Widgets Developer Resources

Common

12/9/2023

Contents

- Developer

Learn how to access Widgets utility functions and dynamically generate the common HTML containers used throughout Genesys Widgets.

Related documentation:

-

Common is a utility object available for import into Plugins/Widgets and Extensions. It is also accessible directly from the path **window._genesys.widgets.common**.

Common provides utility functions and dynamically generates common HTML Containers used throughout Genesys Widgets.

For all examples below, assume that **_genesys.widgets.common** has been stored in a local variable named *Common*.

```
var Common = _genesys.widgets.common;
```

Common.Generate.Container({options})

Dynamically generates a new HTML Container in matching the style of Genesys Widgets with the selected components you request in your options object. Returns the generated container HTML as a jQuery wrapped set.

Example

'Generate an Overlay Container'

```
var ndContainer = Common.Generate.Container({

        type: 'overlay',
        title: 'My Overlay',body: 'Some HTML here as a string or jQuery wrapped set',
        icon: 'call-outgoing',
        controls: 'close',
        buttons: false
}),
```

'Generate a Toast Container'

```
var ndContainer = Common.Generate.Container({

        type: 'generic',
        title: 'My Toast',body: 'Some HTML here as a string or jQuery wrapped set',
        icon: 'chat',
        controls: '',
        buttons: {
```

```
                type:'binary',
                primary: 'OK',
                secondary:'cancel'
        }
}),
```

Arguments

Argument	Type	Description
options	object	An object containing options to apply to the generated container.
options.type	string	`generic` or `overlay`. Overlay containers have special CSS properties for appearing inside the Overlay widget. Default is `generic`.
options.title	string	Title to apply to the container's titlebar area.
options.body	string or jQuery wrapped set	The HTML body you want the container to wrap.
options.icon	string	CSS Classname of icon to use.
options.controls	string	Select from a set of window control buttons to show at the top right. `close` = Show only the close button. `minimize` = Show only the minimize button. `all` = Show both close and minimize buttons.
options.buttons	object	Options for displaying action buttons at the bottom of the container, such as **OK** and **Cancel** buttons.
options.buttons.type	string	Currently, **binary** is the only supported button set at this time. Additional sets and arrangements will be available in a later release. Pass binary as the type here if you wish to show typical accept and dismiss buttons.
options.buttons.primary	string	Display name on the primary button. (for example **OK, Yes, Accept, Continue**, etc.)
options.buttons.secondary	string	Display name on the secondary button. (for example **Cancel, No, Dismiss, Reject**, etc.)

Common.Generate.Buttons({options})

Dynamically generates a new HTML Binary Button set in matching the style of Genesys Widgets with the selected options in your options object. Returns the buttons as a jQuery wrapped set.

Example

'Generate Binary Buttons'

```
var ndButtons = Common.Generate.Buttons({

        type: 'binary',
        primary: 'OK',
        secondary: 'Cancel'
}),
```

Arguments

Argument	Type	Description
options	object	Options for generating buttons, such as **OK** and **Cancel** buttons.
options.type	string	Currently **binary** is the only supported button set at this time. Additional sets and arrangements will be available in a later release. Please pass `binary` as the type here if you wish to show typical **accept** and **dismiss** buttons.
options.primary	string	Display name on the primary button. (for example **OK, Yes, Accept, Continue**, etc.)
options.secondary	string	Display name on the secondary button. (for example **Cancel, No, Dismiss, Reject**, etc.)

Common.Generate.Icon(name)

Dynamically generates an icon from the included icon set. Icons are in SVG format.

Example

'Generate Chat Icon'

```
var ndChatIcon = Common.Generate.Icon('chat');
```

'Insert Chat Icon'

```
$('#your_icon_container').append(Common.Generate.Icon('chat'));
```

Arguments

Argument	Type	Description
name	string	Select the icon you want to generate by name. See the icon reference page for icon names.

Common.Generate.Scrollbar(element, {options})

Dynamically generates a widget scrollbar for selected DOM element.

Example

'Generate Scrollbar for a container'

```
var scrollContainer = Common.Generate.Scrollbar($('#your_container'))
```

Arguments

Argument	Type	Description
element	DOM element or jQuery selector	Select the element to which you would like to apply scrollbar.
options	object	This is an iScroll component. So, all the options that iScroll supports can be passed here. For more details, refer to: http://iscrolljs.com/#configuring

Common.config(object)

Configure some debug options for Common at runtime.

Example

'Enable full debug logging'

```
Common.config({debug: true, debugTimestamps: true});
```

Arguments

Argument	Type	Description
object	object	Supported options are **debug** and **debugTimestamps**. Setting debug to `true` will enable debug messages created by **Common.log()**. Setting **debugTimestamps** to `true` will add timestamps to the front of each debug message created by **Common.log()**. Default value for both is `false`.

Common.checkPath(object, path)

Check for the existence of a sub-property of an object at any depth. Returns the value of that property; if found otherwise it returns **undefined**. Useful for checking configuration object paths without having to check each sub-property level individually.

Example

'Check for window._genesys.main'

```
var oMainConfig = false;

if(oMainConfig = Common.checkPath(window, '_genesys.main')){
        //... Utilize oMainConfig
}
```

Arguments

Argument	Type	Description
object	object	An object you want checked for a particular sub-property at any depth.
path	string	The object path in dot notation you wish to search for.

Common.createPath(object, path, value)

Related to checkPath, createPath lets you specify a target object and path string but lets you create the path and set a value for it. This saves you the pain of defining each node in the path individually. All nodes in your path will be created as objects. Your final node, the property you are trying to create, will be whatever value you assign it.

Example

'Create window._genesys.main'

```
var oMainConfig = false;

if(oMainConfig = Common.createPath(window, '_genesys.main', {debug:true})){
        //... Utilize oMainConfig
}
```

Arguments

Argument	Type	Description
object	object	An object you want to add your new path to.
path	string	The object path in dot notation you wish to create.
value	any	The value you want to assign to the final node (property) in your path.

Common.linkify(string, options)

Search for and convert URLs within a string into HTML links. Returns transformed string.

Example

'Check for window._genesys.main'

```
var sString = 'Please visit www.genesys.com';
sString = Common.linkify(sString, {target: 'self'});
// sString == 'Please visit www.genesys.com
```

Arguments

Argument	Type	Description
string	string	Any string you want to check for URLs and have them converted.
options	object	A list of options to apply to the linkify operation.
options.target	string	Choose the HTML TARGET attribute to apply to the generated links. Default is **_blank**. Set this option to `self` to apply the target **_self** to the generated links.

Common.log(mixed, type)

Log something to the browser's console. When using Common.log, _genesys.main.debug must be set to true to see your logs. This allows you to add debug logging to your code without worrying about unwanted debug messages in production. If timestamps are enabled, they will be prefixed to all messages printed through Common.log.

Example

'Check the contents of window._genesys.main'

```
var Common = _genesys.widgets.common;
Common.log(window._genesys.main);

if(!window._genesys.main){
        Common.log('window._genesys.main is not defined', 'error');
}
```

Arguments

Argument	Type	Description
mixed	Any	Any value or message you'd like to log.
type	string	You can specify the log type, such as `log`, `debug` and `error`. Default type is `log`. Note, if your browser doesn't support the `debug` or `error` log type, use `log` instead.

Common.sanitizeHTML(string)

Search for and escape characters within a string. Returns transformed string. Useful for escaping HTML.

Example

'Check for window._genesys.main'

```
var sString = 'Please visit www.genesys.com';

sString = Common.sanitizeHTML(sString);

 // sString == 'Please visit <a href='www.genesys.com' target='_self'>www.genesys.com</a>''
```

Arguments

Argument	Type	Description
string	string	Any string you want to be transformed.

Common.updateTemplateI18n(element, object)

Searches through an element's contents for i18n string elements to update with new strings. Used when updating the language in real-time. Works by searching for elements with the CSS classname 'i18n' and reading the custom attribute 'data-message' to match the string name in the language object. See example below.

Example

'Check for window._genesys.main'

```
var ndContainer = $('

');

Common.updateTemplateI18n(ndContainer, {CustomButton001: 'Accept'});

// ndContainer ==
Accept
```

Arguments

Argument	Type	Description
element	jQuery wrapped set	Element you want to search within to replace i18n strings.
object	Object of i18n Strings	The list of languages strings you want to update your UI with. This object comes from the App.i18n event or you can define your own custom object inline or using some other system. Object format is a simple name:value pair format. The **data-message** attribute on your HTML element must match one of these property names to be updated.

Common.debugIcons

Returns the list of all the Icons with their names that Widgets support.

Example

'Fetch and Display list of icons present in Widgets'

```
Common.debugIcons()
```

Common.debug

Adds debug logs in to the browser's console. When using Common.debug, _genesys.main.debug must be set to true to see your logs. This allows you to add debug logging to your code without worrying about unwanted debug messages in production. If timestamps are enabled, they will be prefixed to all messages printed through Common.debug.

Example

'Check the File upload limits in WebChatService'

```
Common.debug(data_server_returned_file_limits);
```

Arguments

Argument	Type	Description
mixed	Any	Any value or message you'd like to add debug log. **Note**: This is only supported if your browser supports debug log type.

Common.error

Adds error logs in to the browser's console. When using **Common.error**, **_genesys.main.debug** must be set to `true` to see your logs. This allows you to add error logging to your code without worrying about unwanted error messages in production.

Example

'Logging error messages'

```
Common.error('A widget plugin did not receive the following config: ....');
```

Arguments

Argument	Type	Description
mixed	Any	Any value or message you'd like to add error log. **Note**: This is only supported if your browser supports error log type.

Common.populateAllPlaceholders

Adds place holder content to the input elements in a form with the given text strings.

Example

'Show placeholders strings in a form'

```
Common.populateAllPlaceholders($('#your_form'), {strings})
```

Arguments

Argument	Type	Description
Form Selector	jQuery DOM selector for a form	Form containing input elements. Note: Input elements should contain i18n class name and data attribute **data-message-type** with value `placeholder` for the place holder details to appear.
Key/Value pairs	object	Placeholder messages that needs to be displayed. This is an object with key-value pairs where key should be equal to the **data-message** attribute value of an input element and value can be any text that you would like to display.

Common.populateLanguageStrings

Adds the preferred language place holder text to the given input elements in a form.

Example

'Show placeholders strings in a form'

```
Common.populateLanguageStrings($('#your_form'), {strings})
```

Arguments

Argument	Type	Description
Form Selector	jQuery DOM selector for a form	Form containing input elements. **Note**: Input elements should contain i18n class name and data attribute **data-message-type** with value `placeholder` for the place holder details to appear.
Key/Value pairs	object	Placeholder messages that needs to be displayed. This is an object with key-value pairs where key should be equal to the **data-message** attribute value of an input element and value can be any text that you would like to display.

Common.populateIcons

Show all the Icons on a Widget.

Example

'Populate all Widget Icons'

```
Common.populateIcons($('#your_continer'));
```

Arguments

Argument	Type	Description
element	jQuery DOM selector	Specify the widget container for which all the icons have to be displayed.

Common.insertIcon

Adds an icon before the selected element.

Example

'Insert a check mark icon to an element you desire.'

```
Common.insertIcon($('#your_element'), 'alert-checkmark', 'alert')
```

Arguments

Argument	Type	Description
element	jQuery DOM selector	An html element to which icon is to be displayed.
icon name	string	Name of the icon that you would like to display. **Note**: Refer to **Common.debugIcons** method to find out all the icon names that widgets support.
icon Aria Name	string	Name for the icon to be read by screen readers.

Common.injectScript

Injects javascript code dynamically into widgets with the help of a script tag.

Example

'Inject your Widget WebChat extension plugin.'

```
Common.injectScript('path/to/LoadWebChat.ext.js')
```

Arguments

Argument	Type	Description
Script file name	string path to JavaScript file	JavaScript file name that needs to be injected into widgets.

Common.mobileScreenScale

Re-sizes and fits widget to any mobile screen.

Example

'Fit your widget to any mobile screen.'

```
var mobileScaledWidget = Common.mobileScreenScale($('#your_widget'));
```

Arguments

Argument	Type	Description
element	jQuery DOM Selector	Your main widget wrapper container selector that contains the entire widget with **cx-titlebar, cx-body, cx-footer, cx-button-container** and **cx-message-container** classes in it.

Common.showLoading

Show loading spinner Icon.

Example

'Show loading spinner during an Ajax request'

```
Common.showLoading($('#your_container'))
```

Arguments

Argument	Type	Description
element	jQuery DOM Selector	An html container where loading spinner should appear. This adds a class name **cx-loading**.

Common.hideLoading

Remove loading spinner Icon.

Example

'Remove loading spinner after the Ajax request'

```
Common.hideLoading($('#your_container'))
```

Arguments

Argument	Type	Description
element	jQuery DOM Selector	An html container that contains the loading spinner.

Common.showWaiting

Show waiting icon.

Example

'Show waiting Icon when uploading a file.'

```
Common.showWaiting($('#your_container'),'waiting'))
```

Arguments

Argument	Type	Description
element	jQuery DOM Selector	An html container where waiting symbol should appear. This adds a class name **cx-waiting**.
Aria Label	string	The value of the **aria-label** attribute for the loading screen icon. The default value is `waiting`.

Common.hideWaiting

Remove waiting icon.

Example

'Remove waiting icon after file upload is done.'

```
Common.hideWaiting($('#your_container'))
```

Arguments

Argument	Type	Description
element	jQuery DOM Selector	An html container that contains the waiting symbol.

Common.watch

Repeat your function execution for every 'x' milliseconds (default 1 second) up to a maximum number of times (default - infinite) or till your function returns true.

Example

'Make Request Notifications until none are pending.'

```
Common.watch(function(iteration, maxIterations){

        if(bRequestNotificationsPending){
         // ..POST Request
        }
        return !bRequestNotificationsPending;
}, 3000, 30)
```

Arguments

Argument	Type	Description
function name	function	The function that you would like to execute. It should return `true/false`.
frequency	milliseconds	Execute the function for every *x* milliseconds until it returns `true`.
limit	number	The maximum number of times function is executed.

Common.addDialog

Create your own dialog box and append it in to the widget.

Example

'Add a dialog box on your preferred container div

```
Common.addDialog($('#your_container'), $('#your_dialog_box'), 'my_warning')
```

Arguments

Argument	Type	Description
element	jQuery selector	The parent container that holds the dialog box.
element	jQuery selector	The actual dialog box that you would like to display. This should contain the **data-dialog** attribute with the value equal to the dialog box name.
name	string	Dialog box name.

Common.showDialog

Show the dialog box that you prefer, using the dialog box name created with **Common.addDialog()**.

Example

'Show the dialog box created using Common.addDialog()'

```
Common.showDialog($('#your_container'), 'your_dialog_box_name');
```

Arguments

Argument	Type	Description
element	jQuery Selector	The parent container which has the dialog box appended in to it.
name	string	The actual dialog box name.

Common.hideDialog

Hide the dialog box that you showed using **Common.showDialog()**.

Example

'Hide dialog box'

```
Common.hideDialog($('#your_container'), 'your_dialog_box_name);
```

Arguments

Argument	Type	Description
element	jQuery Selector	The parent container that is showing the dialog box.
name	string	The actual dialog box name.

Common.hideDialogs

Hide all the dialog boxes. Dialog box name is not needed here.

Example

'Hide all dialog boxes.'

```
Common.hideDialogs($('#your_container'));
```

Arguments

Argument	Type	Description
element	jQuery Selector	The parent container that is showing all the dialog boxes.

Common.showAlert

Show a native alert dialog box on the widget you prefer with your own text message. By default, a primary button is added to dismiss the alert dialog.

Example

Show an alert dialog box on the Widget you prefer. But default it adds the dismiss button.

```
Common.showAlert($('.cx-widget.cx-webchat'), {text: 'your alert message', buttonText: 'Ok'})
```

Arguments

Argument	Type	Description
element	jQuery selector	The widget plugin container that should display the alert dialog. This should be the top level container wrapper holding the widget.
options	object	The data options containing the text to be shown on the Alert dialog box.
options.text	string	Display text on the Alert dialog box.
options.buttonText	string	Display text on the primary button (for example: **OK**).

Common.bytesToSize

Convert any number in bytes to Kilobytes, Megabytes, Gigabytes and Terabytes.

Example

'bytes to KB, MB, GB or TB.'

```
var fileSize = Common.bytesToSize(parseInt(fileSizeInBytes));
```

Arguments

Argument	Type	Description
bytes	number	Number in bytes size.

Common.getFormattedTime

Returns time in 12-hour or 24-hour format from the actual date timestamp. If no timestamp is provided, it uses current time.

Example

'convert date timestamp to return time in 12 hrs format'

```
var formattedTime = Common.getFormattedTime(timestamp, 12);
```

Arguments

Argument	Type	Description
timestamp	Date	JavaScript Date timestamp object.
format	number	Time format with value 12 or 24.