Widgets API Reference

App

2/23/2026

Contents

- Developer

Learn how to control your widgets.

Related documentation:
  -

Overview

App is the main controller for Genesys Widgets and has no UI. It controls all startup routines, global configurations, and extensions, and it executes the onReady event and distributes changes to theme, language, mobile mode, and other application-wide effects.

Usage

App's main interface is its configuration. You set all global defaults using the **window._genesys.widgets.main** property. App also has a few commands you can use to change the language and theme.

Customization

App itself cannot be customized, but its *configuration options affect all widgets*.

Mobile support

App has built-in mobile detection and can automatically notify all widgets to switch to mobile mode. You can also control this manually.

Configuration

Description

App uses the configuration property **_genesys.widgets.main**. App controls the Genesys Widgets product as a whole, handling themes, languages, and mobile devices.

Example

```
window._genesys.widgets = {

    main: {
            theme: 'dark',
```

```
        themes: {

                dark: 'cx-theme-dark',
                light: 'cx-theme-light',
                blue: 'cx-theme-blue',
                red: 'cx-theme-red'
        },
        lang: 'en',
        i18n: 'i18n.json',
        mobileMode: 'auto',
        mobileModeBreakpoint: 600,
        debug: true,
        downloadGoogleFont: true,
        googleFontUrl: 'https://apps.mypurecloud.com/webfonts/roboto.css',
        header: {'Authorization': 'value'},
        cookieOptions: {
                secure: true,
                domain: 'genesys.com',
                path: '/',
                sameSite: 'Strict'
        }
    },
    onReady: function(){

        // Do something on Widgets ready
    }
}
```

Options

Name	Type	Description	Default	Required	Introduced/ Updated	
main.themes	object	An object list containing the CSS classname for each theme. The property names are used to select the theme in the theme property, for example {dark: cx-theme-dark, light: cx-theme-light, red: cx-theme-red, blue: cx-theme-blue}. Where dark and light	{dark: 'cx-theme-dark', light: 'cx-theme-light'}	n/a		

Name	Type	Description	Default	Required	Introduced/ Updated	
		are the built-in themes provided in Genesys Widgets, and red and blue are example custom theme names you may create on your own. **Important** It is not necessary to define the dark and light theme as shown in this example. It is included to help show how the formatting works. Whatever you put in this object will be merged with the default themes object internally.				
main.theme	string	Selects the theme to apply to Genesys Widgets from the Themes object. Uses the property name of the theme, for example using the example from themes above, possible values for this could be dark, light, red, blue.	dark	n/a		

Name	Type	Description	Default	Required	Introduced/ Updated	
main.lang	string	Select the language to use from the i18n language pack. Language codes are selected by the customer. Any language code format can be used as long as this property matches one of the language codes in your i18n language pack. For more information about localization, see localization.	en	n/a		
main.i18n	URL string or JSON	Either a path to a remote i18n.json language pack file or an inline JSON language pack definition. For more information about language packs, see localization.	en	Default English language strings are built into each widget and are displayed by default. Defining this i18n language pack overrides the built-in strings.	n/a	
main.header	object	An object containing a key value pair for the authorization header.	n/a	n/a	9.0.002.06	
	array	**Note**: For	none	When lazy		

Name	Type	Description	Default	Required	Introduced/ Updated	
main.preload		use with lazy loading only. A list of plugins you want pre-loaded at startup. You may want certain plugins, such as SideBar, to be shown on screen as soon as possible; to do so, you may add 'sidebar' to this preload plugins array so it will be loaded after Widgets starts up. The names you add to the list must match the first part of the plugin filename you wish to load. Example: sidebar will load **sidebar.min.js** from the **plugins/** folder. All filenames are lowercase. Important This preload array is intended for use when running widgets in lazy loading mode. You may also use this to pre-		loading Widgets		

Name	Type	Description	Default	Required	Introduced/ Updated	
		load your own custom-made plugins.				
main.mobileMode	boolean/ string	Mobile Mode setting. `true` = Force Mobile Mode on all devices. `false` = Disable Mobile Mode completely. `auto` = Genesys Widgets automatically switches between mobile and desktop ,odes using the `mobileModeBreakpoint` property and **UserAgent** detection.	auto	n/a		
main.timeFormat	number/ string	This sets the time format for the timestamps. It can be 12 or 24.	12	n/a		
main.mobileModeBreakpoint	number	The breakpoint width in pixels where Genesys Widgets will switch to Mobile Mode. Breakpoint checked at startup only.	600	n/a		
main.debug	boolean	Enable debug logging from the bus to appear in the browser console.	false	n/a		
main.customStylesheetID	string	The HTML ID of a	n/a	n/a		
main.downloadGoogleFont	boolean	By default, Genesys	true	n/a		

Name	Type	Description	Default	Required	Introduced/ Updated	
		Widgets downloads and uses the Google font Roboto. To disable this download, set value `false`.				
main.googleFontUrl	string	The string used to refer the URL where the Google fonts are hosted in Genesys Hosted Repository. You can configure one of the Genesys Hosted region font URLs specified here Genesys Web Fonts. **Important** This Option is only applicable when the downloadGoog option is set to `true`.	https://apps.mypurecloud.com/ webfonts/ roboto.css	n/a	9.0.018.00	
main.deploymentID	string	The string used to customize cookie names so that multiple Widgets deployments can run in the same domain.	n/a	n/a	9.0.006.02	
main.cookieOptions	object	An object containing cookie attributes that applies	{sameSite:'Strict'}	n/a	9.0.017.01	

Name	Type	Description	Default	Required	Introduced/ Updated
		globally to all Widgets. The following cookie attributes are supported: 1. secure - Either `true` or `false`, indicating if the cookie transmission requires a secure protocol (https). 2. domain - A string indicating a valid domain where the cookie should be visible. 3. path - A string indicating the path where the cookie is visible. 4. expires - Specifies the number of days, either from time of creation or from a date instance,			

Name	Type	Description	Default	Required	Introduced/ Updated
		until the cookie is to be removed. domain and path can be used to make cookies compatible with environments that use a non FQDN URL, such as an intranet hostname. However, the domain should only be manually set in production if the automated values are causing problems. Otherwise, rely on the automated domain and path. 5. sameSite - This maps to the cookie SameSite attribute allowing the cookie to			

Name	Type	Description	Default	Required	Introduced/ Updated
		be restricted to a first-party or same-site context. It can take any of the supported values that SameSite attribute takes.			
		Importa The values are automatically set by Widgets to support cross-sub-domain cookies. Modifying these options overrides the automated values and might break cross-sub-domain cookie support if not properly set. For usage, please refer to the above example			
onReady	function	A callback function that	none	n/a	

Name	Type	Description	Default	Required	Introduced/Updated	
		is invoked when the Widgets are ready and initialized with the configuration provided.				

Localization

No 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 Widgets Extensions for more information about extending Genesys Widgets.

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

oMyPlugin.command('App.themeDemo');
```

setTheme

Sets the theme for Genesys Widgets from the list of registered themes. Default themes are *light* and *dark*. You can register as many new themes as you need.

Example

```
oMyPlugin.command('App.setTheme', {theme: 'light'}).done(function(e){

        // App set theme successfully

}).fail(function(e){

        // App failed to set theme
});
```

Options

Option	Type	Description
theme	string	Name of the theme you want to use. This name is specified in **window._genesys.main.themes**. Default themes are light and dark.

Resolutions

Status	When	Returns
resolved	Theme exists and is successfully changed	The name of the theme that was chosen, for example *light*.
rejected	Theme does not exist	'Invalid theme specified'.

getTheme

Get the CSS classname for the currently selected theme.

Example

```
oMyPlugin.command('App.getTheme').done(function(e){

        // App got theme successfully
        // e == CSS classname for current theme

}).fail(function(e){

        // App failed to get theme
});
```

Resolutions

Status	When	Returns
resolved	Always	CSS classname for the currently selected theme, for example, *cx-theme-light.*
rejected	Never	n/a

reTheme

Accepts an HTML reference (either string or jQuery wrapped set) and applies the proper CSS theme classname to that HTML and returns it back. When widgets receive the **theme** event from App, they pass in their UI containers into App.reTheme to have the old theme classname stripped and the new

classname applied.

Example

```
oMyPlugin.command('App.reTheme', {html: '
Test Theme
'}).done(function(e){

        // App set theme successfully

}).fail(function(e){

        // App failed to set theme
});
```

Options

Option	Type	Description
html	string or jQuery Wrapped Set	HTML string or jQuery Wrapped Set you want to have modified.

Resolutions

Status	When	Returns
resolved	HTML is provided and theme is updated	HTML that was passed-in and modified
rejected	No HTML is provided	'No HTML provided by [plugin name]'

themeDemo

Start an automated demo of each theme. All registered themes will be applied with a default delay between themes of 2 seconds. You can override this delay. This command is useful for comparing themes or testing themes with official or custom widgets.

Example

```
oMyPlugin.command('App.themeDemo', {delay: 1000}).done(function(e){

        // App demo successfully started

}).fail(function(e){

        // App failed to start demo
});
```

Options

Option	Type	Description
delay	number	Number of milliseconds between theme changes. Default value is

Option	Type	Description
		2000 milliseconds.

Resolutions

Status	When	Returns
resolved	Always	n/a
rejected	Never	n/a

setLanguage

Changes the language

Example

```
oMyPlugin.command('App.setLanguage', {lang: 'eng'}).done(function(e){

        // App set language successfully started

}).fail(function(e){

        // App failed to set language
});
```

Options

Option	Type	Description
lang	string	Change the language of Genesys Widgets. Switches all strings in Widgets to selected language.

Resolutions

Status	When	Returns
resolved	Language successfully changed	n/a
rejected	No language code is provided	No language code provided
rejected	No matching language code is specified in your language pack	No matching language code found in language pack

closeAll

Publishes the 'App.closeAll' event that requests all widgets to close.

Example

```
oMyPlugin.command('App.closeAll').done(function(e){
```

```
        // App closed all successfully

}).fail(function(e){

        // App failed to close all
});
```

Resolutions

Status	When	Returns
resolved	Always	n/a
rejected	Never	n/a

updateAJAXHeader

Introduced: 9.0.002.06

Updates the Authorization header.

Example

```
_genesys.widgets.bus.command('App.updateAJAXHeader', {header:

        {'Authorization': 'value'}

});
```

Resolutions

Status	When	Returns
resolved	Header is updated	n/a
rejected	Never	No request header found

removeAJAXHeader

Introduced: 9.0.002.06

Removes the set Authorization header.

Example

```
_genesys.widgets.bus.command('App.removeAJAXHeader');
```

Resolutions

Status	When	Returns
resolved	Always	n/a

registerExtension

Introduced: 9.0.002.06

Allows you to register and initialize new extensions at runtime instead of predefining extensions before Genesys Widgets starts up.

Options

Option	Type	Description
undefined	function	Your extension function. Receives the following arguments: $ (jQuery), CXBus, Common

Resolutions

Status	When	Returns
resolved	Valid 'extension' object provided	n/a
rejected	Invalid 'extension' option provided	n/a

registerAutoLoad

(For use with lazy loading only) Allows you to register a plugin into the preload plugins array so that it can be pre-loaded at the startup rather than lazy loading later. This can be useful when there is an active session maintained by your Widget and you would like to show it immediately at startup during page refresh or navigating across pages.

```
Note: This command is intended for use when running widgets in lazy loading mode.
You may also use this to register and pre-load your own custom-made plugins.
```

Options

Option	Type	Description
name	string	The name of the plugin that needs to be registered for auto loading.

Resolutions

Status	When	Returns
resolved	A plugin is added into the preload list	n/a
rejected	Never	n/a

deregisterAutoLoad

(For use with lazy loading only) Allows you to de-register a plugin from the preload plugins array so that it will not be pre-loaded at startup. This can be useful when there is no more active session

maintained by your Widget and you don't want to show it on the screen immediately at startup.

```
Note: This command is intended for use when running widgets in lazy loading mode.
You may also use this to de-register your own custom-made plugins.
```

Options

Option	Type	Description
name	string	The name of the plugin that needs to be de-registered from auto loading.

Resolutions

Status	When	Returns
resolved	A plugin is removed from the preload list	n/a
rejected	Never	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, do not use the global bus object to register your custom plugins. Instead, see Widgets Extensions for more information about extending Genesys Widgets.

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

oMyPlugin.subscribe('App.ready', function(e){});
```

Name	Description	Data
ready	CallUs is initialized and ready to accept commands.	
i18n	Published when the language for Genesys Widgets is changed or is being set for the first time.	'(language code)'
theme	Published when the theme for Genesys Widgets is changed or is	{theme: '(theme CSS classname)'}

Name	Description	Data
	being set for the first time.	
timeFormat	Published when the time format for Genesys Widgets is changed or is being set for the first time.	{timeFormat: iTimeFormat}