Widgets API Reference

WindowManager

9/7/2025

Contents

- Developer

Learn how to use the WindowManager plugin, which provides a controller for several different types of window group.

Related documentation:

-

Overview

The WindowManager plugin provides a controller for several different types of window group. HTML UIs added to these WindowManager groups are arranged and managed in accordance with each group's purpose.

One group type is *Dock View*, which appears as a toast-like UI docked in the lower-bottom-right of the screen. This group automatically stacks widgets horizontally. When one of the widgets closes, the stack collapses toward the right. Widgets can register themselves into this WindowManager group and let it do all the work.

Another group type is *Side Button*, with its launcher button on the right side of the screen. Like the dock view, buttons are stacked, but in this case they are stacked vertically. As buttons are added and removed from the group, the button stack collapses to fill in the gaps.

Usage

WindowManager has "register" commands for registering your UI into different groups. They all accept one argument, the HTML you want to be handled by WindowManager. You can use 'registerDockView' or 'registerSideButton' at this time. More window management groups will be added in upcoming releases.

Customization

WindowManager does not have customization options.

Screenshot

-

Configuration

WindowManager does not have configuration options.

Localization

WindowManager 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('WindowManager.registerDockView', {html: '
HTML
'});
```

registerDockView

Creates a docked view container to show a widget on the bottom right corner. Its position is adjusted (stacked) to appear beside another widget if already present and is indexed with a tabindex.

Example

```
oMyPlugin.command('WindowManager.registerDockView', {html: '
Template
'}).done(function(e){

        // WindowManager registered a dockView successfully

}).fail(function(e){

        // WindowManager failed to register a dock view
});
```

Options

Option	Type	Description
html	string	A Widget HTML string template that needs to be shown in dock view.

Resolutions

Status	When	Returns
resolved	The HTML template is successfully opened and registered in dock view.	n/a
rejected	No HTML template is found.	'No html content'

registerSideButton

Registers a button to show on the right side of the screen for a particular plugin. Its position is based on the respective plugin order defined in the array configuration. Currently, this is not supported for external plugins.

Example

```
oMyPlugin.command('WindowManager.registerSideButton', {template: '
Button Text
'}).done(function(e){

        // WindowManager registered a side button successfully

}).fail(function(e){

        // WindowManager failed to register a side button
});
```

Options

Option	Type	Description
template	string	Custom HTML string template for a button.

Resolutions

Status	When	Returns
resolved	The HTML button is successfully registered.	n/a

Status	When	Returns
rejected	No HTML template is found.	'No button template found to register'

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

Name	Description	Data
ready	WindowManager is initialized and ready to accept commands.	n/a
changed	WindowManager publishes this event when there is any change in the position of widgets on the screen.	{registry: (object)}