

NGCONSTANTS

 Apr 03, 2024 08:21

Supported Clients

NGClient

Constants Summary

String	FORM_BASED_BROWSER_HISTORY	By default the NGClient appends the name of the current main form to the url in the address bar of the browser using a fragment identifier (#).
String	VALUELIST_CONTAINS_SEARCH	the client property that can be set to always do a like search when filtering over valuelist on a typeahead like component.
String	WINDOW_BRANDING_ICON_192	Same as the WINDOW_BRANDING_ICON_32 client property just for images of size 192x192, usually used as shortcut icon for the web app.
String	WINDOW_BRANDING_ICON_32	Client property used to set the icon of the main window.
String	WINDOW_BRANDING_TITLE	When use branding is enabled (see servoy).
String	WINDOW_TIMEOUT	When the user navigates to another page, closes the browser or is disconnected from the server, the client-session on the server will be kept for a limited time.

Constants Details

FORM_BASED_BROWSER_HISTORY

By default the NGClient appends the name of the current main form to the url in the address bar of the browser using a fragment identifier (#....)

<p>

By setting the FORM_BASED_BROWSER_HISTORY property to false, this is disabled

The value can be true/false

DEFAULT: true

Returns

String

Supported Clients

NGClient

Sample

```
application.putClientProperty(APP_NG_PROPERTY.FORM_BASED_BROWSER_HISTORY, false);
```

VALUELIST_CONTAINS_SEARCH

the client property that can be set to always do a like search when filtering over valuelist on a typeahead like component.

So a component that has a valuelist as a property and uses user input to search in that valuelist.

By default it uses a like search with a % at the end (startsWith search on the fields of the valuelist).

But this makes it a like %value% so a contains search.

This can be set on the element (element.putClientProperty()) or on a application wide level (application.putClientProperty())

DEFAULT: false

Returns

String

Supported Clients

NGClient

Sample

```
elements.typeahead.putClientProperty(APP_NG_PROPERTY.VALUELIST_CONTAINS_SEARCH, true);
```

WINDOW_BRANDING_ICON_192

Same as the WINDOW_BRANDING_ICON_32 client property just for images of size 192x192, usually used as shortcut icon for the web app.

Returns

String

Supported Clients

NGClient

Sample**WINDOW_BRANDING_ICON_32**

Client property used to set the icon of the main window. This should be a PNG image of size 32x32, and it can be the file name that is stored under the web app root or it can be a base64 encoded image from the solution.

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

NGClient

Sample

```
// set image from the web app root onSolutionOpen
application.putClientProperty(APP_NG_PROPERTY.WINDOW_BRANDING_ICON_32, "favicon32x32.png");
// set base64 encoded image from solution onSolutionOpen
var img = solutionModel.getMedia("favicon32x32.png")
var imgAsBase64 = new Packages.org.apache.commons.codec.binary.Base64().encodeAsString(img.bytes);
var imgHref = "data:image/png;base64," + imgAsBase64;
application.putClientProperty(APP_NG_PROPERTY.WINDOW_BRANDING_ICON_32, imgHref);
```

WINDOW_BRANDING_TITLE

When use branding is enabled (see `servoy.branding` setting in Servoy Admin Page / `servoy.properties`) this client property can be used to set the main window title text.

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

NGClient

Sample

```
// set main window title onSolutionOpen
application.putClientProperty(APP_NG_PROPERTY.WINDOW_BRANDING_TITLE, "My app title");
```

WINDOW_TIMEOUT

When the user navigates to another page, closes the browser or is disconnected from the server, the client-session on the server will be kept for a limited time. If the user returns within that time the client session is continued.

<p>

This time can be configured at the server (60 seconds by default) and can overridden for the current ng-client session using

`application.putClientProperty` with `APP_NG_PROPERTY.WINDOW_TIMEOUT`.

<p>

The value is specified in seconds.

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

NGClient

Sample

```
// allow the user to return within 1 hour before the session is cleaned up
application.putClientProperty(APP_NG_PROPERTY.WINDOW_TIMEOUT, 3600);

// get the current active timeout value, when not overridden via putClientProperty this will return the system value.
var timeout = application.getClientProperty(APP_NG_PROPERTY.WINDOW_TIMEOUT);

// reset the value to the system value.
application.putClientProperty(APP_NG_PROPERTY.WINDOW_TIMEOUT, null);
```

