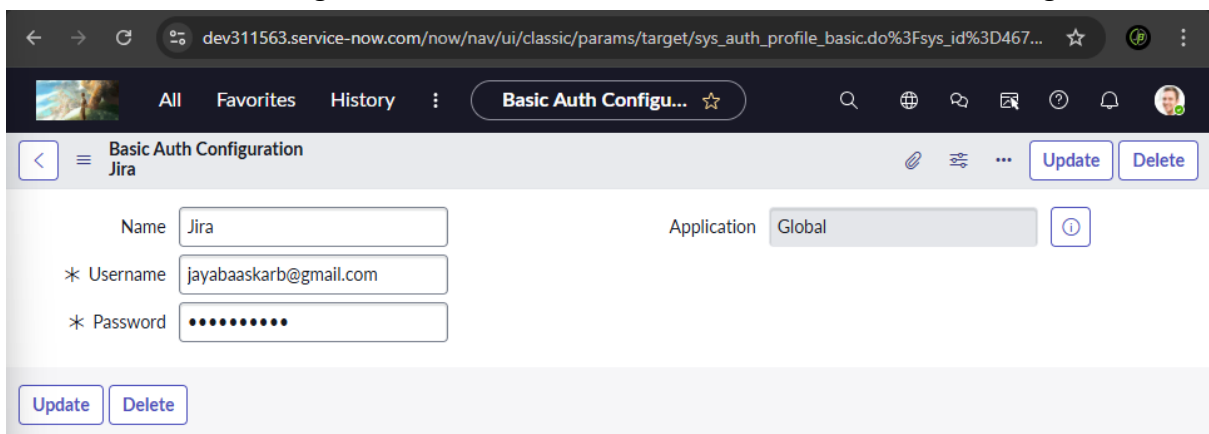


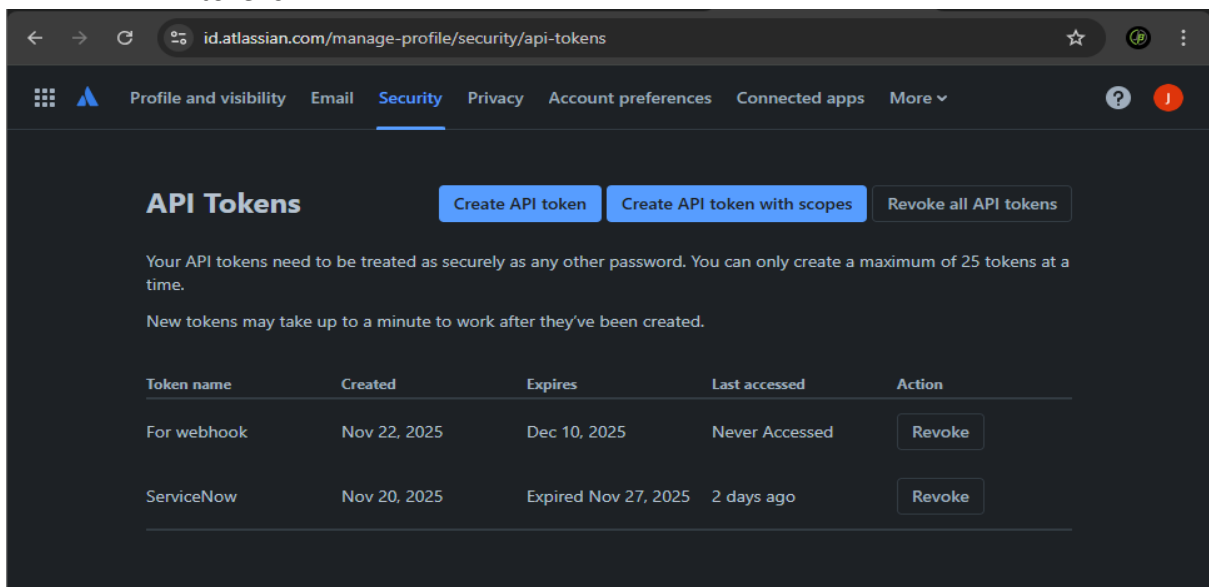
## Jira Integration:

- Create REST Bi - Directional integration with JIRA by using Rest API with incident management in serviceNow and Issue in Jira.
- Outbound
  - Create a Basic Auth Configuration with Jira Credentials and Api Token
  - All > Integration Hub Credentials > Create New > Basic Auth Config.



The screenshot shows the 'Basic Auth Configuration' page in ServiceNow. The browser address bar displays 'dev311563.service-now.com/now/nav/ui/classic/params/target/sys\_auth\_profile\_basic.do%3Fsys\_id%3D467...'. The page title is 'Basic Auth Configuration Jira'. The form includes fields for 'Name' (Jira), 'Application' (Global), '\* Username' (jayabaaskarb@gmail.com), and '\* Password' (masked with dots). There are 'Update' and 'Delete' buttons at the bottom left and top right of the form area.

- Api Token in Jira Nav: Account Settings > Security > Create or Manage Api tokens.



The screenshot shows the 'API Tokens' page in Atlassian. The browser address bar displays 'id.atlassian.com/manage-profile/security/api-tokens'. The page has a navigation bar with 'Profile and visibility', 'Email', 'Security' (active), 'Privacy', 'Account preferences', 'Connected apps', and 'More'. Below the navigation bar, there are buttons for 'Create API token', 'Create API token with scopes', and 'Revoke all API tokens'. A message states: 'Your API tokens need to be treated as securely as any other password. You can only create a maximum of 25 tokens at a time. New tokens may take up to a minute to work after they've been created.' Below this is a table with columns: 'Token name', 'Created', 'Expires', 'Last accessed', and 'Action'.

Token name	Created	Expires	Last accessed	Action
For webhook	Nov 22, 2025	Dec 10, 2025	Never Accessed	Revoke
ServiceNow	Nov 20, 2025	Expired Nov 27, 2025	2 days ago	Revoke

- In Servicenow Go to All > Rest Message > Create New
- Fill in Name and End point as Target Instance url of Jira, i.e, <https://jayabaaskarb.atlassian.net/>
- Select Authentication Type as Required,(Basic Here)
- Choose the Basic Auth Profile created Before.
- Save and Submit

dev311563.service-now.com/now/nav/ui/classic/params/target/sys\_rest\_message.do%3Fsys\_id%3D1073748...

REST Message d- Ji...

REST Message JiralIntegration

\* Name JiralIntegration Application Global

Accessible from This application scope only

Description

\* Endpoint https://jayabaaskarb.atlassian.net/

Authentication HTTP Request

REST Messages support the following Authentication types:

- Basic authentication
- Mutual (two-way authentication)
- OAuth 2.0

Authentication configured on the REST Message will automatically apply to child HTTP Methods. Authentication configured on child HTTP Methods will override the parent configuration.

[More info](#)

Authentication type Basic

Use mutual authentication ☐

Basic auth profile Jira

Update Delete

- Create HTTP Methods for the Rest message with correct http method and endpoint and content using jira documentation

HTTP Methods

REST Message = JiralIntegration

Name	HTTP method	Endpoint
Create	POST	https://jayabaaskarb.atlassian.net/rest/...
Delete	DELETE	https://jayabaaskarb.atlassian.net/rest/...
Receive	GET	https://jayabaaskarb.atlassian.net/rest/...
Update	PUT	https://jayabaaskarb.atlassian.net/rest/...

1 to 4 of 4

- Post method -> Create with proper headers and required Content

REST Message: JiraIntegration

Application: Global

\* Name: Create

\* HTTP method: POST

Endpoint: https://jayabaaskarb.atlassian.net/rest/api/3/issue

Authentication: HTTP Request

Use MID Server: [Search]

HTTP Headers

Name	Value
Accept	application/json
Content-Type	application/json
Insert a new row...	

HTTP Query Parameters

Name	Value	Order
Insert a new row...		

Content

```
{
  "fields": {
    "project": {
      "key": "KAN"
    },
    "issuetype": {
      "name": "Task"
    },
    "summary": "${sd}",
    "description": {
      "type": "doc",
      "version": 1,
    }
  }
}
```

- Test the content by giving Test Values

Variable Substitutions (2) Test Runs (8)

Method = Create

Name	Escape type	Test value
des	No escaping	10
sd	No escaping	20

1 to 2 of 2

- Http status 201 Indicates Proper Cnnection to Jira

The screenshot shows the 'Test Runs' interface in ServiceNow. The test is named 'Create' and was created on 2025-11-20 00:36:02. The HTTP status is 201, indicating a successful creation. The endpoint is `https://jayabaaskarb.atlassian.net/rest/api/3/issue`. The content field displays a JSON response from Jira, and the response field shows the full JSON response including the issue ID.

Test Runs  
Created 2025-11-20 00:36:02

Name: Create

HTTP status: 201

Endpoint: `https://jayabaaskarb.atlassian.net/rest/api/3/issue`

Parameters:

Content:

```
{
  "fields": {
    "project": {
      "key": "KAN"
    },
    "issuetype": {
      "name": "Task"
    },
    "summary": "20",
    "description": {
      "type": "doc",
      "version": 1,
      "content": [
        {
          "type": "paragraph",
          "content": [
            {
              "type": "text",
              "text": "10"
            }
          ]
        }
      ]
    }
  }
}
```

Response:

```
{
  "id": "10037",
  "key": "KAN-5",
  "self": "https://jayabaaskarb.atlassian.net/rest/api/3/issue/10037"
}
```

- Preview Script Usage

The screenshot shows the 'HTTP Method' configuration interface in ServiceNow. A dialog box titled 'Preview REST Message script usage' is open, displaying a JavaScript script for creating a REST message. The script includes comments for overriding authentication, setting a MID server, and handling ECC queue communication. Below the script, there are 'Update' and 'Delete' buttons, and a 'Related Links' section with links to 'Auto-generate variables', 'Preview Script Usage', 'Set HTTP Log level', and 'Test'.

HTTP Method  
Create

Preview REST Message script usage

```
try {
  var r = new sn_ws.RESTMessageV2('JiraIntegration', 'Create');
  r.setStringParameterNoEscape('des', '10');
  r.setStringParameterNoEscape('sd', '20');

  //override authentication profile
  //authentication type = 'basic' / 'oauth2'
  //r.setAuthenticationProfile(authentication type, profile name);

  //set a MID server name if one wants to run the message on MID
  //r.setMIDServer('MY_MID_SERVER');

  //if the message is configured to communicate through ECC queue, either
  //by setting a MID server or calling executeAsync, one needs to set skip_sensor
  //to true. Otherwise, one may get an intermittent error that the response body
  //r.setEccParameter('skip_sensor', true);

  var response = r.execute();
  var responseBody = response.getBody();
  var httpStatus = response.getStatusCode();
} catch(ex) {
  var message = ex.message;
}
```

Update Delete

Related Links

- [Auto-generate variables](#)
- [Preview Script Usage](#)
- [Set HTTP Log level](#)
- [Test](#)

Variable Substitutions (2)

- Copy the Script to Create a Business rule that runs After – Insert,Update,Delete in incident table

The screenshot shows the 'Business Rule d- In...' configuration page in ServiceNow. The 'Name' is 'IntegrationJira BR' and the 'Table' is 'Incident [incident]'. The 'Application' is set to 'Global'. The 'Active' checkbox is unchecked, and the 'Advanced' checkbox is checked. Below the configuration fields, there is a 'When to run' section with tabs for 'When to run', 'Actions', and 'Advanced'. The 'When to run' tab is active, showing a dropdown for 'When' set to 'after' and an 'Order' of '100'. To the right, there are checkboxes for 'Insert' (checked), 'Update' (checked), 'Delete' (checked), and 'Query' (unchecked). Below these are 'Filter Conditions' and 'Role conditions' sections, each with an 'Add Filter Condition' or 'Add OR Clause' button and a dropdown menu for field selection.

- Paste the Copied Preview script and Point Variables to the Incident table Fields.(Create a custom field to populate Jira issue id for future use).

The screenshot shows the 'Script' editor in ServiceNow. The code is a JavaScript function named 'executeRule' that takes 'current' and 'previous' as arguments. It checks the 'current.operation()' and performs different actions based on the operation type: 'insert', 'update', or 'delete'. For 'insert', it creates a REST message to Jira, sets parameters for 'des' (description) and 'sd' (short description), and updates the 'u\_jira\_field' with the response key. For 'update' and 'delete', the code is partially visible and appears to be copied from a previous version.

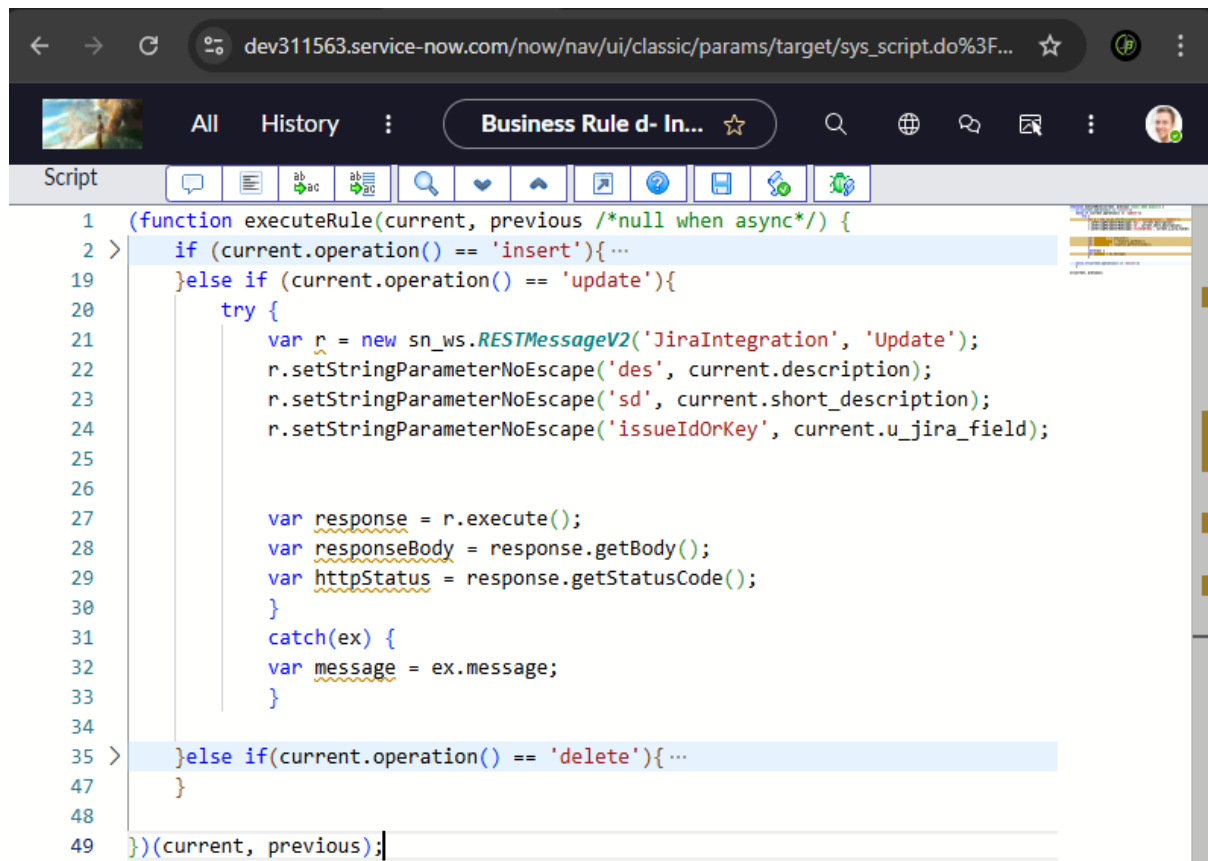
```

1 (function executeRule(current, previous /*null when async*/) {
2   if (current.operation() == 'insert'){
3     try {
4       var r = new sn_ws.RESTMessageV2('JiraIntegration', 'Create');
5       r.setStringParameterNoEscape('des', current.description);
6       r.setStringParameterNoEscape('sd', current.short_description);
7
8
9       var response = r.execute();
10      var responseBody = response.getBody();
11      var res = JSON.parse(responseBody);
12      var httpStatus = response.getStatusCode();
13      current.u_jira_field = res.key;
14      current.update();
15    }
16    catch(ex) {
17      var message = ex.message;
18    }
19  } else if (current.operation() == 'update'){ ...
35  } else if (current.operation() == 'delete'){ ...
47  }
48
49 })(current, previous);

```

- Repeat this process for Update and Delete.
- Note that the endpoint suffix may vary for each http method refer Jira documentation

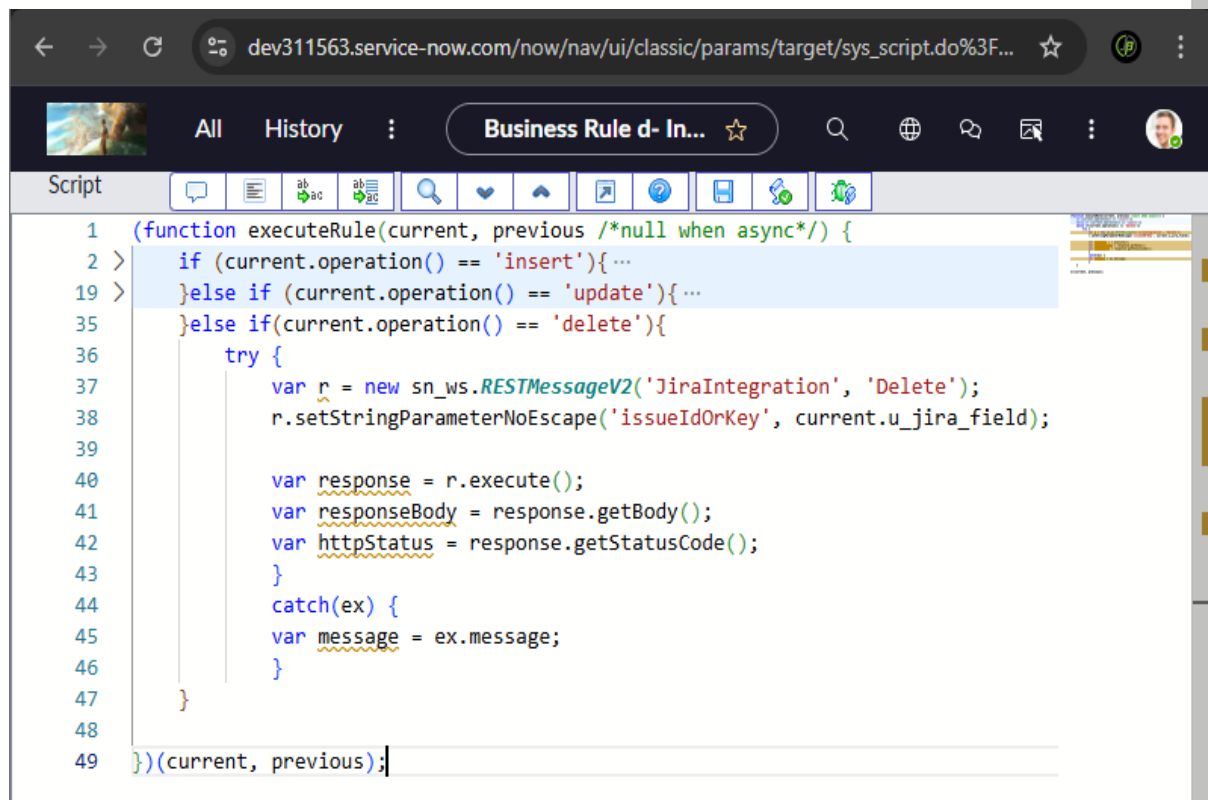
<https://developer.atlassian.com/cloud/jira/platform/rest/v3/intro/#authentication>



```

1 (function executeRule(current, previous /*null when async*/) {
2 >   if (current.operation() == 'insert'){ ...
19   }else if (current.operation() == 'update'){
20     try {
21       var r = new sn_ws.RESTMessageV2('JiraIntegration', 'Update');
22       r.setStringParameterNoEscape('des', current.description);
23       r.setStringParameterNoEscape('sd', current.short_description);
24       r.setStringParameterNoEscape('issueIdOrKey', current.u_jira_field);
25
26
27       var response = r.execute();
28       var responseBody = response.getBody();
29       var httpStatus = response.getStatusCode();
30     }
31     catch(ex) {
32       var message = ex.message;
33     }
34
35 >   }else if(current.operation() == 'delete'){ ...
47   }
48
49 })(current, previous);

```



```

1 (function executeRule(current, previous /*null when async*/) {
2 >   if (current.operation() == 'insert'){ ...
19 >   }else if (current.operation() == 'update'){ ...
35   }else if(current.operation() == 'delete'){
36     try {
37       var r = new sn_ws.RESTMessageV2('JiraIntegration', 'Delete');
38       r.setStringParameterNoEscape('issueIdOrKey', current.u_jira_field);
39
40       var response = r.execute();
41       var responseBody = response.getBody();
42       var httpStatus = response.getStatusCode();
43     }
44     catch(ex) {
45       var message = ex.message;
46     }
47   }
48
49 })(current, previous);

```

- Inbound – From Jira to ServiceNow

- Go to Scripted Rest APIs in ServiceNow > Create New

The screenshot shows the 'Scripted REST API' configuration page in ServiceNow. The browser address bar shows the URL: `dev311563.service-now.com/now/nav/ui/classic/params/target/sys_ws_definition...`. The page title is 'Scripted REST API ...'. The configuration fields are as follows:

- Name:** MyScriptedRest
- API ID:** myscriptedrest
- Active:** ☒
- Protection policy:** -- None --
- Application:** Global
- API namespace:** 1853367
- Base API path:** /api/1853367/myscriptedrest

Below the configuration fields, there are tabs for 'Security', 'Content Negotiation', and 'Documentation'. The 'Security' tab is selected, showing a blue information box with the following text:

Default ACLs may be selected to apply to all resources, but individual resources can override this setting. The Default ACLs are enforced for a resource when:

- The resource 'Requires authentication' and 'Requires ACL authorization' fields are selected, and
- The resource itself does not reference any ACL records

Access is granted if at least one matching ACL record is found. [More info](#)

Below the information box, there is a field for 'Default ACLs' with a lock icon and the text 'Scripted REST External Default'. At the bottom of the 'Security' tab, there are 'Update' and 'Delete' buttons.

- Create new Resources

The screenshot shows the 'Resources' tab in the ServiceNow configuration page. The page title is 'Resources (3)'. The tabs are 'Resources (3)', 'Request Headers', and 'Query Parameters'. The 'Resources (3)' tab is selected, showing a table of resources. The table has columns: 'Name', 'HTTP method', 'Relative path', 'Resource path', and 'API version'. The table contains three rows of resources:

Name	HTTP method	Relative path	Resource path	API version
Delete	DELETE	/delete	/api/1853367/myscriptedrest/delete	(empty)
Post	POST	/post	/api/1853367/myscriptedrest/post	(empty)
Put	PUT	/put	/api/1853367/myscriptedrest/put	(empty)

At the bottom of the table, there is a pagination bar showing '1 to 3 of 3'.

- Select Appropriate http method and add relative path
- Disable Requires Authentication in Security

dev311563.service-now.com/now/nav/ui/classic/params/target/sys\_ws\_operation...

All History Scripted REST Reso...

Scripted REST Resource Post Update Delete

API definition MyScriptedRest Application Global

\* Name Post Active ☒

**Request routing**

The route configuration specifies the 'HTTP method' and 'Relative path'. These fields determine how HTTP clients access this resource.

The relative path identifies the sub-path to this resource relative to the base API path. The relative URI can contain path parameters such as '/abc/{id}'. The requesting client specifies the id value, available to the script at runtime via the: [Request API](#).

[More info](#)

\* HTTP method POST Relative path /post

Resource path /api/1853367/myscriptedrest/post

- Write the appropriate background script using rest commands and pull jira issue data to fill in Incident.

dev311563.service-now.com/now/nav/ui/classic/params/target/sys\_ws\_operation...

All History Scripted REST Reso...

Script

```

1 (function process(/*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {
2   var reqBody = request.body.data;
3   gs.log("jira body", reqBody);
4   var key = reqBody.issue.key;
5   var sum = reqBody.issue.fields.summary;
6   var desc = reqBody.issue.fields.description;
7
8
9   var gr = new GlideRecord('incident');
10  gr.addQuery('u_jiraid', key); gr.query();
11  if (!gr.next()){
12    gr.initialize();
13    gr.short_description = sum;
14    gr.u_jiraid = key;
15    gr.setWorkflow(false);
16    gr.insert();
17  } else {
18    gr.short_description = sum;
19    gr.description = desc;
20    gr.setWorkflow(false);
21    gr.update();
22  }
23 })(request, response);
24

```



- Go to Jira webhooks for integration
- Jira admin Settings > System > Advanced > Webhooks > create new
- Give Resource path as Suffix with the servicenow instance url serving as endpoint
- <https://dev311563.service-now.com/api/1853367/myscriptedrest/post>

The screenshot shows the Jira webhooks configuration interface. At the top, there's a cookie consent banner. Below it, a search bar and a '+ Create' button are visible. The main content area shows a list of webhooks with the filter 'All' selected. Two webhooks are found: 'ServiceNow webhook' and 'snow jira'. The 'snow jira' webhook is selected and its details are shown on the right. The details include the URL, secret, events, and transitions.

Webhook Name	Status	Last Updated	URL	Secret	Events	Transitions
ServiceNow webhook	DISABLED					
snow jira	DISABLED	23/Nov/25 12:18 PM by Jayabaaskar	<a href="https://dev311563.service-now.com/api/1853367/myscriptedrest/post">https://dev311563.service-now.com/api/1853367/myscriptedrest/post</a>	Not set	<b>Jira Software related events</b> <b>Board:</b> created  <b>Issue related events</b> <b>JQL:</b> All issues <b>Issue:</b> created, updated, deleted	No linked transitions.

- Whenever an incident is created,updated,deleted in servicenow same Operation will happen in the Jira issue table bidirectionally