**Bi-Directional Integration between 2 SNOW instances using REST method.**

**Requirement:**

When an Incident is created in one instance, it should be transferred to another instance. Similarly, if the same incident is updated, the change should reflect in another instance record.

Instance 1: <https://dev280591.service-now.com/>

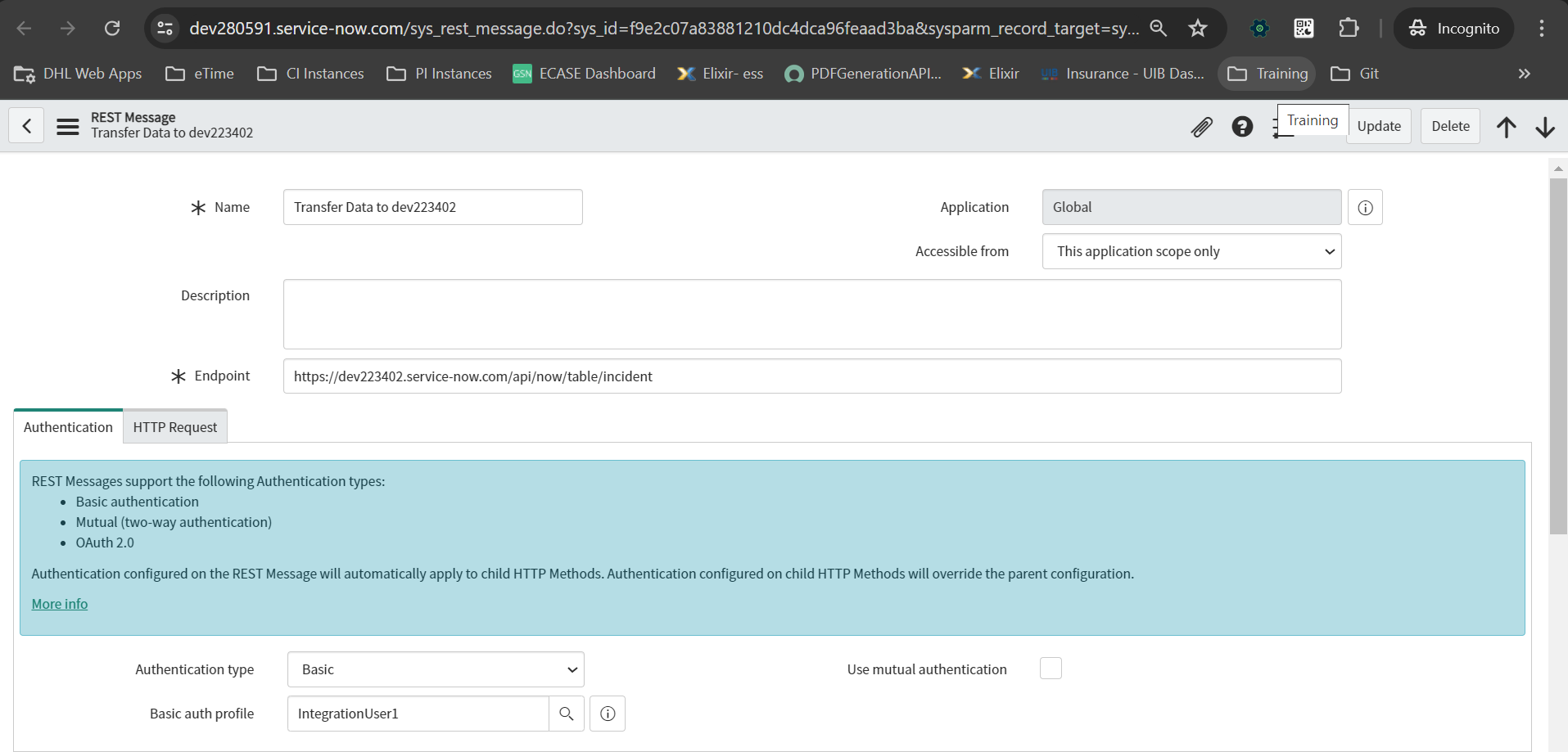
Instance 2: <https://dev223402.service-now.com/>

**Procedure:**

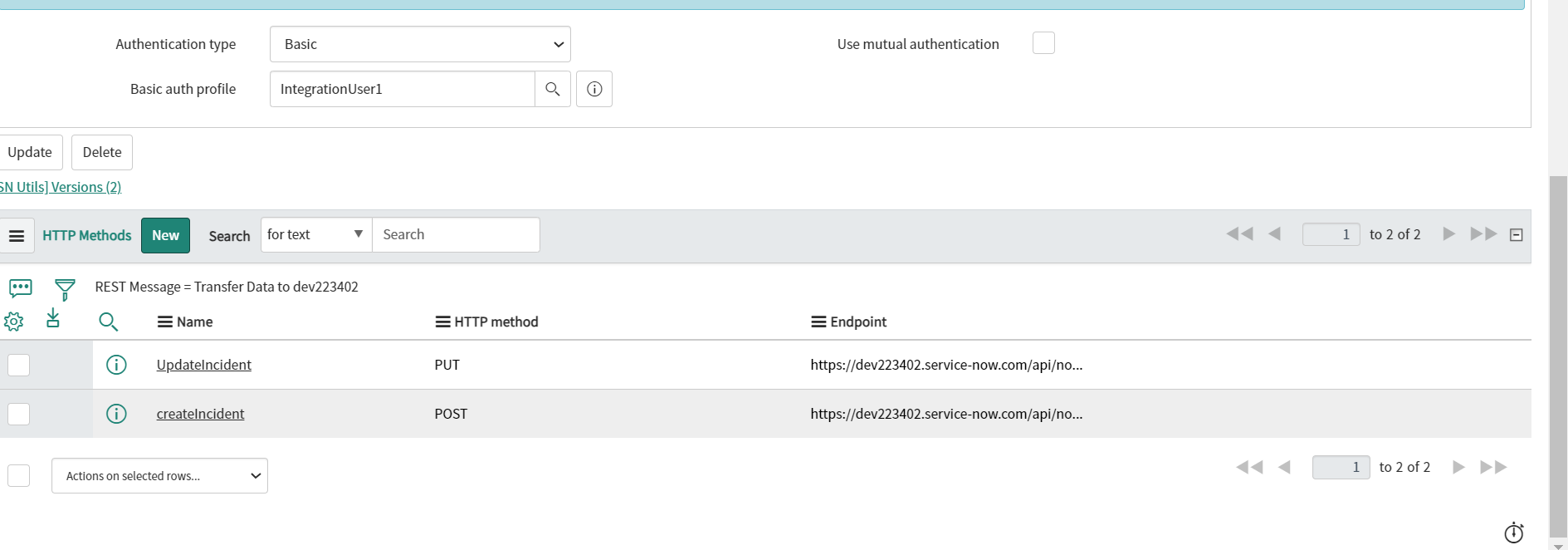
1. Create a User “Integration User1” in dev223402 and assign “rest” related roles.
2. Goto Rest API Explorer 🡪 Select “POST” 🡪 Select “Incident” 🡪 Build a Request.
3. Copy the HTTP method.

<https://dev223402.service-now.com/api/now/table/incident>

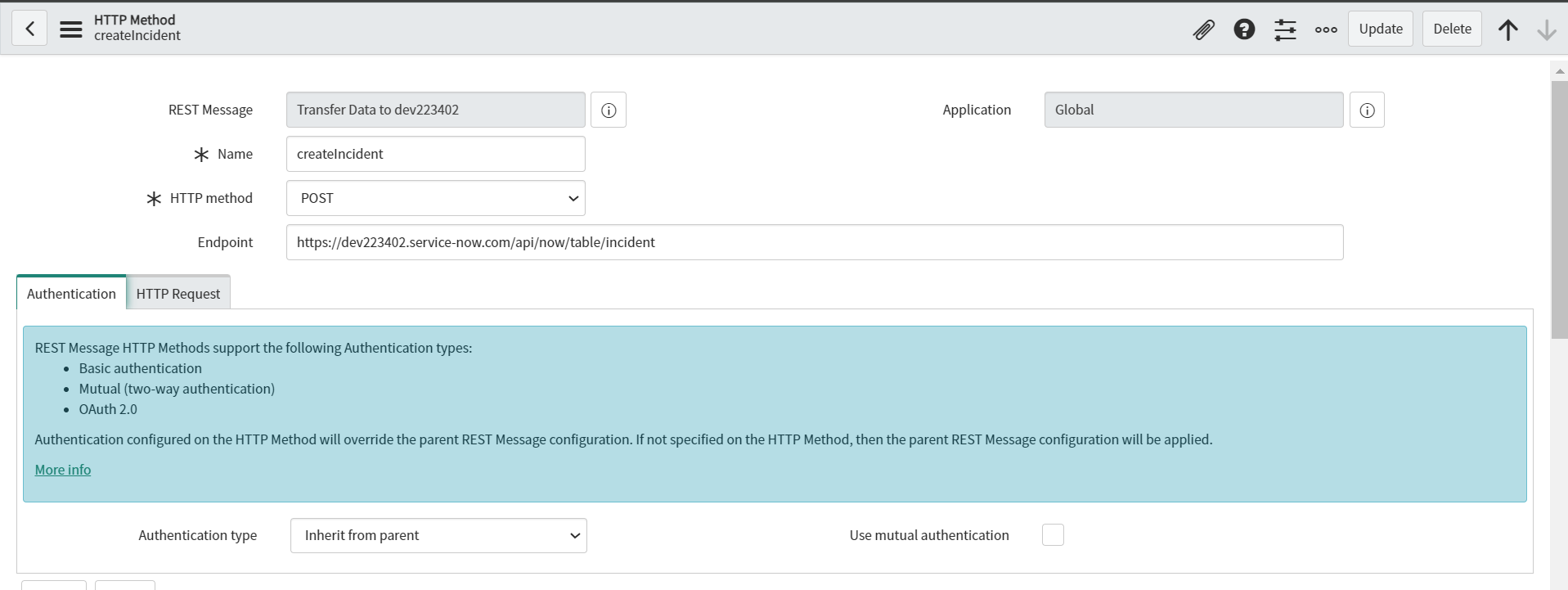
1. Now, Goto First instance 🡪 Open “Rest Message” (Out Bound) 🡪 Click on New.
2. **Enter** the Name & Endpoint.
3. Select Authentication Type 🡪 Basic.
4. Then Select the user which is configured like Integration User1 🡪 Save.



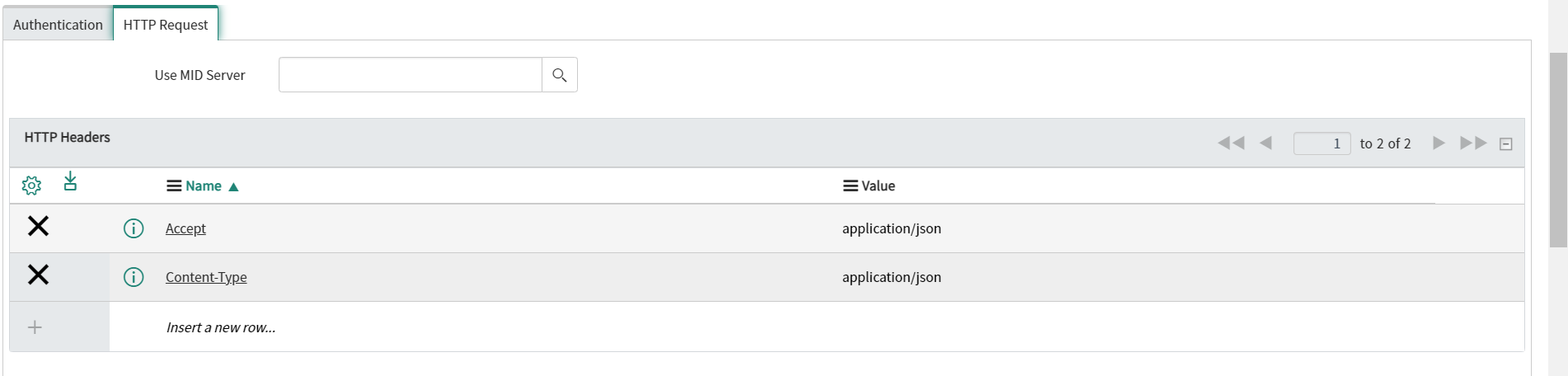
1. Goto Related records 🡪 HTTP Methods 🡪 Click New



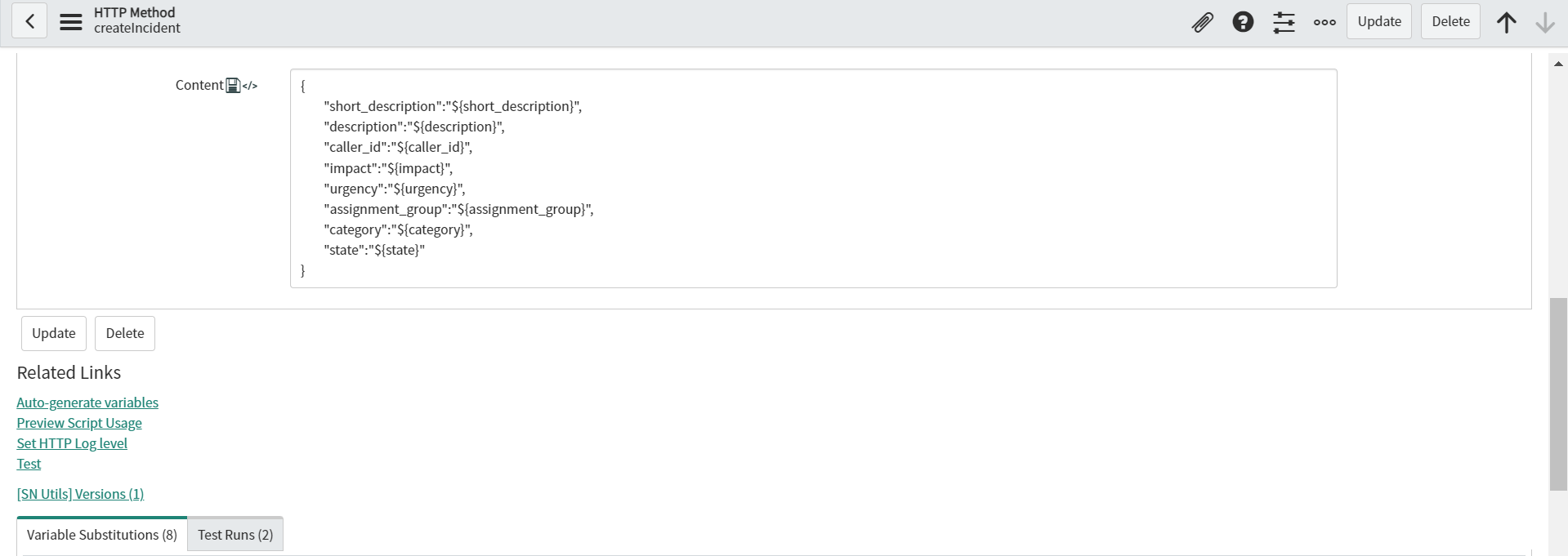
1. Provide the Name, Select HTTP method as “POST” and Endpoint.
2. Authentication 🡪 Inherit from Parent



1. HTTP Request 🡪 HTTP Headers
2. Accept = application/json & Content-Type = application/json



1. Input the below content 🡪 Click on Test to validate.



{

"short\_description":"${short\_description}",

"description":"${description}",

"caller\_id":"${caller\_id}",

"impact":"${impact}",

"urgency":"${urgency}",

"assignment\_group":"${assignment\_group}",

"category":"${category}",

"state":"${state}"

}

1. Now, Goto Business Rules 🡪 Call the Rest message from business rules on Insert a incident record.
2. When to Run 🡪 Asyn && Insert
3. Script section.

(function executeRule(current, previous /\*null when async\*/ ) {

try {

var r = new sn\_ws.RESTMessageV2('Transfer Data to dev223402', 'createIncident');

r.setStringParameterNoEscape('description', current.description);

r.setStringParameterNoEscape('category', current.category);

r.setStringParameterNoEscape('assignment\_group', current.assignment\_group);

r.setStringParameterNoEscape('caller\_id', current.caller\_id);

r.setStringParameterNoEscape('urgency', current.urgency);

r.setStringParameterNoEscape('impact', current.impact);

r.setStringParameterNoEscape('short\_description', current.short\_description);

r.setStringParameterNoEscape('state', current.state);

var response = r.execute();

var responseBody = response.getBody();

var httpStatus = response.getStatusCode();

//Update the Correlation ID field with incident Number from another instance

var obj = JSON.parse(responseBody);

current.correlation\_id = obj.result.sys\_id;

current.update();

} catch (ex) {

var message = ex.message;

}

})(current, previous);

1. Now Create an Incident and check if this data created on second instance.

**\*\*\*\*POST Method Completed\*\*\*\*\***

**// PUT Method Starts – Update the record.**

1. Create one more HTTP Method – PUT
2. End-point: <https://dev223402.service-now.com/api/now/table/incident/>

**Note: PUT method will be processed based on “SYS\_ID” of the Target record.**

**Hence, we should overwrite the above Endpoint.**

1. Use the same HTTP Headers & Content as POST method.
2. Create a BR 🡪 Asyc && Update <Condition> = Correlation ID is not empty

(function executeRule(current, previous /\*null when async\*/ ) {

try {

var r = new sn\_ws.RESTMessageV2('Transfer Data to dev223402', 'UpdateIncident');

r.setEndpoint('https://dev223402.service-now.com/api/now/table/incident/'+current.correlation\_id);

r.setStringParameterNoEscape('desc', current.description);

r.setStringParameterNoEscape('category', current.category);

r.setStringParameterNoEscape('assignment\_group', current.assignment\_group);

r.setStringParameterNoEscape('caller\_id', current.caller\_id);

r.setStringParameterNoEscape('urgency', current.urgency);

r.setStringParameterNoEscape('impact', current.impact);

r.setStringParameterNoEscape('short\_description', current.short\_description);

r.setStringParameterNoEscape('state', current.state);

var response = r.execute();

var responseBody = response.getBody();

var httpStatus = response.getStatusCode();

} catch (ex) {

var message = ex.message;

}

})(current, previous);

**The below query dynamically passes the target record sys\_id**

**r.setEndpoint('https://dev223402.service-now.com/api/now/table/incident/'+current.correlation\_id);**

1. **Now update the incident record in Source instance and validate the details in target instance.**

**\*\*\*\*\*\*\*\*\*\*\*\* One – Directional Integration is completed. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Now, Let’s configure the Bi-Directional Integration**

1. Create Integration.User2 in dev280591 (another instance) and assign “rest” roles.

2. Goto Rest API Explorer 🡪 Select “POST” 🡪 Select “Incident” 🡪 Build a Request.

3. Copy the HTTP method.

<https://dev280591.service-now.com/api/now/table/incident>

4. Now, Goto First instance Open “Rest Message” (Out Bound) Click on New.

5. Enterthe Name & Endpoint.

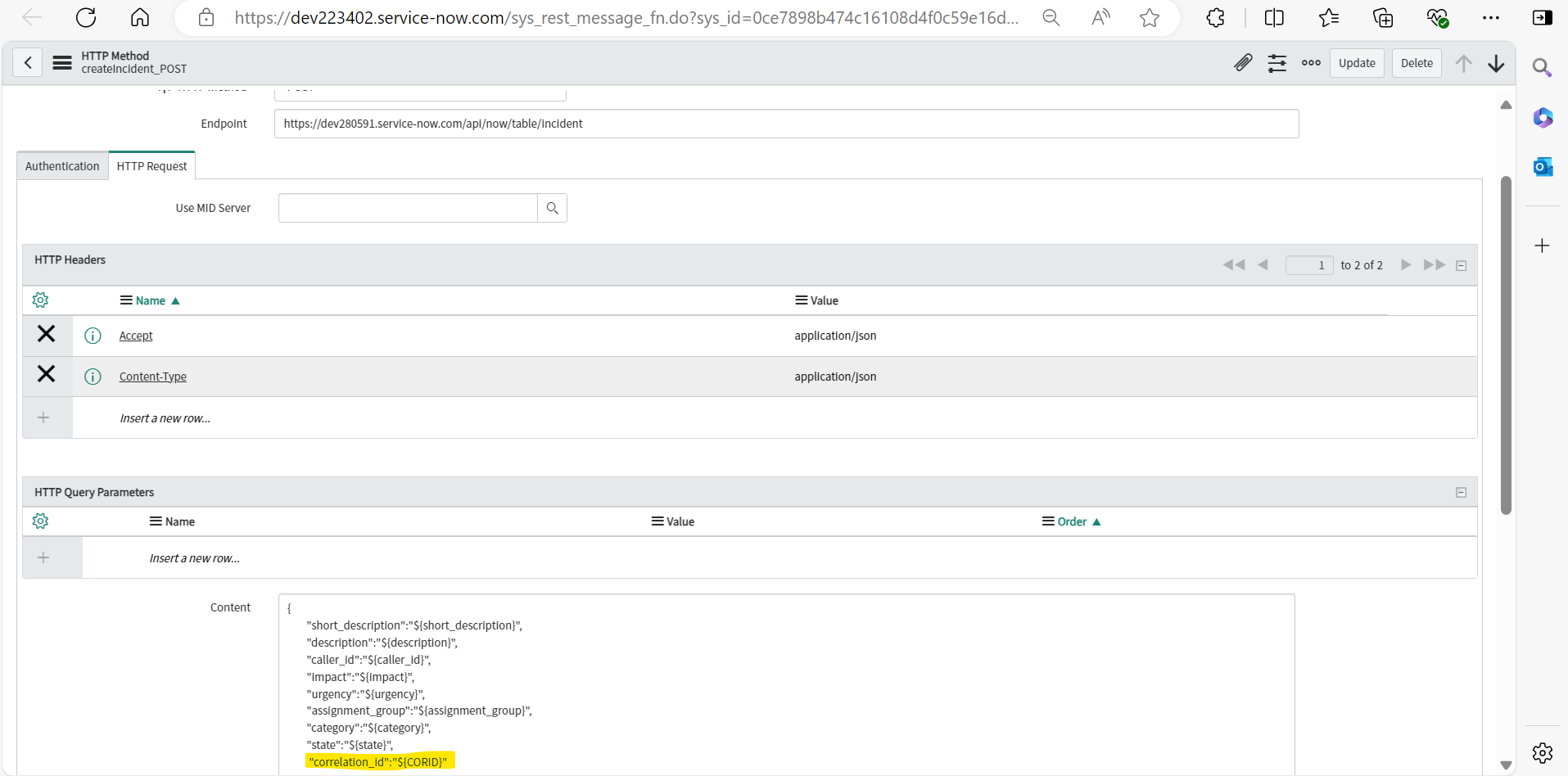
6. Select Authentication Type Basic.

7. Then Select the user which is configured like Integration User2 Save.

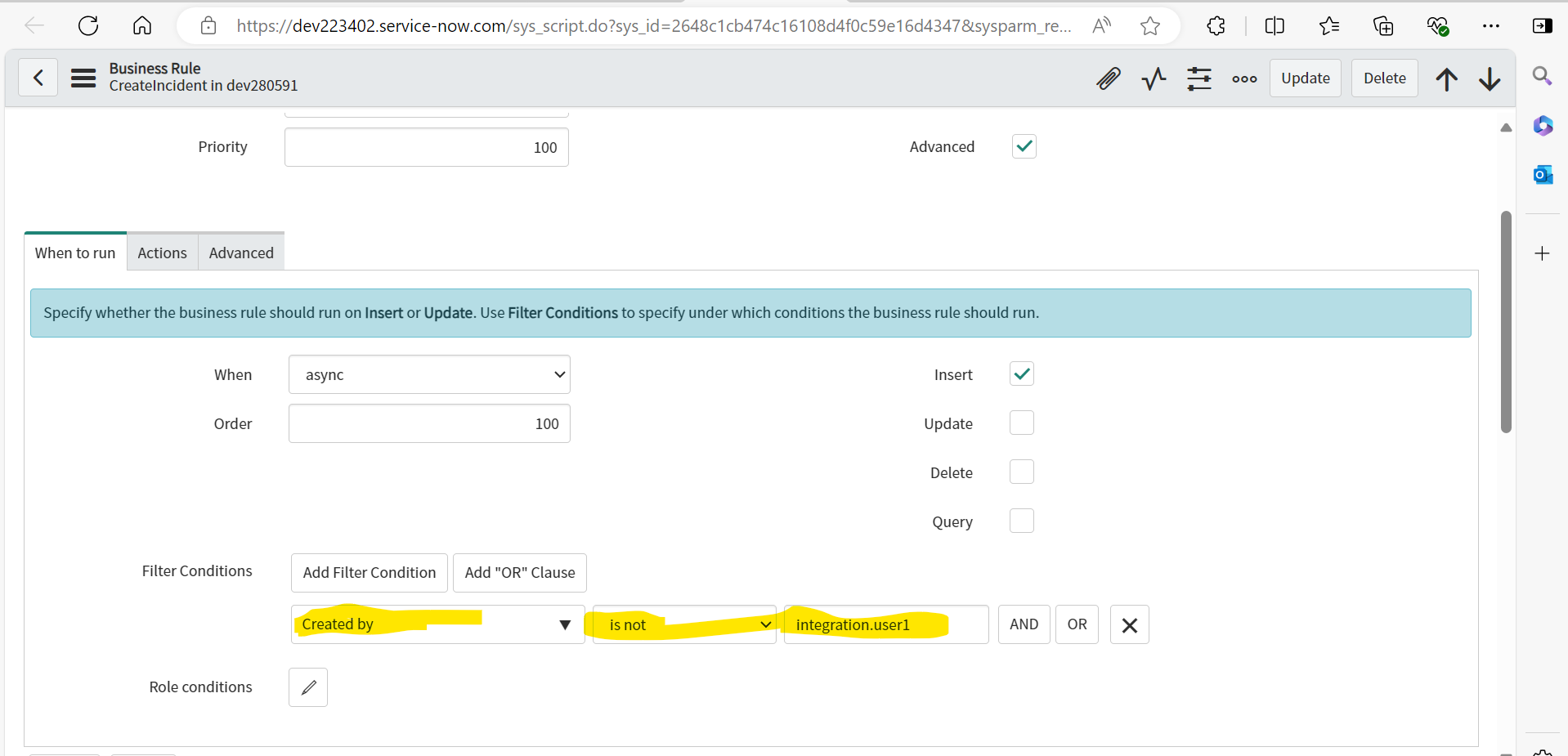
8. Create HTTP Method for “POST”.

**Important:**

**CORRELATION ID is populated with another instance “Incident” SYSID” to use this as dynamic data to “Update” an incident.**



9. Create a business rule.



**Condition:** Created by = is not = integration.user1

Or Add any condition which will stop “Continuous” call between instances.

(function executeRule(current, previous /\*null when async\*/ ) {

try {

var r = new sn\_ws.RESTMessageV2('Transfer Data to dev280591', 'createIncident\_POST');

r.setStringParameterNoEscape('description', current.description);

r.setStringParameterNoEscape('category', current.category);

r.setStringParameterNoEscape('assignment\_group',current.assignment\_group);

r.setStringParameterNoEscape('caller\_id', current.caller\_id);

r.setStringParameterNoEscape('urgency', current.urgency);

r.setStringParameterNoEscape('impact', current.impact);

r.setStringParameterNoEscape('short\_description', current.short\_description);

r.setStringParameterNoEscape('state', current.state);

r.setStringParameterNoEscape('CORID', current.sys\_id);

var response = r.execute();

var responseBody = response.getBody();

var httpStatus = response.getStatusCode();

var obj = JSON.parse(responseBody);

current.correlation\_id = obj.result.sys\_id;

current.update();

} catch (ex) {

var message = ex.message;

}

})(current, previous);

Similarly Create and “PUT” HTTP method to Update an instance.

Business rule:

(function executeRule(current, previous /\*null when async\*/ ) {

try {

var r = new sn\_ws.RESTMessageV2('Transfer Data to dev280591', 'UpdateIncident');

r.setEndpoint("https://dev280591.service-now.com/api/now/table/incident/" + current.correlation\_id);

r.setStringParameterNoEscape('state', current.state);

r.setStringParameterNoEscape('short\_description', current.short\_description);

r.setStringParameterNoEscape('description', current.description);

r.setStringParameterNoEscape('caller\_id', current.caller\_id);

r.setStringParameterNoEscape('impact', current.impact);

r.setStringParameterNoEscape('urgency', current.urgency);

r.setStringParameterNoEscape('assignment\_group', current.assignment\_group);

r.setStringParameterNoEscape('category', current.category);

var response = r.execute();

var responseBody = response.getBody();

var httpStatus = response.getStatusCode();

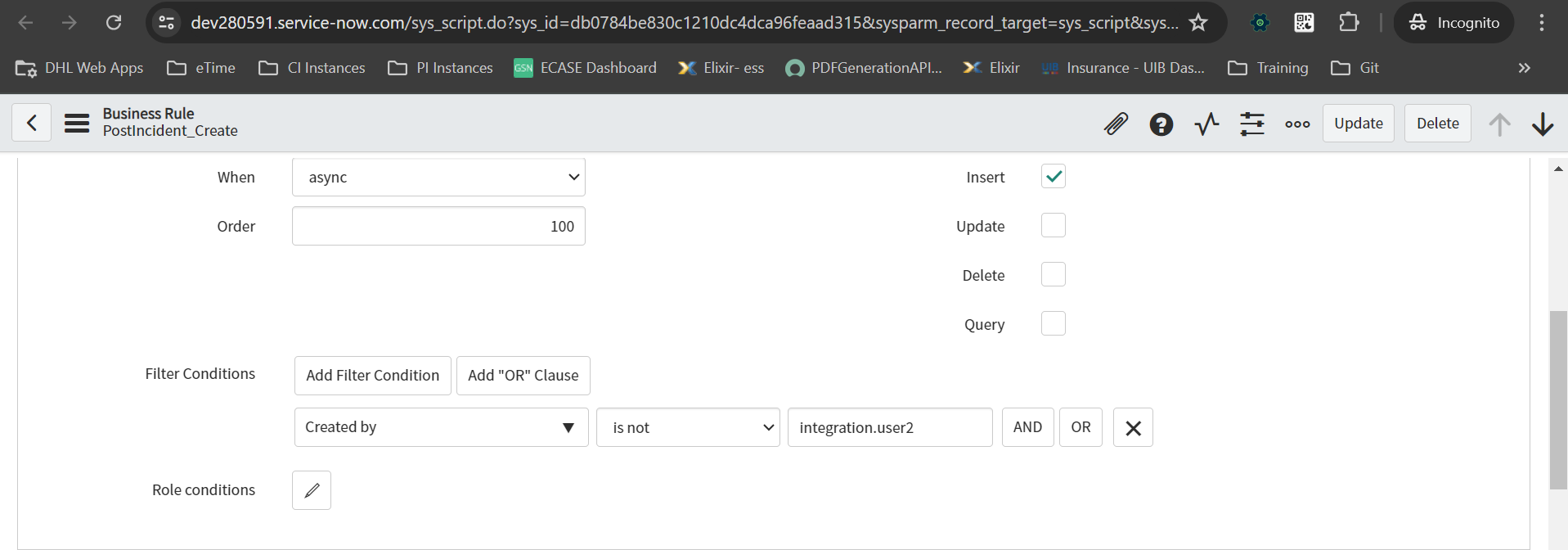
} catch (ex) {

var message = ex.message;

}

})(current, previous);

Now, we need add same condition in business rule for instance 1:



(function executeRule(current, previous /\*null when async\*/ ) {

try {

var r = new sn\_ws.RESTMessageV2('Transfer Data to dev223402', 'createIncident');

r.setStringParameterNoEscape('description', current.description);

r.setStringParameterNoEscape('category', current.category);

r.setStringParameterNoEscape('assignment\_group', current.assignment\_group);

r.setStringParameterNoEscape('caller\_id', current.caller\_id);

r.setStringParameterNoEscape('urgency', current.urgency);

r.setStringParameterNoEscape('impact', current.impact);

r.setStringParameterNoEscape('short\_description', current.short\_description);

r.setStringParameterNoEscape('state', current.state);

r.setStringParameterNoEscape('COR\_ID', current.sys\_id);

var response = r.execute();

var responseBody = response.getBody();

var httpStatus = response.getStatusCode();

var obj = JSON.parse(responseBody);

current.correlation\_id = obj.result.sys\_id;

current.update();

} catch (ex) {

var message = ex.message;

}

})(current, previous);

**POST Content:**

{

"short\_description":"${short\_description}",

"description":"${description}",

"caller\_id":"${caller\_id}",

"impact":"${impact}",

"urgency":"${urgency}",

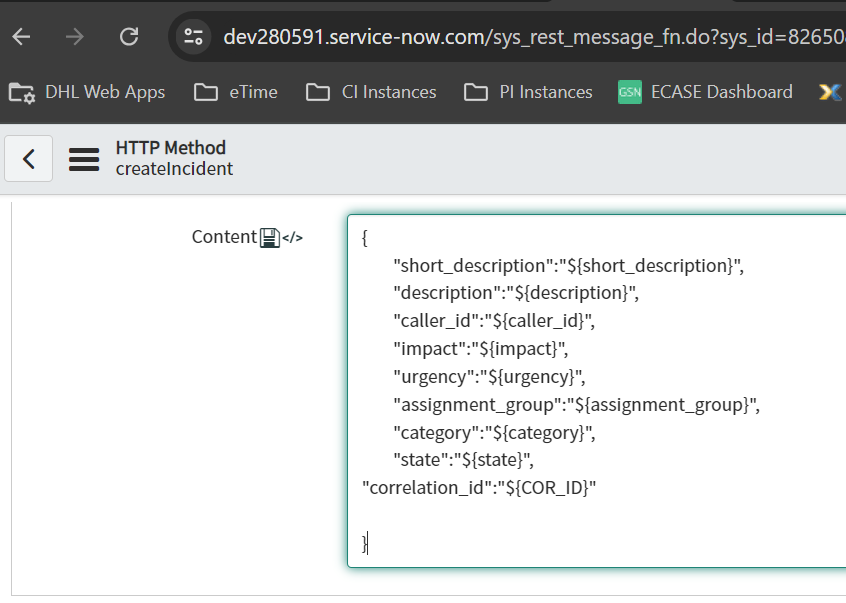
"assignment\_group":"${assignment\_group}",

"category":"${category}",

"state":"${state}",

"correlation\_id":"${COR\_ID}"

}



**//DELETE – HTTP Method – Its throwing “ACL issues” and not letting to Delete the records.**

**Response Body:**

**{"error":{"message":"Operation Failed","detail":"ACL Exception Delete Failed due to security constraints"},"status":"failure"}**

**Update Sets:**



