



Salesforce Service Cloud Adapter for SAP Integration Suite

Version 1.0.0 – October 2026

Contents

1	Introduction	3
1.1	Objective.....	3
1.2	Coding Samples.....	3
1.3	Internet Hyperlinks.....	3
1.4	Overview	3
1.5	Features.....	4
1.6	Quick Start	4
2	Installation and Configuration.....	6
2.1	Prerequisites	6
2.2	Procedure	6
2.2.1	Adapter Installation by creating a New Integration Flow.....	6
2.2.2	Adapter Installation without creating a New Integration Flow	7
3	Getting Started: Salesforce Service Cloud Adapter.....	9
3.1	Architecture Overview.....	9
3.2	Application Configuration	10
3.2.1	Account Creation	10
3.2.2	Application Setup	10
3.3	Authentication	10
3.3.1	Creating Credentials in Security Material	10
3.3.1.1	Creating User Credentials	11
3.3.1.2	Creating Secure Parameter	12
3.3.1.3	Adding Key Pair to Keystore	13
4	Salesforce Service Cloud Adapter Configuration	14
4.1	Connection.....	14
4.2	Processing.....	17
5	Salesforce Service Cloud Operations	23

5.1	Salesforce Object: Create	23
5.1.1	Basic.....	23
5.1.2	Advanced.....	24
5.2	Salesforce Object: Composite	24
5.3	Job: Create.....	26
5.4	Job Query: Get Results.....	28
5.5	Apex Call: Post.....	29
6	Reference	31
6.1	Initialize OAuth Username-Password in Salesforce.....	31
6.2	Initialize OAuth JWT Bearer in Salesforce	34
6.2.1	Create a Connected App in Salesforce	34
6.2.2	Create Security Certificate and JKS file using OpenSSL	36

1 Introduction

1.1 Objective

This is the official guide for the Salesforce Service Cloud Adapter for SAP Integration Suite. This guide covers all relevant information for integration developers to start working with the Salesforce Service Cloud adapter. Read this guide carefully before using the Adapter.

1.2 Coding Samples

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. The correctness and completeness of the Code given herein are not guaranteed.

1.3 Internet Hyperlinks

The documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. The availability and the correctness of this related information or the ability of this information to serve a particular purpose are not warranted.

1.4 Overview

Salesforce Service Cloud adapter for SAP Integration Suite allows you to access Salesforce APIs REST, Bulk 2.0, and Apex Calls. Salesforce Service Cloud is a customer relationship management (CRM) platform designed to help businesses manage and resolve customer inquiries across multiple channels, including phone, email, chat, social media, and messaging.

1.5 Features

Salesforce Service Cloud Adapter has the following features:

- **Receiver** adapter enables access to sObjects like Account, Case, Contact, Goal, Task, User, etc.
- Supports **multiple authentications** using OAuth 2.0 Username-Password, OAuth 2.0 Client Credentials, OAuth 2.0 JWT Bearer.
- **Basic** configuration provides a convenient processing capability for supported versions, whereas **Advanced** enables proficient users to perform calls with greater control while connecting to an API endpoint.
- **Querying and searching:** In addition to all the operations that the adapter offers, it allows the querying of data from a Salesforce system using SOQL and SOSL queries.
- **Support for creation/upsert of Aggregated Structures:** Supports a composite Salesforce call where the request body can contain a combination of different operations (such as Upsert and Read) on multiple sObjects at the same time.

1.6 Quick Start



This Quick Start section outlines the minimal steps required to demo adapter usage. This does not cover all configuration scenarios; always refer to the complete documentation for complete and accurate guidance.

1. Prerequisites:
 - a) Salesforce account (see [To get a Salesforce Developer Edition.](#))
 - b) SAP Integration Suite tenant with Standard License.
2. Authentication
 - a) Set up OAuth2 Client Credential Authentication. (See [Creating User Credentials](#))
3. Create **OAuth2 Client Credentials** in SAP Integration Suite:
 - a) Navigate to **Monitor > Integrations and APIs**.
 - b) Go to **Manage Security > Security Material**.
 - c) Create an OAuth2 Client Credential using Client ID and Client Secret (see [Application Setup](#) point 3)
4. Add Salesforce Service Cloud Adapter to an Integration Flow:
 - a) Go to the Design workspace and create or open an Integration Flow.
 - b) Add Salesforce Service Cloud Adapter as a Receiver.
5. Configure the Connection Tab:
 - a) **Address:** Enter a region-specific address where your account is hosted.
Example: `https://{MyDomain}.my.salesforce.com`
 - b) **Key Name Alias:** Enter the OAuth2 Client Credentials created in step 2c.
 - c) Keep default timeout values.
6. Configure Processing :

- a) Perform the Basic version of the [5.5 Apex Call: Post](#) operation.
- 7. Deploy & Run:
 - a) Save and deploy the Integration Flow.
 - b) Monitor execution under Integrations and APIs.

2 Installation and Configuration

This section details the prerequisites to install and configure the Salesforce Service Cloud adapter.

2.1 Prerequisites



The Salesforce Service Cloud adapter is available as part of your Standard license for SAP Integration Suite. For more information, see [SAP Note](#).

Before you start working with the adapter, you must deploy it to your SAP Integration Suite tenant.

2.2 Procedure

You can deploy the adapter using the following methods:



The following installation procedure is compatible with an Apache Camel 3x environment and Edge Integration Cell (EIC) platform.

2.2.1 Adapter Installation by creating a New Integration Flow

The Salesforce Service Cloud adapter is available for selection in the receiver adapter list. It can be deployed in the **Design** tab directly as you use it in an Integration flow.

Purpose

To install an adapter for use in your Integration flow.

Procedure

1. Go to the **Design** workspace and select the integration package where you want to create a new Integration flow.
2. Click **Edit** to make the package editable.
3. Go to the **Artifacts** tab. Click **Add** and select **Integration Flow**.
4. Enter **Name** and **ID** for your flow. Additionally, select **Runtime Profile** from the drop-down and choose **Sender** and **Receiver** systems from the list . Finally, click **Add** to create the integration flow.
5. Go to the newly created integration flow and click **Edit** to make it editable.

6. In the integration flow, click **End** to add a **Connector**  between the **End** and the **Receiver** Box.
A drop-down with the available adapters appears. The **SalesforceServiceCloud** adapter should show up in the list.
7. Select the **SalesforceServiceCloud** adapter from the list. The adapter is now imported which *triggers* an adapter deployment. Once the Salesforce Service Cloud Adapter is deployed, a success message is displayed.

After all the above steps are completed, the Salesforce Service Cloud Adapter is successfully deployed to your Design workspace in the SAP Integration Suite tenant.

2.2.2 Adapter Installation without creating a New Integration Flow



The following procedure explains how the Salesforce Service Cloud adapter is migrated from the **Discover** workspace to the **Design** workspace of the SAP Integration tenant.

This method is useful for scenarios where integration flow packages are migrated from development to a higher environment such as Production.

The Salesforce Service Cloud adapter can be imported into the Design workspace without creating an integration flow. Use the Transport Management Service (TMS) to import/transport the Salesforce Service Cloud adapter to a higher environment. Alternatively, if the TMS is not available in the landscape, the adapter package can be imported to the Design workspace by copying it from the Discover workspace.

Purpose

To copy the integration package from the Discover workspace and import the Salesforce Service Cloud adapter to the Design workspace, follow these steps:

Procedure

1. Go to **Discover** workspace.
2. In the search box, search for **Salesforce Service Cloud adapter for SAP Integration Suite** package.
3. Select the package and click **Copy**. This copies the package from the Discover workspace to the Design workspace.
4. Go to Design workspace and select the copied **Salesforce Service Cloud adapter for SAP Integration Suite** package.
5. In the **Actions** tab of the selected package, click **Deploy**. This completes the adapter deployment to the Design workspace.

After the adapter deployment is complete, you can check the status in the **Monitor** section.

Purpose

To check the status of the deployed adapter.

Procedure

1. Under the **Monitor** tab, click **Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Integration Content** section and click **All**. This opens the **Integration Content** page with a list of all the deployed adapters.
3. Here, you can check and confirm the deployment status of your adapter.

The screenshot displays the 'Manage Integration Content' page in the Salesforce interface. The breadcrumb trail at the top reads 'Overview / Manage Integration Content'. The main heading is 'Manage Integration Content', with a 'Runtime' dropdown menu set to 'Cloud Integration'. On the left, a table titled 'Integration Content (775)' lists the 'SalesforceServiceCloud' adapter with a status of 'Started'. The right-hand panel provides details for 'SalesforceServiceCloud', including deployment dates, version (1.0.0), and links for 'Undeploy' and 'Download'. A green status message at the bottom confirms: 'The Integration Adapter is deployed successfully.'

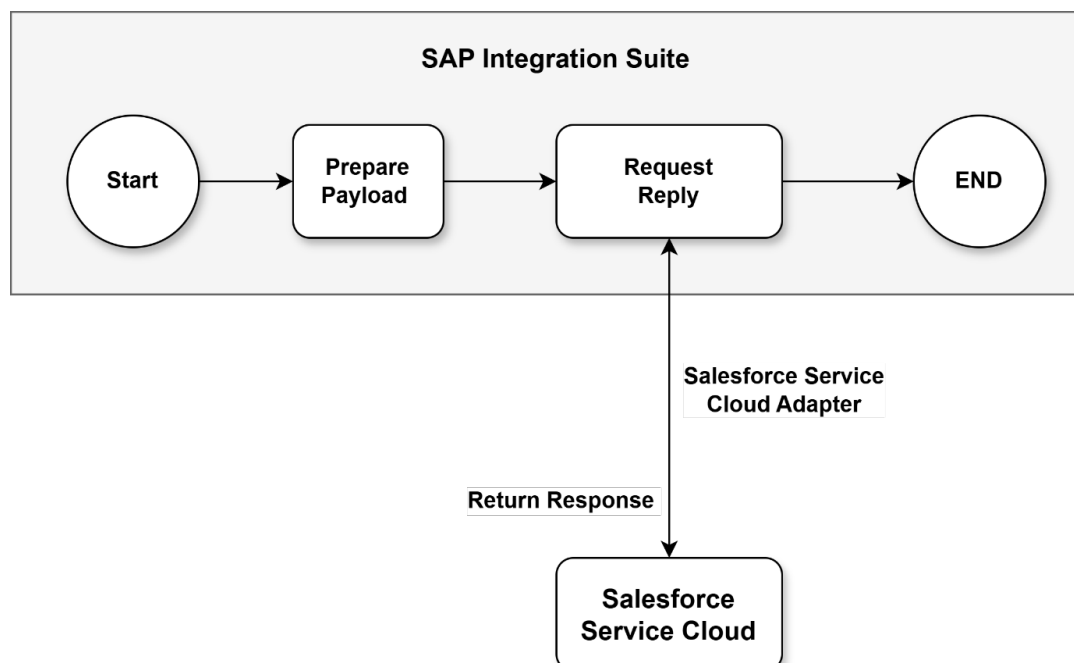
3 Getting Started: Salesforce Service Cloud Adapter

This section contains information to help you get started with the Salesforce Service Cloud Adapter, along with relevant setup and configurations on the application side of Salesforce Service Cloud. You can also find information about adapter architecture and the creation of security artifacts.

3.1 Architecture Overview

The Salesforce Service Cloud adapter is designed to process data from the integration flow and communicate with external Salesforce APIs.

- It is a receiver adapter, designed to receive and process data from an integration flow in SAP Integration Suite and facilitate communication with the Salesforce Service Cloud application (external) using Salesforce REST, Bulk 2.0, and Apex REST APIs.
- SAP Integration Suite acts as the initiator of the calls and sends a request to Salesforce Service Cloud (this is the receiver system) using the adapter. It invokes the Salesforce Service Cloud server to perform operations. Additionally, the adapter also handles responses and logs execution status and errors.



3.2 Application Configuration

The following resources can help you set up your Salesforce Service Cloud application before you start using the adapter in an integration flow.

3.2.1 Account Creation

To create a Salesforce Service Cloud account, see [To get a Salesforce Developer Edition](#).

3.2.2 Application Setup

1. For more information regarding OAuth 2.0 Username-Password, [Authorization OAuth 2.0](#).
2. For more information regarding OAuth 2.0 JWT Bearer, [Access Tokens](#).
3. For more information regarding configuration, see [OAuth 2.0 Client Credentials Flow](#).
 - i. As per the instructions in the above link, you must create an external client app. For more information, see [Create an External Client App](#).
 - ii. After completing the above setup, fetch the Consumer Key and Consumer Secret for your app using External Client App Manager. For more information, see [6.1 Initialize OAuth Username-Password in Salesforce](#).

3.3 Authentication

The Salesforce Service Cloud Adapter supports **OAuth 2.0 Username-Password**, **OAuth 2.0 JWT Bearer**, and **OAuth 2.0 Client Credentials** authentication mechanisms.

You can use Secure Parameter and User Credentials artifacts to safely store OAuth details, user-password combinations, and a key pair for JWT. These security artifacts can then be accessed in the Adapter using aliases.

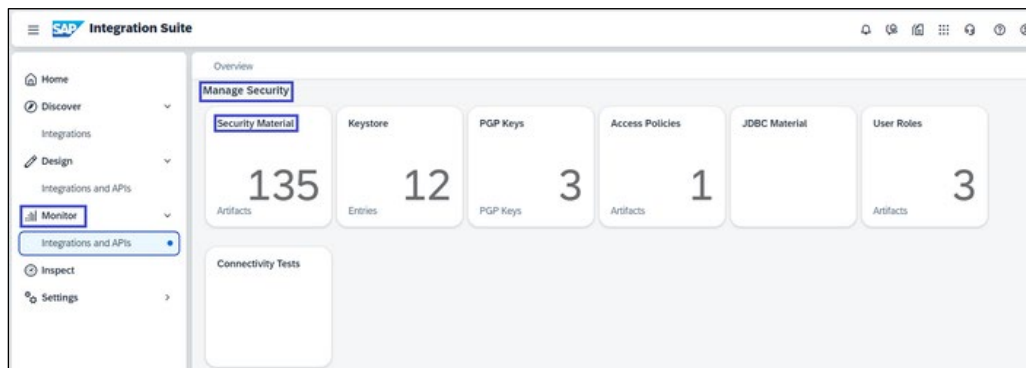
3.3.1 Creating Credentials in Security Material

There are three authentication methods, and each one requires its own set of security artifacts, as outlined below:

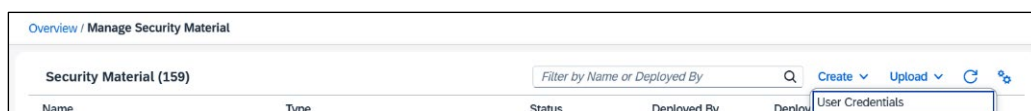
- OAuth 2.0 Username-Password requires you to create two User Credentials artifact (username/password and consumer key/secret), and a Secure Parameter (security token, optional).
- OAuth 2.0 JWT Bearer needs two Secure Parameters (subject and issuer) and a Key Pair.
- OAuth 2.0 Client Credentials authentication requires a User Credentials artifact that stores the consumer key and consumer secret.

3.3.1.1 Creating User Credentials

1. In SAP Integration Suite, navigate to **Monitor** > **Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to the **Manage Security** section and click **Security Material**.



3. On the **Manage Security Material** page, click **Create** to select **User Credentials** from the dropdown.



4. In the **Create User Credentials** popup, provide the below details.

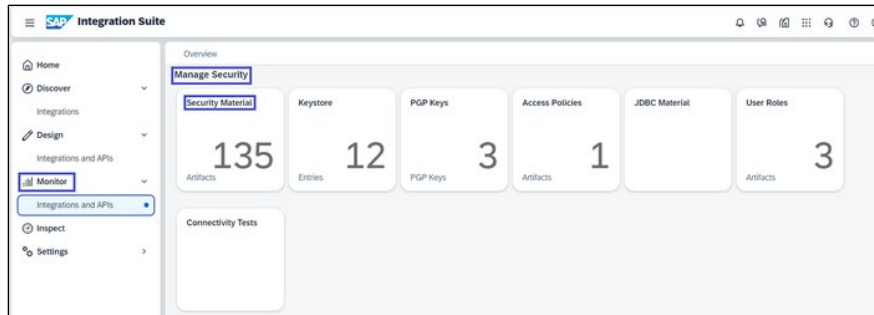
Parameter	Description
Name	Specify the name for the security artifact. The artifact name is used as an alias for the confidential data assigned by this parameter.
Description	Enter a description for the artifact (optional).
Type	Select User Credentials if creating credentials for Salesforce. This allows you to configure a specific system to enable a connection with your integration flow artifact.
User	Specify the username used to invoke the receiver system.
Password	Specify the password against which the user has to be authenticated.

5. Click **Deploy** to complete the process.

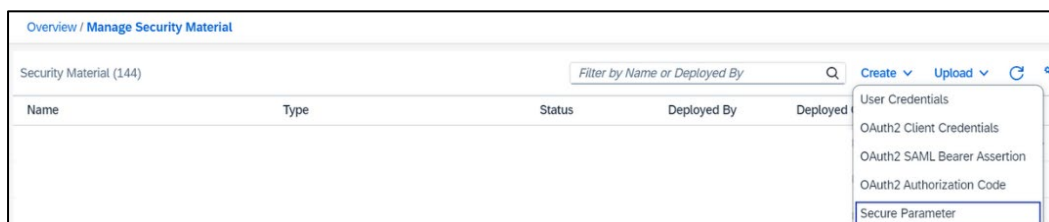
When you refresh the **Manage Security Material** page, the new artifact is displayed (with Type **Credentials**) in the artifact table.

3.3.1.2 Creating Secure Parameter

1. In SAP Integration Suite, navigate to **Monitor > Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Security** section and click **Security Material**.



3. On **Manage Security Material** page, click **Create** to select **Secure Parameter** from the dropdown.



4. In the **Create Secure Parameter** popup, provide the below details.

Parameter	Description
Name	Specify the name of the security artifact. The artifact name is used as an alias for the confidential data.
Description	Enter a description for the artifact (optional).
Secure Parameter	Enter the confidential value of the attribute. The permissible length of the secure parameter for Cloud Foundry is a maximum of 4096 characters.
Repeat Secure Parameter	Repeat the confidential value of the attribute.

5. Click **Deploy** to complete the process.

When you refresh the **Manage Security Material** page, the new artifact is displayed (with Type **Secure Parameter**) in the artifact table.

3.3.1.3 Adding Key Pair to Keystore

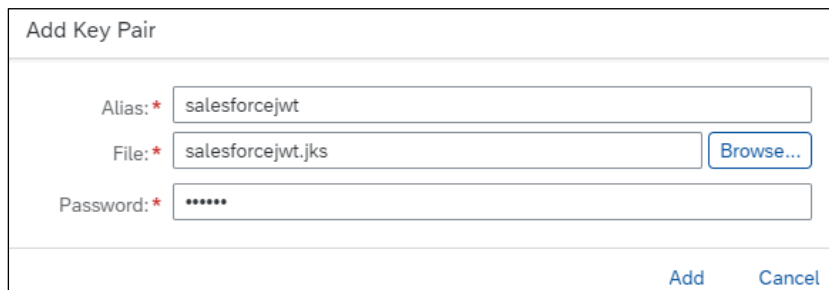
To add **Key Pair** to Cloud Integration Keystore, follow the steps below:

1. Open the **Monitor** Tab in SAP Cloud Integration.
2. Select **Keystore** in the **Manage Security** section.



Overview / Manage Keystore					
Current Backup New SAP keys SAP Key History					
Entries (12)					
Filter by Alias or DN					
Create Add Back Up Download					
Alias	Type	Valid Until	Last Modified At	Subject DN	Issuer DN

3. Select **Add** and choose **Key Pair**. Supply an Alias and the File and Password combination created in 6.2.2.



Add Key Pair

Alias: *

salesforcejwt

File: *

salesforcejwt.jks

Browse...

Password: *

Add

Cancel

4 Salesforce Service Cloud Adapter Configuration

This section defines the configurations for the Salesforce Service Cloud adapter. You need to configure the **Connection** and **Processing** tabs. A description and example usage for every field has been added.

In this section, you will learn how to configure the Salesforce Service Cloud receiver adapter. On selecting the Salesforce Service Cloud adapter from the list of adapters, you must configure the **Connection**, and **Processing** tabs.

4.1 Connection

The Connection tab contains connection and authentication parameters for Salesforce Service Cloud.

The Security artifact created in the previous section ([Creating Credentials in Security Material](#)) should be used in the **Connection tab** of the Adapter as shown in the figure below.

The screenshot shows the configuration window for the 'SalesforceServiceCloud' adapter. The 'Connection' tab is selected, showing fields for authentication and connection details. The 'Authentication' dropdown is set to 'OAuth 2.0 Client Credentials'. The 'Address' field is 'https://{MyDomain}.my.salesforce.com'. The 'OAuth2 Client Credentials' field is 'Client_Credentials_UC'. The 'Connection Timeout (in ms)' and 'Response Timeout (in ms)' are both set to '60000'. The 'Lazy Authentication' checkbox is checked.

SalesforceServiceCloud		
General	Connection	Processing
CONNECTION DETAILS		
Authentication:	OAuth 2.0 Client Credentials	
Address: *	https://{MyDomain}.my.salesforce.com	
OAuth2 Client Credentials: *	Client_Credentials_UC	
Connection Timeout (in ms):	60000	
Response Timeout (in ms):	60000	
Lazy Authentication:	☒	

The connection tab contains the following fields:

Parameter	Description
Authentication	Select the authentication type to connect to Salesforce Service Cloud: • OAuth2.0 Client Credentials • OAuth 2.0 JWT Bearer • OAuth 2.0 Username-Password
Address	• For OAuth2 Client Credentials: Specify the instance URL endpoint to perform the data calls. Example: <code>https://{MyDomain}.my.salesforce.com</code> if MyDomain is enabled on the org. • For OAuth 2.0 Username-Password: Specify the login endpoint to your Salesforce instance URL. Example: <code>https://login.salesforce.com</code> for the Salesforce production environment, <code>https://test.salesforce.com</code> for the Salesforce Sandbox environment and <code>https://{MyDomain}.my.salesforce.com</code> in case MyDomain is enabled on the org.
OAuth2 Client Credentials (When OAuth2 Client Credentials is selected)	Specify the User Credentials security artifact that stores the consumer key and consumer secret.
Audience (When OAuth 2.0 JWT Bearer is selected)	Specify the login endpoint to your Salesforce instance url. Example: <code>https://login.salesforce.com</code> for Salesforce production environment, <code>https://test.salesforce.com</code> for Salesforce Sandbox environment and <code>https://site.force.com/customers</code> if implementing for an Experience Cloud site.
Subject Alias (When OAuth 2.0 JWT Bearer is selected)	Specify the security artifact secure parameter that stores the username.

Parameter	Description
Issuer Alias (When OAuth 2.0 JWT Bearer is selected)	Specify the security artifact secure parameter that stores the OAuth client_id of the connected app in Salesforce for which the certificate was registered.
Keystore Alias (When OAuth 2.0 JWT Bearer is selected)	Specify the keystore alias for Client JWT token generation. It consists of the key and certificate to sign the JWT.
Expiration (in secs) (When OAuth 2.0 JWT Bearer is selected)	Specifies the validity of the assertion in seconds.
Basic Credential Name (When OAuth 2.0 Username-Password is selected)	Specify the security material that stores the basic credential name (username-password pair).
Security Token Alias (When OAuth 2.0 Username-Password is selected)	Specify the security artifact secure parameter that stores the security token. It can be omitted if your IP has been whitelisted on Salesforce.
OAuth Credential Name (When OAuth 2.0 Username-Password is selected)	Specify the security artifact user credentials that stores the Consumer key and Client secret.
Connection Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for the connection to be established.
Response Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for a response message to be received.

Parameter	Description
Lazy Authentication	Enable to delay OAuth token retrieval or generation until the first API call.

4.2 Processing

The Processing tab lists all the operations that can be performed through the adapter.

The screenshot shows the 'SalesforceServiceCloud' interface with the 'Processing' tab selected. The interface includes a top bar with 'General', 'Connection', and 'Processing' tabs. Below the tabs, there are sections for 'PROCESSING DETAILS', 'FORMAT', and 'HEADER DETAILS'. The 'PROCESSING DETAILS' section contains dropdowns for 'Configuration Type' (Basic), 'Type' (REST), 'Operation' (Salesforce Object - Create), 'API Version' (64.0), and 'Salesforce Object' (Account). The 'FORMAT' section has a 'Payload Format' dropdown set to 'Application/JSON'. The 'HEADER DETAILS' section has input fields for 'Request Headers' and 'Response Headers'.

Parameter	Description
Configuration Type	Select the required configuration type: Select Basic to use the dropdowns and parameter text fields, and Advanced to provide the relative URL. Default: Basic
Type	Select the type of API to be used: • Apex Call • REST • REST Bulk 2.0
Operation	Select the operation to be performed.
API Version	Specify the API Version of Salesforce.

Parameter	Description
Salesforce Object	Select the technical Salesforce object name for the data being processed without spaces (e.g. AccountPartner or CustomObject_c).
Salesforce Object ID	Specify the Salesforce object ID which identifies a specific record in Salesforce.
External ID Name	Specify the name of the external field of a specific Salesforce object.
External ID Value	Specify the value of the external field of a specific Salesforce object.
Filter Fields	Enable to apply filters to the Salesforce Object.
Name	Specify the name of the field to be returned in the response.
Method	Select the type of operation to be performed.
List of IDs	Specify the list of up to 200 IDs of objects to be deleted. Note: The IDs can belong to different object types, including custom objects.
All or None	Enable All or None (technical name is allOrNone) to roll back the entire request when the deletion of any object fails or disable it to continue with the independent deletion of other objects in the request.
SOQL Query	Specify the SOQL query to be executed.
SOSL Search	Specify the SOSL search to be executed.
Next Records URL	Specify the URL to retrieve the next batch of records. Example: <code>/services/data/v64.0/subjects/Account/0015Exxxx1pZixuQxx</code>
Method Name	Select the Apex REST method used to specify the Method annotation in the Apex Class.

Parameter	Description
Resource Relative URL (Only available when the Type is selected as Apex Call)	Specify the relative URL of the Apex requestURI. The relative path is the part of the requestURI after the instance URL and needs to start with /. Example: If the request URL is <code>https://instance.salesforce.com/services/apexrest/Account/123</code> , the relative URL will be <code>"/services/apexrest/Account/123"</code> .
Job ID	Specify the ID of the job.
Job Result Type	Select the type of records to be retrieved for the selected job ID. • Failed Results • None • Successful Results • Unprocessed Records
Advanced Job Properties	Enable to specify all the advanced properties in the adapter instead of in the payload.
Operation Name	Select the required operation.
Column Delimiter for Job	Select the column delimiter used for CSV job data: • Backquote Character (``) • Caret Character (^) • Comma Character (,) • Pipe Character () • SemiColon Character (;) • Tab Character
Content Type	Select the content type for the job: • CSV
Line Ending	Select the line ending used for CSV job data, marking the end of a data row: • CRLF—carriage return character • LF—linefeed character
Assignment Rule ID	Specify the ID of an assignment rule to run for a Case or a Lead. The assignment rule can be active or inactive.
External ID Field Name	Specify the external ID field in the object being updated. Only needed for upsert operations. Field values must also exist in CSV job data.

Parameter	Description
Job Type	Select to filter based on the specified job type: • BigObjectIngest • Classic • V2Ingest
Query Locator	Specify a queryLocator with a locator value to get a specific set of job results.
Query Job ID	Specify the Job ID of the Query.
Max Records	Specify the maximum number of records the adapter is expected to receive from Salesforce per page to prevent a timeout. Note: If working with a very large number of query results, then there is a possibility of a timeout before receiving all the data from Salesforce. Example: 2000
Sforce-Locator	Specify the Sforce-Locator to get the individual pages from the query results.
Auto Pagination	Enable to retrieve all records in case the query result has more than one page.
Enable PkChunking	Enable automatic primary key (PK) chunking. PK chunking splits bulk queries on large tables into chunks based on the record IDs, or primary keys, of the queried records.
Upload Job Data from body	Enable this property to create a job and upload all the data for a job using a multipart request.
Include deleted records in query result	Enable to include the records that have been deleted or merged as part of the results.
Handle XML Null Values	Enable to use null values in the payload to nullify properties in Salesforce.
Treat all XML Payload Values as String	Enable to treat all the values in the XML payload as string before sending it to Salesforce.

Parameter	Description
Enable Multiple Batch Processing	Enable to allow multiple round trips to Salesforce in case the number of records exceeds the allowed limit (200 records) for the SObject Collections. Note that this feature does not support allOrNone=true.
Enable Lookup Query	Enable this property to automatically resolve queries in the request body. Note: Enable Lookup Query supports only XML Payload Format.
Payload Format	Select the payload format of the request and response body.
HTTP Method (Only available when Configuration Type is set to Advanced)	Select the HTTP Method. • DELETE • GET • PATCH • POST • PUT
Relative URL (Only available when Configuration Type is set to Advanced)	Specify the endpoint path, excluding the Host. Example: <pre>/services/data/v48.0/subjects/ContentVersion/0000000000000000123/VersionData</pre>
Send Body (Only available when Configuration Type is set to Advanced and HTTP method is set to GET or DELETE)	Enable this checkbox if you want to send the body of the message with the request. For methods GET and DELETE, the body isn't sent by default.

Parameter	Description
Query	Specify the query. When using multiple parameters, ensure they are separated by "&". Example: <code>param1=value1&param2=value2</code>
Request Headers	Enter a list of custom headers, separated by a pipe (), to be sent to the target system. Use an asterisk (*) to send all custom headers to the target system. All Camel-specific headers and HTTP protocol headers except "date" are excluded by default even if you specify them.
Response Headers	Enter a list of headers, separated by a pipe (), coming from the target system's response to be received in the message. Use an asterisk (*) to receive all the headers from the target system, which is also the default value. All Camel-specific headers and HTTP protocol headers except "date" are excluded by default even if you specify them.

5 Salesforce Service Cloud Operations

This section lists and describes some of the operations supported by the Salesforce Service Cloud adapter. For detailed information on any operation, see [Salesforce Service Cloud documentation](#).

The receiver adapter supports two configuration types: **Basic** and **Advanced**. Basic allows users to quickly select predefined operations and endpoints using dropdowns and parameter fields, while Advanced provides flexibility to define custom operations by specifying the HTTP method and relative URL.

5.1 Salesforce Object: Create

Creates a new record for the selected object using the data provided in the request body. For more information, see [Create Records](#).

5.1.1 Basic

With the configuration type set to **Basic**, select the **Type** and **Operation** and enter the parameter values in the adapter UI.

Sample Body

```
{
  "Name": "John Doe",
  "BillingCity": "London",
  "SAPReferenceNumber__c": "123456"
}
```

A sample screenshot of the configuration is provided below.

The screenshot shows the configuration interface for the SalesforceServiceCloud adapter. It has three tabs: General, Connection, and Processing. The Processing tab is active. Under the heading 'PROCESSING DETAILS', there are several dropdown menus: Configuration Type (set to Basic), Type (set to REST), Operation (set to Salesforce Object - Create), API Version (set to 64.0), and Salesforce Object (set to Account). Below this, under the heading 'FORMAT', there is a dropdown for Payload Format (set to Application/JSON).

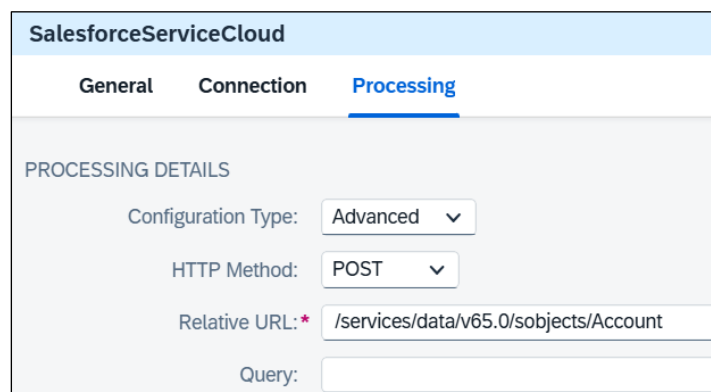
For the complete list of descriptions about the fields below, see [Processing](#).

Parameter	Value
Configuration Type	Select the configuration type as Basic .
Type	Select the type as REST .
Operation	Select the operation as Salesforce Object - Create .
API Version	Select the API Version.
Salesforce Object	Specify the Salesforce Object as Account .

5.1.2 Advanced

With the configuration type set to **Advanced**, users must choose the **HTTP method** and specify the **endpoint URL path**, excluding the host.

A sample screenshot of the configuration is provided below.



The screenshot shows the 'SalesforceServiceCloud' configuration interface. The 'Processing' tab is selected, showing 'PROCESSING DETAILS'. The 'Configuration Type' is set to 'Advanced', the 'HTTP Method' is 'POST', and the 'Relative URL' is '/services/data/v65.0/subjects/Account'. There is also a 'Query' field.

Parameter	Value
Configuration Type	Select Advanced as the configuration type.
HTTP Method	Select POST as the HTTP Method.
Relative URL	Specify the relative URL as /services/data/v65.0/subjects/Account

5.2 Salesforce Object: Composite

The Composite call allows you to execute a series of REST API requests in a single call. You can use the output of one request as the input to a subsequent request.

The response bodies and statuses of the requests are returned in a single response body. The entire series of requests counts as a single call toward your API limits. For more information, see [Composite Call](#).

Sample Body

```
{
  "compositeRequest": [
    {
      "method": "POST",
      "url": "/services/data/v64.0/subjects/Account",
      "referenceId": "refAccount",
      "body": {
        "Name": "Consultancy",
        "Phone": "+1-555-012-3456"
      }
    },
    {
      "method": "POST",
      "url": "/services/data/v64.0/subjects/Contact",
      "referenceId": "refContact",
      "body": {
        "LastName": "John",
        "AccountId": "@{refAccount.id}"
      }
    }
  ]
}
```

A sample screenshot of the configuration is provided below.

SalesforceServiceCloud

General **Connection** Processing

PROCESSING DETAILS

Configuration Type: Basic

Type: REST

Operation: Salesforce Objects - Composite

API Version: * 64.0

Treat all XML Payload Values as String: ☐

FORMAT

Payload Format: Application/JSON

HEADER DETAILS

Request Headers:

Response Headers: *

For the complete list of descriptions about the fields below, see [Processing](#).

Parameter	Value
Configuration Type	Select the configuration type as Basic .
Type	Select the type as REST .
Operation	Select the operation as Salesforce Object - Composite .
API Version	Select the API Version.

5.3 Job: Create

This request initiates a bulk operation by creating a job in Salesforce for asynchronous processing. You can provide the associated data either by submitting an **Upload Job Data** request or by including it directly within a **multipart create job** request. For more information, see [Create a Job](#).

Sample Body

```
FirstName,LastName,Description
Tom,Jones Jr.,Branding guru
Ian,O'Brian,Fuzzy logic expert
```

A sample screenshot of the configuration is provided below.

SalesforceServiceCloud
General
Connection
Processing

PROCESSING DETAILS

Configuration Type:
Basic

Type:
REST Bulk 2.0

Operation:
Job - Create

API Version:
64.0

Advanced Job Properties:
☒

Salesforce Object:
Account

Operation Name:
Insert

Column Delimiter for Job:
Comma character (,)

Content Type:
CSV

Line Ending:
LF—linefeed character

Assignment Rule ID:

Upload Job Data from body:
☒

FORMAT

Payload Format:
Application/XML

For the complete list of descriptions about the fields below, see [Processing](#).

Parameter	Value
Configuration Type	Select the configuration type as Basic .
Type	Select the type as REST Bulk 2.0
Operation	Select the operation as Job - Create .
API Version	Select the API Version.
Advanced Job Properties	Enable the checkbox.
Salesforce Object	Specify as <code>Account</code> .
Operation Name	Select as Insert .
Column Delimiter for Job	Select as Comma Character .
Content Type	Select as CSV .
Line Ending	Select as LF –linefeed character .
Upload Job Data from Body	Enable to use body.

5.4 Job Query: Get Results

The "**Get Results**" operation is the final stage of an asynchronous query workflow. Once the system has finished processing your data in the background and flagged the job as JobComplete, this operation **fetches the materialized data set associated with that specific Job ID**. For more information, see [Get Job Results](#).

Sample Body can be provided in XML format as well

```
<?xml version="1.0" encoding="UTF-8"?>
<createJobQuery>
  <id>7509000000kjySAQAY</id>
  <operation>query</operation>
  <object>Asset</object>
  <createdById>0057Q000000qbejQAA</createdById>
  <createdDate>2026-07-16T12:45:44.000+0000</createdDate>
  <systemModstamp>2026-07-16T12:45:44.000+0000</systemModstamp>
  <state>JobComplete</state>
  <concurrencyMode>Parallel</concurrencyMode>
  <contentType>CSV</contentType>
  <apiVersion>64.0</apiVersion>
  <columnDelimiter>CARET</columnDelimiter>
</createJobQuery>
```

A sample screenshot of the configuration is provided below.

SalesforceServiceCloud

General
Connection
Processing

PROCESSING DETAILS

Configuration Type: Basic

Type: REST Bulk 2.0

Operation: Job Query - Get Results

API Version: * 64.0

Max Records: \${header.maxrecord}

Auto Pagination:

Sforce-Locator:

Query Job ID: * \${property.JobID}

For the complete list of descriptions about the fields below, see [Processing](#).

Parameter	Value
Configuration Type	Select the configuration type as Basic .
Type	Select the type as REST Bulk 2.0
Operation	Select the operation as Job Query – Get Results .
API Version	Select the API Version.
Max Records	Specify as 2000.
Auto Pagination	Enable to retrieve all the records in the specified object in a single API Call.
Query Job ID	Specify the 7509000000kjySAQAY.

5.5 Apex Call: Post

The Apex call helps you to execute custom server-side business logic that cannot be handled by standard Salesforce declarative tools. For more information, see [Apex Call](#).

Sample Body

```
{
  "name" : "Wingo Ducks",
  "phone" : "707-555-1234",
  "website" : "www.wingo.ca.us"
```

```
}
```

A sample screenshot of the configuration is provided below.

The screenshot displays the 'SalesforceServiceCloud' configuration window with the 'Processing' tab selected. The 'PROCESSING DETAILS' section includes the following fields:

- Configuration Type: Basic
- Type: Apex Call
- Method Name: POST
- Resource Relative URL: * /services/apexrest/Account

The 'FORMAT' section includes the following field:

- Payload Format: Application/JSON

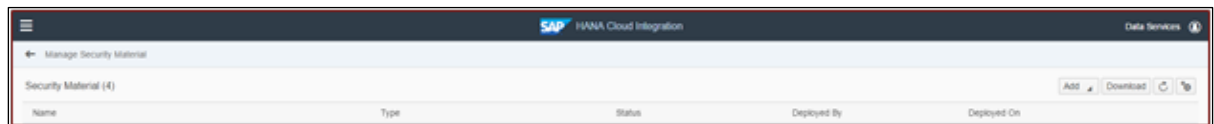
For the complete list of descriptions about the fields below, see [Processing](#).

Parameter	Value
Configuration Type	Select the configuration type as Basic .
Type	Select the type as Apex Call .
Method Name	Select as POST .
Resource Relative URL	Specify as /services/apexrest/Account.
Payload Format	Select as Application/JSON .

6 Reference

6.1 Initialize OAuth Username-Password in Salesforce

1. The Login URL is the login URL that is used by the authorization server of Salesforce. In the documentation of Salesforce, this URL is specified as <https://login.salesforce.com>. This link directs the Salesforce Service Cloud adapter to the login page of Salesforce to authorize.
2. The Basic Credential Name refers to an alias of a Username and Password combination stored in the SAP Cloud Integration Secure Store. This is the exact Username and Password used to log in to the Salesforce console.
3. Open the **Monitor** tab in SAP Cloud Integration.
4. Select **Security Material** in the section **Manage Security Material**.



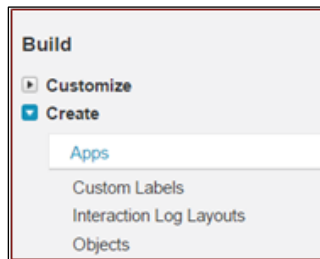
SAP Cloud Integration - Security Material

5. Select **Add**, select **User Credentials**, and enter a **Name** that should be the same as the Basic Credential Name configured in the Connection Tab of the Salesforce Service Cloud Adapter. In the **User** and **Password** fields, fill in the username and password used to log in to the Salesforce console. Click on **Deploy**.

Add User Credentials for Basic Authentication

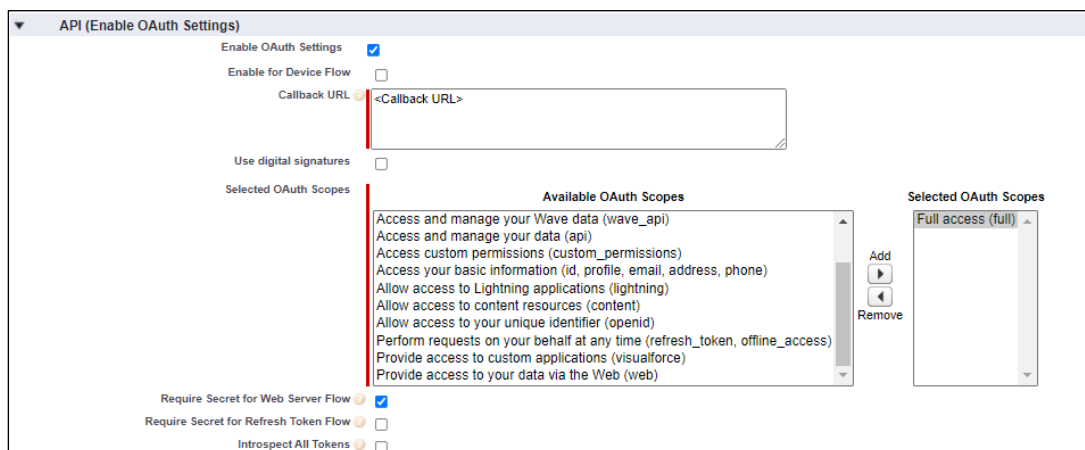
6. For the Security Token and OAuth Credentials, an app is required to be created in the Salesforce tenant. Login to the Salesforce console and select Setup.

- On the left panel of the **Build** overview, select **Create**. Click on **Apps** and select New in the Connected Apps section.



Create Salesforce App

- In the next screen, fill in basic details such as **App Name**, **API Name**, and **Contact Email**.
- In the **API (Enable OAuth Settings)**, select **Enable OAuth Settings**. Figure below displays a sample configuration of the OAuth Settings.
- Click **Save** to complete the creation of the app.



Salesforce API Enable OAuth Settings

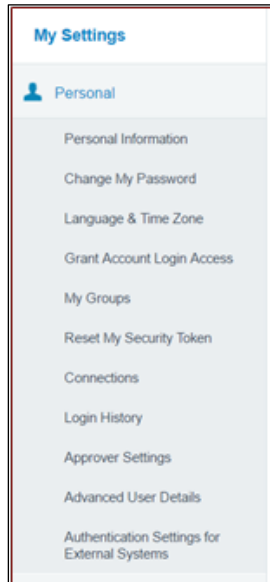
- The next overview displays what can be seen when a specific Connected App is selected. The Consumer Key and Consumer Secret can be seen in the respective Consumer Key and Consumer Secret fields.



OAuth Setting of the app

- Create a new Credential Name and fill in Consumer Key as User and Consumer Secret as Password as User Credentials. Deploy the **Consumer Key** and **Consumer Secret** as done in step 5.

13. In case the IP address of SAP Cloud Integration is not whitelisted in the settings of Salesforce, a Security Token is needed. If the Security Token has not been received via email yet, navigate to My Settings and select Personal. Select Reset My Security Token to receive a new security token.



Obtaining the Security Token

14. To deploy the Security Token, select **Security Material** in the section **Manage Security Material**. Select **Add** and then select **Secure Parameter**.
15. Fill in the Secure Parameter Name. Note that this should match the Security Token Alias used in the Salesforce Service Cloud Adapter. Fill in the Security Token in the **Secure Parameter** field and select **Deploy**.

A screenshot of the 'Add Secure Parameter' dialog box. It contains the following fields: '*Name:' with a placeholder '<Name>', 'Description:', '*Secure Parameter:' with a masked input field, and '*Repeat Secure Parameter:' with a masked input field. At the bottom right, there are 'Deploy' and 'Cancel' buttons.

Deploy Security Token as a Secure Parameter

16. Security Details successfully deployed in SAP Cloud Integration ensure a secure OAuth 2.0 Username-Password connection for the Salesforce Service Cloud Adapter. The names configured for Basic Credential Name, Security Token, and OAuth Credential Name must match the names of the deployed artifacts in the SAP Secure Stores. With the authentication mechanisms in place, the Salesforce Service Cloud Adapter can be configured for specific Salesforce integration scenarios.

6.2 Initialize OAuth JWT Bearer in Salesforce

When using the OAuth JWT Bearer, the following properties need to be maintained in the Connection Tab:

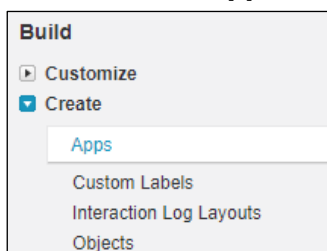
1. The Audience is the login URL used by the authorization server of Salesforce. In the documentation of Salesforce, this URL is specified as <https://login.salesforce.com>. This link directs the Salesforce Service Cloud Adapter to the login page of Salesforce.
2. The Subject Alias refers to an alias of a Secure Parameter stored in the SAP Cloud Integration Security Material. It specifies the username of the Salesforce user.
3. The Issuer Alias refers to an alias of a Secure Parameter stored in the SAP Cloud Integration Security Material. It specifies the OAuth Consumer Key of the connected app for which the certificate was registered.
4. The Keystore Alias refers to the added JKS file in Keystore as a Key Pair. It consists of a key and certificate to sign the JWT.

Refer to Salesforce [documentation](#) for more information.

6.2.1 Create a Connected App in Salesforce

To create a **Connected App**, follow the steps below:

1. Login to the Salesforce console and select **Setup**.
2. On the left panel in the Build overview, select **Create**. Click on **Apps** and select **New** for the **Connected Apps** section.



Create Salesforce App

3. In the next screen, fill in basic details such as **Connected App Name**, **API Name**, and **Contact Email**.
4. In the **API (Enable OAuth Settings)**, select **Enable OAuth Settings**.
5. Select **Use digital signatures** and upload Security Certificate(*.crt). The figure below shows a sample configuration of the OAuth Settings section.

API (Enable OAuth Settings)

6. Click on **Save** to complete the creation of the App.
7. In the next view, the **Consumer Key** can be seen in the respective field.

Salesforce OAuth Setting indicating Consumer Key of the App

8. In the created **App** overview, select **Manage** and choose **Edit Policies**.
9. Under **OAuth Policies**, change the **Permitted Users** to **Admin-approved users are pre-authorized**. See Figure below for a sample configuration.

OAuth Policies	
Permitted Users	Admin approved users are pre-authorized
Usage	View OAuth Usage
Single Logout	Single Logout disabled
Application Permissions:	Perform requests on your behalf at any time
	Full access

Salesforce App OAuth Policies

10. From the **Edit Policies** overview, select **Manage Profiles** then select System Administrator as shown in Figure below.

Salesforce App Manages Profiles

6.2.2 Create Security Certificate and JKS file using OpenSSL

To create Security Certificate and JKS file using OpenSSL, follow the steps below:

1. Generate a private key and store it in a file called server.key

OpenSSL Command:

```
openssl genrsa -out server.pass.key 2048  
openssl rsa -in server.pass.key -out server.key
```

2. Generate a certificate signing request by using the server.key file created in the previous step and store it in a file called server.csr. Enter the information about the organization when prompted.

OpenSSL Command:

```
openssl req -new -key server.key -out server.csr
```

3. Generate a self-signed digital certificate from the server.key and server.csr files and store the certificate in a file called server.crt. This server.crt is uploaded when creating the Connected App in Salesforce.

OpenSSL Command:

```
openssl x509 -req -sha256 -days 365 -in server.csr -signkey  
server.key -out server.crt
```

4. Generate Java Key Store (JKS) file as salesforcejwt.jks and name/alias as salesforcejwt by providing the files server.key and server.crt created in previous steps.

OpenSSL Command:

```
openssl pkcs12 -export -in server.crt -inkey server.key -out salesforcejwt.jks -name  
salesforcejwt  
Password:<PASSWORD>
```

For more information, see [Salesforce Documentation](#).