

Integration : HTTP-based web services allow diverse applications to talk to each other. ServiceNow supports both inbound (provider) and outbound (consumer) web services.

2 types :

I can send to them(outbound) -> servicenow sends -> outbound

they can send to me(Inbound) -> servicenow receives -> inbound

act as API - Application programming interfaces - connection between different applications

To integrate the other system it requires 7 parameters:

1. Type of integration - inbound/outbound
2. Endpoint - inbound/outbound
3. Authentication - Basic/Oauth
4. Method - POST,PUT,GET,DELETE,PATCH
5. Headers - Content-Type : application/json, Accept : application/json
6. payload
7. Mid server

Integration testing tools :

1. SOAP UI
2. Postman
3. REST API Explorer
4. Substituting in the REST Message

*** servicenow has the OOTB ability to create an endpoint with the help of "REST API Explorer"***

Note : Even though you can do CRUD, things is from REST API Explorer too as same as like POSTMAN

SOAP - Simple object access protocol - XML only

REST - (Representational State Transfer) - HTML, XML, plain text, and JSON.
JSON (JavaScript object notation)

Integration can be done in 3 ways:

1. Table API

2. Import set API
3. Scripted REST API

important notes for table API :

1. Table API doesn't validation Mandatory and other validation
2. You can use the HEAD methods in place of GET methods to return a response without a response body
3. You cannot pass multiple records in POST, PUT, and PATCH operations. If you do, only the first record is processed, the rest are ignored.

while testing in the REST API Explorer/POSTMAN, you have to pass the params.

sysparm_query = you can filter the table and copy the query

sysparm_limit = if it is "1" it will show one record

sysparm_fields = based on the fields which you entered it will show only those fields

sysparm_display_value = if true shows "display value" if it All it shows "Display value and sys_id".

sysparm_exclude_reference_link = true to exclude the record link, default : false

Activity:

Table API: REST

1. Create a record - POST

table name and payload is mandatory

2. Retrive a record - GET

table name and sys_id is mandatory

3. update a record - PUT/PATCH

PUT - modifies the entire record

PATCH - specified field in the record

table name, sys_id and payload is mandatory

4. Deleting a record - DELETE

sys_id and table is mandatory

it returns 204 - resCode successful but no response

Note : Dynamically using the response from one request to other request using environment variables

```
var jsonData = pm.response.json();  
var sys_id = jsonData.result.sys_id;  
pm.environment.set("sys_id", sys_id);
```

Need to do in PDI:

check in POSTMAN - Kalil's KT Session

Create a user and give appropriate roles to that user OOTB. create a new role and give access to create an incident or use appropriate existing role..

```
User id = kalil.integration.ootb  
password = AZva}<Qg-3b6]h9MVn72p?  
sys id = d5b62c6b831142104ab7f2a6feaad3fb
```

Table API : SOAP

base URL : <https://dev247558.service-now.com/incident.do?SOAP>

This URL will create all the CRUD operations

FYI, each operations you need to use service account User ID and password

SOAP :

Import set API(SOAP):

service account should have 'import_admin' role

FN > web services > inbound > create new > create import set and create Tmap

How it works?

When import set API creates with the transform map, it generates the WSDL endpoint, through that it provider will send the data it will store it on the source table and the transform map will process the data to the target table

Note :

If web services field are removed by default it will not get reflected in the WSDL endpoint

Web Service Fields		
	Label	Name ▲
×	Active	u_active
×	Activity due	u_activity_due
×	Additional assignee list	u_additional_assignee_list
×	Approval	u_approval
×	Approval history	u_approval_history
×	Approval set	u_approval_set
×	Assigned to	u_assigned_to
×	Assignment group	u_assignment_group
×	Business duration	u_business_duration

if I remove the assigned to and assignment group it will not get reflected in the SOAP app, when importing again the new updated web services in the WSDL.

Task 1: Create WSDL on incident with 4 fields.

check 'Imp set SOAP Kalil V1'. . .

FYI, before submitting the Import set API, fields needs to be created and if we want to make it mandatory, open the import set table > configure dictionary on the field and make it mandatory

Note : still without passing any values in that mandatory field, it will insert to avoid inserting use Tmap script.

Import set validation on the mandatory fields:

1. if doesn't pass the mandatory field still it create the record.
2. In reference record, if passes the value which is not present in the table it will create a record in the reference table, then display it in the field, this is due to in the field map choice action is given as 'create'

Source field	Configuration item ▼
Choice action	create ▼

.. ..

change it to 'reject' -> if record is not present then it will ignores.

Note : if any reference field or choice field, keep choice action as 'reject', but if keep as 'ignore' it won't create record on reference table, but it will keep empty in the field and it will create a record in the import set table

if mandatory fields passes the empty value, and it should not create the record in the import set table, make this

option 'enforce mandatory fields'

Run business rules ☒

Enforce mandatory fields

Only Mapped Fields

Copy empty fields ☐

```
(function transformRow(source, target, map, log, isUpdate) {  
    // Add your code here  
    var des = source.u_description;  
    if (des.startsWith('Soap') == true) {  
        //  
    } else {  
        error = true;  
        source.sys_row_error = 'Incident description should starts with Soap  
word';  
        source.sys_import_state_comment = 'Incident description should starts  
with Soap word';  
    }  
})(source, target, map, log, action === "update");
```

so sys_import_state_comment will be displayed in the SOAP transcation error to know the client what kind of error they got, this field can be stored in sys_import_set_row table

TH0001008	?	Transform Map Script Error.	Incident description should starts with ...
TH0001010	?	Transform Map Script Error.	Incident description should starts with ...
TH0001007	?	Transform Map Script Error.	Incident description should starts with ...

error in SOAP:

```
<table>incident</table>
```

```
<display_name/>
<display_value/>
<status>error</status>
<status_message>Incident description should starts with Soap
word</status_message>
<error_message>Transform Map Script Error.</error_message>
```

Import set API (REST):

Has 3 types,

1. Create a record in import set table - POST {table}
2. create a multiple record in the import set table - POST {table} - after the endpoint '/insertMultiple'
3. Retrive an import set record - GET {table, sys_id}

for multiple insert, there will additional validation in the table:

[Import Set API - Goodbye SRAPIs? - ServiceNow Community](#)

steps :

1. Create a record in the 'sys_rest_insert_multiple' and source table as 'import set table' and transformation as a 'synchronous'.
2. In related list(column mappings) > click New > Type = JSON > column mappings = Column name

to generate the endpoint use 'REST API Explorer'

Scripted Web services:

SOAP:

FN > Scripted soap services.

define the name and description.

it will generate the wsdl link.....

put parameters	put parameters
----------------	----------------

soap if user gives info.	will return back the response
<div> <div>Input Parameters (1)</div> <div>Output Parameters (6)</div> </div> <div> <div> <div>≡</div> <div>🔍</div> <div>💬</div> </div> <div>Order ▾</div> <div>Search</div> </div> <div>Web service = getGroupInfo</div> <div> <div>☐ 🔍</div> <div>Name</div> </div> <div>group_name</div>	<div> <div>Input Parameters (1)</div> <div>Output Parameters (6)</div> </div> <div> <div> <div>≡</div> <div>🔍</div> <div>💬</div> </div> <div>Order ▾</div> <div>Search</div> </div> <div>Web service = getGroupInfo</div> <div> <div>☐ 🔍</div> <div>Name</div> </div> <div>group_name</div> <div>group_manager</div> <div>group_email</div> <div>group_roles</div> <div>group_members</div> <div>results</div>
<pre> soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" <soapenv:Header/> <soapenv:Body> <get:execute> <!--Optional:--> <group_name>?</group_name> </get:execute> </soapenv:Body> </soapenv:Envelope> </pre> <p>Output parameters can be called as 'request'</p>	<p>Input parameters can be called as 'response'</p>

code :

the below code is to 'execute' the SOAP scripted message for creating a record in the table set the function name to 'create'

instead of 'execute'

```

(function scriptedWebServiceOperation(request, response) {
  // Add your code here
  var grp = request.group_name;
  if (!gs.nil(grp)) {
    var gr = new GlideRecord('sys_user_group');

```

```

        if (gr.get('name', grp)) {
            response.group_name = gr.name;
            response.group_manager = gr.manager.name;
            response.group_email = gr.email;
            response.results = 'Group ' + grp + ' is found in the servicenow
group table';
            response.group_members = getMembers(gr.sys_id);
            response.group_roles = getRoles(gr.sys_id);
        } else {
            response.results = 'Group ' + grp + ' not found in the servicenow
group table';
        }
    } else {
        response.results = 'Group name is Mandatory';
    }
}

```

```

function getMembers(grID) {
    var users = '';
    var grm = new GlideRecord('sys_user_grmember');
    var query = 'group=' + grID;
    grm.addEncodedQuery(query);
    grm.query();
    while (grm.next()) { // it will loop all the records
        if (users) {
            users += ',' + grm.user.name;
        } else {
            users += grm.user.name;
        }
    }
    return users;
}

```

```

function getRoles(grID) {
    var roles = '';
    var grR = new GlideRecord('sys_group_has_role');
    var query = 'group=' + grID;

```



```

        grR.addEncodedQuery(query);
        grR.query();
        while (grR.next()) {
            if (roles) {
                roles += ',' + grR.role.name;
            } else {
                roles += grR.role.name;
            }
        }
        return roles;
    }
})(request, response);

```

output : if value passes and found a record in the group table

```

<executeResponse>
    <group_name>CAB Approval</group_name>
    <group_manager>System Administrator</group_manager>
    <group_email>cab@gmail.com</group_email>
    <results>Group CAB Approval is found in the servicenow group
table</results>
    <group_members>Howard Johnson,Luke Wilson,Ron Kettering,Christen
Mitchell,cab approver,Bernard Laboy</group_members>
    <group_roles>itil,change_manager</group_roles>
</executeResponse>

```

If empty values passes in the group name:

```

<executeResponse>
    <results>Group name is Mandatory</results>
</executeResponse>

```

Scripted web service:

REST: you can create a lot of methods in one scripted REST web services

FN > scripted rest api

creating a new record it generates the endpoint
'/api/707171/kalil_rest_scripted_v1'

in 'content negotiations' it declares what kind of request and response that I can alter

Security Content Negotiation Documentation

By default, all resources in an API support the following request and response formats: application/json, application/xml, and text/xml.
The default supported formats can be overridden here at the API level and will apply to all resources belonging to the API, unless an individual resource overrides the defaults.
[More Info](#)

Override default supported request formats ☐

Default supported request formats

Override default supported response formats ☐

Default supported response formats

in the resources, can create the what kind of call/HTTP method that can be used

API analytics

Resources (1) Request Headers Query Parameters

API definition = Kalil REST scripted v1

Name	HTTP method	Relative path	Resource path
getGroupInfo	GET	/	/api/707171/kalil_rest_scripted_v1

it is similar to input and output parameters in the SOAP for the REST services

Request Header Associations Query Parameter Associations

for text Search

API resource = getGroupInfo

API request header

to query the record, then create a query parameters associations record and it can be fetched by 'queryParams', please refer the code

and json value to print the records.

code: GET Method

```
(function process( /*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {  
    var grp = request.queryParams.group_name.toString();
```

```

if (!gs.nil(grp)) {
    var gr = new GlideRecord('sys_user_group');
    var json = {};
    if (gr.get('name', grp)) {
        json.group_name = gr.name;
        json.group_manager = gr.manager.name;
        json.group_email = gr.email;
        json.results = 'Group ' + grp + ' is found in the servicenow group
table';

        json.group_members = getMembers(gr.sys_id);
        json.group_roles = getRoles(gr.sys_id);
        response.setStatus(200);
    } else {
        response.setStatus(400);
        json.results = 'Group ' + grp + ' not found in the servicenow
group table';
    }
} else {
    response.setStatus(400);
    json.results = 'Group name is Mandatory';
}
response.setBody(json);

```

```

function getMembers(grID) {
    var users = '';
    var grm = new GlideRecord('sys_user_grmember');
    var query = 'group=' + grID;
    grm.addEncodedQuery(query);
    grm.query();
    while (grm.next()) { // it will loop all the records
        if (users) {
            users += ',' + grm.user.name;
        } else {
            users += grm.user.name;
        }
    }
}

```

```

    }

    return users;
}

function getRoles(grID) {
    var roles = '';
    var grR = new GlideRecord('sys_group_has_role');
    var query = 'group=' + grID;
    grR.addEncodedQuery(query);
    grR.query();
    while (grR.next()) {
        if (roles) {
            roles += ',' + grR.role.name;
        } else {
            roles += grR.role.name;
        }
    }
    return roles;
}

})(request, response);

```

output :

```

{
  "result": {
    "group_name": "CAB Approval",
    "group_manager": "System Administrator",
    "group_email": "cab@gmail.com",
    "results": "Group CAB Approval is found in the servicenow group
table",
    "group_members": "Howard Johnson,Luke Wilson,Ron Kettering,Christen
Mitchell,cab approver,Bernard Laboy",
    "group_roles": "itil,change_manager"
  }
}

```

code 2: POST Method

note:

All input parameters should be stored in the **`var data = request.body.data`**
then response can be created as JSON payload.

```
(function process( /*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {

    var data = request.body.data; // input parameters - this is the way it can
    be declared

    var outputJson = {};
    var v1 = mandatoryValidation();
    if (v1 == true) {
        var grInc = new GlideRecord("incident");
        grInc.initialize(); //it creates an empty record to populate the value
        grInc.caller_id = data.Caller_ID;
        grInc.short_description = data.Short_Description;
        var grCI1 = new GlideRecord('cmdb_ci');
        grCI1.get('name', data.CI);
        grInc.cmdb_ci = grCI1.sys_id; //because its a reference field you
        need to get sys_id to populate the value
        if (data.Impact) {
            grInc.impact = data.Impact;
        }
        if (data.Urgency) {
            grInc.urgency = data.Urgency;
        }
        grInc.insert();
        outputJson.Result = "Incident Created";
        outputJson.sys_id = grInc.sys_id;
        outputJson.Number = grInc.number;
    } else {
        outputJson.Result = v1;
        response.setStatus(405);
    }
}
```

```

    }

    response.setBody(outputJson); // output parameters stored as a JSON
    structure

    function mandatoryValidation() {

        if (data.Caller_ID) {

            if (data.CI) {

                var grCI = new GlideRecord('cmdb_ci');
                grCI.addQuery('name', data.CI);
                grCI.addQuery('operational_status', 1);
                grCI.query();

                if (grCI.next()) {

                    if (data.Short_Description) {

                        if (data.Impact && data.Impact == 1) {

                            return "Impact cannot be high";

                        } else {

                            if (data.Urgency && data.Urgency == 1) {

                                return "Urgency cannot be high";

                            } else {

                                return true;

                            }

                        }

                    } else {

                        return "Short Description should be mandatory";

                    }

                } else {

                    return "CI is not found and It is not operational";

                }

            } else {

                return "Configuration Item should be mandatory";

            }

        } else {

            return "Caller_ID should be mandatory";

        }

    }

```

```
})(request, response);
```

Note : you cannot create another same method in the same scripted REST API it will not allow to create it, it throws an error

output :

```
{
  "result": {
    "Result": "Incident Created",
    "sys_id": "ab2f2300836d42104ab7f2a6feaad344",
    "Number": "INC0010046"
  }
}
```

```
-----
-----
-----
```

Ebonding - with custom spoke(Integration hub) :

Integration : HTTP-based web services allow diverse applications to talk to each other. ServiceNow supports both inbound (provider) and outbound (consumer) web services.

2 types :

I can send to them(outbound) -> servicenow sends -> outbound

they can send to me(Inbound) -> servicenow receives -> inbound

act as API - Application programming interfaces - connection between different applications

To integrate the other system it requires 7 paramaters:

1. Type of integration - inbound/outbound
2. Endpoint - inbound/outbound
3. Authentication - Basic/Oauth
4. Method - POST, PUT, GET, DELETE, PATCH
5. Headers - Content-Type : application/json, Accept : application/json
6. payload
7. Mid server

Integration testing tools :

5. SOAP UI
6. Postman
7. REST API Explorer
8. Substituting in the REST Message

*** servicenow has the OOTB ability to create an endpoint with the help of "REST API Explorer"***

Note : Even though you can do CRUD, things is from REST API Explorer too as same as like POSTMAN

SOAP - Simple object access protocol - XML only

REST - (Representational State Transfer) - HTML, XML, plain text, and JSON.
JSON (JavaScript object notation)

Integration can be done in 3 ways:

1. Table API
2. Import set API
3. Scripted REST API

important notes for table API :

4. Table API doesn't validation Mandatory and other validation
5. You can use the HEAD methods in place of GET methods to return a response without a response body
6. You cannot pass multiple records in POST, PUT, and PATCH operations.
If you do, only the first record is processed, the rest are ignored.

while testing in the REST API Explorer/POSTMAN, you have to pass the params.

sysparm_query = you can filter the table and copy the query

sysparm_limit = if it is "1" it will show one record

sysparm_fields = based on the fields which you entered it will show only those fields

sysparm_display_value = if true shows "display value" if it All it shows "Display value and sys_id".

sysparm_exclude_reference_link = true to exclude the record link, default : false

Activity:

Table API: REST

1. Create a record - POST

table name and payload is mandatory

2. Retrive a record - GET

table name and sys_id is mandatory

3. update a record - PUT/PATCH

PUT - modifies the entire record

PATCH - specified field in the record

table name, sys_id and payload is mandatory

4. Deleting a record - DELETE

sys_id and table is mandatory

it returns 204 - resCode successful but no response

Note : Dynamically using the response from one request to other request using environment variables

```
var jsonData = pm.response.json();  
var sys_id = jsonData.result.sys_id;  
pm.environment.set("sys_id", sys_id);
```

Need to do in PDI:

check in POSTMAN - Kalil's KT Session

Create a user and give appropriate roles to that user OOTB. create a new role and give access to create an incident or use appropriate existing role..

User id = kalil.integration.ootb

password = AZva}<Qg-3b6]h9MVn72p?

sys id = d5b62c6b831142104ab7f2a6feaad3fb

Table API : SOAP

base URL : <https://dev247558.service-now.com/incident.do?SOAP>

This URL will create all the CRUD operations

FYI, each operations you need to use service account User ID and password

SOAP :

Import set API (SOAP) :

service account should have 'import_admin' role

FN > web services > inbound > create new > create import set and create Tmap

How it works?

When import set API creates with the transform map, it generates the WSDL endpoint, through that it provider will send the data it will store it on the source table and the transform map will process the data to the target table

Note :

If web services field are removed by default it will not get reflected in the WSDL endpoint

Web Service Fields		
	Label	Name ▲
×	Active	u_active
×	Activity due	u_activity_due
×	Additional assignee list	u_additional_assignee_list
×	Approval	u_approval
×	Approval history	u_approval_history
×	Approval set	u_approval_set
×	Assigned to	u_assigned_to
×	Assignment group	u_assignment_group
×	Business duration	u_business_duration

if I remove the assigned to and assignment group it will not get reflected in the SOAP app, when importing again the new updated web services in the WSDL.

Task 1: Create WSDL on incident with 4 fields.

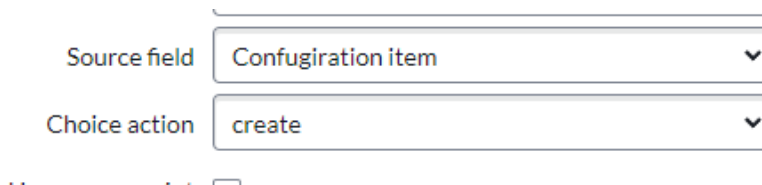
check 'Imp set SOAP Kalil V1'. . .

FYI, before submitting the Import set API, fields needs to be created and if we want to make it mandatory, open the import set table > configure dictionary on the field and make it mandatory

Note : still without passing any values in that mandatory field, it will insert to avoid inserting use Tmap script.

Import set validation on the mandatory fields:

1. if doesn't pass the mandatory field still it create the record.
2. In reference record, if passes the value which is not present in the table it will create a record in the reference table, then display it in the field, this is due to in the field map choice action is given as 'create'



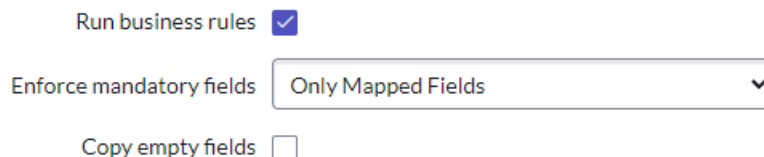
The image shows a configuration interface with two dropdown menus. The first dropdown, labeled 'Source field', has 'Confugiration item' selected. The second dropdown, labeled 'Choice action', has 'create' selected. Below these, there are some faint, partially visible labels like '..' and '...'.

change it to 'reject' -> if record is not present then it will ignores.

Note : if any reference field or choice field, keep choice action as 'reject', but if keep as 'ignore' it won't create record on reference table, but it will keep empty in the field and it will create a record in the import set table

if mandatory fields passes the empty value, and it should not create the record in the import set table, make this

option 'enforce mandatory fields'



The image shows a configuration interface with three options. The first option, 'Run business rules', is checked with a blue checkbox. The second option, 'Enforce mandatory fields', has a dropdown menu set to 'Only Mapped Fields'. The third option, 'Copy empty fields', is unchecked with an empty checkbox.

```
(function transformRow(source, target, map, log, isUpdate) {  
    // Add your code here  
    var des = source.u_description;  
    if (des.startsWith('Soap') == true) {  
        //  
    }  
}
```

```

    } else {
        error = true;
        source.sys_row_error = 'Incident description should starts with Soap
word';
        source.sys_import_state_comment = 'Incident description should starts
with Soap word';
    }
})(source, target, map, log, action === "update");

```

so sys_import_state_comment will be displayed in the SOAP transaction error to know the client what kind of error they got, this field can be stored in sys_import_set_row table

TH0001008	?	Transform Map Script Error.	Incident description should starts with ...
TH0001010	?	Transform Map Script Error.	Incident description should starts with ...
TH0001007	?	Transform Map Script Error.	Incident description should starts with ...

error in SOAP:

```

<table>incident</table>
<display_name/>
<display_value/>
<status>error</status>
<status_message>Incident description should starts with Soap
word</status_message>
<error_message>Transform Map Script Error.</error_message>

```

Import set API (REST):

Has 3 types,

1. Create a record in import set table - POST {table}
2. create a multiple record in the import set table - POST {table} - after the endpoint '/insertMultiple'
3. Retrieve an import set record - GET {table, sys_id}

for multiple insert, there will additional validation in the table:

[Import Set API - Goodbye SRAPIs? - ServiceNow Community](#)

steps :

- 1. Create a record in the 'sys_rest_insert_multiple' and source table as 'import set table' and transformation as a 'synchronous'.
- 2. In related list(column mappings) > click New > Type = JSON > column mappings = Column name

to generate the endpoint use 'REST API Explorer'

Scripted Web services:
SOAP:

FN > Scripted soap services.
define the name and description.
it will generate the wsdl link.....

out parameters	put parameters
soap if user gives info.	will resturn back the response
<div><div>Input Parameters (1)Output Parameters (6)</div><div><div><div><div></div><div></div><div></div></div><div>Order</div><div>Search</div></div><div>Web service = getGroupInfo</div><div><div><div><div></div><div></div><div></div></div><div>Name</div></div><div>group_name</div></div></div></div>	<div><div>Input Parameters (1)Output Parameters (6)</div><div><div><div><div></div><div></div><div></div></div><div>Order</div><div>Search</div></div><div>Web service = getGroupInfo</div><div><div><div><div></div><div></div><div></div></div><div>Name</div></div><div>group_name</div><div>group_manager</div><div>group_email</div><div>group_roles</div><div>group_members</div><div>results</div></div></div></div>
<pre>soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" <soapenv:Header/> <soapenv:Body> <get:execute> <!--Optional:--> <group_name?></group_name> </get:execute> </soapenv:Body> </soapenv:Envelope></pre>	

put parameters can be called as 'request'	put parameters can be called as 'response'
----------------------------------------------	-----------------------------------------------

code :

the below code is to 'execute' the SOAP scripted message for creating a record in the table set the function name to 'create'

instead of 'execute'

```
(function scriptedWebServiceOperation(request, response) {  
    // Add your code here  
    var grp = request.group_name;  
    if (!gs.nil(grp)) {  
        var gr = new GlideRecord('sys_user_group');  
        if (gr.get('name', grp)) {  
            response.group_name = gr.name;  
            response.group_manager = gr.manager.name;  
            response.group_email = gr.email;  
            response.results = 'Group ' + grp + ' is found in the servicenow  
group table';  
            response.group_members = getMembers(gr.sys_id);  
            response.group_roles = getRoles(gr.sys_id);  
        } else {  
            response.results = 'Group ' + grp + ' not found in the servicenow  
group table';  
        }  
    } else {  
        response.results = 'Group name is Mandatory';  
    }  
}  
  
function getMembers(grID) {  
    var users = '';
```

```

var grm = new GlideRecord('sys_user_grmember');
var query = 'group=' + grID;
grm.addEncodedQuery(query);
grm.query();
while (grm.next()) { // it will loop all the records
    if (users) {
        users += ',' + grm.user.name;
    } else {
        users += grm.user.name;
    }
}
return users;
}

function getRoles(grID) {
    var roles = '';
    var grR = new GlideRecord('sys_group_has_role');
    var query = 'group=' + grID;
    grR.addEncodedQuery(query);
    grR.query();
    while (grR.next()) {
        if (roles) {
            roles += ',' + grR.role.name;
        } else {
            roles += grR.role.name;
        }
    }
    return roles;
}

})(request, response);

```

output : if value passes and found a record in the group table

<executeResponse>

<group_name>CAB Approval</group_name>

```

    <group_manager>System Administrator</group_manager>

    <group_email>cab@gmail.com</group_email>

    <results>Group CAB Approval is found in the servicenow group
table</results>

    <group_members>Howard Johnson,Luke Wilson,Ron Kettering,Christen
Mitchell,cab approver,Bernard Laboy</group_members>

    <group_roles>itil,change_manager</group_roles>
</executeResponse>

```

If empty values passes in the group name:

```

<executeResponse>

    <results>Group name is Mandatory</results>
</executeResponse>

```

Scripted web service:

REST: you can create a lot of methods in one scripted REST web services

FN > scripted rest api

creating a new record it generates the endpoint
'/api/707171/kalil_rest_scripted_v1'

in 'content negotiations' it declares what kind of request and response that I can alter

The screenshot shows the 'Content Negotiation' tab in a REST API configuration interface. It contains a blue informational box with the text: 'By default, all resources in an API support the following request and response formats: application/json, application/xml, and text/xml. The default supported formats can be overridden here at the API level and will apply to all resources belonging to the API, unless an individual resource overrides the defaults.' Below this, there are two sections: 'Override default supported request formats' and 'Override default supported response formats', each with an unchecked checkbox. Under each checkbox, there is a text input field containing the default formats: 'application/json,application/xml,text/xml'.

in the resources, can create the what kind of call/HTTP method that can be used

API analytics

Resources (1)

Request Headers

Query Parameters

Name

Search

API definition = Kalil REST scripted v1

<div><div></div><div></div></div>	Name ▲	HTTP method	Relative path	Resource path
	getGroupInfo	GET	/	/api/707171/kalil_rest_scripted_v1

it is similar to input and output parameters in the SOAP for the REST services

Request Header Associations		Query Parameter Associations	
<div> <div> <div></div> <div></div> </div> <div> <div>for text</div> <div>Search</div> </div> </div>			
API resource = getGroupInfo			
<input type="text"/>	<input type="text"/>	API request header	

to query the record, then create a query parameters associations record and it can fetched by 'queryParams', please refer the code

and json value to print the records.

code: GET Method

```
(function process( /*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {
    var grp = request.queryParams.group_name.toString();
    if (!gs.nil(grp)) {
        var gr = new GlideRecord('sys_user_group');
        var json = {};
        if (gr.get('name', grp)) {
            json.group_name = gr.name;
            json.group_manager = gr.manager.name;
            json.group_email = gr.email;
            json.results = 'Group ' + grp + ' is found in the servicenow group
table';
            json.group_members = getMembers(gr.sys_id);
            json.group_roles = getRoles(gr.sys_id);
            response.setStatus(200);
        } else {
```

```

        response.setStatus(400);

        json.results = 'Group ' + grp + ' not found in the servicenow
group table';
    }
} else {
    response.setStatus(400);
    json.results = 'Group name is Mandatory';
}
response.setBody(json);

```

```

function getMembers(grID) {
    var users = '';
    var grm = new GlideRecord('sys_user_grmember');
    var query = 'group=' + grID;
    grm.addEncodedQuery(query);
    grm.query();
    while (grm.next()) { // it will loop all the records
        if (users) {
            users += ',' + grm.user.name;
        } else {
            users += grm.user.name;
        }
    }
    return users;
}

```

```

function getRoles(grID) {
    var roles = '';
    var grR = new GlideRecord('sys_group_has_role');
    var query = 'group=' + grID;
    grR.addEncodedQuery(query);
    grR.query();
    while (grR.next()) {
        if (roles) {
            roles += ',' + grR.role.name;
        } else {

```

```

        roles += grR.role.name;
    }
}
return roles;
}
})(request, response);

```

output :

```

{
  "result": {
    "group_name": "CAB Approval",
    "group_manager": "System Administrator",
    "group_email": "cab@gmail.com",
    "results": "Group CAB Approval is found in the servicenow group
table",
    "group_members": "Howard Johnson,Luke Wilson,Ron Kettering,Christen
Mitchell,cab approver,Bernard Laboy",
    "group_roles": "itil,change_manager"
  }
}

```

code 2: POST Method

note:

All input parameters should be stored in the **var data = request.body.data** then response can created as JSON payload.

```

(function process( /*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {

    var data = request.body.data; // input parameters - this is the way it can
be declared

    var outputJson = {};

    var v1 = mandatoryValidation();

    if (v1 == true) {

```

```

var grInc = new GlideRecord("incident");
grInc.initialize(); //it creates an empty record to populate the value
grInc.caller_id = data.Caller_ID;
grInc.short_description = data.Short_Description;
var grCI1 = new GlideRecord('cmdb_ci');
grCI1.get('name', data.CI);
grInc.cmdb_ci = grCI1.sys_id; //because its a reference field you
need to get sys_id to populate the value
if (data.Impact) {
    grInc.impact = data.Impact;
}
if (data.Urgency) {
    grInc.urgency = data.Urgency;
}
grInc.insert();
outputJson.Result = "Incident Created";
outputJson.sys_id = grInc.sys_id;
outputJson.Number = grInc.number;
} else {
    outputJson.Result = v1;
    response.setStatus(405);
}

response.setBody(outputJson); // output parameters stored as a JSON
structure

function mandatoryValidation() {
    if (data.Caller_ID) {
        if (data.CI) {
            var grCI = new GlideRecord('cmdb_ci');
            grCI.addQuery('name', data.CI);
            grCI.addQuery('operational_status', 1);
            grCI.query();
            if (grCI.next()) {
                if (data.Short_Description) {
                    if (data.Impact && data.Impact == 1) {
                        return "Impact cannot be high";
                    }
                }
            }
        }
    }
}

```

```

        } else {
            if (data.Urgency && data.Urgency == 1) {
                return "Urgency cannot be high";
            } else {
                return true;
            }
        }
    } else {
        return "Short Description should be mandatory";
    }
} else {
    return "CI is not found and It is not operational";
}
} else {
    return "Configuration Item should be mandatory";
}
} else {
    return "Caller_ID should be mandatory";
}
}

})(request, response);

```

Note : you cannot create another same method in the same scripted REST API it will not allow to create it, it throws an error

output :

```

{
  "result": {
    "Result": "Incident Created",
    "sys_id": "ab2f2300836d42104ab7f2a6feaad344",
    "Number": "INC0010046"
  }
}

```


Ebonding - with custom spoke(Integration hub) :

Custom spoke is custom application : [integrationhub-definition-use.pdf \(servicenow.com\)](#)

[ServiceNow Remote instance spoke implementation pr... - ServiceNow Community](#)

[Set up the ServiceNow Remote Instance spoke](#)

updating the worknotes :

get worknotes and coments :[How can I get all list of work notes in incident? - ServiceNow Community](#)

name=incident^element_id=56bee4a6dbc2b190c531a9c813961933

[getJournalEntry \[ServiceNow\]](#)

[ServiceNow Journal Field | ServiceNow getJournalEntry\(\) Method | ServiceNow Journal Entry example](#)

system application > studio

step 1 : create application : eBonding with POC

step 2 : decide which is local and remote instance, in our activity we can take

local : Dev - where communication is initited and established

remote : POC - local instance communicates

step 3 : on remote instance create app registry to provide OAUTH details for external clients

3.1 : FN > app registry > create new > create an Oauth endpoint for external clients

KT: Kalil_V1.IH-appReg_POC

redirect URL : local instance url : https://fiatchryslerdev.servicenow.com/oauth_redirect.do

make a note of client id, client secret and redirect URL

step 4 : register remote instance OAUTH registry in the local instance

4.1 : FN > app registry > create new > connect to third party client

copy and paste the remote instance client id, client secret, default type = authorization code,

authorization URL : remote instance url - https://fcadriveitpoc.service-now.com/oauth_auth.do

token url : remote instance url - https://fcadriveitpoc.service-now.com/oauth_token.do

redirect URL : local instance url : https://fiatchryslerdev.service-now.com/oauth_redirect.do

step 5 : on local instance create credential record

5.1 : FN > connections & credentials > credentials > create new > OAUTH 2.0 credentials

create credentials "KT: Kalil_V1.IH-credentialsOAuth_DEV" - this is used to generate the OAUTH Token

you can see the "Get Oauth Token".

once you hit the "Get Oauth token" it will generate the refresh and access token in the "manage token"

step 6 : on local instance create a credentials & connection alias record to create a connection in the related link

of alias record "KT_KalilV1IHconnCredAliasDEV" - naming convention is important it wont accept more than

special character.

6.1 : after creating a alias record then you can create a own configuration template : KT: Kalil_V1.IH-conn&CredTemplates_DEV

6.2 : under the alias record related list create a new connection record

connection record name as "KT: Kalil_V1.IH-connections_DEV" to store the connection URL

because this connection url will get easily called in the "REST Message in custom action in custom spoke"

step 7 : on remote instance create credential record

7.1 : FN > connections & credentials > credentials > click New > select Basic Auth

give name as "KT: Kalil_V1.IH-credBasicAuth_POC"

create user name and password

user name = Integration.eConnect.POCV1

password = Zaynab@2713

7.2 : Create the same user in the local instance and give appropriate access to the user based on the API consumption

step 8 : on remote instance create credentials & connection alias record and under create a connection record in the alias

related list

8.1 : cred&conn alias record = KT_KalilV1IHconnCredAliasPOC

cred&conn template = KT: Kalil_V1.IH-conn&CredTemplates_POC

connection record needs to get created in the related list of alias record, connection name = KT: Kalil_V1.IH-connections_POC

connection Url : local instance URL : <https://fiatchryslerdev.service-now.com/>

Once all the connections is done then start creating the custom actions..

step 1 : create custom actions in the custom spoke...

in custom action you have inputs,data actions,outputs

step 1 : inputs

Note : create the field based on the requirements and for any reference field try to use the reference table

example : in incident table, affected_user field is referring to the sys_user table, so use the appropriate references

step 2: create a payload, how to make the inputs as a payload/JSON structure to happen transaction in the rest message

add "script action/JSON Builder step" after the input action.

option 1 : script atep

if it is "script step"

you have to create inputs for script from "inputs(step 1)"

1. Create Payload

Input Variables

Name	Value
<input type="text" value="impacted_application"/>	<input type="text" value="a... ▶ ... ▶ Configuration Item ... X"/>
<input type="text" value="configuration_item"/>	<input type="text" value="a... ▶ ... ▶ Configuration Item ... X"/>
<input type="text" value="short_description"/>	<input type="text" value="action ▶ Short Description X"/>
<input type="text" value="description"/>	<input type="text" value="action ▶ Description X"/>
<input type="text" value="affected_user"/>	<input type="text" value="action ▶ ... ▶ User ID X"/>
<input type="text" value="reported_by"/>	<input type="text" value="action ▶ ... ▶ User ID X"/>
<input type="text" value="priority"/>	<input type="text" value="action ▶ Priority X"/>
<input type="text" value="correlation_display"/>	<input type="text" value="action ▶ Correlation Display X"/>
<input type="text" value="correlation_id"/>	<input type="text" value="action ▶ Correlation ID X"/>
<input type="text" value="assignment_group"/>	<input type="text" value="action ▶ ... ▶ Name X"/>

[⊕ Create Variable](#)

call those input as variables in the script by using :

```
(function execute(inputs, outputs) {  
    // ... code ...  
  
    var payload = {};  
    payload.impacted_application = inputs['impacted_application'];  
    payload.configuration_item = inputs['configuration_item'];  
    payload.short_description = inputs['short_description'].trim();  
    payload.description = inputs['description'].trim();  
    payload.affected_user = inputs['affected_user'];  
    payload.reported_by = inputs['reported_by'];  
    payload.priority = inputs['priority'];  
    payload.correlation_display = inputs['correlation_display'].trim();  
    payload.correlation_id = inputs['correlation_id'].trim();  
    payload.assignment_group = inputs['assignment_group'];
```

```
outputs['payload'] = JSON.stringify(payload);
```

```
})(inputs, outputs);
```

those input needs to be supplied in to output variable, you have create a output variable and pass the JSON into it.

Output Variables

Label	Name	Type	Mandatory	
:: payload	payload	String	<input type="checkbox"/>	

[+ Create Variable](#)

option 2.2: JSON Builder step

get inputs from step1 using data pills to generate the payload

in the input of JSON builder you need to give the name, it should not be used from the inputs

1. JSON Builder: create paylo...

Name	Value	Type
impacted_application	action ▶ impacted_application ✕	String
confugiration_item	action ▶ confugiration_item ✕	String
short_description	action ▶ short_description ✕	String
description	action ▶ description ✕	String
affected_user	action ▶ affected_user ✕	String
correlation_id	action ▶ correlation_id ✕	String
work_notes	action ▶ work_notes ✕	String

Include Outer: ☐

Structure

preview ⓘ

```
{
  "impacted_application": "action ▶ impacted_application",
  "confugiration_item": "action ▶ confugiration_item",
  "short_description": "action ▶ short_description",
  "description": "action ▶ description",
  "affected_user": "action ▶ affected_user",
  "correlation_id": "action ▶ correlation_id",
  "work_notes": "action ▶ work_notes"
}
```

step 3: configure the rest message by calling the alias of the OAUTH credentials

use connection alias

Connection Details ⓘ

Connection Use Connection Alias

Connection Alias x_fich_ebondingpoc.KT_KaliIV1IHconnCredAliasDEV

Base URL https://fcadriveitpoc.service-now.com/

Test REST Step

fill the request details

Request Details ⓘ

Build Request Manually

Resource Path api/now/import/x_fich_econnect_fca_api_inc_create_v2

HTTP Method POST

Query Parameters

Name	Value
sysparm_exclude_reference_link	true

Headers

Name	Value
Accept	application/json
Content-Type	application/json

pass the JSON builder output into the request body


Request Content

Request Type

Request Body[Text] step ▶ ... ▶ Output ✕

9947 characters remaining

step 4 : handle the response output using script.

3. Handle response Script 

step ▶ ... ▶ Status Code ✕

step ▶ ... ▶ Response Body ✕

[⊕ Create Variable](#)

in the script : decode and parse is same it will give us [object Object]

```
(function execute(inputs, outputs) {  
  // ... code ...  
  
  var responseBody = new global.JSON().decode(inputs['response_body']);  
  gs.info('ARK:::create remote incident[responseBody-stringify] =' +  
    JSON.stringify(inputs['response_body']));  
  gs.info('ARK:::create remote incident[responseBody-decode] =' + new  
    global.JSON().decode(inputs['response_body']));  
  gs.info('ARK:::create remote incident[responseBody-info] =' +  
    JSON.parse(inputs['response_body']));  
  
  if(responseBody.result[0].status == 'inserted'){  
    outputs['status'] = 'Success';  
    outputs['incident_number'] = responseBody.result[0].display_value;  
    outputs['message'] = 'Incident is created in the remote instance';  
  } else {  
    outputs['status'] = responseBody.result[0].status;  
  }  
}
```

```

        outputs['message'] = responseBody.result[0].error_message;
        outputs['incident_number'] = 'No record is created, Ignored by script';
    }

```

```

    })(inputs, outputs);

```

you can pass the output variable(customized response)

Output Variables

Label	Name	Type	Mandatory
:: Status	status	String	<input type="checkbox"/>
:: Incident number	incident_number	String	<input type="checkbox"/>
:: Message	message	String	<input type="checkbox"/>

[+ Create Variable](#)

Generate the refresh token using schedule job:

To generate the refresh token using schedule job

step 1 : create a schedule jon with repeat interval of 90 days

Name: generate refresh token

Active: ☒

Application: Global












Conditional: ☐

Scheduled job types that require an entered time, you have the option to enter an associated time zone. If no time zone is selected, the time zone of the instance running the job.

Run: Periodically

* Repeat Interval: Days 90 Hours 00

Starting (yyyy-MM-dd HH:mm:ss): 2021-08-22 11:26:31

Run this script:           

```

1 new TestDemoUtils().generateRefreshToken();

```

step 2: create a scriptInclude utils to generate and store it to the system properties

demo_auth_server is Oauth credentials

```
generateRefreshToken: function(){  
  
    var oauthCli = new GlideOAuthClient();  
    var parameters = {grant_type:"password", username: "demo_oauth", password:"demo_oauth"};  
    var jsonParams = new JSON();  
  
    var tokenResp = oauthCli.requestToken('demo_oauth_server', jsonParams.encode(parameters));  
    var tokens = tokenResp.getToken();  
  
    gs.setProperty('snow.integ.refreshToken', tokens.getRefreshToken());  
  
},
```

step 3: create a another function in scriptInclude to get the access token from the refresh token

```
getDemoAccessToken: function(refreshToken){  
  
    var cliReq = new GlideOAuthClientRequest();  
    cliReq.setGrantType('refresh_token');  
    cliReq.setRefreshToken(refreshToken);  
    cliReq.setScope(null);  
  
    var oauthCli = new GlideOAuthClient();  
    var tokResp = oauthCli.requestToken('demo_oauth_server', cliReq);  
    var tokens = tokResp.getToken();  
  
    return tokens.getAccessToken();  
}
```

step 4: this access token can be passed into the Business rule

Name [?] send incident to instance B Application [?] Global

Table [?] Incident [incident] Active ☒

Advanced [?] ☒

Condition

Script [?]

```

1 (function executeRule(current, previous /*null when async*/) {
2
3     var demoAccToken = new TestDemoUtils().getDemoAccessToken(gs.getProperty('snow.integ.refreshToken'));
4
5     try {
6         var r = new sn_ws.RESTMessageV2('demo oauth msg', 'create incident');
7         r.setStringParameterNoEscape('access_token', demoAccToken);
8
9         var response = r.execute();
10        var responseBody = response.getBody();
11        var httpStatus = response.getStatusCode();
12
13        gs.addInfoMessage(responseBody);
14    }
15    catch(ex) {
16        var message = ex.message;
17    }
18
19 })(current, previous);

```

step 5: create a REST message and for getting access token, use an variable to store it \${access_token}

REST Message [?] demo oauth msg Application [?] Global

* Name [?] create incident

* HTTP method [?] POST

Endpoint [?] https://dev81442.service-now.com/api/now/table/incident

Authentication HTTP Request

Use MID Server [?]

HTTP Headers

Name	Value
Authorization	Bearer \${access_token}
content-type	application/json
Insert a new row...	

Creating incident to other instance using integration hub script action : [Two way to configure OAuth 2.0 -Outbound Rest inte... - ServiceNow Community](#)

Generating access token and refresh token:

For generating access token you can do in 2 way

first way : This is only for getting refresh token

POST ▼ https://fiatchryslerlertest.service-now.com/oauth_token.do

Params Authorization Headers (9) **Body** ● Pre-request Script Tests Settings

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> grant_type	password	
<input checked="" type="checkbox"/> client_id	4f3b9932d5c36d10a24cf1a58a7e502c	
<input checked="" type="checkbox"/> client_secret	~b8,.***?9p	
<input checked="" type="checkbox"/> username	Integration.AWSLZAccount.API	
<input checked="" type="checkbox"/> password	ak6feESF7IXu8Rn4*\$npVm#S6uOisoX&	
Key	Value	Description

Second way :

with the help of **client ID, Client secret and refresh token** you can able to generate the access token(This is need to be given to stakeholders)

Refer this community link to generate Access token with the help of ClientID, Secret and refresh token :

<https://www.servicenow.com/community/developer-blog/oauth-2-0-with-inbound-rest/ba-p/2278926>

Link to API reference :

https://developer.servicenow.com/dev.do#!/reference/api/sandiego/server/sn_ws-namespaces/c_RESTMessageV2API

converting JSON to escaped JSON : <https://www.lambdatest.com/free-online-tools/json-escape>

HTTP Method :

POST - post the data from drive it to AWS

GET - get the data from AWS to Drive it

PATCH - updating the record

DELETE

task var - sn_sc -> update variable(PUT) --> it will update th hidden API
result from AWS

task - now -> table API ->patch(update a record) --> it needs to change the
state and put worknotes