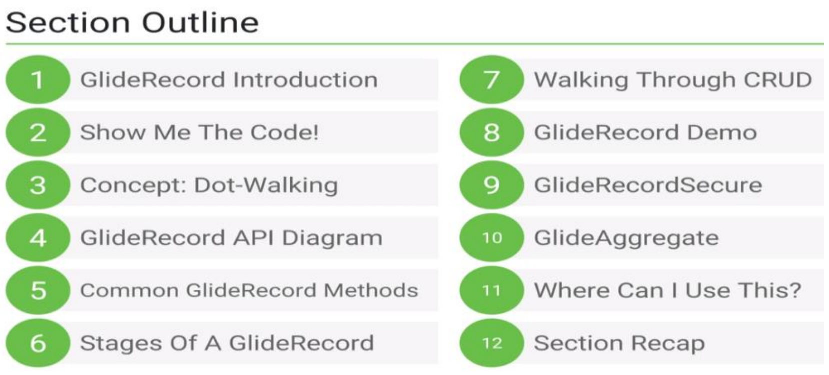
Service Now Development

Scripting Locations in Service Now

1. Business Rules
2. Client Scripts
3. UI Actions
4. Script Include
5. UI policies
6. Workflow Scripting
7. Scheduled Jobs
8. API s
9. Where to customization in service now
10. Transform Maps
11. UI Pages & UI Macros
12. Web Services
13. Service Portal Widgets
14. More……

Glide Record



Section Outline

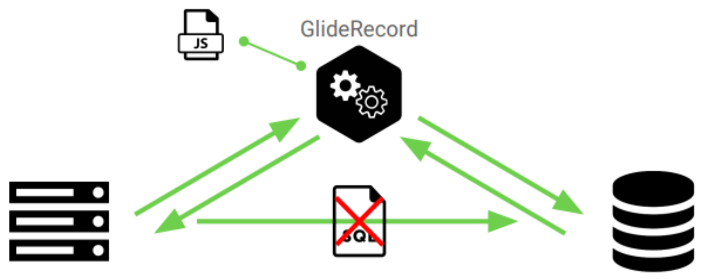
1. GlideRecord Introduction
2. Show Me The Code!
3. Concept: Dot-Walking
4. GlideRecord API Diagram
5. Common GlideRecord Methods
6. Stages Of A GlideRecord
7. Walking Through CRUD
8. GlideRecord Demo
9. GlideRecordSecure
10. GlideAggregate
11. Where Can I Use This?
12. Section Recap

What is Glide Record?

Glide Record is a special Java class (GlideRecord.java) that can be used in JavaScript exactly as if it was a native JavaScript class.

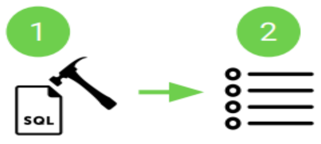
1. It is used for database operations instead of writing **SQL queries.**
2. It is an object that contains **zero or more records from one table**. Another way to say this is that a Glide Record is an ordered list.
3. A Glide Record contains both records **(rows) and fields (columns).** The field names are the same as the underlying database column names.

Application Storage



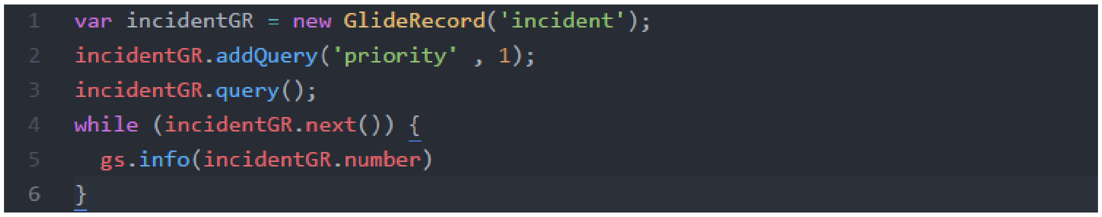
Glide Record Introduction

- Most common API
- Server side
- Used for database operations (CRUD)
- Generates SQL for you
- 2 stages
  - Building a query
  - Process records



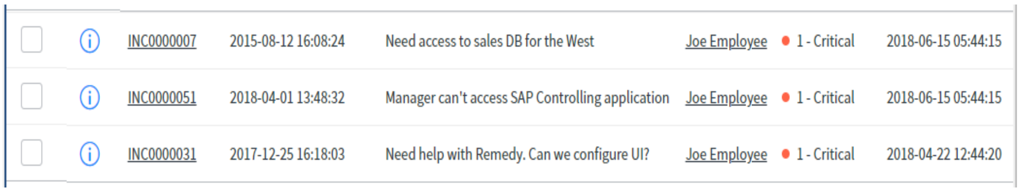
Show Me The Code!

- Print a list of all priority 1 incidents to the screen



```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('priority' , 1);
incidentGR.query();
while (incidentGR.next()) {
  gs.info(incidentGR.number)
}
```

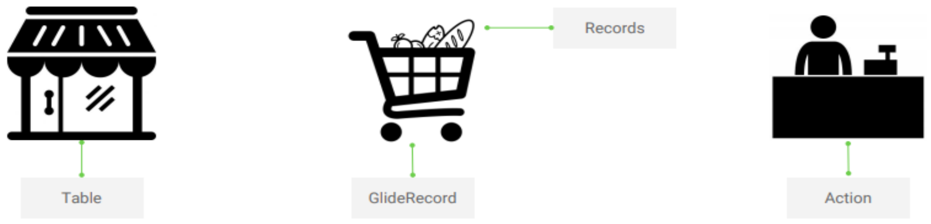
- Print a list of all priority 1 incidents to the screen using by filter condition



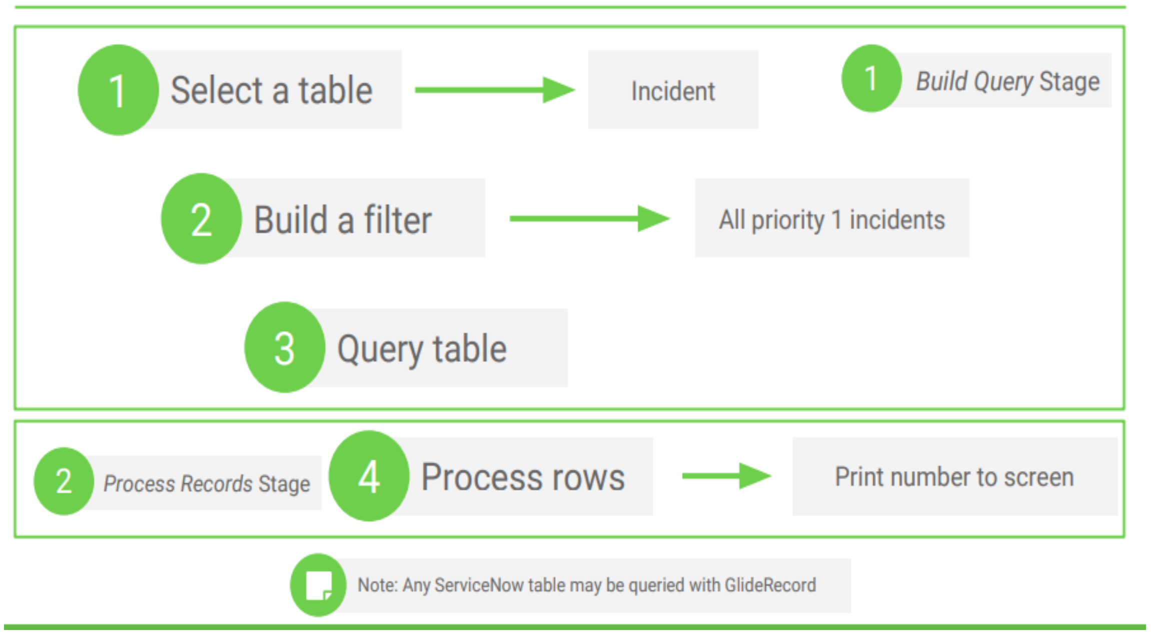
☐	ⓘ	INC0000007	2015-08-12 16:08:24	Need access to sales DB for the West	Joe Employee	● 1 - Critical	2018-06-15 05:44:15
☐	ⓘ	INC0000051	2018-04-01 13:48:32	Manager can't access SAP Controlling application	Joe Employee	● 1 - Critical	2018-06-15 05:44:15
☐	ⓘ	INC0000031	2017-12-25 16:18:03	Need help with Remedy. Can we configure UI?	Joe Employee	● 1 - Critical	2018-04-22 12:44:20

Glide Record by Analogy: Grocery Shopping

1. Go to a grocery store
2. Grab a shopping cart
3. Place groceries in the shopping cart
4. Checkout at cashier

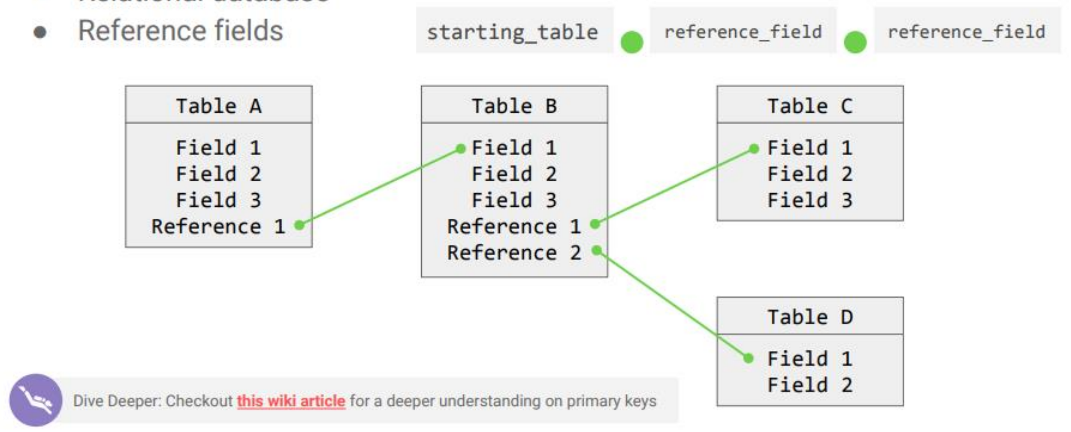


General Glide Record Steps



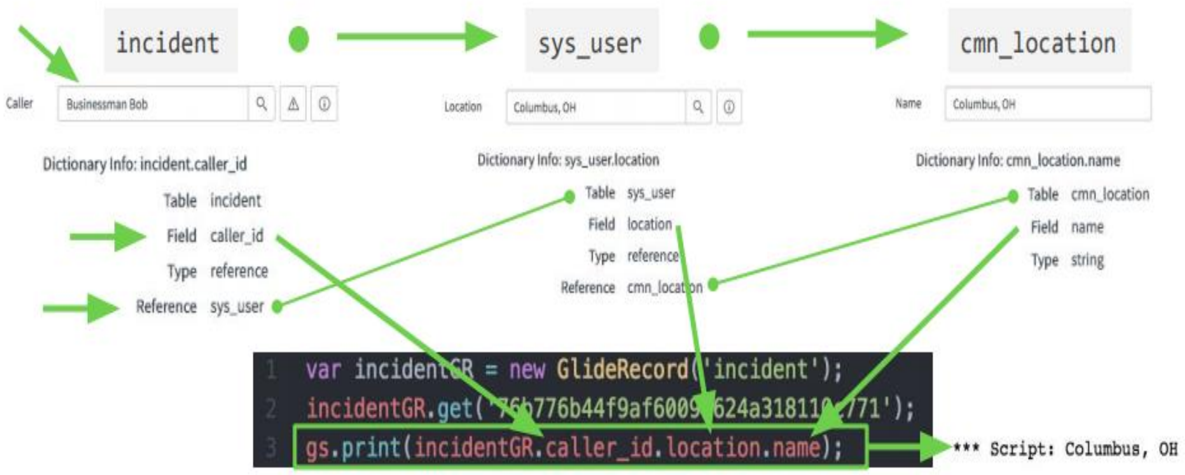
Concept: Dot-Walking

- Relational database
- Reference fields



Example: Dot-Walking

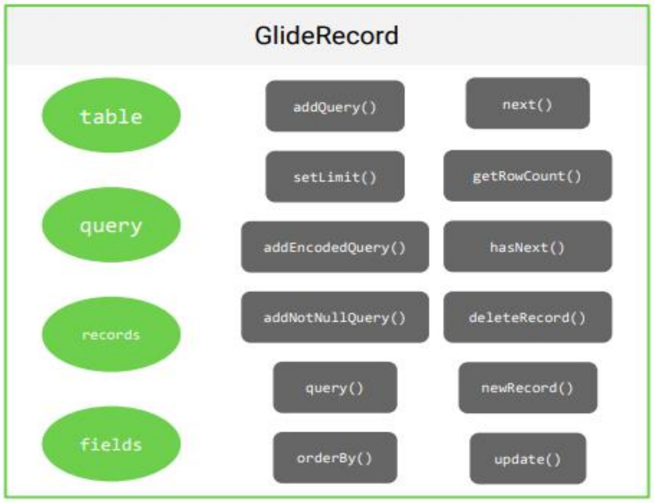
- Example:
  - For a specific incident, you would like to find the location of the caller



```
var incidentGR = new GlideRecord('incident');
incidentGR.get('76b776b44f9af6009[illegible]624a31811[illegible]771');
gs.print(incidentGR.caller_id.location.name);
```

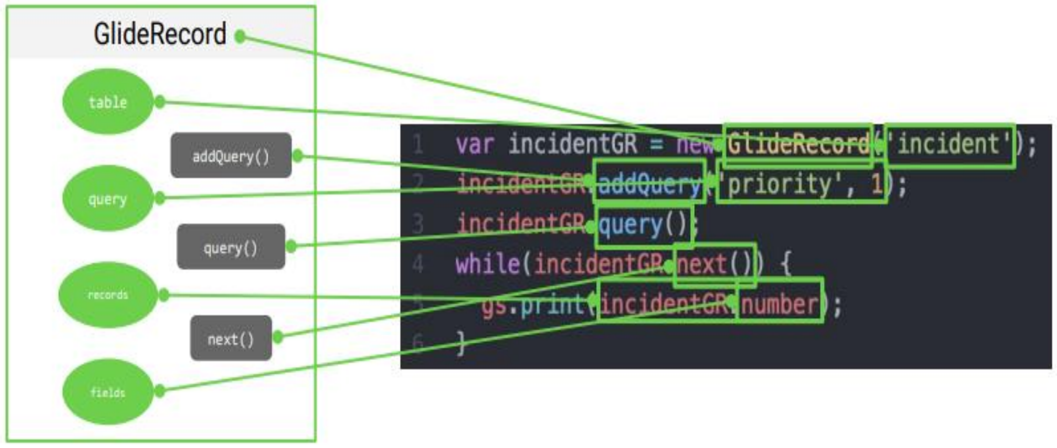
```
*** Script: Columbus, OH
```

Glide Record API Diagram



Note: Ovals represent properties or topics associated with API, while rounded rectangles represent methods

Glide Record API Mapping



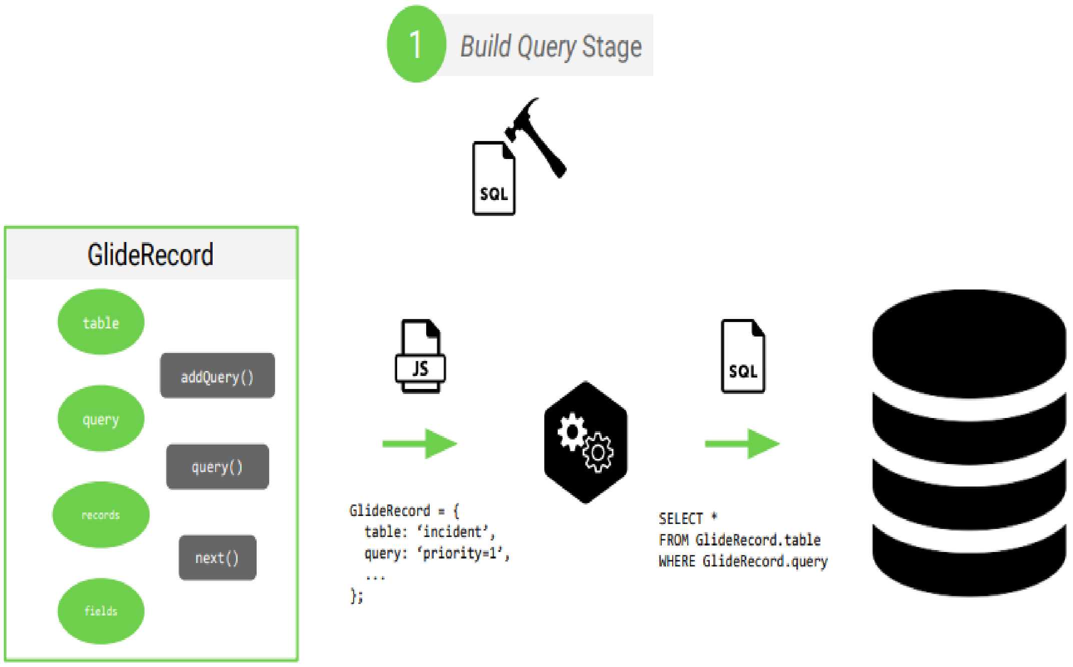
```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('priority', 1);
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number);
}
```

Common Glide Record Methods

- query()
- newRecord()
- insert()
- update()
- deleteRecord()
- addQuery()
- addEncodedQuery()
- hasNext()
- next()
- get()
- orderBy()
- orderByDesc()
- canCreate()
- canWrite()

- canRead()

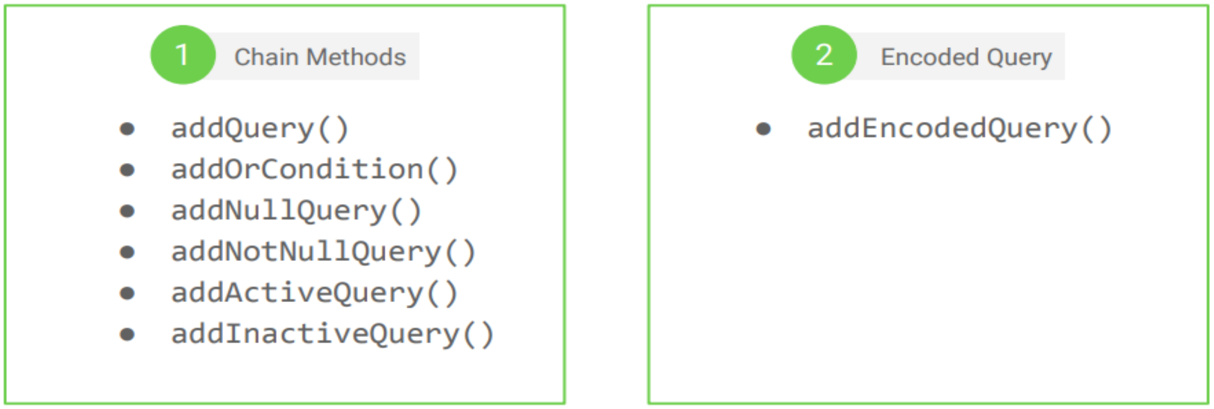
Glide Record Stage 1: Building Query



```
GlideRecord = {
  table: 'incident',
  query: 'priority=1',
  ...
};
```

```
SELECT *
FROM GlideRecord.table
WHERE GlideRecord.query
```

Glide Record Stage 1: Options to Build Queries



1 Chain Methods

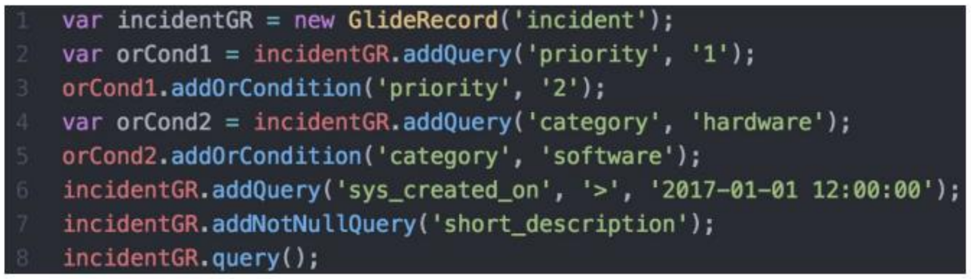
- addQuery()
- addOrCondition()
- addNullQuery()
- addNotNullQuery()
- addActiveQuery()
- addInactiveQuery()

2 Encoded Query

- addEncodedQuery()

Glide Record Stage 1: Option 1 - Chain Methods

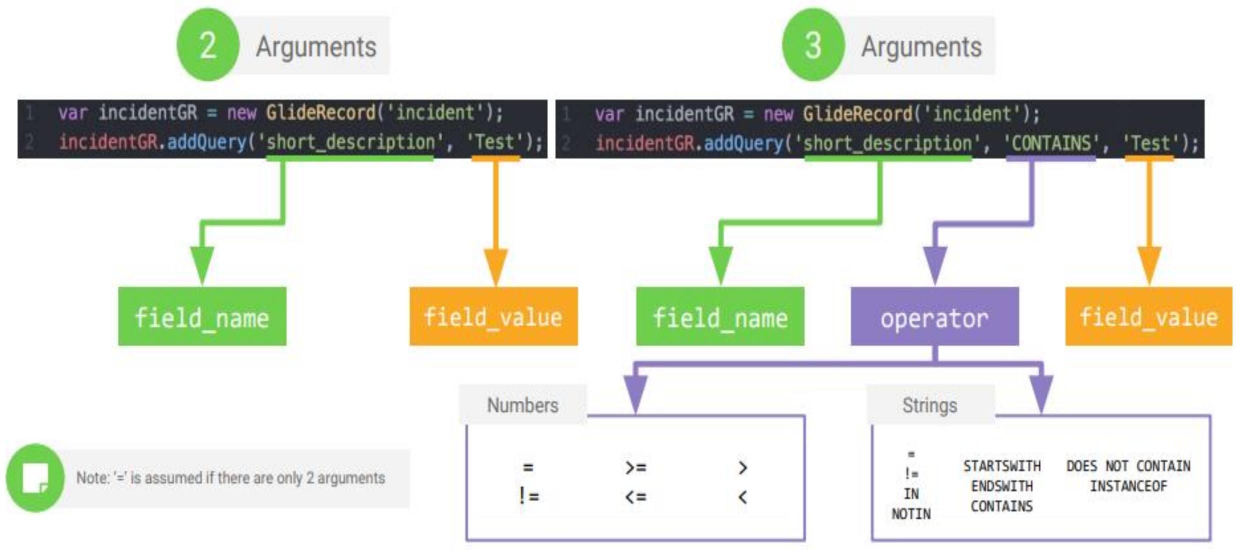
Add GlideRecord methods onto the current GlideRecord object



```
var incidentGR = new GlideRecord('incident');
var orCond1 = incidentGR.addQuery('priority', '1');
orCond1.addOrCondition('priority', '2');
var orCond2 = incidentGR.addQuery('category', 'hardware');
orCond2.addOrCondition('category', 'software');
incidentGR.addQuery('sys_created_on', '>', '2017-01-01 12:00:00');
incidentGR.addNotNullQuery('short_description');
incidentGR.query();
```

Glide Record addQuery() Method

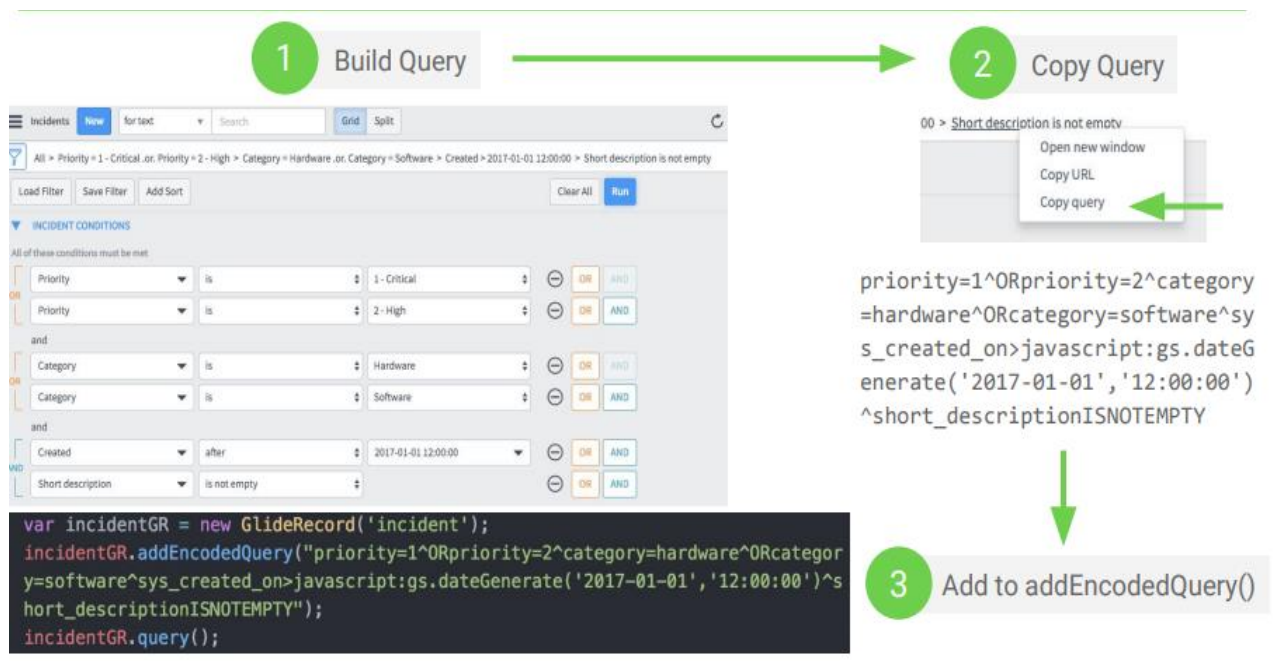
- Accepts 2 or 3 arguments



```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('short_description', 'Test');
```

```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('short_description', 'CONTAINS', 'Test');
```

Glide Record Stage 1: Option 2 - Encoded Query



All > Priority = 1 - Critical .or. Priority = 2 - High > Category = Hardware .or. Category = Software > Created > 2017-01-01 12:00:00 > Short description is not empty

All of these conditions must be met

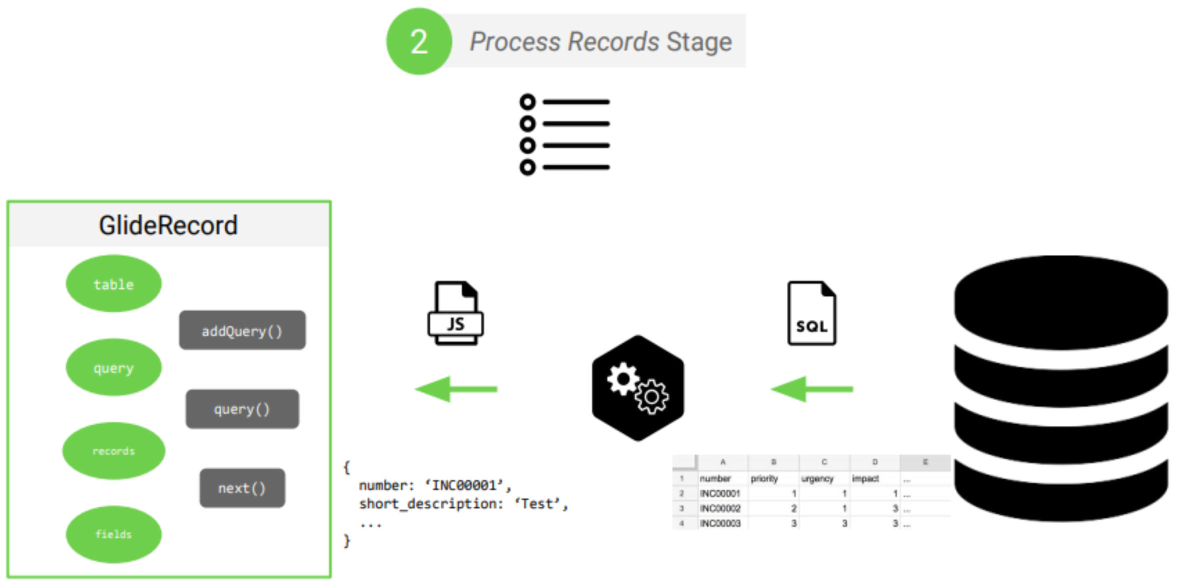
Priority	is	1 - Critical
Priority	is	2 - High
and		
Category	is	Hardware
Category	is	Software
and		
Created	after	2017-01-01 12:00:00
Short description	is not empty	

00 > Short description is not empty

```
priority=1^ORpriority=2^category
=hardware^ORcategory=software^sy
s_created_on>javascript:gs.dateG
enerate('2017-01-01','12:00:00')
^short_descriptionISNOTEMPTY
```

```
var incidentGR = new GlideRecord('incident');
incidentGR.addEncodedQuery("priority=1^ORpriority=2^category=hardware^ORcategor
y=software^sys_created_on>javascript:gs.dateGenerate('2017-01-01','12:00:00')^s
hort_descriptionISNOTEMPTY");
incidentGR.query();
```

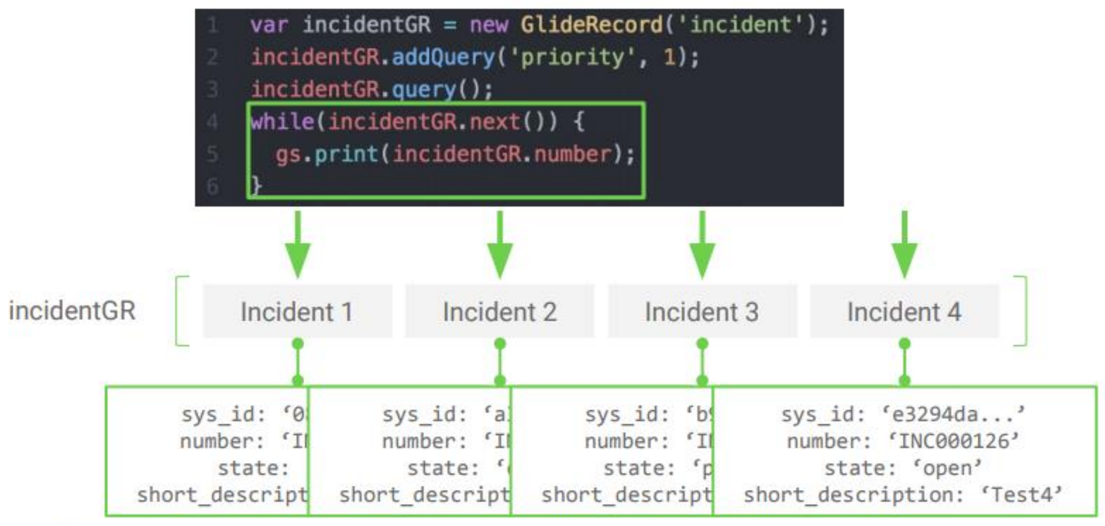
Glide Record Stage 2: Process Records



```
{
  number: 'INC00001',
  short_description: 'Test',
  ...
}
```

	A	B	C	D	E
1	number	priority	urgency	impact	...
2	INC00001	1	1	1	...
3	INC00002	2	1	3	...
4	INC00003	3	3	3	...

Glide Record next() Method & Iteration



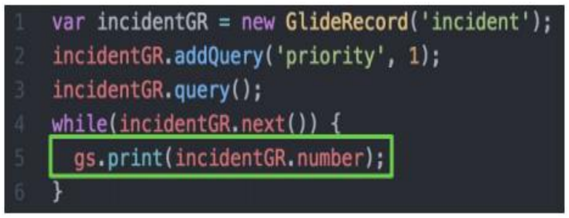
```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('priority', 1);
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number);
}
```

```
        sys_id: '0
        number: 'I
         state:
short_descript

        sys_id: 'a
        number: 'I
         state: '
short_descript

        sys_id: 'b
        number: 'I
         state: 'p
short_descript

        sys_id: 'e3294da...'
        number: 'INC000126'
         state: 'open'
short_description: 'Test4'
```

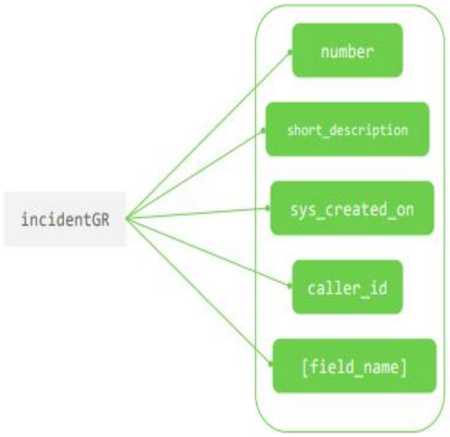
Dive Deeper: Checkout **this wiki article** and this YouTube video if you'd like to learn more on iterators

Accessing a Record's Fields

- Once query() method is executed and stage 2 begins, all fields are just a dot away
- Fields become GlideRecord properties

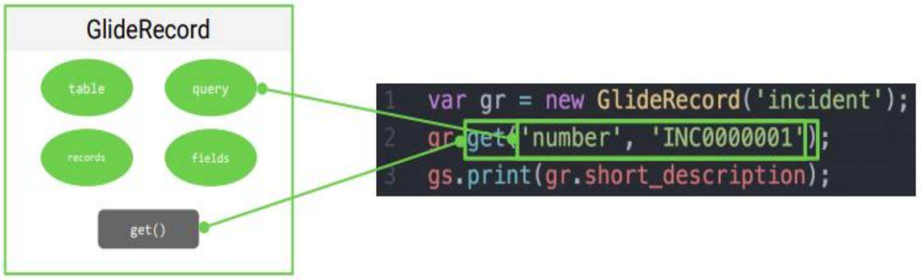


```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('priority', 1);
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number);
}
```



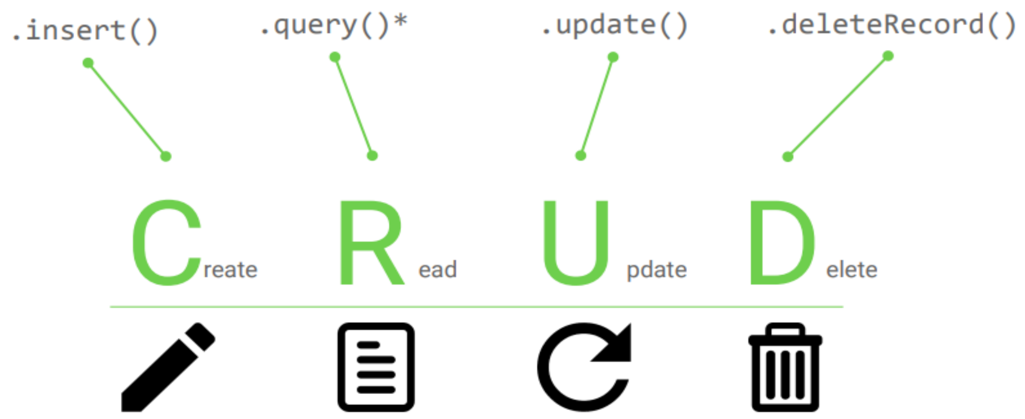
Glide Record get() Method

- Shortcut
- Only grabs 1 record
- Commonly used with record sys_id



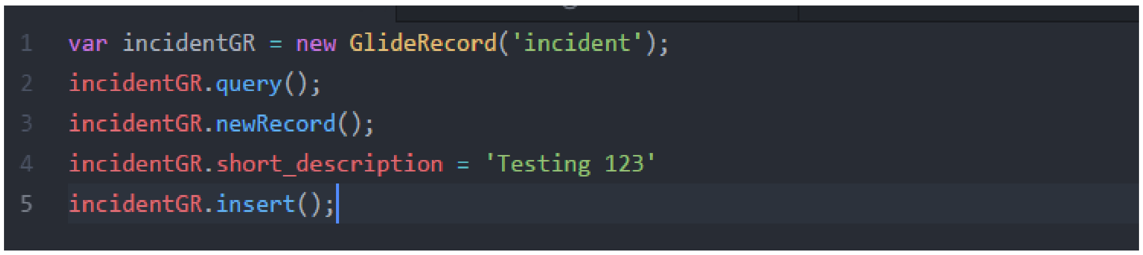
```
var gr = new GlideRecord('incident');
gr.get('number', 'INC0000001');
gs.print(gr.short_description);
```

CRUD Glide Record Mapping



CRUD - Create

1. Build GlideRecord
2. query()
3. newRecord()
4. Set field values
5. insert()

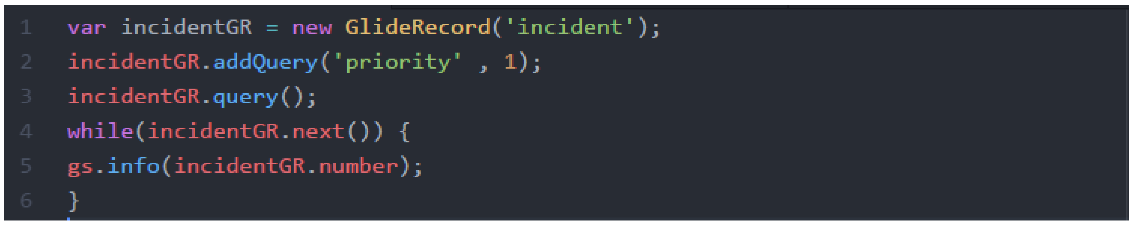


```
var incidentGR = new GlideRecord('incident');
incidentGR.query();
incidentGR.newRecord();
incidentGR.short_description = 'Testing 123'
incidentGR.insert();
```

Note: After executed this script check in incident table

CRUD - Read

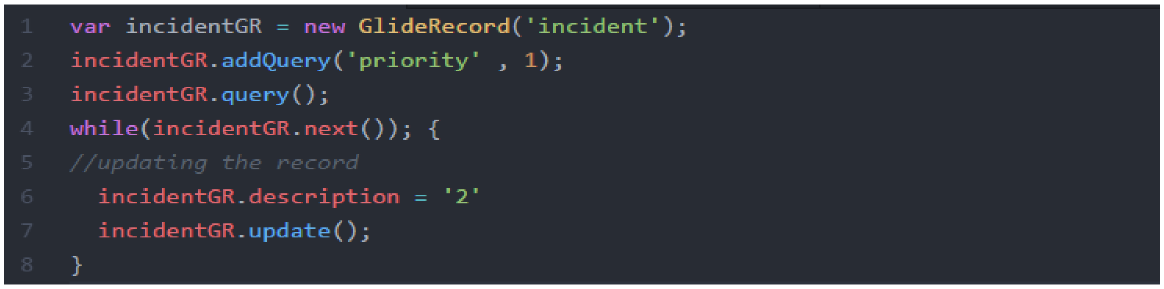
1. Build Glide Record
2. Add filter conditions (optional)
3. query()
4. next()
5. Print or copy variables



```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('priority' , 1);
incidentGR.query();
while(incidentGR.next()) {
gs.info(incidentGR.number);
}
```

CRUD - Update

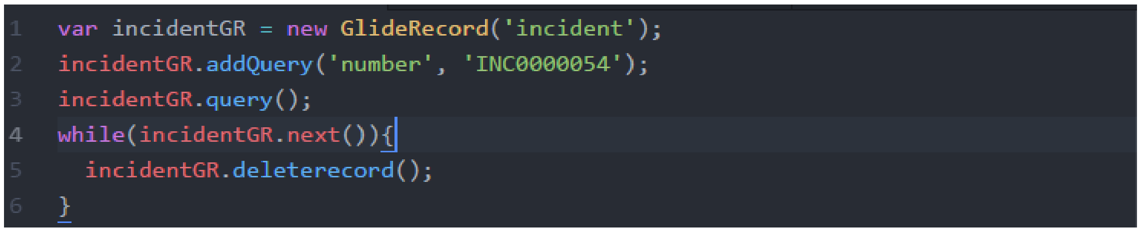
1. Build Glide Record
2. Add filter conditions (optional)
3. query()
4. next()
5. Set field values
6. update()



```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('priority' , 1);
incidentGR.query();
while(incidentGR.next()); {
//updating the record
  incidentGR.description = '2'
  incidentGR.update();
}
```

CRUD - Delete

1. Build GlideRecord
2. Add filter conditions (optional)
3. query()
4. next()
5. deleteRecord()



```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('number', 'INC0000054');
incidentGR.query();
while(incidentGR.next()){
  incidentGR.deleterecord();
}
```

Glide Record Secure

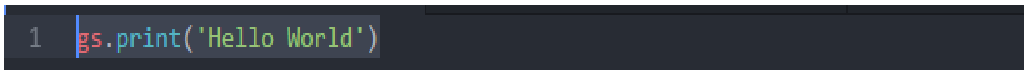
- Glide Record Secure class is inherited from Glide Record

  ✓ Has all of the same methods
  ✓ Performs ACL checking

- Used to secure Script Includes
- Replaces canWrite(), canRead(), canUpdate(), canDelete() GlideRecord methods

Dive Deeper: Watch episode 15 of TechNow to learn more about GlideRecordSecure

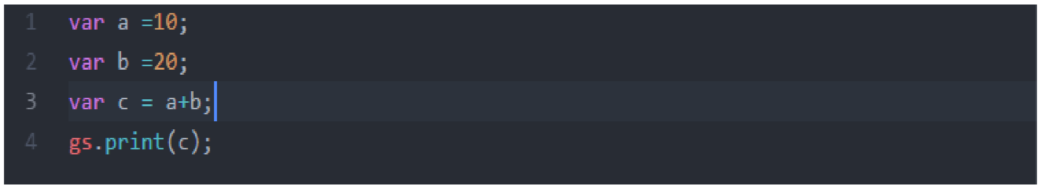
Working with gs.print ()method



```
gs.print('Hello World')
```

Result: Just print the Output is "Hellow World"

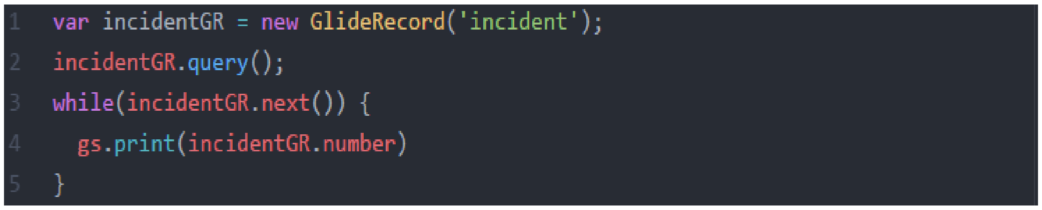
Adding two numbers a simple program:



```
var a =10;
var b =20;
var c = a+b;
gs.print(c);
```

Result: given the total = 30

Working with query () method:

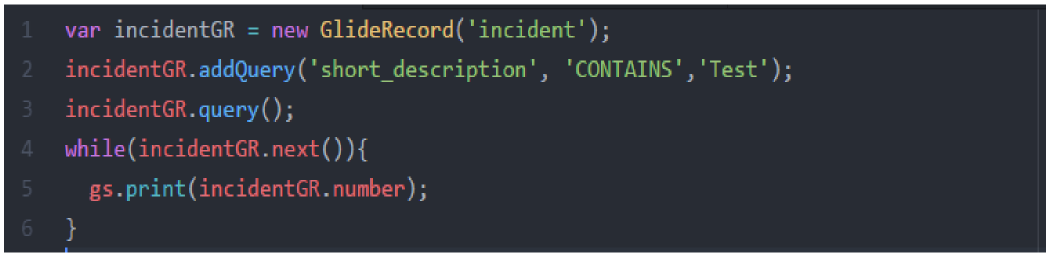


```
var incidentGR = new GlideRecord('incident');
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number)
}
```

Result: Display all the incident numbers in a specified table

Working with addQuery () method

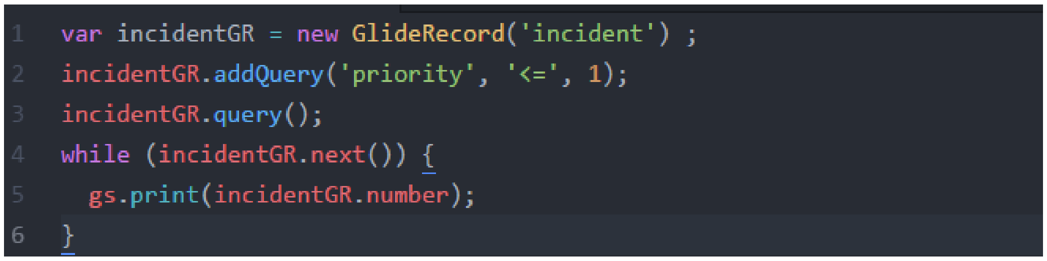
Example 1:



```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('short_description', 'CONTAINS','Test');
incidentGR.query();
while(incidentGR.next()){
  gs.print(incidentGR.number);
}
```

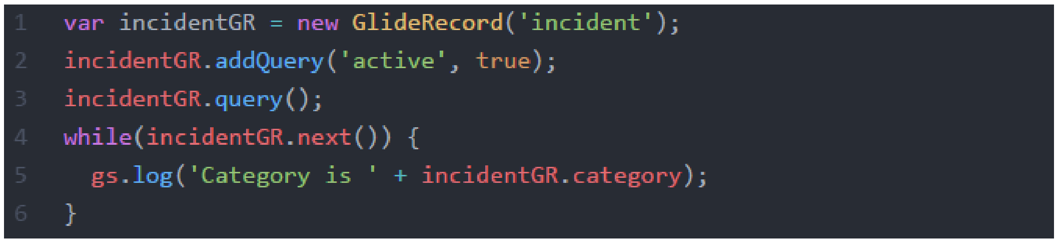
Result: Given the incident number where the description field is “Test” value

Example 2:



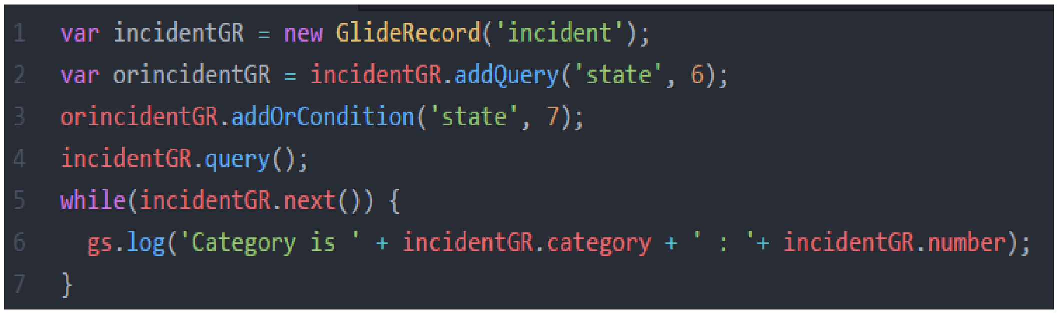
```
var incidentGR = new GlideRecord('incident') ;
incidentGR.addQuery('priority', '<=', 1);
incidentGR.query();
while (incidentGR.next()) {
  gs.print(incidentGR.number);
}
```

Example 3:



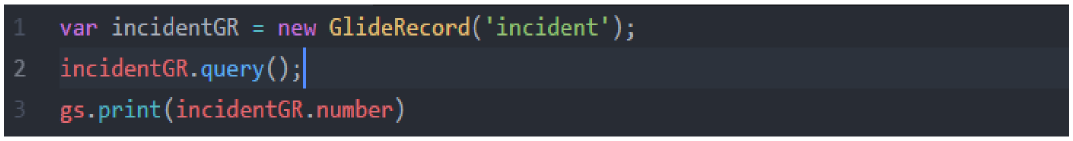
```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('active', true);
incidentGR.query();
while(incidentGR.next()) {
  gs.log('Category is ' + incidentGR.category);
}
```

Working addOrCondition () method



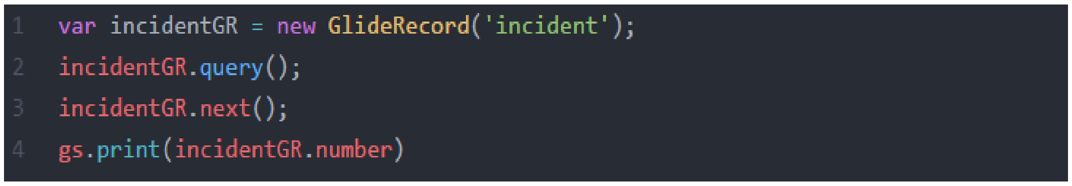
```
var incidentGR = new GlideRecord('incident');
var orincidentGR = incidentGR.addQuery('state', 6);
orincidentGR.addOrCondition('state', 7);
incidentGR.query();
while(incidentGR.next()) {
  gs.log('Category is ' + incidentGR.category + ' : '+ incidentGR.number);
}
```

Working with next () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.query();
gs.print(incidentGR.number)
```

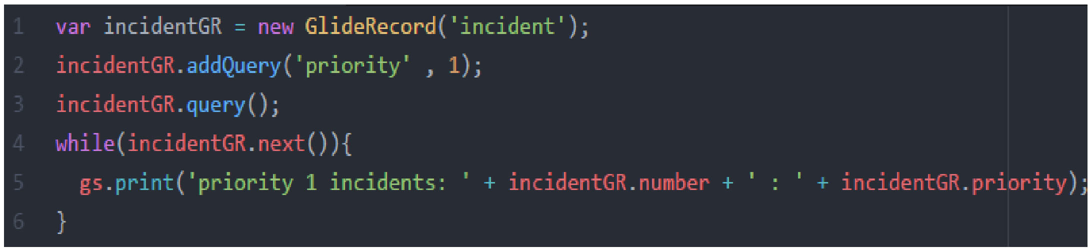
Result: Empty



```
var incidentGR = new GlideRecord('incident');
incidentGR.query();
incidentGR.next();
gs.print(incidentGR.number)
```

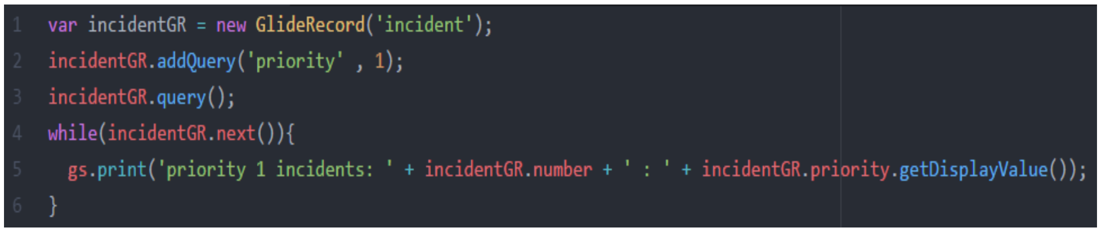
Result: Given first incident number from incident Table

Working with addQuery () method



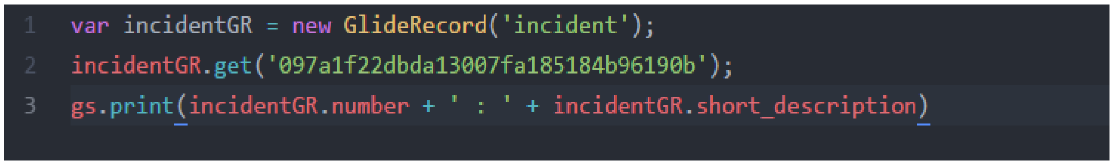
```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('priority' , 1);
incidentGR.query();
while(incidentGR.next()){
  gs.print('priority 1 incidents: ' + incidentGR.number + ' : ' + incidentGR.priority);
}
```

Result: Given all priority 1 incidents from incident Table



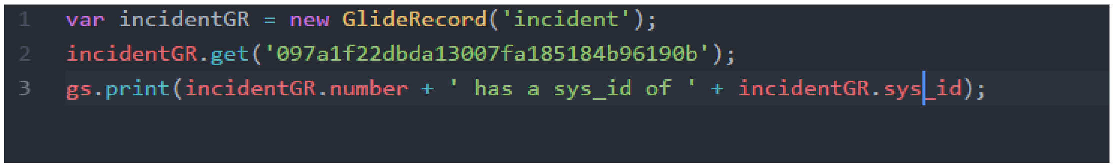
```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('priority' , 1);
incidentGR.query();
while(incidentGR.next()){
  gs.print('priority 1 incidents: ' + incidentGR.number + ' : ' + incidentGR.priority.getDisplayValue());
}
```

Working with get () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.get('097a1f22dbda13007fa185184b96190b');
gs.print(incidentGR.number + ' : ' + incidentGR.short_description)
```

Result: Given incident number and short description of sys_id

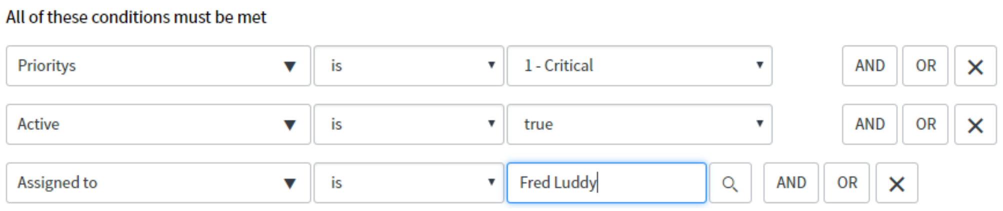


```
var incidentGR = new GlideRecord('incident');
incidentGR.get('097a1f22dbda13007fa185184b96190b');
gs.print(incidentGR.number + ' has a sys_id of ' + incidentGR.sys_id);
```

Working with addEncodedQuery () method

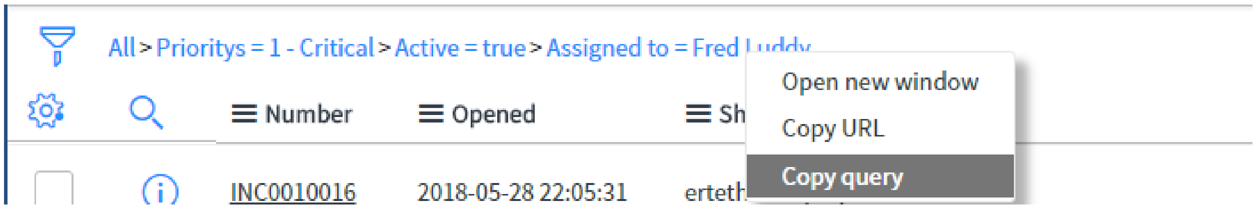
Step 1: Navigate to Incident list view

Step 2: Build the query using by filter condition shown below → Run


All of these conditions must be met

Prioritys	is	1 - Critical
Active	is	true
Assigned to	is	Fred Luddy

Step 3: Copy the query from list view


All > Prioritys = 1 - Critical > Active = true > Assigned to = Fred Luddy

INC0010016 2018-05-28 22:05:31

Copy Query = **priority=1^active=true^assigned_to=5137153cc611227c000bbd1bd8cd2005**

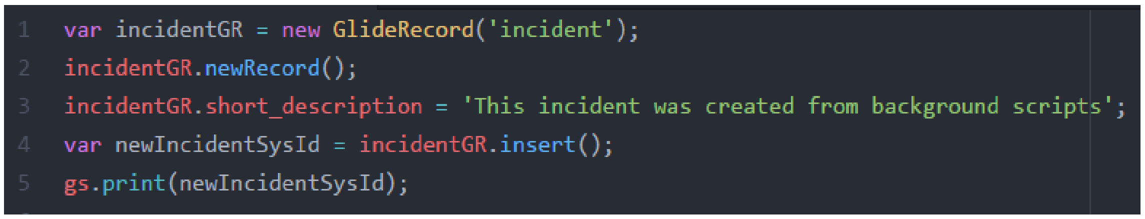
```
var queryString = 'priority=1^active=true^assigned_to=5137153cc611227c000bbd1bd8cd2005';
var incidentGR = new GlideRecord('incident');
incidentGR.addEncodedQuery(queryString);
incidentGR.query();
while(incidentGR.next()){
  gs.print(incidentGR.number);
}
```

Result: Given incidents based on you condition query matched

Working with newRecord insert () method

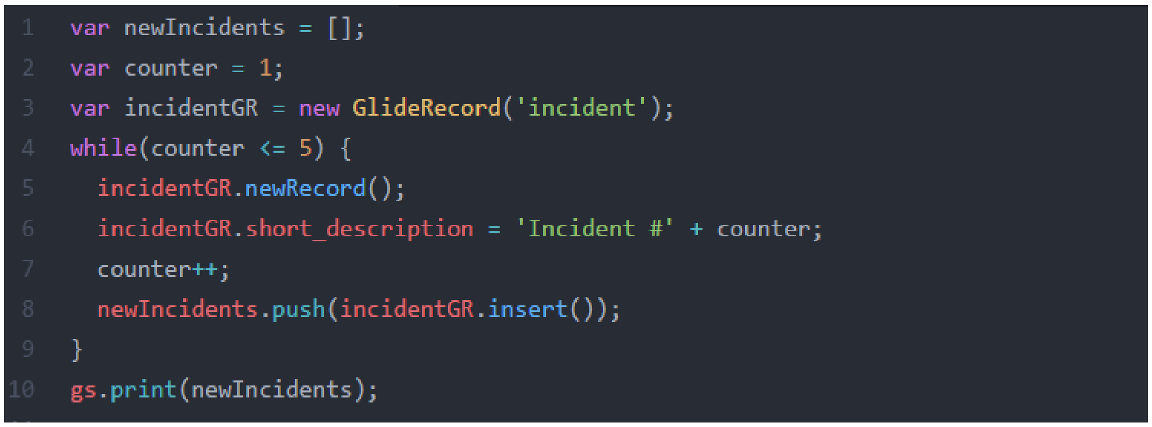
```
var incidentGR = new GlideRecord('incident');
incidentGR.newRecord();
incidentGR.short_description = 'This incident was created from background scripts';
incidentGR.insert();
```

Result: A new record is created in Incident table and navigates to list view to confirm



```
var incidentGR = new GlideRecord('incident');
incidentGR.newRecord();
incidentGR.short_description = 'This incident was created from background scripts';
var newIncidentSysId = incidentGR.insert();
gs.print(newIncidentSysId);
```

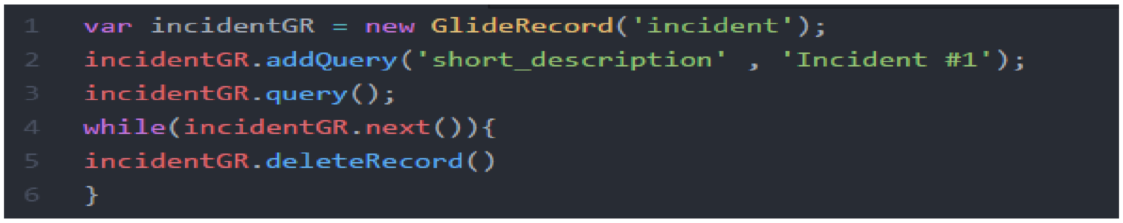
Working with create multiple records insert () method



```
var newIncidents = [];
var counter = 1;
var incidentGR = new GlideRecord('incident');
while(counter <= 5) {
  incidentGR.newRecord();
  incidentGR.short_description = 'Incident #' + counter;
  counter++;
  newIncidents.push(incidentGR.insert());
}
gs.print(newIncidents);
```

Result: Created multiple records then check in to your list view

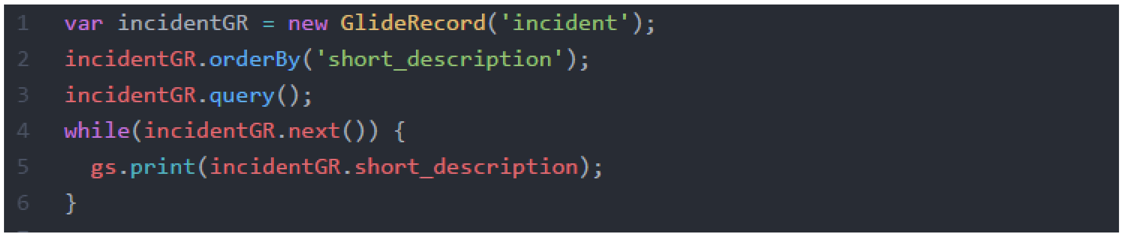
Working with delete records () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('short_description' , 'Incident #1');
incidentGR.query();
while(incidentGR.next()){
incidentGR.deleteRecord()
}
```

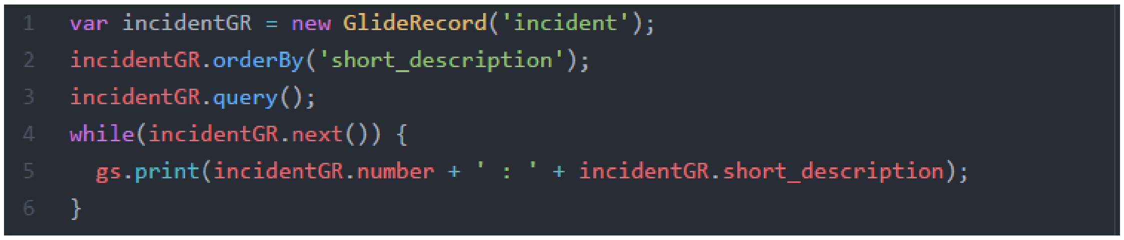
Result: Deleted a record then check in to your list view

Working with orderBy () method



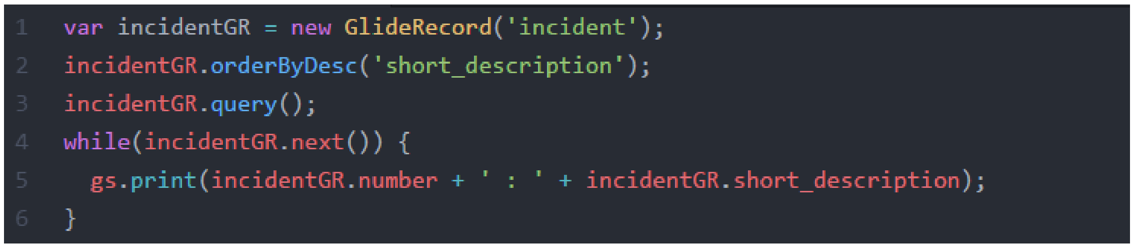
```
var incidentGR = new GlideRecord('incident');
incidentGR.orderBy('short_description');
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.short_description);
}
```

Result: Returned all the incidents short_descrption value in Order by Ascending



```
var incidentGR = new GlideRecord('incident');
incidentGR.orderBy('short_description');
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number + ' : ' + incidentGR.short_description);
}
```

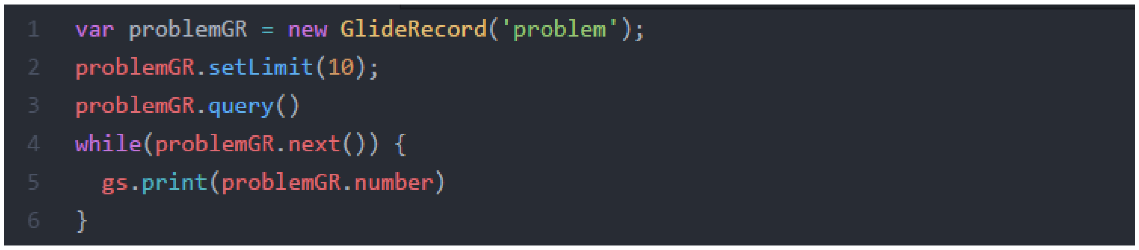
Working with orderByDesc () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.orderByDesc('short_description');
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number + ' : ' + incidentGR.short_description);
}
```

Result: Returned all the incidents short_descrption value in Order by Descending

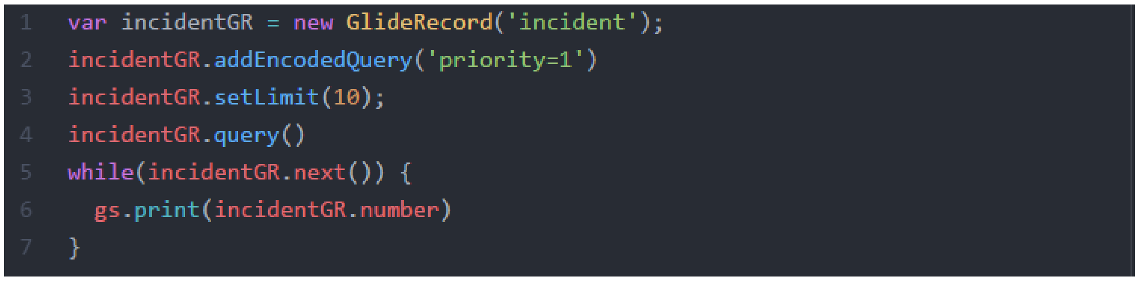
Working with setLimit () method



```
var problemGR = new GlideRecord('problem');
problemGR.setLimit(10);
problemGR.query()
while(problemGR.next()) {
  gs.print(problemGR.number)
}
```

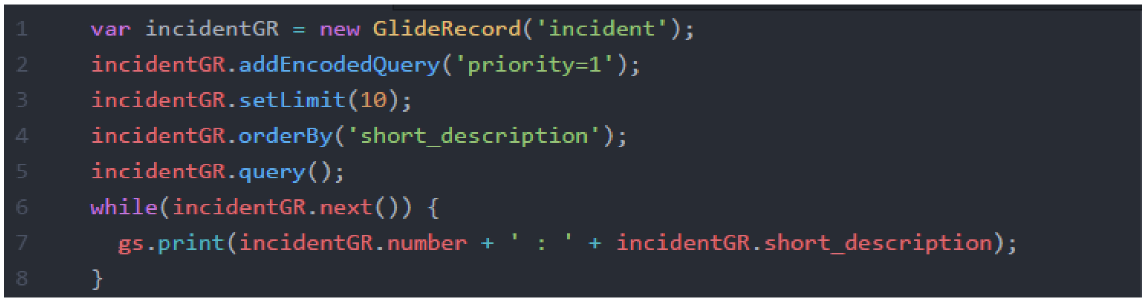
Result: Returned 10 records from problem table for setLimit

Working with setLimit () method



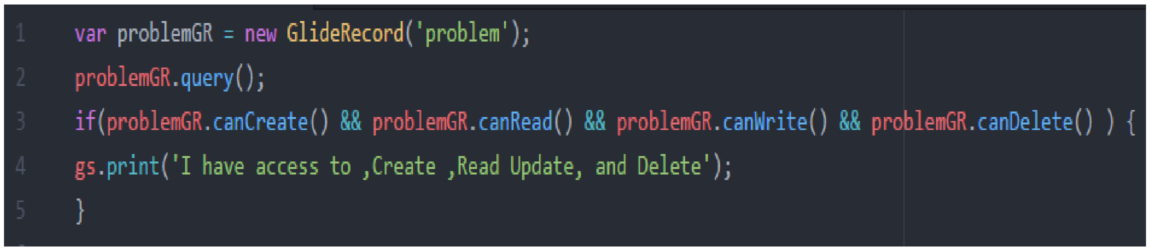
```
var incidentGR = new GlideRecord('incident');
incidentGR.addEncodedQuery('priority=1')
incidentGR.setLimit(10);
incidentGR.query()
while(incidentGR.next()) {
  gs.print(incidentGR.number)
}
```

Example 2:



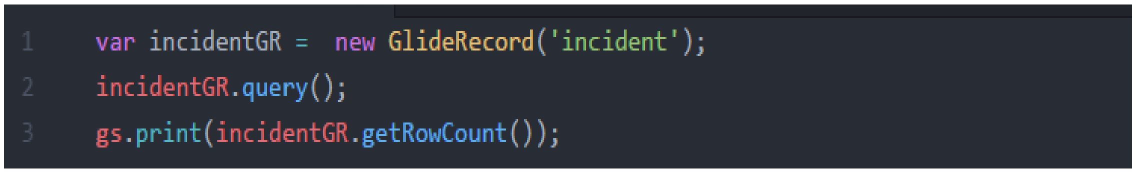
```
var incidentGR = new GlideRecord('incident');
incidentGR.addEncodedQuery('priority=1');
incidentGR.setLimit(10);
incidentGR.orderBy('short_description');
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number + ' : ' + incidentGR.short_description);
}
```

Working with Access Control List canCreate, canWrite, canRead, canDelete



```
var problemGR = new GlideRecord('problem');
problemGR.query();
if(problemGR.canCreate() && problemGR.canRead() && problemGR.canWrite() && problemGR.canDelete() ) {
gs.print('I have access to ,Create ,Read Update, and Delete');
}
```

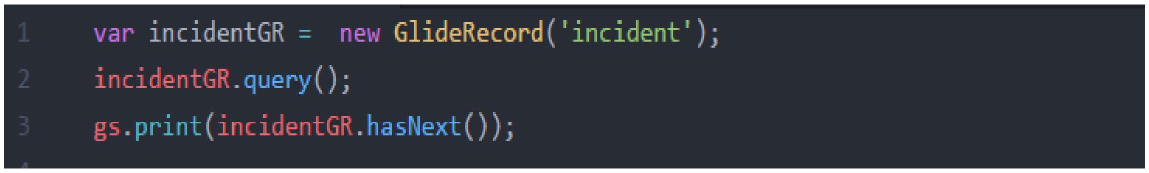
Working with getRowCount () method



```
var incidentGR =  new GlideRecord('incident');
incidentGR.query();
gs.print(incidentGR.getRowCount());
```

Result: Returned total records in incident Table

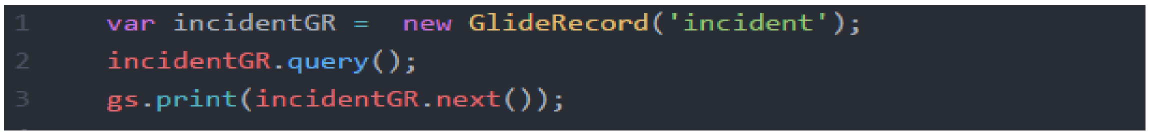
Working with hasNext () method



```
var incidentGR =  new GlideRecord('incident');
incidentGR.query();
gs.print(incidentGR.hasNext());
```

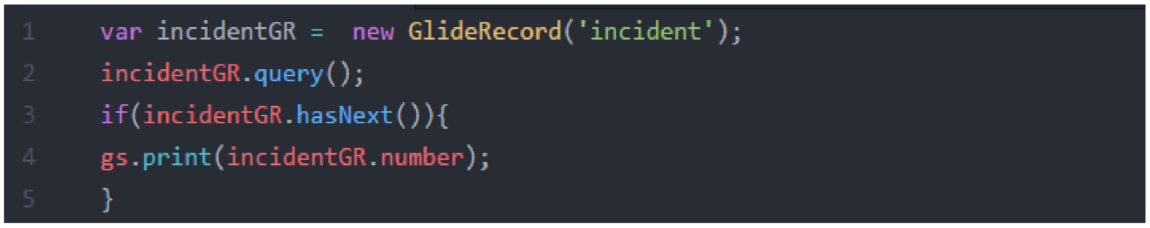
Result: Returned bullion value as a True

Example 2:



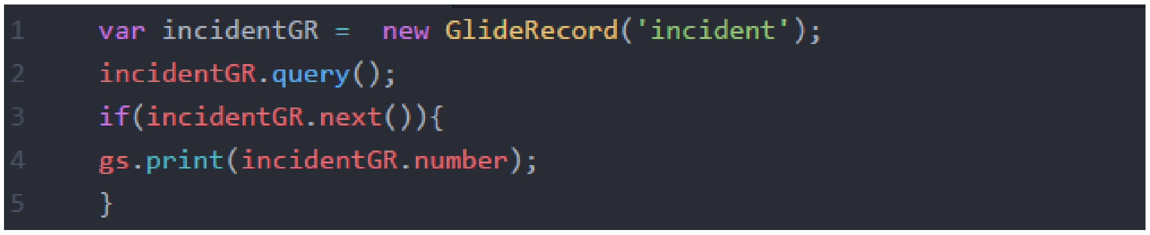
```
var incidentGR =  new GlideRecord('incident');
incidentGR.query();
gs.print(incidentGR.next());
```

Example 3: This is not working properly the hasNext method



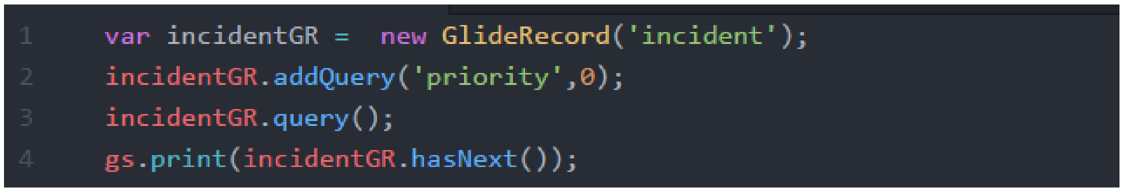
```
var incidentGR =  new GlideRecord('incident');
incidentGR.query();
if(incidentGR.hasNext()){
gs.print(incidentGR.number);
}
```

Example 4: This will work properly and returned first record from incident table



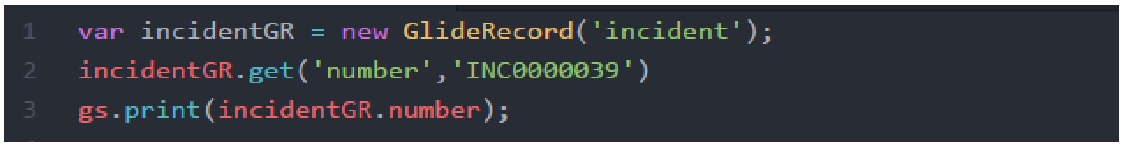
```
var incidentGR =  new GlideRecord('incident');
incidentGR.query();
if(incidentGR.next()){
gs.print(incidentGR.number);
}
```

Example 5:



```
var incidentGR =  new GlideRecord('incident');
incidentGR.addQuery('priority',0);
incidentGR.query();
gs.print(incidentGR.hasNext());
```

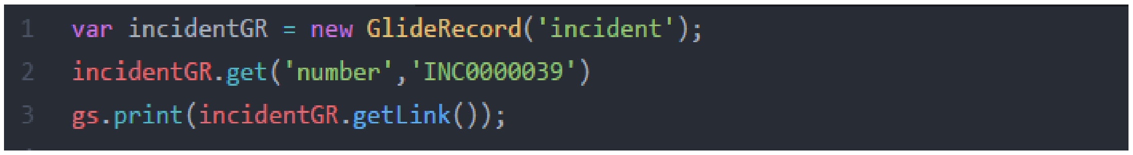
Working with get () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.get('number','INC0000039')
gs.print(incidentGR.number);
```

Result: Returned incident number as provided in get method

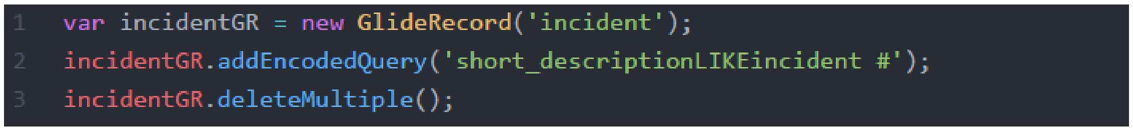
Working with getLink () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.get('number','INC0000039')
gs.print(incidentGR.getLink());
```

Result:
incident.do?sys_id=471bfbc7a9fe198101e77a3e10e5d47f&sysparm_stack=incident_list.do?sysparm_query=active=true

Working with deleteMultiple () method

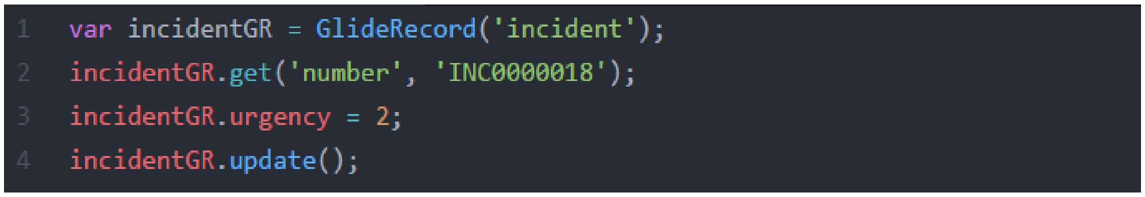


```
var incidentGR = new GlideRecord('incident');
incidentGR.addEncodedQuery('short_descriptionLIKEincident #');
incidentGR.deleteMultiple();
```

Result: Deleted multiple records form expected table

Working with update () method to change urgency

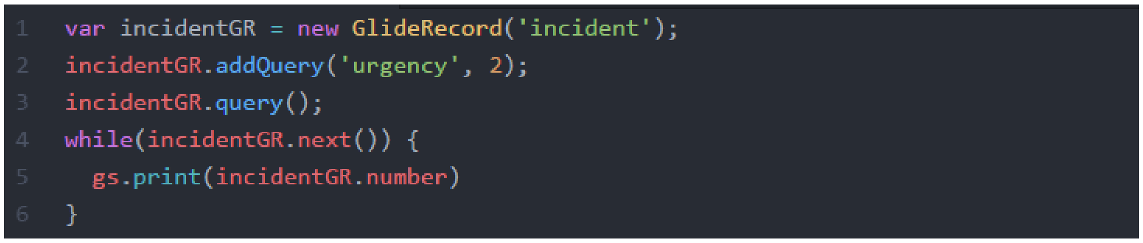
Example 1:



```
var incidentGR = GlideRecord('incident');
incidentGR.get('number', 'INC0000018');
incidentGR.urgency = 2;
incidentGR.update();
```

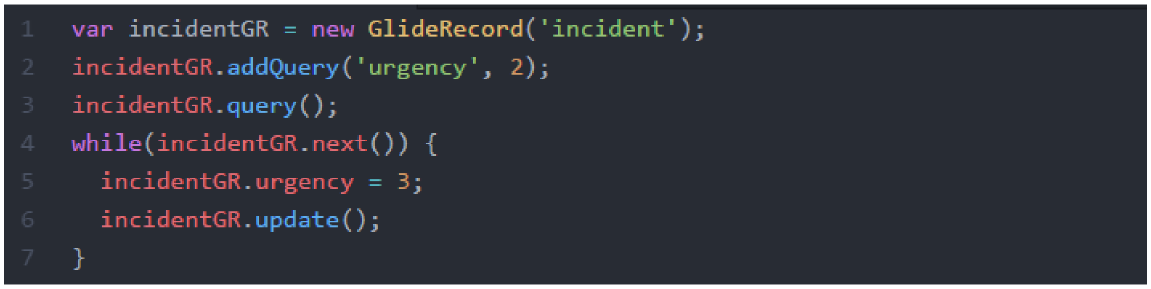
Result: Urgency value has been changed

Example 2:



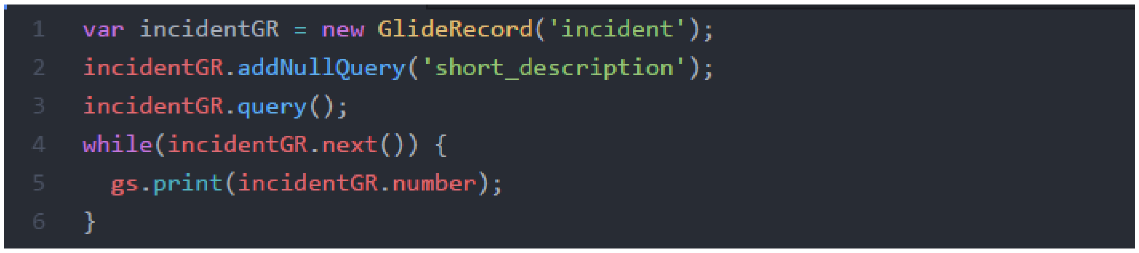
```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('urgency', 2);
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number)
}
```

Example 3:



```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('urgency', 2);
incidentGR.query();
while(incidentGR.next()) {
  incidentGR.urgency = 3;
  incidentGR.update();
}
```

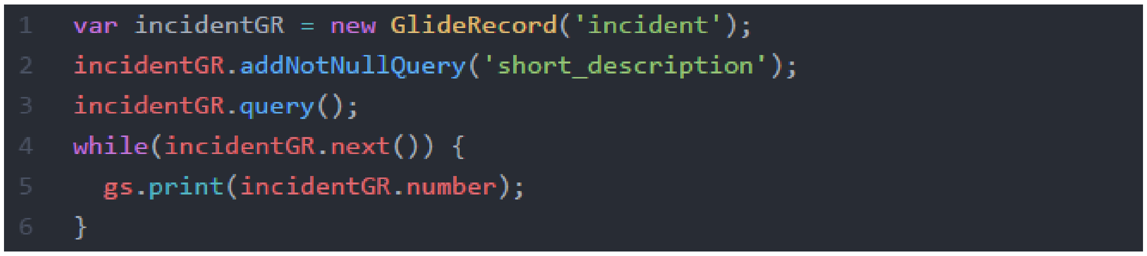
Working with addNullQuery () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.addNullQuery('short_description');
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number);
}
```

Result: Return all records where the description field value is null

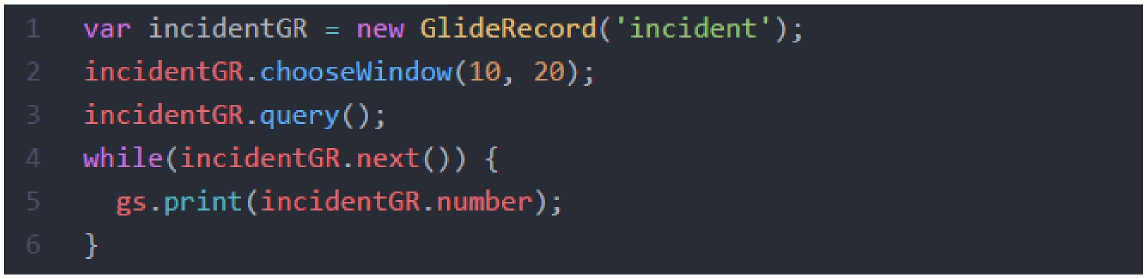
Working with addNotNullQuery () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.addNotNullQuery('short_description');
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number);
}
```

Result: Return all records where the description field contain the value

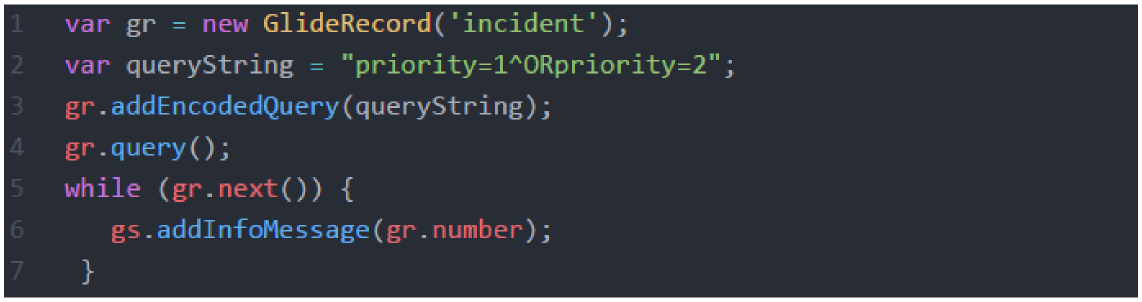
Working with chooseWindow () method



```
var incidentGR = new GlideRecord('incident');
incidentGR.chooseWindow(10, 20);
incidentGR.query();
while(incidentGR.next()) {
  gs.print(incidentGR.number);
}
```

Result: Choose Window will return all records between the first parameter(inclusive) and the second parameter(exclusive), so this example will return the 10 incidents between record 10-19 both inclusive. Works with orderBy

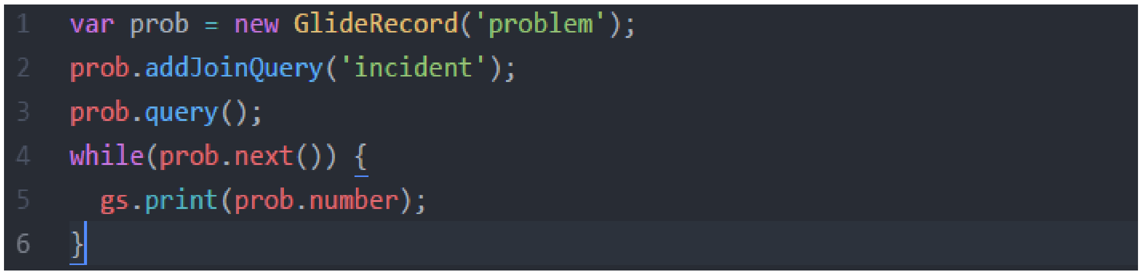
Working with addInfoMessage () method



```javascript
var gr = new GlideRecord('incident');
var queryString = "priority=1^ORpriority=2";
gr.addEncodedQuery(queryString);
gr.query();
while (gr.next()) {
   gs.addInfoMessage(gr.number);
 }
```

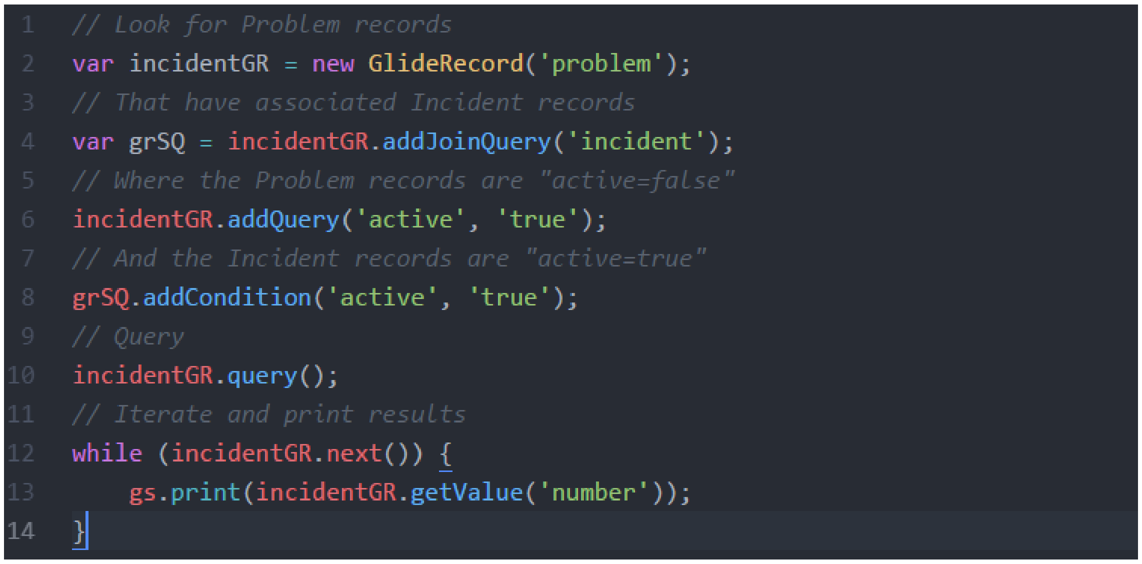
Working with addJoinQuery() method

Example 1:



```javascript
var prob = new GlideRecord('problem');
prob.addJoinQuery('incident');
prob.query();
while(prob.next()) {
  gs.print(prob.number);
}
```

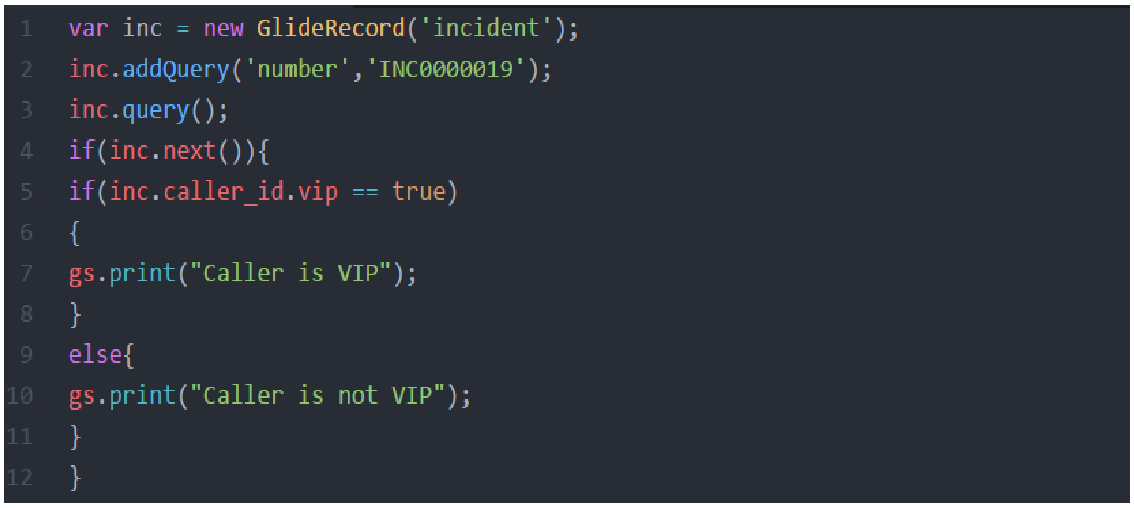
Example 2:



```javascript
// Look for Problem records
var incidentGR = new GlideRecord('problem');
// That have associated Incident records
var grSQ = incidentGR.addJoinQuery('incident');
// Where the Problem records are "active=false"
incidentGR.addQuery('active', 'true');
// And the Incident records are "active=true"
grSQ.addCondition('active', 'true');
// Query
incidentGR.query();
// Iterate and print results
while (incidentGR.next()) {
    gs.print(incidentGR.getValue('number'));
}
```

Result: Find inactive problems with associated incidents

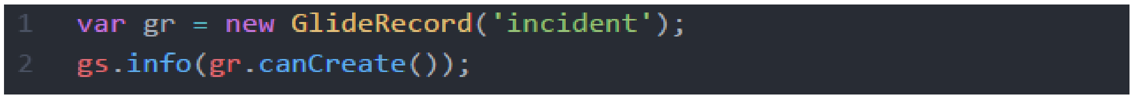
Just byou want to know caller is VIP or not methods



```
var inc = new GlideRecord('incident');
inc.addQuery('number','INC0000019');
inc.query();
if(inc.next()){
if(inc.caller_id.vip == true)
{
gs.print("Caller is VIP");
}
else{
gs.print("Caller is not VIP");
}
}
```

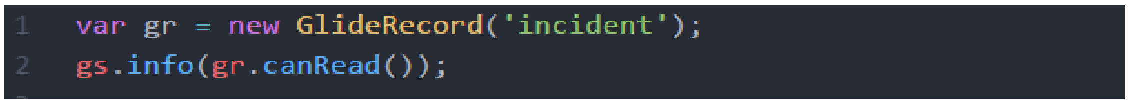
Secure canCreate , canCreate, canUpade, canDelete () methods

Example 1:



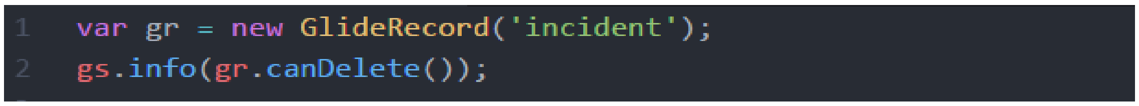
```
var gr = new GlideRecord('incident');
gs.info(gr.canCreate());
```

Example 2:



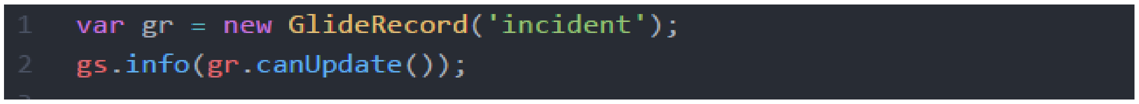
```
var gr = new GlideRecord('incident');
gs.info(gr.canRead());
```

Example 3:

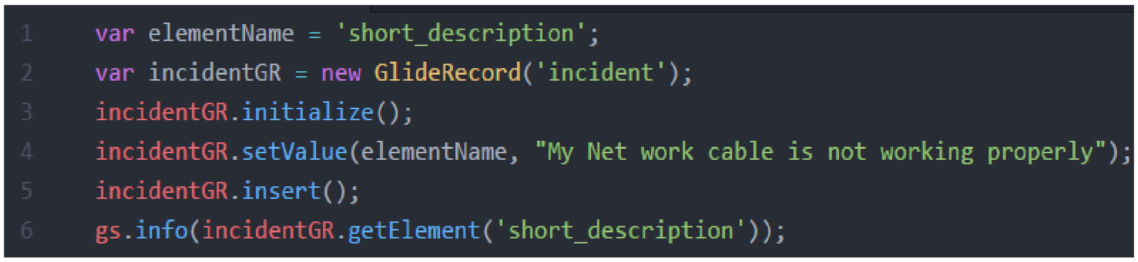


```
var gr = new GlideRecord('incident');
gs.info(gr.canDelete());
```

Example 3:

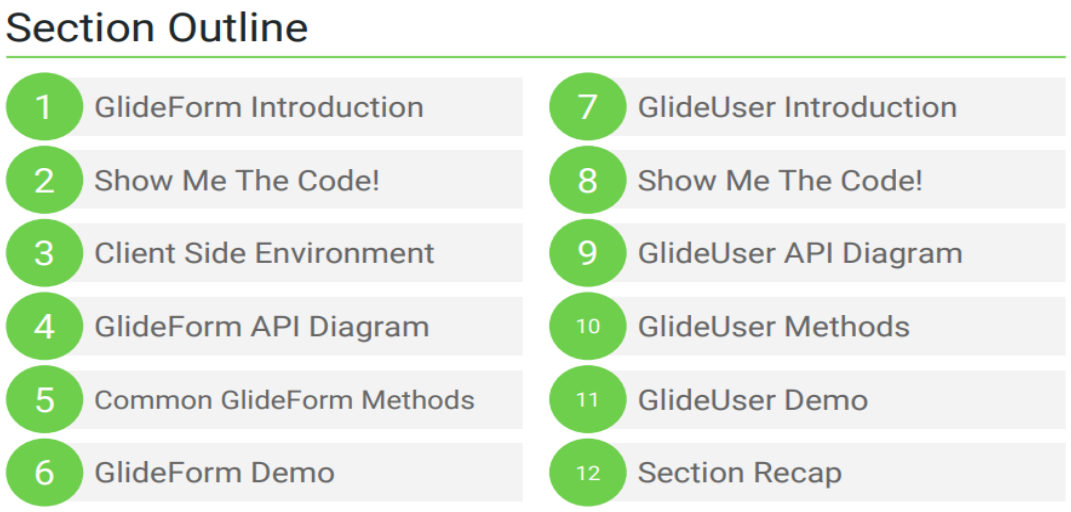


```
var gr = new GlideRecord('incident');
gs.info(gr.canUpdate());
```



```
var elementName = 'short_description';
var incidentGR = new GlideRecord('incident');
incidentGR.initialize();
incidentGR.setValue(elementName, "My Net work cable is not working properly");
incidentGR.insert();
gs.info(incidentGR.getElement('short_description'));
```

Glide Form

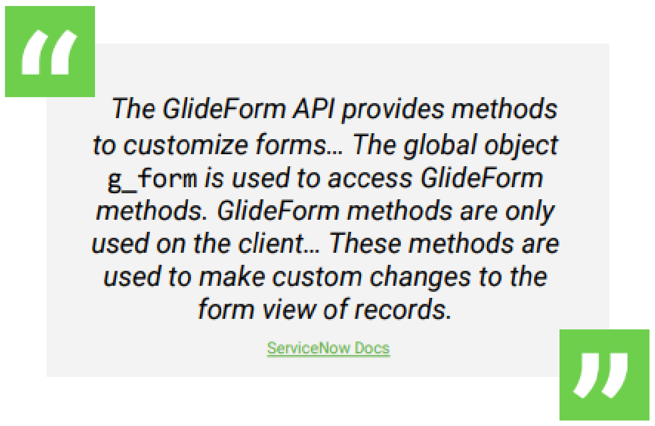


Section Outline

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

Glide Form Introduction

- Run from 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

The Glide Form API provides methods to customize forms. **GlideForm.js** is the JavaScript class containing the methods. The global object **g_form** is used to access Glide Form methods. Glide Form methods are only used on the client.

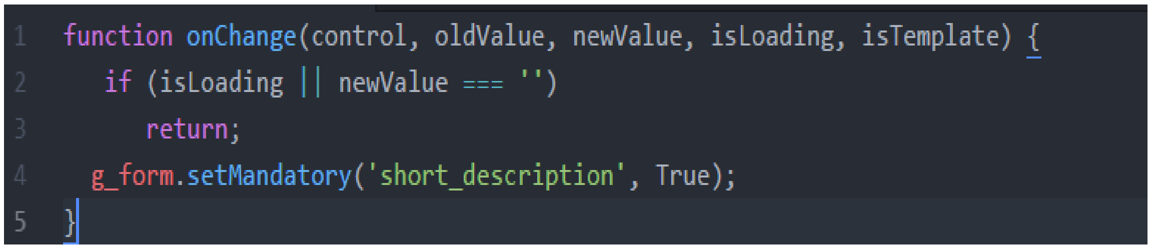
These methods are used to make custom changes to the form view of records. All validation of examples was done using Client Scripts.

Some of these methods can also be used in other client scripts (such as Catalog Client Scripts or Wizard Client Scripts), but must first be tested to determine whether they will work as expected.

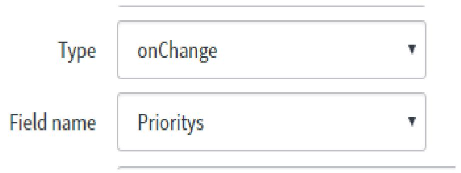
Note: The methods **getControl()**, **getHelpTextControl()**, **getElement()**, and **getFormElement()** are deprecated for mobile devices. For information on using GlideForm for mobile, see Mobile Client GlideForm (g_form) Scripting and Migration .

Show Me The Code!

- ✓ Set short description to mandatory when priority changes.



```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
   if (isLoading || newValue === '')
      return;
  g_form.setMandatory('short_description', True);
}
```



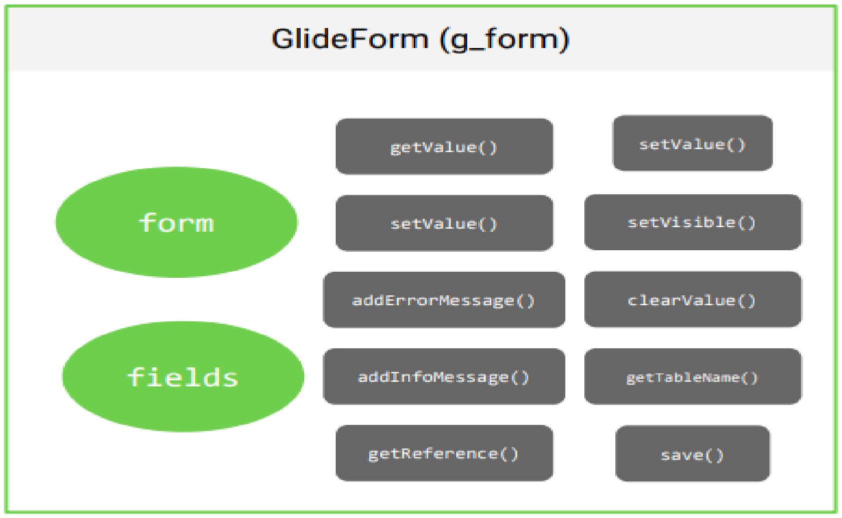
Type	onChange
Field name	Prioritys

Client-Side Environment

- 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 Service Now w/ client scripts and UI actions, use g_form instead

Glide Form API Overview

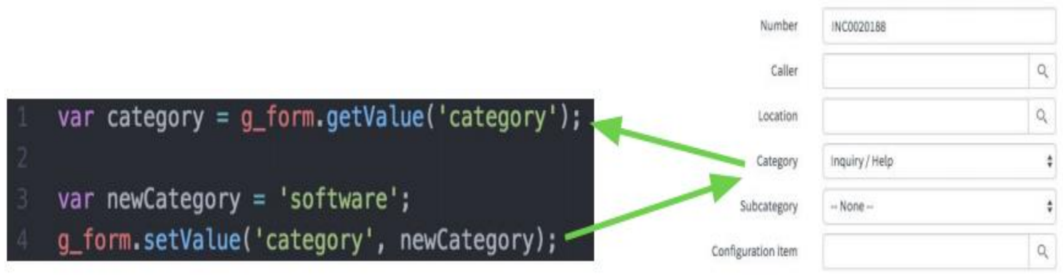


Common Glide Form Methods

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

Working with 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

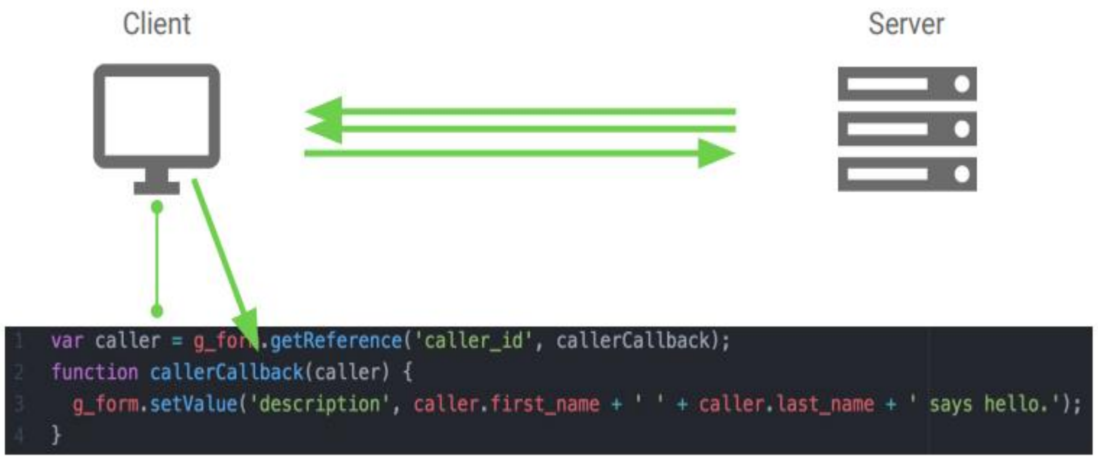


```
var category = g_form.getValue('category');

var newCategory = 'software';
g_form.setValue('category', newCategory);
```

Number	INC0020188
Caller	
Location	
Category	Inquiry / Help
Subcategory	-- None --
Configuration item	

Working with getReference() Method

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



```
var caller = g_form.getReference('caller_id', callerCallback);
function callerCallback(caller) {
  g_form.setValue('description', caller.first_name + ' ' + caller.last_name + ' says hello.');
}
```

Working with getValue () Method

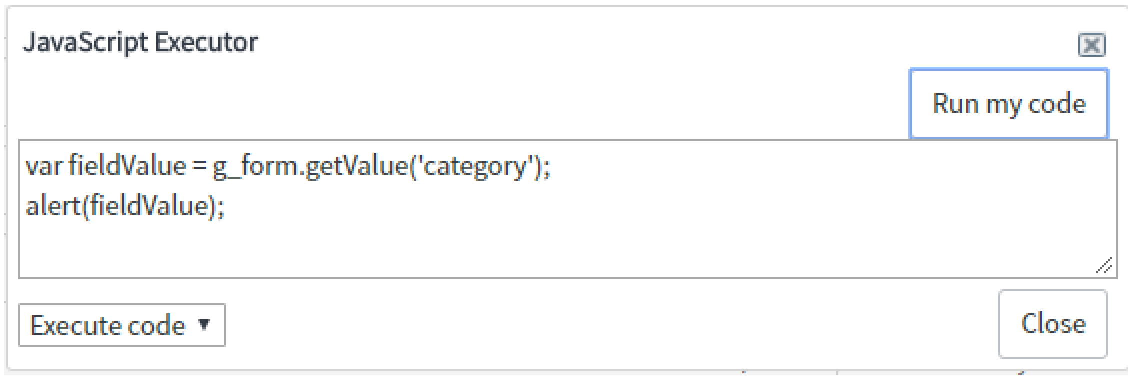
Step 1: Ctr+Shift+j

Step 2: Open ATOM

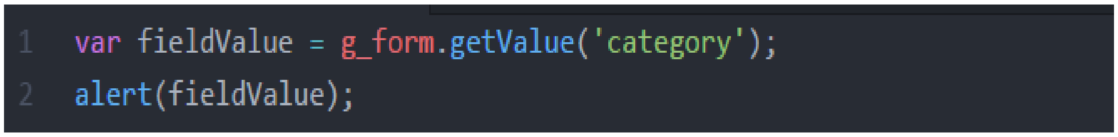
Write code **var fieldValue = g_form.getValue('category');**

alert(fieldValue);

Step 3: Copy code and paste into JavaScript Excutor



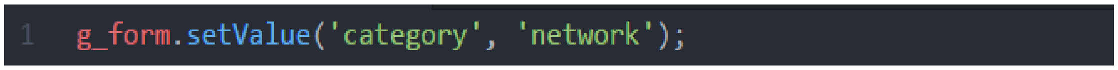
```
var fieldValue = g_form.getValue('category');
alert(fieldValue);
```



```
var fieldValue = g_form.getValue('category');
alert(fieldValue);
```

Result: Get Current category field value on the form

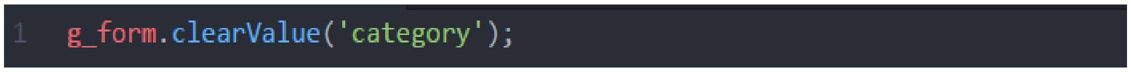
Working with setValue () Method



```
g_form.setValue('category', 'network');
```

Result: Set Current category field value on the form

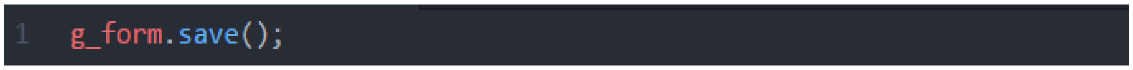
Working with clearValue () Method



```
g_form.clearValue('category');
```

Result: Clear the value in category field

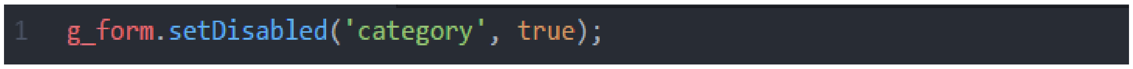
Working with save() Method



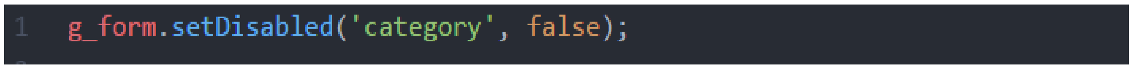
```
g_form.save();
```

Result: Saves current form

Working with setDisabled() Method



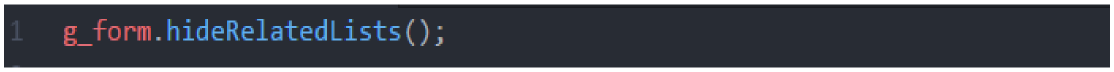
```
g_form.setDisabled('category', true);
```



```
g_form.setDisabled('category', false);
```

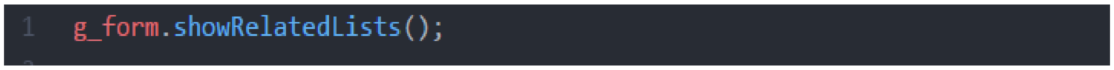
Result: Disabled form fields

Working with hideRelateLists() and shoeRelatedLists() Method



```
g_form.hideRelatedLists();
```

Result: Hide related lists on form

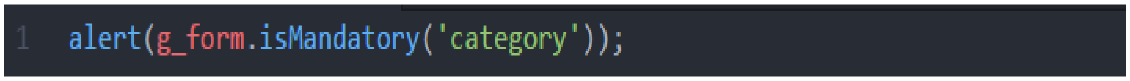


```
g_form.showRelatedLists();
```

Result: Show related lists on form

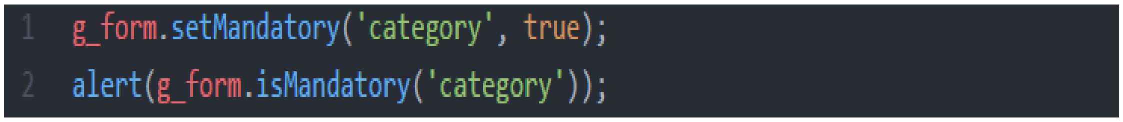
Working with isMandatory() Method

Example 1:



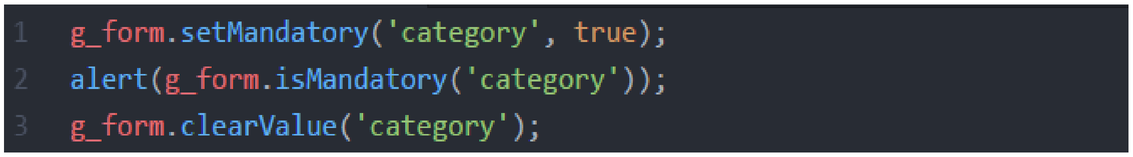
```
alert(g_form.isMandatory('category'));
```

Example 1:



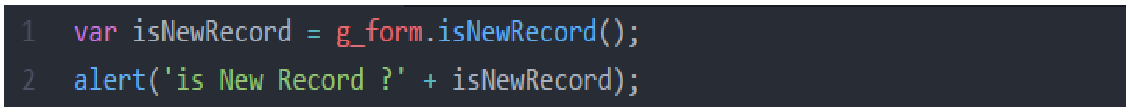
```
g_form.setMandatory('category', true);
alert(g_form.isMandatory('category'));
```

Example 3:



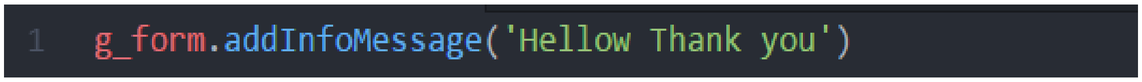
```
g_form.setMandatory('category', true);
alert(g_form.isMandatory('category'));
g_form.clearValue('category');
```

Working with isMandatory() Method

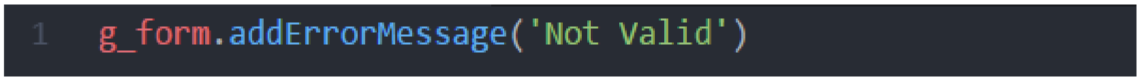


```
var isNewRecord = g_form.isNewRecord();
alert('is New Record ?' + isNewRecord);
```

Working with addInfoMessage() and addInfoMessage() Method

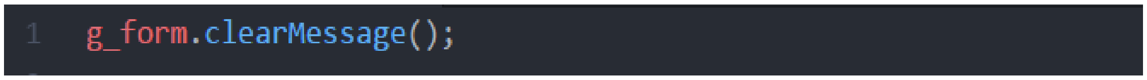


```
g_form.addInfoMessage('Hellow Thank you')
```



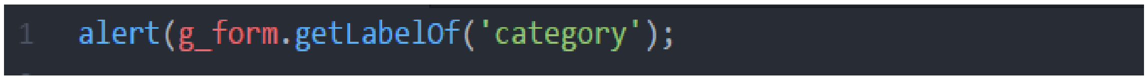
```
g_form.addErrorMessage('Not Valid')
```

Working with clearMessage () Method



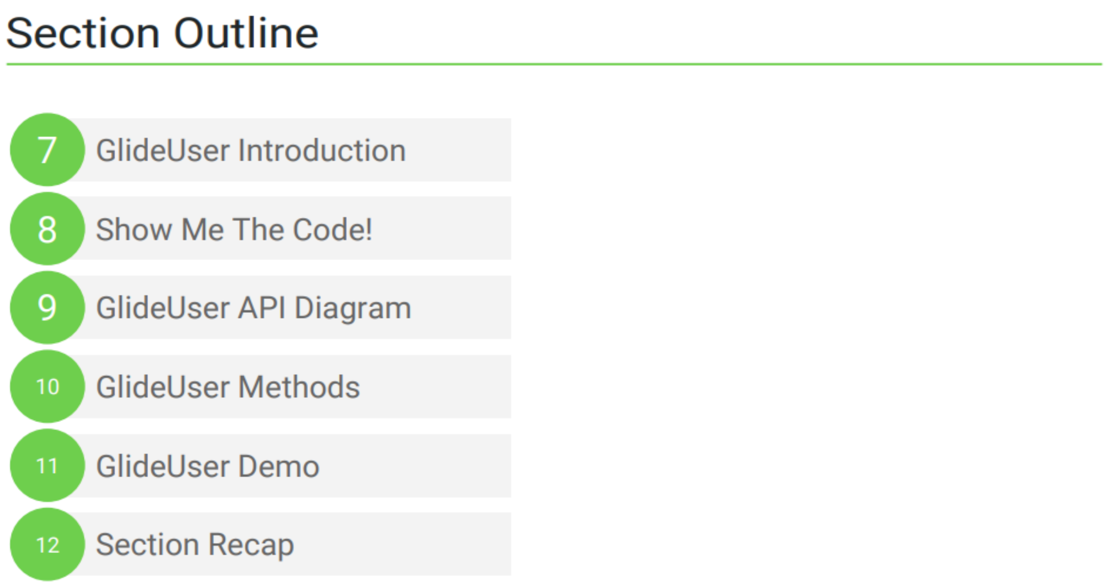
```
g_form.clearMessage();
```

Working with getLabelOf () Method



```
alert(g_form.getLabelOf('category'));
```

Glide User

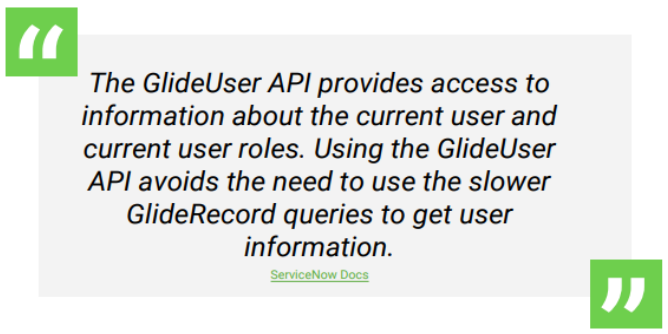


Section Outline

7 GlideUser Introduction
8 Show Me The Code!
9 GlideUser API Diagram
10 GlideUser Methods
11 GlideUser Demo
12 Section Recap

Glide User 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.

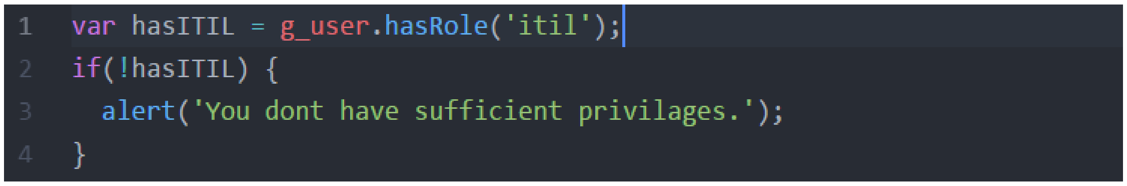
ServiceNow Docs

Glide User

- Contains name and role information about the current user.
- It is typically used in client scripts and UI policies but is also found in UI actions that run
- Cannot be used in business rules or UI actions that run on the server.
- Avoids the need for Glide Record queries to get user information.
- Session information about the current user
- And current user roles are contained in the client (web browser).
- All Glide User methods except **getClientData()** access the session information that is available by default.
- The **getClientData()** method requires setup on the server and use of **putClientData()** to make session information available.

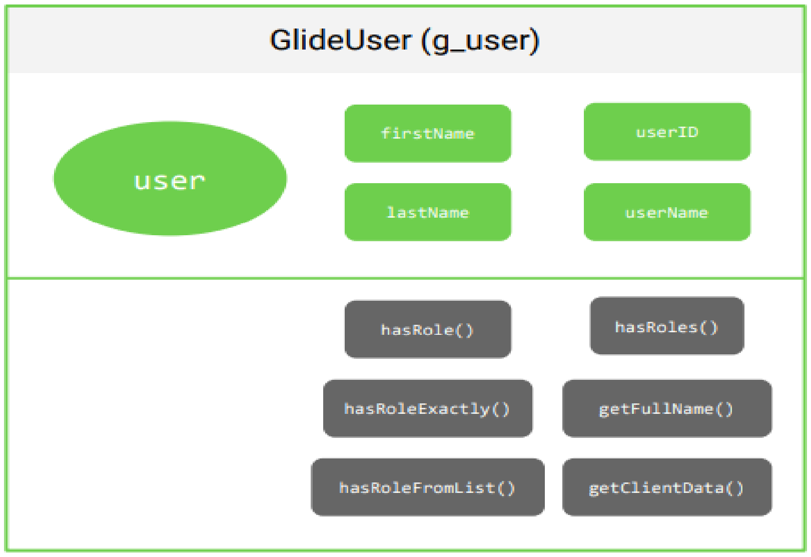
Show Me The Code!

- Check if the current user has the ITIL role



```
var hasITIL = g_user.hasRole('itil');
if(!hasITIL) {
  alert('You dont have sufficient privilages.');
}
```

Glide User API Overview

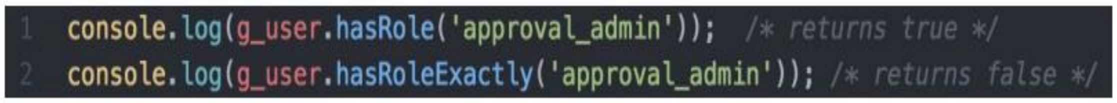


Glide User Methods & Properties

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

Glide User 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



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

Glide User Properties

- Object properties
- Don’t need getter methods