



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Widgets API Reference

CallbackService

Contents

- [1 Overview](#)
 - [1.1 Usage](#)
 - [1.2 Namespace](#)
 - [1.3 Customization](#)
- [2 Configuration](#)
 - [2.1 Description](#)
 - [2.2 Example](#)
 - [2.3 Options](#)
- [3 Localization](#)
- [4 API Commands](#)
 - [4.1 configure](#)
 - [4.2 schedule](#)
 - [4.3 availability](#)
- [5 API Events](#)

-
- Developer

Learn how to use CallbackService to schedule a callback with customer service.

Related documentation:

-

Overview

CallbackService exposes high-level API access to Genesys Callback services, allowing you to use our Callback Widget to schedule a callback with customer service—or to develop your own custom Callback Widget. CallbackService dramatically simplifies integration, improving the reliability, feature set, and compatibility of every widget on the bus.

Usage

Callback Service and the matching Callback widget work together, and they share a configuration object. Using Callback uses CallbackService.

You can also use Callback Service as a high-level API using bus commands and events to build your own Callback widget.

Namespace

The CallbackService plugin has the following namespaces tied to each of the following types:

| Type | Namespace |
|---------------------------------|-----------------|
| Configuration | Sendmessage |
| CXBus—API commands & API events | CallbackService |

Customization

CallbackService does not have customization options. It is a Plug and Play plugin and works as is.

Configuration

Description

Callback and CallbackService share the **_genesys.widgets.callback** configuration namespace. Callback contains the UI options and CallbackService contains the connection options.

Example

```
window._genesys.widgets.callback = {  
  apikey: 'n3eNkgXXXXXXXXXXXXXXXXXA',  
  apiVersion: 'v3',  
  serviceName: 'service',  
  dataURL: 'http://host:port/callbacks',  
  userData: {},  
  countryCodes: true  
};
```

Options

| Name | Type | Description | Accepted Values | Default | Required |
|-------------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|---------|-----------------------------------|
| apikey | string | If apiVersion is v3, this holds the x-api-key value. | n/a | n/a | Yes, if using Apigee Proxy |
| dataURL | URL String | URL to the API endpoint for Callback. Important The base URL for your API endpoints is: <code>https://gapi-.genesyscengagement/v3</code> You will receive the information from Genesys at the same time that you receive your API key. | n/a | n/a | Always |
| apiVersion | string | Version of Callback API. Important This value determines the version of Callback API. | 'v3' | 'v1' | Yes, if using Callback v3 dataURL |
| serviceName | string | Name of the Callback virtual queue. | n/a | n/a | Yes, if using Callback v3 dataURL |
| userData | object | Arbitrary | n/a | {} | |

| Name | Type | Description | Accepted Values | Default | Required |
|-------------|--------|-------------------------------------------------------|-----------------|---------|----------|
| | | attached data to include while scheduling a callback. | | | |
| ajaxTimeout | number | Number of milliseconds to wait before AJAX timeout. | n/a | 3000 | |

Localization

CallbackService does not have localization options.

API Commands

Once you've registered your plugin on the bus, you can call commands on other registered plugins. Here's how to use the global bus object to register a new plugin on the bus.

Important

The global bus object is a debugging tool. When implementing Widgets on your own site, do not use the global bus object to register your custom plugins. Instead, see Genesys Widgets Extensions for more information about extending Genesys Widgets.

```
var oMyPlugin = window._genesys.widgets.bus.registerPlugin('MyPlugin');
oMyPlugin.command('CallbackService.schedule', {
  userData: {},
  firstname: 'Bob',
  lastname: 'Jones',
  email: 'b.jones@mail.com',
  subject: 'product questions',
  desiredTime: '2017-04-04T00:24:17.804Z',
  phonenumber: '4151110000'
});
```

configure

Internal use only. The main App plugin shares configuration settings with widgets using each widget's configure command. The configure command can only be called at startup. Calling configure again after startup may result in unpredictable behavior.

schedule

Schedule a callback service with the callback schedule API.

Example

```
oMyPlugin.command('CallbackService.schedule', {  
    userData: {},  
    serviceName: 'service' // service name from callback API v3 version,  
    firstname: 'Bob',  
    lastname: 'Jones',  
    email: 'b.jones@mail.com',  
    subject: 'product questions',  
    desiredTime: '2017-03-03T00:24:17.804Z',  
    phonenumber: '4151110000'  
});
```

Options

| Option | Type | Description |
|-------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| firstname | string | Receive a Call entry Form Data: 'firstname'. |
| lastname | string | Receive a Call entry Form Data: 'lastname'. |
| phonenumber | string | Receive a Call entry Form Data: 'phonenumber'. |
| subject | string | Receive a Call entry Form Data: 'notes'. |
| email | string | Receive a Call entry Form Data: 'email'. |
| desiredtime | string | The preferred desired time user would like to get the callback scheduled. Time should be in UTC format. |
| userData | object | Arbitrary data that is to be attached with callback schedule. Properties defined here will be merged with default userData set in the configuration object. |
| serviceName | string | Service Name of Callback API to be passed if the apiVersion is v3. |

Resolutions

| Status | When | Returns |
|----------|---------------------------------------|------------------------------------------|
| resolved | Server confirms callback is scheduled | 200 OK AJAX Response - Schedule Callback |

| Status | When | Returns |
|----------|--------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| | | For Callback API v3, refer to 'Responses' in Schedule Callback V3 |
| rejected | Selected time slot is not available | 400 Bad Request AJAX Error Response For Callback API v3, refer to 'Responses' in Schedule Callback V3 |
| rejected | AJAX exception occurs | 429 Too Many Requests AJAX Error Response For Callback API v3, refer to 'Responses' in Schedule Callback V3 |
| rejected | Server exception occurs | 500 Internal Server Error Response For Callback API v3, refer to 'Responses' in Schedule Callback V3 |
| rejected | No form data is found to schedule callback | 'No data found to schedule callback' |

availability

Get the list of available callback time slots using the callback service.

Example

```
oMyPlugin.command('CallbackService.availability', {
    serviceName: 'service' // service name from callback API v3 version,
    startDate: '2017-04-03T00:24:17.804Z',
    numberOfDays: '5',
    maxTimeSlots: 20
}).done(function(e){
    // CallbackService successfully showing availability
}).fail(function(e){
    // CallbackService failed to show availability
});
```

Options

| Option | Type | Description |
|--------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| startDate | string | The start date is specified in ISO 8601 format, using UTC as the timezone (yyyy-MM-ddTHH:mm:ss.SSSZ). |
| endDate | string | The end date is specified in ISO 8601 format, using UTC as timezone (yyyy-MM-ddTHH:mm:ss.SSSZ). If neither endDate nor numberOfDays is specified, the end date is assumed to be the same as the start date. |
| numberOfDays | string | Used as an alternative to the end date. If neither endDate nor numberOfDays is specified, the end date is assumed to be the same as the start date. |
| maxTimeSlots | number | The maximum number of time slots to be included in the response. |
| serviceName | string | Service Name of Callback API to be passed if the apiVersion is v3. |

Resolutions

| Status | When | Returns |
|----------|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| resolved | Server confirms the list of available callback time slots | 200 OK AJAX Response - Query Callback Availability For Callback API v3, refer to 'Responses' in Availability Callback V3 |
| rejected | Time slots are not available for selected period | 400 Bad Request AJAX Response For Callback API v3, refer to 'Responses' in Availability Callback V3 |
| rejected | AJAX exception occurs | 400 Bad Request AJAX Response For Callback API v3, refer to 'Responses' in Availability Callback V3 |
| rejected | Server exception occurs | 500 Internal Server Error Response |

| Status | When | Returns |
|----------|------------------------|-----------------------------------------------------------------------|
| | | For Callback API v3, refer to 'Responses' in Availability Callback V3 |
| rejected | No query data is found | 'No query parameters passed for callback availability service' |

API Events

Once you've registered your plugin on the bus, you can subscribe to and listen for published events. Here's how to use the global bus object to register a new plugin on the bus.

Important

The global bus object is a debugging tool. When implementing Widgets on your own site, do not use the global bus object to register your custom plugins. Instead, see Genesys Widgets Extensions for more information about extending Genesys Widgets.

```
var oMyPlugin = window._genesys.widgets.bus.registerPlugin('MyPlugin');
oMyPlugin.subscribe('CallbackService.ready', function(e){});
```

| Name | Description | Data |
|-------------------|---------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| ready | CallbackService is initialized and ready to accept commands. | n/a |
| scheduled | Callback is scheduled successfully. | 200 OK AJAX Response - Schedule Callback For Callback API v3, refer to 'Responses' in Schedule Callback V3 |
| scheduleError | An error occurred between the client and the server during a callback schedule. | The JSON data returned by Callback server. For Callback API v3, refer to 'Responses' in Schedule Callback V3 |
| availableSlots | Callback available slots fetched successfully. | 200 OK AJAX Response - Query Callback Availability For Callback API v3, refer to 'Responses' in Availability Callback V3 |
| availabilityError | An error occurred between the | The JSON data returned by |

| Name | Description | Data |
|------|---------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| | client and the server while fetching the available timeslots. | Callback server. For Callback API v3, refer to 'Responses' in Availability Callback V3 |