

SF.UI.WIDGET

**** This class will not be supported in 2015.**

All functionalities of *sf.ui.widget* class are more improved, integrating with CAPH. Therefore *sf.ui.widget* class is not supported since 2015 Smart TV. To use functions of *sf.ui.widget* class, refer to [here](#).

This is the base class for all widgets.

Add the following line for *sf.ui.widget* class into a html file your own :

```
<script type="text/javascript" src="$MANAGER_WIDGET/Common/af/2.0.0/loader.js"></script>
```

You can declare *sf.ui.widget* class like this :

```
ex) var widget = sf.ui.widget;
```

Contents

Properties

Methods

[addCallback](#)

[destroy](#)

[extend](#)

[hide](#)

[option](#)

[removeAllCallbacks](#)

[removeCallback](#)

[show](#)

[widget](#)

Properties

Name	Type	Value	Description
baseCSSClass	String		Base CSS class added to the widget instance
eventNamespace	String		Event namespace added by default to all the events triggered by the widget
widgetName	String		The Widget name

Methods

addCallback

Description

This function adds the callback function for the specified event.

Parameters	<ul style="list-style-type: none">■eventName<ul style="list-style-type: none">- String Object- Name of the event or literal object with pairs eventName: Function Array<Function>. It should be event name without eventNamespace■Fn<ul style="list-style-type: none">- Function Array<Function>- Callback function or Array of callback functions
Return	<ul style="list-style-type: none">■Object<ul style="list-style-type: none">- this
Emulator Support	N
SDK Constraint	None

Example

Add more callbacks for various events.

```
$("#instance").myWidget("addCallback", {  
  "show": function () {  
    alert("Can You see me?");  
  },  
  "beforeshow": foo,  
  "hide": [foo, Obj.bar]  
});
```

destroy

Description

This function destroys the widget instance.

Parameters	none
Return	<ul style="list-style-type: none">■Object<ul style="list-style-type: none">- this
Emulator Support	N
SDK Constraint	None

Example

//Destroy widget instance.

```
$("#instance").myWidget("destroy");
```

extend

Description

This function extends the widget instance.

Parameters	<ul style="list-style-type: none">■name<ul style="list-style-type: none">- String Object- Property name or literal object of name:value pairs■Value<ul style="list-style-type: none">- Function Array<Function>- Callback function or Array of callback functions
Return	<ul style="list-style-type: none">■Object<ul style="list-style-type: none">- this
Emulator Support	N
SDK Constraint	None

Example

<pre>//Extend widget instance \$("#instance").myWidget().myWidget("extend", "newFunction", function (value) { alert(value); return this; // let's make new function chainable }) .myWidget("newFunction", "Is there anybody out there?");</pre>	
hide	
Description	
This function hides the element returned by the widget function.	
Parameters	none
Return	<div>■Object</div> <div>- this</div>
Emulator Support	N
SDK Constraint	None
Example	
<pre>//Hide element \$("#instance").myWidget("hide");</pre>	
option	
Description	
This function is the setter/getter for properties of "options" object.	
Parameters	<div>■name</div> <div>- String Object</div> <div>- Name</div> <div>■value</div> <div>- Any type</div> <div>- Value</div>
Return	<div>[get option]</div> <div>■Any Types</div> <div>- if 'value'; is not defined, returns the value.</div> <div>[set option]</div> <div>■Object</div> <div>- if 'value' is defined, returns this.</div>
Emulator Support	N
SDK Constraint	None
Example	
<pre>//Set value for bigValue property \$("#instance").myWidget("option", "bigValue", 12345678);</pre>	
removeAllCallbacks	
Description	
This function removes all the callbacks. Only the callbacks added with the addCallbacks function, or passed on initialization of instance, are removed.	
Parameters	none
Return	<div>■Object</div> <div>- this</div>
Emulator Support	N

SDK Constraint	None
Example	
<pre>//Remove all callbacks for widget instance \$("#instance").myWidget("removeAllCallbacks");</pre>	
removeCallback	
Description	
<p>This function removes the specified callback or all the callbacks for the specified event. Only the callbacks added with the addCallbacks function, or passed on initialization of the instance are removed with this function. If fn is undefined, all the callbacks for the specified event are removed.</p>	
Parameters	<ul style="list-style-type: none"> ■eventName <ul style="list-style-type: none"> - String - Name of event without 'eventNamespace' ■Fn <ul style="list-style-type: none"> - Function - Callback function to remove
Return	<ul style="list-style-type: none"> ■Object <ul style="list-style-type: none"> - this
Emulator Support	N
SDK Constraint	None
Example	
<p>[Example 1]</p> <pre>//Remove specific callback for event 'show' function onShow() { alert("Can you see me?"); } \$("#instance").myWidget("addCallback", "show", onShow); ... \$("#instance").myWidget("removeCallback", "show", onShow);</pre> <p>[Example 2]</p> <pre>//Callbacks passed "on the fly" to addCallback function can not be removed \$("#instance").myWidget("addCallback", "show", function () { alert("Can you see me?"); }); // This will not work \$("#instance").myWidget("removeCallback", "show", function () { alert("Can you see me?"); });</pre>	
show	
Description	
<p>This function shows the element returned by the widget function.</p>	
Parameters	none
Return	<ul style="list-style-type: none"> ■Object <ul style="list-style-type: none"> - this
Emulator Support	N
SDK Constraint	None
Example	
<pre>\$("#instance").myWidget("show");</pre>	

widget

Description

This function returns the DOM element (wrapped as jQuery object) representing the widget instance. It is, by default, the element passed to the constructor.

Parameters	none
Return	■Object - jQuery object
Emulator Support	N
SDK Constraint	None

Example

```
var widgetHandler = $("#instance").myWidget("widget");
```