CSS Sequence (Execution Order)

» ServiceNow Specific StyleSheet
› Generated Bootstrap CSS
✱ Including Portal and Theme records' CSS at the top of this blob
» Patch
» Each CSS Include's Style Sheet record's CSS
✱ Including Portal and Theme records' CSS at the top of each blob
» Font-Awesome
» Page record's CSS
✱ Including Portal and Theme records' CSS at the top of this blob
» Alternating Sequence of:
» Each Widget record's CSS
✱ Including Portal and Theme record CSS at the top of each blob
» Each Widget's Instance records' Stylesheets
✱ Including Portal and Theme record CSS at the top of each blob

Angular Directives

Evaluate the given expression when the user changes the input.	`ng-cha nge -="ex pre ssi -on"`
Sets the checked attribute on the element	`ng-che cke -d="b ool ean "`
This directive sets the disabled attribute on the element	`ng-dis abl -ed= " boo lea -n"`
The ngHide directive shows or hides the given HTML element based on the expression provided to the ngHide attribute.	`ng-hid e|s -how ="bo ole -an"`
The ngMousedown directive allows you to specify custom behavior on mousedown event.	`ng-mou sed -own ="ex pre -ssi on"`
Specify custom behavior on mouseover event	`ng-mou seo -ver ="ex pre -ssi on"`
Specify custom behavior on blur event. A blur event fires when an element has lost focus.	`ng-blu r="e -xpr ess ion "`
Run a loop with the data in array	`ng-rep eat =" ([ key,] value) in object |ar -ray "`
Evaluate an if condition	`ng-if= " exp -res sio n"`

Client script global functions

`this.s -erv er.g -et ([O -bject])`	Calls the server and sends custom input. Returns Promise.
`this.s -erv er.u -pd ate()`	Calls the server and posts this.data to the server script. Returns Promise.
`this.s -erv er.r -ef resh()`	Calls the server and automatically replaces the current options and data from the server response. Returns Promise.

A promise represents the eventual result of an asynchronous operation. For more information on promises, see https://promise-saplus.com/ or AngularJS documentation.

spAriaUtil - Client

Announce a message to a screen reader.	`spAria Uti l.s end Liv eMe ssa -ge('Say Hello to screen reader');`

The spAriaUtil service is an angular service included as part of the Service Portal angular application. The spAriaUtil service is available in the client script block of Service Portal widgets.

To use the spAriaUtil please add the same to client controller

```
functi on( spA ria Util) {
/ widget controller /
var c =this;
}
```

spUtil - Client

Displays a notification error message.	`spUtil.ad dEr ror Mes sag -e("There has been an error processing your reques t")`
Displays a notification info message.	`spUtil.ad dIn foM ess age ("Your order has been placed ")`
Displays a trivial notification message. *Trivial messages disappear after a short period of time.*	`spUtil.ad dTr ivi alM ess age -("Thanks for your order")`

spUtil - Client (cont)

Use this method to embed a widget model in a widget client script. The callback function returns the full widget model.	`spUtil.ge t(' pps -li - st- modal', {title: c.data.ed itA llo cat - ions, table: 'resou rce _al - loc ation', queryS tring: 'GROUP BYu - ser ^re sou rce _plan=' + c.data.sysId, view: 'resou rce _po rta - l_a llo cat ions' }).the - n(f unc tio n(r esp onse) { var formModal = response; c.allo cat ion Lis tModal = response; });`
Formats a string as an alternative to string concatenation. *Use this method to build a string with variables.*	`spUtil.fo rma t('An error ocurred: {error} when loading {widget}', {error: '404', widget: 'sp-wi - dget'})`
Calls the server and replaces the current options and data with the server response.	`spUtil.re fre sh( $scope)`
Updates the data object on the server within a given scope.	`spUtil.up dat e($ scope)`
Watches for updates to a table or filter and returns the value from the callback function.	`spUtil.re cor dWa tch - ($s cope, " pro ble m", " act ive =tr ue", functi on( res ponse) { // Returns the data inserted or updated on the table consol e.l og( res pon - se.d ata); });`

Utility methods to perform common functions in a Service Portal widget client script.

To use the spUtil functions in the client controller, ensure that you add to client function

```
functi on( spU til ,$s cope) {
/ widget controller /
var c =this;
}
```

Internationalize a widget

✱Use the ${} or gs.getMessage() syntax in widgets to tag strings for translation so you can localize your Service Portal content.

```
<di v>
<p> ${This message will be transl ate d.} </p>
<p> How ever, this will nessage will NOT be transl ate d.< /p>
</d iv>
```

Now to translate the above, add the text in sys_ui_message to get the respective translation

ex:

- Key: This message will be translated.
- Language : 'de' / 'fr' / 'it'
- Message : Add the translated text

GlideSPScriptable - Scoped

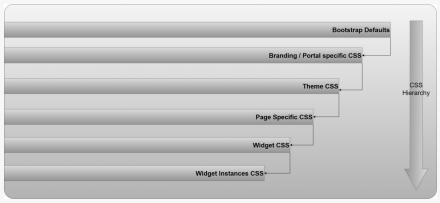
if the user can read the specified GlideR ecord.	$sp.ca nRe adR - eco rd(<gl ide - rec ord >)
if the user can read the specified record with table and sysID	$sp.ca nRe adR - eco rd('ta ble', 'sys_id')
if the currently logged in user has permission to view the specified page	GlideS PSc rip - tab le.c an See - Pag e("p age - _id ")
Returns a model and view model for a sc_cat _item or sc_cat _it em_ guide.	$sp.ge tCa tal - ogI tem (sy s_id)
Returns the display value of the specified field (if it exists and has a value) from either the widget's sp_ins - tance or the sp_portal record.	$sp.ge tDi spl - ayV alu e("s c_c - ata log ")
Returns the portal record from the Service Portals [sp_po rtal] table.	$sp.ge tPo rta - lRe cord()
Returns an array of Service Catalog variables associated with a record.	$sp.ge tRe cor - dVa ria ble sAr - ray ('s c_r eq_ - ite mGlide record ',t rue(if you want variables with no values));
Gets a widget by id or sys_id, executes that widget's server script using the provided options, then returns the widget model.	$sp.ge tWi dge - t(' wid get _id', {p1: param1, p2: param2});

Interact with data and perform record operations in Service Portal widgets.
You access GlideS PSc rip table methods by using the global $sp object.

ServicePortal Cheat Sheet Cheat Sheet

by Bibin Gokuldas (bibingokuldas) via cheatography.com/69673/cs/23975/

CSS Hierarchy



Global objects in widgets

» **Server Side Objects**

› ***input*** : *Sent from Client script*

› ***options*** : *Configurations mentioned in widget options when widget is loaded*

› ***data*** : *Object or properties/value pair sent to client*

» **Client Side Objects**

› ***options*** : *Configurations mentioned in widget options when widget is loaded*

› ***data*** : *Serialized data object from Server script*

The "snRecordPicker"

✱ Creating a reference Field

```
<sn -re cor d-p icker field= " "
                                    tab le= " 'sy -
s_u ser '"
                                    dis pla y-f -
iel d="' nam e'"
                                    dis pla y-f -
iel ds= " 'us er_ nam e'"
                                    val ue- fie -
ld= " 'sy s_i d'"
                                    sea rch -fi -
eld s="' nam e'"
                                    def aul t-q -
uer y="' nam eLI KEA dmi n'"
                                    pag e-s ize -
="10 0" ></ sn- rec ord -pi cke r>
```

When you try and include a reference field (such as "department") in the "display-fields" parameter, the field that you specify doesn't render other non reference fields do render correctly. This is an existing issue can be found in the KB0717097
you can pass in the filter you need to apply on reference field using default-query attribute

The "sp-date-picker"

FormController -(Widgets)

True if user has not interacted with the form yet.	`$pristine`
True if user has already interacted with the form.	`$dirty`

spModal - Client

Displays an alert.	`spModa l.a ler t(' Thanks for the update ').t he n(f unction (answer) { c.simple = answer; });`
Displays a confirmation message.	`spModa l.c onf irm ("Can you confirm or deny this?").th en( -fun cti on( con firmed) { c.conf irmed = confirmed; // true or false })`
Opens a modal window using the specified options.	`spModa l.o pen({ title: 'Give me a name', message: 'What would you like to name it?', input: true, value: c.name }).the n(f unc tio n(name) { c.name = name; })`
Displays a prompt for user input.	`spModa l.p rom pt( "Your name please ", c.name ).t hen (fu -nct ion (name) { c.name = name; })`

The spModal class provides an alternative way to show alerts, prompts, and confirmation dialogs. The SPModal class is available in Service Portal client scripts.
You can use spModal.open() to display a widget in a modal dialog.
To use spModal in the client controller ensure that you add the spModal function in the client function

```
functi on( spM odal) {
/ widget controller /
var c =this;
}
```

Embed a widget in an HTML template

✱ The sp-date-picker allows to select a date and time in widgets

```
<sp -da te- picker field= " fro mda te"
                                ng- mod el= " c.f -
rom "
                                ng- mod el- opt -
ion s="{ get ter Setter: true}"
                                sn- inc lud e-t -
ime ="tr ue"
                                sn- dis abl ed= " -
fal se"> </s p-d ate -pi cke r>
```

sn-include-time property when set to true it will allow to select date and time. and when false it will allow to select only date.

✱ Use the <widget></widget> element to embed a widget in an HTML template. Pass in the ID of the widget you are trying to embed as a parameter.

```
<di v>
<widget id=" wid get -co ol- clo ck"> </w idg et>
</d iv>
```

✱ If a widget has an option schema, you can define instance options in JSON format.

```
<widget id=" wid get -co ol- clo ck" option s=' -
{"zo ne": " Ame ric a/L os_ Ang ele s","t itl e":
"San Diego, CA"} '></w idg et>
```

By **Bibin Gokuldas**
(bibingokuldas)

cheatography.com/bibingokuldas/
bibingokuldas.com/

Published 26th September, 2020.
Last updated 26th September, 2020.

spContextManager - Client

Initializes a key and adds widget data as the value.	`spCont ext Man age r.a ddC ont ext - ('a gen t-c hat', { 'appro val _co - unt': 5 });`
Returns each key and associated data object defined by any widget on the page.	`spCont ext Man age r.g etC ont ext();`
Returns the widget data associated with a key.	`spCont ext Man age r.g etC ont ext - For Key ('a gen t-c hat', true).t he - n(f unc tio n(c ontext) { context = context || {}; contex t.a ppr ova - l_count = 5; spCont ext Man age r.u - pda teC ont ext For Key ('a gen t-c - hat', context); });`
Sends data to an existing key.	`spCont ext Man age r.g etC ont ext - For Key ('a gen t-c hat', true).t he - n(f unc tio n(c ontext) { context = context || {}; contex t.a ppr ova - l_count = 5; spCont ext Man age r.u - pda teC ont ext For Key ('a gen t-c - hat', context); });`

Make data from a Service Portal widget available to other applications and services in a Service Portal page.

to use the spContextManager, ensure to add this in the function for client controller

```
function ($scope, spCont ext Man ager) {
var c =this;
}
```

By **Bibin Gokuldas**
(bibingokuldas)

cheatography.com/bibingokuldas/
bibingokuldas.com/

Published 26th September, 2020.
Last updated 26th September, 2020.