Widgets API Reference

Calendar

4/26/2026

Contents

- 1 Overview
  - 1.1 Usage
  - 1.2 Customization
  - 1.3 Namespace
  - 1.4 Mobile support
  - 1.5 Screenshots
- 2 Configuration
  - 2.1 Description
  - 2.2 Example
  - 2.3 Options
- 3 Localization
  - 3.1 Usage
  - 3.2 Example i18n JSON
- 4 API commands
  - 4.1 configure
  - 4.2 generate
  - 4.3 showAvailability
  - 4.4 reset
- 5 API events

- Developer

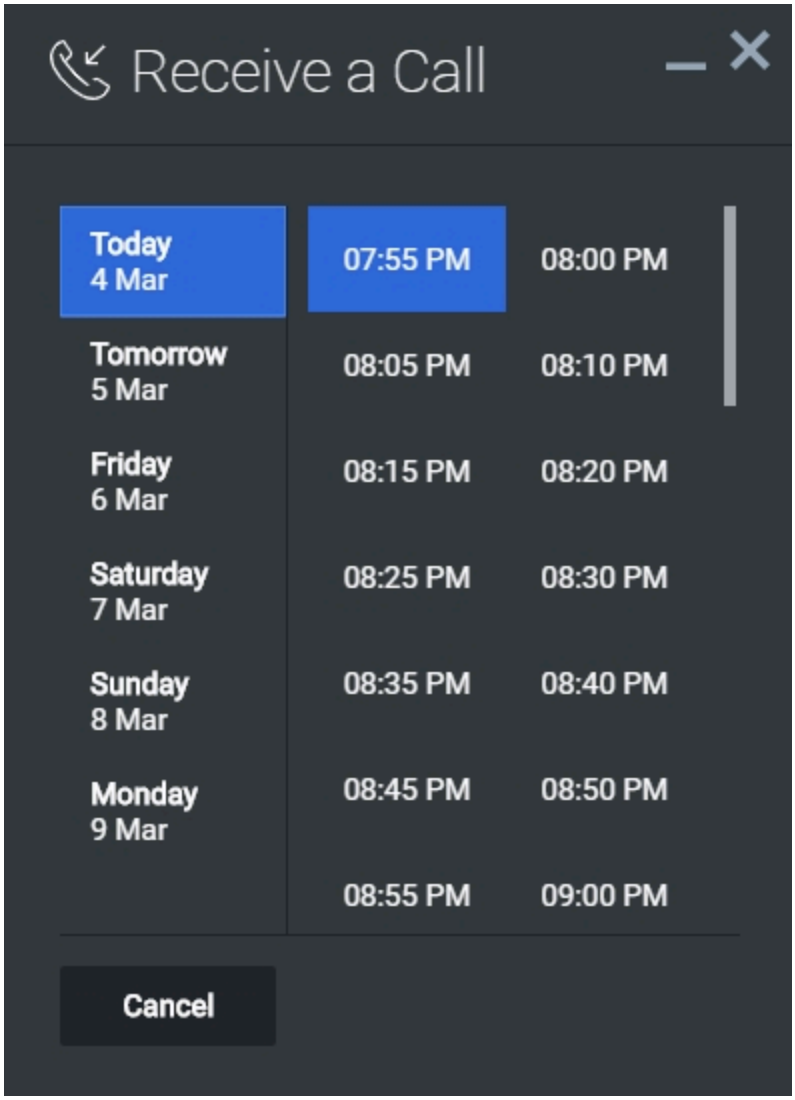
Learn how to display a calendar, so your customers can choose when they want to be contacted.

Related documentation:

-

Overview

The Calendar UI plugin displays time slots for a selected day. The number of days to display—and the opening and closing times for a day—are configurable, as shown in the configuration section.



Usage

> **Important**
> By default, the Calendar widget needs a UI container to display itself properly. For information about how to create and display a calendar, see the API events section.

- Enable or disable certain sections of a day using calendarHours.section.enable
- Define your own business hours for each section of a day using calendarHours.section.openTime and calendarHours.section.closeTime
- Use showAvailability to enable only those time slots for which a customer service agent is available and

disable the rest.

- Define your own time interval between each time slot.

How does the Calendar widget render time slots in local time zones?

1. The Calendar widget uses the command `showAvailability` which calls `CallbackService.availability` with the start date. This start date is then converted into the ISO 8601 format, using UTC as the timezone by `toISOString()`, internally.
2. The Callback service fetches the available time slots from the server.
3. The Calendar gets the available time slots from `CallbackService.availableSlots` in the ISO 8601 format, using UTC as the timezone.
4. Each and Every Time Slot is converted according to the user's local time zone internally through `Date()` and `toTimeString()` methods in the Calendar Plugin.

Customization

All the texts displayed by the Calendar Widget are fully localizable.

Namespace

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

Type	Namespace
Configuration	calendar
i18n - Localization	calendar
CXBus—API commands & API events	Calendar
CSS	.cx-calendar

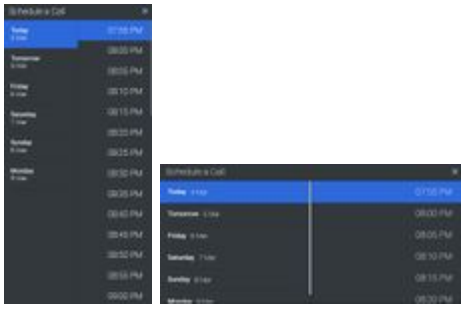
Mobile support

Calendar supports both desktop and mobile devices. Like all Genesys Widgets, there are two main modes: Desktop and Mobile. Desktop is used for monitors, portable computers, and tablets, while Mobile is used for mobile devices. When a mobile device is detected, Calendar switches to special full-screen templates that are optimized for both portrait and landscape orientations.

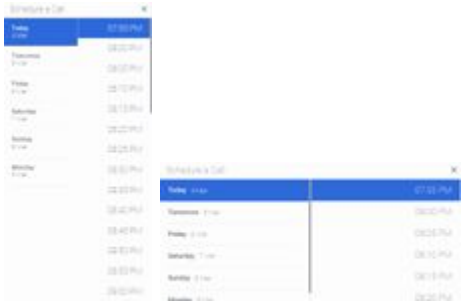
Switching between Desktop and Mobile mode is done automatically by default. You can also configure Genesys Widgets to switch between Desktop and Mobile mode manually.

Screenshots

Dark theme



Light theme



Configuration

Description

Calendar shares the **_genesys.widgets.calendar** configuration namespace. It also has UI options.

Example

```
window._genesys.widgets.calendar = {

        showAvailability: true,
        numberOfDays: 5,
        hideUnavailableTimeSlots: false

        calendarHours: {

                interval: 10,
                allDay: {

                        openTime: '09:00',
                        closeTime: '23:59'
                }
        }
};
```

Options

Name	Type	Description	Default	Required
showAvailability	boolean	Enable or disable calendar to update the time slots based on the callback availability. The unavailable time slots are grayed out.	true	n/a
numberOfDays	number	The number of days to display on calendar starting today.	5	n/a
timeFormat	number/string	This sets the time format for the timestamps in this widget. It can be 12 or 24.	12	n/a
hideUnavailableTimeSlots	boolean	Show hide the unavailable callback time slots.	false	n/a
calendarHours.interval	number	The time interval between each consecutive time slot displayed on calendar.	15	n/a
calendarHours.allDay.openTime	number	Opening time in 'HH:MM' 24 Hr format.	17:00	n/a
calendarHours.allDay.closeTime	number	Closing time in 'HH:MM' 24 Hr format.	23:59	n/a

Localization

> **Important**
> For information on how to set up localization, refer to Localize widgets and services.

Usage

You must use the **calendar** namespace when you're defining localization strings for the Calendar plugin in your i18n JSON file.

The following example shows how to define new strings for the **en** (English) language. You can use any language codes you wish, as there isn't a standard format. When selecting the active language in your configuration, you must match one of the language codes defined in your i18n JSON file. Note that you must only define a language code once in your i18n JSON file. Inside each language object you must define new strings for each widget.

Example i18n JSON

```
{
        "en": {
                "calendar": {
                        "CalendarDayLabels": [
                                "Sunday",
                                "Monday",
                                "Tuesday",
                                "Wednesday",
                                "Thursday",
                                "Friday",
                                "Saturday"
                        ],
                        "CalendarMonthLabels": [
                                "Jan",
                                "Feb",
                                "Mar",
                                "Apr",
                                "May",
                                "Jun",
                                "Jul",
                                "Aug",
                                "Sept",
                                "Oct",
                                "Nov",
                                "Dec"
                        ],
                        "CalendarLabelToday": "Today",
                        "CalendarLabelTomorrow": "Tomorrow",
                        "CalendarTitle": "Schedule a Call",
                        "CalendarOkButtonText": "Okay",
                        "CalendarError": "Unable to fetch availability details.",
                        "CalendarClose": "Cancel",
                        "AriaWindowTitle": "Calendar Window",
                        "AriaCalendarClose": "Cancel",
                        "AriaYouHaveChosen": "You have chosen",
                        "AriaNoTimeSlotsFound": "No time slots found for selected date"
                }
        }
}
```

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, you must not use the global bus object to register your custom plugins. For more information about extending Genesys Widgets, see Genesys Widgets Extensions.

```
var oMyPlugin = window._genesys.widgets.bus.registerPlugin('MyPlugin');

oMyPlugin.command('Calendar.reset');
```

configure

Internal use only. The main App plugin shares widget configuration settings using each widget's **configure** command. The **configure** command can only be called once, at startup. If you call **configure** after startup, the results are unpredictable.

generate

Builds and generates the calendar. Subscribe to the **generate** events to get the generated calendar and display it where you would like to.

Example

```
oMyPlugin.command('Calendar.generate', {date: 'Mon Mar 20 2017 19:51:47 GMT-0700
(PDT)'}).done(function(e){

        // Calendar generated successfully

}).fail(function(e){

        // Calendar failed to generate
});
```

Options

Option	Type	Description
date	Date string/object	To pre-select the date and time on calendar.

Resolutions

Status	When	Returns
resolved	When the calendar is successfully generated	n/a
rejected	When Invalid date is passed to calendar	'Invalid data'

showAvailability

Update the calendar time slots with the callback availability. This enables only those time slots that have the callback facility and disables the rest.

Example

```
oMyPlugin.command('Calendar.showAvailability', {date: '03/22/17'}).done(function(e){

        // Calendar showed availability successfully

}).fail(function(e){

        // Calendar failed to show availability
});
```

Options

Option	Type	Description
date	Date string/object	Update the available time slots in the Calendar plugin for the selected Date. Note that, after calling this command, the internal `showAvailability` value is set to `true` for this session and the Calendar only shows the available time slots when switching between other dates.

Resolutions

Status	When	Returns
resolved	When time slots are successfully updated	n/a
rejected	When no date value is found to	'No date found to check

Status	When	Returns
	check the availability	availability'
rejected	When invalid date value is found	'Invalid date'

reset

Resets the calendar with no pre-selected values.

Example

```
oMyPlugin.command('Calendar.reset').done(function(e){

        // Calendar reset successfully

}).fail(function(e){

        // Calendar failed to reset
});
```

Resolutions

Status	When	Returns
resolved	When calendar is successfully reset	n/a

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, you must 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('Calendar.ready', function(e){});
```

Name	Description	Data
ready	Calendar is initialized and ready to accept commands.	n/a
generated	Calendar UI has been generated. Use this event to get the calendar UI and display where you would like to.	{ ndCalendar: }
selectedDateTime	Date and time selected on calendar.	{ dayString: , dateString: , timeString: , date: }