

### 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.	<code>ng-change = "expression"</code>
Sets the checked attribute on the element	<code>ng-checked="boolean"</code>
This directive sets the disabled attribute on the element	<code>ng-disabled="boolean"</code>
The ngHide directive shows or hides the given HTML element based on the expression provided to the ngHide attribute.	<code>ng-hide/show="boolean"</code>
The ngMousedown directive allows you to specify custom behavior on mousedown event.	<code>ng-mousedown="expression"</code>
Specify custom behavior on mouseover event	<code>ng-mouseover="expression"</code>
Specify custom behavior on blur event. A blur event fires when an element has lost focus.	<code>ng-blur="expression"</code>
Run a loop with the data in array	<code>ng-repeat = "[ key, ] value in object   array "</code>
Evaluate an if condition	<code>ng-if= " expression "</code>

### Client script global functions

<code>this.\$server.get([Object])</code>	Calls the server and sends custom input. Returns Promise.
<code>this.\$server.update()</code>	Calls the server and posts this.data to the server script. Returns Promise.
<code>this.\$server.refresh()</code>	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-sapplus.com/> or AngularJS documentation.*

### spAriaUtil - Client

Announce a message to a screen reader.

```
spAriaUtil.speak('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**

```
function(spAriaUtil) {
  // widget controller /
  var c=this;
}
```

### spUtil - Client

Displays a notification error message.

```
spUtil.adError('There has been an error processing your request');
```

Displays a notification info message.

```
spUtil.adInfo('Your order has been placed');
```

Displays a trivial notification message.

```
spUtil.adTrivial('Thanks for your order');
```

*Trivial messages disappear after a short period of time.*



By **Bibin Gokuldas**  
(bibingokuldas)

Published 26th September, 2020.  
Last updated 26th September, 2020.  
Page 1 of 6.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>

### 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.get('pps -list-modal', {title:
c.data.editions,
table: 'resource_allocation',
queryString: 'GROUP BY user ^resource_plan=' +
c.data.sysId,
view: 'resource_portals_
l_ations'}).then(function(response)
{
var formModal = response;
c.allocationListModal
= response;
});
```

Formats a string as an alternative to string concatenation. *Use this method to build a string with variables.*

```
spUtil.format('An error
occurred: {error} when
loading {widget}', {error:
'404', widget: 'sp-widget'})
```

Calls the server and replaces the current options and data with the server response.

```
spUtil.refresh($scope)
```

Updates the data object on the server within a given scope.

```
spUtil.update($scope)
```

Watches for updates to a table or filter and returns the value from the callback function.

```
spUtil.watch($scope, "problem",
"active=true",
function(response) {
// Returns the data
inserted or updated on the
table
console.log(response.data);
});
```

*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*

```
function(spUtil,$scope) {
/ 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.

```
<div>
<p> ${This message will be translated.} </p>
<p> However, this message will NOT be
translated.</p>
</div>
```

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 GlideRecord.	<code>\$sp.canRecord(&lt;glideRecord&gt;)</code>
if the user can read the specified record with table and sysID	<code>\$sp.canRecord('table', 'sys_id')</code>
if the currently logged in user has permission to view the specified page	<code>GlideScriptTable.canSeePage("page_id")</code>
Returns a model and view for a sc_cat_item or sc_cat_item_guide.	<code>\$sp.getItem(sys_id)</code>
Returns the display value of the specified field (if it exists and has a value) from either the widget's sp_instance or the sp_portal record.	<code>\$sp.getItemDisplayValue("sc_catalog")</code>
Returns the portal record from the Service Portals [sp_portal] table.	<code>\$sp.getPortalsRecord()</code>
Returns an array of Service Catalog variables associated with a record.	<code>\$sp.getItemRecordVariablesArray('sc_record', true (if you want variables with no values));</code>
Gets a widget by id or sys_id, executes that widget's server script using the provided options, then returns the widget model.	<code>\$sp.getItemWidget('widget_id', {p1: param1, p2: param2});</code>

*Interact with data and perform record operations in Service Portal widgets.  
You access GlideScriptTable methods by using the global \$sp object.*



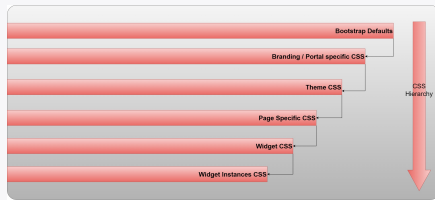
By **Bibin Gokuldas**  
(bibingokuldas)

[cheatography.com/bibingokuldas/](http://cheatography.com/bibingokuldas/)  
[bibingokuldas.com/](http://bibingokuldas.com/)

Published 26th September, 2020.  
Last updated 26th September, 2020.  
Page 2 of 6.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>

### 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-record-picker field= " "
                                table= " 'sys -
s_user "'
                                display-f -
iel d="' name'"
                                display-f -
iel ds= " 'user_ name'"
                                value- fie -
ld= " 'sys_id'"
                                sea rch -fi -
eld s="' name'"
                                def aul t-q -
uer y="' nameLI 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.

```
spModal.alert(' Thanks for the
update ').then(function
(answer) {
c.simple = answer;
});
```

Displays a confirmation message.

```
spModal.confirm("Can you
confirm or deny this?").then( -
function( confirmed) {
c.confirmed = confirmed; // true
or false
})
```

Opens a modal window using the specified options.

```
spModal.open({
title: 'Give me a name',
message: 'What would you like to
name it?',
input: true,
value: c.name
}).then(function(name) {
c.name = name;
})
```

Displays a prompt for user input.

```
spModal.prompt("Your name
please ", c.name ).then (fu -
nction (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**

```
function( spModal) {
/ 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-date-picker field= "from date"
                 ng-model= "current"
                 from="
                 options="{getter Setter: true}"
                 sn-include-time="true"
                 sn-disabled="false"></sp-date-picker>
```

*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.

```
<div>
<widget id="widget-color-clock"></widget>
</div>
```

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

```
<widget id="widget-color-clock" options=' -
{"zone": "America/Los_Angeles", "title":
"San Diego, CA"} '></widget>
```



By **Bibin Gokuldas**  
(bibingokuldas)

Published 26th September, 2020.  
Last updated 26th September, 2020.  
Page 3 of 6.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>

[cheatography.com/bibingokuldas/](http://cheatography.com/bibingokuldas/)  
[bibingokuldas.com/](http://bibingokuldas.com/)

### spContextManager - Client

Initializes a key and adds widget data as the value.

```
spContextManager.addContext('a-gen-t-hat', { 'approval-count': 5 });
```

Returns each key and associated data object defined by any widget on the page.

```
spContextManager.getContext();
```

Returns the widget data associated with a key.

```
spContextManager.getContext(
  For Key ('a-gen-t-hat', true).then(
    function(context) { context =
    context || {}; context.approval-
    count = 5; spContextManager.
    updateContext(For Key ('a-gen-t-
    hat', context)); });
```

Sends data to an existing key.

```
spContextManager.getContext(
  For Key ('a-gen-t-hat', true).then(
    function(context) { context =
    context || {}; context.approval-
    count = 5; spContextManager.
    updateContext(For Key ('a-gen-t-
    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, spContextManager) {
  var c = this;
}
```



By **Bibin Gokuldas**  
(bibingokuldas)

Published 26th September, 2020.

Last updated 26th September, 2020.

Page 4 of 6.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>