

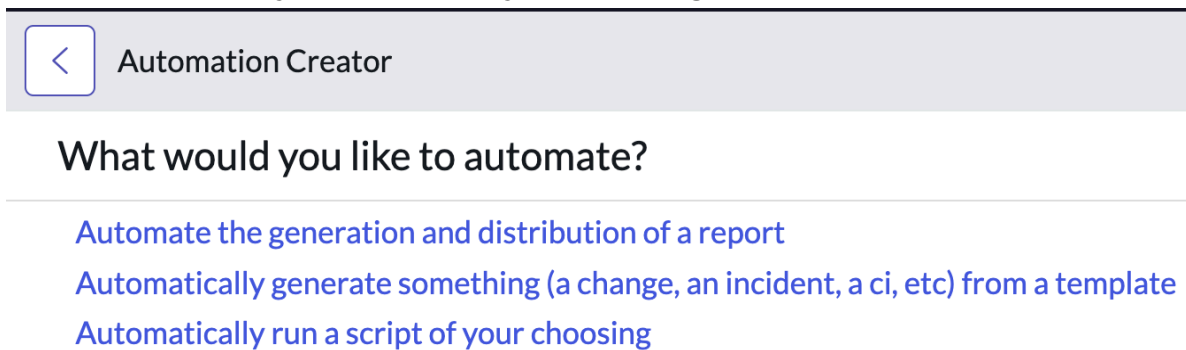
# Moveworks Creator Studio - Events: Agent Nudge

## Requirements:

- Install the Moveworks API SDK
- Have access to Moveworks Creator Studio

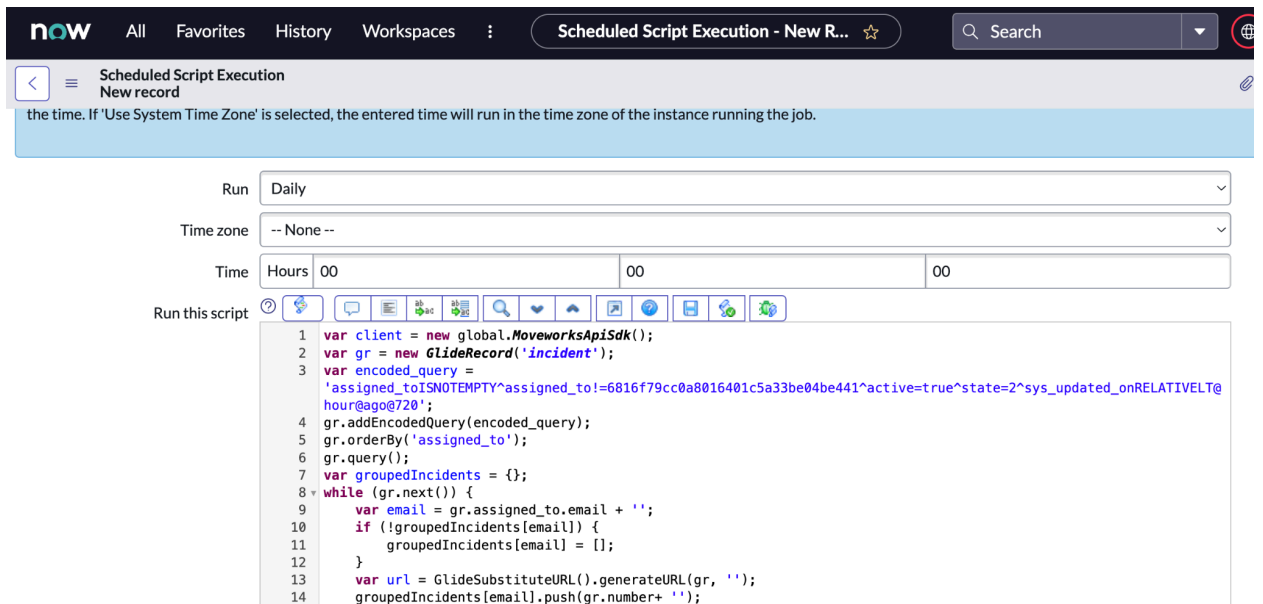
## Set up Steps:

1. Navigate to Scheduled jobs and select **New**
2. Select: **Automatically run a script of your choosing**



The screenshot shows the 'Automation Creator' interface. At the top, there is a navigation bar with a back arrow and the text 'Automation Creator'. Below this, a large heading asks 'What would you like to automate?'. Three blue links are provided: 'Automate the generation and distribution of a report', 'Automatically generate something (a change, an incident, a ci, etc) from a template', and 'Automatically run a script of your choosing'.

3. Once you have verified your logic, paste the script



The screenshot shows the 'Scheduled Script Execution - New R...' configuration page. The top navigation bar includes 'now', 'All', 'Favorites', 'History', 'Workspaces', and a search bar. The main content area shows the configuration for a new record. The 'Run' dropdown is set to 'Daily'. The 'Time zone' dropdown is set to '-- None --'. The 'Time' section shows 'Hours' set to '00' and '00'. Below the configuration fields is a 'Run this script' section with a toolbar and a code editor. The code editor contains the following JavaScript code:

```
1 var client = new global.MoveworksApiSdk();
2 var gr = new GlideRecord('incident');
3 var encoded_query =
  'assigned_toISNOTEMPTY^assigned_to!=6816f79cc0a8016401c5a33be04be441^active=true^state=2^sys_updated_onRELATIVE@
  hour@ago@720';
4 gr.addEncodedQuery(encoded_query);
5 gr.orderBy('assigned_to');
6 gr.query();
7 var groupedIncidents = {};
8 while (gr.next()) {
9   var email = gr.assigned_to.email + '';
10  if (!groupedIncidents[email]) {
11    groupedIncidents[email] = [];
12  }
13  var url = GlideSubstituteURL().generateURL(gr, '');
14  groupedIncidents[email].push(gr.number+ '');
```

### Example Script:

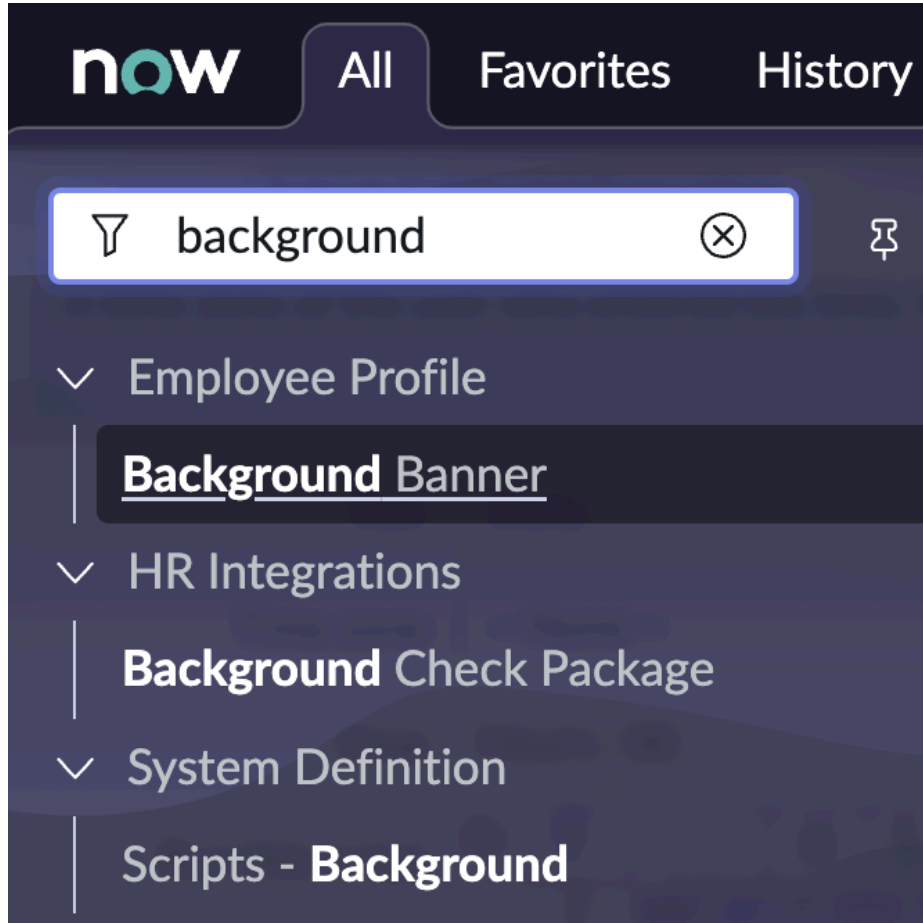
```
var client = new global.MoveworksApiSdk();
var gr = new GlideRecord('incident');
var encoded_query =
'assigned_toISNOTEMPTY^assigned_to!=6816f79cc0a8016401c5a33be04be441^
active=true^state=2^sys_updated_onRELATIVELT@hour@ago@720';
gr.addEncodedQuery(encoded_query);
gr.orderBy('assigned_to');
gr.query();
var groupedIncidents = {};
while (gr.next()) {
    var email = gr.assigned_to.email + '';
    if (!groupedIncidents[email]) {
        groupedIncidents[email] = [];
    }
    var url = GlideSubstituteURL().generateURL(gr, '');
    groupedIncidents[email].push(gr.number+ '');
}

for (var email in groupedIncidents) {
    gs.log('Prepared the following message to send to: '+ email)
    var message = "Please respond to the following tickets that are
overdue and pending your action.:";
    message += groupedIncidents[email].join('<br>');
    gs.log(message);
    // client.send_message(recipients, msg, event_id);
    // handle failures appropriately
}
}
```

### Testing instructions:

When testing, make sure you use a test API key scoped to only message a subset of test users.

You can test the script using Background Scripts, which allow you to run server side code in ServiceNow.



For example, running the example script above, shows the example messages that would be sent out:

```
now All Favorites History : Scheduled Script E... ☆ Search [Globe] [Help] [Profile]
[0:00:00.124] Script completed in scope global: script
Script execution history and recovery available here
*** Script: Prepared the following message to send to: beth.anglin@example.com
*** Script: Please respond to the following tickets that are overdue and pending your action.: INC0000047<br>INC0000055<br>INC0000050
*** Script: Prepared the following message to send to: bud.richman@example.com
*** Script: Please respond to the following tickets that are overdue and pending your action.: INC0000019
*** Script: Prepared the following message to send to: david.loo@example.com
*** Script: Please respond to the following tickets that are overdue and pending your action.: INC0000031
*** Script: Prepared the following message to send to: don.goodliffe@example.com
*** Script: Please respond to the following tickets that are overdue and pending your action.: INC0000029<br>INC0000049<br>INC0000051
*** Script: Prepared the following message to send to: fred.luddy@example.com
*** Script: Please respond to the following tickets that are overdue and pending your action.: INC0013174<br>INC0000052<br>INC0000044
*** Script: Prepared the following message to send to: howard.johnson@example.com
*** Script: Please respond to the following tickets that are overdue and pending your action.: INC0000037
*** Script: Prepared the following message to send to: itil@example.com
*** Script: Please respond to the following tickets that are overdue and pending your action.: INC0000018<br>INC0000020<br>INC0000041
*** Script: Prepared the following message to send to:
*** Script: Please respond to the following tickets that are overdue and pending your action.: INC0010252<br>INC0010564
```

When moving the script to production, be sure to uncomment the **client.send\_message** portion of the script.