Callback Administrator's Guide

Callback scenarios

8/6/2025

Contents

-

- Administrator

This article summarizes the scenarios that Genesys Callback supports. In addition to the traditional Immediate and Scheduled callback scenarios, Callback also supports Click-To-Call-In scenarios, allowing consumers to initiate a call to your contact center by tapping a Click-To-Call button in your mobile app.

Related documentation:
-

General information about callbacks

You configure callback scenarios in Designer.

It does not matter how a callback originates, the voice interaction is always converted to a virtual call and added to the queue where it is monitored so the system can provide information such as the Estimated Wait Time (EWT) to future callers.

Immediate callbacks

An *Immediate* callback is a callback that is set in motion when your customer (a *consumer*) makes a request to be called back as soon as an agent who can provide assisted service becomes available. A consumer can request an Immediate callback in the following ways:

- While the consumer is in-queue on an IVR:
    - A consumer's call arrives and the caller is offered Immediate callback. If the caller accepts, he or she confirms the phone number for the callback.

- Through an API call; in other words, the consumer makes the request from a mobile app or website:
    - A consumer is using your company's or organization's mobile app or website and encounters a situation where he or she requires assisted service by voice. The consumer taps or clicks a button to request a callback, confirms or provides the number at which he or she would like to receive the call, and receives a confirmation message that the request was received.

No matter how the Immediate callback is requested, when the consumer reaches the top of the queue, the callback is attempted as soon as an agent who satisfies the required skill expression enters the Ready state. The agent that triggered the callback is not reserved, though, and might take another call while the system tries to reconnect with the consumer. When the system has successfully connected to the consumer, then the call is routed to the next agent with the correct skills who enters the Ready state.

Because user/context data might be attached to the API request for a callback, key components of the consumer's app or web journey can be preserved for agent or reporting use.

For information about provisioning Callback, see Provisioning Callback in Designer.

Scheduled callbacks

A *Scheduled* callback is a callback that is set in motion when your customer (a consumer) makes a request to be called back in the future, at an approximate time that works for the consumer's schedule. A consumer can request a Scheduled callback in the following ways:

- While in-queue on an IVR:
  - A consumer's call arrives and the caller is offered the option to schedule a callback for some time in the future. If the caller accepts, he or she confirms the phone number for the callback and is prompted to input the time at which he or she would like to receive it. Using the caller's requested time, the system searches for the closest-matching available time to connect with an agent.

- Through an API call (from a mobile app or website):
  - A consumer is using your company's or organization's mobile app or website and encounters a situation where he or she requires assisted service by voice. The consumer taps or clicks a button to request a callback, confirms or provides the number at which he or she would like to receive the call, interacts with a date/time picker to search for availability, and receives a confirmation message that the request was received.

No matter how the Scheduled callback is requested, the system – using the desired callback time as the target time slot – makes the outbound call to the consumer when the consumer reaches the top of the queue and an agent with the required skills enters the Ready state. The agent that triggered the callback is not reserved, though, and might take another call while the system tries to reconnect with the consumer. When the system has successfully connected to the consumer, then the call is routed to the next agent with the correct skills who enters the Ready state.

Because user/context data might be attached to the API request for a callback, key components of the consumer's app or web journey can be preserved for agent or reporting use.

For information about provisioning Callback, see Provisioning Callback in Designer.

Click-To-Call-In (Immediate)

> ### Important
> To implement this scenario, you need to use the corresponding Call-In API to initiate the Click-To-Call-In request.

A Click-To-Call-In (Immediate) interaction is set in motion when your customer (a consumer) taps a button in a mobile app that is designed to trigger a Call-In API request:

- A consumer is using your company's or organization's mobile app and encounters a situation where he or she requires assisted service by voice. The consumer taps a button that you have provisioned in your app to connect consumers to your contact center.

- The system responds with call-in details immediately. Using that information, the app triggers a call to your contact center. The system attempts to match the caller to existing information. For more information, see Provisioning the Click-to-Call-In scenario.

- If the attempt to match the caller to a Call-In request is successful and your Designer application is configured to route the call when a match is made, then the call is queued on hold like any other call to the contact center. If the consumer is placed in a queue where the EWT is above the configured threshold, then the consumer might be offered a callback option.

Because user/context data might be attached to the API request, key components of the consumer's app journey can be preserved for agent or reporting use.

Click-To-Call-In (Delayed)

Important

To implement this scenario, you need to use the corresponding Call-In API to initiate the Click-To-Call-In request.

A Click-To-Call-In (Delayed) interaction is set in motion when your customer (a consumer) taps a button in a mobile app that is designed to trigger a Callback API request. The following description of what happens next is a brief summary. See How Click-To-Call-In (Delayed) Works for additional information. To provision the Click-To-Call-In scenario, see Provisioning the Click-to-Call-In scenario.

- A consumer is using your company's or organization's mobile app and encounters a situation where he or she requires assisted service by voice. The consumer taps a button to contact your center.

- The mobile app sends a callback request, which includes Push parameters that the system uses to contact the consumer to provide information about the callback.

- The call is queued in the Click-To-Call-In (Delayed) virtual queue.

- When the consumer reaches the top of the queue, the system sends a Push Notification to the mobile app to notify the consumer that the callback is ready.

- When the consumer accepts the callback, the system immediately replies with a Push Notification that provides a phone number to call and, if configured, a unique access code that the consumer will be asked to enter before the call is initiated.

- When the consumer calls in and enters the access code, if required, the system attempts to match the caller to an existing callback request. When a match is made, the caller is queued and routed to the next available agent who satisfies the required skill expression.

Because user/context data might be attached to the API request, key components of the consumer's app journey can be preserved for agent or reporting use.

How Click-To-Call-In (Delayed) works

Your mobile app sends a Callback request when the consumer taps the button or link that you have provisioned for the Click-To-Call-In Delayed feature. The call is then queued in the Click-To-Call-In (Delayed) virtual queue. The Callback request includes Push parameters. To use Push Notifications with Callback, see Provisioning Push Notifications.

A Click-To-Call-In (Delayed) request is routed to an agent when the following criteria are met:

1. The Call-In request can be matched to an existing Callback request in the system.
2. Your Click-to-Call-In Match Designer application is configured to route the call.
3. An agent (with the correct skills if skills are configured) is ready to accept the call.

There are, however, a number of things that can happen during an active Click-To-Call-In (Delayed) session that might impact the session's flow. For example, the system might fail to match a Click-To-Call-In (Delayed) request on the first attempt. As long as the Click-To-Call-In and Callback requests remain valid and outstanding, though, the consumer can call and try again.

There are also Designer settings that can purge the callback from the system or remove the callback from its queue:

1. The **Callback Purge Time (minutes)** for the virtual queue is reached before the consumer responds to Push Notifications or before an agent is available to assist the caller. In this case, the callback is purged from the system.
2. The end of the business day, based on the configured Business Hours, occurs before the **Callback Purge Time (minutes)** is reached, before the consumer responds to Push Notifications, or before an agent is available to assist the caller. In this case, the callback is purged from the system.
3. The **Pushed Callback Expiry Time (minutes)** setting in Designer causes the callback session to terminate because the specified time interval expires. When this happens, the callback is removed from the queue, but not purged from the system until one of the preceding two events occurs.

A Call-In API request that is associated with a Callback request that was purged or terminated will fail because no match can be made between the Call-In and Callback requests.