Key API ServiceNow Scripts Cheat Sheet

by Bibin Gokuldas (bibingokuldas) via cheatography.com/69673/cs/23998/

ArrayUtil API

Merges two arrays	`new ArrayUtil().concat(array1,array2)`
Searches the array for the element. Returns true if the element exists in the array, otherwise returns false	`new ArrayUtil().contains(array,searchobject)`
Converts an object to an array.	`new ArrayUtil().convertArray(Object)`
Finds the differences between two or more arrays	`new ArrayUtil().diff(array1,array2,arrayn)`
Returns an array from the object.	`new ArrayUtil().ensureArray(Object)`
Searches the array for the element. Returns the element index if found, -1 otherwise.	`new ArrayUtil().indexOf(array1, seachelement)`
Finds the elements common in all arrays.	`new ArrayUtil().intersect(Array a, Array b)`
Merges two or more arrays (common element is addded once)	`new ArrayUtil().union(Array a, Array b)`
Removes duplicate items from an array.	`new ArrayUtil().unique(Array a)`

The syntax of the above APIs can be Like this

```
var arUtil = new ArrayUtil();
arUtil.unique(array1);
```

CIUtils

Determine which business services are affected by a specific CI.	`new CIUtils().servicesAffectedByCI(ci_sys_id);`
Determine which business services are affected by a task.	`new CIUtils().servicesAffectedByTask(taskgliderecord)`

Sample Usage of the script can be like below

```
var CIUtil = new CIUtils();
var bsaffected = CIUtil.servicesAffectedByCI(ci_sys_id);
```

GlideFilter

Compares a specified filter to the contents of a specified GlideRecord.	`GlideFilter.c` `"filtercondit`

If the specified filter contains one condition, the method
the condition of the filter.
matchparameter indicates whether all conditions must b
contains multiple conditions.
Valid values:
true: all conditions must be met for the method to return
false: only one of the conditions must be met for the met

TableUtils

Checks to see if a table exists.	`new` `ME").`
Drops a database table. (use with caution)	`new` `LENAM`
Drops a database table and cleans up references to the table.	`new` `lean`
Drops a database table, all of it's extended tables, and cleans up references to the tables.	`new` `eAndD` `AME")`
Returns the base table name from which the table was extended.	`new` `me").`
Returns the list of tables that extend a table, includes the base table in array	`new` `me").`
Returns a list of all classes participating in the hierarchy of the specified table.	`new` `me").`
Returns the table hierarchy.	`new` `me").`
Returnss a list of tables that extend a table.	`new` `me").`
Determines if a table has been extended.	`new` `me").`

By **Bibin Gokuldas** (bibingokuldas)
cheatography.com/bibingokuldas/
bibingokuldas.com/

Published 24th August, 2020.
Last updated 24th August, 2020.
Page 1 of 5.

TableUtils (cont)

Determines if a table is a base class, meaning it has no parents and has extensions.	`new TableUtils("tablename").isBaseClass();`
Determines if the table has no parents and no extensions.	`new TableUtils("tablename").isSoloClass();`

Use Drop the table function with extreme caution as it will delete the table and the data forever

GlideSysAttachment

Copies attachments from the source record to the target record.	`new GlideSysAttachment().copy(String sourceTable, String sourceID, String targetTable, String targetID)`
Deletes the specified attachment.	`new GlideSysAttachment().deleteAttachment(attachmentsysID)`
Returns a GlideRecord containing the matching attachment metadata such as name, type, or size.	`new GlideSysAttachment().getAttachments('<table_name>', '<record_sys_id>');`
Returns the attachment content as a string.	`new GlideSysAttachment().getContent(GlideRecord sysAttachment)`
Returns the attachment content as a string with base64 encoding.	`new GlideSysAttachment().getContentBase64( GlideRecord sysAttachment)`
Inserts an attachment for the specified record using base64 encoded content.	`new GlideSysAttachment().writeBase64( GlideRecord gr, String fileName, String contentType, String content_base64Encoded)`

SLARepair

Repair the task SLAs associated with the passed-in filter and source table.
Repair the task SLAs associated with the passed in GlideRecord.
Repair the task SLAs associated with the passed in sys_id and source table.
Enables or disables auditing when running a repair.*
Enables or disables running a workflow for each of the Task SLA records being repaired.
Validates the repair request.

*The SLARepair API first deletes th recreates them from each task's hi *auditing is set to the value in the p air.audit. You can override this wit false to disable auditing.*

GlideEmailOutbound

Adds the address to either the cc or bcc list.	`email.a 'emaila ss2');`
Returns the email's subject line.	`email.g`
Returns the email's watermark.	`email.g`
Sets the body of the email.	`email.s`
Sets the sender's address.	`email.set
Sets the reply to address.	`email.s ess');`
Sets the email's subject line.	`email.s ess');`

GlideSession

Sets a session client value that can be retrieved with getClientData().	`gs.getSession().putClientData('paramname', 'paramvalue');`
Determines if the current user is currently logged in.	'gs.getSession().isLoggedIn()'
Determines if the current session is interactive.	'gs.getSession().isInteractive()'
Gets the name of the session's time zone.	'gs.getSession().getTimeZoneName()'
Gets a list of roles for the current user.	'gs.getSession().getRoles()'
Gets the session's language code.	'gs.getSession().getLanguage()'
Returns a session client value previously set with putClientData().	`gs.getSession().getClientData('paramname');`
Clears a session client value previously set with putClientData().	`gs.getSession().clearClientData('paramname');`

The GlideSession API allows you to find information about the current session.

```
var session = gs.getSession();
session.putClientData('cheatsheet', 'OOBAPI');
```

CartJS

Adds the request for a catalog item to the current cart.	`new sn_sc.CartJS().addToCart(request);`
Performs the cart checkout. If the two-step checkout is enabled, returns the order summary. If the two-step checkout is disabled, the cart is submitted and details of the generated request are returned.	`new sn_sc.CartJS().checkoutCart();`
Deletes the current cart.	`new sn_sc.CartJS().empty();`
Returns the cart id of the current cart.	`new sn_sc.CartJS().getCartID();`

CartJS (cont)

Returns the GlideRecord for the cart item (sc_cart_item) in the current cart
Gets the sys_id from the sys_user record of the user for whom the cart is requested.
Orders a single item.
Specifies if the current user has the required role to edit the Request for field.
Sets the sys_id in the sys_user record of the user for whom the cart is requested.

The variable used in the scripts above JSON Object which has the details of cart or to order. Sample is below

```
var item ={
'sysparm_id': '0d08837237153
// Catalog item sys_id
'sysparm_quantity': '1',
'variables':{ // Pass the ca
'carrier': 'at_and_t_mobilit
'data_plan': '500MB',
'duration': 'eighteen_months
'color': 'slate',
'storage': 'sixtyfour'
}};
```

TemplatePrinter (Mail Script)

Prints the string to the email body.	`template.`
	`- " + cur`

Adds non-breaking spaces to the email body.	`template.`

GlideLocale

Returns the GlideLocale object.	`GlideLocale.get`

Returns the decimal separator.	`var locale = Gl`
	`dseperatr = loc`
	`rator()`

Key API ServiceNow Scripts Cheat Sheet

by Bibin Gokuldas (bibingokuldas) via cheatography.com/69673/cs/23998/

GlideLocale (cont)

Returns the grouping separator.	`var locale = GlideLocale.get();var groupingSeparator = locale.getGroupingSeparator();`

GlideRecordUtil

Returns a GlideRecord instance positioned to the given CI sys_id, and of the right class (table).	`new GlideRecordUtil().getCIGR("cisysid");`
Returns a list of all the fields in the specified GlideRecord.	`new GlideRecordUtil().getFields(gliderecord-object)`
Returns a GlideRecord instance for the given table, positioned to the given sys_id, and of the right class (table).	`new GlideRecordUtil().getGR("table", "sysID");`
Returns a Java ArrayList of the ancestors of the given table name.	`new GlideRecordUtil().getTables("tablename")`
Sets the fields in the specified GlideRecord with the field values contained in the specified hashmap, unless that field name is in the ignore hashmap.*	`var gr = new GlideRecordUtil().getGR("table", "sysid"); var obj = {"fieldname": "fieldvalue"}; var ignore = {"sys_created_by": true}; new GlideRecordUtil().mergeToGR(obj, gr, ignore); gr.update();`
Populates the given hashmap from the given GlideRecord instance. Each field in the GlideRecord becomes a property in the hashmap.*(returns field/value pairs from the GlideRecord)*	`var objectToPopulate = {}; var gr = new GlideRecordUtil().getGR("table", "sysid"); var ignore = {"sys_created_on": true, "sys_updated_by": true}; new GlideRecordUtil().populateFromGR(objectToPopulate, gr, ignore);`

GlideSchedule

Adds a new schedule segment to the current schedule.	n d
Determines the elapsed time in the schedule between two date time values using the timezone of the schedule or, if that is not specified, the timezone of the session.	v G u i s r
Retrieves the schedule name.	n i
Determines if the given datetime is within the current schedule.	n d l
Determines if the current schedule is valid. A schedule is valid if it has at least one schedule span.	n d
Sets the timezone for the current schedule.	v G l i
Determines how much time (in milliseconds) until start time of the next schedule item.	n e o

The scoped GlideSchedule API provid operations on GlideSchedule objects, s segments to a schedule, determining i schedule, or setting the schedule timez

FlowAPI

Cancels a paused or running flow, subflow, or action.	`sn_f ow('`
Run an action from a server-side script synchronously.	`sn_f ion(`

FlowAPI (cont)

Run an action from a server-side script synchronously from the current user session without creating execution details or other related records	`sn_fd.FlowAPI.executeAction-Quick('flowname', inputs);`
Runs a Data Stream action synchronously from a server-side script and returns a ScriptableDataStream object.	`sn_fd.FlowAPI.executeDataSt-reamAction('datastream_n-ame',input,timeout);`
Run a flow from a server-side script synchronously.	`sn_fd.FlowAPI.executeFlow('-flowname', inputs,timeout);`
Run a flow, subflow, or action from a server-side script synchr-onously or asynchronously without creating execution details or other related records.	`sn_fd.FlowAPI.executeFlowQu-ick('flowname', inputs);`
Run an subflow from a server-side script synchronously.	`var results = sn_fd.FlowAP-I.executeSubflow('flowname', inputs);`
Run a subflow from a server-side script synchronously from the current user session without creating execution details or other related records	`var results = sn_fd.FlowAP-I.executeSubflowQuick('flow-name', inputs);`
Build password2 values inside a script step.	`var encrypteddata=sn_fd.Glid-eActionUtil.setEncryptedOut-put(Val);`
Run an action from a server-side script asynchronously.	`sn_fd.FlowAPI.startAction('-flowname', inputs);`
Run an action from a server-side script asynchronously without creating execution details	`sn_fd.FlowAPI.startActionQu-ick('flowname', inputs);`

FlowAPI (cont)

Run a flow from a server-side script.
Run a flow from a server-side script asynchronously without creating execution details
Run a subflow from a server-side script.
Run a subflow from a server-side script asynchronously without creating execution details

Use FlowAPI methods to execute act server-side scripts using either block