**Scripting in ServiceNow**

**Scripts in ServiceNow fall into two categories:**

* **Client-side**
* **Server-side**

**This learning module is about client-side scripting. Client-side scripts execute within a user's browser and are used to manage forms and form fields. Examples of things client-side scripts can do include:**

* **Place the cursor in a form field on form load**
* **Generate alerts, confirmations, and messages**
* **Populate a form field in response to another field's value**
* **Highlight a form field**
* **Validate form data**
* **Modify choice list options**

**Hide/Show fields or sections**

1. **Client-side scripting: Used for Validation (Validating Form)**

* **UI Policies: Make field mandatory Read only, Visible or not visible on condition based.**
* **Client scripts**
* **On load**
* **On Change**
* **On Submit**
* **On Cell Edit**

**The API’s used in Client-side scripting are *g\_form* , *g\_user* and Glide Ajax each api has different methods.**

* **The GlideForm API provides methods to customize forms. GlideForm.js is the JavaScript class containing the methods. Only use GlideForm methods on the client. You can use these methods to make custom changes to the form view of records. All validation of examples was done using client scripts.**
* **The GlideUser API provides methods that access information about the current user and current user roles. Using this API avoids the need to use the slower GlideRecord queries to get user information. GlideUser methods and properties are accessed through a global object (g\_user) that is only available in client scripts.**
* **The GlideAjax class enables a client script to call server-side code in a script include.**
* **Create a GlideAjax instance by calling the GlideAjax constructor. As the argument to the constructor, specify the name of the script include class that contains the method you want to call.**
* **Call the addParam method with the sysparm\_name parameter and the name of the script-include method you want to call.**
* **Optional. Call the addParam method one or more times to provide the script-include code with other parameters it needs.**
* **Execute the server-side code by calling getXML().**
* **Note: getXML() is the preferred method for executing the code, because it is asynchronous and does not hold up the execution of other client code. Another method, getXMLWait(), is also available but is not recommended. Using getXMLWait() ensures the order of execution, but can cause the application to seem unresponsive, significantly degrading the user experience of any application that uses it. getXMLWait() is not available to scoped applications.**

**A Client Script executes client-side script logic when forms are:**

* **Loaded**
* **Changed**
* **Submitted**
* **Cell edition.**

1. **Server-side scripting**

* **UI actions**
* **Business Rules**
* **Script Include**
* **Data Policies**

**The API’s used in server-side scripting are glide System (gs), Glide Date, Glide date and time, Glide Record and objects like *current* and *previous*.**

1. **On cell edit : Will be applied on List module only.**

**No User must edit cell -State field cell from a List Module.**

**function onCellEdit(sysIDs, table, oldValues, newValue, callback) {**

**var saveAndClose = true;**

**//Type appropriate comment here, and begin script below**

**// old valuses**

**if (newValue == 1) {**

**alert("You cannot change the state field of Incident to 'NEW' from a List Module");**

**saveAndClose = false;**

**}**

**if (newValue == 2) {**

**alert("You cannot change the state field of Incident to 'In Progess' froma a List Module");**

**saveAndClose = false;**

**}**

**if (newValue == 3) {**

**alert("You cannot change the state field of Incident to 'On Hold' froma a List Module");**

**saveAndClose = false;**

**}**

**if (newValue == 6) {**

**alert("You cannot change the state field of Incident to 'Resolved' froma a List Module");**

**saveAndClose = false;**

**}**

**if (newValue == 7) {**

**alert("You cannot change the state field of Incident to 'closed' froma a List Module");**

**saveAndClose = false;**

**}**

**if (newValue == 8) {**

**alert("You cannot change the state field of Incident to 'Canceled' froma a List Module");**

**saveAndClose = false;**

**}**

**callback(saveAndClose);**

**}**

**On Change:**

**getReference(String fieldName, Function callBack)**

**Returns the GlideRecord for a specified field.**

**If a callback function is present, this routine runs asynchronously. The browser (and script) processing continues normally until the server returns the reference value, at which time, the callback function is invoked. If a callback function is not present, this routine runs synchronously and processing halts (causing the browser to appear to hang) while waiting on a server response.**

**Important: It is strongly recommended that you use a callback function.**

**Callback function support for ServiceCatalogForm.getReference is available**

**Get caller details on Change of caller**

**function onChange(control, oldValue, newValue, isLoading, isTemplate) {**

**if (isLoading || newValue === '') {**

**return;**

**}**

**//Type appropriate comment here, and begin script below**

**var vCallerInfo = g\_form.getReference('caller\_id', populateCallerdetails);**

**////This is Synchronus way of calling this is not right approch**

**// always do Asynchornus way of calling**

**// In Asynchronus way of calling You need to Specify callback function**

**//returns glide record of specified field. vcallerinfo variable  contains**

**// all caller information of adela cervantz or the user selected**

**//populateCallerdetails is a call back function**

**function populateCallerdetails(vCallerInfo) {**

**var vFirst\_Name = vCallerInfo.first\_name;**

**var vLast\_Name = vCallerInfo.last\_name;**

**var vemail = vCallerInfo.email;**

**var vManager = vCallerInfo.manager;**

**var vMobilePhone = vCallerInfo.mobile\_phone;**

**g\_form.setValue('u\_callers\_first\_name', vFirst\_Name);**

**g\_form.setValue('u\_callers\_last\_name', vLast\_Name);**

**g\_form.setValue('u\_callers\_email', vemail);**

**g\_form.setValue('u\_callers\_manager', vManager);**

**g\_form.setValue('u\_callers\_mobile\_phone', vMobilePhone);**

**}**

**}**

**Important configuration item**

1. function onChange(control, oldValue, newValue, isLoading, isTemplate) {
2. if (isLoading || newValue === '') {
3. return;
4. }
5. //Type appropriate comment here, and begin script below
6. var affectedCI = g\_form.getReference('cmdb\_ci', checkCI);
7. //it contains glide record of the ci selected.
8. //CheckCI is a call back fucntion
9. function checkCI(affectedCI) {
10. if (affectedCI.u\_important\_config\_item == 'true') {
11. g\_form.setValue('priority', 1);
12. g\_form.setValue('risk', 2);
13. g\_form.setValue('impact', 1);
14. g\_form.setReadOnly('priority', true);
15. g\_form.setReadOnly('risk', true);
16. g\_form.setReadOnly('impact', true);
17. } else {
18. g\_form.setValue('priority', 4);
19. g\_form.setValue('risk', 4);
20. g\_form.setValue('impact', 3);
21. g\_form.setReadOnly('priority', false);
22. g\_form.setReadOnly('risk', false);
23. g\_form.setReadOnly('impact', false);
24. }
25. }
26. }