



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

Toaster

Contents

- [1 Overview](#)
 - [1.1 Usage](#)
 - [1.2 Namespace](#)
 - [1.3 Customization](#)
 - [1.4 Mobile support](#)
- [2 Configuration](#)
- [3 Localization](#)
- [4 API Commands](#)
 - [4.1 open](#)
 - [4.2 close](#)
- [5 API Events](#)

-
- Developer

Learn how to use a toast view control into which widgets can inject their UI.

Related documentation:

-

Overview

The Toaster plugin provides a toast view control that widgets can inject their UI into, accepting the HTML UI, placing it inside a toast view, and displaying the UI on-screen at the lower-bottom-right. When it opens, it slides up from the bottom. When it closes, it slides down until it is off-screen.

Toaster provides these benefits:

- Shows UI as a slide-up toast view in the lower-bottom-right of the screen.
- Open and close transition animations.
- No overlapping toasts; only one at a time. Automatically managed by the Toaster plugin.

Usage

Toaster is easy to use - you simply open and close it. When you call `Toaster.open`, you pass in the HTML content you want to show. If you call `Toaster.open` again while a toast is already open, it will automatically close the previous toast before showing yours (unless the previous toast has reserved the view to prevent new toasts).

Namespace

The Toaster plugin has the following namespaces tied to each of the following types.

Type	Namespace
CXBus—API commands & API events	Toaster
CSS	.cx-toaster

Customization

Toaster does not have customization options.

Mobile support

Toaster does not have mobile-specific styles at this time.

Configuration

Toaster does not have configuration options.

Localization

Toaster 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('Toaster.close');
```

open

Opens the Toaster UI.

Example

```
oMyPlugin.command('Toaster.open', {  
  type: 'generic',  
  title: 'Toaster Title',  
  body: 'Toaster Body',  
  icon: 'chat',  
  controls: 'close',  
  immutable: false,  
  buttons:{  
    type: 'binary',  
    primary: 'Accept',  
    secondary: 'Decline'  
  }  
})
```

```

}).done(function(e){
    // Toaster opened successfully
}).fail(function(e){
    // Toaster failed to open properly
});

```

Options

Option	Type	Description
type	string	Specifies the type of body content that can be provided to Toaster window. Generic type shows the default body content and custom type overrides the default html body content.
title	string	Heading title to display on the Toaster window.
body	string	Holds text value for Generic Toaster type and html string template for Custom Toaster type.
icon	string	The CSS class name for an icon.
controls	string	Show close and minimize controls on Toaster window.
buttons	object	Define the type of buttons.
buttons.type	string	Shows two buttons on the Toaster .
buttons.primary	string	Text to be shown on primary button.
buttons.secondary	string	Text to be shown on secondary button.
immutable	boolean	When set to true, Toaster cannot be closed by other plugins.

Resolutions

Status	When	Returns
resolved	Toaster is successfully opened	n/a
rejected	No Toaster type is specified	'No content was provided. Toaster has ignored your command'
rejected	Toaster is already opened	'Toaster view is currently reserved'

close

Closes the Toaster UI.

Example

```
oMyPlugin.command('Toaster.close').done(function(e){
    // Toaster closed successfully
}).fail(function(e){
    // Toaster failed to close
});
```

Resolutions

Status	When	Returns
resolved	Toaster is successfully closed	n/a
rejected	Toaster is already closed	'Toaster view is already closed'
rejected	Toaster view is immutable	'Toaster view is currently reserved'

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('Toaster.ready', function(e){});
```

Name	Description	Data
ready	The Toaster plugin is initialized and ready to accept commands	n/a
closed	The Toaster plugin has been removed from the screen	n/a