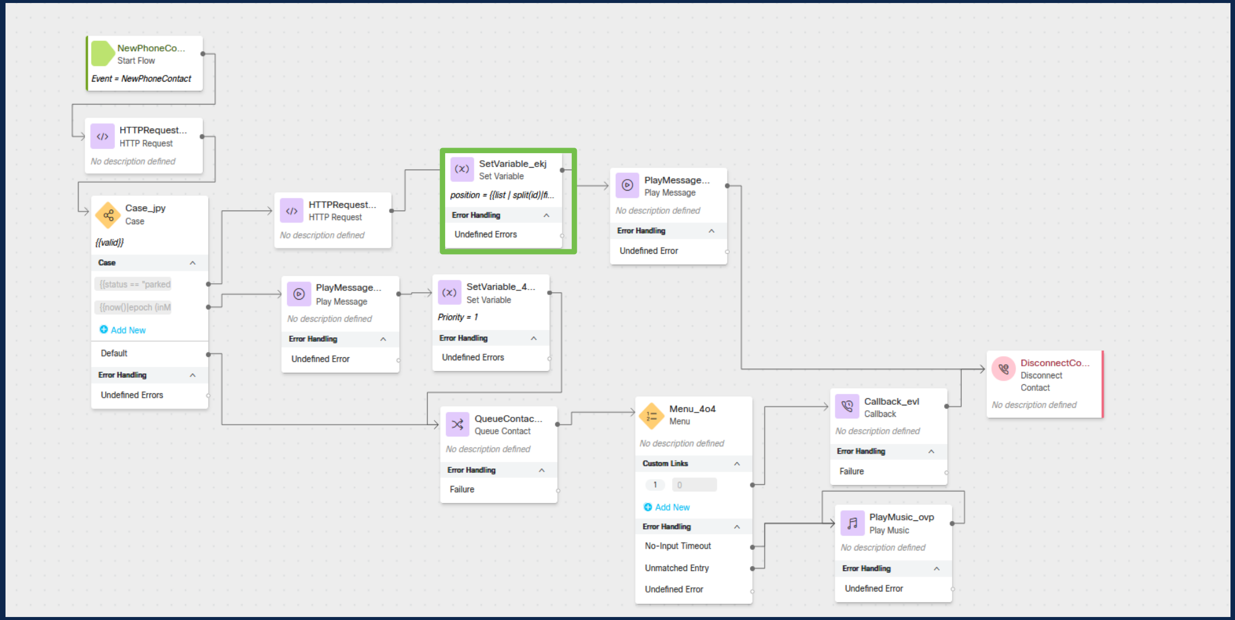
JSON

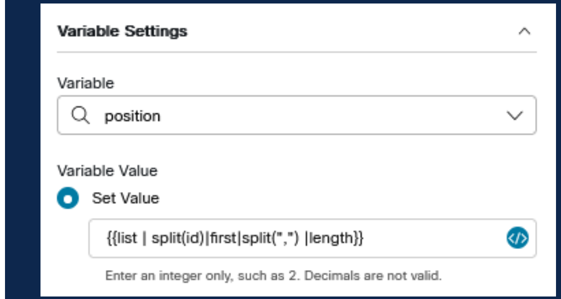
Output Variable

list

Path Expression ⓘ

$.data.task.tasks..id





Variable Settings

Variable

position

Variable Value

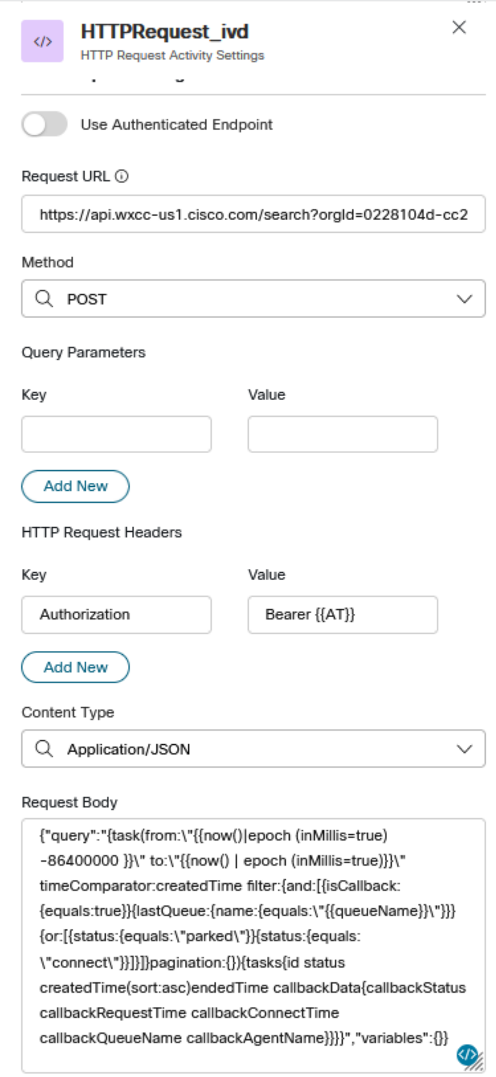
● Set Value

{{list | split(id)|first|split(",") |length}}

Enter an integer only, such as 2. Decimals are not valid.

- If the Status is parked or connect (in case of RONA)
  - Query Search API for all calls in queue
  - Return a list of tasks
  - Split the list on the id of the existing callback and count how many tasks are ahead of the existing callback

Catching and handling a potential duplicate callback



HTTPRequest_ivd

HTTP Request Activity Settings

Use Authenticated Endpoint

Request URL ⓘ

https://api.wxcc-us1.cisco.com/search?orgId=0228104d-cc2

Method

POST

Query Parameters

Key	Value

Add New

HTTP Request Headers

Key	Value
Authorization	Bearer {{AT}}

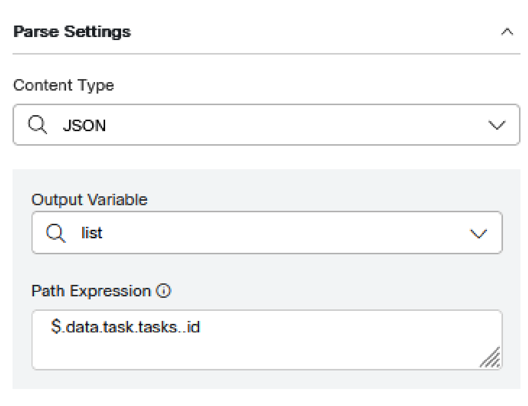
Add New

Content Type

Application/JSON

Request Body

```
{"query":"{task(from:\"{{now()|epoch (inMillis=true)
-86400000 }}\" to:\"{{now() | epoch (inMillis=true)}}\"
timeComparator:createdTime filter:{and:[{isCallback:
{equals:true}}{lastQueue:{name:{equals:\"{{queueName}}\"}}}
{or:[{status:{equals:\"parked\"}}{status:{equals:
\"connect\"}}]}]}pagination:{}){tasks{id status
createdTime(sort:asc)endedTime callbackData{callbackStatus
callbackRequestTime callbackConnectTime
callbackQueueName callbackAgentName}}}}","variables":{}}
```



Parse Settings

Content Type

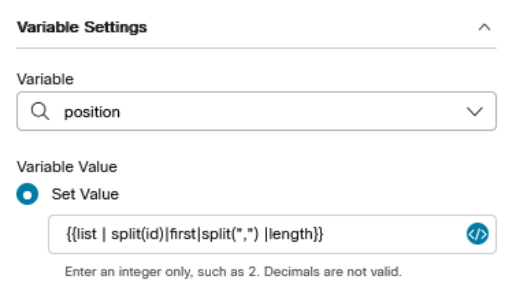
JSON

Output Variable

list

Path Expression ⓘ

$.data.task.tasks..id



Variable Settings

Variable

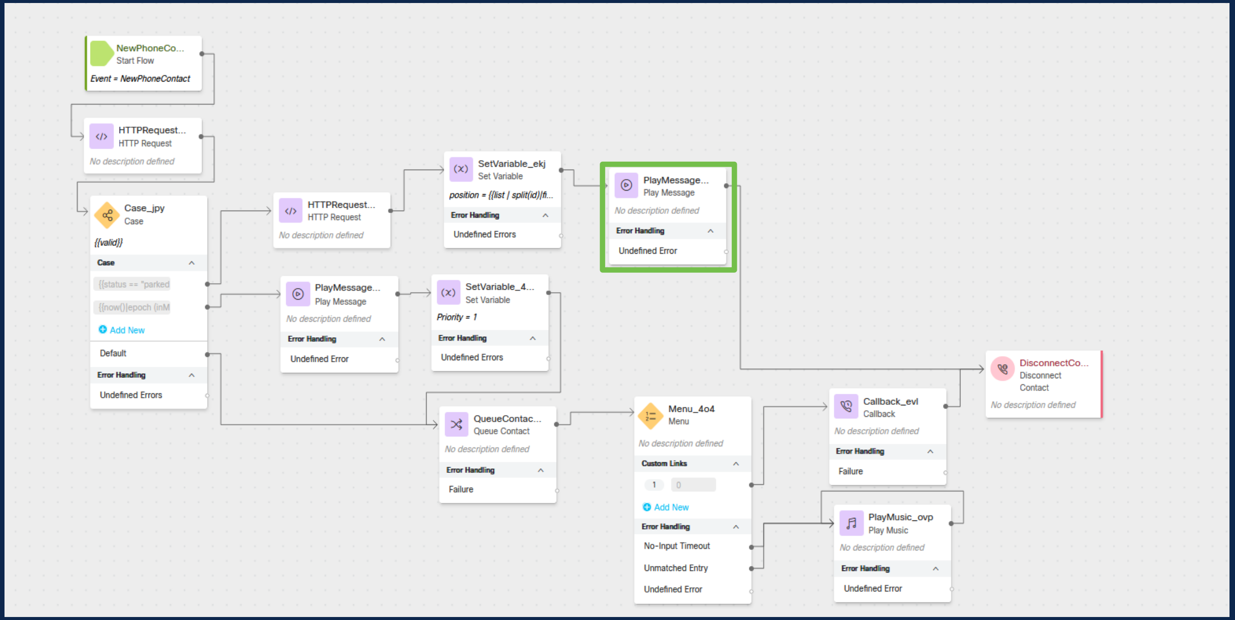
position

Variable Value

● Set Value

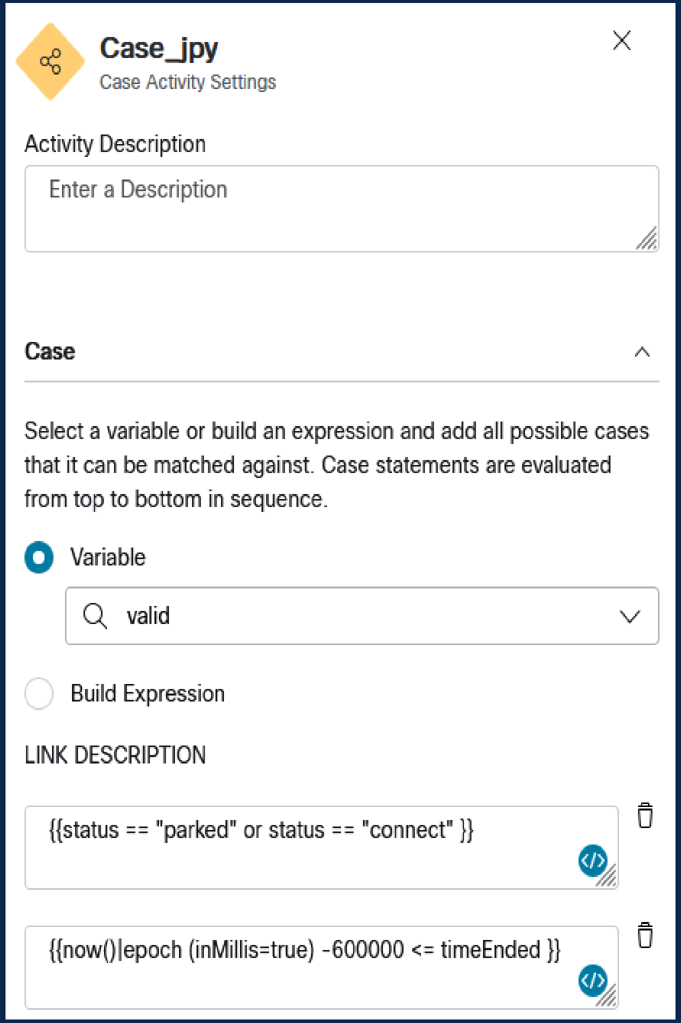
{{list | split(id)|first|split(",") |length}}

Enter an integer only, such as 2. Decimals are not valid.



- If the Status is parked or connect (in case of RONA)
  - Query Search API for all calls in queue
  - Return a list of tasks
  - Split the list on the id of the existing callback and count how many tasks are ahead if the existing callback
  - Read back the position in queue

Catching and handling a potential duplicate callback



Case_jpy

Case Activity Settings

Activity Description

Enter a Description

Case

Select a variable or build an expression and add all possible cases that it can be matched against. Case statements are evaluated from top to bottom in sequence.

● Variable

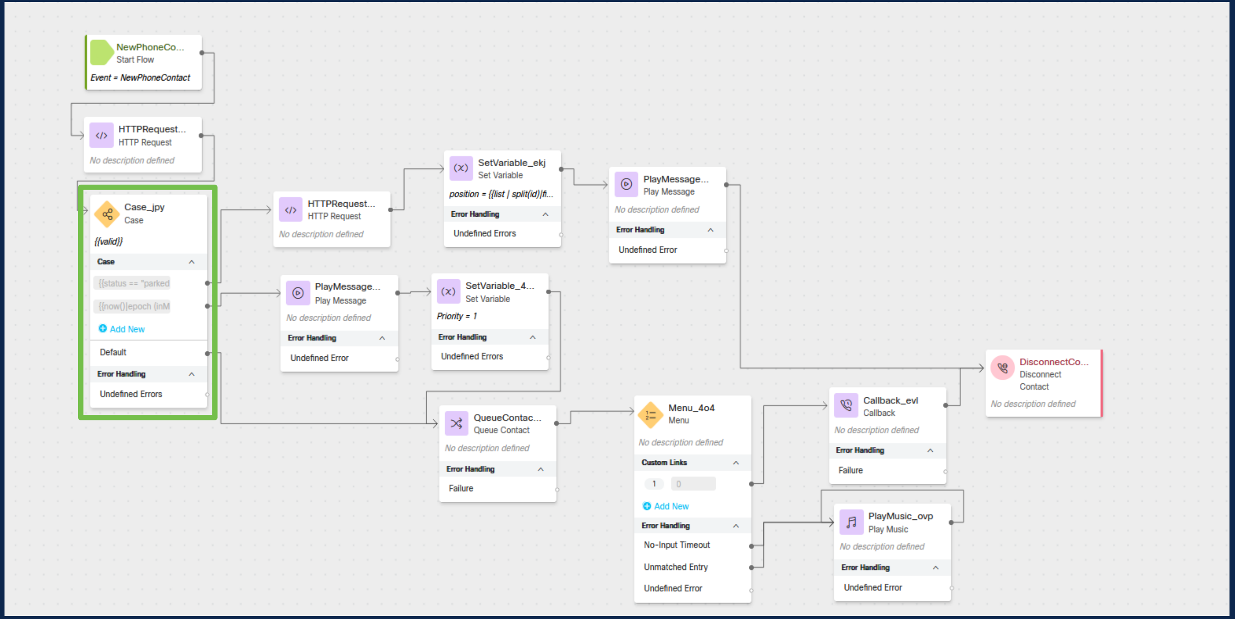
valid

○ Build Expression

LINK DESCRIPTION

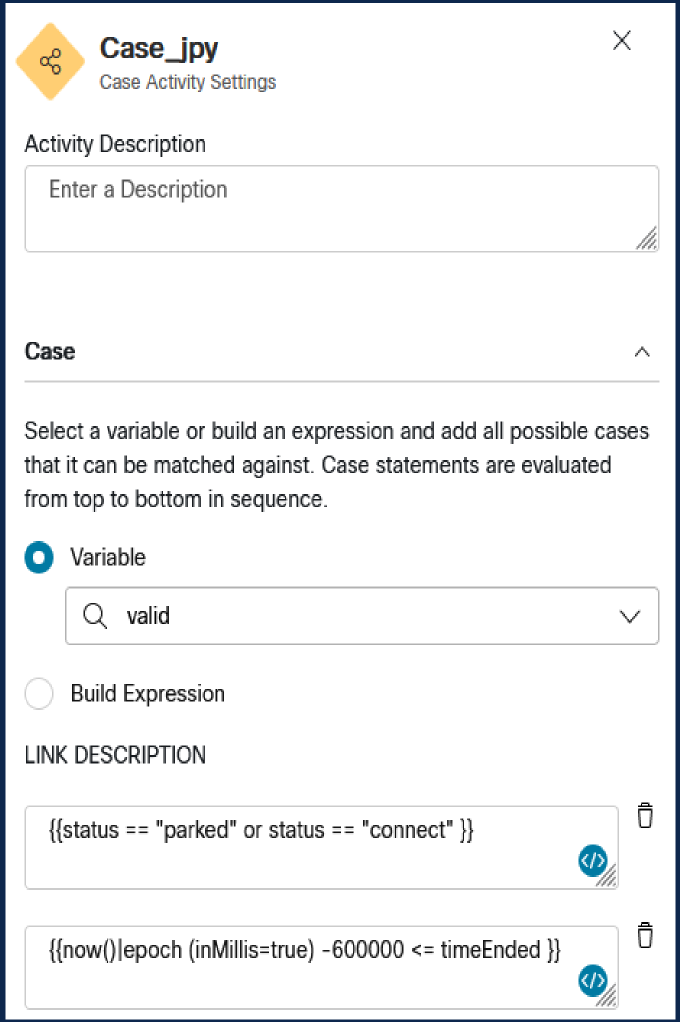
{{status == "parked" or status == "connect" }}

{{now()|epoch (inMillis=true) -600000 <= timeEnded }}



- If the timeEnded is less than 10 minutes ago

Catching and handling a potential duplicate callback



Case_jpy

Case Activity Settings

Activity Description

Enter a Description

Case

Select a variable or build an expression and add all possible cases that it can be matched against. Case statements are evaluated from top to bottom in sequence.

● Variable

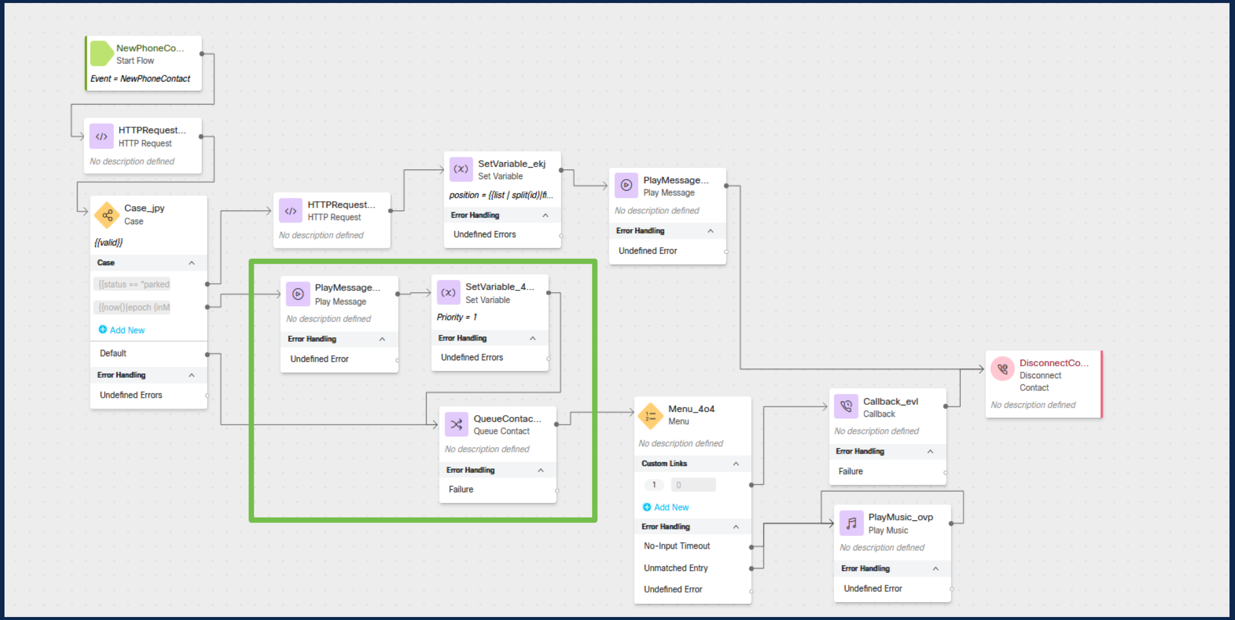
valid

○ Build Expression

LINK DESCRIPTION

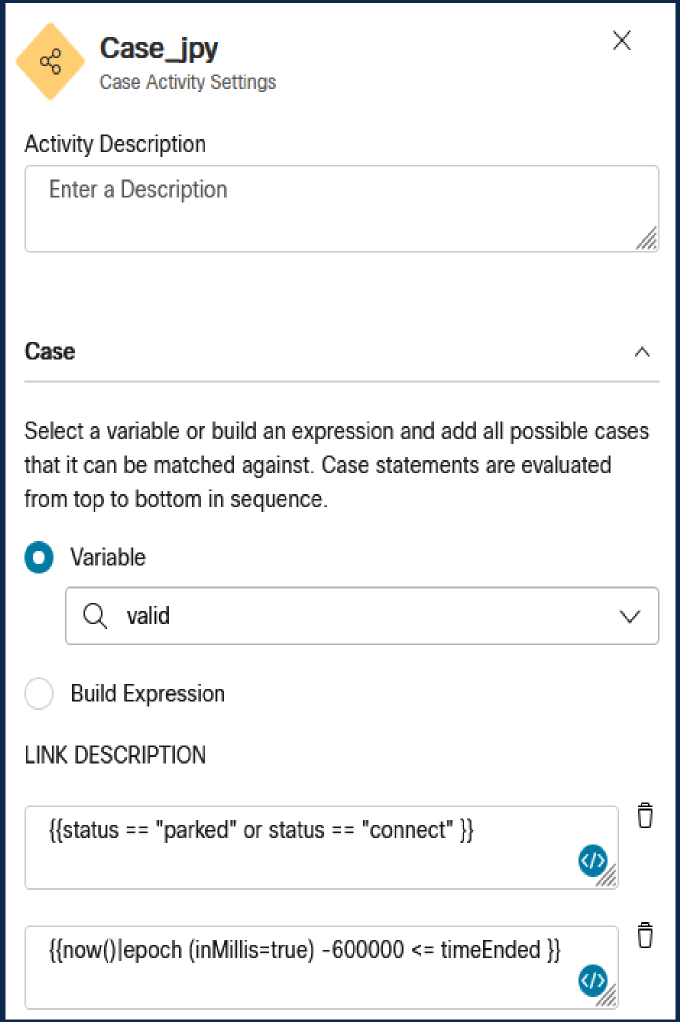
{{status == "parked" or status == "connect" }}

{{now()|epoch (inMillis=true) -600000 <= timeEnded }}



- If the timeEnded is less than 10 minutes ago
  - Play a message that the call is being prioritized
  - Set the queue priority to 1
  - Place the call in the queue

Catching and handling a potential duplicate callback



Case_jpy

Case Activity Settings

Activity Description

Enter a Description

Case

Select a variable or build an expression and add all possible cases that it can be matched against. Case statements are evaluated from top to bottom in sequence.

● Variable

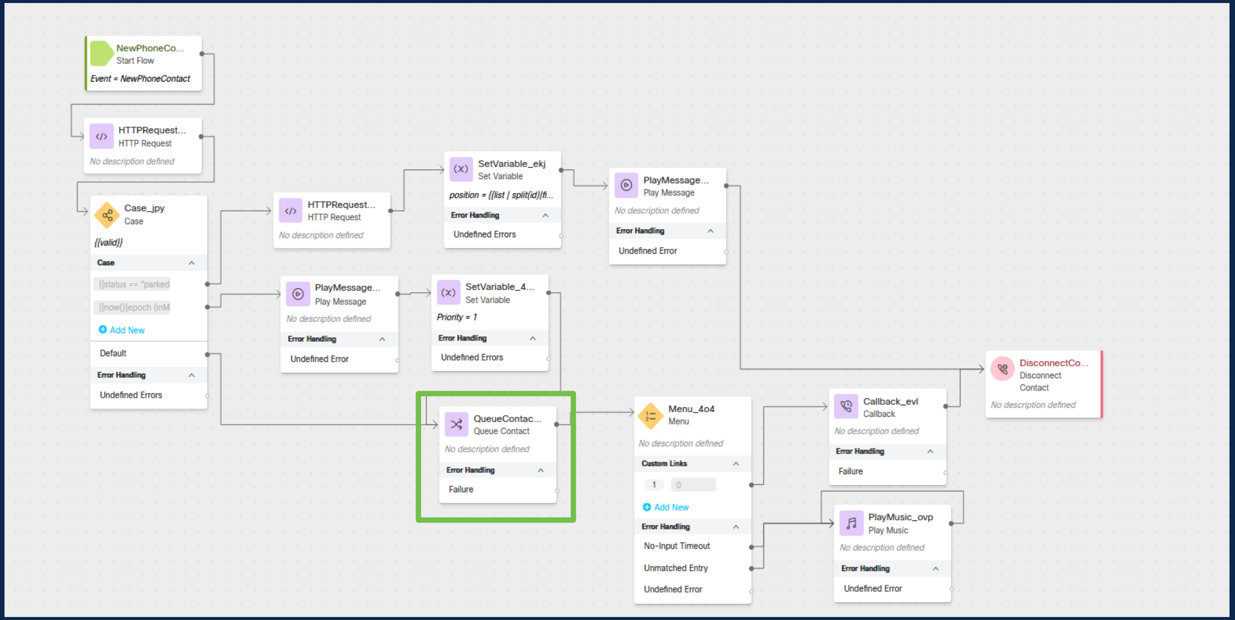
valid

○ Build Expression

LINK DESCRIPTION

{{status == "parked" or status == "connect" }}

{{now()|epoch (inMillis=true) -600000 <= timeEnded }}



- If neither case is true
  - The call is new
  - Place the call in the queue

How could this be improved?

- Use a subflow so you can easily add the feature to additional flows
- Use Business Hours to only run the logic when call volume is high
- Offer to send an SMS message when the callback is getting close to the top of the list
- Offer to schedule a callback

Scheduling a callback

Scheduling a callback

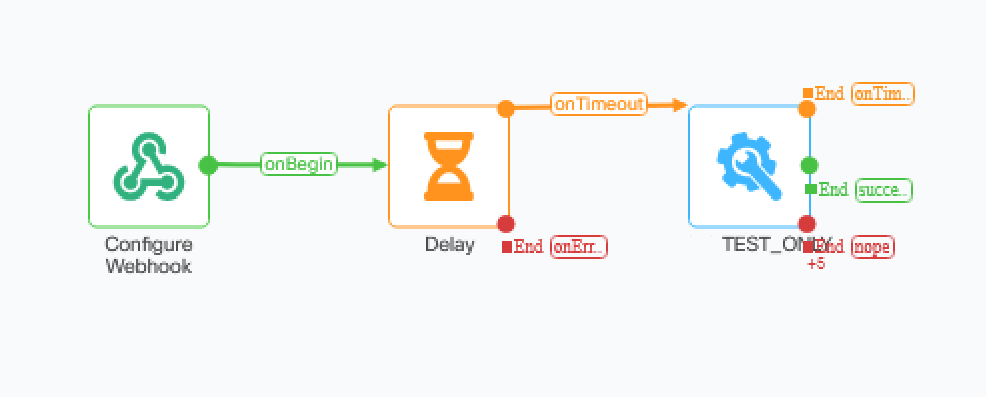
- Use case:
  - Agent wants to schedule a callback for a customer.
  - Customer wants to schedule a callback for a different day instead of waiting in queue.

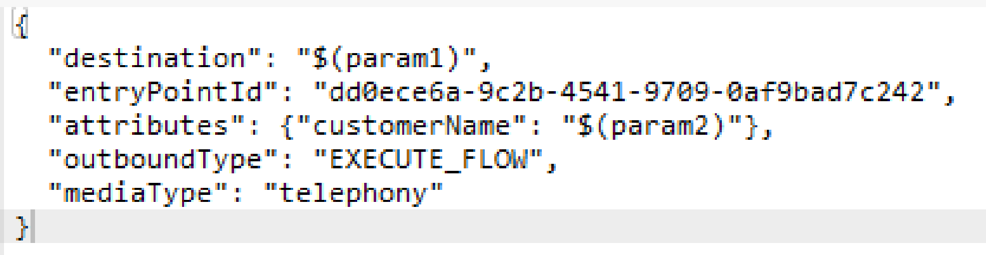
- Features used:
  - Webex Connect Webhook
  - Digital Flow Builder

- APIs used:
  - Task

Scheduling a callback

- Webhook accepts
  - Caller’s name
  - Callback Time
  - Callback Number
- Delay
  - Holds flow until callback time is reached
- Call Task API
  - Callback Number
  - Caller Name as attribute
  - Entry Point ID to execute flow from





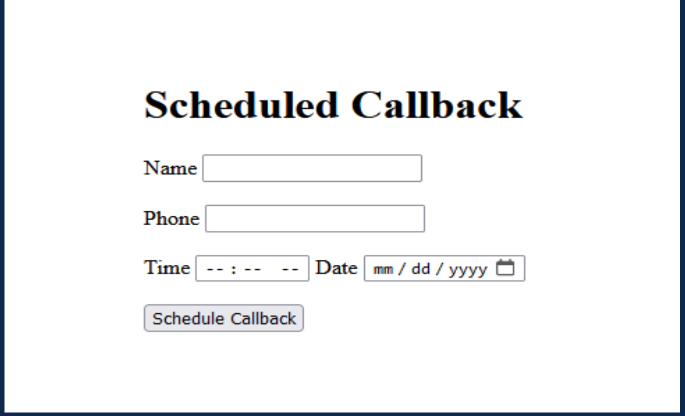
```
{
  "destination": "$(param1)",
  "entryPointId": "dd0ece6a-9c2b-4541-9709-0af9bad7c242",
  "attributes": {"customerName": "$(param2)"},
  "outboundType": "EXECUTE_FLOW",
  "mediaType": "telephony"
}
```

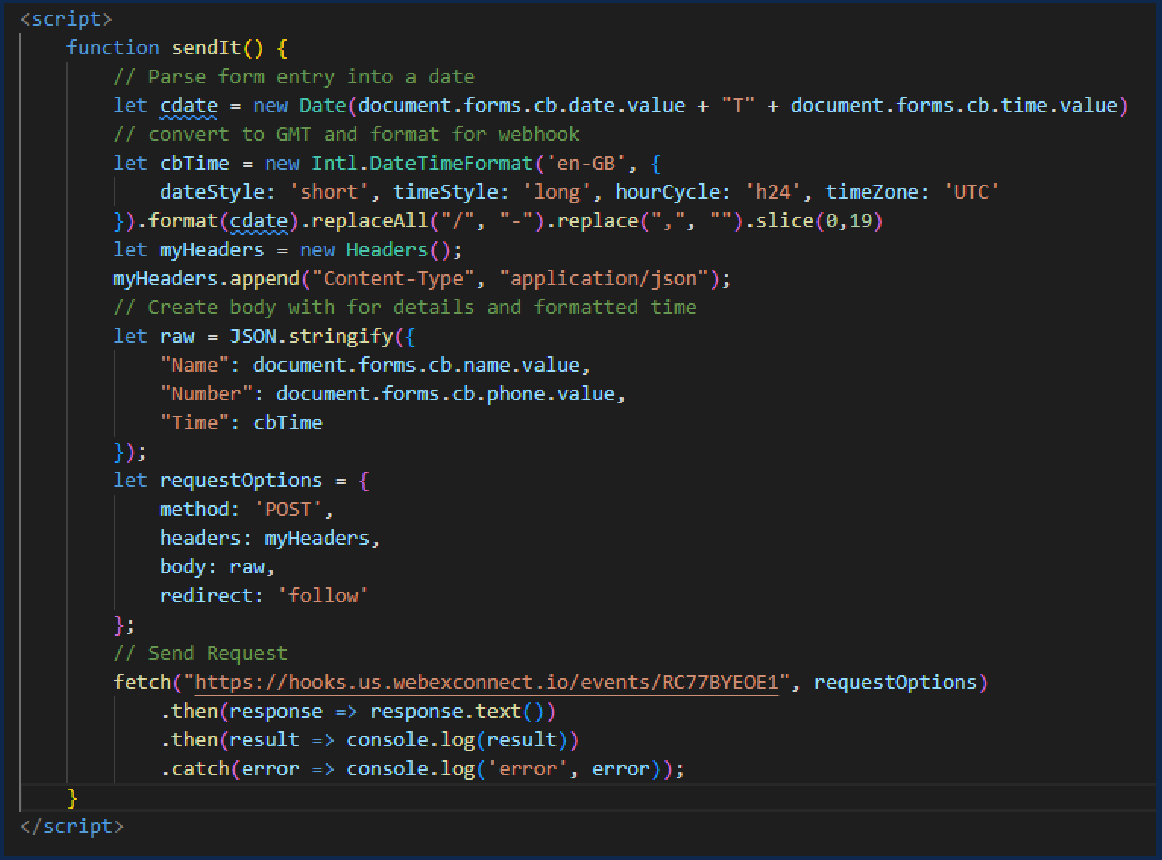
Scheduling a callback

- The user fills out the information

- The form data gets parsed, and the date is formatted to be accepted by the webhook

- The request is sent to the webhook





```
<script>
    function sendIt() {
        // Parse form entry into a date
        let cdate = new Date(document.forms.cb.date.value + "T" + document.forms.cb.time.value)
        // convert to GMT and format for webhook
        let cbTime = new Intl.DateTimeFormat('en-GB', {
            dateStyle: 'short', timeStyle: 'long', hourCycle: 'h24', timeZone: 'UTC'
        }).format(cdate).replaceAll("/", "-").replace(",", "").slice(0,19)
        let myHeaders = new Headers();
        myHeaders.append("Content-Type", "application/json");
        // Create body with for details and formatted time
        let raw = JSON.stringify({
            "Name": document.forms.cb.name.value,
            "Number": document.forms.cb.phone.value,
            "Time": cbTime
        });
        let requestOptions = {
            method: 'POST',
            headers: myHeaders,
            body: raw,
            redirect: 'follow'
        };
        // Send Request
        fetch("https://hooks.us.webexconnect.io/events/RC77BYEOE1", requestOptions)
            .then(response => response.text())
            .then(result => console.log(result))
            .catch(error => console.log('error', error));
    }
</script>
```

How could this be improved?

- Offer to schedule a callback window in a callflow
- Send an SMS message reminder message
- Capture the reason for the call so that you can best route to a resource

Transfer a Skilled Chat to Another Queue

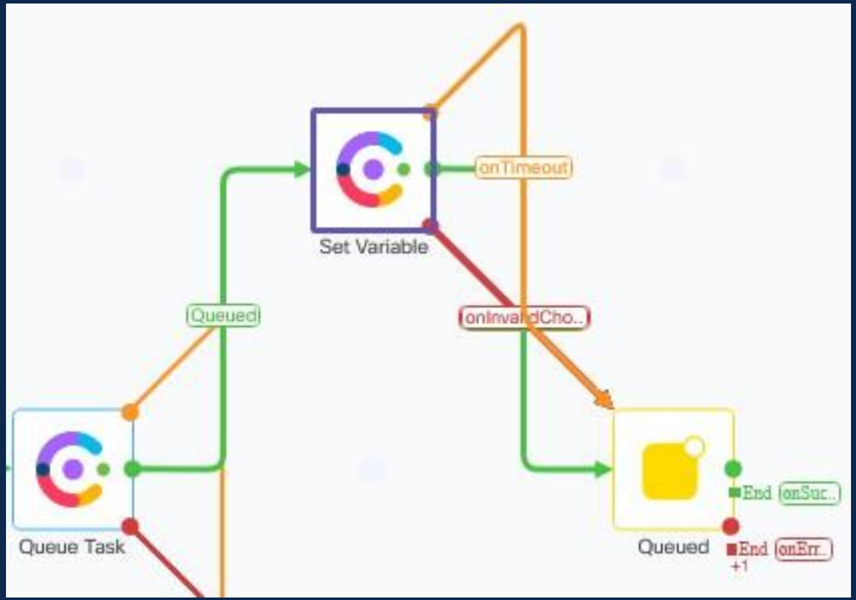
Transfer a Skilled Chat to Another Queue

- Use case:
  - An agent needs to transfer a skilled chat to another queue
- Features used:
  - Webex Connect Task Modified Flow
  - Digital Flow Builder
  - Reportable Global Variable which is Agent Editable
  - Integration Custom Node
  - Developer Integration App
  - Text Skill for transfering
- APIs used:
  - Search API

Transfer a Skilled Chat to Another Queue

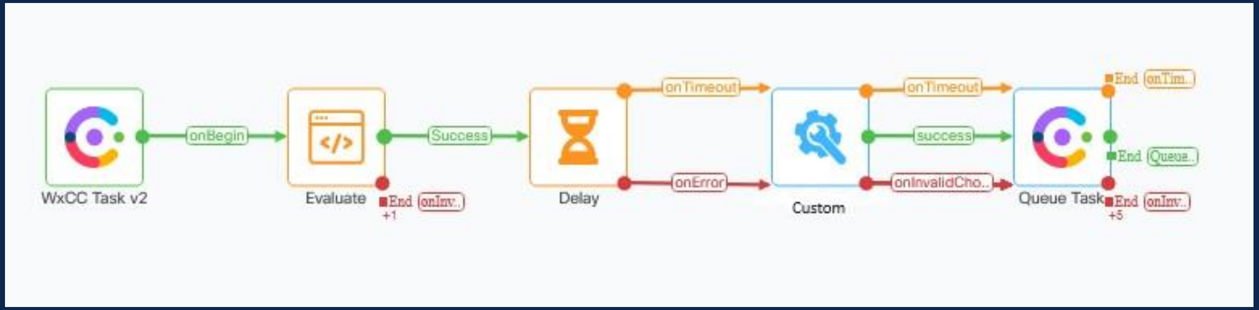
- Agent Editable Global Variable created on the tenant
  - Will be used to set the text skill

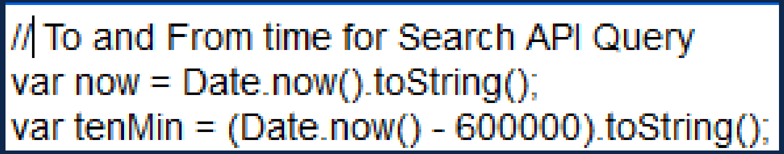
- Set Variable Node added to existing chat flow
  - Used to add the Global Variable to the flow



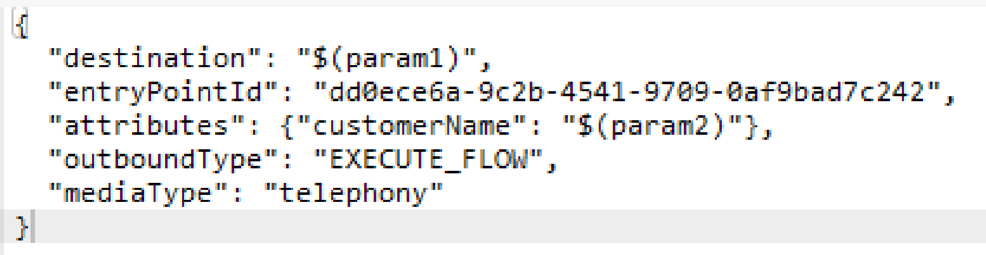
Transfer a Skilled Chat to Another Queue

- Agent will transfer the chat to a queue with no teams assigned
- Flow will run when a chat is transferred
- Evaluate Node
  - Create time variables for the Search API Query
- Delay (2 seconds)
  - Allows the Global Variable to update in the Analyzer data





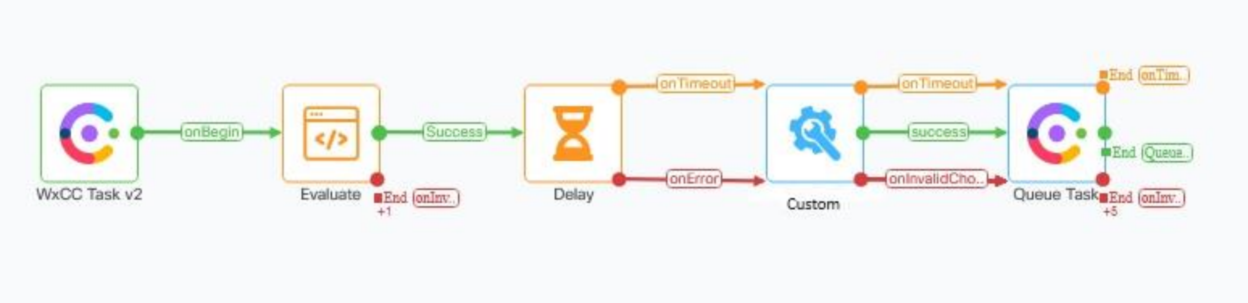
```
// To and From time for Search API Query
var now = Date.now().toString();
var tenMin = (Date.now() - 600000).toString();
```

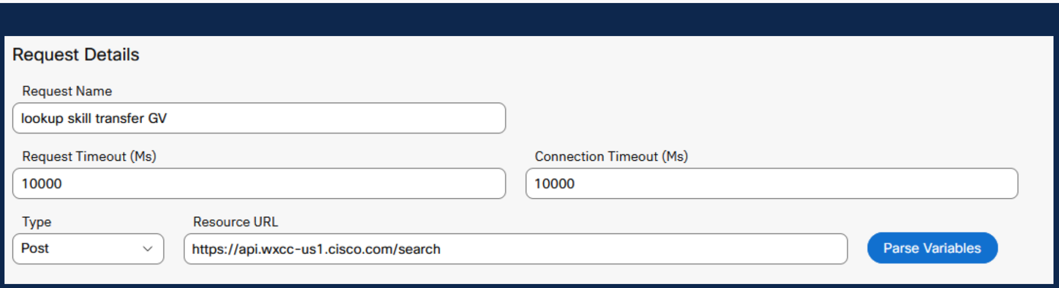


```
{
  "destination": "$(param1)",
  "entryPointId": "dd0ece6a-9c2b-4541-9709-0af9bad7c242",
  "attributes": {"customerName": "$(param2)"},
  "outboundType": "EXECUTE_FLOW",
  "mediaType": "telephony"
}
```

Transfer a Skilled Chat to Another Queue

- Custom Integration
  - POST to the Search API
  - Parameters
    - From Time
    - To Time
    - Task ID
    - Global Variable name which the agent updated before the transfer
  - Query will filter on the Task ID between 10 minutes ago and now
    - Returns the value of the Global Variable
      - $.data.task.tasks[0].stringGlobalVariables.value




Request Details

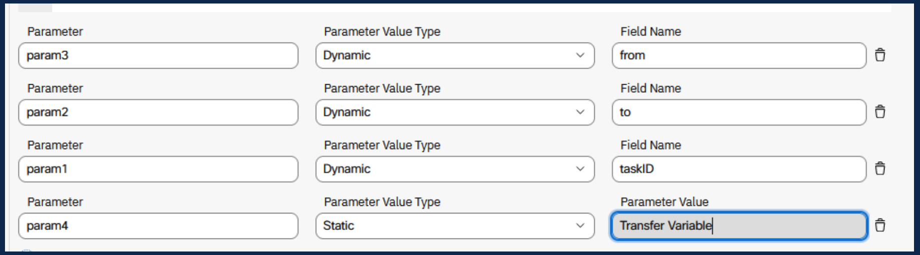
Request Name: lookup skill transfer GV

Request Timeout (Ms): 10000

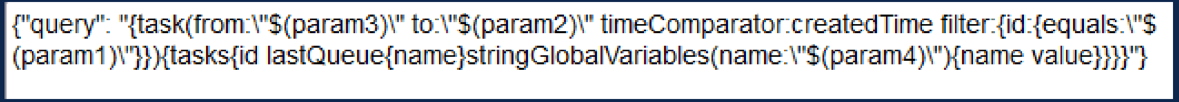
Connection Timeout (Ms): 10000

Type: Post

Resource URL: https://api.wxcc-us1.cisco.com/search

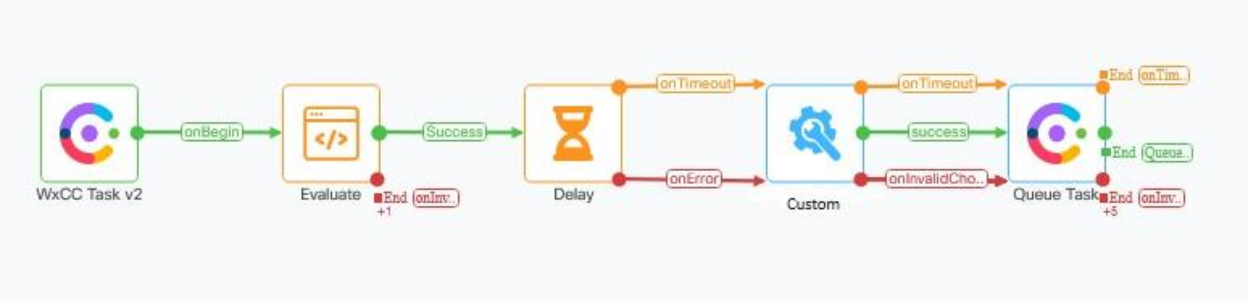


Parameter: param3	Parameter Value Type: Dynamic	Field Name: from
Parameter: param2	Parameter Value Type: Dynamic	Field Name: to
Parameter: param1	Parameter Value Type: Dynamic	Field Name: taskID
Parameter: param4	Parameter Value Type: Static	Parameter Value: Transfer Variable

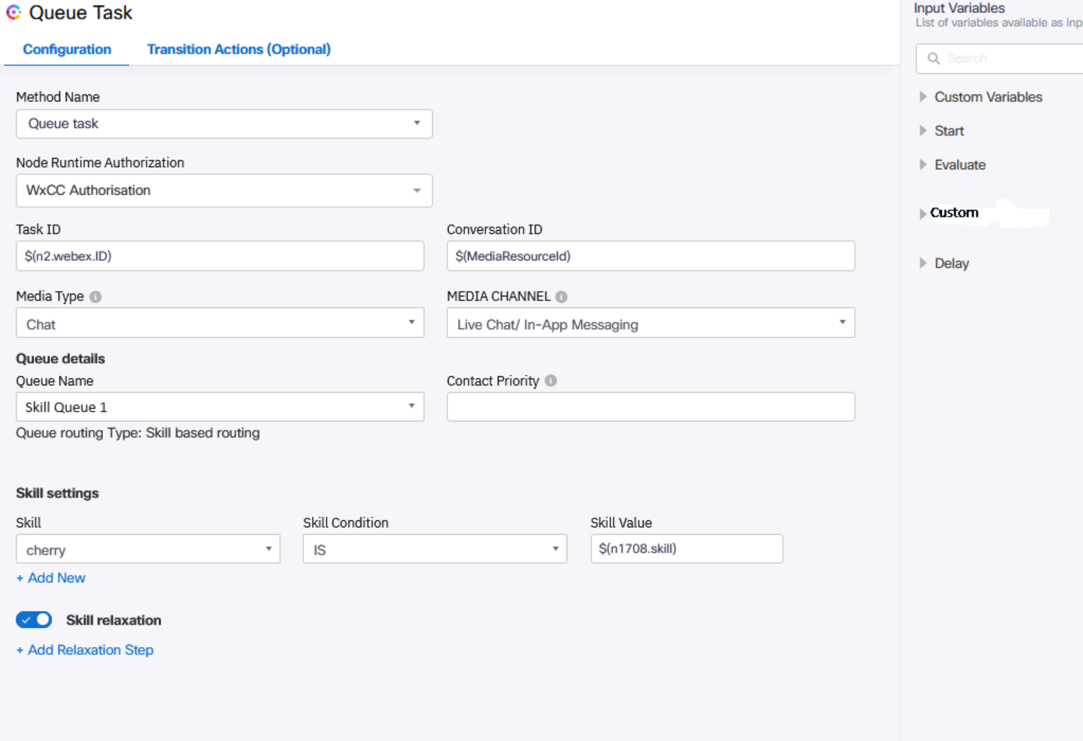


```
{"query": "{task(from:\"$(param3)\" to:\"$(param2)\" timeComparator:createdTime filter:{id:{equals:\"$
(param1)\"}}){tasks{id lastQueue{name}stringGlobalVariables(name:\"$(param4)\"){name value}}}}"}
```

Transfer a Skilled Chat to Another Queue



- Queue Task
  - Use the returned value from the Custom Node’s Search API Query to set the skill value
  - Queue is statically set
  - Task will be delivered using the new skill



Queue Task

Configuration | Transition Actions (Optional)

Method Name: Queue task

Node Runtime Authorization: WxCC Authorisation

Task ID: $(n2.webex.ID)

Conversation ID: $(MediaResourceId)

Media Type: Chat

MEDIA CHANNEL: Live Chat/ In-App Messaging

Queue details

Queue Name: Skill Queue 1

Contact Priority:

Queue routing Type: Skill based routing

Skill settings

Skill	Skill Condition	Skill Value
cherry	IS	$(n1708.skill)

+ Add New

Skill relaxation

+ Add Relaxation Step

Input Variables

List of variables available as inp

How could this be improved?

- Use a Custom Desktop Widget with dropdown values
- Select a different queues based on Global Variables
- Use Skills which are not text skills based on what the agent updated the Global Variable to reflect

Supervisor - Voice Flow changes from Desktop

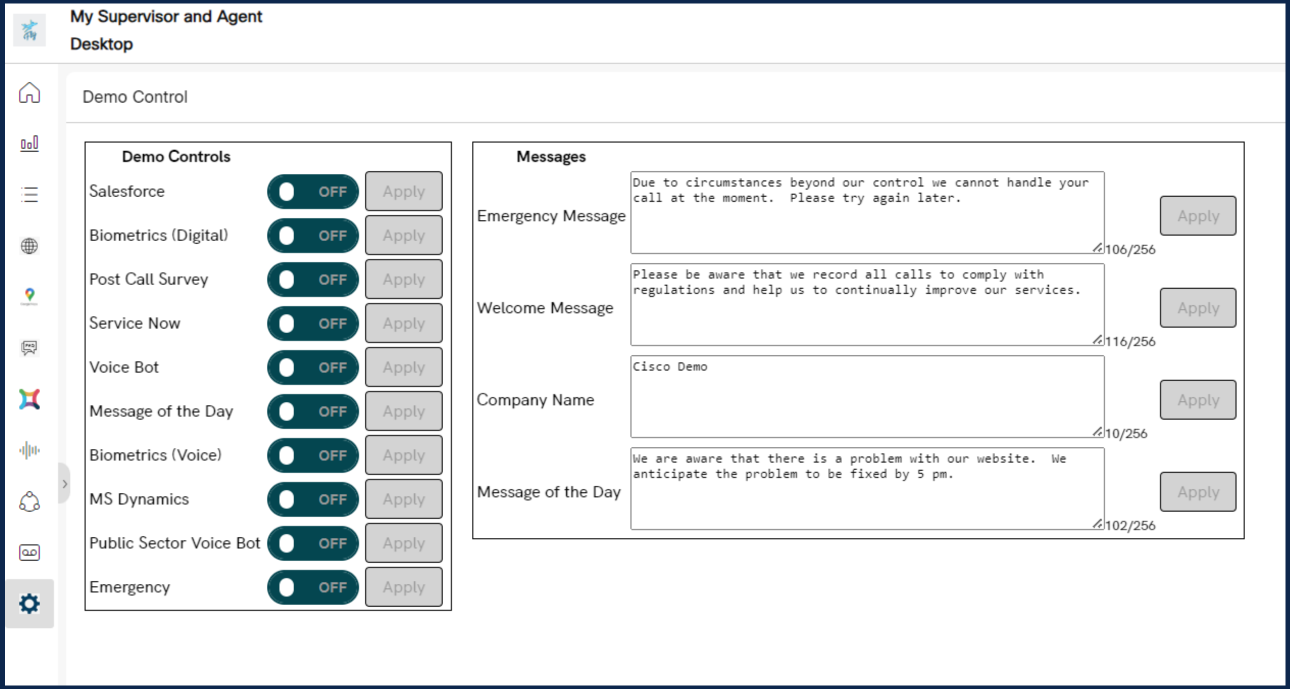
Supervisor - Voice Flow changes from Desktop

Use Case

- Change Messaging including ad-hoc messages
- Put Contact Centre into Emergency Mode
- Control Surveys
- And much more...

Features Used

- Global Variables APIs (read/write)
- (Glitch) webpage
- Supervisor Desktop Layout



My Supervisor and Agent Desktop

Demo Control

Demo Controls

Salesforce	OFF	Apply
Biometrics (Digital)	OFF	Apply
Post Call Survey	OFF	Apply
Service Now	OFF	Apply
Voice Bot	OFF	Apply
Message of the Day	OFF	Apply
Biometrics (Voice)	OFF	Apply
MS Dynamics	OFF	Apply
Public Sector Voice Bot	OFF	Apply
Emergency	OFF	Apply

Messages

Emergency Message	Due to circumstances beyond our control we cannot handle your call at the moment. Please try again later.	106/256	Apply
Welcome Message	Please be aware that we record all calls to comply with regulations and help us to continually improve our services.	116/256	Apply
Company Name	Cisco Demo	10/256	Apply
Message of the Day	We are aware that there is a problem with our website. We anticipate the problem to be fixed by 5 pm.	102/256	Apply

Supervisor Voice Flow changes from Desktop

Standard Functionality

1

Create Global variables (Boolean and String types are currently supported) – naming convention required

2

Add Global Variables to your inbound routing flow

Enhanced Functionality

3

Create Glitch website to run in the Desktop

4

Use WxCC APIs within webpage to read and update global variables used in the Flow.

5

Add webpage to your Agent/Supervisor Desktop profile

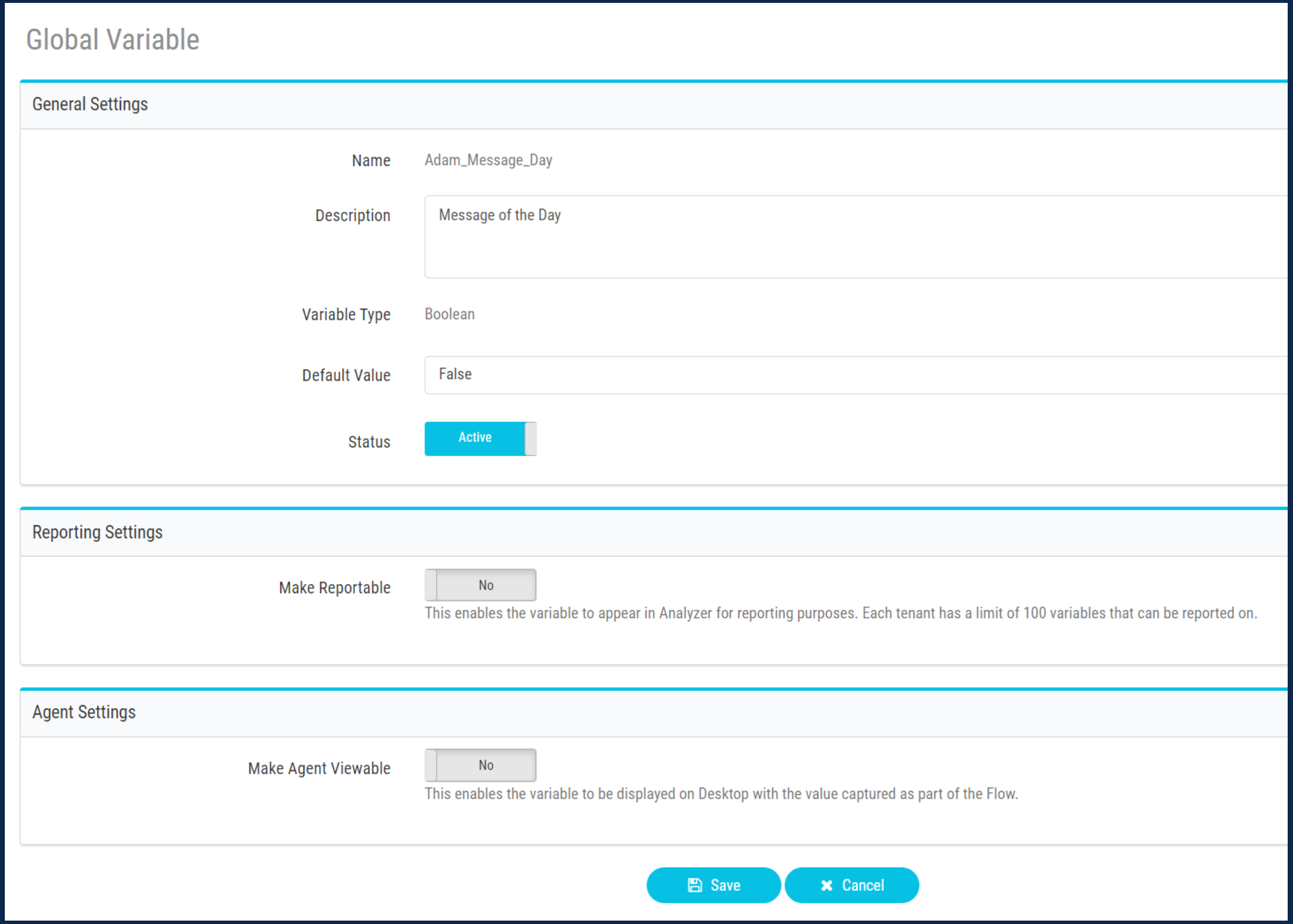
Step 1 – Create Global Variable(s)

Use the WxCC (Legacy) Admin Portal to create either **Boolean** or **String** Global Variables that can be used in your voice flow.

This example will add an ad-hoc message (Message of the Day) to the Voice Flow.

- Prefix your global variables with your name
  - e.g. Adam_Message_Day

- Give your Global Variable a Description – this is what will be shown in the webpage in the Desktop
- Choose either Boolean or String as the type of the Global variable
- Set a default value
- Save
- Repeat with a String Global Variable to contain the actual message to be played.



Global Variable

General Settings

Name	Adam_Message_Day
Description	Message of the Day
Variable Type	Boolean
Default Value	False
Status	Active

Reporting Settings

Make Reportable: No

This enables the variable to appear in Analyzer for reporting purposes. Each tenant has a limit of 100 variables that can be reported on.

Agent Settings

Make Agent Viewable: No

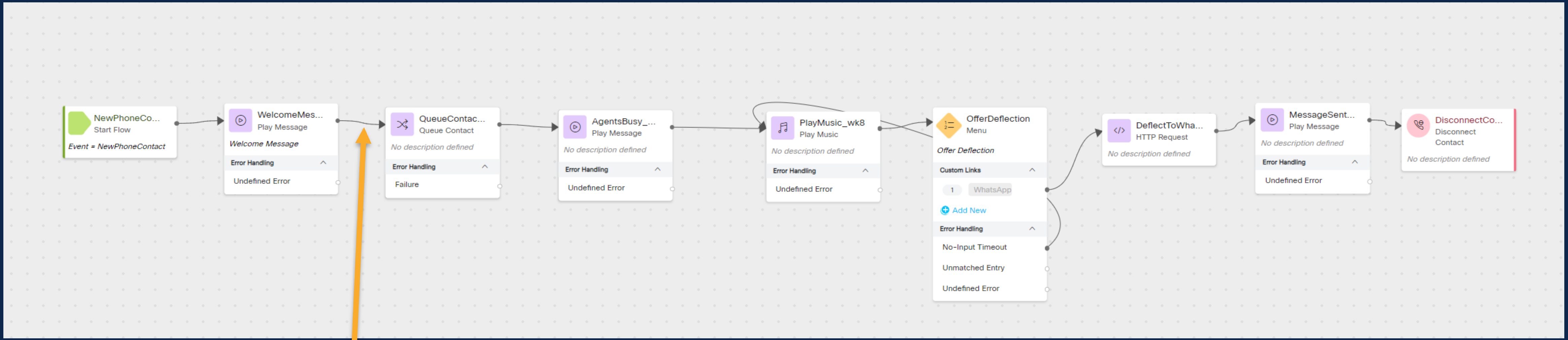
This enables the variable to be displayed on Desktop with the value captured as part of the Flow.

Step 2 – Add Global Variables to your Voice Flow

Use the WxCC Flow Editor to add the Global Variable to your Voice Flow.

We will build on the Call Deflection Flow from the previous example.

Add the Global variable before the queue node. (Note: you can add the Global Variable anywhere that is relevant)

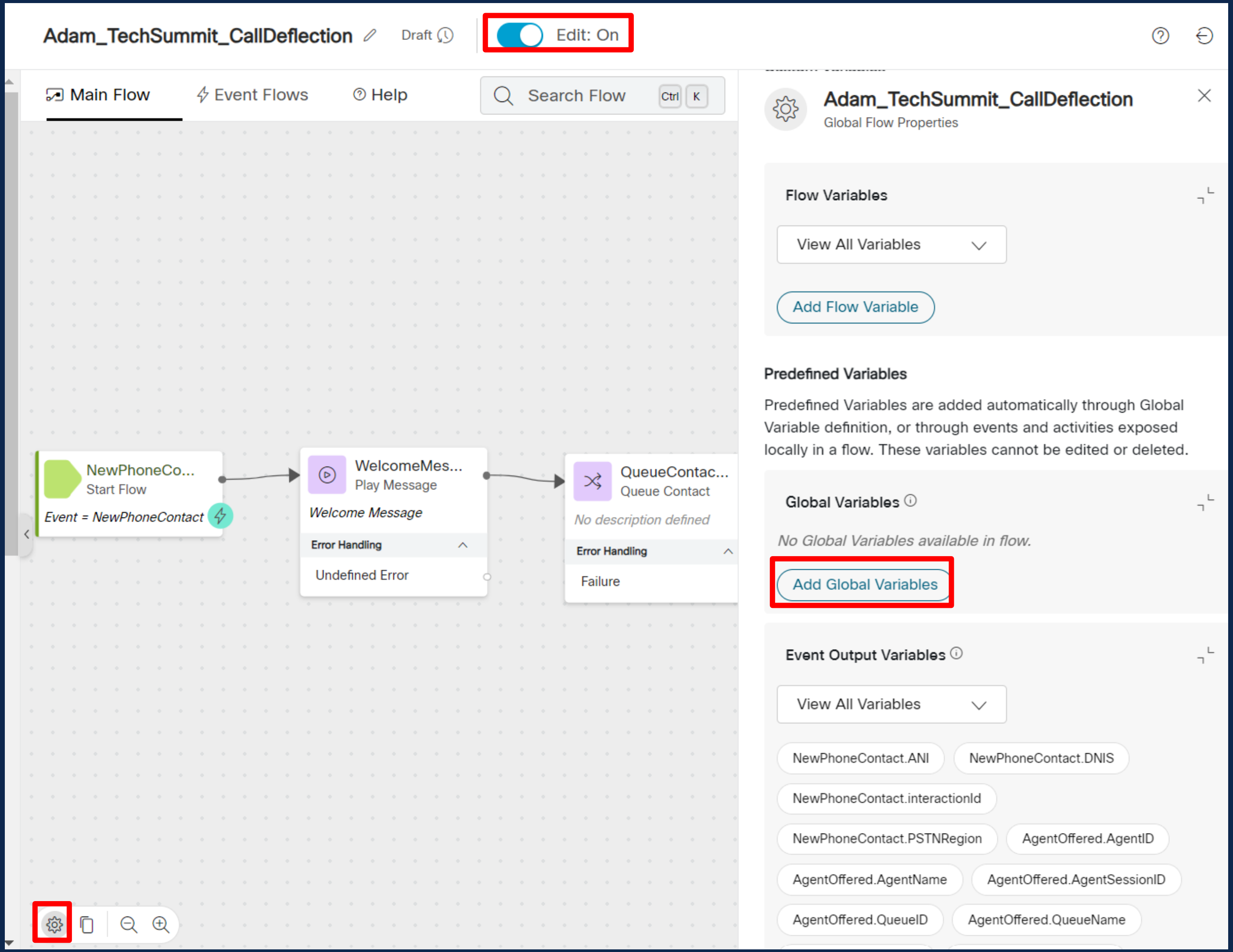


We will place the Global Variable here

Step 2 – Add Global Variables to your Voice Flow

From Flow Designer

- Edit your flow.
- Select the **Global Flow Properties** Cog
- Scroll down to Global Variables and select Add Global Variable

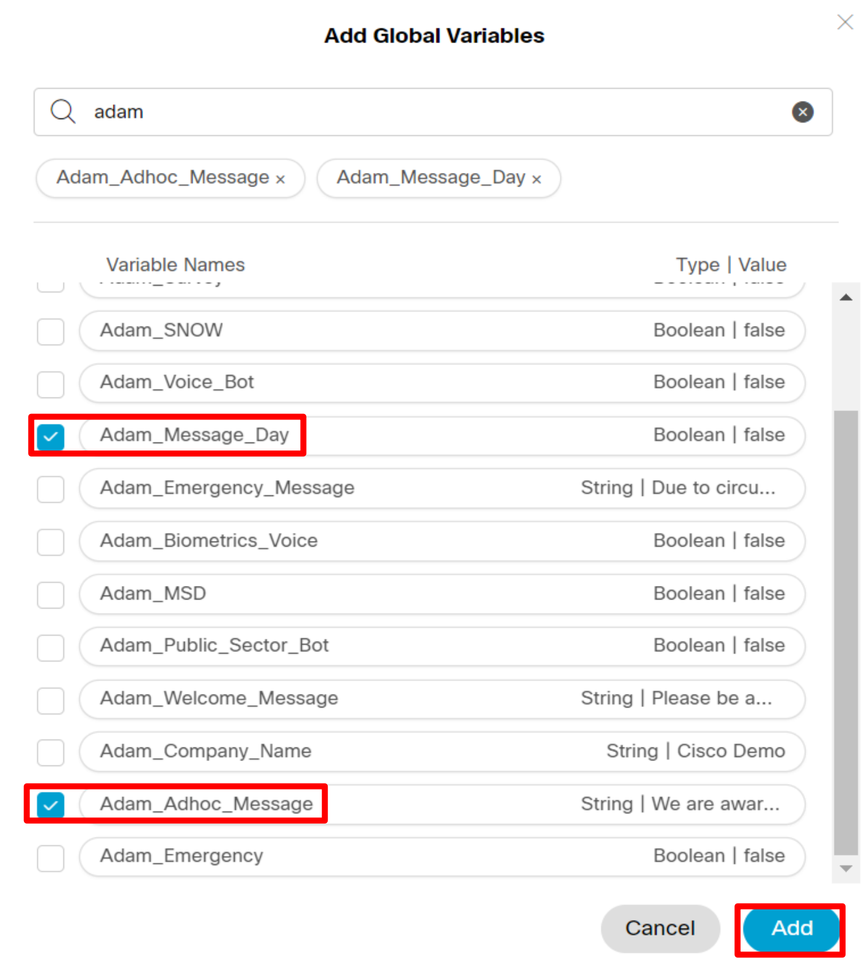


Predefined Variables are added automatically through Global Variable definition, or through events and activities exposed locally in a flow. These variables cannot be edited or deleted.

Step 2 – Add Global Variables to your Voice Flow

From Flow Designer

- Select the Global Variables added above
- Select Add



Add Global Variables

adam

Adam_Adhoc_Message × Adam_Message_Day ×

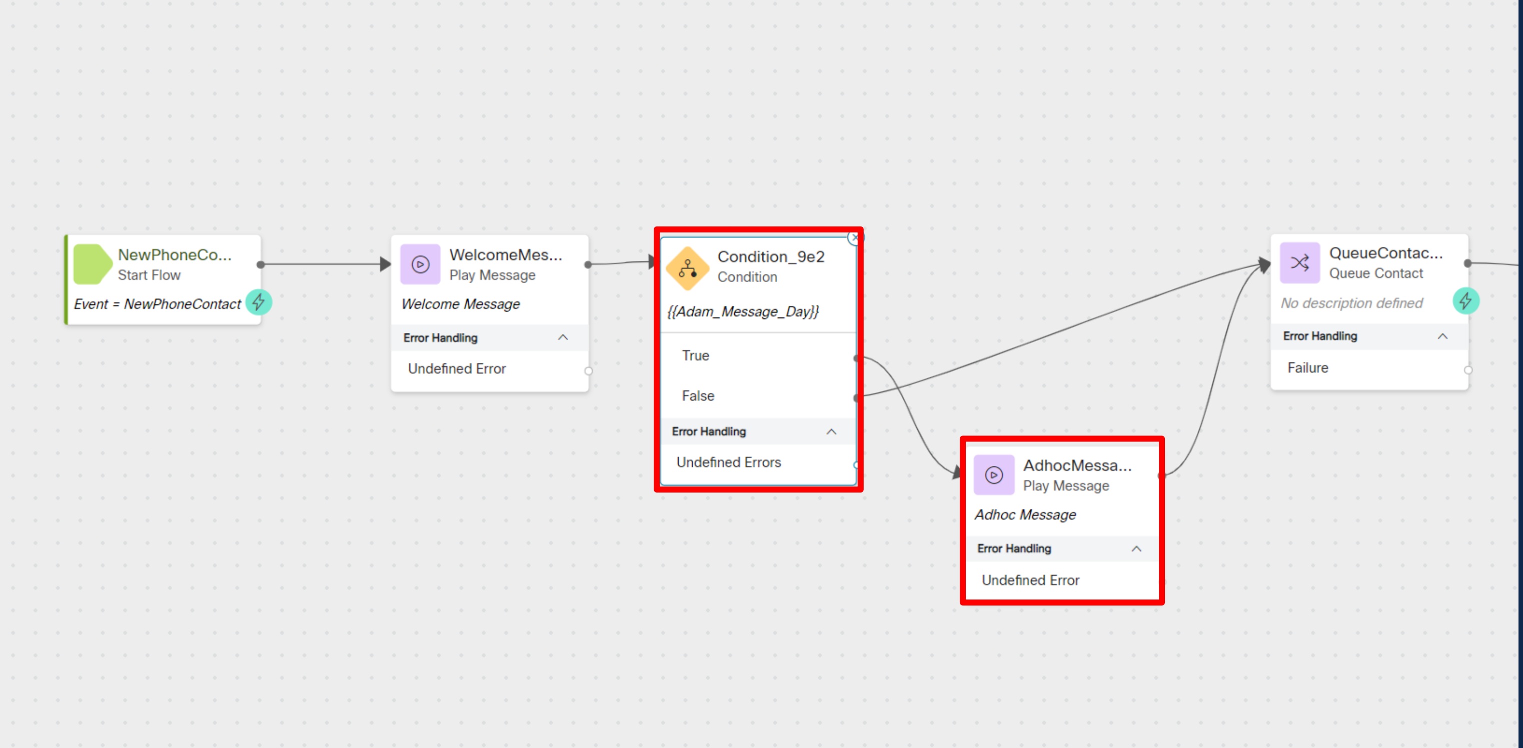
	Variable Names	Type | Value
□	Adam_SNOW	Boolean | false
□	Adam_Voice_Bot	Boolean | false
☑	Adam_Message_Day	Boolean | false
□	Adam_Emergency_Message	String | Due to circu...
□	Adam_Biometrics_Voice	Boolean | false
□	Adam_MSD	Boolean | false
□	Adam_Public_Sector_Bot	Boolean | false
□	Adam_Welcome_Message	String | Please be a...
□	Adam_Company_Name	String | Cisco Demo
☑	Adam_Adhoc_Message	String | We are awar...
□	Adam_Emergency	Boolean | false

Step 2 – Add Global Variables to your Voice Flow

From Flow Designer

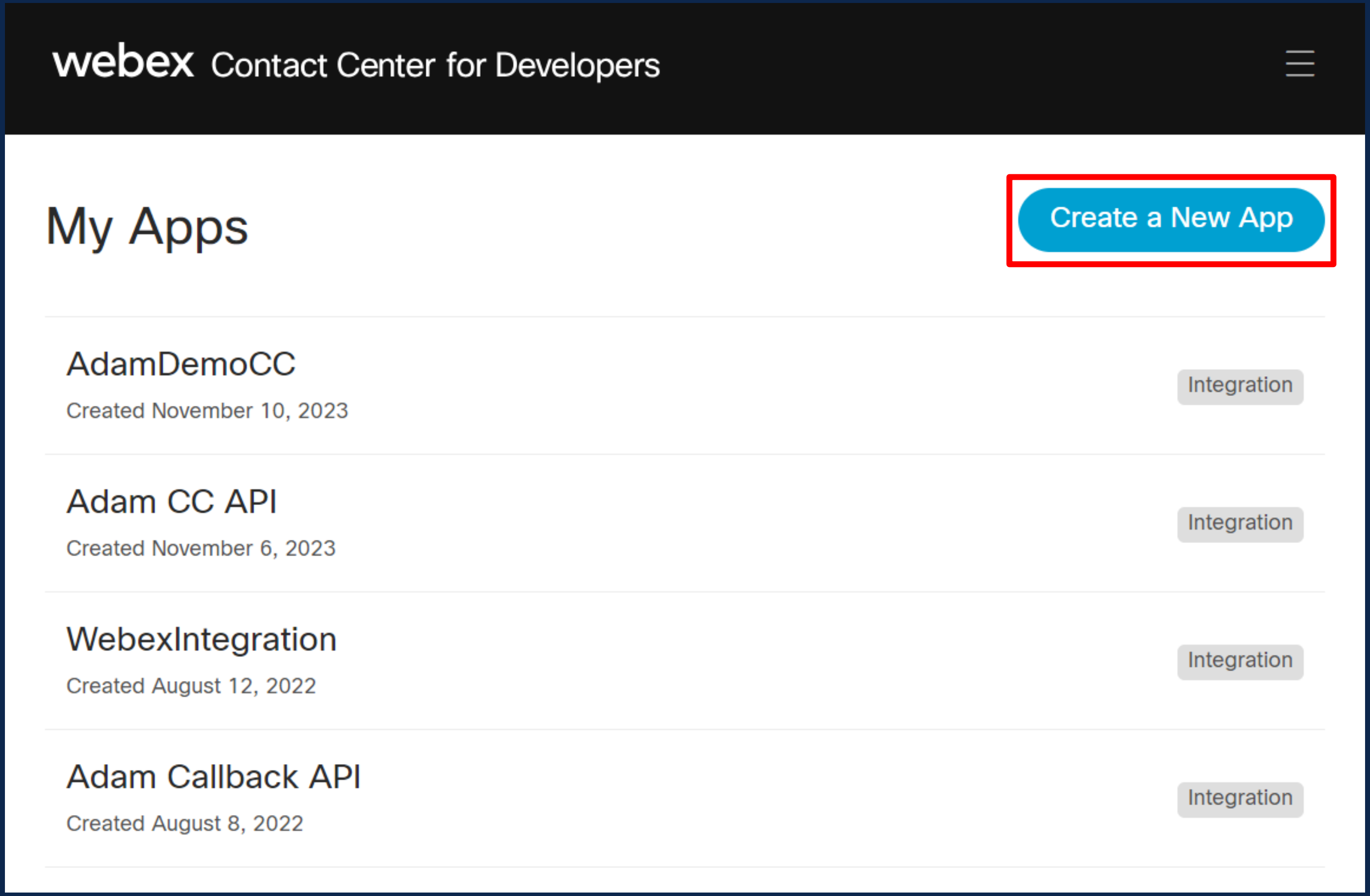
- Now, add to the main Flow:
- Condition
  - {{Adam_Message Day}}
- Play Message
  - If Condition True
  - {{Adam_Adhoc_Message}}

- Validate and Publish



Step 3 – Use WxCC APIs to read/write Global Variables

- From developer.webex-cx.com, create a new WxCC API App
  - Secret
  - Client Id
- Create an Access Token mechanism to provide access token in your website
  - Details available in Cisco github repository.

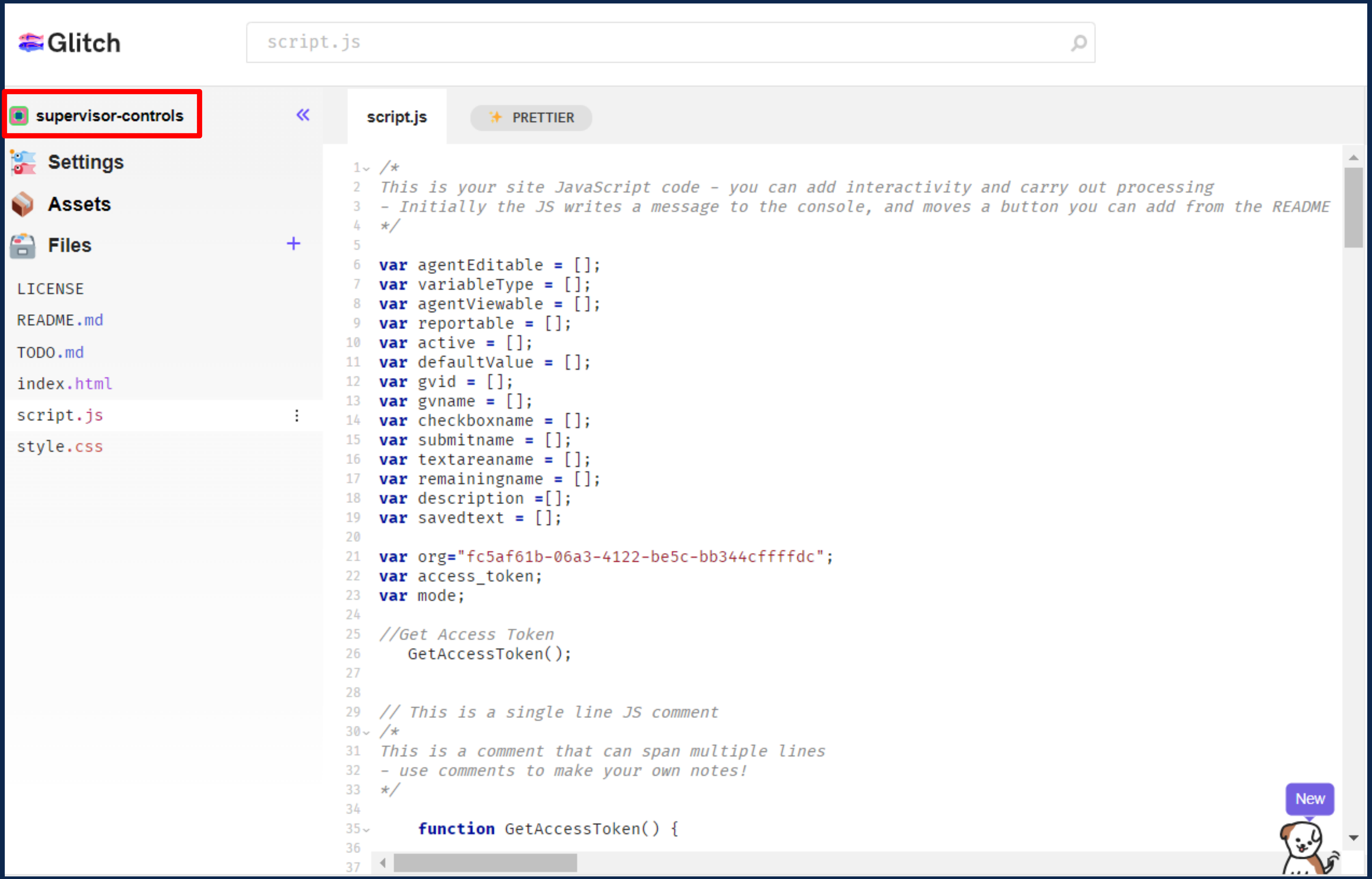


My Apps

AdamDemoCC Created November 10, 2023	Integration
Adam CC API Created November 6, 2023	Integration
WebexIntegration Created August 12, 2022	Integration
Adam Callback API Created August 8, 2022	Integration

Step 4 – Create (Glitch) website to enable view/update Global Variables

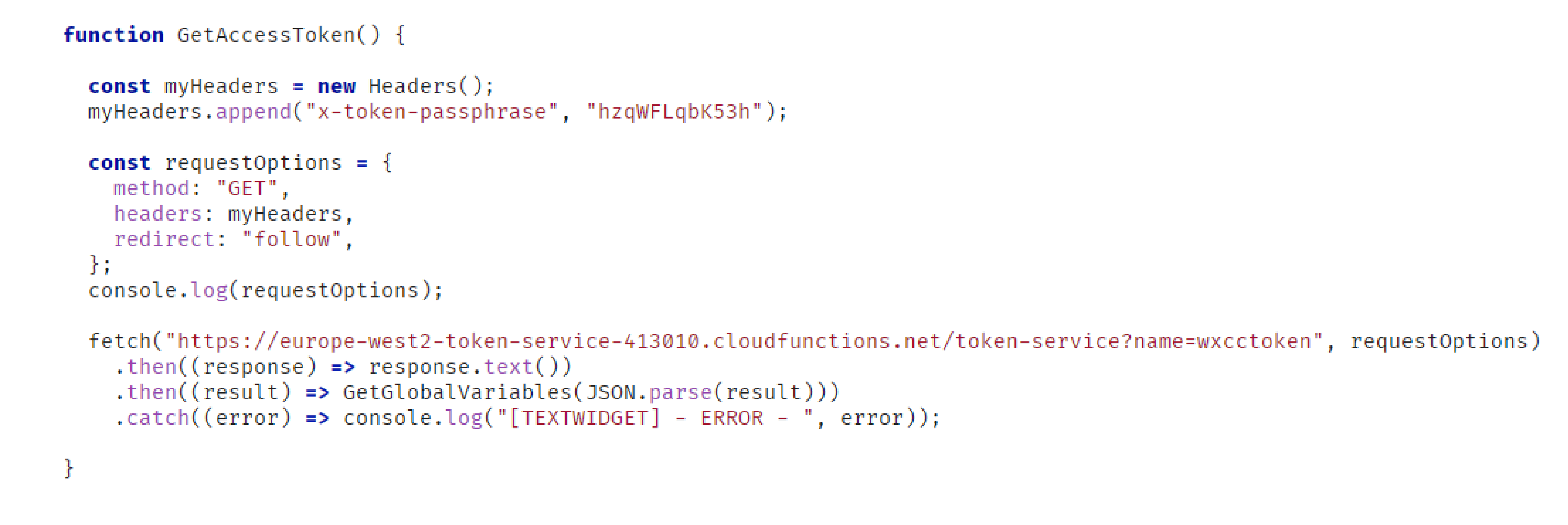
- Browse to Glitch.com
- Create a new Project
- Optional (rename project to something other than the random 3 words that Glitch defaults to)



```
/*
This is your site JavaScript code - you can add interactivity and carry out processing
- Initially the JS writes a message to the console, and moves a button you can add from the README
*/

var agentEditable = [];
var variableType = [];
var agentViewable = [];
var reportable = [];
var active = [];
var defaultValue = [];
var gvid = [];
var gvname = [];
var checkboxname = [];
var submitname = [];
var textareaname = [];
var remainingname = [];
var description =[];
var savedtext = [];

var org="fc5af61b-06a3-4122-be5c-bb344cffffdc";
var access_token;
var mode;

//Get Access Token
  GetAccessToken();


// This is a single line JS comment
/*
This is a comment that can span multiple lines
- use comments to make your own notes!
*/

    function GetAccessToken() {
```

Step 4 – Use WxCC Apis to read/write Global Variables

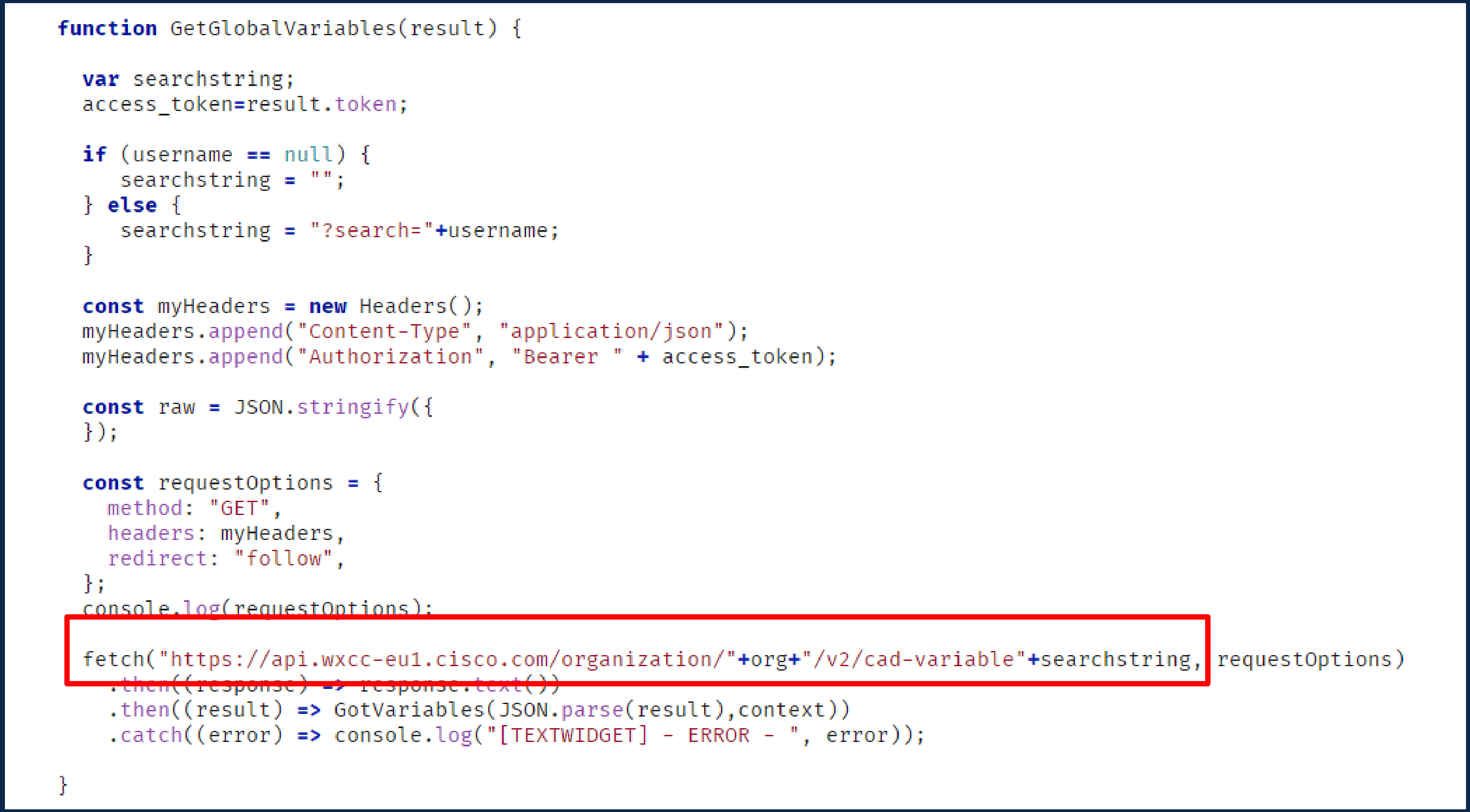
- From webpage javascript file, use the APIs to list the current status of the global variables.
- Get an access token



```
function GetAccessToken() {

  const myHeaders = new Headers();
  myHeaders.append("x-token-passphrase", "hzqWFLqbK53h");

  const requestOptions = {
    method: "GET",
    headers: myHeaders,
    redirect: "follow",
  };
  console.log(requestOptions);

  fetch("https://europe-west2-token-service-413010.cloudfunctions.net/token-service?name=wxcctoken", requestOptions)
    .then((response) => response.text())
    .then((result) => GetGlobalVariables(JSON.parse(result)))
    .catch((error) => console.log("[TEXTWIDGET] - ERROR - ", error));

}
```

Step 4 – Use WxCC Apis to read/write Global Variables

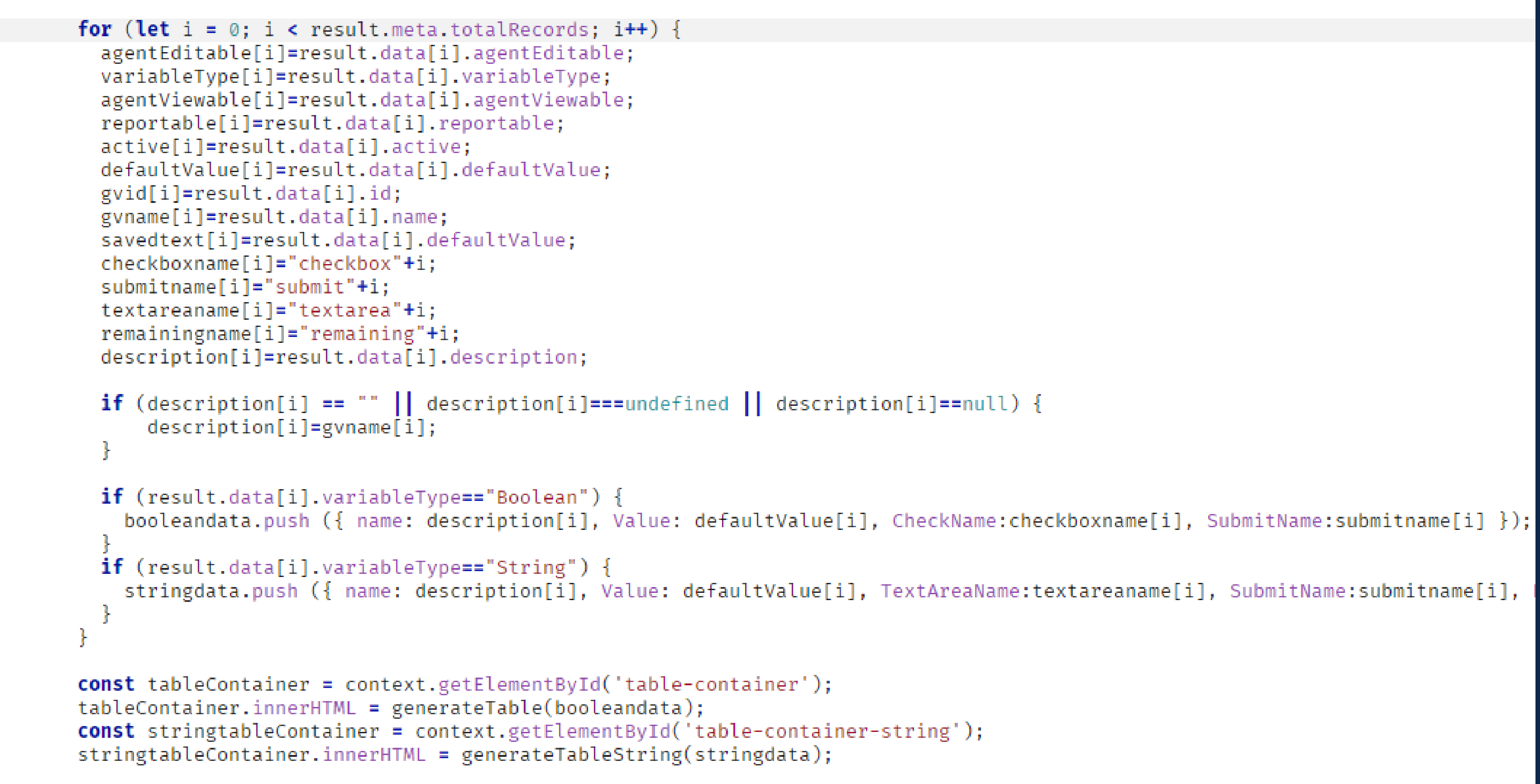
- Use the Global Variables API to retrieve a list of all Global variables with the prefix added to the search.



```
function GetGlobalVariables(result) {

  var searchstring;
  access_token=result.token;

  if (username == null) {
     searchstring = "";
  } else {
     searchstring = "?search="+username;
  }

  const myHeaders = new Headers();
  myHeaders.append("Content-Type", "application/json");
  myHeaders.append("Authorization", "Bearer " + access_token);

  const raw = JSON.stringify({
  });

  const requestOptions = {
    method: "GET",
    headers: myHeaders,
    redirect: "follow",
  };
  console.log(requestOptions);

  fetch("https://api.wxcc-eu1.cisco.com/organization/"+org+"/v2/cad-variable"+searchstring, requestOptions)
    .then((response) => response.text())
    .then((result) => GotVariables(JSON.parse(result),context))
    .catch((error) => console.log("[TEXTWIDGET] - ERROR - ", error));

}
```

Step 4 – Use WxCC Apis to read/write Global Variables

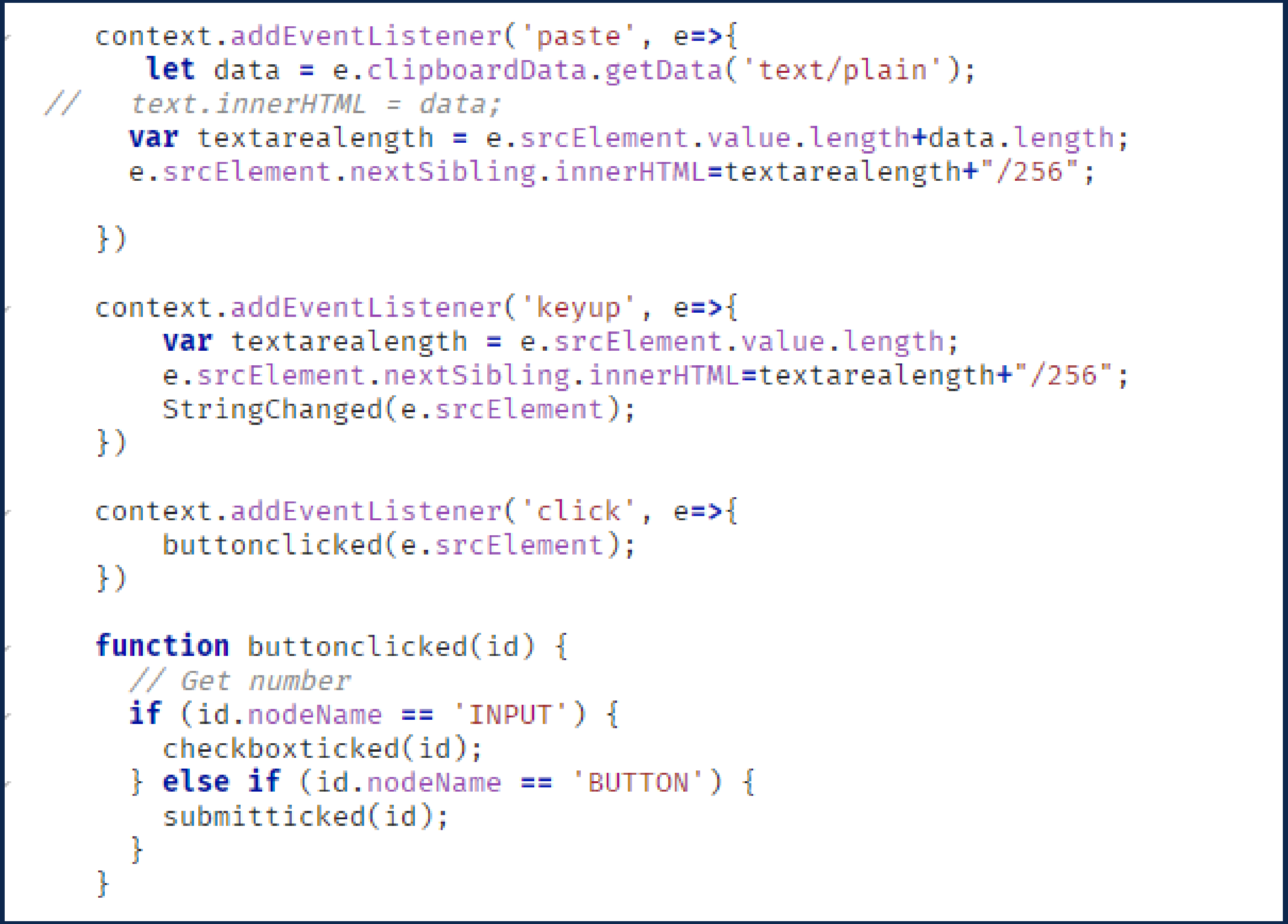
- Parse the results of the search to create an HTML table of results



```
for (let i = 0; i < result.meta.totalRecords; i++) {
  agentEditable[i]=result.data[i].agentEditable;
  variableType[i]=result.data[i].variableType;
  agentViewable[i]=result.data[i].agentViewable;
  reportable[i]=result.data[i].reportable;
  active[i]=result.data[i].active;
  defaultValue[i]=result.data[i].defaultValue;
  gvid[i]=result.data[i].id;
  gvname[i]=result.data[i].name;
  savedtext[i]=result.data[i].defaultValue;
  checkboxname[i]="checkbox"+i;
  submitname[i]="submit"+i;
  textareaname[i]="textarea"+i;
  remainingname[i]="remaining"+i;
  description[i]=result.data[i].description;

  if (description[i] == "" || description[i]===undefined || description[i]==null) {
      description[i]=gvname[i];
  }

  if (result.data[i].variableType=="Boolean") {
    booleandata.push ({ name: description[i], Value: defaultValue[i], CheckName:checkboxname[i], SubmitName:submitname[i] });
  }
  if (result.data[i].variableType=="String") {
    stringdata.push ({ name: description[i], Value: defaultValue[i], TextAreaName:textareaname[i], SubmitName:submitname[i],
  }
}

const tableContainer = context.getElementById('table-container');
tableContainer.innerHTML = generateTable(booleandata);
const stringtableContainer = context.getElementById('table-container-string');
stringtableContainer.innerHTML = generateTableString(stringdata);
```

Step 4 – Use WxCC Apis to read/write Global Variables

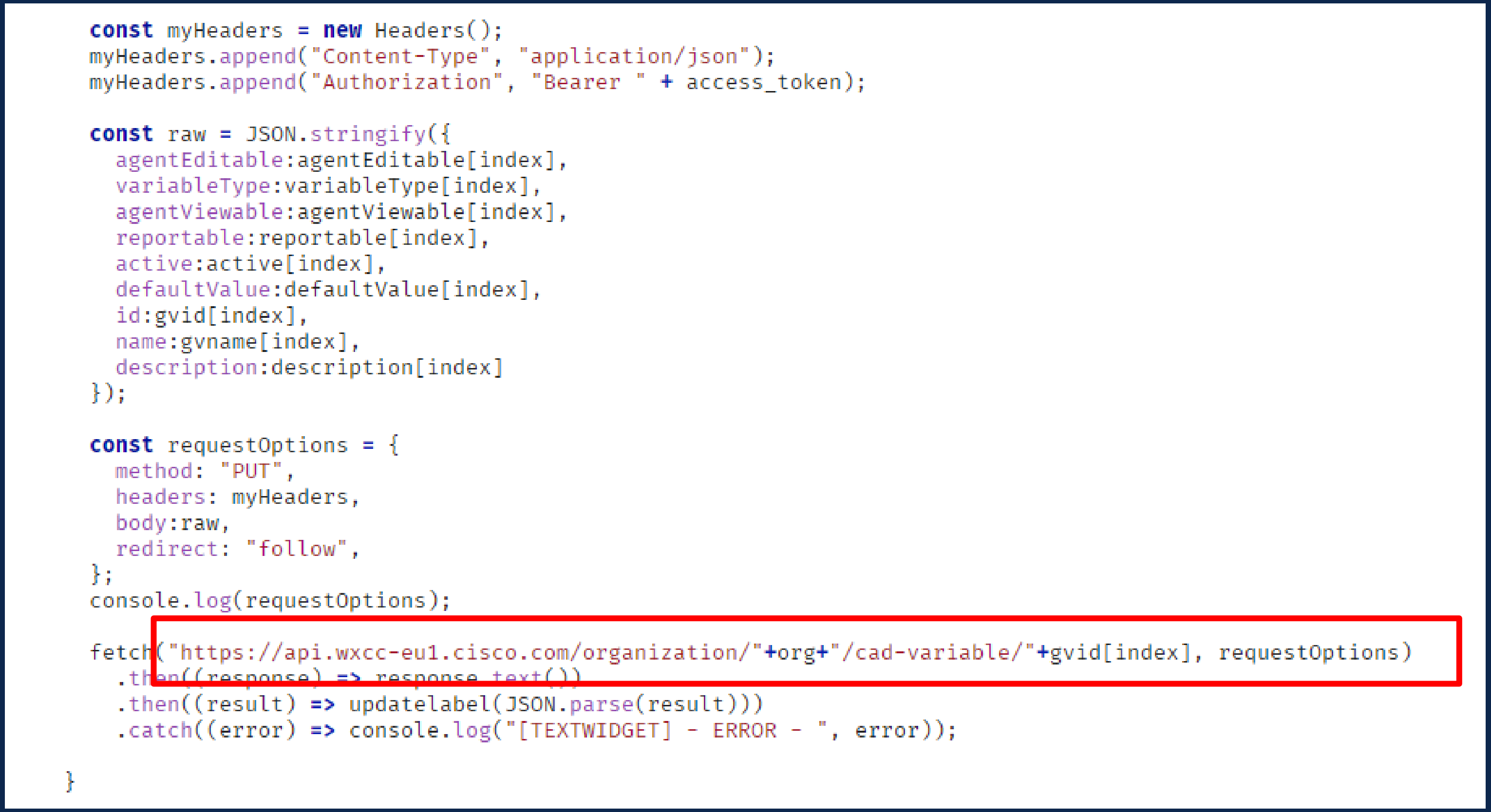
- Add some listeners



```
    context.addEventListener('paste', e=>{
       let data = e.clipboardData.getData('text/plain');
//     text.innerHTML = data;
     var textarealength = e.srcElement.value.length+data.length;
     e.srcElement.nextSibling.innerHTML=textarealength+"/256";

    })

    context.addEventListener('keyup', e=>{
      var textarealength = e.srcElement.value.length;
      e.srcElement.nextSibling.innerHTML=textarealength+"/256";
      StringChanged(e.srcElement);
    })

    context.addEventListener('click', e=>{
      buttonclicked(e.srcElement);
    })

    function buttonclicked(id) {
     // Get number
     if (id.nodeName == 'INPUT') {
      checkboxticked(id);
     } else if (id.nodeName == 'BUTTON') {
      submitticked(id);
     }
    }
```

Step 4 – Use WxCC Apis to read/write Global Variables

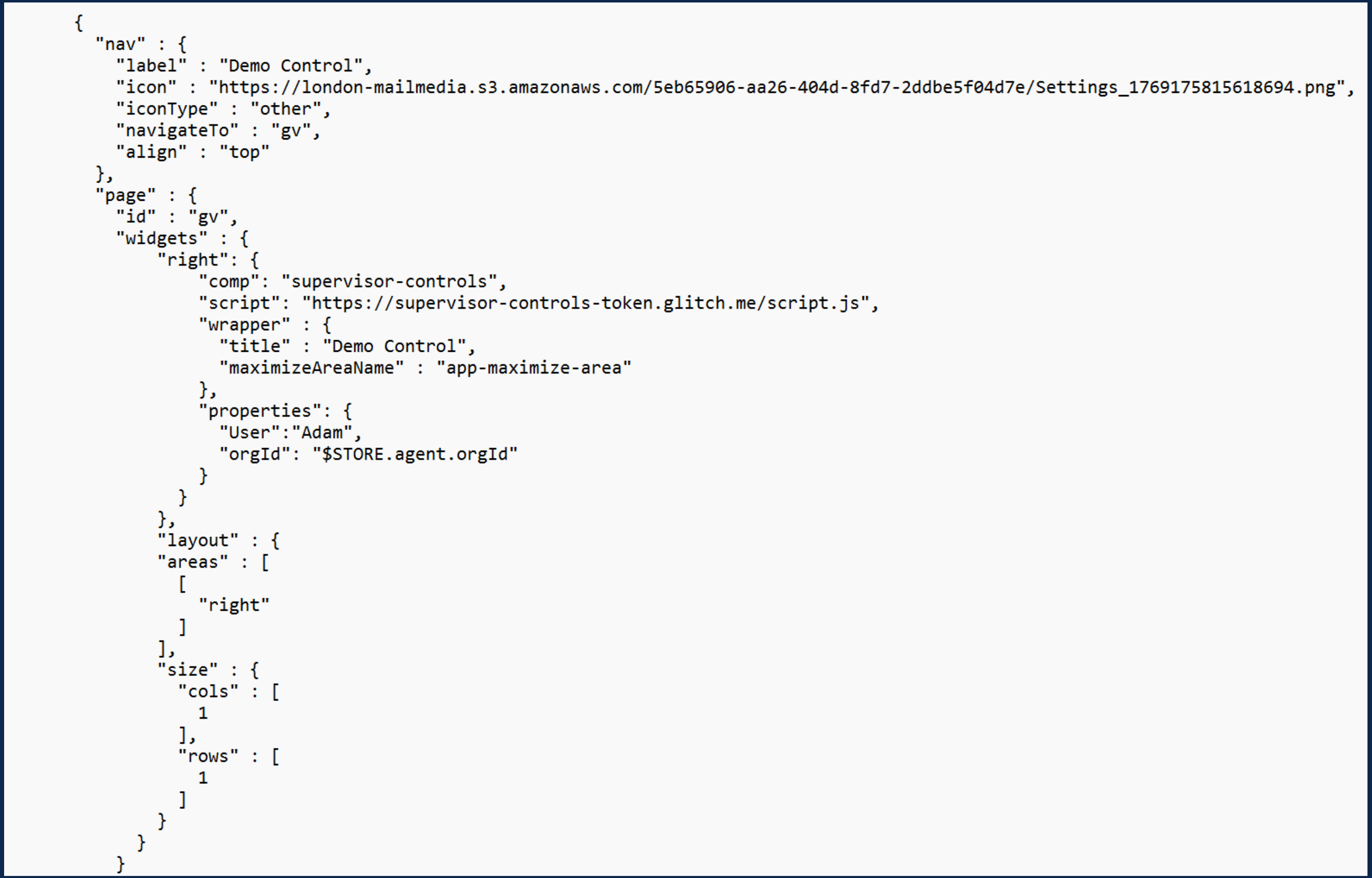
- If anything changes and the Apply button is clicked, use the Set Global variable API to update the Global Variable.



```
    const myHeaders = new Headers();
    myHeaders.append("Content-Type", "application/json");
    myHeaders.append("Authorization", "Bearer " + access_token);

    const raw = JSON.stringify({
      agentEditable:agentEditable[index],
      variableType:variableType[index],
      agentViewable:agentViewable[index],
      reportable:reportable[index],
      active:active[index],
      defaultValue:defaultValue[index],
      id:gvid[index],
      name:gvname[index],
      description:description[index]
    });

    const requestOptions = {
      method: "PUT",
      headers: myHeaders,
      body:raw,
      redirect: "follow",
    };
    console.log(requestOptions);

    fetch("https://api.wxcc-eu1.cisco.com/organization/"+org+"/cad-variable/"+gvid[index], requestOptions)
      .then((response) => response.text())
      .then((result) => updatelabel(JSON.parse(result)))
      .catch((error) => console.log("[TEXTWIDGET] - ERROR - ", error));

  }
```

Step 5 – Add webpage to Supervisor Desktop

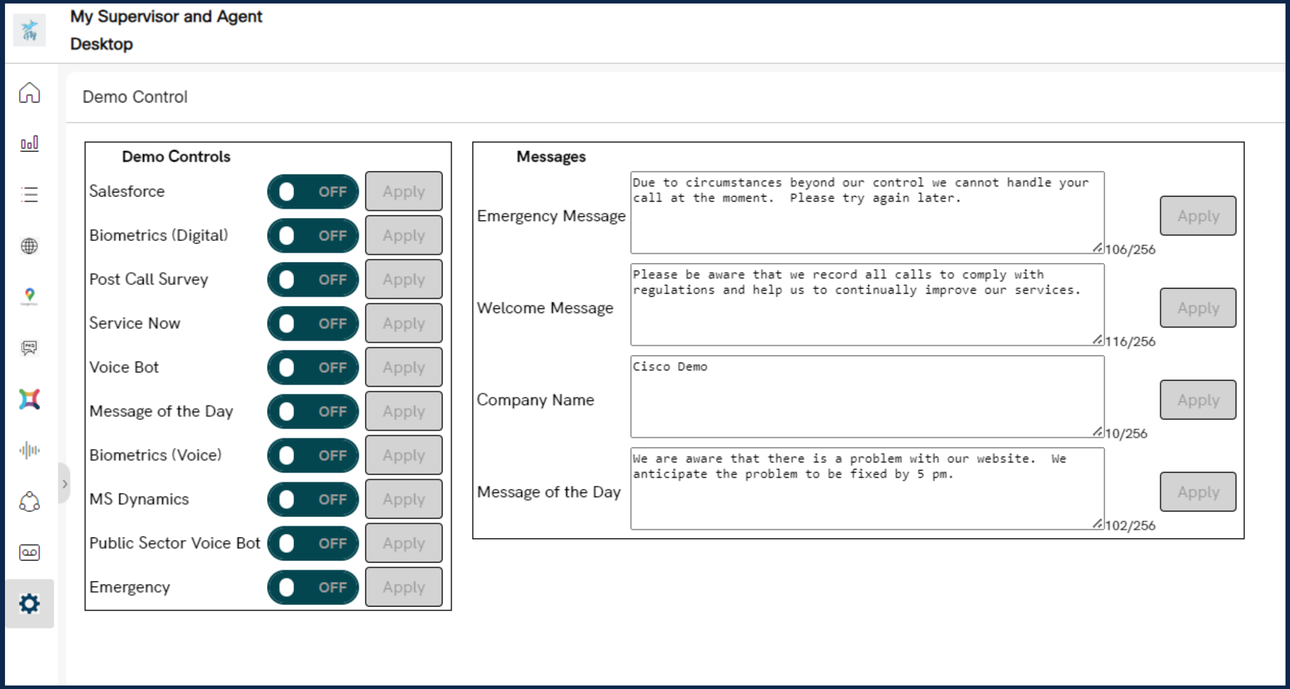
- Edit Desktop JSON File
- Add the following code to the **Navigation** section
- Use the User and orgId parameters to pass data to the script.
- Save and use Control Hub to upload to your Desktop Profile



```
{
  "nav" : {
    "label" : "Demo Control",
    "icon" : "https://london-mailmedia.s3.amazonaws.com/5eb65906-aa26-404d-8fd7-2ddbe5f04d7e/Settings_1769175815618694.png",
    "iconType" : "other",
    "navigateTo" : "gv",
    "align" : "top"
  },
  "page" : {
    "id" : "gv",
    "widgets" : {
        "right": {
            "comp": "supervisor-controls",
            "script": "https://supervisor-controls-token.glitch.me/script.js",
            "wrapper" : {
              "title" : "Demo Control",
              "maximizeAreaName" : "app-maximize-area"
            },
            "properties": {
              "User":"Adam",
              "orgId": "$STORE.agent.orgId"
            }
          }
        },
        "layout" : {
        "areas" : [
          [
            "right"
          ]
        ],
        "size" : {
          "cols" : [
            1
          ],
          "rows" : [
            1
          ]
        }
      }
    }
```

Supervisor - Voice Flow changes from Desktop

- Refresh or log in as your supervisor. You should see a cog on the Navigation bar.
- Select the cog and you will see your Global Variables created above.



My Supervisor and Agent Desktop

Demo Control

Demo Controls

Salesforce	OFF	Apply
Biometrics (Digital)	OFF	Apply
Post Call Survey	OFF	Apply
Service Now	OFF	Apply
Voice Bot	OFF	Apply
Message of the Day	OFF	Apply
Biometrics (Voice)	OFF	Apply
MS Dynamics	OFF	Apply
Public Sector Voice Bot	OFF	Apply
Emergency	OFF	Apply

Messages

Emergency Message	Due to circumstances beyond our control we cannot handle your call at the moment. Please try again later.	106/256	Apply
Welcome Message	Please be aware that we record all calls to comply with regulations and help us to continually improve our services.	116/256	Apply
Company Name	Cisco Demo	10/256	Apply
Message of the Day	We are aware that there is a problem with our website. We anticipate the problem to be fixed by 5 pm.	102/256	Apply

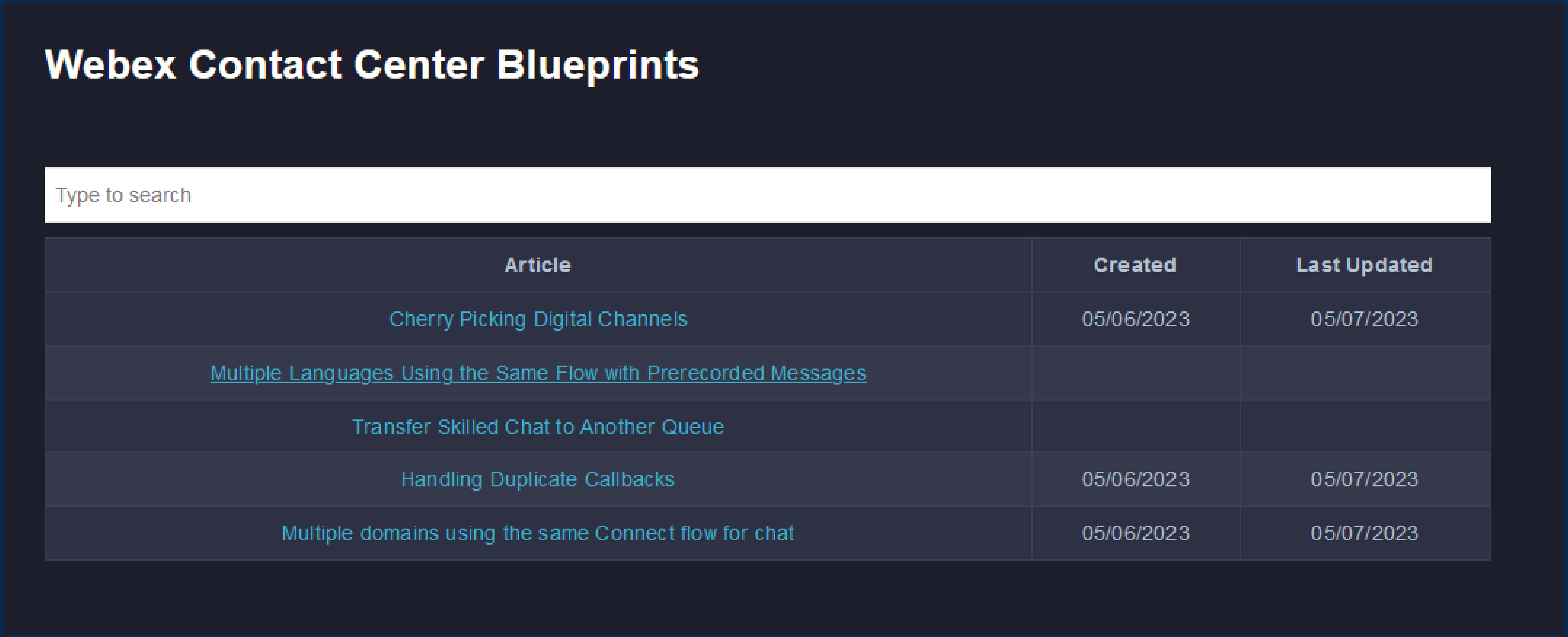
What solutions do you want to create?

ⓘ Start presenting to display the poll results on this slide.

Webex Contact Center Blueprints

A place to find the solutions to your challenges

Searchable



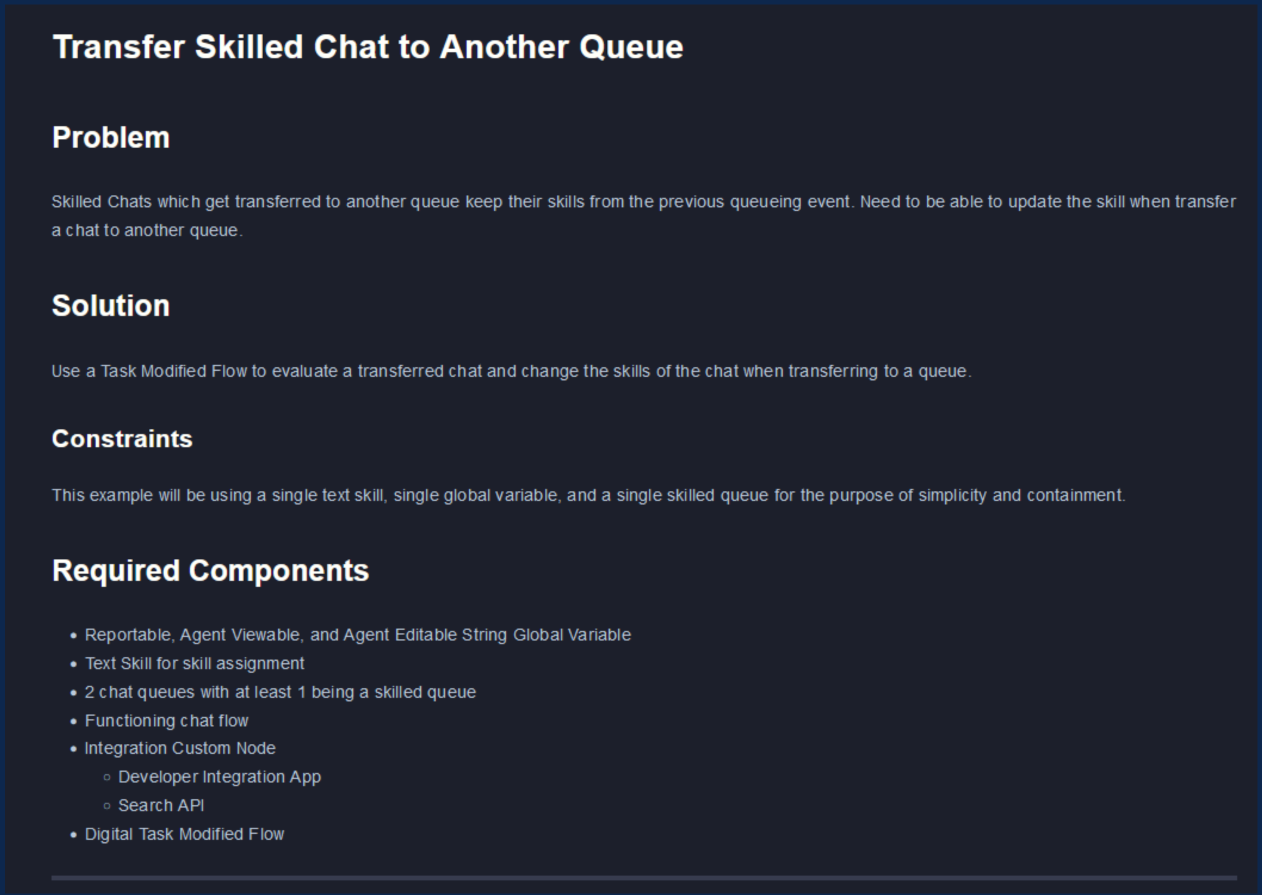
Webex Contact Center Blueprints

Type to search

Article	Created	Last Updated
Cherry Picking Digital Channels	05/06/2023	05/07/2023
Multiple Languages Using the Same Flow with Prerecorded Messages		
Transfer Skilled Chat to Another Queue		
Handling Duplicate Callbacks	05/06/2023	05/07/2023
Multiple domains using the same Connect flow for chat	05/06/2023	05/07/2023

Use Case Based

- Problem
  - What we are trying to solve
- Solution
  - High level summary of how we are going to solve the problem
- Constraints
  - The parameters we worked with to create the solution
- Required Components
  - The ingredients in the recipe



Transfer Skilled Chat to Another Queue

Problem

Skilled Chats which get transferred to another queue keep their skills from the previous queueing event. Need to be able to update the skill when transfer a chat to another queue.

Solution

Use a Task Modified Flow to evaluate a transferred chat and change the skills of the chat when transferring to a queue.

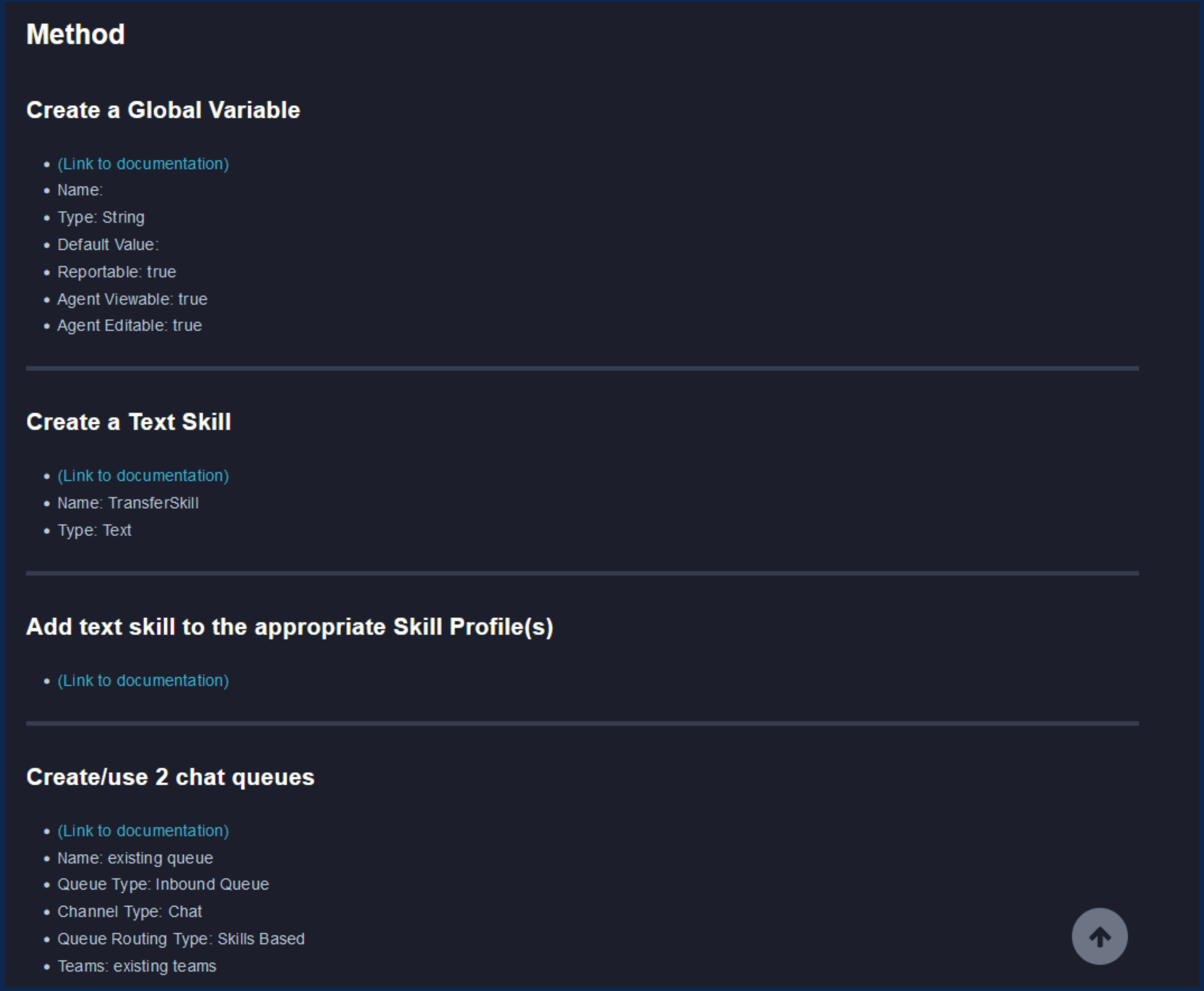
Constraints

This example will be using a single text skill, single global variable, and a single skilled queue for the purpose of simplicity and containment.

Required Components

- Reportable, Agent Viewable, and Agent Editable String Global Variable
- Text Skill for skill assignment
- 2 chat queues with at least 1 being a skilled queue
- Functioning chat flow
- Integration Custom Node
  - Developer Integration App
  - Search API
- Digital Task Modified Flow

Method – How did we put the parts together



Method

Create a Global Variable

- (Link to documentation)
- Name:
- Type: String
- Default Value:
- Reportable: true
- Agent Viewable: true
- Agent Editable: true

Create a Text Skill

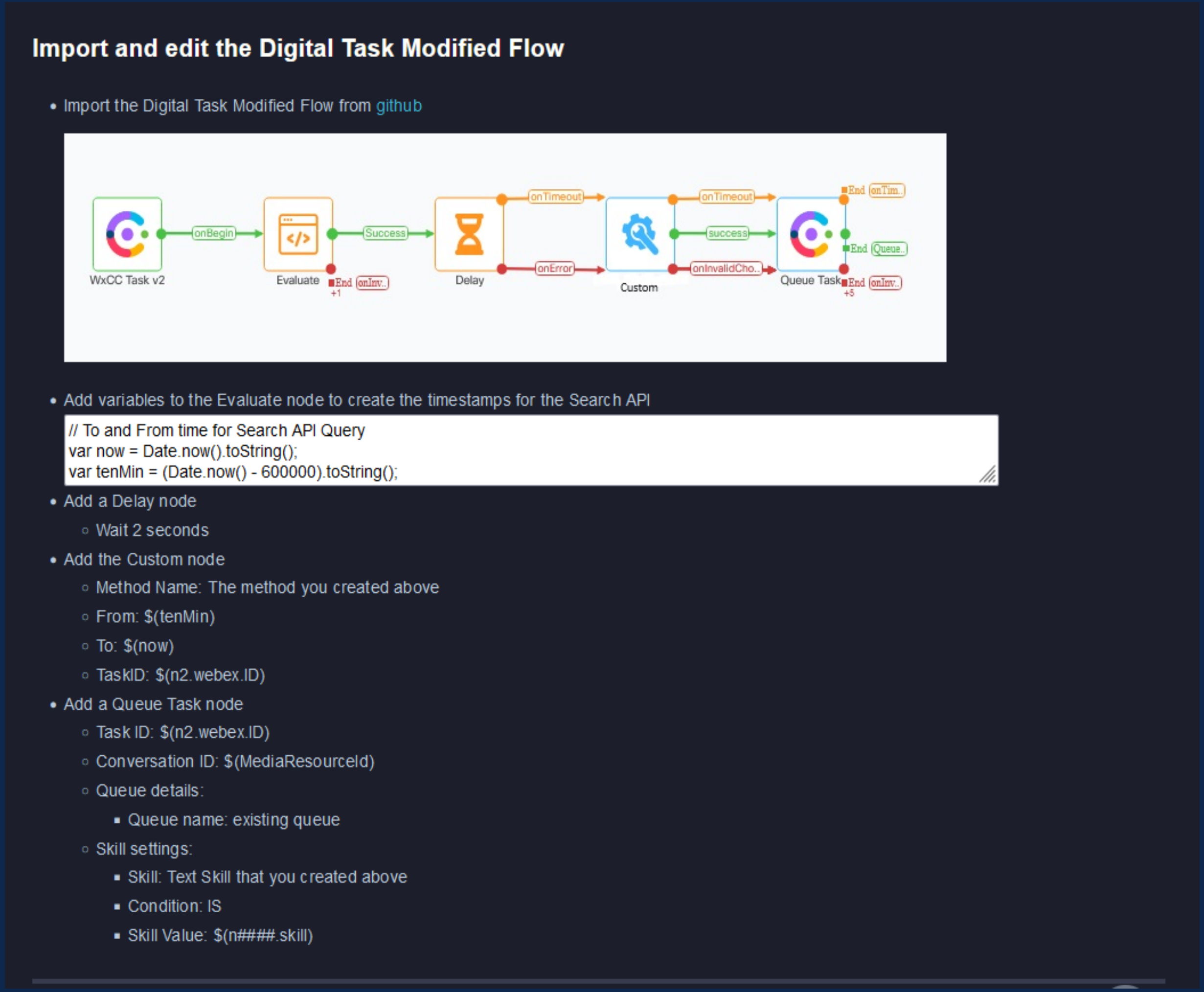
- (Link to documentation)
- Name: TransferSkill
- Type: Text

Add text skill to the appropriate Skill Profile(s)

- (Link to documentation)

Create/use 2 chat queues

- (Link to documentation)
- Name: existing queue
- Queue Type: Inbound Queue
- Channel Type: Chat
- Queue Routing Type: Skills Based
- Teams: existing teams



Import and edit the Digital Task Modified Flow

- Import the Digital Task Modified Flow from github

- Add variables to the Evaluate node to create the timestamps for the Search API

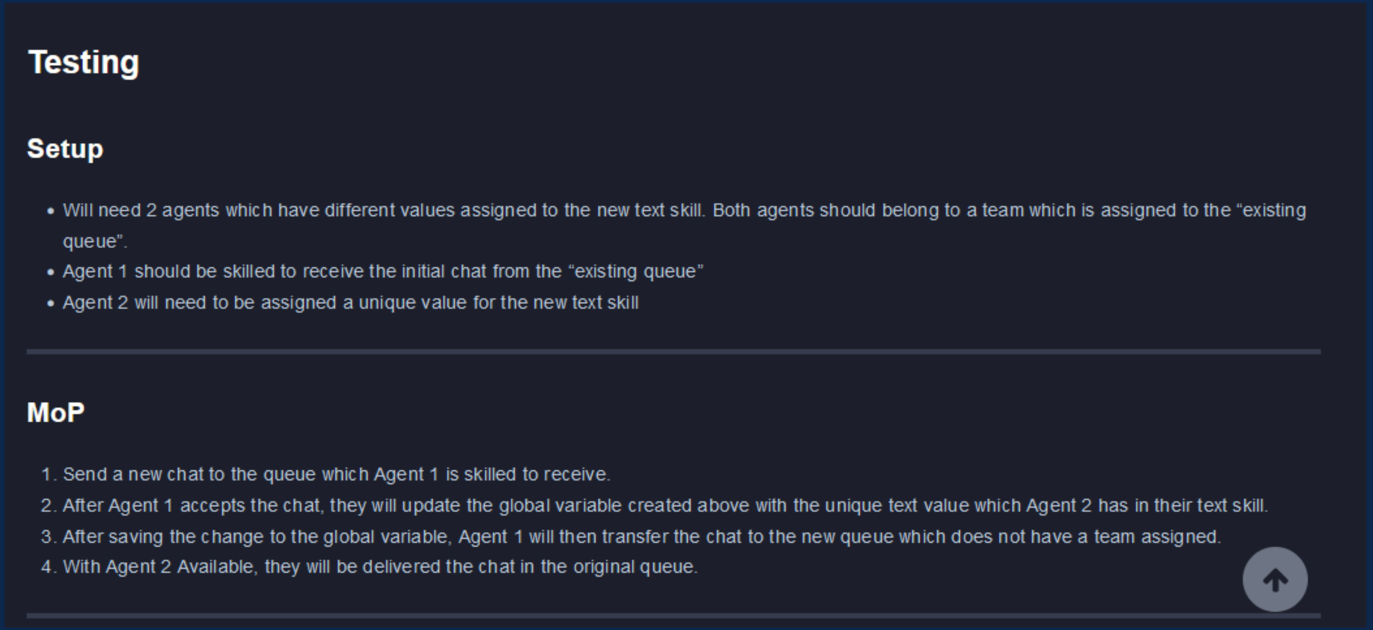
```
// To and From time for Search API Query
var now = Date.now().toString();
var tenMin = (Date.now() - 600000).toString();
```

- Add a Delay node
  - Wait 2 seconds
- Add the Custom node
  - Method Name: The method you created above
  - From: $(tenMin)
  - To: $(now)
  - TaskID: $(n2.webex.ID)
- Add a Queue Task node
  - Task ID: $(n2.webex.ID)
  - Conversation ID: $(MediaResourceId)
  - Queue details:
    - Queue name: existing queue
  - Skill settings:
    - Skill: Text Skill that you created above
    - Condition: IS
    - Skill Value: $(n####.skill)

Steps to Test With

- Setup
  - What is necessary to test successfully

- MoP (Method or Procedure)
  - The steps you will take to recreate the solution on your tenant
  - Confirmation that everything is working as described



Testing

Setup

- Will need 2 agents which have different values assigned to the new text skill. Both agents should belong to a team which is assigned to the "existing queue".
- Agent 1 should be skilled to receive the initial chat from the "existing queue"
- Agent 2 will need to be assigned a unique value for the new text skill

MoP

1. Send a new chat to the queue which Agent 1 is skilled to receive.
2. After Agent 1 accepts the chat, they will update the global variable created above with the unique text value which Agent 2 has in their text skill.
3. After saving the change to the global variable, Agent 1 will then transfer the chat to the new queue which does not have a team assigned.
4. With Agent 2 Available, they will be delivered the chat in the original queue.

Coming soon!

Watch your Partner Success Space and Knowledge Hub

What solutions do you want to create?

ⓘ Start presenting to display the poll results on this slide.

The bridge to possible

Channel Shift – Voice Deflection to WhatsApp from IVR

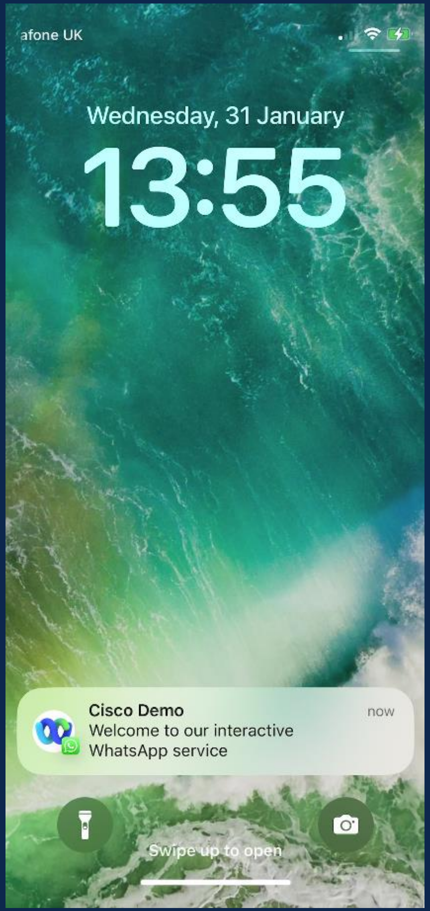
Channel Shift – Voice Deflection to WhatsApp from IVR

Use Case

- Reactive deflection – customer in queue but all agents busy – reduce abandon rate – positive abandons.
- Proactive deflection – channel shift – upfront messaging to divert before customer reaches the queue.
- Saving the customer £/€
  - Reduction in headcount
  - Multi-tasking agents

Features Used

- HTTP Node in Voice Flow
- Connect webhook flow to move to digital channels
- Utilise Bots and real agents if necessary
- Can elevate back to voice if necessary (links back to Scheduling a callback above)

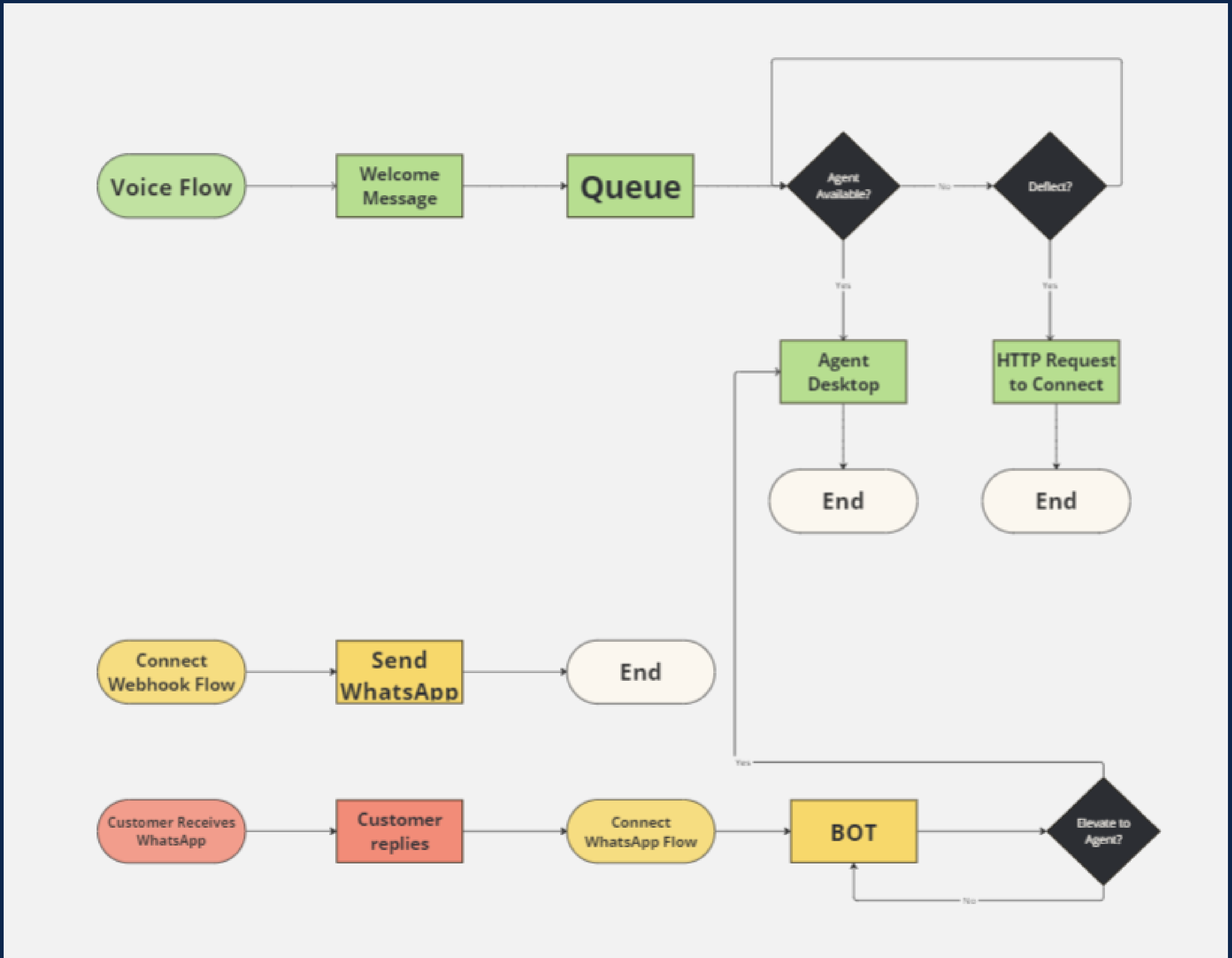


Cisco Demo

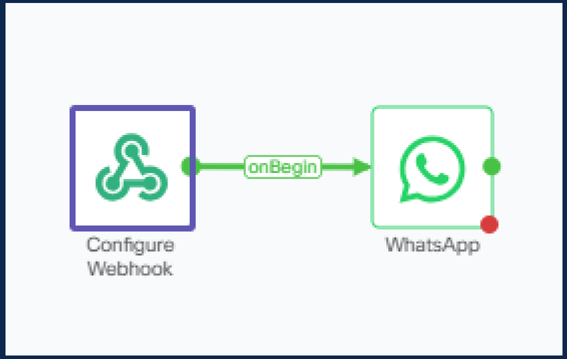
Welcome to our interactive WhatsApp service

Channel Shift – Voice to WhatsApp from Queue

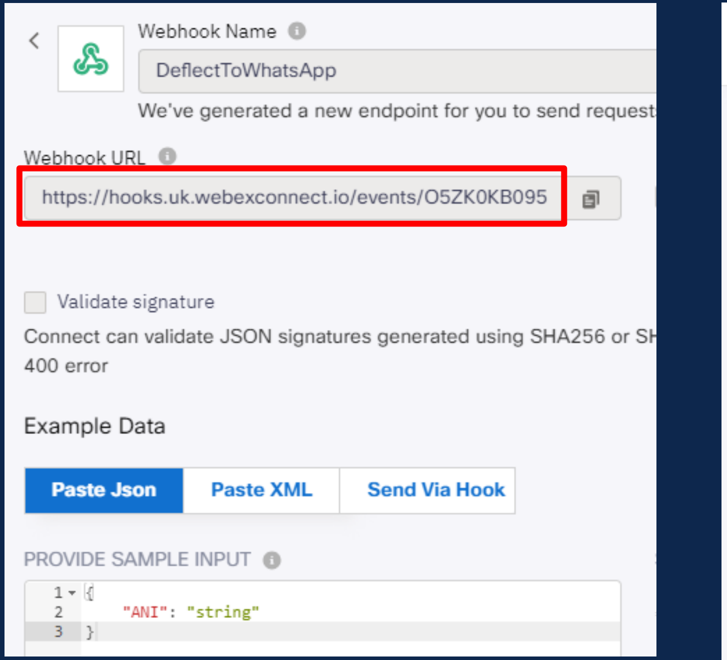
- Components
- **Webex Contact Centre**
- Voice Flow
- **Webex Connect**
- Webhook Flow
- WhatsApp Flow



Channel Shift – Voice to WhatsApp from Queue



- **Webex Connect**
- Webhook Flow
- Webhook URL will be used in Voice flow.
- Use a WhatsApp template for initial outbound message.



Webhook Name
DeflectToWhatsApp
We've generated a new endpoint for you to send request

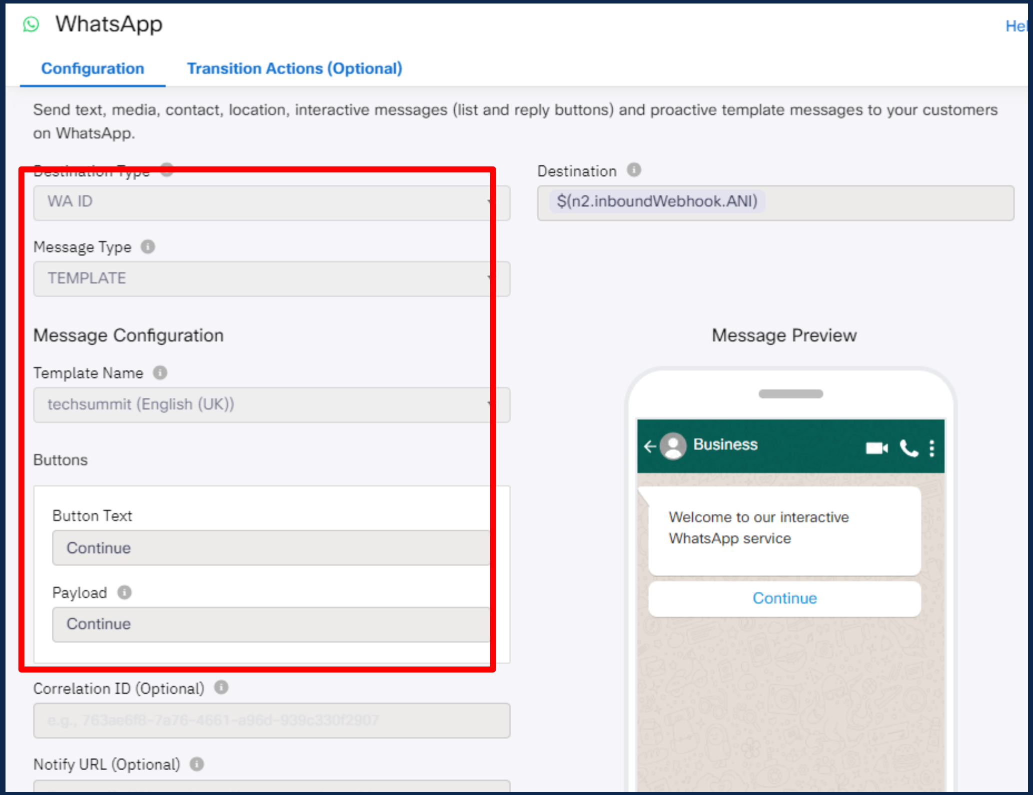
Webhook URL
https://hooks.uk.webexconnect.io/events/O5ZK0KB095

□ Validate signature
Connect can validate JSON signatures generated using SHA256 or SH
400 error

Example Data

PROVIDE SAMPLE INPUT

```
{
    "ANI": "string"
}
```



WhatsApp

Send text, media, contact, location, interactive messages (list and reply buttons) and proactive template messages to your customers on WhatsApp.

Destination Type: WA ID

Destination: $(n2.inboundWebhook.ANI)

Message Type: TEMPLATE

Message Configuration

Template Name: techsummit (English (UK))

Buttons

Button Text: Continue

Payload: Continue

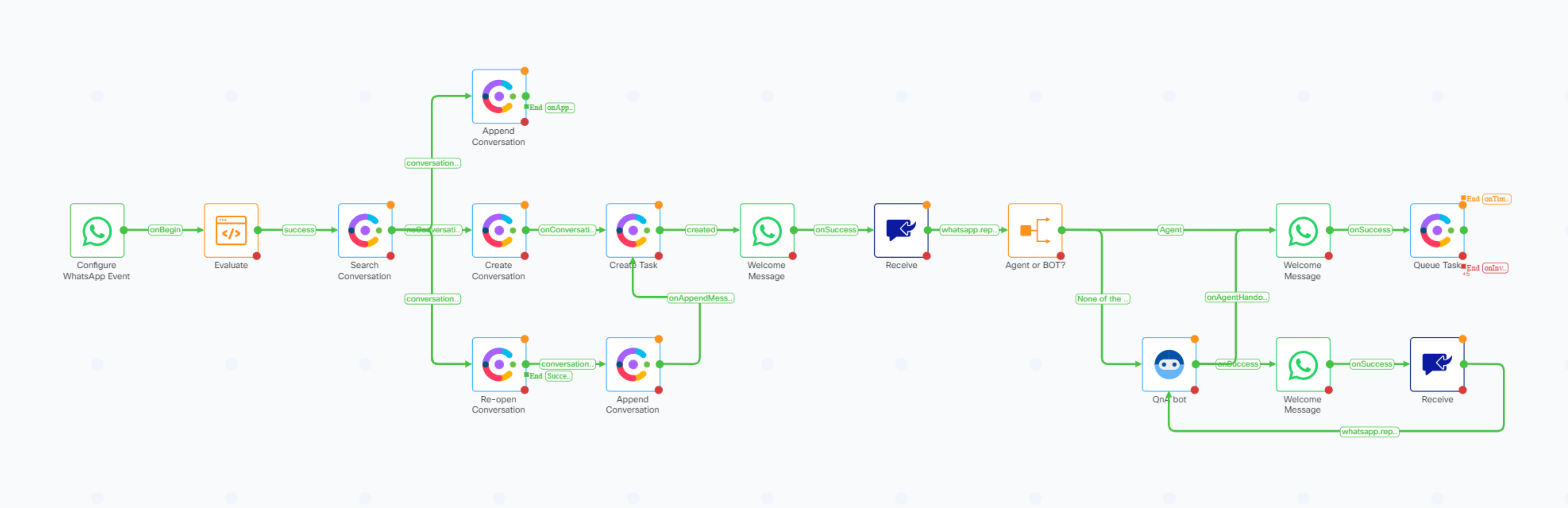
Correlation ID (Optional): e.g., 763ae6f8-7a76-4661-a96d-939c330f2907

Notify URL (Optional)

Message Preview

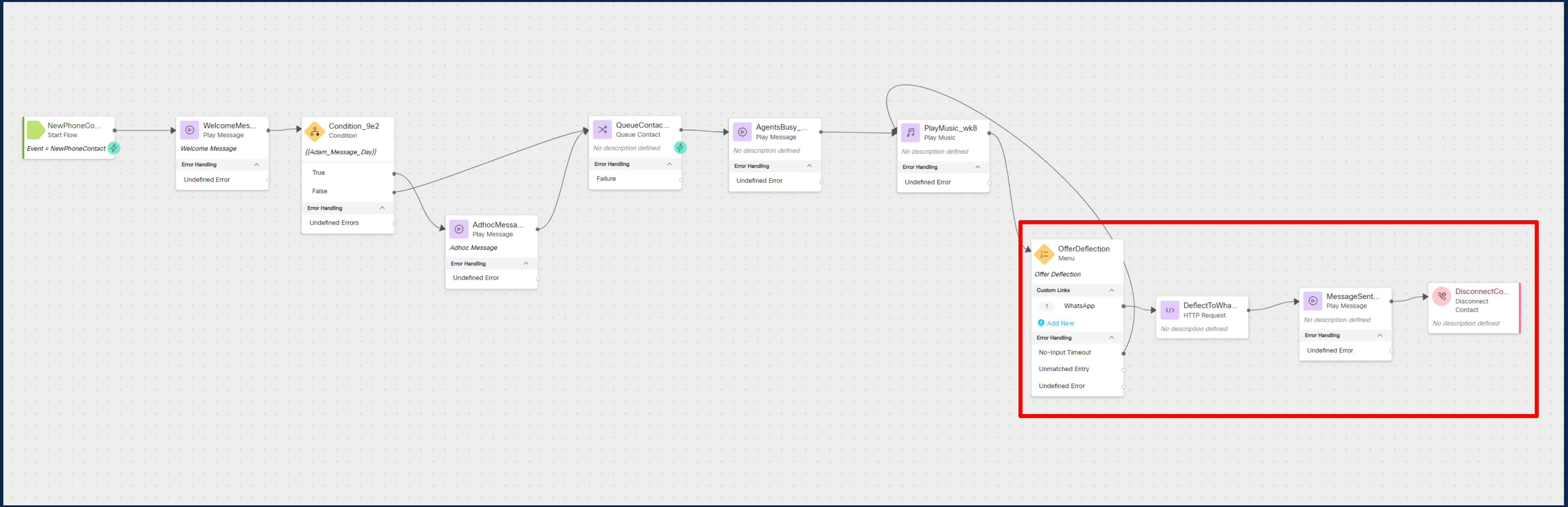
Channel Shift – Voice to WhatsApp from Queue

- Webex Connect
  - ○ WhatsApp flow to handle customer interactions



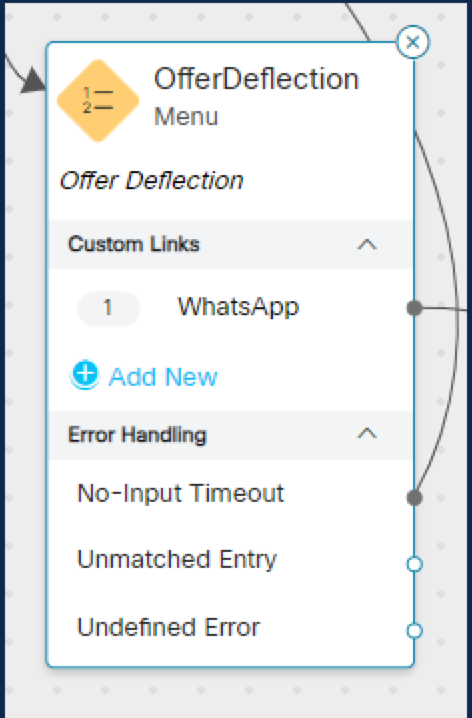
Channel Shift – Voice to WhatsApp from Queue

- Webex Contact Centre
  - Voice Flow



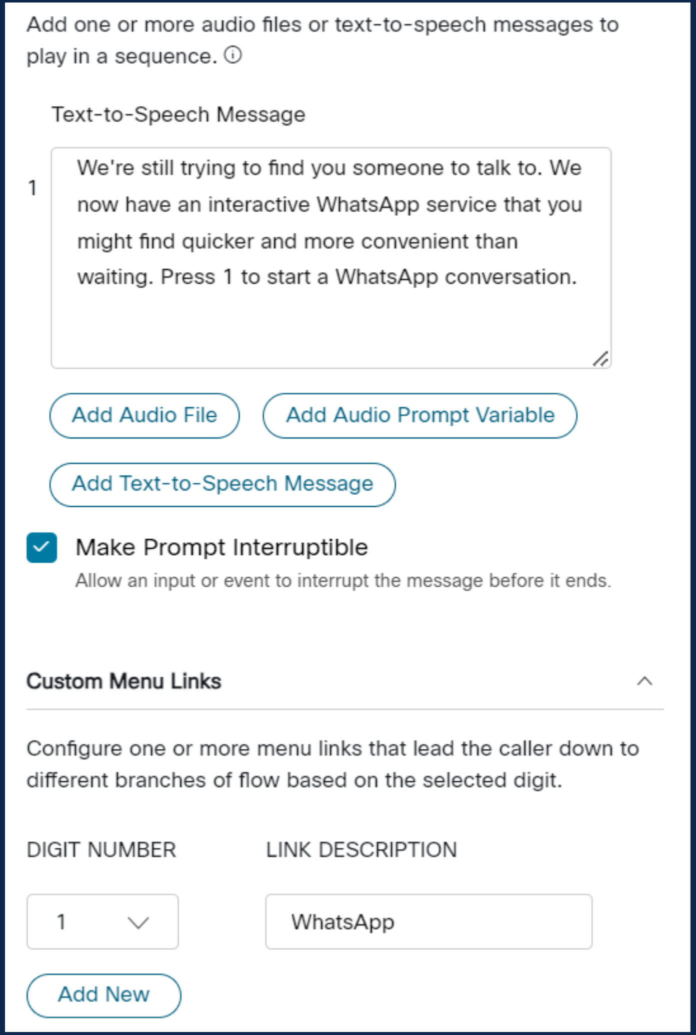
Channel Shift – Voice to WhatsApp from Queue

- Webex Contact Centre
  - Voice Flow
  - Use Menu node to offer deflection



OfferDeflection
Menu

Offer Deflection



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

Text-to-Speech Message

1 We're still trying to find you someone to talk to. We now have an interactive WhatsApp service that you might find quicker and more convenient than waiting. Press 1 to start a WhatsApp conversation.

☑ Make Prompt Interruptible
Allow an input or event to interrupt the message before it ends.

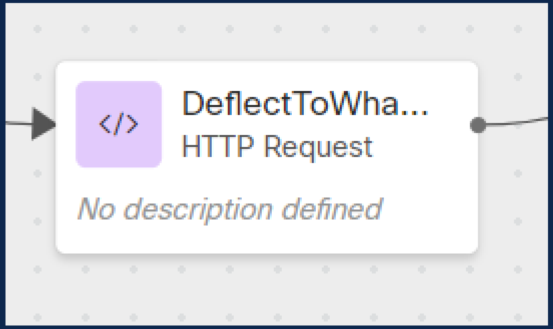
Custom Menu Links

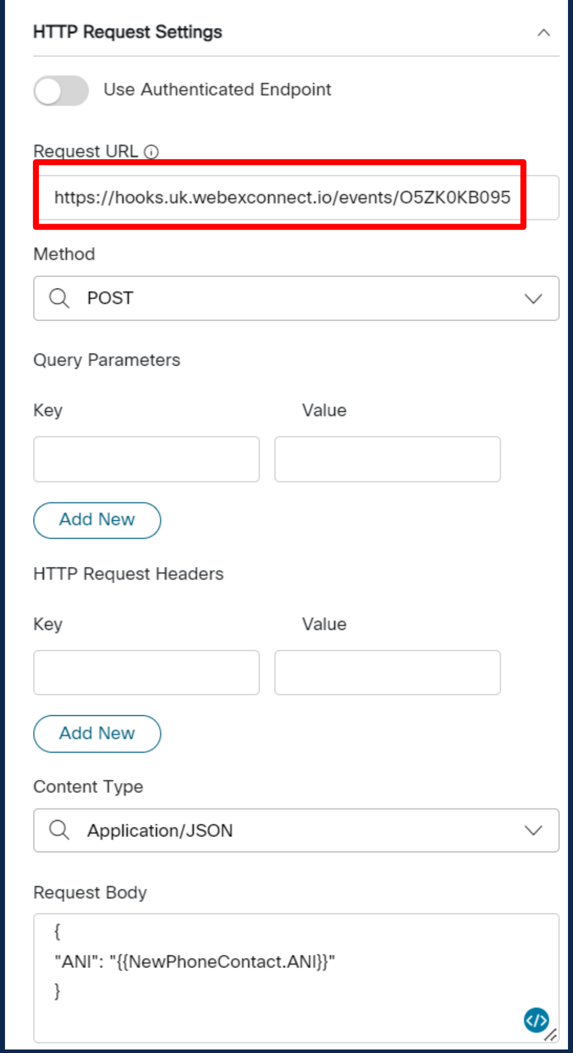
Configure one or more menu links that lead the caller down to different branches of flow based on the selected digit.

DIGIT NUMBER	LINK DESCRIPTION
1	WhatsApp

Channel Shift – Voice to WhatsApp from Queue

- **Webex Contact Centre**
  - Voice Flow
  - Customer presses one on their phone.
  - Use HTTP node to send request to Connect webhook flow created above.
  - Tell customer what has happened in a Message Node.




HTTP Request Settings

Use Authenticated Endpoint

Request URL: https://hooks.uk.webexconnect.io/events/O5ZK0KB095

Method: POST

Query Parameters

Key	Value

Add New

HTTP Request Headers

Key	Value

Add New

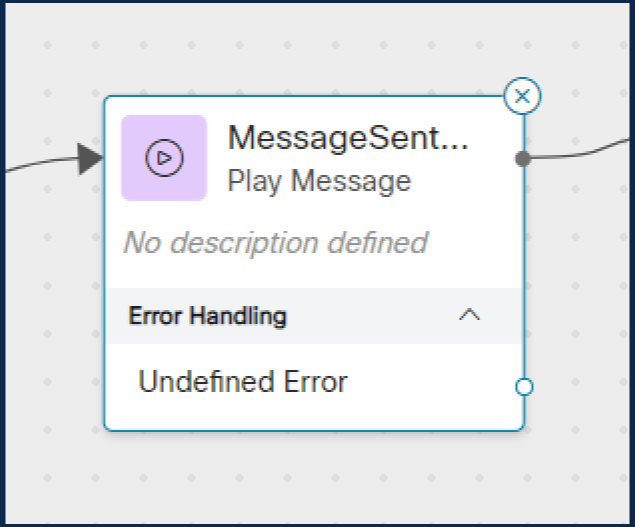
Content Type: Application/JSON

Request Body

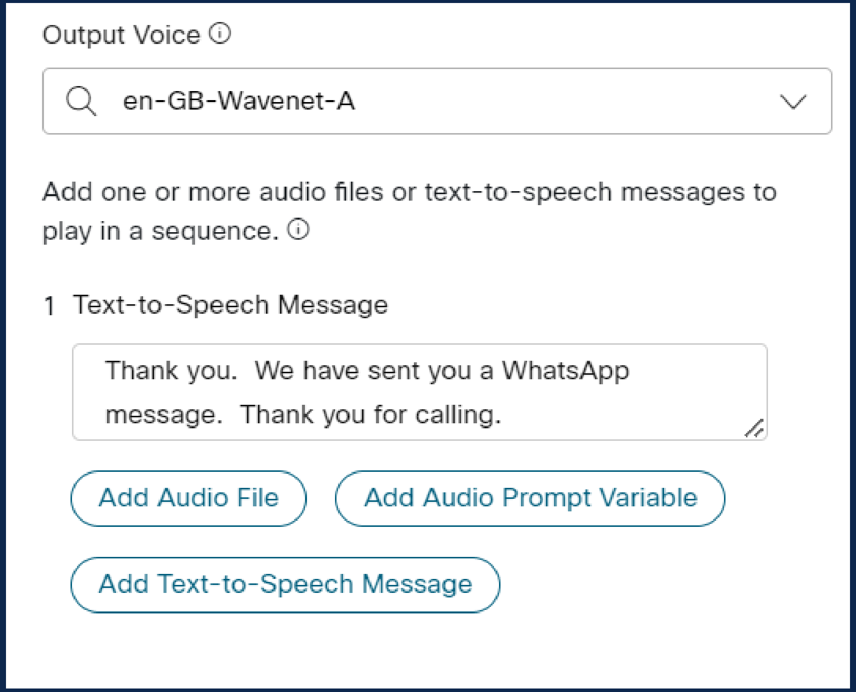
```
{
"ANI": "{{NewPhoneContact.ANI}}"
}
```

Channel Shift – Voice to WhatsApp from Queue

- **Webex Contact Centre**
- Voice Flow
- Tell customer what has happened in a Message Node.



MessageSent...
Play Message
No description defined
Error Handling
Undefined Error



Output Voice ⓘ

en-GB-Wavenet-A

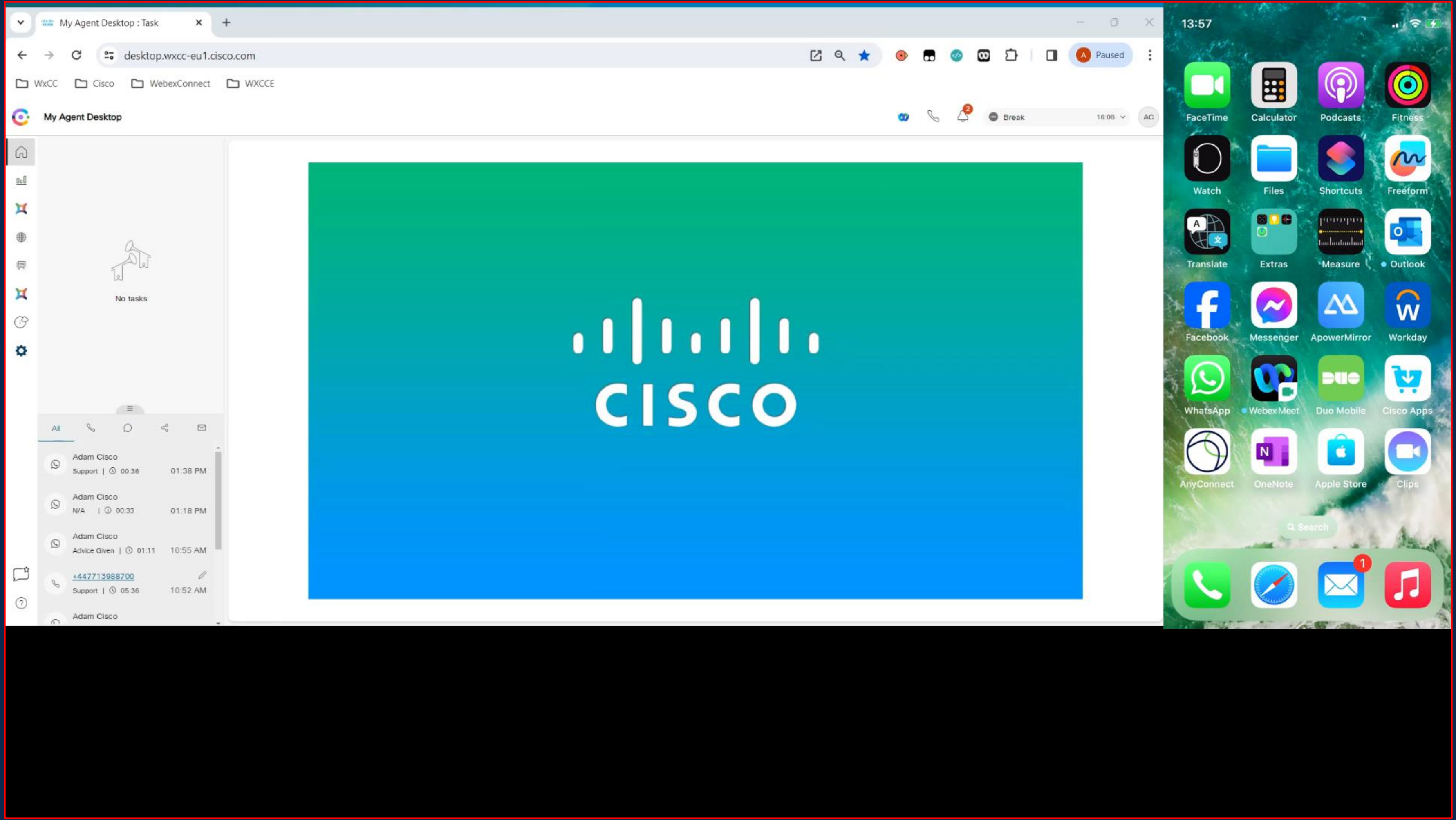
Add one or more audio files or text-to-speech messages to play in a sequence. ⓘ

1 Text-to-Speech Message

Thank you. We have sent you a WhatsApp message. Thank you for calling.

Channel Shift – Voice to SMS from Queue

Demonstration



Adam Cisco
Support | 00:36 01:38 PM

Adam Cisco
N/A | 00:33 01:18 PM

Adam Cisco
Advice Given | 01:11 10:55 AM

+447713988700
Support | 05:36 10:52 AM

Adam Cisco

CISCO

The bridge to possible