

Field Based APIs

Hide/Display variable	<code>g_form.setDisplay(<fld>,true/false);</code>
Disable a field	<code>g_form.setDisable(<fld>,true/false);</code>
Field ReadOnly	<code>g_form.setReadOnly(<fld>,true/false);</code>
Clear Value from Field	<code>g_form.clearValue(<fld>);</code>
Change Label of Field	<code>g_form.setLabelOf(<fld>,newlabel);</code>
Make field mandatory	<code>g_form.setMandatory(<fld>,true/false);</code>
Set Value of Field	<code>g_form.setValue(<fld>,value);</code>
Show error and scroll to field	<code>g_form.showErrorBox(<fld>,msg,true/false);</code>
Add Options to Choice field	<code>g_form.addOption(<fld>,value,label,index);</code>
Remove Options from Choice field	<code>g_form.removeOption(<fld>,value);</code>
Clear all Options from Choice Field	<code>g_form.clearOptions(<fld>);</code>
Get Reference field Value and Attributes	<code>g_form.getReference(<fld>,callbackfunction);</code>
Flash a particular field	<code>g_form.flash(table.name,color,no of flashes)</code>
Get Integer Value of Field	<code>g_form.getIntValue(<fieldname>);</code>
Get Decimal Value of Field	<code>g_form.getDecimalValue(<fieldname>);</code>

<fld> Represent the field name on which the function has to work
value - is something what you have to set.
true / false - true is for enabling, false is for disable

User Objects

User name of the current user e.g. employee	<code>g_user.userName</code>
First name of the current user e.g. Joe	<code>g_user.firstName</code>
Last name of the current user e.g. Employee	<code>g_user.lastName</code>
sys_id of the current user e.g. 681ccaf9c0a8016400-b98a06818d57c7	<code>g_user.userID</code>
if the current user has the role specified, false otherwise. ALWAYS returns true if the user has the 'admin' role.	<code>g_user.hasRole(<rolename>)</code>
if the current user has the exact role specified, false otherwise, regardless of 'admin' role.	<code>g_user.hasRoleExactly(<rolename>)</code>
if the current user has at least one role specified, false otherwise.	<code>g_user.hasRoles(<rolename1>,<rolename2>)</code>

Related List APIs

Show Related List	<code>g_form.showRelatedList(<relatedlisttablename>)*</code>
Hide Related List	<code>g_form.hideRelatedList(<relatedlisttablename>)</code>
Show All Related Lists	<code>g_form.showRelatedLists();</code>
Hide All Related Lists	<code>g_form.hideRelatedLists();</code>

Sections based API

Show/Hide a Section	<code>g_form.setSectionDisplay(<sectionname>,true/false);</code>
Get All Sections*	<code>g_form.getSectionNames();</code>
Get All Sections**	<code>g_form.getSections();</code>

Sections based API (cont)

Check whether section is Visible	<code>g_form.isSectionVisible(<sectionname>);</code>
*Returns all the sections including the invisible ones, **Not available on Mobile Platform <sectionname> : the section name in ServiceNow is created by replacing the 1st space by "_" and trimming rest of the spaces, for ex :if a section on form is created as "Section Specific API" then section name is "section_specificapi"	

Decorations API

Add decorations(not applicable for Catalog Client Scripts)	<code>g_form.addDecoration(<fld>,iconname,-titleforicon);</code>
Remove Decorations(not applicable for Catalog Client Scripts)	<code>g_form.removeDecoration(<fld>,iconname,titleforicon);</code>
LiveUpdate running on form*	<code>g_form.isLiveUpdating();</code>

*Live update can detect any change of the data on the form from another session.

HTML based APIs

Control a Field	<code>g_form.getControl(<fldname>)</code>
Get an HTML Element from Parameter*	<code>g_form.getElement(<parameter>)</code>
Get HTML element of Option**	<code>g_form.getOption(<fieldname>,value);</code>
Get HTML element of Form	<code>g_form.getFormElement();</code>
Get HTML element of HelpText	<code>g_form.getHelpTextControl(<fldname>);</code>

*Not available in the Mobile and Service Portal

**The method will not work on readonly fields, and if the field is not present on the form it will result in null.



By Bibin Gokuldas (bibingokuldas)

Published 25th October, 2018.
Last updated 11th August, 2020.
Page 1 of 2.

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

Action Specific API

Submit a form	<code>g_form.submit(actionName)</code>
Save a form	<code>g_form.save()</code>

Best Practices Client Scripting

Use UI policies instead of Client scripts : For basics like making field readonly, mandatory best practice suggests to use UI Policies.

Always use isLoading check when in onChange Client Scripts : All the onChange client script will get executed onLoad as well, always check whether the form is loading by isLoading function

Avoid Using GlideRecord function in Client script : Replace the calls by GlideAjax

Avoid DOM(manipulation of form elements)
. As all the elements might be changed in the future release, and to avoid complications avoid doing DOM

Use g_scratchpad to minimise server calls :
When working with lot of data write a Display Business Rule and make the data available to Client Side

When using JQuery in Client Scripts :
Instance before london, has no possibility of isolating a JQuery Script, but instances in London and above has a strict client mode, where methods like window, eval would only work if isolation is turned on.



By **Bibin Gokuldas**
(bibingokuldas)

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

Published 25th October, 2018.
Last updated 11th August, 2020.
Page 2 of 2.

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