

ServiceNow 201:

6 - *GlideForm & GlideUser*



Course Outline

1 Course Introduction

2 Development Overview

3 Scripting Locations

4 GlideRecord

5 GlideSystem

6 GlideForm & GlideUser

7 GlideAjax

8 Exploring Other APIs

9 Becoming A Scripting Master

10 Creating A Custom App

Section Outline

1 GlideForm Introduction

2 Show Me The Code!

3 Client Side Environment

4 GlideForm API Diagram

5 Common GlideForm Methods

6 GlideForm Demo

7 GlideUser Introduction

8 Show Me The Code!

9 GlideUser API Diagram

10 GlideUser Methods

11 GlideUser Demo

12 Section Recap

GlideForm Introduction



- Client-side
- Changes to form & fields
- Referenced by g_form

“

The GlideForm API provides methods to customize forms... The global object g_form is used to access GlideForm methods. GlideForm methods are only used on the client... These methods are used to make custom changes to the form view of records.

[ServiceNow Docs](#)

”

Show Me The Code!

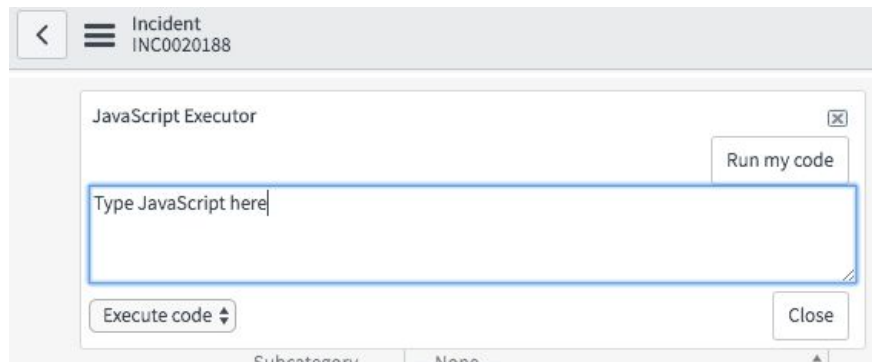
- Set short description to mandatory when priority changes.

Type	<input type="text" value="onChange"/>
Field name	<input type="text" value="Priority"/>

```
1 function onChange(control, oldValue, newValue) {  
2     if (oldValue == newValue)  
3         return;  
4  
5     g_form.setMandatory('short_description', true);  
6  
7 }
```

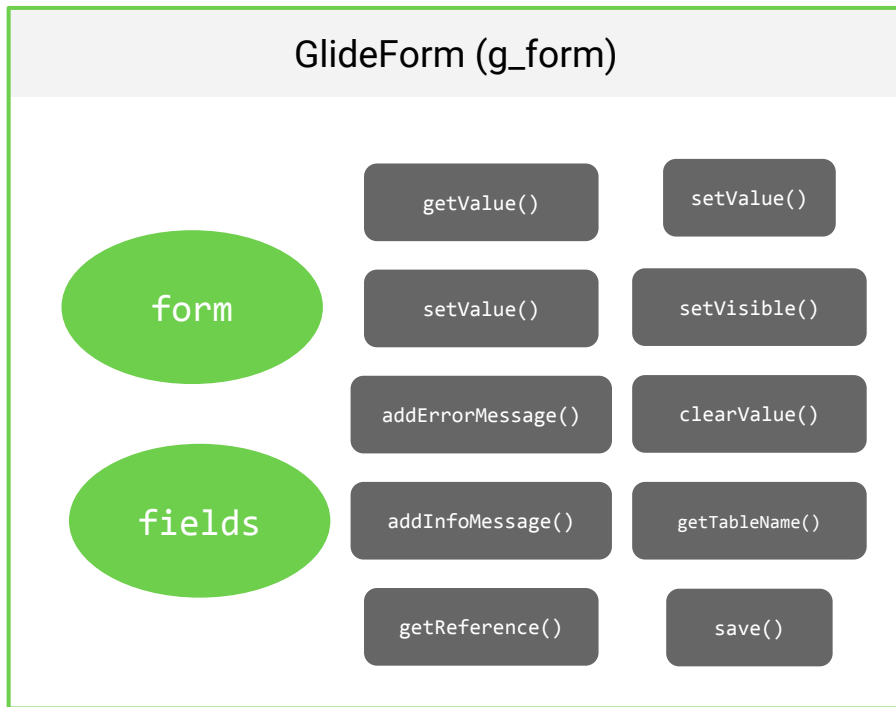
Client-Side Environment

- In browser
- Access to client-side APIs
 - Most are accessible via global scope
- Ctrl + Shift + j
- Debug to browser console
 - `console.log()`
 - `console.dir()`



Caution: Avoid manipulating the DOM directly in ServiceNow w/ client scripts and UI actions, use `g_form` instead

GlideForm API Overview



Common GlideForm Methods

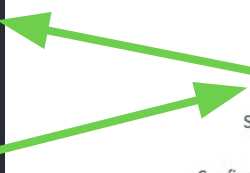
- `addErrorMessage()`
- `addInfoMessage()`
- `addOption()`
- `clearOptions()`
- `clearValue()`
- `disableAttachments()`
- `enableAttachments()`
- `getLabelOf()`
- `getOption()`
- `getReference()`
- `hideRelatedLists()`
- `isMandatory()`
- `removeOption()`
- `setDisabled()`
- `setReadOnly()`
- `setVisible()`
- `setValue()`

GlideForm getValue() & setValue() Methods

- Getter and setter methods for form fields
- `getValue()` accepts the field name as an argument
- `setValue()` accepts the field name and new value as arguments

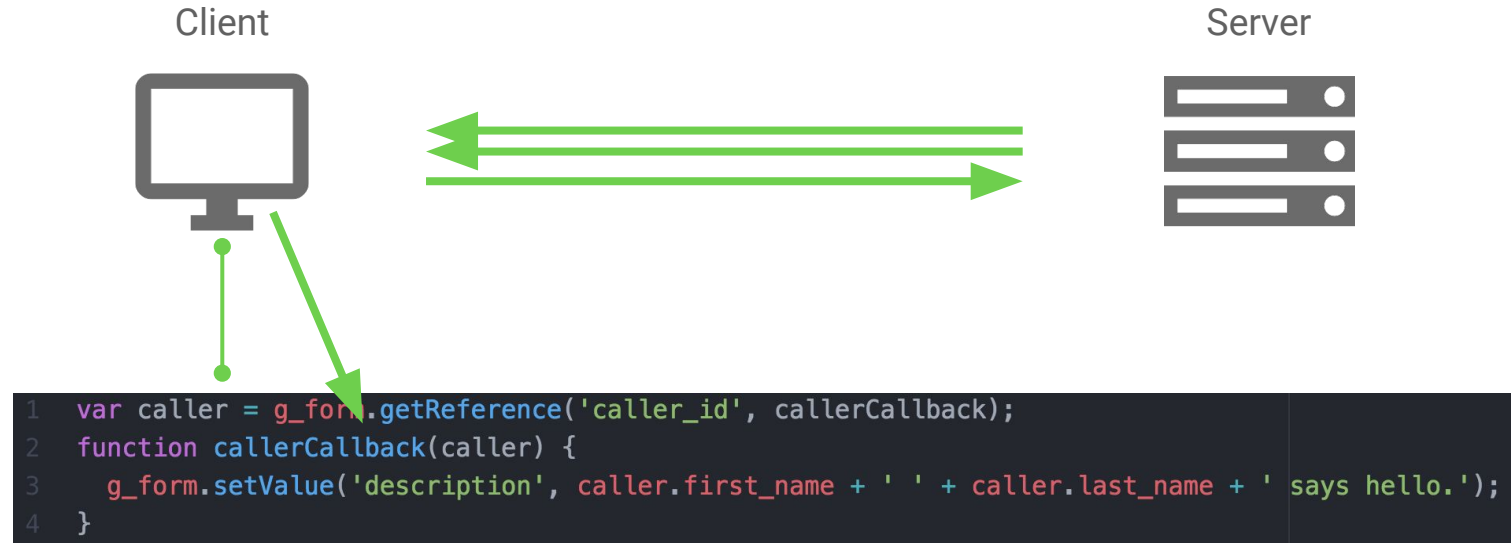
```
1 var category = g_form.getValue('category');  
2  
3 var newCategory = 'software';  
4 g_form.setValue('category', newCategory);
```

Number	<input type="text" value="INC0020188"/>
Caller	<input type="text"/> <input type="button" value="Q"/>
Location	<input type="text"/> <input type="button" value="Q"/>
Category	<input type="text" value="Inquiry / Help"/> <input type="button" value="v"/>
Subcategory	<input type="text" value="-- None --"/> <input type="button" value="v"/>
Configuration item	<input type="text"/> <input type="button" value="Q"/>



GlideForm getReference() Method

- Form only loads fields associated with record on form
- Use `getReference()` to retrieve referenced field values
- Leverages JavaScript callbacks



GlideUser Introduction



- Client-side
- User information
- Referenced by g_user
- Relatively small & simple API



The GlideUser API provides access to information about the current user and current user roles. Using the GlideUser API avoids the need to use the slower GlideRecord queries to get user information.

[ServiceNow Docs](#)

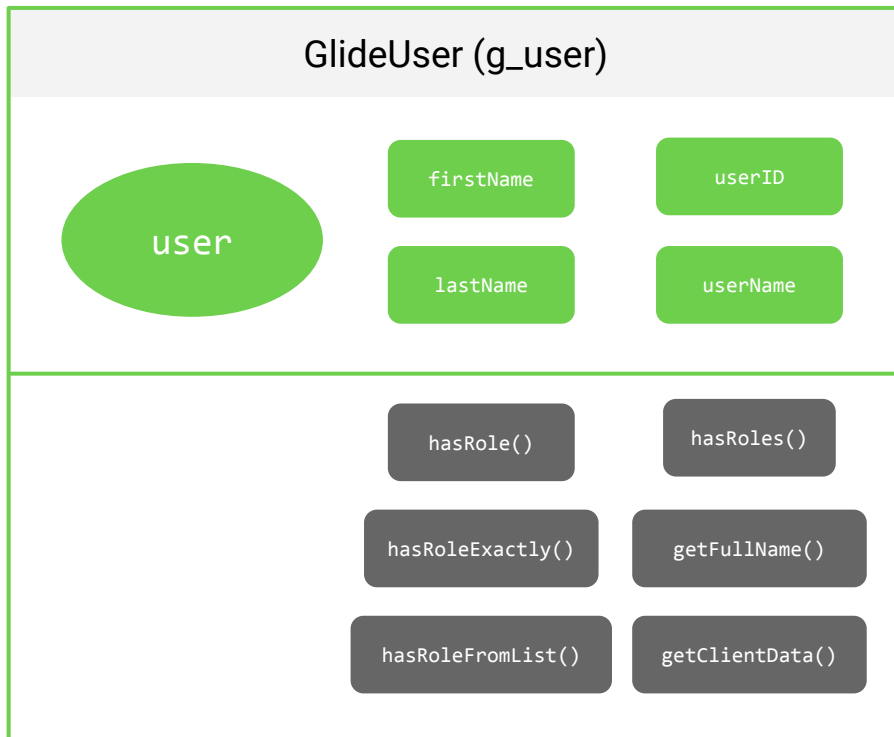


Show Me The Code!

- Check if the current user has the *ITIL* role

```
1  var hasITIL = g_user.hasRole('itil');  
2  if(!hasITIL) {  
3      alert('You do not have sufficient privileges.');
```

GlideUser API Overview



GlideUser Methods & Properties

- `firstName`
- `lastName`
- `userID`
- `userName`
- `getClientData()`
- `getFullName()`
- `hasRole()`
- `hasRoleExactly()`
- `hasRoleFromList()`
- `hasRoles()`

GlideUser hasRoleExactly() Method

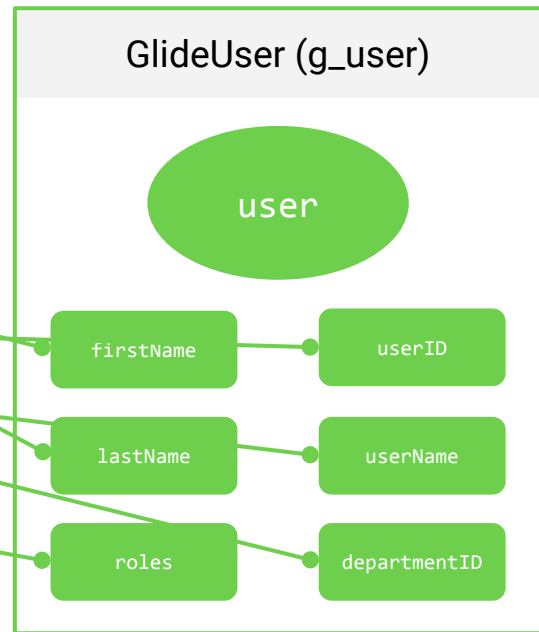
- Takes the name of a role as its argument
- Returns true only if user has the specified role, even if current user is admin

```
1 console.log(g_user.hasRole('approval_admin')); /* returns true */  
2 console.log(g_user.hasRoleExactly('approval_admin')); /* returns false */
```

GlideUser Properties

- Object properties
- Don't need getter methods

```
1 console.log(g_user.firstName + ' ' + g_user.lastName);
2 console.log('User ID: ' + g_user.userID);
3 console.log('Username: ' + g_user.userName);
4 console.log('Department: ' + g_user.departmentID);
5 console.log('Roles: ' + g_user.roles);
```



Where Can I Use This?

Client Side

GlideForm

GlideUser



Client Scripts



UI Actions



UI Scripts



Service Portal

Server Side

GlideRecord

GlideSystem

GlideDateTime



Business Rules



Script Includes



Service Portal



Web Services



UI Actions



Scheduled Jobs



Workflows

Section Recap

- Use GlideForm (g_form) to access information, or manipulate a form in ServiceNow
- Use GlideUser (g_user) to access information about the current user
- Both GlideForm and GlideUser are client-side APIs and cannot be used on the server side
- Use ctrl + shift + j to access the ServiceNow client-side console or use your browser's web console

