servicenow

Lab Guide

Advanced Service Portal

Goal

In this exercise, we will create a list widget and then we will modify it to load data client side.

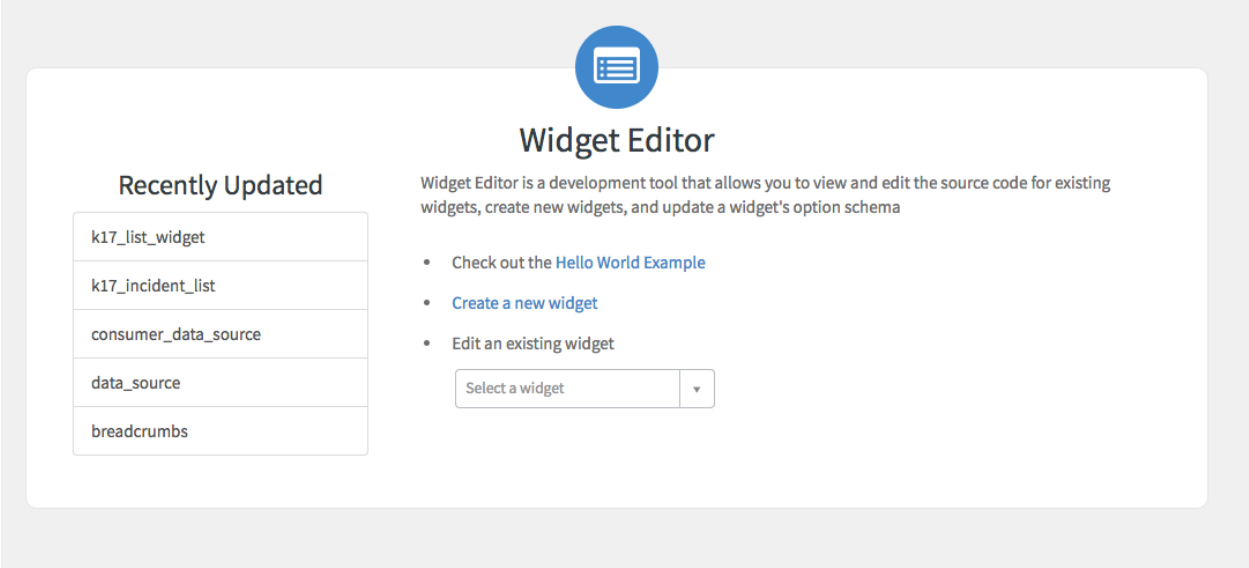
Exercise 1
Create list widget

Log on to Your Training Instance

1. Navigate to the unique instance URL provided to you.
2. Log on with provided credentials.

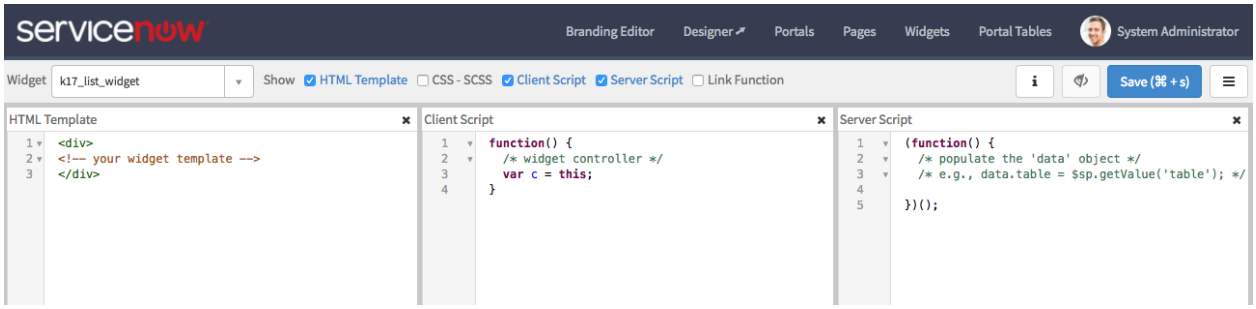
Create List Widget

1. Navigate to Service Portal > Service Portal Configuration.
2. Select **Widget Editor**
3. Select **Create a new widget**



4. Enter the following information:
   - Widget Name: dd18_list_widget
   - Widget ID: dd18_list_widget
   - Create Test Page: checked
   - Page Id: dd18_list_widget
5. Click the **Submit** button.

6. Your Screen should look like this



Enter Widget Information

1. Enter the following information in the Server Script Window:

```
(function () {
    /* populate the 'data' object */
    /* e.g., data.table = $sp.getValue('table'); */

    //define the variable to hold our list
    data.list = [];

    //Query the table
    var gr = new GlideRecord('incident');
    gr.addQuery('active', true);
    gr.query()
    while (gr.next()) {
        //load values into object
        var record_obj = {};
        record_obj.number = gr.number.getDisplayValue();
        record_obj.short_description = gr.short_description.getDisplayValue();
        record_obj.priority = gr.priority.getDisplayValue();
        record_obj.category = gr.category.getDisplayValue();
        //add object to the list
        data.list.push(record_obj);
    }
})();
```

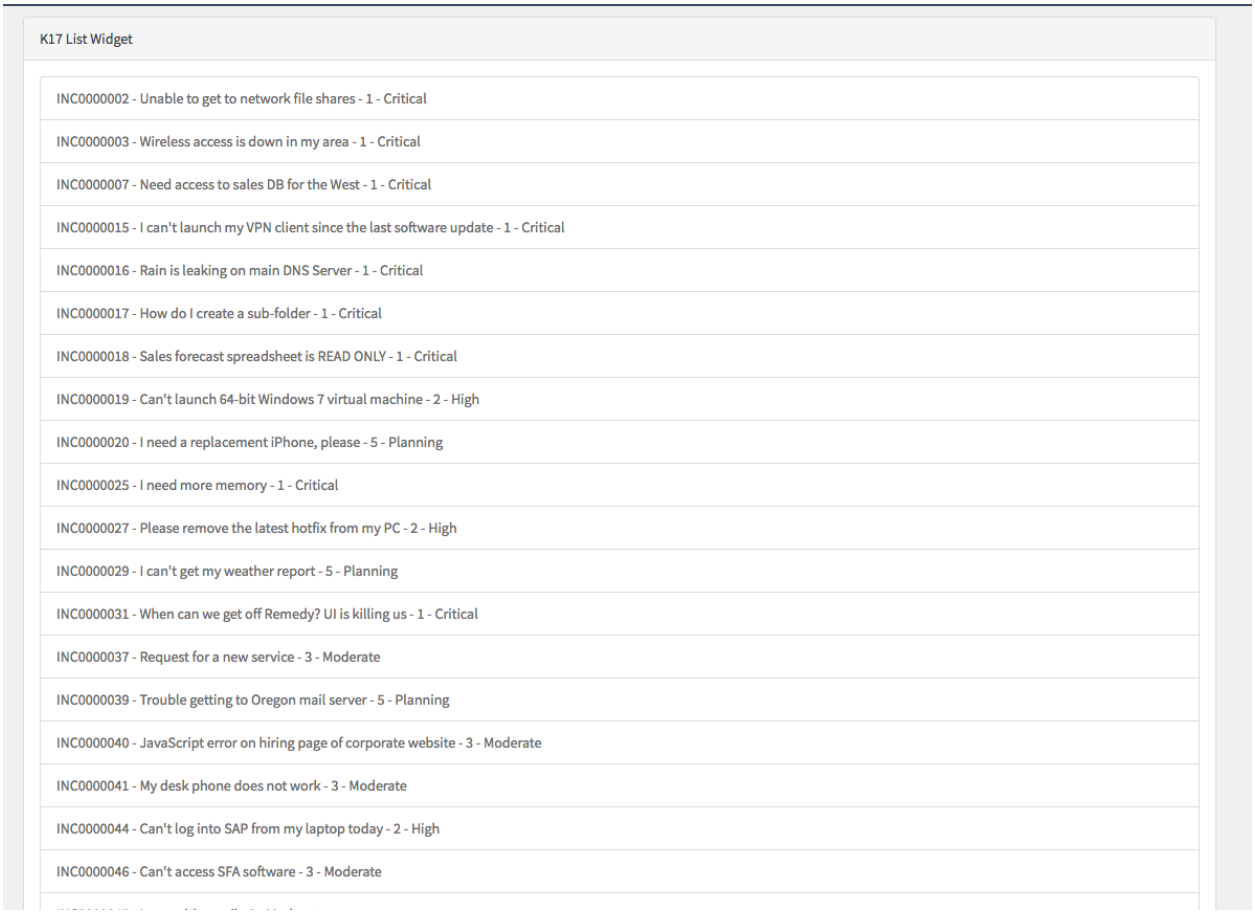
2. Enter the following information in the HTML template box

```
<div class="panel panel-default">
    <!-- your widget template -->
    <div class="panel-heading">
        DD18 List Widget
    </div>
    <div class="panel-body">
```

```
        <ul class="list-group">
            <li class="list-group-item" ng-repeat="item in
c.data.list">{{item.number}} - {{item.short_description}} -
{{item.priority}}</li>
        </ul>
    </div>
</div>
```

3. Click **Save**

4. Navigate to [your instance] .service-now.com/sp_config/?id=dd18_list_widget and check your results



Load data client side

1. Navigate back to the widget editor

2. Modify the Server Script to look like this

```
(function() {
    /* populate the 'data' object */
    /* e.g., data.table = $sp.getValue('table'); */

    if(input){
        //define the variable to hold our list
        data.list = [];
        //Query the table
        var gr = new GlideRecord('incident');
        gr.addQuery('active', true);
        gr.query()
        while (gr.next()) {
            //load values into object
            var record_obj = {};
            record_obj.number = gr.number.getDisplayValue();
            record_obj.short_description =
gr.short_description.getDisplayValue();
            record_obj.priority = gr.priority.getDisplayValue();
            record_obj.category = gr.category.getDisplayValue();
            //add object to the list
            data.list.push(record_obj);
        }
    }
})();
```

3. Enter the following information in the Client Script

```
function($scope) {
    /*widget controller */
    var c = this;
    c.data.loading = true;
    $scope.server.update().then(function(){
        c.data.loading = false;
    })
}
```

4. Modify the html template to look like this

```
<div class="panel panel-default">
<!-- your widget template -->
    <div class="panel-heading">
    DD18 List Widget
    </div>
    <div class="panel-body">
```

```
        <span ng-if="c.data.loading"><i class="fa fa-spinner fa-spin fa-3x fa-
fw"></i>
        <span class="sr-only">Loading\...</span>
        </span>
        <ul class="list-group">
            <li class="list-group-item" ng-repeat="item in
c.data.list">{{item.number}} - {{item.short_description}} -{{item.priority}}</li>
        </ul>
    </div>
</div>
```

5. Click **Save**

6. Navigate to [your instance] .service-now.com/sp_config/?id=dd18_list_widget and check your results

Goal

In this exercise, we will modify our list widget to make use of the REST API.

Exercise 2
REST API

Modify List Widget

1. Navigate to the widget editor and open the dd18_list_widget.

2. Modify the Server Script to look like this

```
(function() {
    /*populate the 'data' object */
    /*e.g., data.table = $sp.getValue('table'); */
})();
```

3. Modify the Client Script to look like this:

```
function($scope,$http) {
    /*widget controller */
    var c = this;
    c.data.loading = true;


$http.get('/api/now/table/incident?sysparm_query=active%3Dtrue').success(function
(response){
        c.data.loading=false;
        c.data.list = response.result;
    })
}
```

4. Click **Save**

5. Navigate to [your instance] .service-now.com/sp_config/?id=dd18_list_widget and check your results

Goal

Exercise 3
NG Template

In this exercise, we will create two ng-templates to be used by our widgets. We will alter the HTML Template to use one and then the other to see how that affects the rendering.

Create Templates

1. Navigate to Service Portal > Widgets
2. Open the dd18_list_widget.
3. Scroll to the bottom and select the Angular **ng-templates** tab
4. Click **New**
5. Enter the following information
   - ID: task-priority
   - Template:

```
<span>{{item.number}} - {{item.short_description}} - {{item.priority}}</span>
```

6. Click Submit
7. Scroll to the bottom and select the Angular **ng-templates** tab
8. Click **New**
9. Enter the following information
   - ID: task-category
   - Template:

```
<span>{{item.number}} - {{item.short_description}} - {{item.category}}</span>
```

10. Click **Submit**

Update List widget

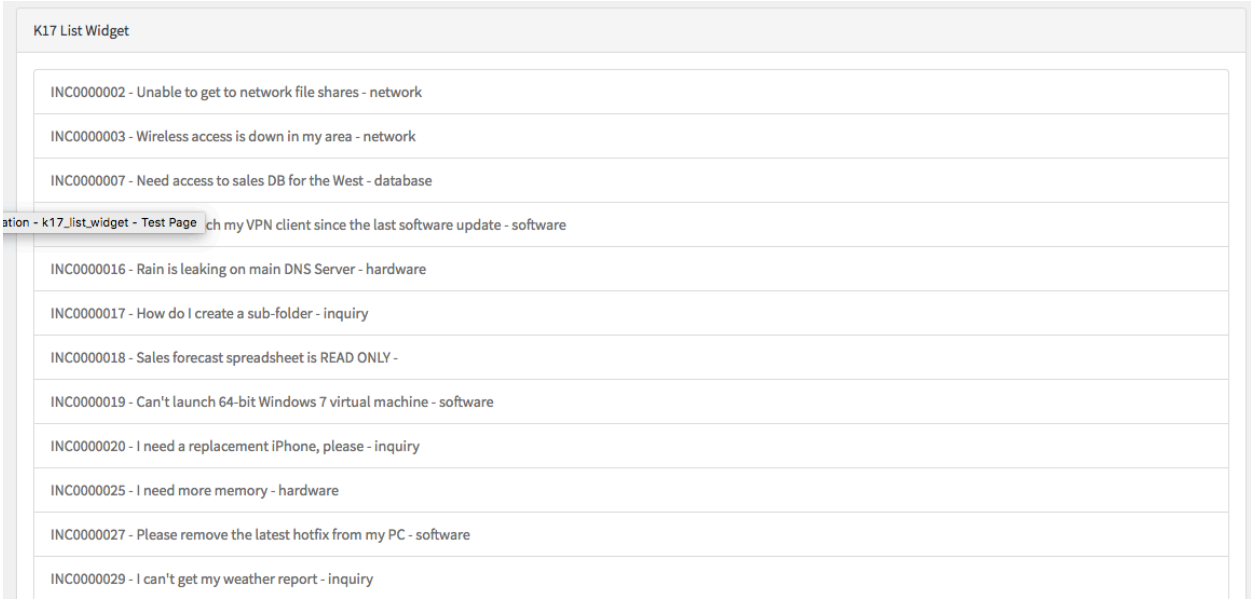
1. Open the dd18_list_widget using the widget editor

2. Modify the HTML template to look like this

```
<div class="panel panel-default">
    <!-- your widget template -->
    <div class="panel-heading">
        DD18 List Widget
    </div>
    <div class="panel-body">
    <span ng-if="c.data.loading"><i class="fa fa-spinner fa-spin fa-3x fa-
fw"></i>
        <span class="sr-only">
            Loading\...
        </span>
    </span>
    <ul class="list-group">
        <li class="list-group-item" ng-repeat="item in c.data.list" ng-
include="'task-category'"></li>
    </ul>
    </div>
</div>
```

3. Click **Save**

4. Navigate to [your instance] .service-now.com/sp_config/?id=dd18_list_widget and check your results



5. Return to your HTML Template and change it to the task-priority template and reload.

Goal

In this section, we will define options for our widget and use them to alter the rendering of a widget as it appears on a page.

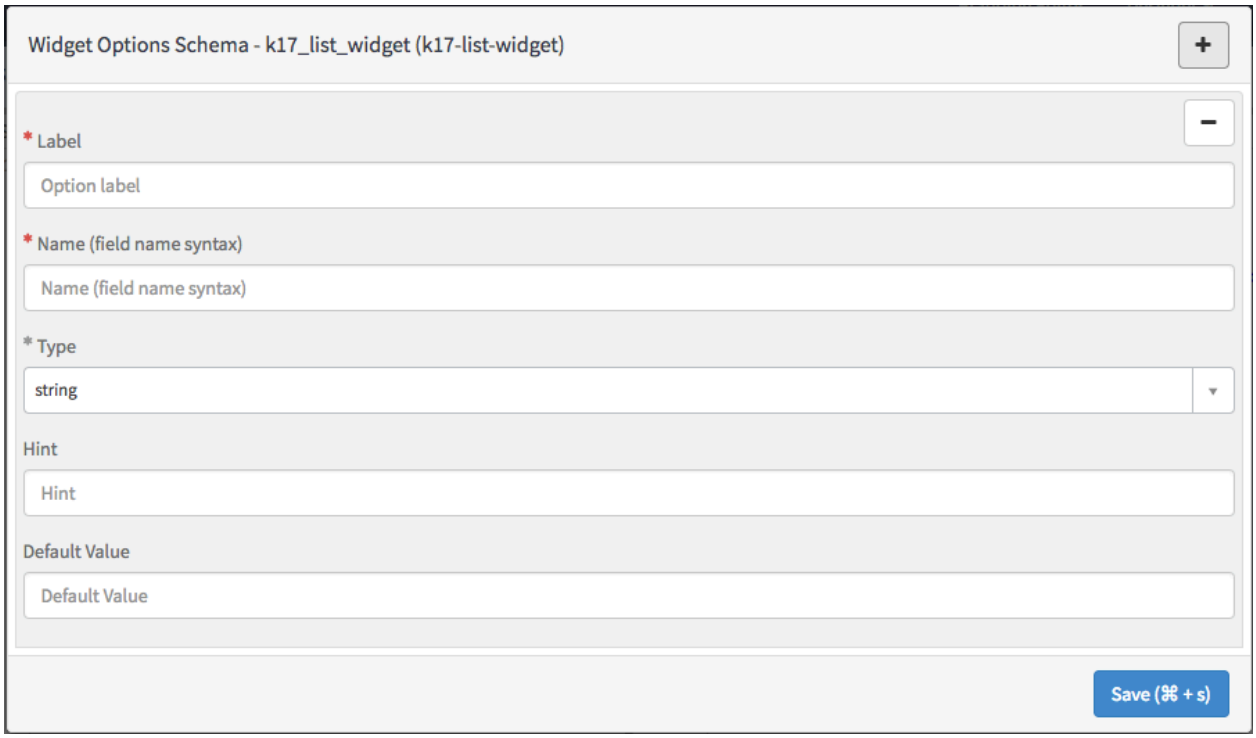
Exercise 4 Widget Options

Add out of box options

1. Navigate to Service Portal > Widgets
2. Open the dd18_list_widget.
3. Click on the lock icon next the **Fields** field
4. Add the following fields:
   - Title
   - Bootstrap Color
5. Click **Update**

Modify list widget

1. Open the dd18_list_widget using the widget editor. NOTE: If you had the editor open, refresh it before continuing.
2. Click on the  icon in the top right corner
3. Select Edit Option Schema
4. Click on the  icon in the top right corner on of the modal window



Widget Options

5. Enter the following information

   - Label: Table
   - Name: table
   - Type: String
   - Default Value: incident

6. Click on the  icon in the top right corner on of the modal window

7. Enter the following information

   - Label: Query
   - Name: query
   - Type: String
   - Default Value: leave empty

8. Click on the  icon in the top right corner on of the modal window

9. Enter the following information

   - Label: Template
   - Name: template
   - Type: String
   - Default Value: leave empty

10. Click **Save**

11. Modify Client Script to look like this

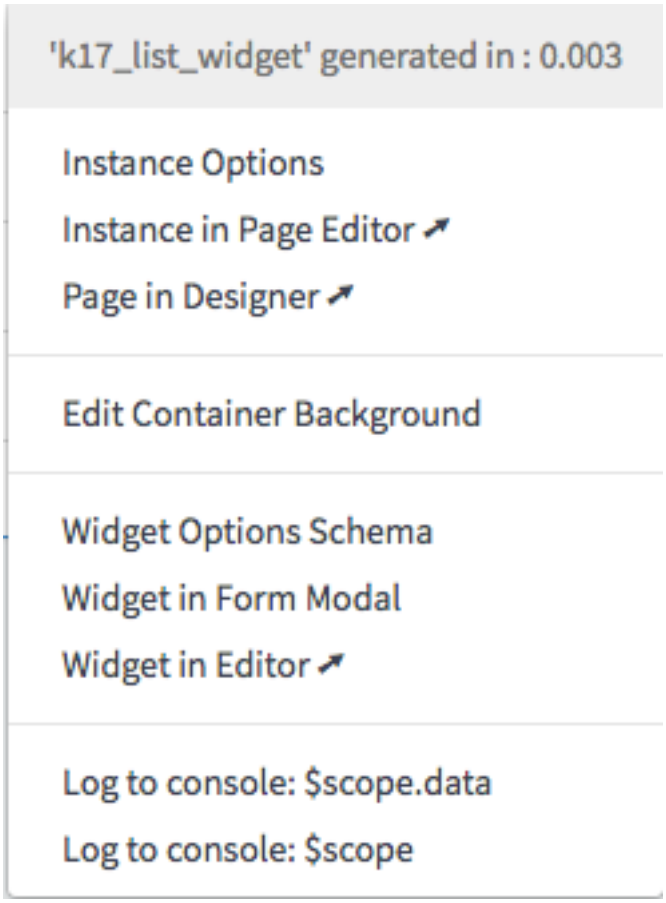
```
function($scope,$http) {
    /*widget controller */
    var c = this;
    c.data.loading = true;
    c.data.table = c.options.table || "incident";
    c.data.query = c.options.query || "";
    c.data.template = c.options.template || "task-category";


$http.get('/api/now/table/'+c.data.table+'?sysparm_query='+c.data.query).success(
function(response){
        c.data.loading=false;
        c.data.list = response.result;
    })
}
```

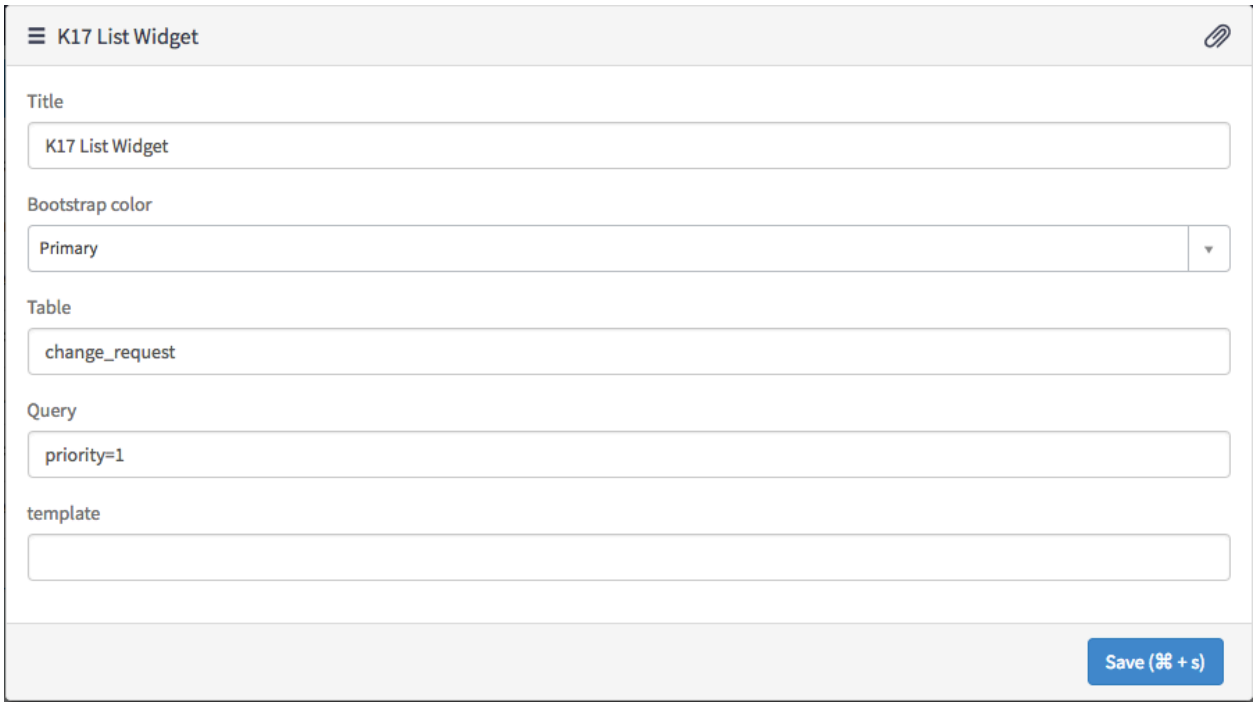
12. Modify HTML Template to look like this

```
<div class="panel panel-{{c.options.color}}">
    <!-- your widget template -->
    <div class="panel-heading">
        {{c.options.title}}
    </div>
    <div class="panel-body">
        <span ng-if="c.data.loading"><i class="fa fa-spinner fa-spin fa-3x fa-
fw"></i>
            <span class="sr-only">Loading\..</span>
        </span>
        <ul class="list-group">
            <li class="list-group-item" ng-repeat="item in c.data.list" ng-
include="c.data template"></li>
        </ul>
    </div>
</div>
```

13. Click **Save**
14. Navigate to [your instance] .service-now.com/sp_config/?id=dd18_list_widget
15. Press **“Control” right click** on top of your widget
16. Select Instance Options

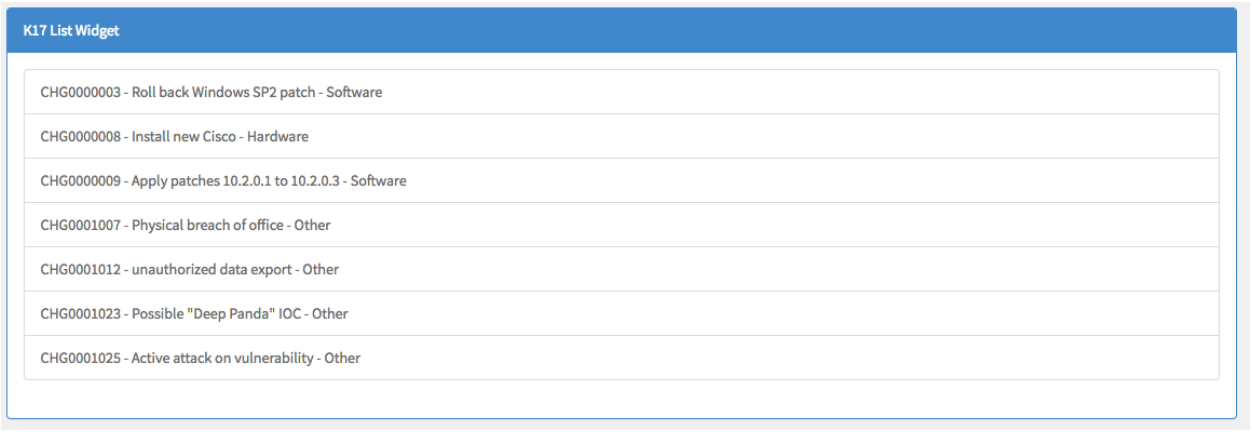


17. Enter the following information



18. Click **Save**

19. Check your results



Goal

In this exercise, we will create a record watcher so our widget is automatically updated without having to refresh the page.

Exercise 5 Record Watchers

Set up record watcher

1. Open the dd18_list_widget using the widget editor.
2. Modify the Client Script to look like this:

```
function($scope,$http,snRecordWatcher) {
    /*widget controller */
    var c = this;
    c.data.loading = true;
    c.data.table = c.options.table || "incident";
    c.data.query = c.options.query || "";
    c.data.template = c.options.template || "task-category";

    function getData(){

$http.get('/api/now/table/'+c.data.table+'?sysparm_query='+c.data.query).success(
function(response){
            c.data.loading=false;
            c.data.list = response.result;
        })
    }

    getData();

    snRecordWatcher.initList(c.data.table, c.data.query);
    $scope.$on('record.updated', function(name, data) {
        getData();
    });
}
```

3. Click **Save**
4. Navigate to [your instance] .service-now.com/sp_config/?id=dd18_list_widget
5. How Many Changes are displayed?
6. Open a new window and open a new critical priority change request.

7. Go back to your widget window, is the new change request now showing?

Goal

In this exercise, we will modify our widget to broadcast an event and have another widget receive that event.

Set up event

1. Open the dd18_list_widget using the widget editor.
2. Modify the Client Script to look like this:

Exercise 6 Broadcast events

```
function($scope,$http,snRecordWatcher,$rootScope) {
    /*widget controller */
    var c = this;
    c.data.loading = true;
    c.data.table = c.options.table || "incident";
    c.data.query = c.options.query || "";
    c.data.template = c.options.template || "task-category";

    function getData(){

$http.get('/api/now/table/'+c.data.table+'?sysparm_query='+c.data.query).success(
function(response){
            c.data.loading=false;
            c.data.list = response.result;
            $rootScope.$broadcast("DD18ListWidgetUpdated",c.data.list);
        })
    }

    getData();
    snRecordWatcher.initList(c.data.table, c.data.query);

    $scope.$on('record.updated', function(name, data) {
        getData();
    });
}
```

3. Click **Save**
4. Click 

5. Select Create New Widget

6. Enter the following information:

   - Widget Name: dd18_count_display_widget
   - Widget ID: dd18_count_display_widget

7. Click Submit

8. Enter the following code in the HTML Template Field:

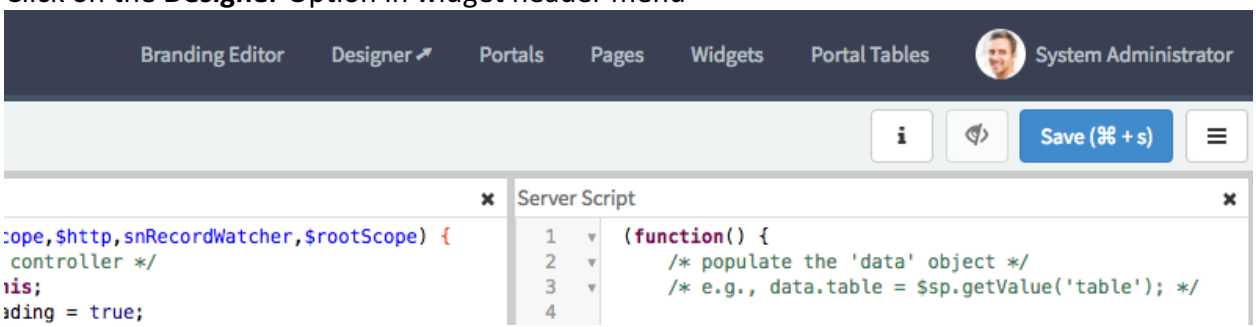
```
<div>
    <!-- your widget template -->
    <h2 class="jumbotron">
    Total Records:{{c.data.count}}
    </h2>
</div>
```

9. Enter the following code in the Client Script field:

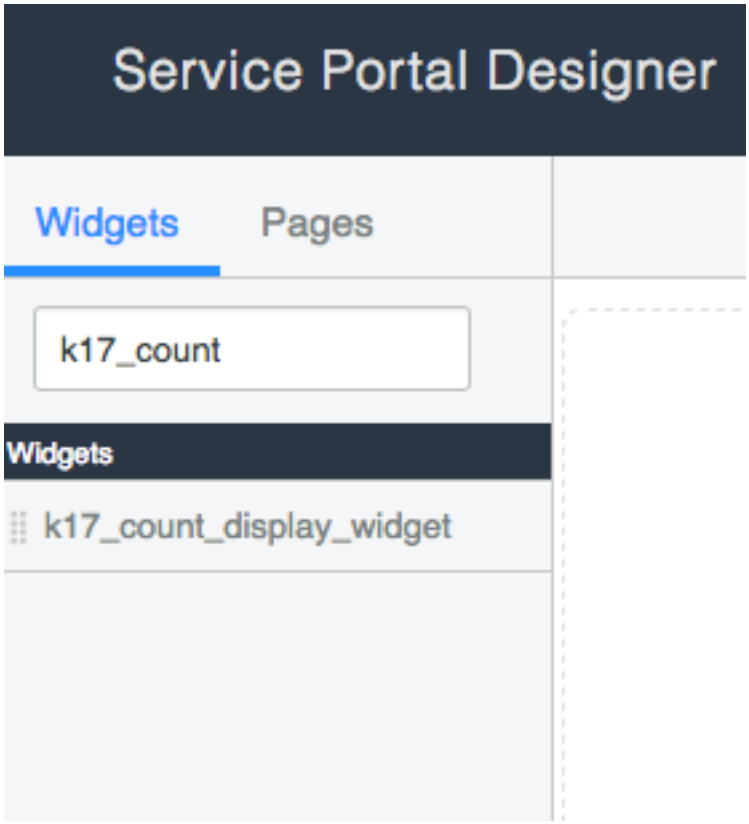
```
function($scope) {
    /*widget controller */
    var c = this;
    c.data.count = 0;
    $scope.$on("DD18ListWidgetUpdated",function(evt,results){
        console.log(results)
        c.data.count = results.length;
    })
}
```

10. Click **Save**

11. Click on the **Designer** Option in widget header menu



12. Use the filter to look for dd18_list_widget Test Page
13. Select the Page
14. Use the filter to search for dd18_count_display_widget



15. Drag and drop dd18_count_display_widget above the dd18 List Widget
16. Navigate to [your instance] .service-now.com/sp_config/?id=dd18_list_widget
17. Is the count of records populating?
18. Create another new Change Request record. Does the change reflect in both your list and the count widget? If not, try to troubleshoot the issue.