**ASSIGNMENT – SCHEDULED SCRIPT**

**Que.1 What are the entry points and the execution type of scheduled script.**

Schedule script is type of script that we set to run in some time interval without hampering the peak hours of any staff or employee of an organization.

* It is useful for long-running tasks.
* In scheduled script, we have only a single entry point. i.e. execute ().
* It contains the logic that will be executed when the entry point will get triggered.

There are two execution types of scheduled script.

1. ***Scheduled*** –

* This script runs the way we schedule it such as on hourly basis, daily basis, weekly basis, monthly basis and hourly basis.
* We can schedule through script deployment.

1. ***OnDemand*** -

* It can be triggered manually by the user.
* It can be done by going to the script deployment page and after clicking on Save & Execute button.

**Que.2 Create and deploy a scheduled script and print a log.**

* I have used **'N/log'** module to get the **log.debug()** in the Execution log after executing this file.
* We can see the debug output under execution log.

***Code:***

/\*\*

\*@NApiVersion 2.0

\*@NScriptType ScheduledScript

\*/

define(['N/search', 'N/record', 'N/email', 'N/runtime', 'N/log'],

function (search, record, email, runtime, log) {

function execute(scriptContext) {

log.debug('This is my ScheduledScript.');

}

return {

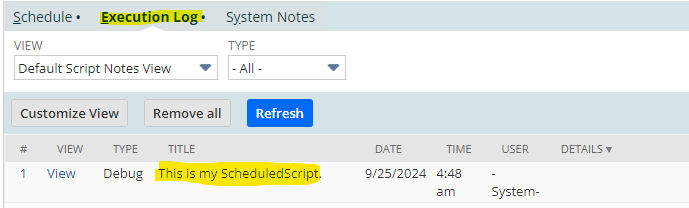
execute: execute

};

}

);

***Output:***



**Que.3 Pass the parameter to scheduled script through SuiteLet.**

* Here I have used **'N/task'** module and **create()** API to create tasks by having the task type, script id (internal id) and deploymentId of the scheduled script.
* I used **'N/runtime'** module to get the information about the current script execution context.

***SuiteLet Code:***

/\*\*

\* @NApiVersion 2.0

\* @NScriptType Suitelet

\*/

define(['N/ui/serverWidget','N/log', 'N/task', 'N/runtime'],

function(serverWidget, log, task, runtime) {

function onRequest(scriptContext) {

if(scriptContext.request.method == 'GET') {

var form = serverWidget.createForm({

title: 'Scheduled Script'

})

form.addField({

id: 'custpage\_param',

type: serverWidget.FieldType.TEXT,

label: 'Parameter'

})

form.addSubmitButton({

label: 'Submit'

})

scriptContext.response.writePage(form);

}else{

var paramValue = scriptContext.request.parameters.custpage\_param

var scheduleTask = task.create({

taskType: task.TaskType.SCHEDULED\_SCRIPT,

scriptId: 'customscript\_sch\_assignment\_sendmail',

deploymentId: 'customdeploy\_test\_d2'

})

scheduleTask.params = {

custscript\_param\_email: paramValue

}

var taskId = scheduleTask.submit()

log.debug('Scheduled script submitted' + taskId)

}

}

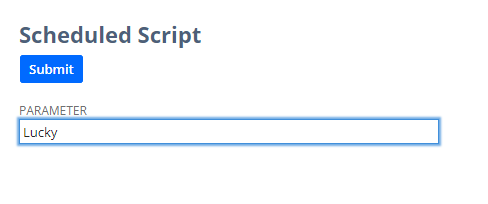
return {

onRequest: onRequest

}

}

)



***Scheduled Script:***

/\*\*

\*@NApiVersion 2.0

\*@NScriptType ScheduledScript

\*/

define(['N/search', 'N/record', 'N/email', 'N/runtime', 'N/log'],

function (search, record, email, runtime, log) {

function execute(scriptContext) {

// Q3

{

var paramValue = runtime.getCurrentScript().getParameter({

name: 'custscript\_param\_email'

})

log.debug(paramValue)

}

}

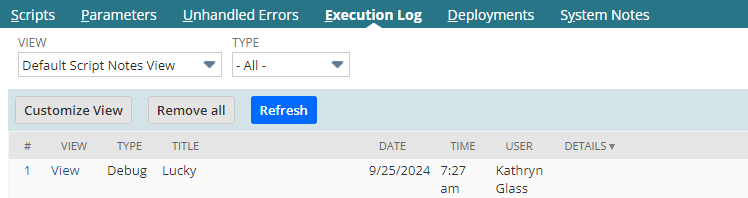
return {

execute: execute

};

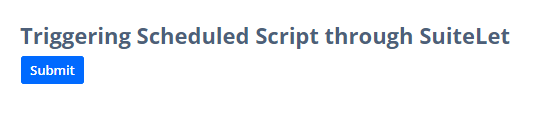
}

);



**Que.4 Trigger the scheduled script through a SuiteLet script on submit button.**

* Here I have used '**N/search'** and ‘**N/email**’ module, which will help me to create and run on-demand or saved searches and analyze and iterate through the search results and to send the email from within NetSuite respectively.
* I used **'N/task'** module and **create()** API to create tasks by having the task type, script id (internal id) and deploymentId of the scheduled script.
* I created a form using SuiteLet having a submit button. So, when I click on the Submit button the Scheduled script will get triggered, and it will send an email to the mentioned recipient.



***SuiteLet Code:***

/\*\*

\* @NApiVersion 2.0

\* @NScriptType Suitelet

\*/

define(['N/ui/serverWidget','N/log', 'N/task'],

function(serverWidget, log, task) {

function onRequest(scriptContext) {

if(scriptContext.request.method == 'GET') {

var form = serverWidget.createForm({

title: 'Triggering Scheduled Script through SuiteLet'

})

form.addSubmitButton({

label: 'Submit'

})

scriptContext.response.writePage(form);

}else{

var schScriptTask = task.create({

taskType: task.TaskType.SCHEDULED\_SCRIPT,

scriptId: 'customscript\_sch\_assignment\_sendmail',

deploymentId: 'customdeploy\_test\_d2'

})

var taskId = schScriptTask.submit()

log.debug('Scheduled script submitted' + taskId)

scriptContext.response.write('Scheduled script submitted with ' + taskId)

}

}

return {

onRequest: onRequest

}

}

)

***Scheduled Code:***

/\*\*

\*@NApiVersion 2.0

\*@NScriptType ScheduledScript

\*/

define(['N/search', 'N/record', 'N/email', 'N/runtime', 'N/log'],

function (search, record, email, runtime, log) {

function execute(scriptContext) {

// Q4

email.send({

author: -5,

recipients: '[adam@gilchristpty.com](mailto:adam@gilchristpty.com)',

subject: 'Hy Adam',

body: 'How are you?'

})

log.debug('Mail sent successfully')

}

return {

execute: execute

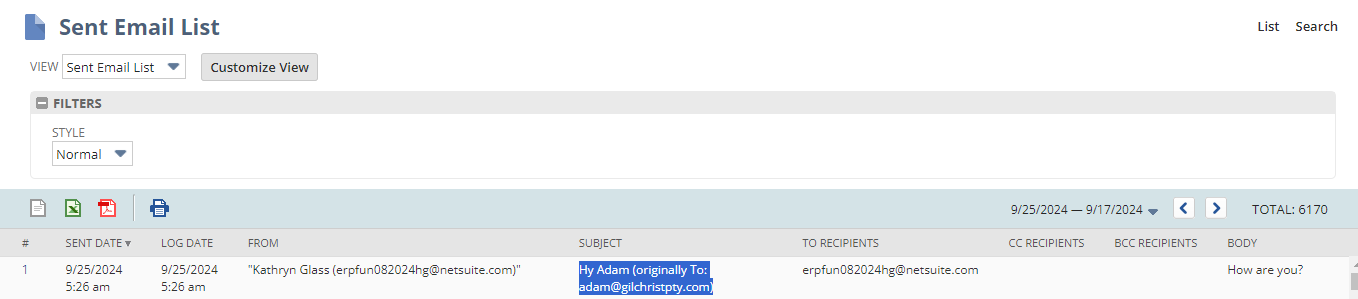
};

}

);



Now we can also cross check whether the mail has been sent or not by navigating to, Setup -> Company -> Sent Email List.



**Que.5 Create a scheduled script and make four deployments of the script. Print a different message on memo field through each deployment.**

* Here first of all I deployed the scheduled script four times and got their unique id.
* Then I got the deployment id of the current execution record.
* Then I run a switch loop by providing the deployment id of each deployment and respectively pass the message to the memo field of a Sales Order transaction record by providing the sales order id.
* Then simply save the record and log the memo message.

***Code:***

/\*\*

\*@NApiVersion 2.0

\*@NScriptType ScheduledScript

\*/

define(['N/search', 'N/record', 'N/email', 'N/runtime', 'N/log'],

function (search, record, email, runtime, log) {

function execute(scriptContext) {

// Q5

{

var scriptObj = runtime.getCurrentScript();

var deploymentId = scriptObj.deploymentId

switch (deploymentId) {

case 'customdeploy\_deployment\_memo1':

memo\_msg = 'MEMO MESSAGE OF 1ST DEPL';

break;

case 'customdeploy\_deployment\_memo2':

memo\_msg = 'MEMO MESSAGE OF 2ND DEPL';

break;

case 'customdeploy\_deployment\_memo3':

memo\_msg = 'MEMO MESSAGE OF 3RD DEPL ';

break;

case 'customdeploy\_deployment\_memo4':

memo\_msg = 'MEMO MESSAGE OF 4TH DEPL';

break;

default:

memo\_msg = 'Unknown deployment';

}

var salesOrder = record.load({

type: record.Type.SALES\_ORDER,

id: 22051

});

salesOrder.setValue({

fieldId: 'memo',

value: memo\_msg

})

salesOrder.save();

log.debug('Memo field updated : ' + memo\_msg);

}

}

return {

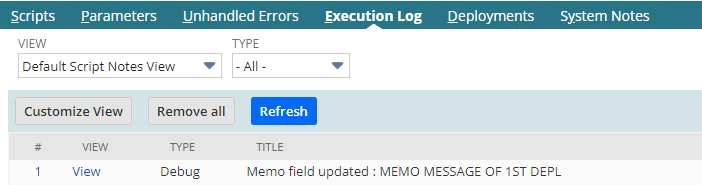
execute: execute

};

}

);

***For 1st deployment:***



***Similarly, below are the for 2nd, 3rd and 4th deployment respectively.***

