Docusign Adapter for SAP Integration Suite

Version 1.0.1 – February 2026

THE BEST RUN SAP

1.Introduction

1.1 Objective

This is the official guide for the Docusign Adapter for SAP Integration Suite. This guide covers all relevant information for integration developers to start working with the Docusign 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

The Docusign receiver adapter connects SAP Integration Suite to Docusign and accelerates communication between the two systems.

The Docusign adapter enables you to manage and configure Docusign objects to integrate features like embedded sending, e-signatures, etc into your applications.

1.5 Features

- Allows you to perform several operations such as Create, Read, Update, and Delete (**CRUD**) for Entity such as Envelopes, SigningGroups, and Template etc.
- Allows secure authentication via **OAuth 2.0 Authorization Code** and **JSON Web Token** based authentication.
- Supports **Basic** configuration for convenient processing capability whereas **Advanced** enables proficient users to perform calls with greater control while connecting to any API endpoint.
- Allows flexibility while retrieving metadata using **Query Parameter** field.
- Supports response output formats like JSON, XML.

2. Installation and Configuration

This section describes the prerequisites and procedure to install the Docusign adapter.

2.1 Prerequisites

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

2.2.1 Adapter Installation by creating a New Integration Flow

The Docusign adapter is available for selection in the 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 **Name** and **ID** for your flow. Additionally, select **Runtime Profile** from the drop-down and choose **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.

5. 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 **Docusign** adapter should show up in the list.
6. Select the **Docusign** adapter from the list. The adapter is now imported which *triggers* an adapter deployment. Once Docusign Adapter is deployed, a success message is displayed.
   After the above steps are done, the Docusign Adapter is successfully deployed in your Design workspace of the SAP Integration Suite tenant.

2.2.2 Adapter Installation without Creating a New Integration Flow

> The following procedure describes how the Docusign 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 Docusign adapter can be imported into the Design workspace without creating an integration flow. Use the Transport Management Service (TMS) to import/transport the Docusign 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 Docusign adapter to the Design workspace, follow these steps:

Procedure

1. Go to **Discover** workspace.
2. In the search box, search for the **Docusign 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 **Docusign 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.

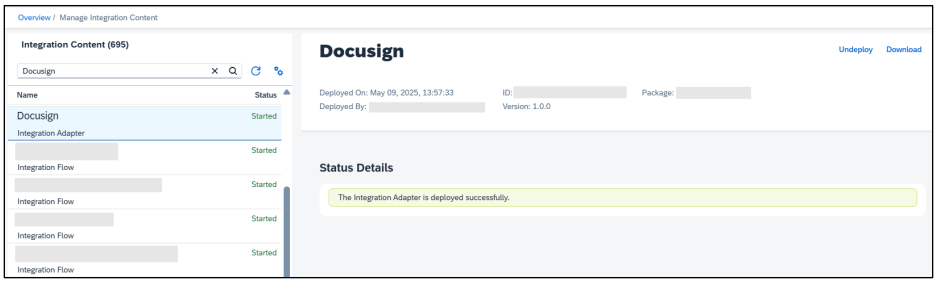
After the adapter deployment is complete, you can check the status in the **Monitor** section.

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 **Integration Content** page with a list of all the deployed adapters.
3. Here, you can check and confirm the deployment status of your adapter.



3. Getting Started: Docusign Adapter

This section explains how to configure the Docusign adapter for SAP Integration Suite. You can find information about adapter architecture, application configuration and authentication.

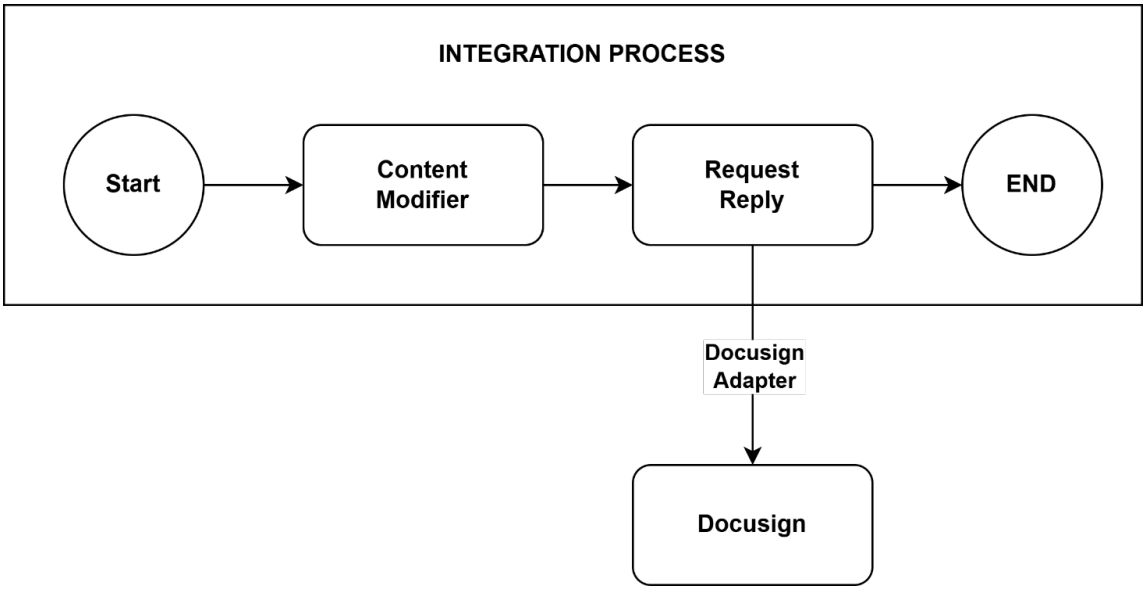
3.1 Architecture Overview

The Docusign adapter is designed to function as a receiver adapter.

In such a scenario where Docusign Adapter is used as a receiver adapter, SAP Cloud Integration acts as the initiator of the calls.

A sample flow contains *Request-Reply* step and *End* step in the integration process, responsible for connecting and interacting with Docusign to invoke its operations. The Docusign adapter supports CRUD operations. For more information, see Operations supported in Docusign.

The figure below demonstrates provides a high-level representation of how the adapter works.



3.2 Application Configuration

- To get an overview of Docusign, see Docusign Overview.
- For more information about accounts, see Create Account.
- For more information regarding JWT, see JWT Grant.
- For more information regarding OAuth 2.0, see Authorization Code Grant
- For more information about connecting to Docusign, see Authentication.

3.3 Authentication

This section details the authentication mechanisms supported by the Docusign Adapter in SAP Integration Suite.

The Docusign adapter supports the OAuth mechanism and JSON Web Token (JWT):

- Docusign JWT is an authentication mechanism that increases overall system security.
- OAuth enables client applications to use an access token to access Docusign through APIs.

Before setting up the authentication, you must create the Credentials in **Security Material** in the SAP Cloud Integration.

3.3.1 Add RSA Key in Cloud Integration Keystore

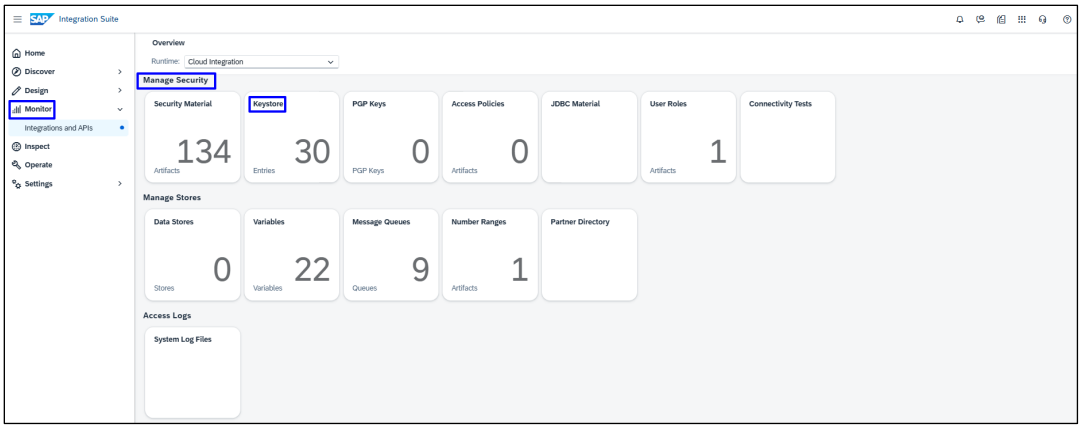
Purpose

To create credentials for authentication.

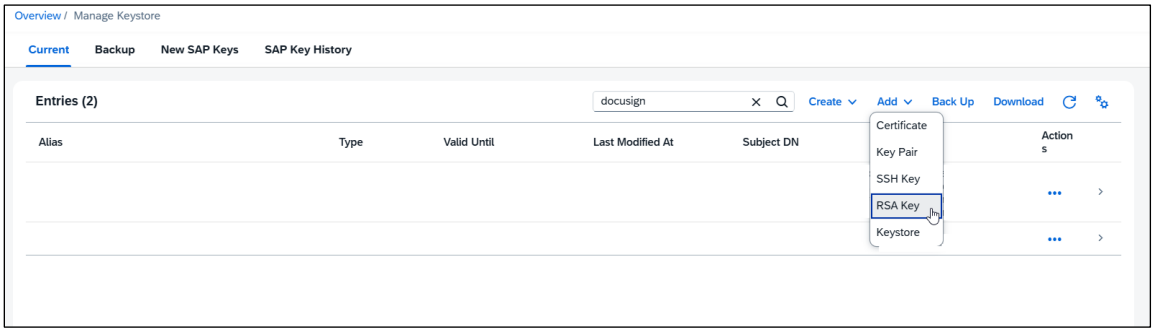
Procedure

1. In SAP Integration Suite, navigate to **Monitor** > **Integrations and APIs**. This opens the **Overview** page.

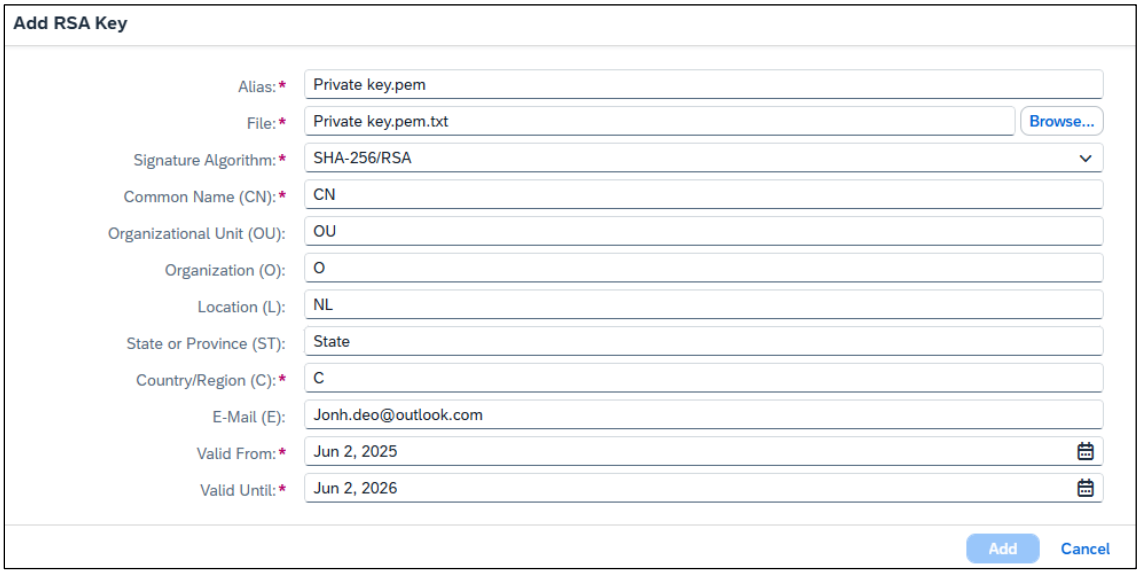
2. On the **Overview** page, go to **Manage Security** section and click **Keystore**.



3. On **Keystore** page, click **Add** and select **RSA Key** from the dropdown.



4. In the **RSA Key** popup, provide the below details.



Parameter	Description
Alias	Specify the name for the security artifact. The artifact name is used as an alias for the confidential data assigned by this parameter.
File	Enter a file name for the artifact.
Signature Algorithm	Select the required signature algorithm.
Common Name (CN)	Specify the primary identifier for the entity associated with the artifact.
Organizational Unit (OU)	Specify the organizational unit.
Organization (O)	Specify the organization.
Location (L)	Specify the location.
State or Province (ST)	Specify the state or province.
Country/Region (C)	Specify the country.
Email	Specify the email.
Valid From	Specify the start date for the artifact's validity.
Valid Until	Specify the end date for the artifact's validity.

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

When you refresh the **Manage Security Material** page, the new artifact is displayed in the artifact table. For more information, refer to the Generate a keypair with DocuSign using Keystore Generation.

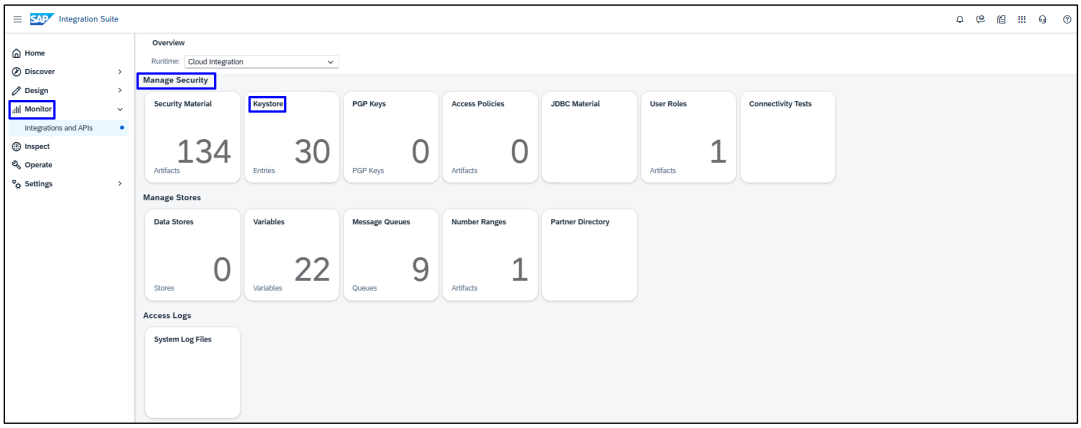
3.3.2 Add RSA Key-Pair in Cloud Integration Keystore

Purpose

To create credentials in Security Material for authentication.

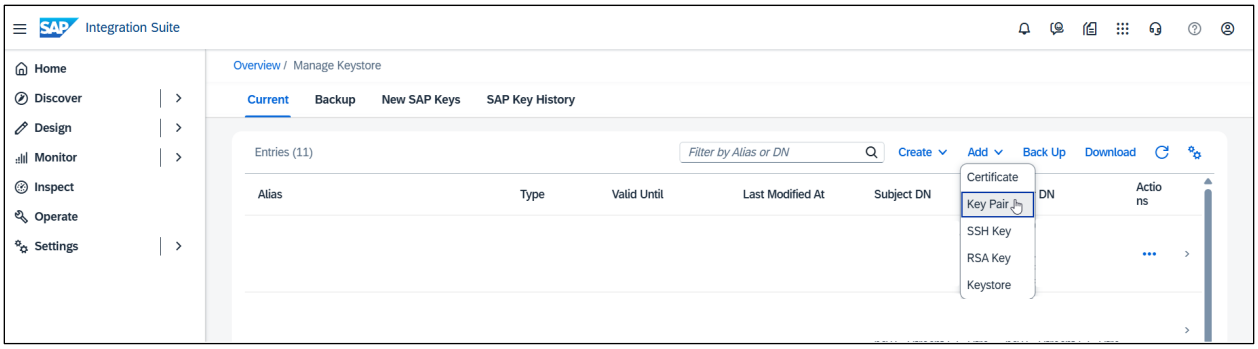
Procedure

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**.



4. On **Keystore** page, click **Add** and select **Key-Pair** from the dropdown.

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



Parameter	Description
Alias	Specify the name for the security artifact. The artifact name is used as an alias for the confidential data assigned by this parameter.
File	Select the certificate file to upload.
Password	Specify the password for the artifact.

6. Click **Add** to complete the process.

When you refresh the **Manage Security Material** page, the new artifact is displayed in the artifact table. For more information, refer to the Generate an RSA Key and Certificate to Upload to Docusign.

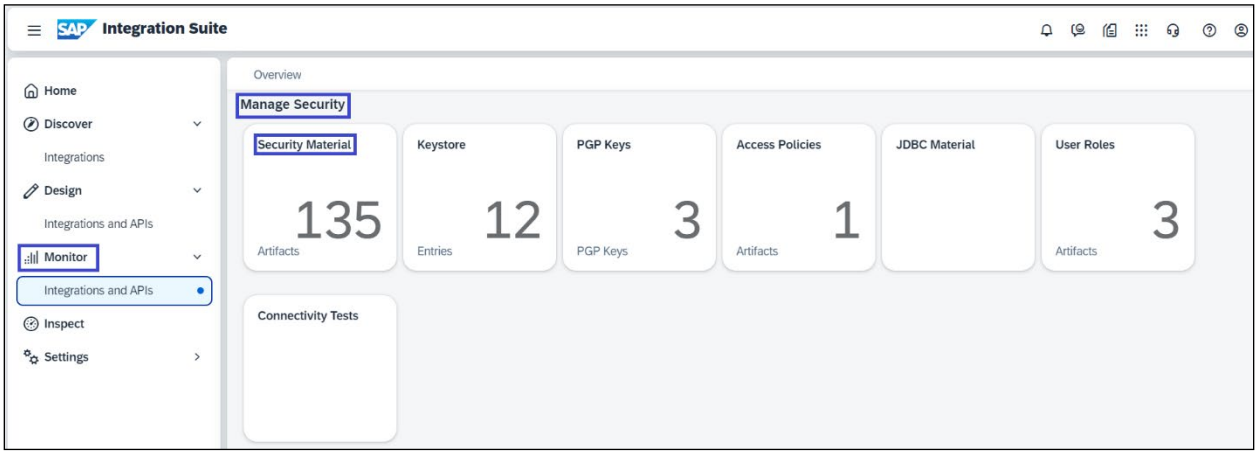
3.3.3 OAuth2 Authentication

Purpose

To create credentials in Security Material for OAuth2 authentication.

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 Authorization Code** from the dropdown.

4. In the **Create OAuth2 Authorization Code** popup, provide the details below.

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).
Provider	Enter the name of the provider of the platform on which you created the OAuth2 client. Example: Generic
Authorization URL	Provide the Authorization URL for authorizing the OAuth client to access resources of a user.
Token Service URL	Address of the token service that issues the access token.
Redirect URL	Displays the URL you need when creating the OAuth Clients/App in OAuth Authorization Server/Token Server.
Client ID	Specify the ID of the client you want to connect to.
Client Secret	Specify the Secret key of the client to which you are connecting.
Send As	Basic Authentication Header: Send the Client ID and Client Secret in the request body when calling the Authorization URL or Token Service URL.
User Name	Name of the user whose resources the OAuth2 client gets access to.
Scope	OAuth2 scopes protect access to the resources. If you add more than 1 scope, you need to separate your scopes by a comma.

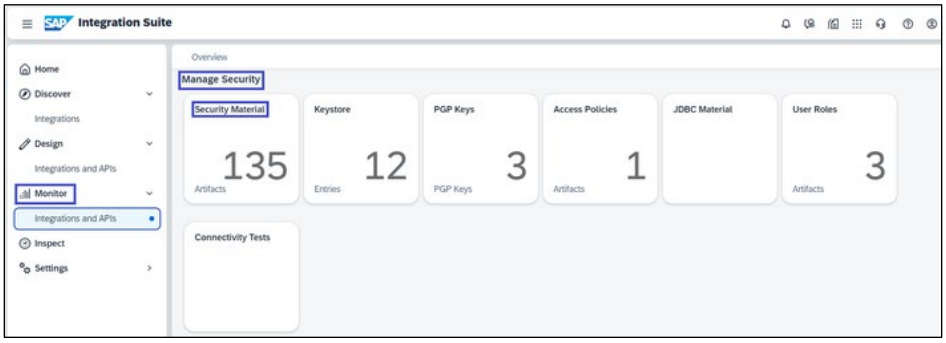
5. Click **Deploy** to complete the process.

3.3.5 Creating Secure Parameter in Security Material

The creation of credentials to support the authentication mechanism can be done by following the steps below:

3. In SAP Integration Suite, navigate to **Monitor** > **Integrations and APIs**. This opens the **Overview** page.

1. On the **Overview** page, go to **Manage Security** section and click **Security Material**.



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

3. In the Create Secure Parameter popup, provide the details below.

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).

Parameter	Description
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 **Secure Parameter**) in the artifact table.

4. Docusign Adapter Configuration

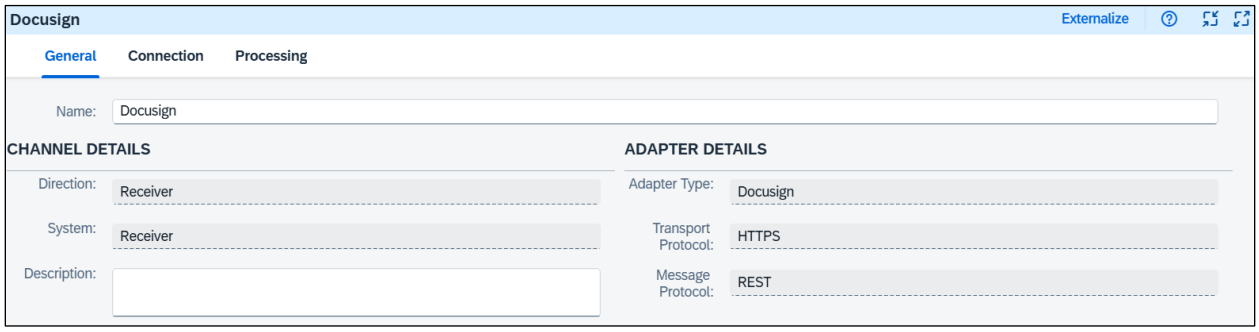
This section describes the parameters to be configured for your Docusign adapter. You need to configure the **General**, **Connection,** and **Processing** tabs. A description and example usage for every field has been added.

4.1 Receiver Adapter

The Configuration of the Docusign Receiver Adapter for each one of the supported variants is mentioned below.

4.1.1 General tab

The General tab provides an overview of basic adapter information, including **Channel** and **Adapter** details.



Only the Name and Description fields are editable.

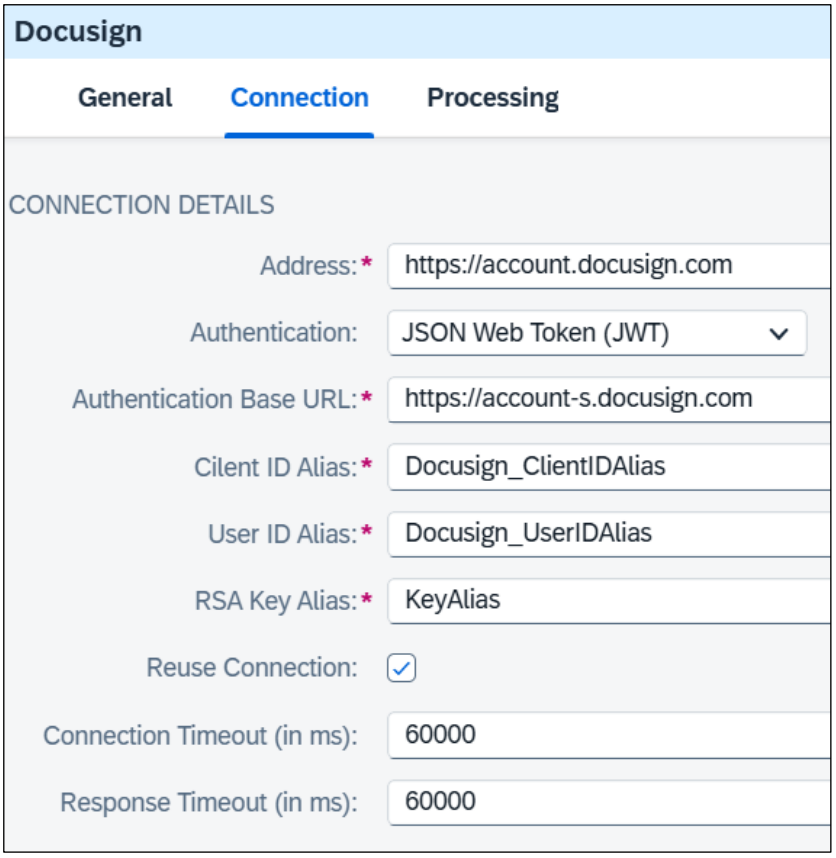
Parameter	Description
Name	Name of the adapter flow
Description	Description of the adapter

4.1.2 Connection Tab

The Connection tab contains connection and authentication parameters for Docusign.

Using Credentials

The Security artifact created in the Creating Credentials in Security Material should be used in the **Connection tab** of the Adapter, as shown below.



The connection tab contains the following fields:

Parameter	Description
Address	Specify the address of Docusign to be used for the connection. Example: `https://account.docusign.com`

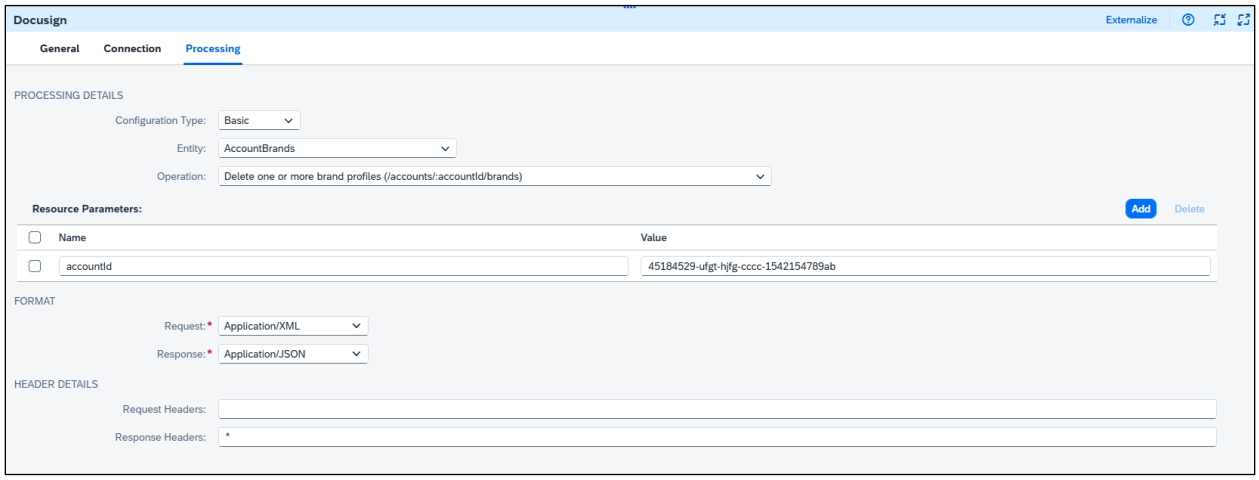
Parameter	Description
Authentication	Select your Authentication Mechanism to connect to Docusign: • **JSON Web Token (JWT)** • **OAuth 2.0 Authorization Code**
Authentication Base URL	Specify the base URL for the authentication service of Docusign.
Client ID Alias	Specify the Secure Parameter artifact that contains the Client ID needed to connect to Docusign. For more information, see Creating Secure Parameters in Security Material.
User ID Alias	Specify the Secure Parameter artifact that contains the User ID needed to connect to Docusign. For more information, see Creating Secure Parameters in Security Material.
RSA Key Alias (Only available when **JSON Web Token (JWT)** is selected)	Specify the RSA key alias for connecting to the Docusign account. For more information, see Add RSA Key in Cloud Integration Keystore and Add RSA Key-Pair in Cloud Integration Keystore.
Authorization Code Credentials (Only available when **OAuth 2.0 Authorization Code** is selected)	Specify the Secure Parameter key alias that stores the Docusign OAuth Authorization Code. For more information, see OAuth2 Authentication.
Reuse Connection	Enable this property to reuse the HTTP connection.
Connection Timeout (in ms)	Specify the connection timeout in milliseconds. You can configure the maximum waiting time for SAP until a response is received from Docusign. Example: `60000`

Parameter	Description
Response Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for a response message. Example: `60000`

The adapter validates the certificate while connecting to Docusign. In case the certificates of Docusign are not present in the SAP Cloud Integration Keystore, then an error is shown.

4.1.3 Processing Tab

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



Parameter	Description
Configuration Type	Select the required configuration type: • **Basic** to use the dropdowns and parameter text fields. • **Advanced** to specify the relative URL.
Entity	Specify the entity on which the operation to be performed.
Operation	Select the operation to be performed.

Parameter	Description
File Name	Specify the file name of the resource being uploaded.
Query Parameter	Specify the expression containing the query parameter and value. Example: `"include_external_references,include_logos"`
Count	Specify the maximum number of results to return. Valid values range from 1 to 100.
Start Position	Specify the zero-based index of the result from which to start returning results.
Resource Parameters	Specify the **Name** and **Value** in case the resource path includes parameters. Example: Set **Name** as `propertyKey` and **Value** as `76548`
HTTP Method	Select the required HTTP method from the available dropdown: • **DELETE** • **GET** • **PATCH** • **POST** • **PUT**
Relative URL (Only available when the **Configuration Type** is **Advanced**)	Specify the relative endpoint, excluding the Host. Example: `/restapi/v2.1/accounts/1234`

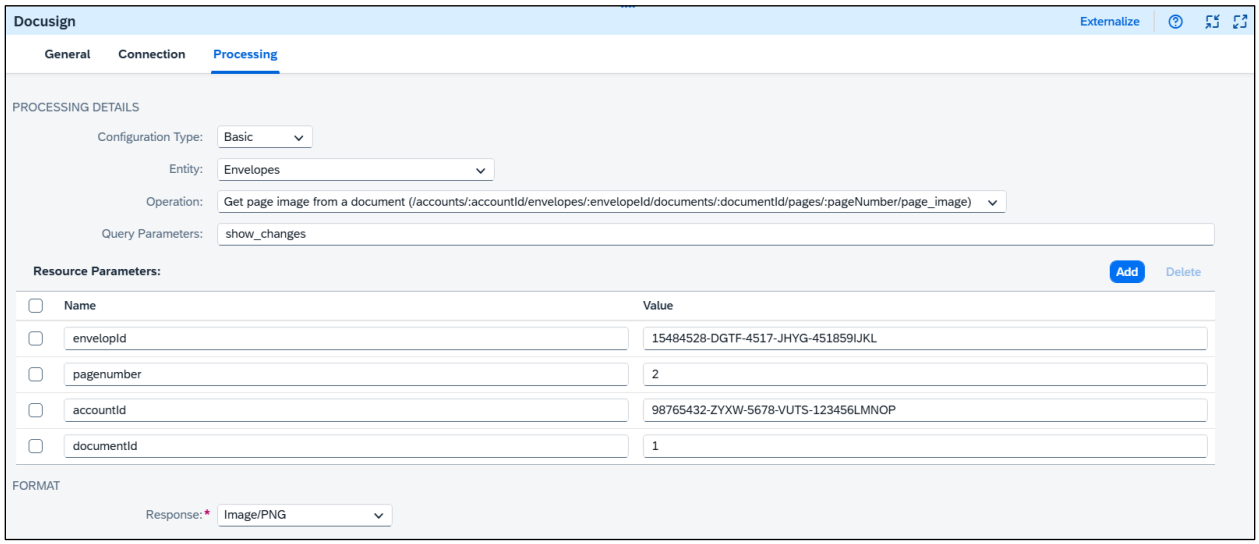
Parameter	Description
Request	Select the format for request payload: (based on the selected **Operation,** only applicable options will be available) • **Application/JSON** • **Application/XML** • **Application/PDF** • **Image/PNG** • **Image/GIF** You must dynamically assign a value through a **header** or **property** when the desired option isn't available in the dropdown. Example: ${property.request} where request is set to image/jpeg
Response	Select the format for response payload: (based on the selected **Operation,** only applicable options will be available) • **Application/JSON** • **Application/XML** • **Application/Octet-stream** • **Application/PDF** • **Image/PNG** • **Image/GIF** • **Text/PLAIN** You must dynamically assign a value through a **header** or **property** when the desired option isn't available in the dropdown. Example: ${property.request} where request is set to image/jpeg

Parameter	Description
Request Header	Enter a list of custom headers, separated by a pipe (|), to send to the target system. By default, no custom headers are sent. Use an asterisk (*) to send all custom headers to the target system. Alternatively, you can dynamically pass on the values by defining a property that includes a list of headers.
Response Header	Enter a list of headers coming from the target system's response, separated by a pipe (|), to be received in the message. Use an asterisk (*) to receive all the headers from the target system, which is also the default value.

5. Docusign Adapter Operations

5.1 Envelopes: Get page image from a document (/accounts/:accountId/envelopes/:envelopeId/documents/:documentId/pages/:pageNumber/page_image)

Retrieves a specific page image from a document inside a Docusign envelope, enabling display or preview functionality.

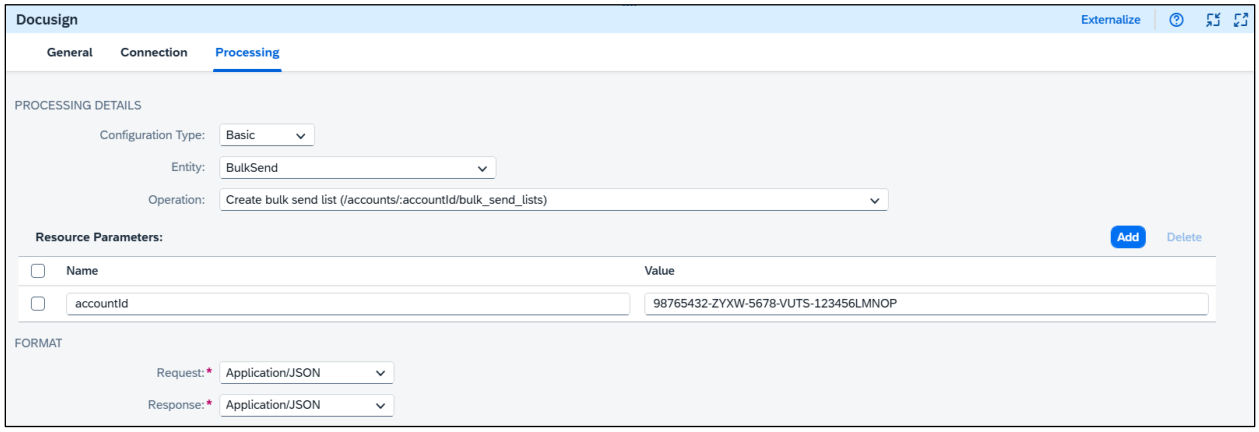


Parameter	Values	
Configuration Type	Select the configuration type as **Basic**.	
Entity	Select the entity as **Envelopes**.	
Operation	Select operation as **Gets a page image from a document**	
Query Parameter	Set as `show_changes`	
Resource Parameter	**Name**	**Values**
	envelopId	`15484528-DGTF-4517-JHYG-451859IJKL`

	pageNumber	2
	accountId	98765432-ZYXW-5678-VUTS-123456LMNOP
	documentId	1
Response	Select as **Image/PNG**	

5.2 BulkSend: Create bulk send list (/accounts/:accountId/bulk_send_lists)

Allows you to create a new account. You can define the summary and description of the issue inside fields in the payload/body.

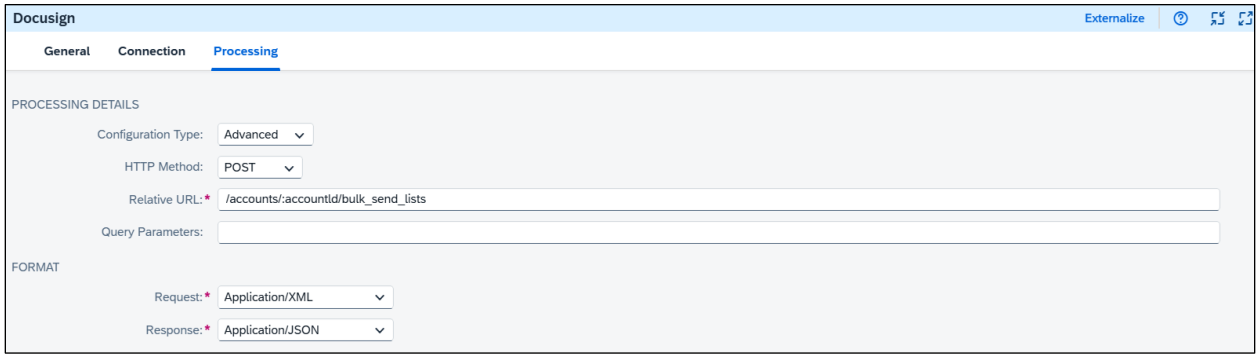


Parameter	Values	
Configuration Type	Select the configuration type as **Basic**.	
Entity	Select the entity as **BulkSend**.	
Operation	Select operation as **Create bulk send list**	
Resource Parameter	**Name**	**Values**
	accountId	98765432-ZYXW-5678-VUTS-123456LMNOP
Request	Select as **Application/JSON**	
Response	Select as **Application/JSON**	

Sample Payload:


```
{
  "name": "Sample Bulk List",
  "bulkCopies": [
    {
      "recipients": [
        {
          "name": "John Doe",
          "email": "john.doe@example.com",
          "roleName": "Signer"
        }
      ],
      "customFields": [
        {
          "name": "CustomField1",
          "value": "Value1"
        }
      ]
    },
    {
      "recipients": [
        {
          "name": "Jane Smith",
          "email": "jane.smith@example.com",
          "roleName": "Signer"
        }
      ],
      "customFields": [
        {
          "name": "CustomField1",
          "value": "Value2"
        }
      ]
    }
  ]
}
```

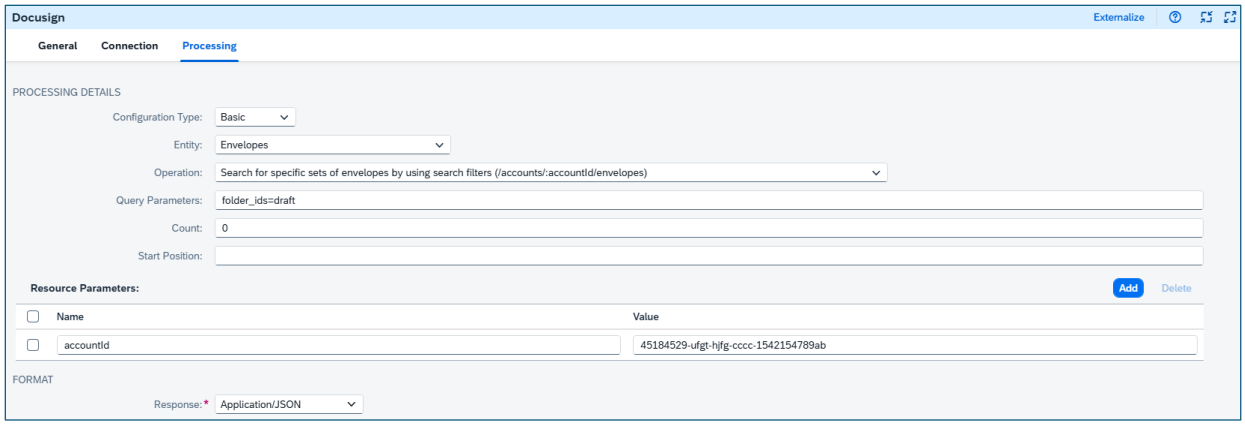
You can achieve the same using **Advanced** configuration type as well.



Parameter	Values
Configuration Type	Select the configuration type as **Advanced**.
HTTP Method	Select the method as **POST**.
Relative URL	Set as `/accounts/:accountId/bulk_send_lists`
Request	Select as **Application/XML**
Response	Select as **Application/JSON**

5.3 Envelopes: Search for specific sets of envelopes by using search filters(/accounts/:accountId/envelopes)

Envelope statuses of multiple envelopes within an account, enabling efficient tracking and management of document workflows.



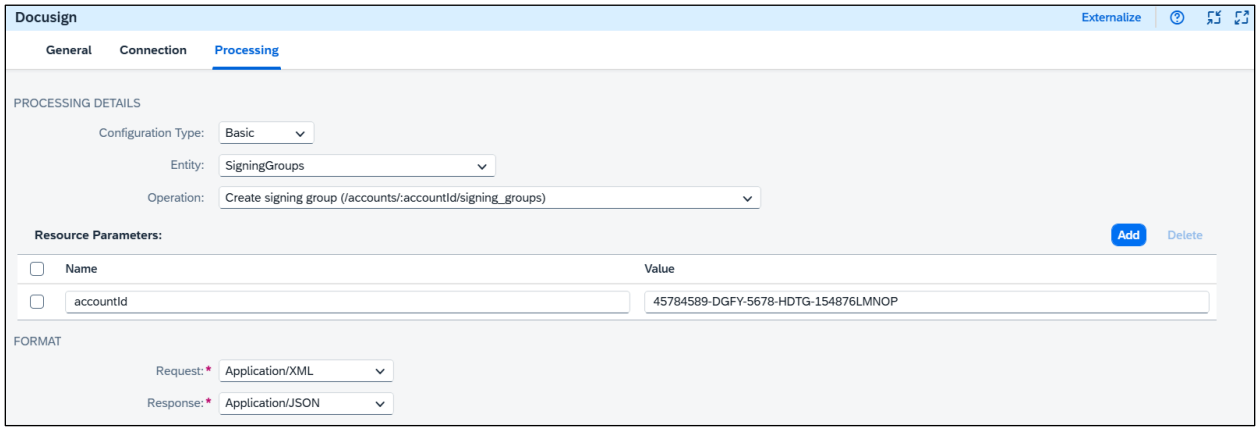
Parameter	Values
Configuration Type	Select the configuration type as **Basic**.
Entity	Select the entity as **Envelopes**.
Operation	Select operation as **Search for specific sets of envelopes by using search filters**
Query Parameter	`folder_ids=draft`
Count	Set as 0
Resource Parameter	**Name**: **accountId** — **Value**: `98765432-ZYXW-5678-VUTS-123456LMNOP`
Response	Select as **Application/JSON**

Sample Payload:

```
{
  "from_date": "2025-05-01T00:00:00Z",
  "to_date": "2025-05-27T23:59:59Z",
  "status": "completed",
  "include": [
    "recipients",
    "documents",
    "sender"
  ],
  "page_size": 50,
  "start_position": 1
}
```

5.4 SigningGroups: Create signing group (/accounts/:accountId/signing_groups)

Contacts allow users to store and manage contact details within their accounts, facilitating seamless document transactions.

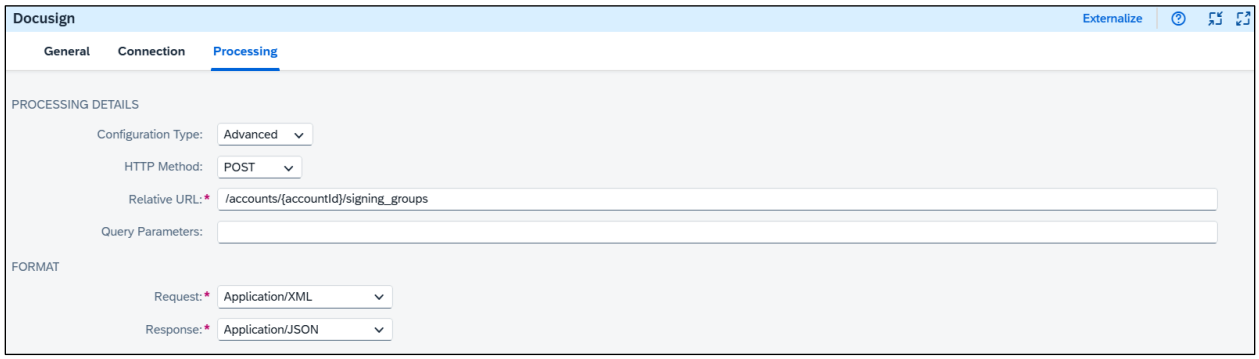


Parameter	Values
Configuration Type	Select the configuration type as **Basic**.
Entity	Select the entity as **SigingGroups**.
Operation	Select operation as **Create signing group**
Resource Parameters	**Name**: **accountId** — **Value**: `45784589-DGFY-5678-HDTG-154876LMNOP`
Request	Select as **Application/XML**
Response	Select as **Application/JSON**

Sample Payload:


```
{
  "contacts": [
    {
      "signingGroupUsers": [
        {
          "userName": "John Deo",
          "userId": "67890",
          "email": "john.deo@outlook.com",
          "userType": "member",
          "userStatus": "inactive",
          "uri": "/users/67890",
          "loginStatus": "logged_out",
          "sendActivationEmail": "false",
          "activationAccessCode": "XYZ7890"
        }
      ],
      "contactId": "54321",
      "name": "Bob Williams",
      "emails": [
        "bob.williams@outlook.com"
      ],
      "organization": "Tech Innovations",
      "shared": "true",
      "contactUri": "/contacts/54321",
      "signingGroup": "Finance Team",
      "signingGroupName": "Authorized Signers",
      "contactPhoneNumbers": [
        {
          "phoneNumber": "+1-555-987-6543",
          "phoneType": "mobile"
        }
      ]
    }
  ]
}
```

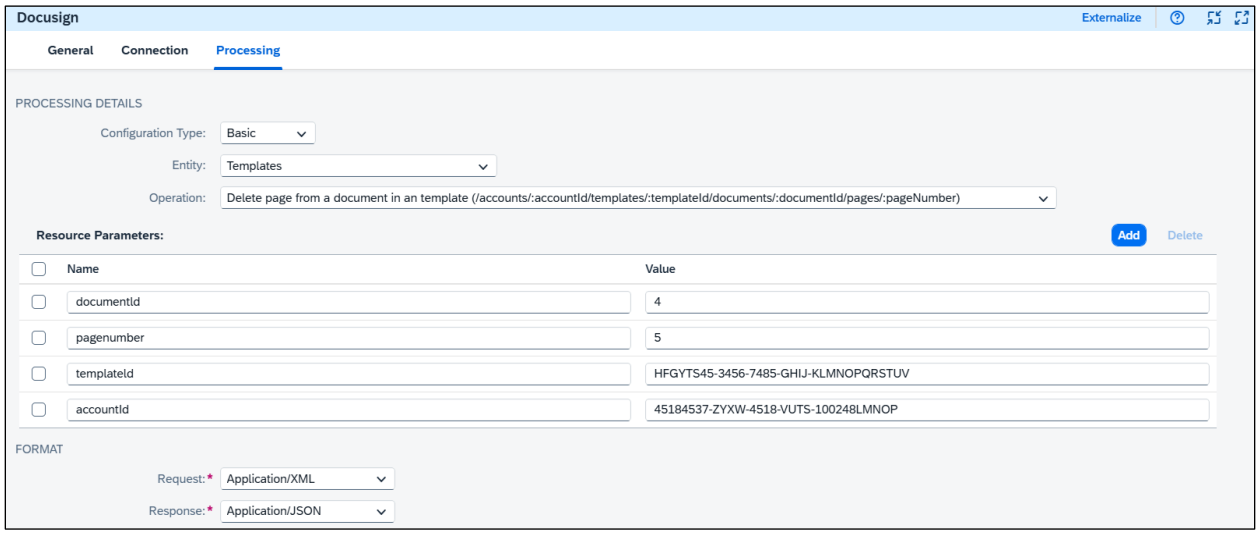

You can achieve the same using **Advanced** configuration type as well.



Parameter	Values
Configuration Type	Select the configuration type as **Advanced**.
HTTP Method	Select the method as **POST**.
Relative URL	Set as `/accounts/{accountId}/signing_groups`
Request	Select as **Application/XML**
Response	Select as **Application/JSON**

5.5 Templates: Delete page from a document in an template (/accounts/accountId/templates/templateId/documents/documentId/pages/pageNumber)

Allows you to remove a specific page from a document within a template. Specify the account ID, template ID, document ID, and page number in the endpoint URL.



Parameter	Values
Configuration Type	Select the configuration type as **Basic**.

<table>
<tr><td>Entity</td><td colspan="2">Select the entity as Templates.</td></tr>
<tr><td>Operation</td><td colspan="2">Select operation as Delete page from a document in a template</td></tr>
<tr><td>Resource Parameter</td><td>Name</td><td>Value</td></tr>
<tr><td></td><td>pagenumber</td><td>4</td></tr>
<tr><td></td><td>documentId</td><td>5</td></tr>
<tr><td></td><td>templateId</td><td>HFGYTS45-3456-7485-GHIJ-KLMNOPQRSTUV</td></tr>
<tr><td></td><td>accountId</td><td>45184537-ZYXW-4518-VUTS-100248LMNOP</td></tr>
<tr><td>Request</td><td colspan="2">Select as Application/XML</td></tr>
<tr><td>Response</td><td colspan="2">Select as Application/JSON</td></tr>
</table>

Sample Payload

```
{
  "rotate": "90",
  "password": "SecurePass123!"
}
```

6. References

6.1 Initialize JSON Web Token (JWT) in Docusign

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

1. The Audience is the login URL used by the authorization server of Docusign. In the documentation of Docusign, this URL is specified as https://account.docusign.com. This link directs the Docusign Adapter to the login page of Docusign.
1. The Client ID Alias specify the name of the Secure Parameter artifact that contains the client ID needed to connect to Docusign. It specifies the Integration key when new app was created.
2. The User ID Alias specify the name of the Secure Parameter artifact that contains the user ID needed to connect to Docusign. It specifies the OAuth Consumer Key of the connected app for which the certificate was registered.
3. The RSA Keystore Alias refers to the added .pkcs12 (Key-Pair) or .pem (RSA Key) file in Keystore as a Key Pair. It consists of a key and certificate to sign the JWT.

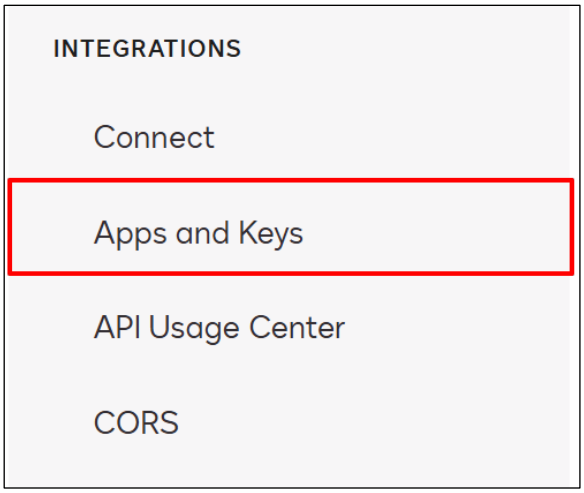
In case you encounter a consent related error, ensure that you complete the consent step.

For more information, see the official Docusign documentation.

6.1.1 Create a Connected App in Docusign Using Authorization Code Grant

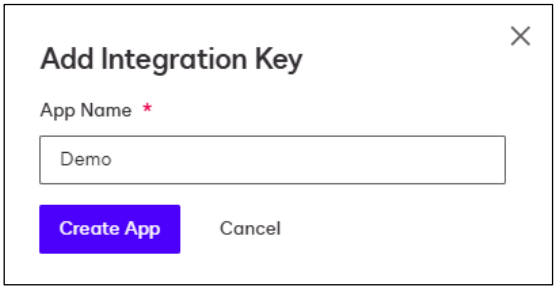
To create a Connected App, follow the steps below:

1. Login to the Docusign console and select Admin.
2. On the left panel, under Integrations. Click on Apps and keys.

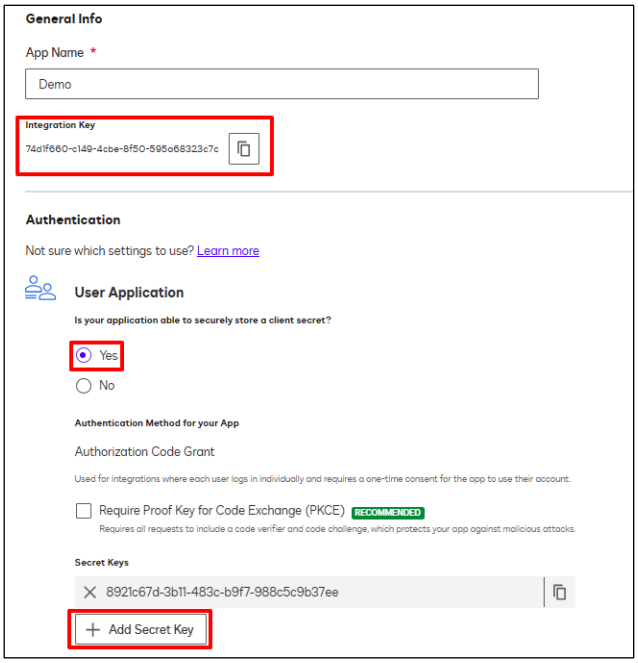


Apps and Keys

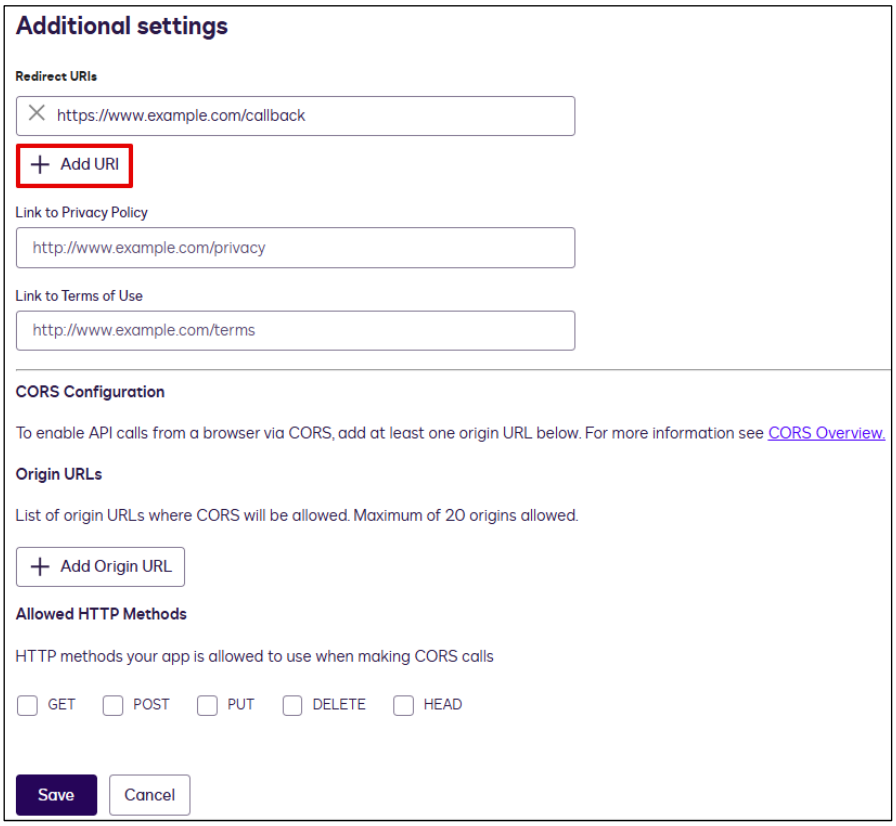
3. In the next screen, select Add App and Integration Key, enter App Name and click Create App.



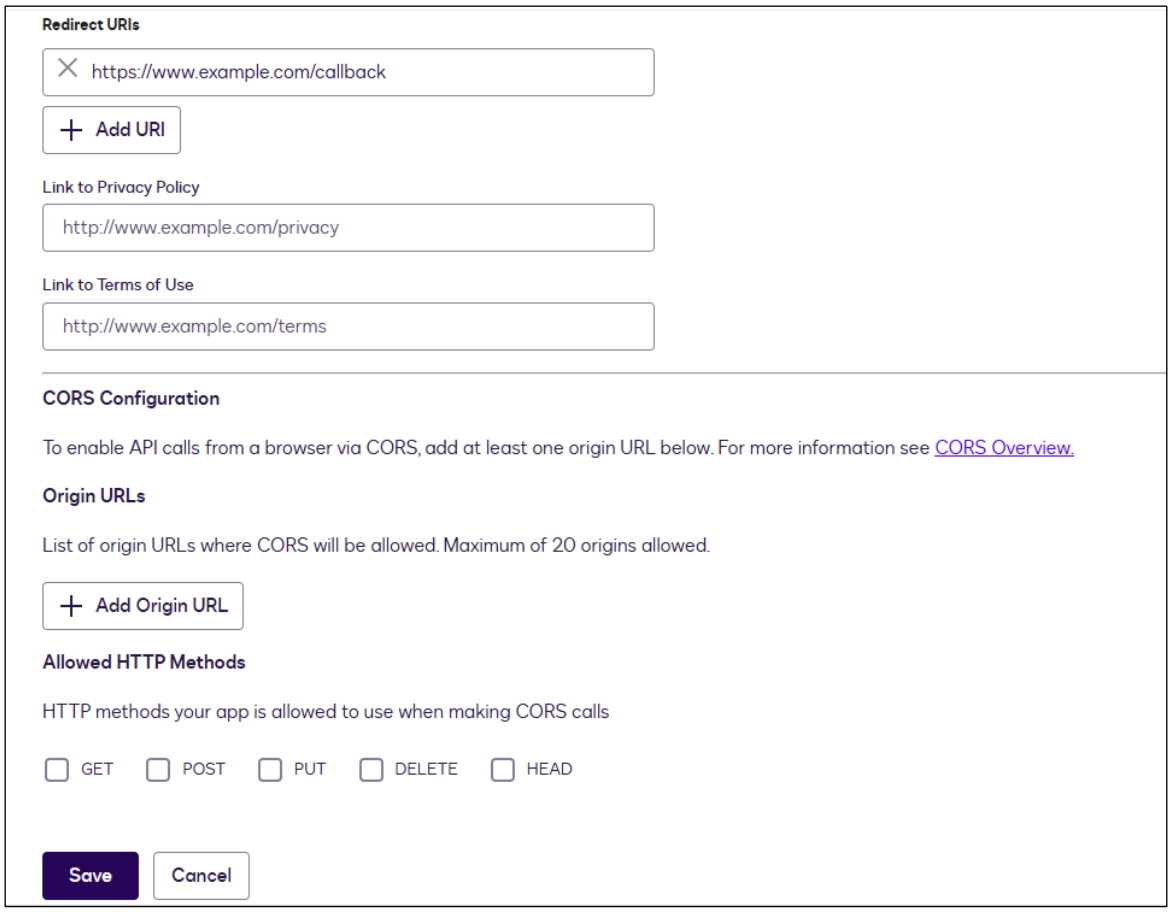
4. In next tab, Copy Integration Key, select yes for the client secret key and select Add Secret Key.



5. Add all the additional Field according to the requirement.



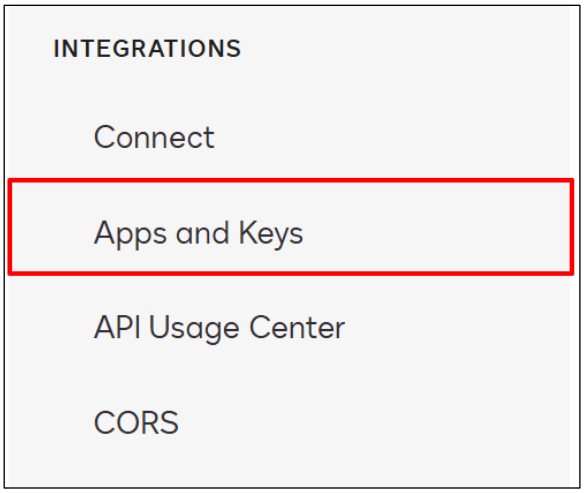
6. Click **Save**.



6.1.2 Create a Connected App in Docusign Using JWT

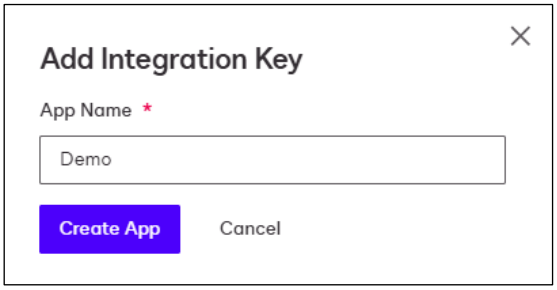
To create a Connected App, follow the steps below:

1. Login to the Docusign console and select Admin.
2. On the left panel, under Integrations. Click on Apps and keys.

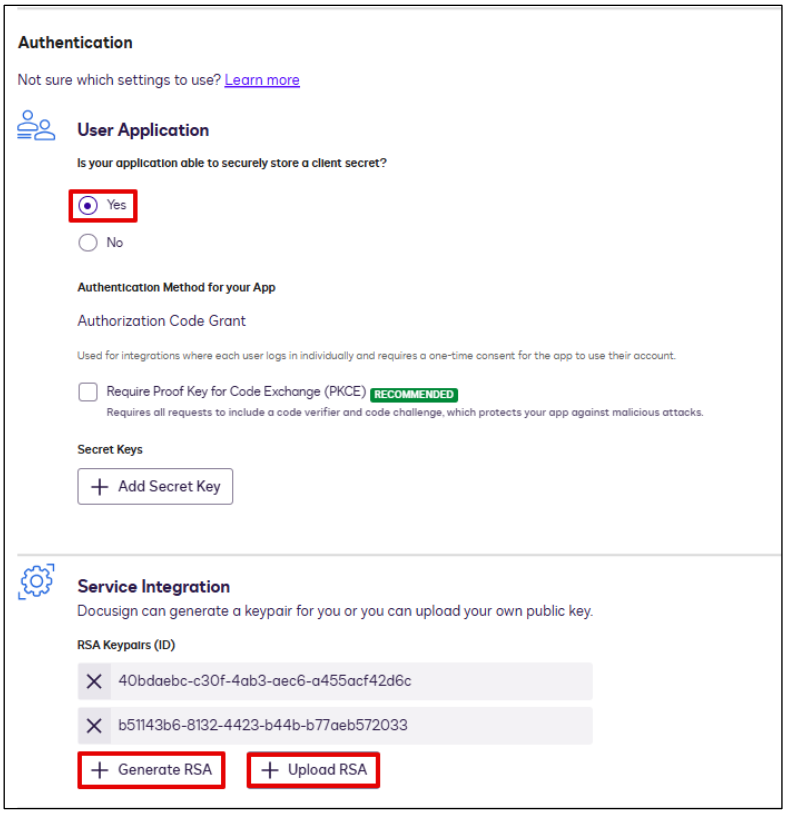


Apps and Keys

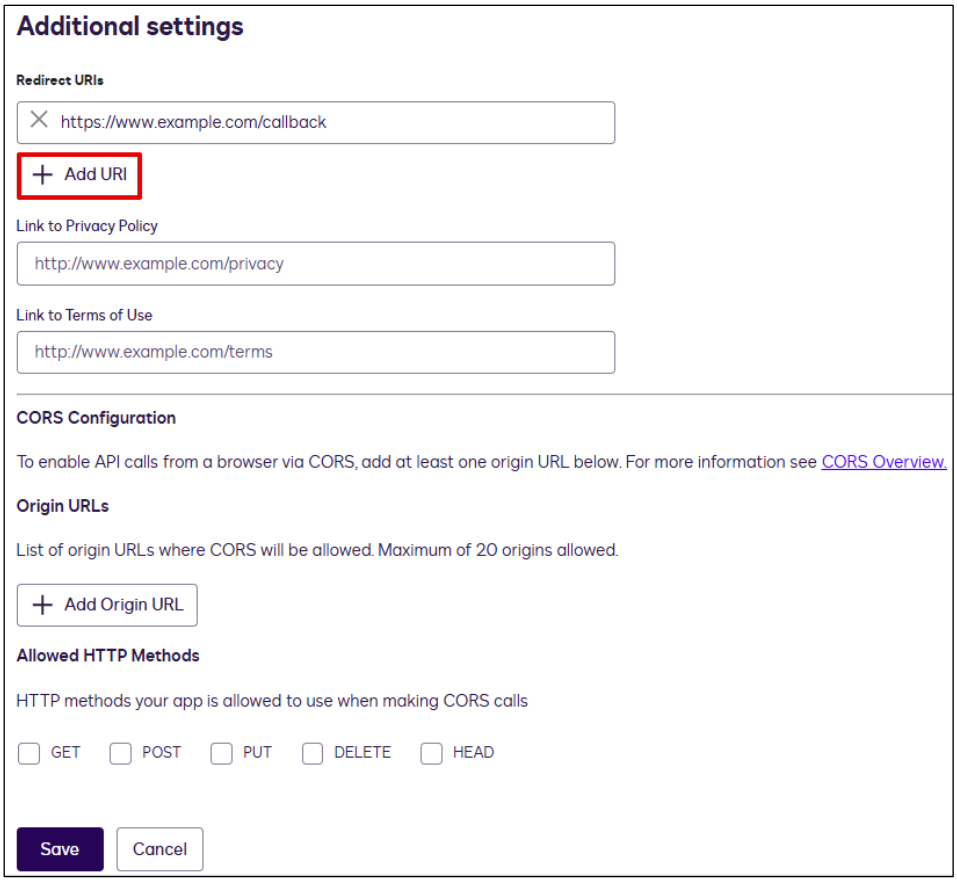
3. In the next screen, select Add App and Integration Key, enter App Name and click Create App.



4. In next tab, Copy Integration Key (Client ID), and click Generate RSA (RSA Key) or Upload RSA (key-pair).



5. Add URL and all other additional Field according to the requirement.



6. Click **Save**.

6.1.3 Generate a Keypair with Docusign and Upload to SAP Keystore

To create a Keypair with help of Docusign application, follow below steps:

1. Follow reference documentation Generate RSA Key.
2. Copy the Private Key.
3. Paste the key in notepad and save as .pem format.

6.1.4 Generate an RSA Key and Certificate to Upload to Docusign

To create RSA Private Key and certificate using OpenSSL, follow the steps below:

1. Since a .pem file contains both the private key and the public key, you first need to generate them.

   OpenSSL Command:

   openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048

2. Extract the public key from the private key.

   OpenSSL Command:

   openssl rsa -in private_key.pem -pubout -out public_key.pem

3. Now generate a self-signed certificate.

   OpenSSL Command:

   openssl req -new -x509 -key private_key.pem -out certificate.pem -days 365

4. You will be prompted to enter details shown below.

Country Name (2 letter code) : **<Enter your Country Name>** State or Province Name (full name) [Some-State]: **<Enter your State Name>** Locality Name : **<Enter your Locality Name>** Organization Name : **<Enter your Organization Name>** Organizational Unit Name : **<Enter your Organizational Unit Name>** Common Name : **<Enter your Integration Key>** Email Address : **<Enter your Email Address >**

5. Now export the key in .pkcs12 and enter the password for accessing the certificate.

OpenSSL Command:

```
openssl pkcs12 -export -inkey private_key.pem -in
certificate.pem -out certificate.pfx -name "MyCertificate"
```

Enter Export Password:**12345**
Verifying – Enter Export Password:**12345**