Integration Package Documentation: Figaf ServiceNow Integration

Version: 1.0.1
Last Updated: 19.01.2026
)

Contents

Contents 2
1. Package Overview 3
2.Prerequisites 3
3. Integration Flows 4
3.1. Search ServiceNow Issues 4
3.1.1. Configuration: 4
3.1.2. Using the IFlow 5
3.2. Update ServiceNow tasks 6
3.2.1. Configuration: 6
3.2.2. Options: 7
3.2.3. Using the IFlow 9
3.3. Update Figaf Transport from ServiceNow 10
3.3.1. ServiceNow Setup 10
3.3.2. Configuration (Integration Suite): 12
4. Deployment and General Setup 14

1. Package Overview

This integration package facilitates communication and process automation between the Figaf DevOps tools and ServiceNow. It enables organizations to synchronize ticket and transport information, automating ServiceNow tasks updates based on Figaf events, and providing utilities for querying ServiceNow data.

The package currently includes the following key integration flows (iFlows):

- **Search ServiceNow Task:** Provides an endpoint to search for ServiceNow tasks based on specified criteria, called by Figaf.
- **Update ServiceNow Task:** Processes webhook events from Figaf to update corresponding ServiceNow tasks from a Virtual Task Board (Freedom Board).
- **Update Figaf Transport from ServiceNow:** Allows ServiceNow events to trigger imports in Figaf.
- This document provides details on the configuration and functionality of each iFlow.

2.Prerequisites

- SAP Integration Suite (CPI) tenant.
- Access to a Figaf instance with webhook capabilities.
- Access to a ServiceNow instance, with a Virtual Task Board (freedom board).
- Necessary authorizations in both Figaf and ServiceNow for the integration user.
- User Credentials artifact must be deployed on the CPI tenan (ex: ServiceNow_BasicAuth, containing a ServiceNow user and password, with admin user role).

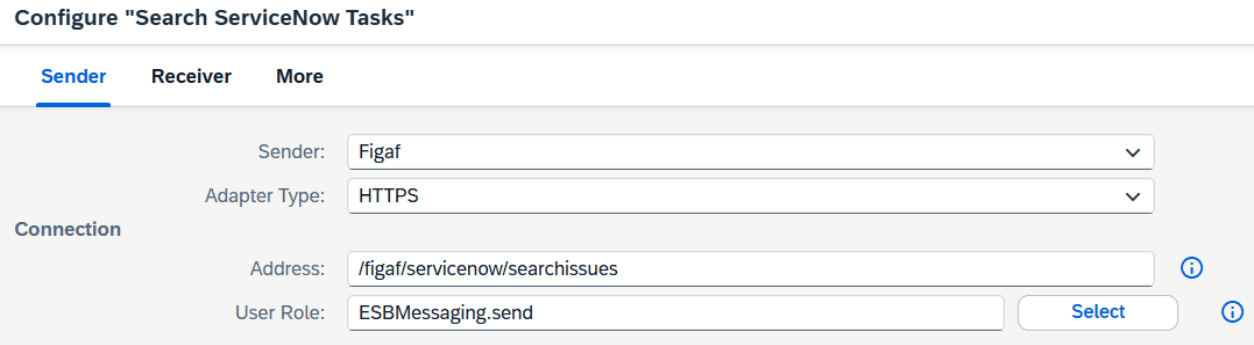
3. Integration Flows

3.1. Search ServiceNow

Provides an endpoint for Figaf (or other systems) to query ServiceNow and retrieve essential field data (key, summary, description, status) for tasks.

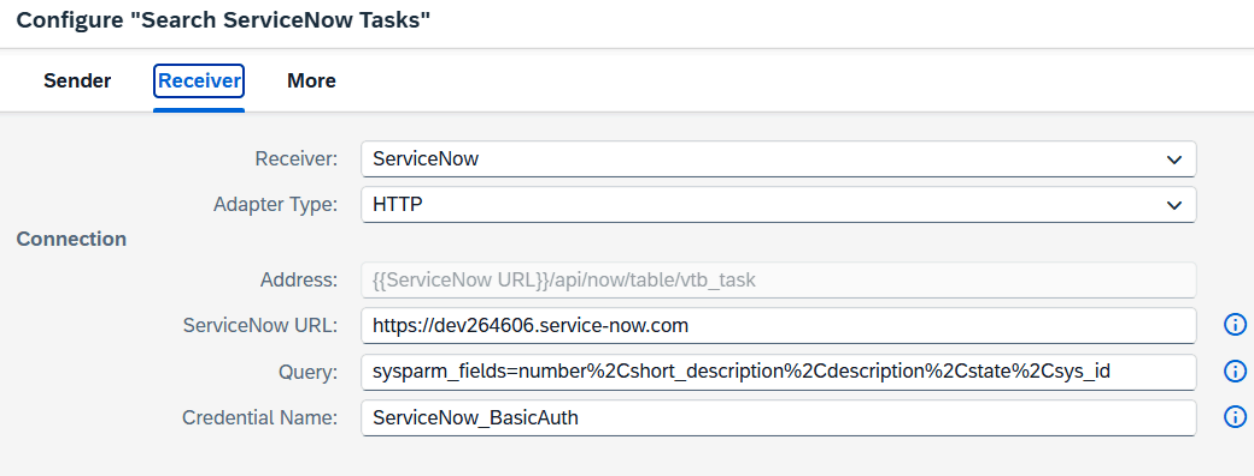
3.1.1. Configuration (from iFlow's "Configure"

Sender Details:



o **Address:** This is the path appended to your Integration Suite tenant URL to call this iFlow.
o **User Role:** The calling system must authenticate with a user having this role.

Receiver Details (Connection to ServiceNow):



o **ServiceNow URL**: The base ServiceNow search API endpoint. This should be changed to your ServiceNow instance's URL.
o **Query**: The fields retrieved.
o **Credential Name**: The alias of the User Credential artifact in Integration Suite for ServiceNow authentication (Basic Auth).

3.1.2. Using the

IFlow Trigger: HTTPS call, typically from

Input: None explicitly via body for this configuration.

Output: JSON response from SN containing the search results.

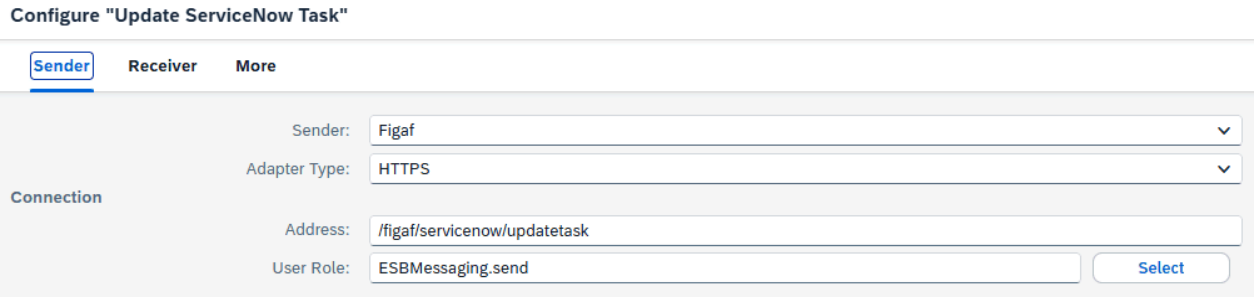

```
[
  {
    "ID": "PTSK0001002",
    "URL": "https://dev264606.service-
now.com/nav_to.do?uri=vtb_task.do?sys_id=2d506281c358321035fc7cec0501318a",
    "Title": "Task2",
    "Description": "Description 2",
    "Status": "1"
  },
  {
    "ID": "PTSK0001001",
    "URL": "https://dev264606.service-
now.com/nav_to.do?uri=vtb_task.do?sys_id=d450a281c358321035fc7cec05013134",
    "Title": "Task1",
    "Description": "Description 1",
    "Status": "2"
  }
]
```

3.2. Update ServiceNow

This iFlow is triggered by webhooks from Figaf whenever tickets or transports undergo changes. Based on a central JSON configuration, it dynamically performs actions in ServiceNow, such as transitioning tasks statuses or adding comments.

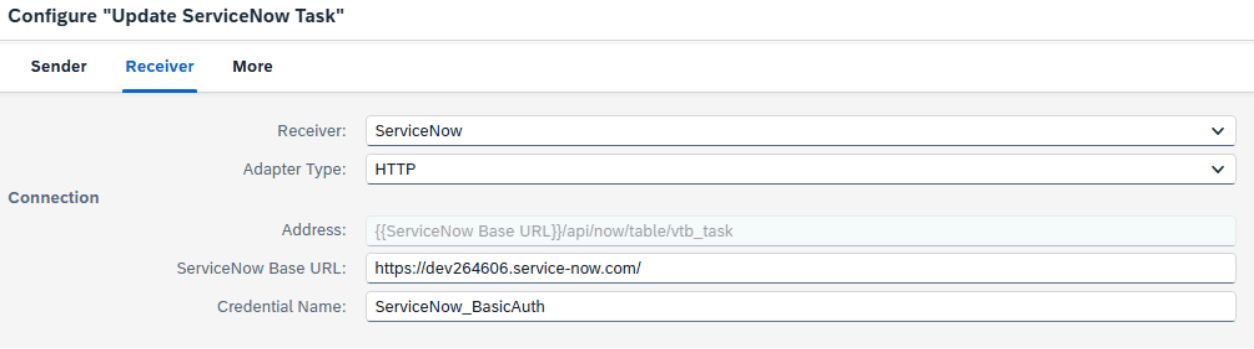
3.2.1. Configuration (from iFlow's "Configure"

Sender Details:



o **Address:** This is the path appended to your Integration Suite tenant URL to call this iFlow.
o **User Role:** ESBMessaging.send as default in this case.

Receiver:



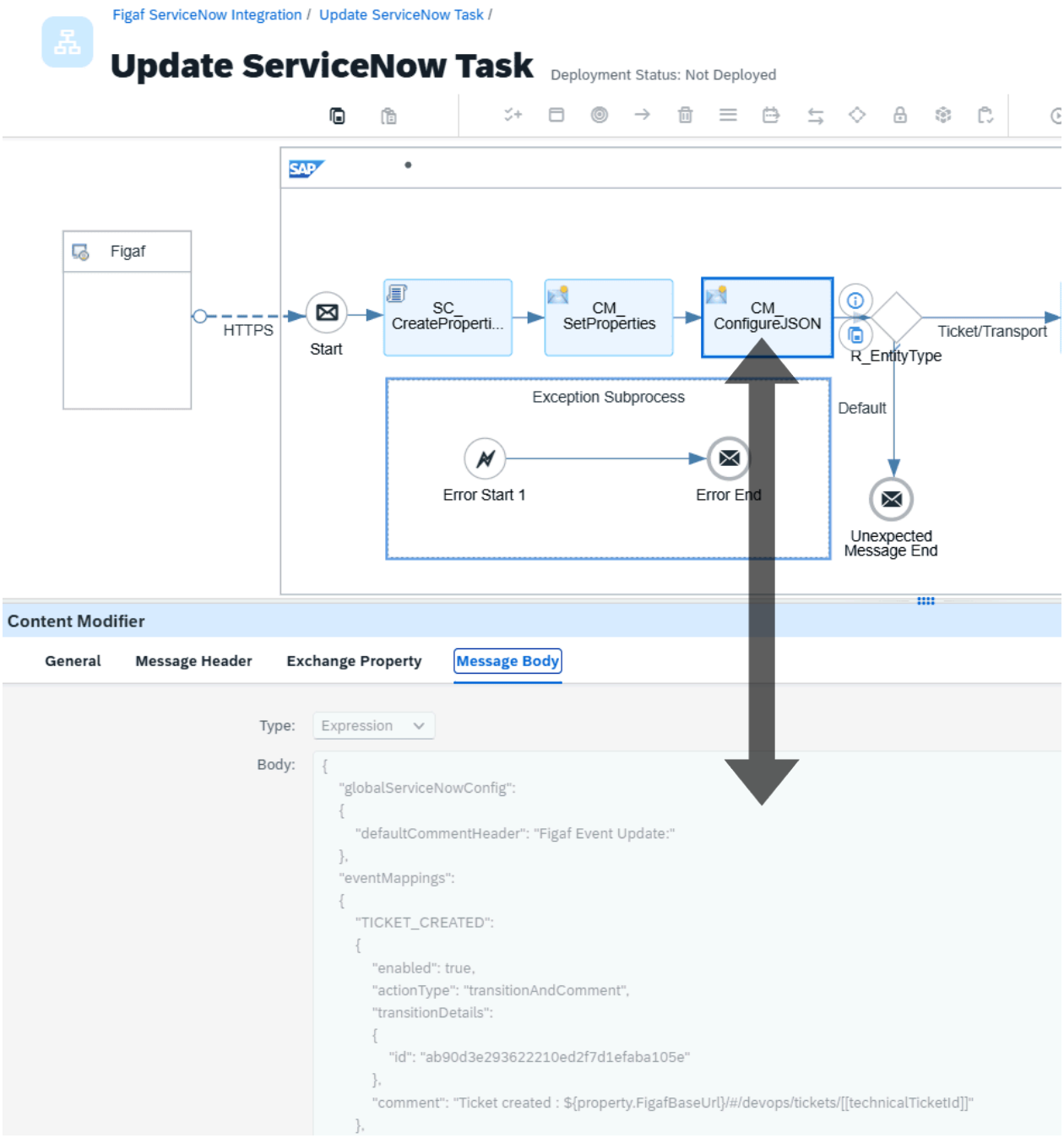
o **ServiceNow Base URL: address of your ServiceNow instance.**
o **ServiceNow Token Credential Name: Client credentials – user and key for ServiceNow**

More:

o **Figaf Base URL :** This is the path to your Figaf instance.

3.2.2. Options:

The primary configuration for this iFlow is managed through a JSON structure defined within the *CM_Initialize* Content Modifier step, specifically on its *Message Body* tab. This JSON dictates how



Key Configuration Sections within the

- **globalServiceNowConfig**:
  - defaultCommentHeader: An optional prefix automatically added to comments posted by this integration (e.g., "Figaf Event Update:").

```
"globalServiceNowConfig": {
  "defaultCommentHeader": "Figaf Event Update:"
}
```

eventMappings: This is the core section where you define actions for each Figaf eventType. For each event (e.g., TICKET_CREATED, TRANSPORT_APPROVED):

- enabled: (boolean) Set to true to process this event type, false to ignore it.
- actionType: Defines the primary ServiceNow action. Only allowed values:
  - commentOnly: Only adds a comment to the related ServiceNow issue.
  - transitionAndComment: Performs a ServiceNow workflow transition and then adds a comment.
- transitionDetails: (Object, used if actionType involves a transition)
  - id: The sys_id of the ServiceNow vtb_lane to be changed to. This ID must be found from your specific ServiceNow project.
- comment: (String) The template for the comment to be added to the ServiceNow issue. Placeholders in the format [[placeholderName]] will be replaced with values from the Figaf webhook payload.
  - Example: "Transport [[technicalTransportId]] for Figaf ticket [[webhookTicketDtoList[0].technicalTicketId]] is now [[status]]."

This is an example of configuration for the TRANSPORT_APPROVED Figaf event.

```
"eventMappings": {
  "TRANSPORT_APPROVED": {
    "enabled": true,
    "actionType": " transitionAndComment",
    "transitionDetails": {"id": "6d402281c358321035fc7cec0501318d"},
    "comment": "Transport [[technicalTransportId]] for Figaf ticket [[webhook-
TicketDtoList[0].technicalTicketId]] on landscape [[landscape.name]] is now [[sta-
tus]]."
  }
}
```

Explanation:

"enabled": **true**

The TRANSPORT_APPROVED event will be

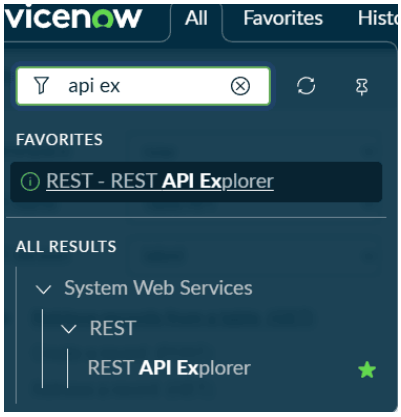
"actionType": " transitionAndComment",

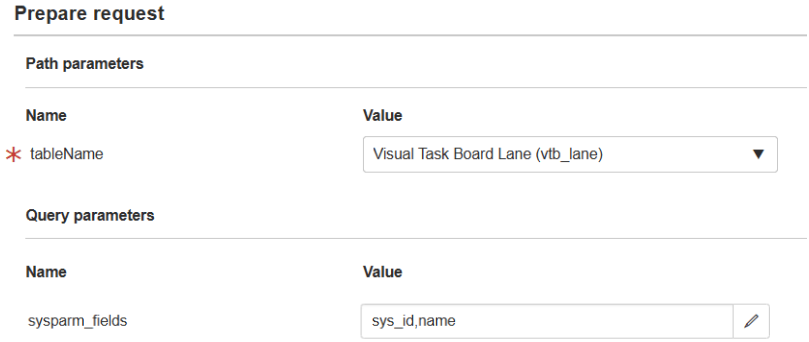
The event will produce a transition and a comment for the corresponding ServiceNow task.

"transitionDetails": { "id": "6d402281c358321035fc7cec0501318d"},
The **new lane** of the task will be that of sys_id “6d402281c358321035fc7cec0501318d”.

"comment": "Transport [[technicalTransportId]] for Figaf ticket [[webhook-TicketDtoList[0].technicalTicketId]] on landscape [[landscape.name]] is now [[status]]."
This will add a comment with some essential info.

You can find the **sys_id** of the cards lanes by using the REST API Explorer, and creating **GET** request





3.2.3. Using the

- **Trigger:** HTTPS call from Figaf
- **Input:** JSON - This iFlow expects to be called by Figaf webhooks and receive a JSON
- **Output:** JSON - Calls to ServiceNow containing data about transitions and/or comments.

A list of suggested placeholders for comments:

- [[technicalTransportId]]
- [[status]]
- [[landscape.name]]
- [[webhookTicketDtoList[0].externalTicketId]] (for the ServiceNow key if present in the Figaf payload)
- [[webhookTicketDtoList[0].technicalTicketId]] (for the Figaf ticket
- [[figafEntityId]]
- [[eventType]]

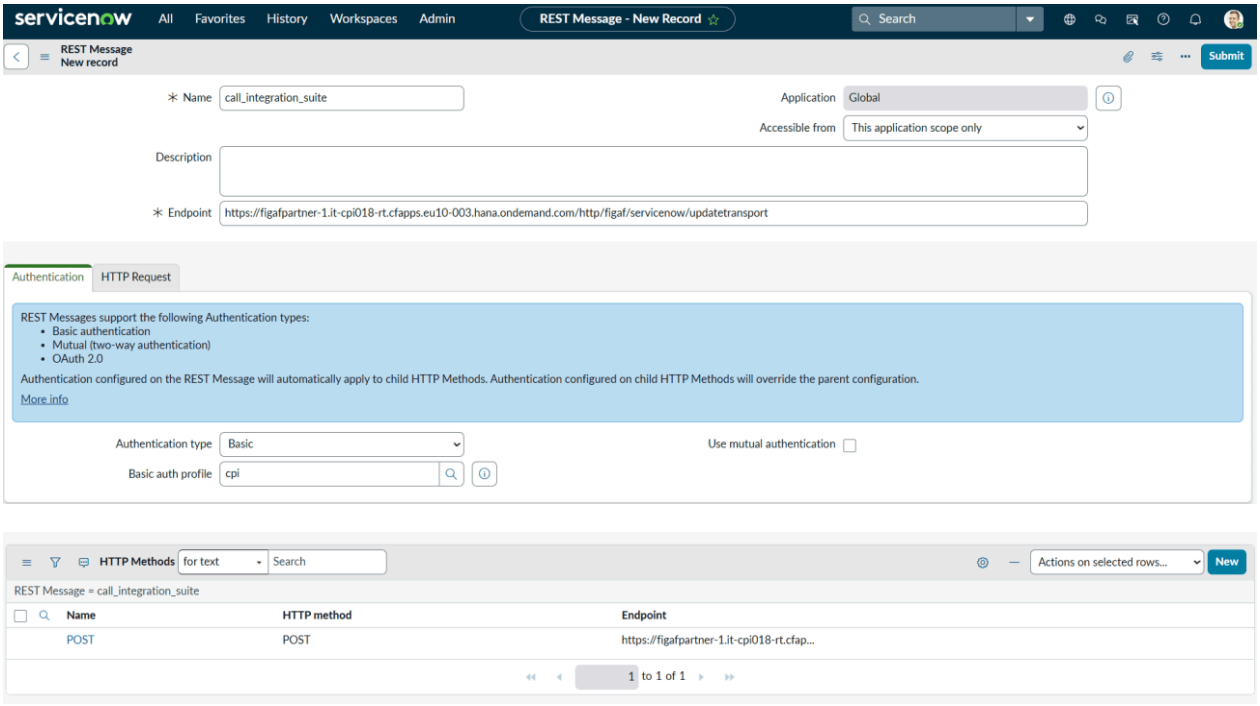
3.3. Update Figaf Transport from

This IFlow provides a way to automatically import a transport in Figaf via a ServiceNow task state change (for example, moving a task to “Done” status).

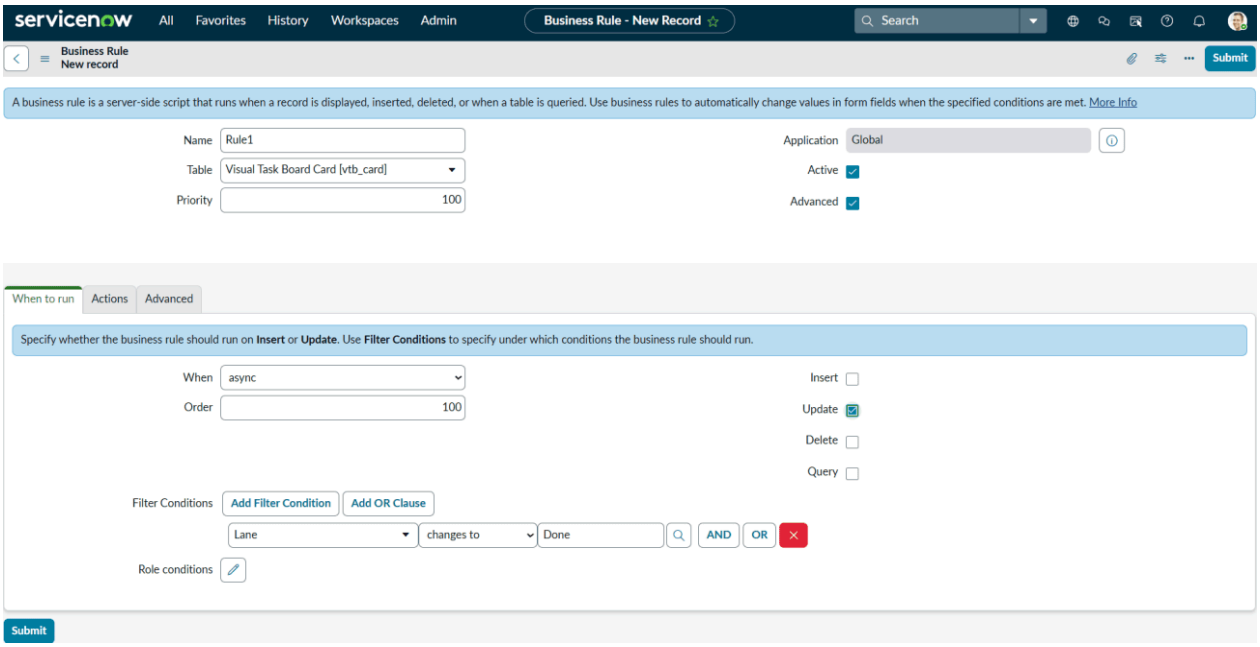
3.3.1. ServiceNow

We need to create a REST Message and a Business Rule that will call our IFlow.

1. Create a new REST Message.
2. Configure CPI basic auth. (Credentials from BTP Cockpit).
3. Create a POST method (after creating and accessing again the REST Message);



4. Create a new Business Rule. Here you can set up when will the flow be triggered (Filter Conditions).



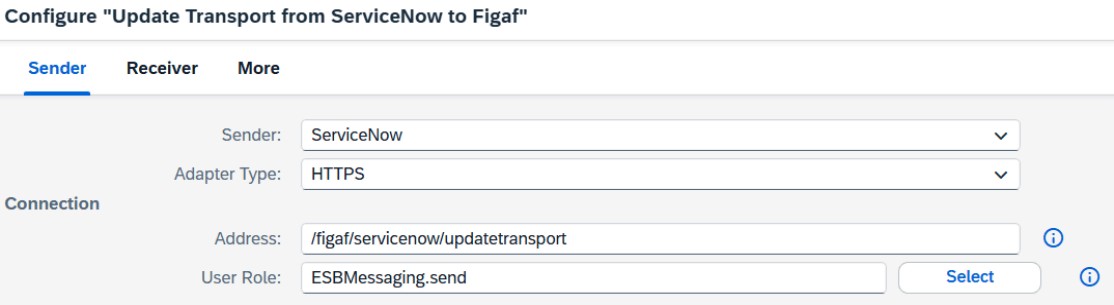
5. Under “Advanced” upload the script below, then submit the rule:

```
(function executeRule(current, previous) {
var taskNumber = '';
if (current.task) {
var taskGR = new GlideRecord('task');
if (taskGR.get(current.task.toString())) {
taskNumber = taskGR.getValue('number');
}
}
// Optionally get the state or lane
var taskState = current.getValue('lane');
// Build payload later using these values

var payload = {
id: current.sys_id.toString(),
number: taskNumber,
// state: taskState, // Uncomment if you also want state
updated: current.sys_updated_on.toString()
};

var restMessage = new sn_ws.RESTMessageV2('call_integration_suite',
'POST');
restMessage.setRequestBody(JSON.stringify(payload));
var response = restMessage.execute();
var responseBody = response.getBody();
```

```
var httpStatus = response.getStatusCode();
gs.info('Webhook response: ' + httpStatus + ' ' + responseBody);

})(current, previous);
```

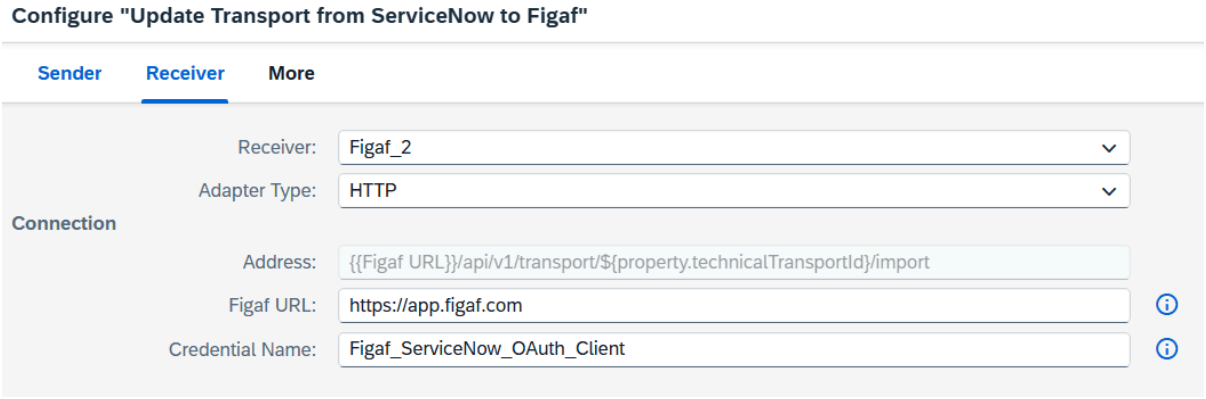
3.3.2. Configuration (Integration

• Sender Details:



o **Address:** This is the path appended to your Integration Suite tenant URL to call this iFlow.
o **User Role:** User role for authentication (ESBMessaging.send in this case).

- Receiver Details (Connection to ServiceNow):

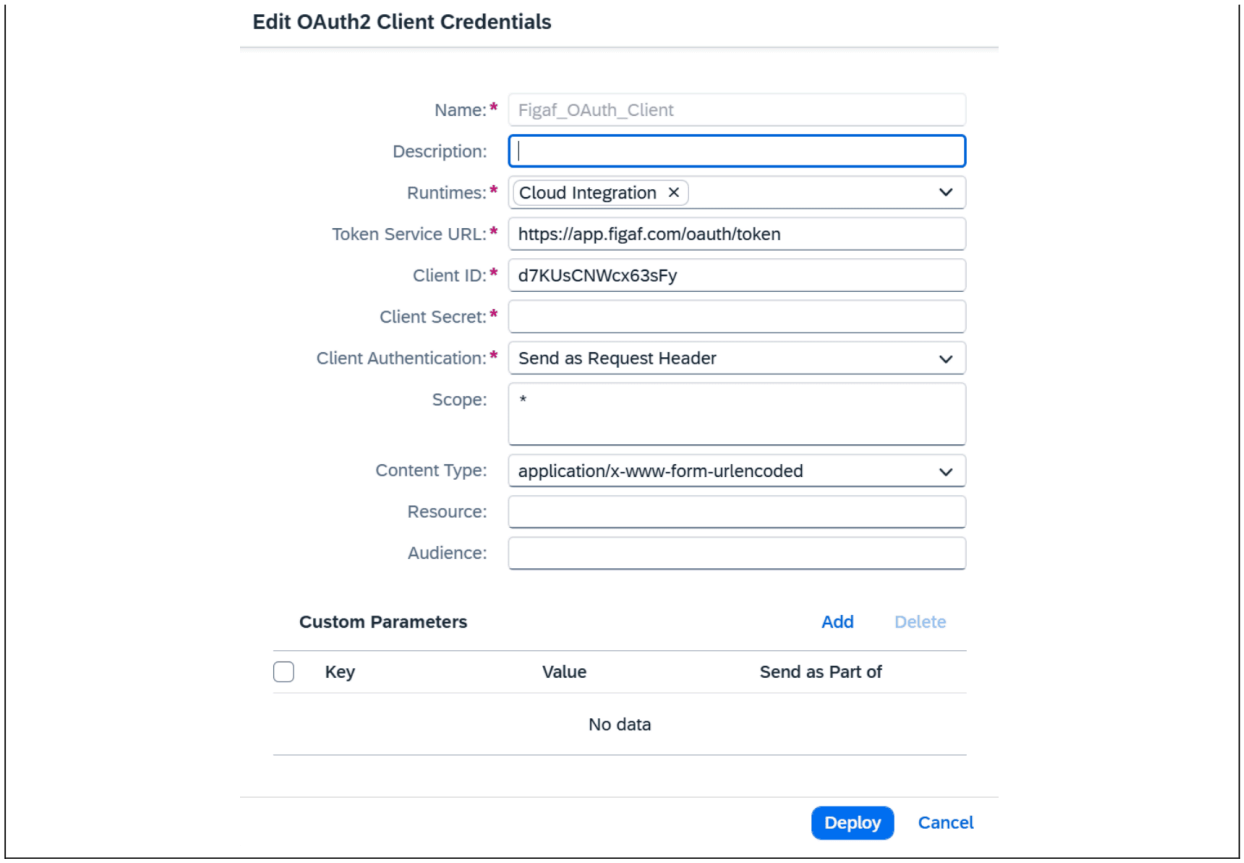


o **Figaf URL**: The base Figaf URL. Will be used to build the address to call the API.
o Credential Name: The alias for storing the Figaf OAuth Client Credentials.

To create the Figaf OAuth Client, in Figaf go to Configuration -> OAuth Clients -> Add OAuth Client. Select the following scopes:

ticket:read, ticket:run, ticket:import, ticket:resolve,

Then add the credentials in Integration Suite like below. Token Service Url is your Figaf url followed by /oauth/token.



- More config.

o **Triggering State**: The state of the card triggering a transport import.

4. Deployment and General Setup

1. **Deploy Credentials:**

    - Ensure the User Credential artifact (containing valid ServiceNow credentials) is deployed on your Cloud Integration tenant and that the iFlows configurations reflect the same alias.

    - Ensure the User Credential containing Figaf OAuth Client certificate is deployed.

2. **Configure “Search ServiceNow Tasks”:**
    - Access the iFlow and go to the "Configure" view.
    - Update the Receiver Address to your ServiceNow instance URL.
    - Modify the Receiver Query parameter to match your desired JQL (project, fields, etc.).

3. **Configure “Update ServiceNow Tasks”:**
    - Access the iFlow;
    - Configure the sender/receiver/more settings;
    - Modify the CM_ConfigureJson Content Modifier's Message Body with your preferences and the desired logic (sys_id of cards lane, comment templates, etc.) as detailed in section 3.2.

4. **Configure “Update Transport from ServiceNow to**
    - Follow the Setup Steps from sections 3.3.1.
    - Update the IFlow’s Configuration.

5. **Deploy Integration Package:** Deploy this integration package and it’s iFlow’s to your Integration Suite tenant.

6. **Configure Figaf Webhooks:**

    - In Figaf, set up a new integration and then the webhooks to point to the endpoint URL of the deployed **Update ServiceNow Issues iFlow**.