# Configuration Guide - Medallia Customer Experience Integration with SAP Signavio Process Intelligence

April 2025

SAP BTP

# Table of contents

# Prerequisites

Prerequisites to establish the Integration between Medallia Customer Experience and Signavio Process Intelligence are:

## Access to Medallia

1. Integration role for OAuth client

- Go to Admin Suite
- Go to Organization > Roles
- Click on New Role
- Fill the form as follows:
    - o Role Name: SAP Signavio API
    - o Role Type: Other
    - o Role Group: Integrations
- Click on Save
- Click on Administrative Permissions
- Enable the View Data from Query API permission
- Click on Save

2. OAuth client for API access

a. Go to Advanced Setup (Classic)
b. Go to Integrations > OAuth
c. Click on AppID Accounts tab
d. Click on New
e. Fill the form as follows:
    a. Account Name: sap_signavio
    b. Account Type: Custom
    c. Enabled: must be checked
    d. Role: Select SAP Signavio API
f. Click on Save
g. Click on Clients tab
h. Click on New
i. Fill the form as follows:
    a. Client ID: sap_signavio_oauth
    b. Confidential: must be checked
    c. Grants: Client Credentials Grant (for backend services)
    d. Client subject: sap_signavio
j. Click on Save
k. The Generate new client secret button will now be visible,
    a. Check Confirm next to it
    b. Click the Generate new client secret button

l. Scroll down to the end of the page and you should see a green box saying "Your request was successful!"
m. Copy the client_secret value (note that after navigating away from this screen the client_secret will no longer be visible, make sure to store it securely)
n. Click on Configuration tab
o. Copy the value for OAuth Token endpoint

3. API Hostname

- Go to Advanced Setup (Classic)
- Go to Integrations > Public APIs Setup
- Copy the value on Public APIs Hostname

## Access to Signavio Process Intelligence

In order to bring the customer, experience free text data into Process Intelligence for further analysis, a license for SAP Signavio Process Intelligence is needed.
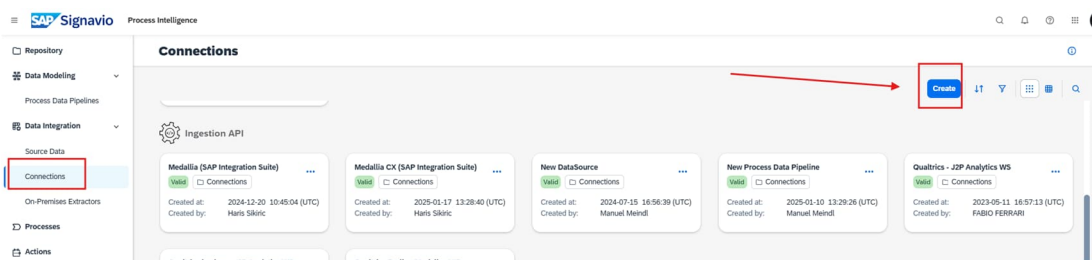


To create a connection to Medallia go to Process Intelligence, then navigate to Data Integration, click on Connections and then select Create

Choose the Name (Example Medallia CX). Set the Connection Type to Ingestion API and choose your Repository.



After the connection for Medallia is created, we will receive the API endpoint needed for configuring the receiver adapter in the iFlow "Import Medallia Data to SAP Signavio Process Intelligence" (see chapter: Configuration steps on SAP Cloud Integration) and we will receive the token for authentication, which must be stored as a secure parameter in the Security Material section (see chapter: Access to SAP Cloud Integration).
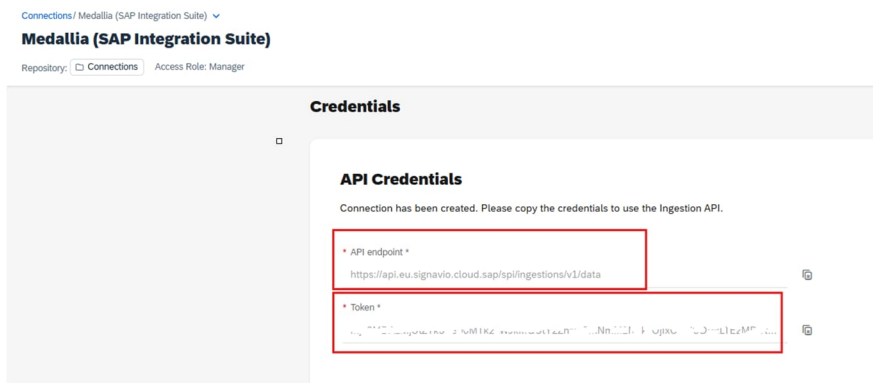
## Access to SAP Cloud Integration

Before configuring the iFlows, you need to set up the credentials for authentication from SAP Cloud Integration to Medallia and Signavio.

Navigate to the "Manage Security" section within the Monitoring area and then click on Security Material.
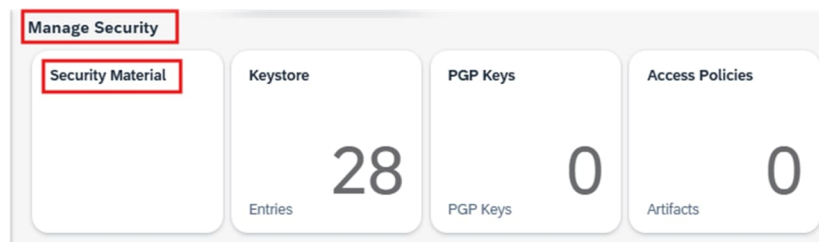


After that, we first set up the OAuth credentials for Medallia, configure the Values and Deploy.
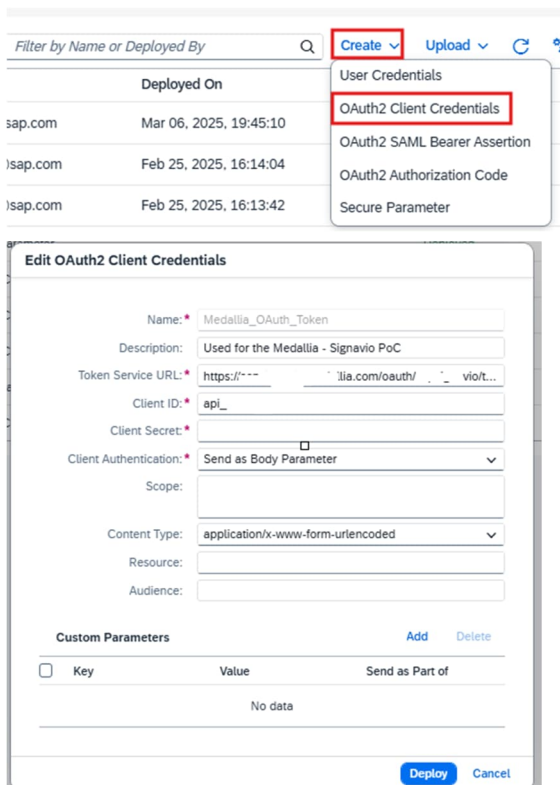
| Parameter | Value |
|---|---|
| Name | Select an alias of your choice, e.g., *Medallia_OAuth_Token.* |
| Description | Choose a description of your preference, such as *Medallia – Signavio Integration* |
| Token Service URL | See Chapter Prerequsities → Access to Medallia |
| Client ID | See Chapter Prerequsities → Access to Medallia |
| Client Secret | See Chapter Prerequsities → Access to Medallia |
| Client Authentication | Send Body as Parameter |
| Content Type | application/x-www-form-urlencoded |

After that, we create the secure parameter for Signavio Process Intelligence authentication and deploy it.



| Parameter | Value |
|---|---|
| Name | Select an alias of your choice, e.g., *Medallia_Ingestion_Token.* |
| Description | Choose a description of your preference, such as *Medallia – Signavio Integration* |
| Secure Parameter | See Chapter Prerequsities → Access to Signavio Process Intelligence |

# Integration Logic Description

The package includes a total of 5 interfaces. The "Replicate Customer Experience Data from Medallia to Signavio" interface is the main interface, and all other iFlows are called synchronously via the Process Direct Adapter. The interfaces "Replicate Customer Experience Data from Medallia to Signavio," "Import Medallia Data to SAP Signavio Process Intelligence," and "Medallia Errorhandler" should not be edited.
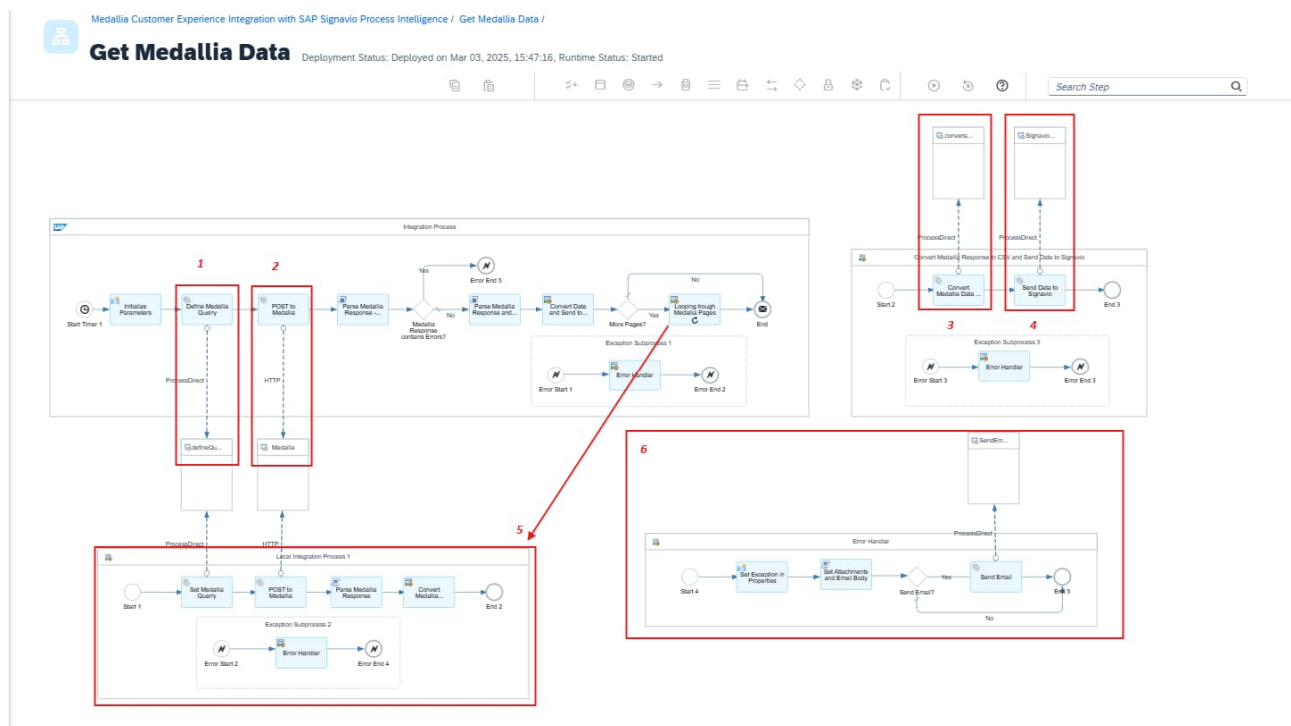
The interfaces "Define Medallia Query" and "Convert Medallia Data to CSV" can be customized according to needs, use cases, and specific customization requirements. The package is designed such that the Process Direct endpoints are configurable, meaning custom iFlows can be developed and then called via Process Direct Adapter.



Data flow:

1. Prepare the Medallia query (GraphQL)

2. Send the query to Medallia

3. Convert the Medallia response into CSV

4. Send the CSV to the Signavio Handler iFlow

5. Looping logic (Required if the Medallia response has multiple pages based on the configuration)

6. Error handler


The "Replicate Customer Experience Data from Medallia to Signavio" interface is a so-called time scheduled full load interface, meaning it always retrieves data based on the configuration (filter & query) set in the "Define Medallia Query" interface. Essentially, a JSON response is returned (if multiple pages are present), all pages are fetched, converted to CSV, and then prepared for the upload to Signavio Process Intelligence in the "Import Medallia Data to SAP Signavio Process Intelligence" interface.

# Configuration steps on SAP Cloud Integration

## Replicate Customer Experience Data from Medallia to Signavio Configuration

This is the main iFlow; for more details, see "Integration Logic Description."

### Configure Timer

During the initial run, you should trigger the iFlow for the first time upon deployment. Afterward, adjust the scheduler according to your needs.



### Configure Receiver Adapter

In the receiver section, there are a total of 5 receivers, but only Medallia needs to be configured.

The other Process Direct endpoints can be configured according to your use case if you want to experiment with your own logic.

Configure "Replicate Customer Experience Data from Medallia to Signavio"

The host_name parameter is the Medallia endpoint, which must be provided by the Medallia admin, and the Medallia credentials are the alias of the Medallia OAuth credentials that were stored in the chapter "Prerequisites Access to SAP Cloud Integration."



Configure "Get Medallia Data"

## Additional Parameters

In the "More" section, you should configure only the parameters error_email_notification (email dispatch in case of failure) and error_log_attachment (logging in case of an error). Adjust these settings according to your needs.



Configure "Get Medallia Data"

# Define Medallia Query Configuration

The Query API is a powerful analytics API that harnesses GraphQL to give you access to Medallia Experience Cloud data through a single endpoint. Send a standard HTTP POST request to the server through a single endpoint, specify what data you need, and get back exactly that.

More Details: Query API | Medallia Documentation

## Configure Sender Adapter

In the Sender section, everything remains as it is. The Process Direct endpoint can be changed if a different query is needed or if the existing one, along with all configurations, is insufficient.

**Configure "Define Medallia Query"**

**Sender**    More

Connection

| | |
|---|---|
| Sender: | Cloud_Integration |
| Adapter Type: | ProcessDirect |
| Address: | /medallia/querry |

## Additional Parameters

In our case, we use the standard query *getSurveyDetails* for the record-level response extraction use case.

The exact values for the *filter* parameter will depend on the use case, but commonly filters include date ranges and program conditions to fetch desired data. By leaving the filters as null your query is extracting everything the OAuth client has access to.

With the parameter *node_structure,* you specify the fields that are relevant for the use case. The nodes are specific fields to pull for each record.

The parameter *number_per_page* specifies the number of records per page. The maximum number of records per request will vary depending on the data requested for each one. The more fields the query includes, the fewer records will fit in the response. The limitation that must be considered to define page size is the cost limit, which provides a measurement of how many resources are needed to serve the request. While there is no clear recommendation as it depends on the data volume, it would be advisable to test with up to 500 records per page and monitor the cost measurement (ideally kept in the low hundreds) and response time to adjust accordingly for your use case and avoid timeouts.

More Details about limit can be find here: Restrictions and limits | Medallia Documentation

**Configure "Define Medallia Querry"**

For example, if you want to request a specific date range the filters could look something like the following:

```
{
        "variables": {
                "filter": {
                        "and": [
                                {
                                        "fieldIds": "e_responsedate",
                                        "lt": "2025-01-01"
                                },
                                {
                                        "fieldIds": "e_responsedate",
                                        "gte": "2024-10-01"
                                }
                        ]
                }
        }
}
```

You need to convert the JSON Representation into a single line "compact representation."

*{"and":[{"fieldIds":"e_responsedate","lt":"2025-01-01"},{"fieldIds":"e_responsedate","gte":"2024-10-01"}]}*



## Convert Medallia Data to CSV

As described in the chapter "Integration Logic Description," the Medallia JSON response is converted into CSV format. A Groovy script was used for this conversion. If there is a need for more complex conversions, you can implement your own JSON-to-CSV conversion, ensuring that the format of the CSV header adheres to specific guidelines.

## Configure Sender Adapter



**Configure "Convert Medallia Data to CSV"**

| Sender | More |

**Connection**

Sender: Cloud_Integration
Adapter Type: ProcessDirect
Address: /convert/medallia/data

## Additional Parameters

More Information: https://help.sap.com/docs/signavio-process-intelligence/api-guide/ingestion-api-data-upload?locale=en-US



**Configure "Convert Medallia Data to CSV"**

| Sender | More |

Type: All Parameters
csv_header: id,email_id,comment

# Import Medallia Data to SAP Signavio Process Intelligence

After the Medallia JSON response has been converted into CSV, the final step is to upload the CSV payload to Signavio Process Intelligence using the Ingestion API.

## Configure Sender Adapter

In the configuration for the sender adapter, nothing needs to be done.



**Configure "Import Medallia Data to SAP Signavio Process Intelligence"**

| Sender | Receiver | More |

**Connection**

Sender: Cloud_Integration
Adapter Type: ProcessDirect
Address: /medallia/signavio/pint

## Configure Receiver Adapter

In the receiver adapter configuration, the Ingestion API endpoint must be added, which can be obtained from the chapter "Prerequisites - Access to Signavio Process Intelligence."

Configure "Import Medallia Data to SAP Signavio Process Intelligence"

## Additional Parameters

The *connection_token* parameter is the alias of the secure parameter (for Process Intelligence) that was created in the chapter "Prerequisites - Access to SAP Cloud Integration."

The *fileName* parameter is defined arbitrarily. It appears in the logs after the import.



The *logger* parameter is used to log the request and response payloads to and from Signavio. The default value is false.

Regarding *primary_keys* and *schema*, please see the explanation in the documentation:
https://help.sap.com/docs/signavio-process-intelligence/api-guide/sending-ingestion-request?locale=en-US



# Send Error Notification Emails (Medallia Error Handler)

This interface serves as a central point for email dispatch in case of an error. If emails are to be sent, the SMTP server must be configured for email delivery.

## Configure Sender Adapter

# Configure Receiver Adapter

**Configure "Send Error Notification Emails"**

**Sender**  **Receiver**

| | |
|---|---|
| Receiver: | EMail_Receiver ⌄ |
| Adapter Type: | Mail ⌄ |

**Connection**

| | |
|---|---|
| Address: | TBD |
| Proxy Type: | Internet ⌄ |
| Timeout (in ms): | 30000 |
| Protection: | STARTTLS Mandatory ⌄ |
| Authentication: | Encrypted User/Password ⌄ |
| Credential Name: | TBD |

**Processing**

| | |
|---|---|
| From: | TBD@from.com |
| To: | TBD@to.com |
| Cc: | |
| Bcc: | |
| Subject: | |
| Body Mime-Type: | Text/Plain ⌄ |
| Body Encoding: | UTF-8 ⌄ |
| Add Message Attachments: | ☐ |
| Content Transfer Encoding: | Automatic ⌄ |

**Security**

| | |
|---|---|
| Signature and Encryption Type: | None ⌄ |