Digital Channels Developer's Guide

6/26/2022

Table of Contents

Contents

Find information about how to develop using the Digital Channels APIs.

Related documentation:

-

Digital Channels provides APIs you can use to build custom chat applications and deliver messages between Digital Channels and third-party services such as SMS or email.

Developer tasks

Explore these topics for examples of how to use the Digital Channels APIs.

- Genesys Widgets: Close the chat when an agent disconnects

Digital Channels APIs

Learn more about the Digital Channels APIs on the Genesys Multicloud CX Developer Center.

- Consumer Messaging API
- Third-Party Messaging API

Genesys Widgets: Close the chat when an agent disconnects

Contents

Learn how to finalize asynchronous chat interactions so that customers can no longer type in the Genesys Widget after an agent marks a chat as done.

Related documentation:

-

Digital Channels supports an asynchronous chat model that lets your customers join or rejoin a chat session at their convenience. This divides a chat session into multiple segments, which are represented by separate interactions that agents can handle within your contact center. During these interactions, an agent connects to a chat session and conducts the conversation with a customer. Each interaction can be ended only from the agent side or, for chat bot conversations, a workflow. The interaction ends when the agent sends a reply to the customer that isn't likely to generate an immediate response, or when the customer goes offline (disconnects from chat or doesn't reply in a timely manner). The next interaction in the chat is started by either a new message from the customer or when an agent requests the creation of a new interaction to follow up with customer. From the customer's perspective, these interactions are one large seamless and ongoing conversation.

There are some edge cases when a customer might initiate a new segment of conversation by sending a follow up that isn't actionable for the agent. For example, after being offline the customer comes back and sends a thank you message after the agent has already closed the segment. This follow up from the customer creates a new segment (interaction) to be routed to an agent.

The sections below describe a solution you can implement to modify the Chat Widget behavior to prevent these non-actionable messages, by either notifying the customer or closing the chat window.

Note: It's possible that only one interaction is expected for the whole chat session, which resembles non-asynchronous chat behavior. In this case, you must also adjust the Chat Widget's behavior accordingly.

How to implement the solution

To handle this scenario in your Chat Widget, first set up the widget to respond to state changes from Digital Channels using the Consumer Messaging API.

When Digital Channels detects a state change, it adds a System message to the chat session history and returns it in the response of a long poll or history GET /chat/sessions/{sessionId}/messages request.

For example:

```
{
  "status": {
    "code": "SUCCESS"
  },
  "data": {
```

```
    "messages": [
      {
        "type": "System",
        "from": {
          "participantId": "00000000-0000-0000-0000-000000000000",
          "type": "System"
        },
        "visibility": "All",
        "id": "dba9117a-38d4-4c06-ad67-665409093314",
        "event": "InteractionStateChange",
        "data": {
          "interactionState": "Finalizing"
        },
        "utcTime": "2021-07-09T10:25:38.829Z",
        "streamId": "1625826338845-0",
        "index": "1625826338845-0"
      }
    ],
    "lastDeliveredIndex": "1625826338845-0",
    "chatEnded": false
  },
  "operationId": "69be0dc50e-de8a9-093ca-52684-51ce7aa716db743",
  "referenceId": "1d9c9d1e-e518-446c-9889-57fadc98fd16"
}
```


The widget can capture this message by using the Widgets Bus API to subscribe to the WebChatService.messageReceived event. The way you process this event depends on whether the chat is asynchronous. See the table below for details:

State	Notify customer	Close the chat
Finalizing OR Completed	Inject a message into the widget that says the chat interaction is over. This leaves the option for the customer to start a new interaction. Example message: "Thank you for your query. Please do not respond to this automated message if you do not have additional questions."	End the chat on the widget side. This removes the participant in Digital Channels.

Deploying the widget example

The code below shows an example of how you could handle the event change. If the chat is asynchronous, it posts a message to the customer when the agent closes the segment; otherwise, it closes the chat window when the agent closes the segment.

The sample uses the global CXBus instance, but you can also use other methods of accessing Widgets Bus.

```
// Widgets config and initialization code is omitted for clarity
// ...
CXBus.loadPlugin('widgets-core');
```

```
CXBus.subscribe('WebChatService.messageReceived', (event) => {
  // taking the latest interaction state change event when fetching the history, for long-
poll there will be only one
  const lastInteractionStateChange = event.data.originalMessages
    .filter(message => message.type === 'System' && message.event ===
'InteractionStateChange')
    .pop();

  // nothing to do if no event found
  if (!lastInteractionStateChange) return;

  // digging for current interaction state
  const interactionState = lastInteractionStateChange.data.interactionState;
  if (!interactionState) return;

  // processing the state
  if (interactionState === 'Finalizing' || interactionState === 'Completed') {
    // different handling for async and regular chat
    const isAsync = window._genesys.widgets.webchat.transport.async &&
window._genesys.widgets.webchat.transport.async.enabled;
    if (isAsync) {
      CXBus.command('WebChat.injectMessage', {
        type: 'text',
        text: 'Thank you for your query. Please do not respond to this automated message if
you do not have additional questions.',
        bubble: { time: false }
      });
    } else {
      CXBus.command('WebChat.endChat');
    }
  }
});
```