



# Google Pub/Sub Adapter for SAP Integration Suite

Version 1.2.1 – April 2026

# Contents

- 1 Introduction ..... 4
  - 1.1 Objective ..... 4
  - 1.2 Coding Samples ..... 4
  - 1.3 Internet Hyperlinks ..... 4
  - 1.4 Overview..... 4
  - 1.5 Features ..... 5
- 2 Installation and Configuration..... 6
  - 2.1 Adapter Installation on Cloud Foundry..... 6
    - 2.1.1 Prerequisite ..... 6
    - 2.1.2 Procedure ..... 6
      - 2.1.2.1 Adapter Installation by Creating a New Integration Flow ..... 6
      - 2.1.2.2 Adapter Installation without Creating a New Integration Flow.....7
    - 2.1.3 Monitor the Deployment Status ..... 8
- 3 Getting Started: Google Pub/Sub Adapter .....10
  - 3.1 Architecture Overview .....10
  - 3.2 Application Configuration..... 11
  - 3.3 Authentication.....12
    - 3.3.1 Set up Workload Identity Federation.....12
      - 3.3.1.1 Creating OAuth2 Client Credentials .....12
    - 3.3.2 Creating Keys for OAuth2 Service Account.....15
      - 3.3.2.1 Creating Keystore for JSON key .....15
      - 3.3.2.2 Creating Key-Pair for P12 key.....17
- 4 Google Pub/Sub Adapter Configuration .....19
  - 4.1 Sender Adapter .....19
    - 4.1.1 Sender (gRPC) .....19
      - 4.1.1.1 Connection Tab.....19
      - 4.1.1.2 Processing Tab.....22

4.1.2	Sender (Rest).....	24
4.1.2.1	Connection Tab.....	24
4.1.2.2	Processing Tab.....	27
4.2	Receiver Adapter .....	29
4.2.1	Connection Tab.....	29
4.2.2	Processing Tab.....	32
5	Google Pub/Sub Adapter Operations.....	36
5.1	Sender Adapter .....	36
5.1.1	gRPC.....	36
5.1.2	Rest.....	38
5.2	Receiver Adapter .....	39
5.2.1	Topics: Publish Message.....	39
5.2.1.1	Basic.....	39
5.2.1.2	Advanced.....	40
5.2.2	Subscriptions: Modify Acknowledgement Deadline.....	42
6	Troubleshooting.....	44
7	References.....	45
7.1	Keystore Generation (JKS File Creation).....	45
7.2	Configuring Post Processing Actions: Best Practices .....	50

# **1 Introduction**

## **1.1 Objective**

This is the official guide for the Google Pub/Sub Adapter for SAP Integration Suite. This guide covers all relevant information for integration developers to start working with the Google Pub/Sub 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**

In many integration scenarios, messages must be exchanged between applications or systems. The Google Pub/Sub adapter is a Cloud Integration capability available on SAP Integration Suite that facilitates message publishing and subscription service. Create your message-based integration solutions with ease using the Google Pub/Sub Adapter.

## 1.5 Features

Google Pub/Sub Adapter has the following features:

- Receiver adapter allows reliable **publishing** of messages to Google Pub/Sub. It offers single message (**Basic**) and multiple message publishing (**Advanced**) capabilities.
- Support for listing Topics with **pagination** using **Page Token**.
- Allows you to modify the acknowledgement deadline of a message using the **Modify Acknowledgement Deadline** operation.
- Sender adapter allows **consuming** messages from Google Pub/Sub. You can read messages in real time (**gRPC**) or using a specified polling interval (**REST**).
- Provides **multiple post-processing** options: *Acknowledge after Successful Processing, Acknowledge Immediately, and Do not Acknowledge*.
- Support for On-Premise Connectivity.
- Offers secure authentication via **OAuth2 Service Account** option (uses Key Pair authentication) and **Workload Identity Federation**.
- Support for **decoding data for incoming messages** in Sender Adapter (REST variant only).

## 2 Installation and Configuration

This section details the prerequisites to install and configure the Google Pub/Sub adapter.



The Google Pub/Sub adapter is available as part of your SAP Integration Suite license.

### 2.1 Adapter Installation on Cloud Foundry

Before the Google Pub/Sub adapter can be used in the Cloud Foundry environment, it must be deployed to the SAP Integration Suite tenant.

#### 2.1.1 Prerequisite

To deploy the Google PubSub adapter, you must have access to “*GooglePubSub Adapter for SAP Integration Suite*” as part of your SAP Integration Suite license.

#### 2.1.2 Procedure



The below installation procedure is compatible with Apache Camel 2, Apache Camel 3, and Edge Integration Cell (EIC) platform.

You can deploy the adapter using the following methods:

##### 2.1.2.1 Adapter Installation by Creating a New Integration Flow




The Google Pub/Sub adapter is available for selection in the Sender and Receiver adapter list and 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**


Go to **Design** workspace and select the integration package where you want to create a new Integration flow.

1. Click **Edit** to make the package editable.
2. Go to the **Artifacts** tab. Click **Add** and select **Integration Flow**.
3. Enter the **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.
4. Go to the newly created integration flow and click **Edit** to make it editable.
  - i) For the Sender, in the integration flow add a **Connector**  between the **Sender box** and the **Start**.
  - ii) For the Receiver, in the integration flow, click **End** to add a **Connector**  between the **End** and the **Receiver Box**.
5. A drop-down with the available adapters appears. The **GooglePubSub** adapter should show up in the list.
6. Select the **GooglePubSub** adapter from the list.

The adapter is now imported which *triggers* an adapter deployment. Once the Google Pub/Sub Adapter is deployed, a success message is *displayed*.

After the above steps are done, the Google Pub/Sub Adapter is successfully deployed in your **Design** workspace of the SAP Integration Suite tenant.

### 2.1.2.2 Adapter Installation without Creating a New Integration Flow

 The following procedure describes how the Google Pub/Sub 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 Google Pub/Sub adapter can be imported into the Design workspace without creating an integration flow.

Use the Transport Management Service (TMS) to import/transport the Google Pub/Sub adapter to a higher environment. Alternatively, if the TMS is not available in the landscape, the adapter package can be imported into the Design workspace by copying it from the Discover workspace.

## Purpose

To import the GooglePubSub adapter to **Design** workspace by copying the integration package from **Discover** workspace.

## Procedure

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

This completes the adapter deployment to **Design** workspace.

## 2.1.3 Monitor the Deployment Status

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 **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 shows a web interface for managing integration content. On the left, a sidebar titled "Integration Content (562)" contains a table with one entry: "GooglePubSub Integration Adapter" with a status of "Started". The main content area is titled "GooglePubSub" and includes a search bar, "Undeploy" and "Download" buttons, and deployment metadata: "Deployed On: Oct 29, 2024, 13:41:00", "Deployed By:", "ID:", "Version: 1.0.0", and "Package:". Below this, a "Status Details" section features a green notification bar stating "The Integration Adapter is deployed successfully."

# 3 Getting Started: Google Pub/Sub Adapter

Before you start using the adapter, you can get to know underlying [Architecture](#) for the adapter and [Application Configuration](#) for Google Pub/Sub.

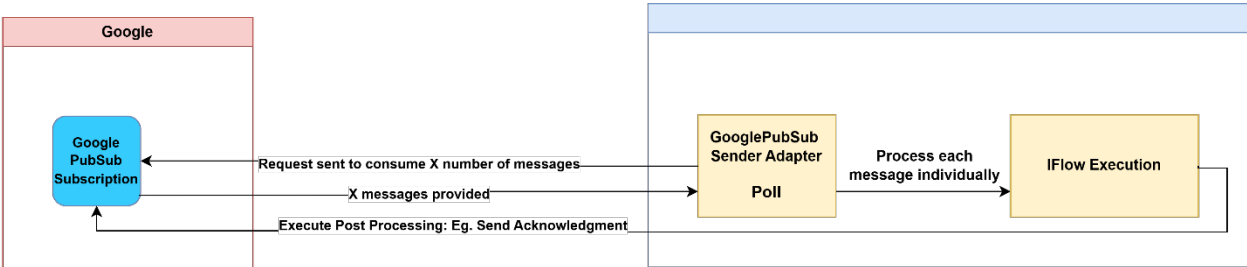
## 3.1 Architecture Overview

### How Google Pub/Sub Sender Adapter Works:

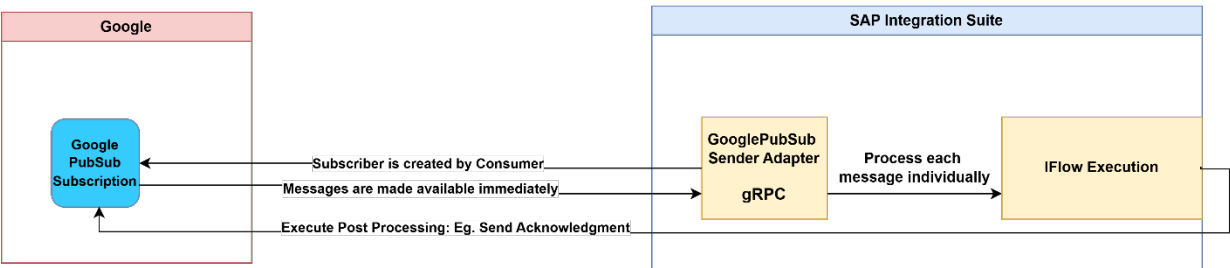
In a common message consumption/read scenario, sender adapter acts as the initiator of the calls.

The **Poll** variant sends a request to consume messages from a Google PubSub subscription. The messages are made available to the adapter on request. In case of **gRPC** variant, the sender is an active listener. After creating a subscriber, the messages are made available immediately.

Once the messages are available to the sender adapter, each message would be individually processed by the iFlow. Next, the post processing action is executed after IFlow processing is completed and consequently ACK or NACK is sent depending on the selection for Post Processing action.



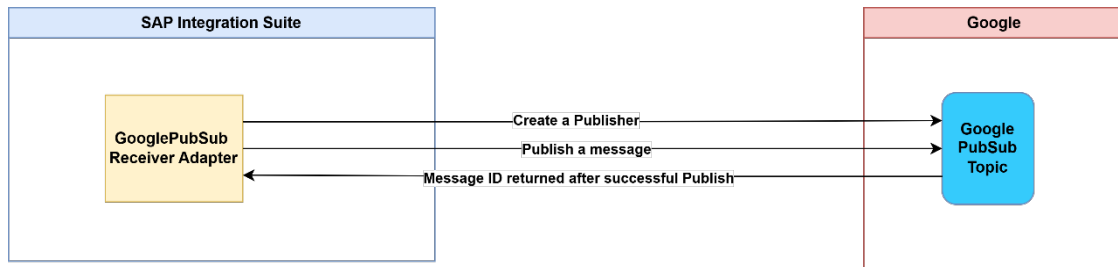
**Poll Variant**



**gRPC Variant**

## How Google Pub/Sub Receiver Adapter Works

In a common message publish scenario, the receiver adapter creates a publisher for a specific Google PubSub topic. The publisher sends a request for message publish to the topic. After successful publish, the message ID is returned by the topic.



## 3.2 Application Configuration

- To get an overview of Pub/Sub, see [Google Pub/Sub Overview](#).
- For more information about connecting to Pub/Sub, see [OAuth for Service Accounts](#).
- For more information about roles for your service account, see [Access Control](#).
- To learn more about Google Pub/Sub publish and subscribe, see [Publish & receive messages in Pub/Sub](#).

## 3.3 Authentication

The Google Pub/Sub adapter supports authentication using OAuth Service Account and Workload Identity Federation (WIF).

Find more information about how to create service account, see [Create Keys](#).

### 3.3.1 Set up Workload Identity Federation

- To learn more about WIF authentication, see [Workload Identity Federation](#).
- To set up WIF authentication, see WIF with Azure. You must create a security artifact of the type **OAuth2 Client Credentials**.
- For more information, see [Creating OAuth2 Client Credentials](#).

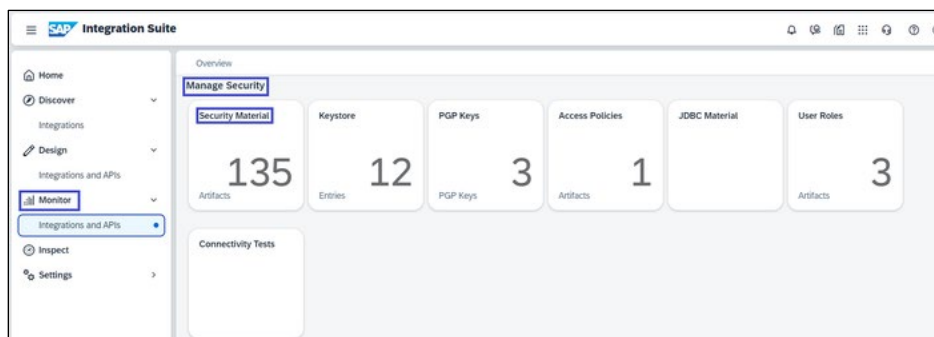
#### 3.3.1.1 Creating OAuth2 Client Credentials

##### Purpose

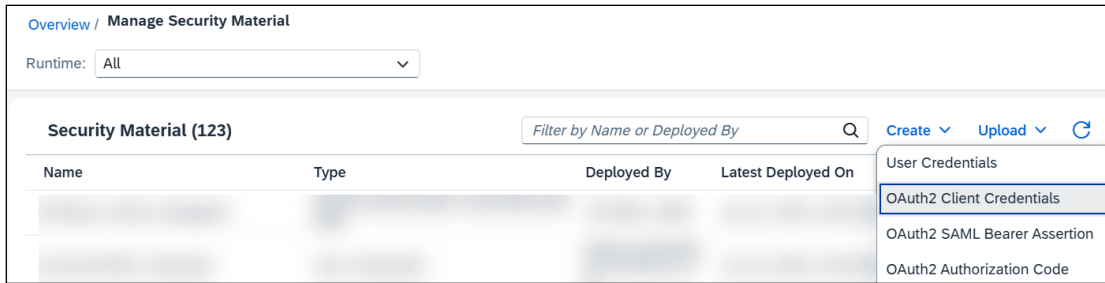
To create credentials in Security Material for **OAuth2 Client Credentials**.

##### Procedure

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 **OAuth2 Client Credentials** from the dropdown.



4. In the Create **OAuth2 Client Credentials** popup, provide the below details.

Parameter	Description
<b>Name</b>	Specify the name for the credentials.
<b>Description</b>	Specify the description for the credentials.
<b>Token Service URL</b>	Specify the URL of the OAuth2 authorization server that issues the access token.  Example: <code>https://login.microsoftonline.com/&lt;tenant ID&gt;/oauth2/token</code>
<b>Client ID</b>	Specify the ID of the client to which you are connecting.
<b>Client Secret</b>	Specify the Secret key of the client to which you are connecting.
<b>Client Authentication</b>	Select the Client Authentication which allows you to access an application using Client ID and Client Secret.  By default, the Send as Body Parameter is selected, this option sends the Client ID and Client Secret as a JSON content to the authentication server in the request body.  Example: <b>Send as Body Parameter</b>
<b>Scope</b>	Specify the OAuth2 scope information to be included in the request body.

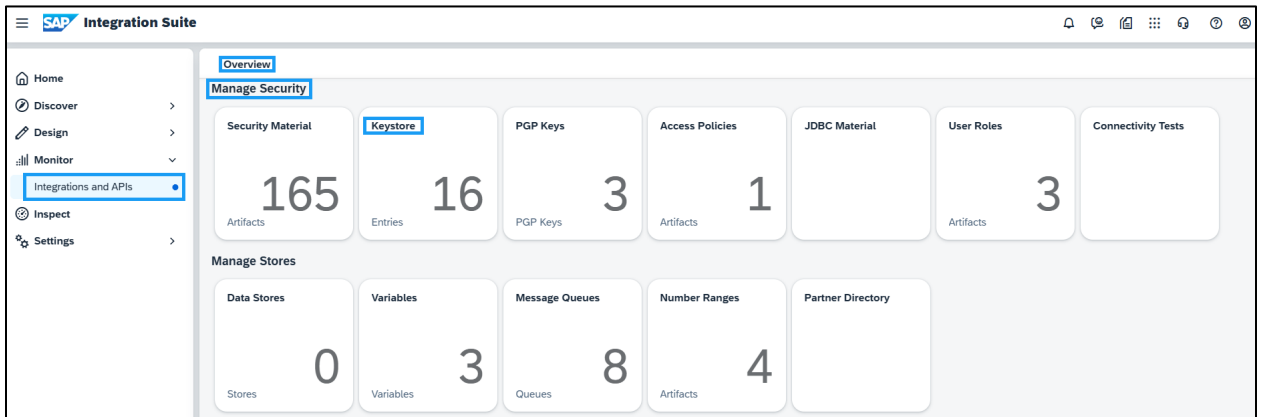
Parameter	Description
	Example: <a href="https://iam.googleapis.com/projects/&lt;projectid&gt;/locations/global/workloadIdentityPools/&lt;wiff-pool&gt;/providers/&lt;wif-provider&gt;/default">https://iam.googleapis.com/projects/&lt;projectid&gt;/locations/global/workloadIdentityPools/&lt;wiff-pool&gt;/providers/&lt;wif-provider&gt;/default</a>
<b>Content Type</b>	Select content type to indicate the media type.  <div style="background-color: #e0e0e0; padding: 10px; border: 1px solid #ccc;">  To use application permissions in the adapter, deploy an OAUTH2 Client Credential security artifact and ensure Content Type is set to '<b>application/x-www-form-urlencoded</b>' : </div>
<b>Resource</b>	Specify the identifier of the application or service that shares the same client secret. The identifier varies depending on the service that you want to connect with.
<b>Audience</b>	Specify the identifier of the application or service that shares the same client secret. The identifier varies depending on the service that you want to connect with.

- Click Deploy to complete the process.

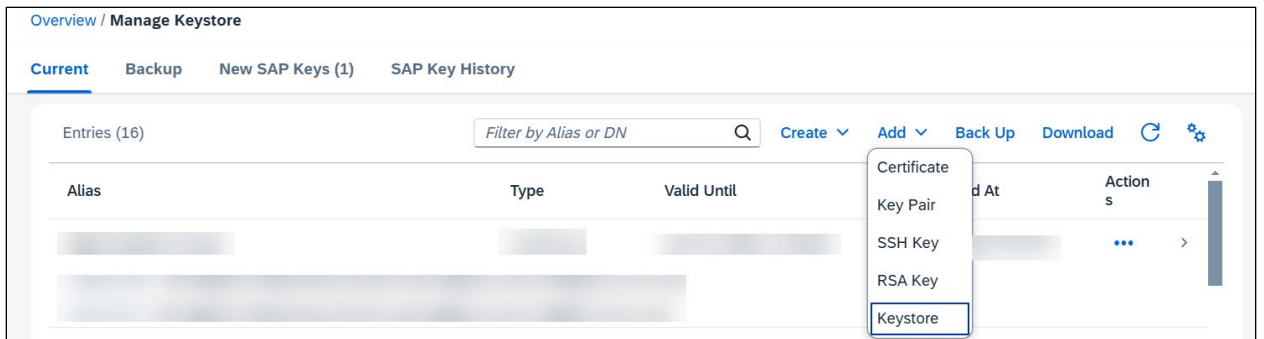
When you refresh the Manage Security Material page, the new artifact is displayed (with Type **OAuth2 Client Credentials**) in the artifact table.



2. Create a keystore file with `.jks` extension. For detailed information, see [Keystore Generation \(JKS file creation\)](#).
3. In SAP Integration Suite, navigate to **Monitor** > **Integrations and APIs**. This opens the **Overview** page.
4. On the **Overview** page, go to **Manage Security** section and click **Keystore**.



5. On **Manage Keystore** page, click **Add** to select **Keystore** from the dropdown.



6. In the **Add Keystore** popup, provide the below details.

### Add Keystore

Keystore: \*  Browse...

Passphrase: \*

Action: Add ▼

Overwrite existing entries

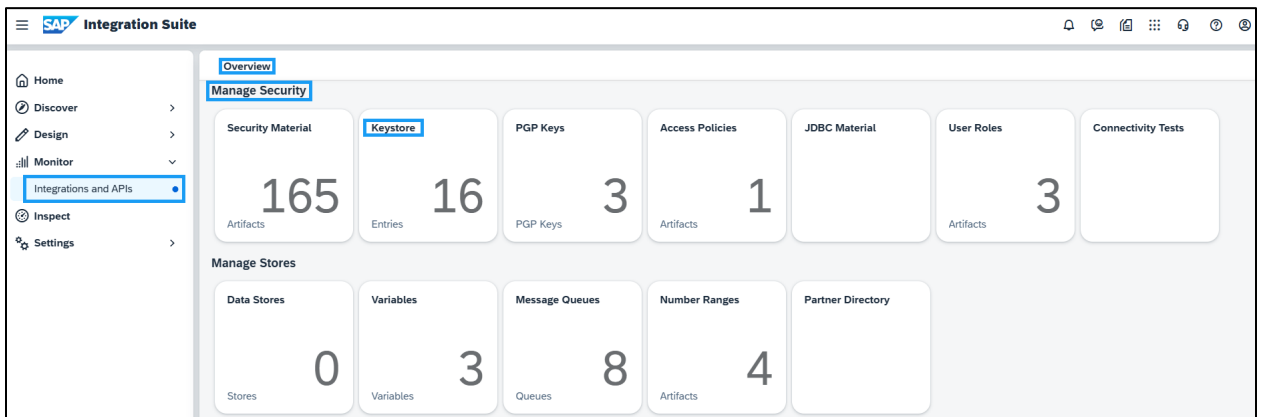
Deploy
Cancel

Parameter	Description
<b>Keystore</b>	Choose a keystore for upload. Select the .jks file. For detailed information, see <a href="#">Keystore Generation (JKS file creation)</a> .
<b>Passphrase</b>	Specify the password used while creating the .jks file.
<b>Action</b>	Specify the action to be performed as <b>Add</b> .
<b>Overwrite Existing Entries</b>	Enable or Disable depending on your need to overwrite existing entries.

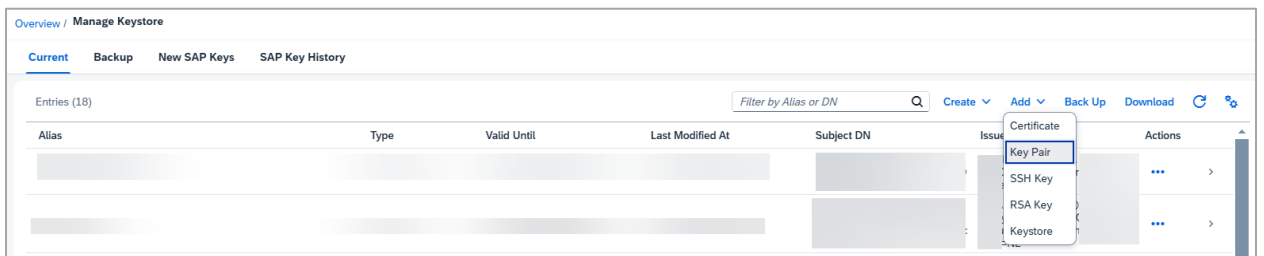
7. Click **Deploy** to complete the process.

### 3.3.2.2 Creating Key-Pair for P12 key

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 **Keystore**.



3. On **Manage Keystore** page, click **Add** to select **Key Pair** from the dropdown.



- In the **Key Pair** popup, provide the below details.

**Add Key Pair**

Alias: \*

File: \*

Password: \*

Parameter	Description
<b>Alias</b>	Specify the alias for the key pair.
<b>File</b>	Select the P12 key for upload.
<b>Password</b>	Specify the password used while creating the .jks file.

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

# 4 Google Pub/Sub Adapter Configuration

This section describes the parameters to be configured for your Google Pub/Sub sender and receiver adapters. You need to configure the **Connection** and **Processing** tabs.

A description and example usage for each field has been added.

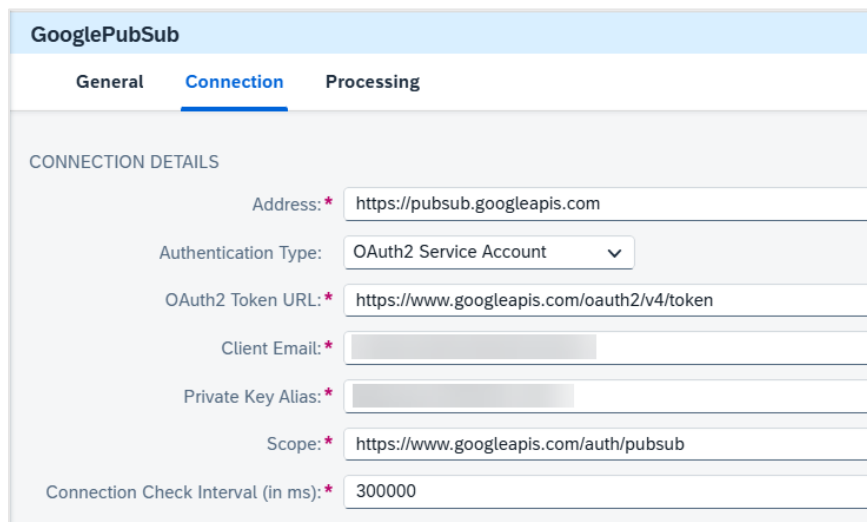
## 4.1 Sender Adapter

The Google Pub/Sub Sender Adapter is available as **gRPC** and **REST** variants.

### 4.1.1 Sender (gRPC)

#### 4.1.1.1 Connection Tab



The **Connection** tab contains connection and authentication parameters for Google Pub/Sub. Before you set the connection details, see [Creating a Keystore](#).






The screenshot shows the configuration interface for the GooglePubSub adapter, specifically the Connection tab. The interface has three tabs: General, Connection (selected), and Processing. Under the Connection tab, there is a section titled 'CONNECTION DETAILS' with the following fields:

Address: *	https://pubsub.googleapis.com
Authentication Type:	OAuth2 Service Account
OAuth2 Token URL: *	https://www.googleapis.com/oauth2/v4/token
Client Email: *	[Redacted]
Private Key Alias: *	[Redacted]
Scope: *	https://www.googleapis.com/auth/pubsub
Connection Check Interval (in ms): *	300000

The connection tab contains the following fields:

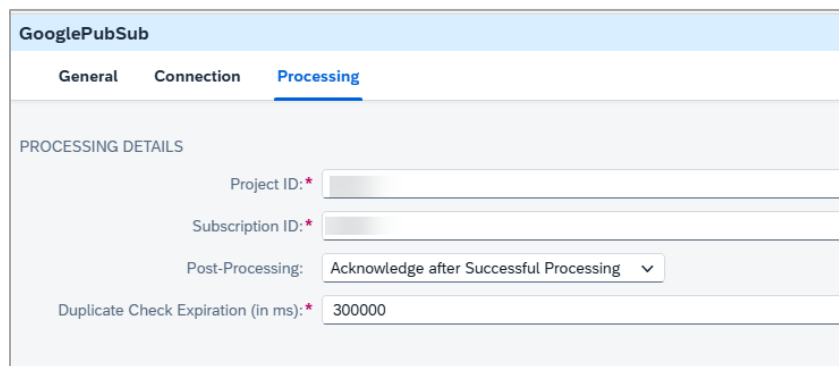
Parameter	Description
<b>Address</b>	Specify the hostname pointing to the Google Pub/Sub service.  Default: <a href="https://pubsub.googleapis.com">https://pubsub.googleapis.com</a>
<b>Authentication Type</b>	Select the method for authentication to Google Pub/Sub. <ul style="list-style-type: none"> <li>• <b>OAuth2 Service Account</b></li> <li>• <b>Workload Identity Federation</b></li> </ul>
 The below fields are applicable when <b>Authentication Type</b> is set to <b>OAuth2 Service Account</b> .	
<b>OAuth2 Token URL</b>	Specify the OAuth2 Token URL which identifies as the authorization server for producing a JWT token internally.  Default: <a href="https://www.googleapis.com/oauth2/v4/token">https://www.googleapis.com/oauth2/v4/token</a>
<b>Client Email</b>	Specify the Google Service Account Client Email.  Example: <a href="mailto:limited-svc-account@192843.iam.gserviceaccount.com">limited-svc-account@192843.iam.gserviceaccount.com</a>
<b>Private Key Alias</b>	Specify the Google Service Account Private Key for authentication.
<b>Scope</b>	Specify the scope of the connection to Google Pub/Sub service.  Default: <a href="https://www.googleapis.com/auth/pubsub">https://www.googleapis.com/auth/pubsub</a>
 The below fields are applicable when <b>Authentication Type</b> is set to <b>Workload Identity Federation</b> .	
<b>Microsoft Entra ID Credential Name</b>	Specify the OAuth2 Client Credentials security artifact created for Microsoft Entra ID.
<b>STS Token URL</b>	Specify the Security Token Service URL. Example: <a href="https://sts.googleapis.com/v1/token">https://sts.googleapis.com/v1/token</a>
<b>STS Audience</b>	Specify the audience i.e. the full resource name of the identity provider. Example: <code>//iam.googleapis.com/projects/&lt;project-number&gt;/locations/global/workloadIdentityPools/&lt;pool-id&gt;/providers/&lt;provider-id&gt;</code>

Parameter	Description
<b>STS Scopes</b>	Specify the OAuth 2.0 scopes to include on the resulting access token, formatted as a list of space-delimited, case-sensitive strings. Example: <code>https://www.googleapis.com/auth/cloud-platform</code>
<b>STS Options</b>	Specify the additional options to use in the STS token call.
<b>Use Service Account Impersonation</b>	Enable to use Service Account Impersonation.
 The below fields are applicable when <b>Use Service Account Impersonation</b> is enabled.	
<b>Access Token URL</b>	Specify the URL for generating access token with Service Account Impersonation.  Example: <code>https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/{ACCOUNT_EMAIL_OR_UNIQUEID}:generateAccessToken</code>   The - wildcard character is required; replacing it with a project ID is invalid.
<b>Service Account Impersonation Scopes</b>	Specify the comma-separated scopes to be included in the resulting OAuth 2.0 access token.  Example: <code>https://www.googleapis.com/auth/pubsub,https://www.googleapis.com/auth/cloud-platform</code>
<b>Delegates</b>	Specify the comma-separated sequence of service accounts in a delegation chain.  Example: <code>projects/-/serviceAccounts/{ACCOUNT_EMAIL_OR_UNIQUEID}</code>

Parameter	Description
 The below fields are applicable are available for all authentication methods.	
<b>Connection Check Interval (in ms)</b>	Specify the interval in milliseconds to verify the connection validity of gRPC.

### 4.1.1.2 Processing Tab



The **Processing** tab contains the operational configurations.




The screenshot shows the 'Processing' tab of the GooglePubSub configuration. It includes the following fields:

- Project ID:** A text input field with an asterisk indicating it is required.
- Subscription ID:** A text input field with an asterisk indicating it is required.
- Post-Processing:** A dropdown menu currently set to 'Acknowledge after Successful Processing'.
- Duplicate Check Expiration (in ms):** A text input field containing the value '300000'.

Parameter	Description
<b>Project ID</b>	Specify the Google Pub/Sub project ID.
<b>Subscription ID</b>	Specify the Google Pub/Sub Subscription ID from which the messages will be read.

Parameter	Description
<b>Post-Processing</b>	<p>Select the action to be performed after polling a message:</p> <ul style="list-style-type: none"> <li>• <b>Acknowledge after Successful Processing</b></li> <li>• <b>Acknowledge Immediately</b></li> <li>• <b>Do not Acknowledge</b></li> </ul> <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> For gRPC, <b>Do not Acknowledge</b> is only meant for testing and is not recommended for a production scenario.</p> </div> <p>For more information about best post processing practices, see <a href="#">Configuring Post Processing Actions: Best Practices</a>.</p>
<b>Duplicate Check Expiration (in ms)</b>	<p>Specify the expiry time in milliseconds while handling the Duplicate check.</p> <p>The default value is 300000.</p> <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Duplicate Check uses the Google PubSub message id to suppress the duplicates. This is only valid for success scenarios.</p> </div>

The message body for Sender Adapter adheres to a specific format. For more information regarding format, see below links:



- <https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.subscriptions/pull#ReceivedMessage> (For gRPC variant, `ackId` and `deliveryAttempt` are not applicable)
- <https://cloud.google.com/pubsub/docs/reference/rest/v1/PubsubMessage>

## 4.1.2 Sender (Rest)


### 4.1.2.1 Connection Tab



The screenshot shows the 'Connection' tab of the 'GooglePubSub' configuration. The 'CONNECTION DETAILS' section includes the following fields:




- Address:
- Authentication Type:
- Microsoft Entra ID Credential Name:
- STS Token URL:
- STS Audience:
- STS Scopes:
- STS Options:
- Use Service Account Impersonation:
- Polling Interval (in ms):
- Proxy Type:
- Reuse Connection:
- Connection Timeout (in ms):
- Response Timeout (in ms):

The **Connection** tab contains connection and authentication parameters for Google Pub/Sub. Before you set the connection details, see [Creating a Keystore](#).

The connection tab contains the following fields:

Parameter	Description
<b>Address</b>	<p>Specify the hostname pointing to the Google Pub/Sub service.</p> <p>Default: <a href="https://pubsub.googleapis.com">https://pubsub.googleapis.com</a></p> <div style="background-color: #e0f0ff; padding: 10px;"><p> When <b>Proxy Type</b> is set to <b>On-Premise</b>, you must use virtual host and port from cloud connector.</p></div>

Parameter	Description
<b>Authentication Type</b>	Select the method for authentication to Google Pub/Sub. <ul style="list-style-type: none"> <li>• <b>OAuth2 Service Account</b></li> <li>• <b>Workload Identity Federation</b></li> </ul>
 The below fields are applicable when <b>Authentication Type</b> is set to <b>OAuth2 Service Account</b> .	
<b>OAuth2 Token URL</b>	Specify the OAuth2 Token URL which identifies as the authorization server for producing a JWT token internally.  Default: <code>https://www.googleapis.com/oauth2/v4/token</code>
<b>Client Email</b>	Specify the Google Service Account Client Email.  Example: <a href="mailto:limited-svc-account@192843.iam.gserviceaccount.com">limited-svc-account@192843.iam.gserviceaccount.com</a>
<b>Private Key Alias</b>	Specify the Google Service Account Private Key for authentication.
<b>Scope</b>	Specify the scope of the connection to Google Pub/Sub service.  Default: <code>https://www.googleapis.com/auth/pubsub</code>
 The below fields are applicable when <b>Authentication Type</b> is set to <b>Workload Identity Federation</b> .	
<b>Microsoft Entra ID Credential Name</b>	Specify the OAuth2 Client Credentials security artifact created for Microsoft Entra ID.
<b>STS Token URL</b>	Specify the Security Token Service URL.  Example: <code>https://sts.googleapis.com/v1/token</code>
<b>STS Audience</b>	Specify the audience i.e. the full resource name of the identity provider.  Example: <code>//iam.googleapis.com/projects/&lt;project-number&gt;/locations/global/workloadIdentityPools/&lt;pool-id&gt;/providers/&lt;provider-id&gt;</code>

Parameter	Description
<b>STS Scopes</b>	Specify the OAuth 2.0 scopes to include on the resulting access token, formatted as a list of space-delimited, case-sensitive strings.  Example: <code>https://www.googleapis.com/auth/cloud-platform</code>
<b>STS Options</b>	Specify the additional options to use in the STS token call.
<b>Use Service Account Impersonation</b>	Enable to use Service Account Impersonation.
 The below fields are applicable when <b>Use Service Account Impersonation</b> is enabled.	
<b>Access Token URL</b>	Specify the URL for generating access token with Service Account Impersonation.  Example: <code>https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/{ACCOUNT_EMAIL_OR_UNIQUEID}:generateAccessToken</code>   The - wildcard character is required; replacing it with a project ID is invalid.
<b>Service Account Impersonation Scopes</b>	Specify the comma-separated scopes to be included in the resulting OAuth 2.0 access token.  Example: <code>https://www.googleapis.com/auth/pubsub,https://www.googleapis.com/auth/cloud-platform</code>
<b>Delegates</b>	Specify the comma-separated sequence of service accounts in a delegation chain.  Example: <code>projects/-/serviceAccounts/{ACCOUNT_EMAIL_OR_UNIQUEID}</code>
 The below fields are applicable are available for all authentication methods.	

Parameter	Description
<b>Polling Interval (in ms)</b>	Specify the Polling Interval in milliseconds.
<b>Proxy Type</b>	Select the proxy type: <ul style="list-style-type: none"> <li>• <b>Internet</b></li> <li>• <b>On-Premise</b></li> </ul>
<b>Location ID</b>	Specify the Location ID from Cloud Connector.
<b>Reuse Connection</b>	Enable to reuse this connection.
<b>Connection Timeout (in ms)</b>	Specify the maximum waiting time (in milliseconds) for the connection to be established with Google Pub/Sub service.
<b>Response Timeout (in ms)</b>	Specify the maximum waiting time (in milliseconds) for a response message to be received with Google Pub/Sub service.

#### 4.1.2.2 Processing Tab

**GooglePubSub**

General
Connection
Processing

PROCESSING DETAILS

Project ID: \*

Subscription ID: \*

Max Messages to Read: \*

Decode Data:




Post-Processing:  ▼

Duplicate Check Expiration (in ms): \*

HEADER DETAILS

Response Headers:

The **Processing** tab contains the operational configurations for the Google Pub/Sub Sender adapter.

Parameter	Description
<b>Project ID</b>	Specify the Google Pub/Sub project ID.
<b>Subscription ID</b>	Specify the Google Pub/Sub Subscription ID from which messages will be read.
<b>Max Messages to Read</b>	<p>Specify the maximum number of messages to be returned for this request. This must be a positive integer. The Google Pub/Sub system may return fewer than the number specified.</p> <p> Maximum number of messages allowed in a Pull response is 1000.</p>
<b>Decode Data</b>	<p>Enable to decode incoming data.</p> <p> The data field in the incoming message will be decoded as per Base64 format.</p>
<b>Post-Processing</b>	<p>Select the action to be performed after polling a message:</p> <ul style="list-style-type: none"> <li>• <b>Acknowledge after Successful Processing</b></li> <li>• <b>Acknowledge Immediately</b></li> <li>• <b>Do not Acknowledge</b></li> </ul> <p>For more information about best post processing practices, see <a href="#">Configuring Post Processing Actions: Best Practices</a>.</p>
<b>Duplicate Check Expiration (in ms)</b>	<p>Specify the expiry time in milliseconds while handling the Duplicate check. The default value is 300000ms.</p> <p> Duplicate Check uses the Google PubSub message id to suppress the duplicates. This is only valid for success scenarios.</p>
<b>Response Headers</b>	<p>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. Note that this value can also be read dynamically using an exchange header or property.</p>

The message body for Sender Adapter adheres to a specific format. For more information regarding format, see below links:



- <https://cloud.google.com/pubsub/docs/reference/rest/v1/projects.subscriptions/pull#ReceivedMessage>
- <https://cloud.google.com/pubsub/docs/reference/rest/v1/PubsubMessage>

## 4.2 Receiver Adapter




### 4.2.1 Connection Tab



The **Connection** tab contains connection and authentication parameters for Google Pub/Sub. Before you set the connection details, see [Creating a Keystore](#).


The screenshot shows the 'GooglePubSub' configuration interface with the 'Connection' tab selected. The form is titled 'CONNECTION DETAILS' and contains the following fields:

- Address: \*
- Authentication Type:
- Microsoft Entra ID Credential Name: \*
- STS Token URL: \*
- STS Audience: \*
- STS Scopes: \*
- STS Options:
- Use Service Account Impersonation:
- Proxy Type:
- Location ID: \*
- Reuse Connection:
- Connection Timeout (in ms):
- Response Timeout (in ms):

The connection tab contains the following fields:

Parameter	Description
<b>Address</b>	Specify the name for the Google Pub/Sub service.  Default: <a href="https://pubsub.googleapis.com">https://pubsub.googleapis.com</a>   When <b>Proxy Type</b> is set to <b>On-Premise</b> , you must use virtual host and port from cloud connector.
<b>Authentication Type</b>	Select the method for authentication to Google Pub/Sub: <ul style="list-style-type: none"> <li>• <b>OAuth2 Service Account</b></li> <li>• <b>Workload Identity Federation</b></li> </ul>
 The below fields are applicable when <b>Authentication Type</b> is set to <b>OAuth2 Service Account</b> .	
<b>OAuth2 Token URL</b>	Specify the OAuth2 Token URL which identifies as the authorization server for producing a JWT token internally.  Default: <a href="https://www.googleapis.com/oauth2/v4/token">https://www.googleapis.com/oauth2/v4/token</a>
<b>Client Email</b>	Specify the Google Service Account Client Email.  Example: <a href="mailto:limited-svc-account@192843.iam.gserviceaccount.com">limited-svc-account@192843.iam.gserviceaccount.com</a>
<b>Private Key Alias</b>	Specify the Google Service Account Private Key for authentication.
<b>Scope</b>	Specify the scope of the connection to Google Pub/Sub service.  Default: <a href="https://www.googleapis.com/auth/pubsub">https://www.googleapis.com/auth/pubsub</a>
 The below fields are applicable when <b>Authentication Type</b> is set to <b>Workload Identity Federation</b> .	
<b>Microsoft Entra ID Credential Name</b>	Specify the OAuth2 Client Credentials security artifact created for Microsoft Entra ID.
<b>STS Token URL</b>	Specify the Security Token Service URL. Example: <a href="https://sts.googleapis.com/v1/token">https://sts.googleapis.com/v1/token</a>



Parameter	Description
<b>STS Audience</b>	Specify the audience i.e. the full resource name of the identity provider. Example: <code>//iam.googleapis.com/projects/&lt;project-number&gt;/locations/global/workloadIdentityPools/&lt;pool-id&gt;/providers/&lt;provider-id&gt;</code>
<b>STS Scopes</b>	Specify the OAuth 2.0 scopes to include on the resulting access token, 0formatted as a list of space-delimited, case-sensitive strings. Example: <code>https://www.googleapis.com/auth/cloud-platform</code>
<b>STS Options</b>	Specify the additional options to use in the STS token call.
<b>Use Service Account Impersonation</b>	Enable to use Service Account Impersonation.
 The below fields are applicable when <b>Use Service Account Impersonation</b> is enabled.	
<b>Access Token URL</b>	Specify the URL for generating access token with Service Account Impersonation.  Example: <code>https://iamcredentials.googleapis.com/v1/projects/-/serviceAccounts/{ACCOUNT_EMAIL_OR_UNIQUEID}:generateAccessToken</code>   The - wildcard character is required; replacing it with a project ID is invalid.
<b>Service Account Impersonation Scopes</b>	Specify the comma-separated scopes to be included in the resulting OAuth 2.0 access token.  Example: <a href="https://www.googleapis.com/auth/pubsub">https://www.googleapis.com/auth/pubsub</a> , <a href="https://www.googleapis.com/auth/cloud-platform">https://www.googleapis.com/auth/cloud-platform</a>

Parameter	Description
<b>Delegates</b>	Specify the comma-separated sequence of service accounts in a delegation chain.  Example:  <code>projects/- /serviceAccounts/{ACCOUNT_EMAIL_OR_UNIQUEID}</code>
 The below fields are applicable are available for all authentication methods.	
<b>Proxy Type</b>	Select the proxy type: <ul style="list-style-type: none"> <li>• <b>Internet</b></li> <li>• <b>On-Premise</b></li> </ul>
<b>Location ID</b>	Specify the Location ID from Cloud Connector.
<b>Reuse Connection</b>	Enable this property to reuse the connection.
<b>Connection Timeout (in ms)</b>	Specify the maximum waiting time (in milliseconds) for the connection to be established with Google Pub/Sub service.
<b>Response Timeout (in ms)</b>	Specify the maximum waiting time (in milliseconds) for a response message to be received with Google Pub/Sub service.

## 4.2.2 Processing Tab

The **Processing** tab contains the operational configurations for the Google Pub/Sub receiver adapter.

Parameter	Description
<b>Entity</b>	Select the entity for message processing: <ul style="list-style-type: none"> <li>• <b>Topics</b></li> <li>• <b>Subscriptions</b></li> </ul>
<b>Operation</b>	Select the operation to be performed.
<b>Project ID</b>	Specify the Google Pub/Sub project ID.
<b>Topic ID</b>	Specify the Topic ID to be used.
<b>Operation Type</b>	Select the type of operation <ul style="list-style-type: none"> <li>• <b>Basic</b></li> <li>• <b>Advanced</b></li> </ul> <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p><b>i</b> In case of multiple (Advanced) mode, body must be set in the expected JSON format before using this adapter.</p> </div>
<b>Encode Data</b>	Enable to encode data to Base64. <div style="background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p><b>i</b> Google Pub/Sub can only accept Base64 encoded data. If not enabled data must be converted to Base64 before using the adapter.</p> </div>

Parameter	Description
<b>Ordering Key</b>	<p>Specify the ordering key for this message.</p> <p> For subscriptions with message ordering enabled, messages with the same ordering key are sent in the order in which they were published.</p>
<b>Subscription ID</b>	Specify the Subscription ID.
<b>Acknowledgement ID</b>	Specify the Acknowledgement ID for the messages being used.
<b>Acknowledgement Deadline (in sec)</b>	Specify the new acknowledgement deadline with respect to the time when this request is sent to the Google Pub/Sub system.
<b>Page Size</b>	<p>Specify the maximum number of topics to be returned.</p> <p>Example: 10</p>
<b>Page Token</b>	<p>Specify the value returned by the last ListTopicsResponse; indicates that this is a continuation of a prior topics.list call, and that the system should return the next page of data.</p> <p> A value of null or an empty string retrieves the first page.</p>
<b>Message Attributes</b>	<p>Specify multiple message attributes as key value pairs.</p> <p>Example: Set <b>Key</b> to <code>status</code> and <b>Value</b> to <code>end_of_message</code></p>
<b>Request Headers</b>	<p>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. Note that this value can also be read dynamically using an exchange header or property.</p>

Parameter	Description
<b>Response Headers</b>	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. Note that this value can also be read dynamically using an exchange header or property.

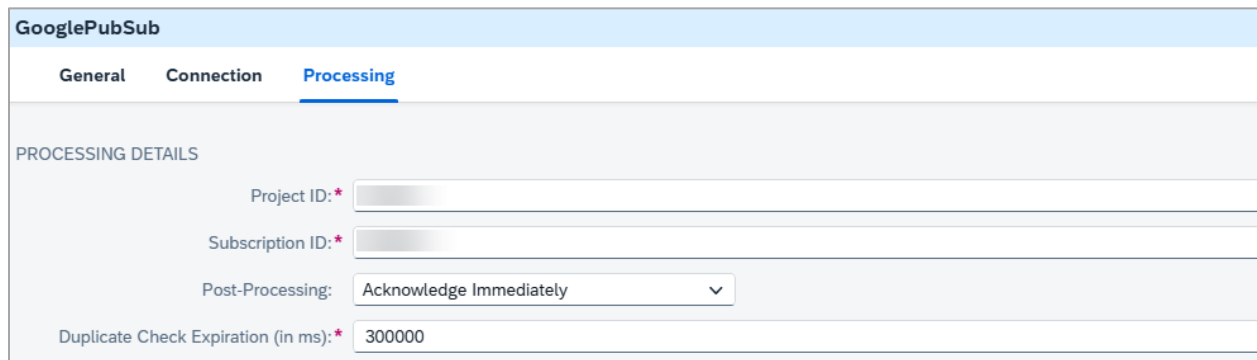
# 5 Google Pub/Sub Adapter Operations

This section describes some of the operations supported by the Google Pub/Sub adapter.

## 5.1 Sender Adapter

You can read messages from a subscribed topic using the Google Pub/Sub adapter. For more information, see [Subscriptions](#).



### 5.1.1 gRPC



The screenshot shows the 'Processing' tab of the 'GooglePubSub' configuration. It includes the following fields:

- Project ID:** A text input field with a red asterisk indicating it is required.
- Subscription ID:** A text input field with a red asterisk indicating it is required.
- Post-Processing:** A dropdown menu currently set to 'Acknowledge Immediately'.
- Duplicate Check Expiration (in ms):** A text input field containing the value '300000'.

Parameter	Values
<b>Project ID</b>	Specify the ID for your Project.
<b>Subscription ID</b>	Specify the Subscription ID associated to your Project.

Parameter	Values
<p><b>Post-Processing</b></p>	<p>Select the action to be performed after polling a message:</p> <ul style="list-style-type: none"> <li>• <b>Acknowledge after Successful Processing</b></li> <li>• <b>Acknowledge Immediately</b></li> <li>• <b>Do not Acknowledge</b></li> </ul> <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> For gRPC, <b>Do not Acknowledge</b> is only meant for testing and is not recommended for a production scenario.</p> </div> <p>For more information about best post processing practices, see <a href="#">Configuring Post Processing Actions: Best Practices</a>.</p>
<p><b>Duplicate Check Expiration (in ms)</b></p>	<p>Specify the expiry time in milliseconds while handling the Duplicate check.</p> <p>The default value is 300000.</p> <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> Duplicate Check uses the Google PubSub message id to suppress the duplicates. This is only valid for success scenarios.</p> </div>

## 5.1.2 Rest

**GooglePubSub**

General   Connection   **Processing**

---

PROCESSING DETAILS

Project ID: \*

Subscription ID: \*

Max Messages to Read: \*

Decode Data:



Post-Processing:  ▾

Duplicate Check Expiration (in ms): \*

---

HEADER DETAILS

Response Headers:

Parameter	Values
<b>Project ID</b>	Specify the ID for your Project.
<b>Subscription ID</b>	Specify the Subscription ID associated to your Project.
<b>Max Messages to Read</b>	<p>Specify the maximum number of messages to be returned for this request. This must be a positive integer. The Google Pub/Sub system may return fewer than the number specified.</p> <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #ccc;"> <p> Maximum number of messages allowed in a Pull response is 1000.</p> </div> <p>Example: 10</p>
<b>Decode Data</b>	<p>Enable to decode incoming data.</p> <div style="background-color: #e6f2ff; padding: 5px; border: 1px solid #ccc;"> <p> Google PubSub sends data in Base 64 encoded format.</p> </div>

Parameter	Values
<b>Post-Processing</b>	<p>Select the action to be performed after polling a message:</p> <ul style="list-style-type: none"> <li>• <b>Acknowledge after Successful Processing</b></li> <li>• <b>Acknowledge Immediately</b></li> <li>• <b>Do not Acknowledge</b></li> </ul> <p>For more information about best post processing practices, see <a href="#">Configuring Post Processing Actions: Best Practices</a>.</p>
<b>Duplicate Check Expiration (in ms)</b>	Set value is 300000




## 5.2 Receiver Adapter

### 5.2.1 Topics: Publish Message

For more information about this operation, see [Publish Message Topics](#).

#### 5.2.1.1 Basic

Parameter	Values
<b>Entity</b>	Select <b>Topics</b> from the available dropdown.
<b>Operation</b>	Select Operation as <b>Publish Message</b>

Parameter	Values
<b>Project ID</b>	Specify the Google Pub/Sub project ID.
<b>Topic ID</b>	Specify the Topic ID to be used.
<b>Operation Type</b>	<p>Select the type of operation</p> <ul style="list-style-type: none"> <li>• <b>Basic</b></li> <li>• <b>Advanced</b></li> </ul> <p> Basic allows you to specify configurations on the adapter UI.</p>
<b>Encode Data</b>	<p>Enable this field to encode data to Base64.</p> <p> Google Pub/Sub can only accept Base64 encoded data. If not enabled data must be converted to Base64 before using the adapter.</p>
<b>Ordering key</b>	<p>Specify the ordering key for this message.</p> <p> For subscriptions with message ordering enabled, messages with the same ordering key are sent in the order in which they were published.</p> <p>Example: <code>ord-key-3904</code></p>
<b>Message Attributes</b>	<p>Specify message attributes as per your requirement.</p> <p>Set Key to <code>status</code> and Value to <code>end_of_message</code>.</p>

**5.2.1.2 Advanced**

**GooglePubSub**

General   Connection   **Processing**

PROCESSING DETAILS

Entity: Topics

Operation: Publish Message

Project ID: \* \${property.ProjectID}

Topic ID: \* \${property.Topic\_ID}

Operation Type: Advanced


HEADER DETAILS

Request Headers:

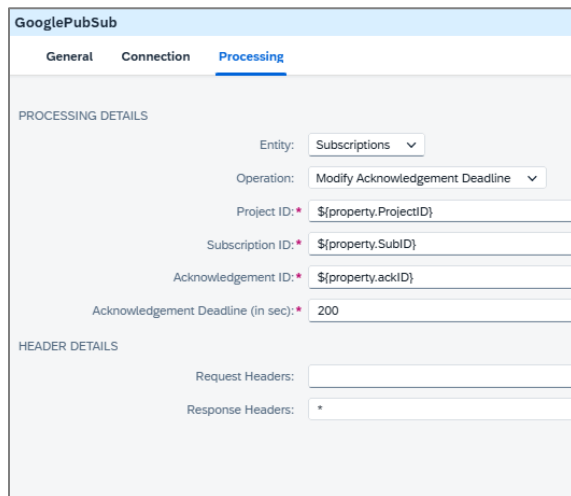
Response Headers: \*

### Sample Payload:

```
{
  "messages": [
    {
      "data": "dFeWzdC1kYXRh",
      "attributes" : {},
      "orderingKey": "1"
    },
    {
      "attributes" : {
        "data1" : "ERfsdS",
        "data2" : "eqwWRfs"
      }, "data" : "dTVzWCR1kYXRh",
      "orderingKey": "1"
    },
    {
      "data": "",
      "attributes" : {
        "data3" : "value"
      },
      "orderingKey": "1"
    }
  ]
}
```


Parameter	Values
<b>Entity</b>	Select <b>Topics</b> from the available dropdown.
<b>Operation</b>	Select Operation as <b>Publish Message</b>
<b>Project ID</b>	Specify the Google Pub/Sub project ID.
<b>Topic ID</b>	Specify the Topic ID to be used.
<b>Operation Type</b>	<p>Select the type of operation:</p> <ul style="list-style-type: none"> <li>• <b>Basic</b></li> <li>• <b>Advanced</b></li> </ul> <div style="border: 1px solid #ccc; background-color: #e6f2ff; padding: 5px; margin-top: 10px;"> <p> In case of multiple (Advanced) mode, body must be set in the expected JSON format before using this adapter.</p> </div>

## 5.2.2 Subscriptions: Modify Acknowledgement Deadline



The screenshot shows the 'Processing' tab of the GooglePubSub configuration interface. Under 'PROCESSING DETAILS', the 'Entity' is set to 'Subscriptions' and the 'Operation' is 'Modify Acknowledgement Deadline'. The 'Project ID' is set to '\$(property.ProjectID)', 'Subscription ID' to '\$(property.SubID)', and 'Acknowledgement ID' to '\$(property.ackID)'. The 'Acknowledgement Deadline (in sec)' is set to '200'. Under 'HEADER DETAILS', the 'Request Headers' and 'Response Headers' fields are empty.

Parameter	Values
<b>Entity</b>	Select <b>Subscriptions</b> from the available dropdown.
<b>Operation</b>	Select Operation as <b>Modify Acknowledgement Deadline</b>
<b>Project ID</b>	Specify the Google Pub/Sub project ID.
<b>Subscription ID</b>	Specify the Topic ID to be used.

Parameter	Values
<b>Acknowledgement ID</b>	Specify the new acknowledgement deadline with respect to the time when this request is sent to the Google Pub/Sub system.
<b>Acknowledgement Deadline (in sec)</b>	Specify the new acknowledgement deadline with respect to the time when this request is sent to the Google Pub/Sub system.  Example: 200   This value ranges from 0 to 600. For more information, see <a href="#">modifyAckDeadline</a> .

## 6 Troubleshooting

A few issues that you might encounter and possible solutions for them have been documented below.

Error	Error Message	Possible Solution
Publish Message Failure if <b>Encode Data</b> is unchecked and exchange body has invalid characters.	<pre>"message": "Invalid value at 'messages[0].data' (TYPE_BYTES), Base64 decoding failed for \"bkZpZWxkIjogdHJ1ZX0=\r\n\r\n\""</pre>	You must ensure encoded data being sent is properly encoded, check and remove invalid characters.

# 7 References

## 7.1 Keystore Generation (JKS File Creation)

1. Generate a certificate signing request by using the `gps_rsa_key.p8` file and store it in a file called `gps_server.csr`.

### OpenSSL Command:

```
openssl req -new -key <source_path>/gps_rsa_key.p8 -out  
<target_path>/gps_server.csr
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:<enter\_country\_code>

State or Province Name (full name) [Some-State]:<enter\_city\_name>

Locality Name (eg, city) []:<enter\_locality\_name>

Organization Name (eg, company) [Internet Widgits Pty Ltd]:<enter\_org\_name>

Organizational Unit Name (eg, section) []:<enter\_org\_unit\_name>

Common Name (e.g. server FQDN or YOUR name) []:<enter\_common\_name>

Email Address []:<enter\_email\_address>

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:<password>

An optional company name []:<company\_name>

2. It is possible to sign the certificate using a self-signing mechanism or via CA.



Please refer to your organisation policy to choose a suitable signing mechanism for the certificate.

You can generate a self-signed digital certificate using our private key

(`gps_rsa_key.p8`) and certificate signing request (`gps_server.csr`) file and store the certificate in a file called (`gps_server.crt`).

This `gps_server.crt` is uploaded when creating the jks file.

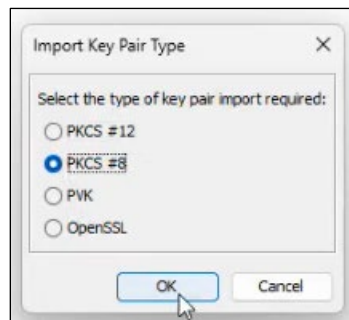
### OpenSSL Command:

```
openssl x509 -req -sha256 -days 365 -in  
<source_path>/gps_server.csr -signkey  
<source_path>/gps_rsa_key.p8 -out <target/path>/gps_server.crt
```

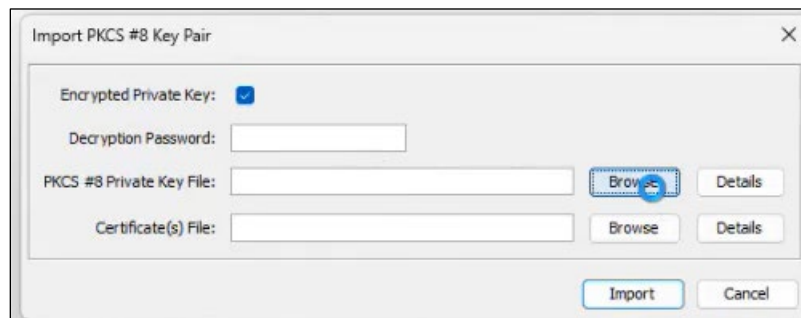
3. To create a JKS file using Keystore in Windows follow the below steps:
  - i. Go to Keystore and **Import key Pair**.

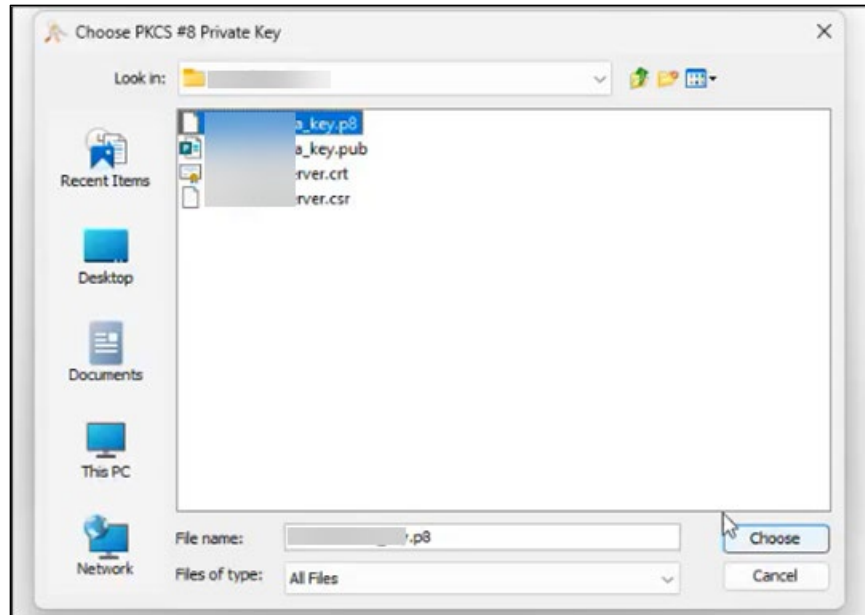


- ii. Select the type of key pair as **PKCS #8**.

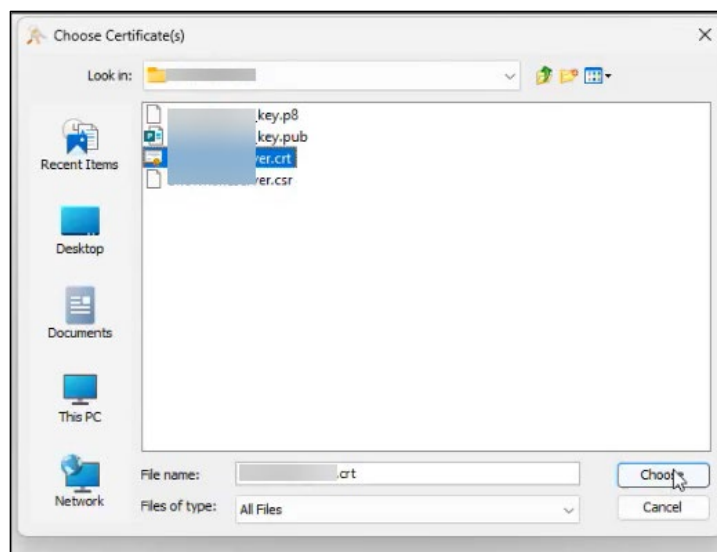
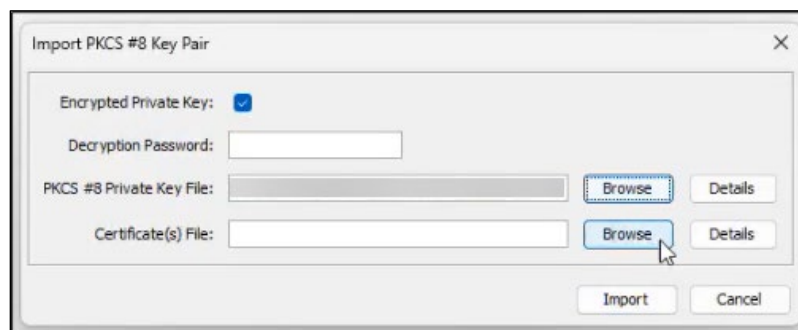


- iii. Add the private key file by uploading the PKCS **Private Key** file.

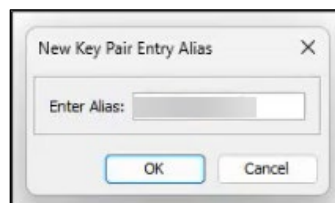
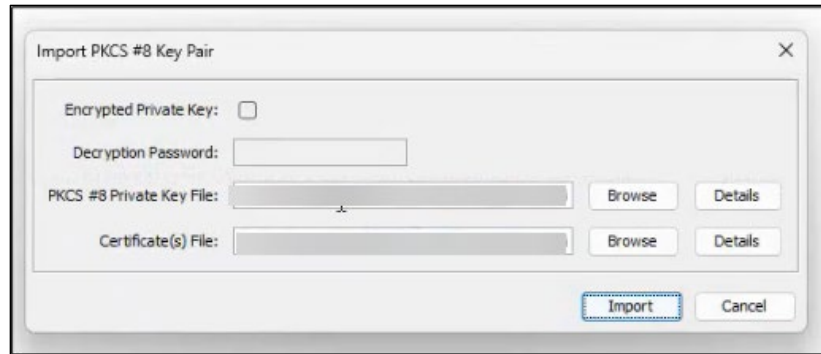




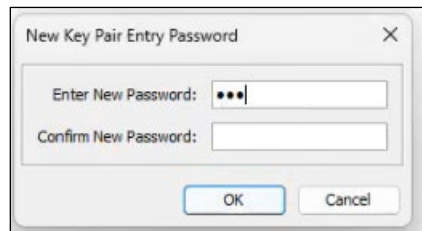
iv. Add the Google PubSub **Certificate(s) File** created above.



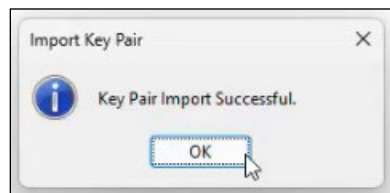
- v. Click **Import** and provide the **Alias**.



- vi. Provide a **New Key Pair Entry Password**.



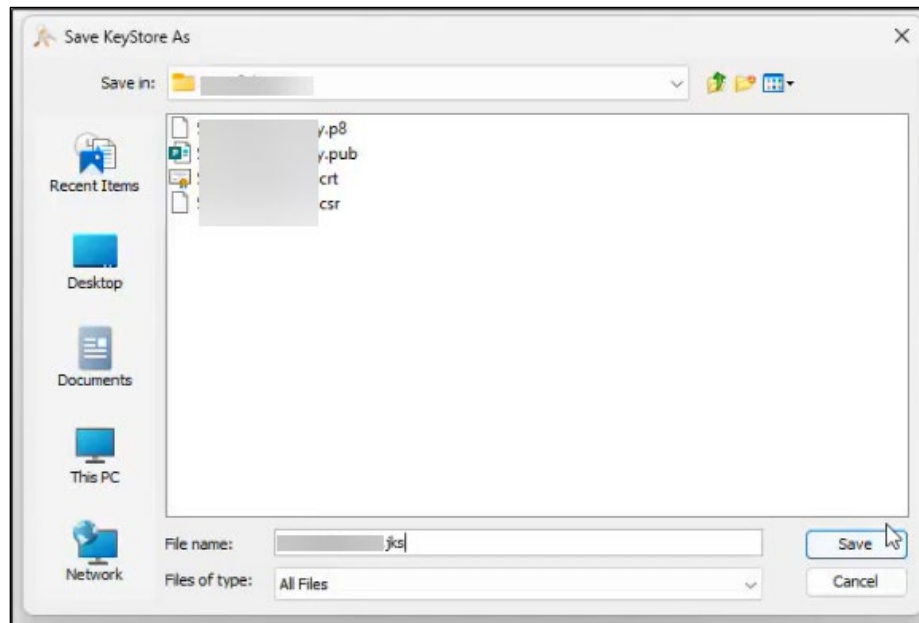
- vii. You will get a message that indicates key pair import was successful.



viii. Provide the **KeyStore Password**.




ix. Finally, save the .jks file.



## 7.2 Configuring Post Processing Actions: Best Practices

This section clarifies the post processing behaviour and best practices for gRPC and REST variants.

- Post processing is executed only after the iFlow execution is completed.
-  In case of gRPC, the ack deadline configured in the Google Subscription is not relevant. Even if the ack deadline is expired, Google will not send the message again to the subscriber and would wait for ACK or NACK.

- **Acknowledge Immediately**


Once the message is received by the adapter, message processing or execution can begin. After iFlow processing is completed, an ACK is sent to Google and the message is removed.

- **Acknowledge after Successful Processing**

There are two pathways under this action:

- In case of successful execution of the iFlow, an acknowledgement is sent to Google and the message is removed.
- However, if the iFlow execution fails, NACK is sent to Google, and message is made available for redelivery. This causes consumption of the same message in Cloud integration and results in higher MPLS.

To avoid a scenario where iFlow execution fails due to corrupted data, you can take the following precautions:

-  Enable **Dead lettering** which forwards undelivered messages to a dead-letter topic after a set number of retries.

- Set **Retry policy** to **Retry after exponential backoff** to add progressively longer delays between retry attempts.

- **Do not Acknowledge**



For gRPC, **Do not Acknowledge** is only meant for testing and is not recommended for a production scenario.

If you have selected Do not Acknowledge, NACK is sent to Google, and message is made available for redelivery.