

JSWebComponent

 Apr 19, 2024 02:59

Supported Clients

NGClient

Extends

JSComponent

Property Summary

Number	anchors	Enables a component to stick to a specific side of form and/or to grow or shrink when a window is resized.
CSSPosition	cssPosition	CSS position is a replacement for anchoring system making it more intuitive to place a component.
Boolean	enabled	The enable state of the component, default true.
Number	formIndex	The Z index of this component.
Number	height	The height in pixels of the component.
String	name	The name of the component.
String	typeName	The webcomponent type (name from the spec).
Boolean	visible	The visible property of the component, default true.
Number	width	The width in pixels of the component.
Number	x	The x coordinate of the component on the form.
Number	y	The y coordinate of the component on the form.

Methods Summary

Object	getAttribute(name)	Get the value of an attribute of the element.
Array	getAttributes()	Returns the attribute names of an element.
String	getComment()	Returns the comment of this component.
Object	getDesignTimeProperty(key)	Get a design-time property of an element.
Array	getDesignTimePropertyNames()	Get the design-time properties of an element.
String	getFormName()	Returns the name of the form.
JSMETHOD	getHandler(handlerName)	Returns the JSMETHOD handler with the given name.
Object	getJSONProperty(propertyName)	Get the design-time value of the given property.
Array	getJSONPropertyNames(includeAll)	Get the list of property names this component supports.
UUID	getUUID()	Returns the UUID of this component.
Boolean	isJSONPropertySet(propertyName)	Returns whatever the design-time value of the given property was set.
Object	putDesignTimeProperty(key, value)	Set a design-time property of an element.
String	removeAttribute(name)	Remove the attribute of an element.
Object	removeDesignTimeProperty(key)	Clear a design-time property of an element.
void	resetHandler(handlerName)	Similar to resetJSONProperty but for handlers.
void	resetJSONProperty(propertyName)	Reset the design-time value of the given property.
void	setAttribute(name, value)	Set the attribute value of an element.
void	setHandler(handlerName, method)	Set the JSMETHOD handler for the given handler name.
void	setJSONProperty(propertyName, value)	Set the design-time value for the given property.

Property Details

anchors

Enables a component to stick to a specific side of form and/or to grow or shrink when a window is resized.

If opposite anchors are activated then the component with grow or shrink with the window. For example if Top and Bottom are activated, then the component will grow/shrink when the window is vertically resized. If Left and Right are activated then the component will grow/shrink when the window is horizontally resized.

If opposite anchors are not activated, then the component will keep a constant distance from the sides of the window which correspond to the activated anchors.

Returns

[Number](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```

var form = solutionModel.newForm('mediaForm', 'db:/example_data/parent_table', null, false, 400, 300);
var stretchAllDirectionsLabel = form.newLabel('Strech all directions', 10, 10, 380, 280);
stretchAllDirectionsLabel.background = 'red';
stretchAllDirectionsLabel.anchors = SM_ANCHOR.ALL;
var stretchVerticallyLabel = form.newLabel('Strech vertically', 10, 10, 190, 280);
stretchVerticallyLabel.background = 'green';
stretchVerticallyLabel.anchors = SM_ANCHOR.WEST | SM_ANCHOR.NORTH | SM_ANCHOR.SOUTH;
var stretchHorizontallyLabel = form.newLabel('Strech horizontally', 10, 10, 380, 140);
stretchHorizontallyLabel.background = 'blue';
stretchHorizontallyLabel.anchors = SM_ANCHOR.NORTH | SM_ANCHOR.WEST | SM_ANCHOR.EAST;
var stickToTopLeftCornerLabel = form.newLabel('Stick to top-left corner', 10, 10, 200, 100);
stickToTopLeftCornerLabel.background = 'orange';
stickToTopLeftCornerLabel.anchors = SM_ANCHOR.NORTH | SM_ANCHOR.WEST; // This is equivalent to SM_ANCHOR.DEFAULT
var stickToBottomRightCornerLabel = form.newLabel('Stick to bottom-right corner', 190, 190, 200, 100);
stickToBottomRightCornerLabel.background = 'pink';
stickToBottomRightCornerLabel.anchors = SM_ANCHOR.SOUTH | SM_ANCHOR.EAST;

```

cssPosition

CSS position is a replacement for anchoring system making it more intuitive to place a component. CSS position should be set on form, an absolute position form can either work with anchoring or with css position. This is only working in NGClient.

Returns

[CSSPosition](#)

Supported Clients

NGClient

Sample

```

var label = form.newLabel('Label', -1);
label.cssPosition.r("10").b("10").w("20%").h("30px")

```

enabled

The enable state of the component, default true.

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```

var form = solutionModel.newForm('printForm', 'db:/example_data/parent_table', null, false, 400, 300);
var field = form.newField('parent_table_text', JSField.TEXT_FIELD, 10, 10, 100, 20);
field.enabled = false;

```

formIndex

The Z index of this component. If two components overlap, then the component with higher Z index is displayed above the component with lower Z index.

Returns

[Number](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var labelBelow = form.newLabel('Green', 10, 10, 100, 50);
labelBelow.background = 'green';
labelBelow.formIndex = 10;
var fieldAbove = form.newField('parent_table_text', JSField.TEXT_FIELD, 10, 10, 100, 30);
fieldAbove.background = '#FF0000';
fieldAbove.formIndex = 20;
```

height

The height in pixels of the component.

Returns

[Number](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var field = form.newField('parent_table_text', JSField.TEXT_FIELD, 10, 10, 100, 20);
application.output('original width: ' + field.width);
application.output('original height: ' + field.height);
field.width = 200;
field.height = 100;
application.output('modified width: ' + field.width);
application.output('modified height: ' + field.height);
```

name

The name of the component. Through this name it can also accessed in methods.
Must be a valid javascript name. (no - in the name or start with number)

Returns

[String](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = solutionModel.newForm('someForm', 'db:/example_data/parent_table', null, false, 620, 300);
var label = form.newLabel('Label', 10, 10, 150, 150);
label.name = 'myLabel'; // Give a name to the component.
forms['someForm'].controller.show()
// Now use the name to access the component.
forms['someForm'].elements['myLabel'].text = 'Updated text';
```

typeName

The webcomponent type (name from the spec).

Returns

[String](#)

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent');
application.output(bean.typeName);
```

visible

The visible property of the component, default true.

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var form = solutionModel.newForm('printForm', 'db:/example_data/parent_table', null, false, 400, 300);
var field = form.newField('parent_table_text', JSField.TEXT_FIELD, 10, 10, 100, 20);
field.visible = false;
```

width

The width in pixels of the component.

Returns[Number](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var field = form.newField('parent_table_text', JSField.TEXT_FIELD, 10, 10, 100, 20);
application.output('original width: ' + field.width);
application.output('original height: ' + field.height);
field.width = 200;
field.height = 100;
application.output('modified width: ' + field.width);
application.output('modified height: ' + field.height);
```

x

The x coordinate of the component on the form.

Returns[Number](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var field = form.newField('parent_table_text', JSField.TEXT_FIELD, 10, 10, 100, 20);
application.output('original location: ' + field.x + ', ' + field.y);
field.x = 90;
field.y = 90;
application.output('changed location: ' + field.x + ', ' + field.y);
```

y

The y coordinate of the component on the form.

Returns[Number](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var field = form.newField('parent_table_text', JSField.TEXT_FIELD, 10, 10, 100, 20);
application.output('original location: ' + field.x + ', ' + field.y);
field.x = 90;
field.y = 90;
application.output('changed location: ' + field.x + ', ' + field.y);
```

Methods Details**getAttribute(name)**

Get the value of an attribute of the element.

Parameters

[String](#) name the name of the attribute

Returns

[Object](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var frm = solutionModel.getForm('orders')
var fld = frm.getField('fld')
var attributes = fld.getAttributes();
for (var i = 0; i < attributes.length; i++)
{
    application.output(fld.getAttribute(attributes[i]));
}
```

getAttributes()

Returns the attribute names of an element.

Returns

[Array](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var frm = solutionModel.getForm('orders')
var fld = frm.getField('fld')
var attributes = fld.getAttributes();
for (var i = 0; i < attributes.length; i++)
{
    application.output(fld.getAttribute(attributes[i]));
}
```

getComment()

Returns the comment of this component.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var comment = solutionModel.getForm("my_form").getButton("my_button").getComment();
application.output(comment);
```

getDesignTimeProperty(key)

Get a design-time property of an element.

Parameters

[String](#) key the name of the property

Returns

[Object](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var frm = solutionModel.getForm('orders')
var fld = frm.getField('fld')
var prop = fld.getDesignTimeProperty('myprop')
```

getDesignTimePropertyNames()

Get the design-time properties of an element.

Returns

[Array](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var frm = solutionModel.getForm('orders')
var fld = frm.getField('fld')
var propNames = fld.getDesignTimePropertyNames()
```

getFormName()

Returns the name of the form. (may be empty string as well)

Returns

[String](#) The name of the form.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var name = %%elementName%%.getFormName();
```

getHandler(handlerName)

Returns the JSMethod handler with the given name.

Parameters

[String](#) handlerName the name of the handler to get

Returns

[JSMethod](#)

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent');
var handler = wc.getHandler('onActionMethodID');
```

getJSONProperty(propertyName)

Get the design-time value of the given property. If the value was never set and the component has default value for the give property, that value will be returned. To check if the value was set, use `isJSONPropertySet`.

Parameters

[String](#) propertyName the name of the property to get

Returns

[Object](#)

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent');
application.output(wc.getJSONProperty('mytext')); // will output a string value if present for a string typed
property
application.output(wc.getJSONProperty('mynumber')); // getter will return a number if present
application.output(JSON.stringify(wc.getJSONProperty('mycustomtype'), null, ' ')); // getter returns an object
if present for custom types is spec files
application.output(JSON.stringify(wc.getJSONProperty('myarray'), null, ' ')); // getter returns an array type
if present for array types
application.output(JSON.stringify(wc.getJSONProperty('myfoundset'), null, ' ')); // getter returns an object
representing the design settings of the given property if present

//In case you want to change a more complex property (e.g. tabs) obtained with getJSONProperty, setJSONProperty
must be used:
var tabs = jsTabPanel.getJSONProperty("tabs");
tabs[0].containsFormId = jsForm;
jsTabPanel.setJSONProperty("tabs", tabs);
```

getJSONPropertyNames(includeAll)

Get the list of property names this component supports.

It can return all properties specified in spec file or only the properties that are set in json.

Parameters

Boolean includeAll whether to return all properties from specification file or only the ones that are set in json

Returns

Array

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent');
application.output(wc.getJSONPropertyNames(true)); // will output all property names from the spec file
application.output(wc.getJSONPropertyNames(false)); // will output all property names that are set in component
json
```

getUUID()

Returns the UUID of this component.

Returns

UUID

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var button_uuid = solutionModel.getForm("my_form").getButton("my_button").getUUID();
application.output(button_uuid.toString());
```

isJSONPropertySet(propertyName)

Returns whatever the design-time value of the given property was set.

Parameters

String propertyName the name of the property to check

Returns

Boolean

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent');
var isSet = wc.isJSONPropertySet('mytext'); // returns false
wc.setJSONProperty('mytext', 'Hello World Extended!');
isSet = wc.isJSONPropertySet('mytext'); // returns true
```

putDesignTimeProperty(key, value)

Set a design-time property of an element.

Parameters

[String](#) key the name of the property
[Object](#) value the value to store

Returns

[Object](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var frm = solutionModel.getForm('orders')
var fld = frm.getField('fld')
fld.putDesignTimeProperty('myprop', 'strawberry')
```

removeAttribute(name)

Remove the attribute of an element.

Parameters

[String](#) name the name of the attribute

Returns

[String](#) the deleted attribute value

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var frm = solutionModel.getForm('orders')
var fld = frm.getField('fld')
fld.removeAttribute('keylistener')
```

removeDesignTimeProperty(key)

Clear a design-time property of an element.

Parameters

[String](#) key the name of the property

Returns

[Object](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var frm = solutionModel.getForm('orders')
var fld = frm.getField('fld')
fld.removeDesignTimeProperty('myprop')
```

resetHandler(handlerName)

Similar to resetJSONProperty but for handlers.

Parameters

[String](#) handlerName the name of the handler to reset

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent'); // form is extending another form who's 'onActionMethodID' does something nice
wc.setHandler('onActionMethodID', form.getMethod('onCoolerAction'));
// hmm, I changed my mind - I like 'nice' action better
wc.resetHandler('onActionMethodID');
```

resetJSONProperty(propertyName)

Reset the design-time value of the given property.

This makes it as if it was never set. It can be useful when working with form inheritance.

Parameters

[String](#) propertyName the name of the property to reset

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent'); // form is extending another form who's 'mycomponent' has property 'mytext' set to something...
wc.setJSONProperty('mytext', 'Hello World Extended!');
// hmm I changed my mind - I like simple 'Hello World' better
wc.resetJSONProperty('mytext');
```

setAttribute(name, value)

Set the attribute value of an element.

Parameters

[String](#) name the name of the attribute

[String](#) value the value of the attribute

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var frm = solutionModel.getForm('orders')
var fld = frm.getField('fld')
fld.setAttribute('keylistener', 'callback')
```

setHandler(handlerName, method)

Set the JSMETHOD handler for the given handler name. The handlerName is checked for existence in the component spec file, if the component does not declare this handler, an error is thrown.

If the handler is already set, it will be replaced with the new JSMETHOD.

Parameters

[String](#) handlerName the name of the handler to get

[JSMETHOD](#) method the JSMETHOD to attach to the handler (can be also JSMETHOD that has arguments)

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent');
wc.setHandler('onActionMethodID', form.getMethod('onAction'));
```

setJSONProperty(propertyName, value)

Set the design-time value for the given property. For primitive property types you can just set the value.

For more complex property types you can set a JSON value similar to what would be generated in the .frm file if you would design what you need using editor/properties view.

Some property types can be assigned values in the runtime accepted format (for example border, font typed properties have a string representation at runtime and here as well).

Parameters

[String](#) propertyName the name of the property to set
[Object](#) value the new value of the property

Supported Clients

NGClient

Sample

```
var wc = form.getWebComponent('mycomponent');
wc.setJSONProperty('mytext', 'Hello World!');
wc.setJSONProperty('mynumber', 1);
wc.setJSONProperty('myborder', 'LineBorder,1,#ccffcc');
wc.setJSONProperty('mydynamicfoundset', { dataproviders: { dp1: "city", dp2: "country" }, foundsetSelector: ""
}); // foundset property type using
    // the parent form's foundset and providing two columns of the foundset to client; see foundset property
type wiki page for more information
```