Adobe Sign Adapter for SAP Integration Suite

Version 1.0.0 – March 2026

THE BEST RUN SAP

Contents

1. Introduction

1.1 Objective

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

Adobe Sign is a cloud-based electronic signature service that enables organizations to send, sign, track, and manage documents digitally. The **Adobe Sign Adapter for SAP Integration Suite** enables seamless connectivity to the Adobe Sign platform. You can utilize electronic signatures and embedded sending to automate and manage document processes for your applications. Revolutionize your business with an integrated digital signing experience.

1.5 Features

Adobe Sign Adapter has the following features:

- Support for Create, Retrieve, Update, and Delete (CRUD) operations for entities such as **Agreement**, **Library Documents**, and **Message Templates**, etc.
- Secure authentication via **Integration Key**.
- Flexibility metadata retrieval using the **Query** Parameter field.
- Supports **Basic** configuration for convenient processing capability, whereas **Advanced** allows proficient users to connect to any API endpoint.
- **Transient Documents** operation to upload a document for signing.
- **Mega Sign** operation to manage and track signatures from multiple signers or approvers.

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) Adobe Sign account (see Log in to Adobe Sign) and Integration Key (see 3.2.3 Create Integration Key).
    b) SAP Integration Suite tenant with Standard License.
2. Create Secure Parameter in SAP Integration Suite:
    a) Navigate to Monitor > Integrations and APIs.
    b) Go to Manage Security > Security Material.
    c) Create a Secure Parameter using the Integration Key created in step 1a. (see 3.3 Authentication)
3. Add Adobe Sign Adapter to an Integration Flow:
    a) Go to the Design workspace and create or open an Integration Flow.
    b) Add Adobe Sign Adapter as a Receiver.
4. Configure Connection Tab:
    a) Address: `https://api.na1.adobesign.com`
    b) Key Name Alias: Enter Secure Parameter artifact created in step 2c.
    c) Keep default timeout values.
5. Configure Processing :
    a) Perform Upload Transient Document operation using Basic Configuration Type.
6. Deploy & Run:
    a) Save and Deploy the Integration Flow.
    b) Monitor execution under Integrations and APIs.

2. Installation and Configuration

This section describes the prerequisites and procedure to install the Adobe Sign adapter

2.1 Prerequisites

> The Adobe Sign 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 the Apache Camel 3x environment and Edge Integration Cell (EIC) platform.

2.2.1 Adapter Installation by Creating a New Integration Flow

The Adobe Sign adapter is available for selection in the Receiver adapter list and can be deployed in the **Design** tab directly as you use it in Integration flow.

Purpose

To install an adapter for use in your Integration flow.

Procedure

Go to the **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 **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.

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.
6. Select the **Adobe Sign Adapter** from the list.

   The adapter is now imported, which *triggers* the adapter deployment. Once the adapter is deployed, a success message is displayed indicating successful deployment.

   After the above steps are completed, the adapter is available in the Design workspace of the SAP Integration Suite tenant.

2.2.2 Adapter Installation without Creating a New Integration Flow

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

Procedure

1. Go to the **Discover** workspace.
2. In the search box, search for **Adobe Sign 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 the Design workspace and select the copied **Adobe Sign 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.

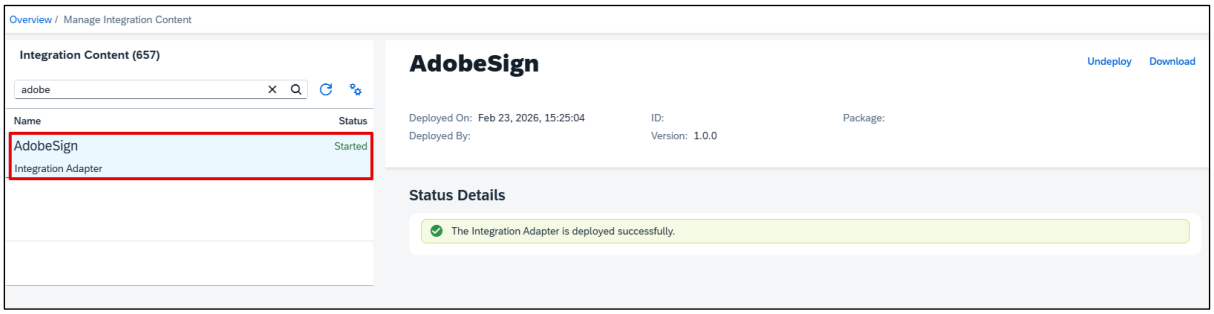
Purpose

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 the **Manage Integration Content** section and click **All**. This opens the **Integration Content** page with a list of all the deployed adapters.

You can check and confirm the deployment status of your adapter as shown in the screenshot below.

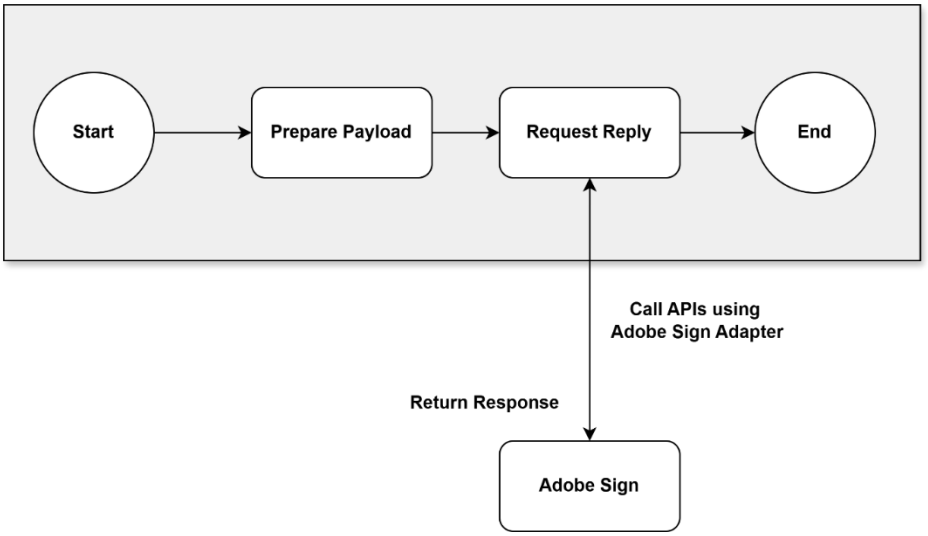


3. Getting Started: Adobe Sign Adapter

This section explains the setup required before configuring the Adobe Sign Adapter, you can find information about adapter architecture, application configuration, and authentication for the Adobe Sign Adapter.

3.1 Architecture Overview

How Adobe Sign Receiver Adapter Works:



- The Adobe Sign Receiver adapter is designed to receive and process data from an integration flow in SAP Integration Suite and facilitate communication with the Adobe Sign application (external) using REST-based APIs.
- SAP Integration Suite tenant acts as the initiator of the calls and sends a request to Adobe Sign (this is the receiver system) using the adapter.
- Additionally, the adapter also handles responses and logs execution status and errors before returning them to the SAP Integration Suite tenant.

3.2 Application Configuration

This section contains links to setup and configure the Adobe Sign application. These steps are a prerequisite before you start using the adapter in an integration scenario.

3.2.1 Introduction

- To explore Adobe Sign, see Getting Started with Adobe Sign.
- To get started with Adobe Sign, see Log in to Adobe Sign.

3.2.2 API Overview

- For an API overview of Adobe Sign, see API Overview.
- To explore APIs in detail, see Adobe Sign REST API.

3.2.3 Creating Integration Key

- For more information on how to create an integration key, see Integration Key.
    - To learn more about Permission, see Adobe Sign Scopes.

3.3 Authentication

This section details the authentication mechanism supported by the Adobe Sign Adapter in the SAP Integration Suite.

The adapter supports standard authentication, such as an **Integration Key**. You can securely store the security artifacts in SAP Security Material. This ensures that credentials can be safely provided to the Adapter.

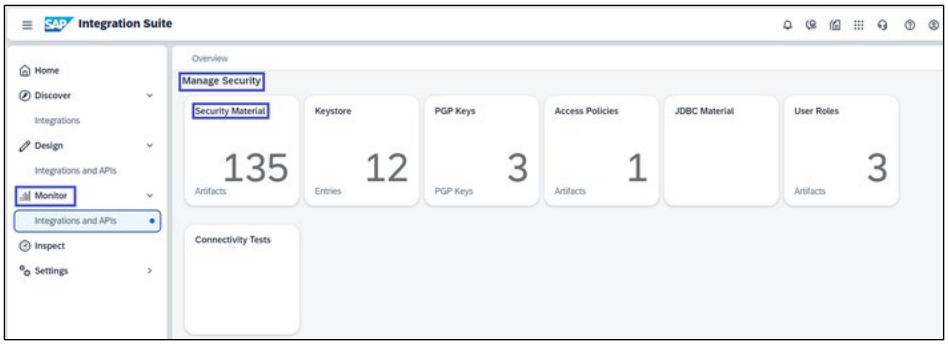
3.3.1 Creating a Secure Parameter in Security Material

Purpose

To create a security artifact that stores the Integration Key.

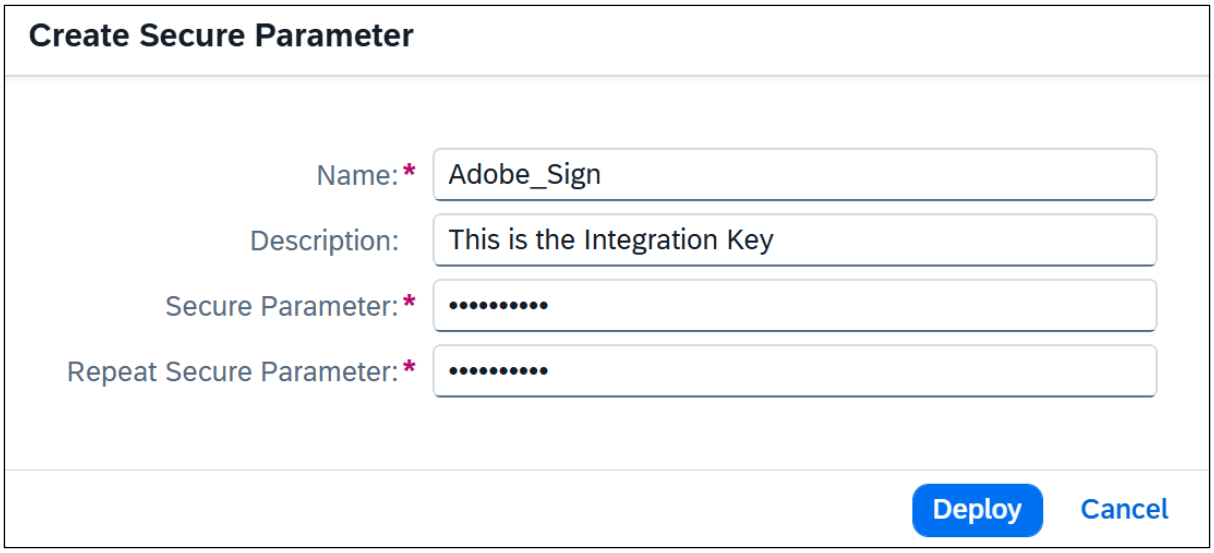
Procedure

1. Navigate to **Monitor > Integrations** and APIs in SAP Integration Suite. 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** and select **Secure Parameter** from the dropdown.

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



Parameter	Description
Name	Specify the name of the security artifact. The artifact name is used as an alias for confidential data. The artifact name is used as an alias for the confidential data assigned by this parameter.

Parameter	Description
Description	Enter a description for the artifact (optional).
Secure Parameter	Enter the confidential attribute value (**Integration Key**). For more information about Integration Key, see 3.2.3 Creating Integration Key
Repeat Secure Parameter	Repeat the confidential attribute value (**Integration Key**).

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. Adobe Sign Adapter Configuration

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

4.1 Receiver Adapter

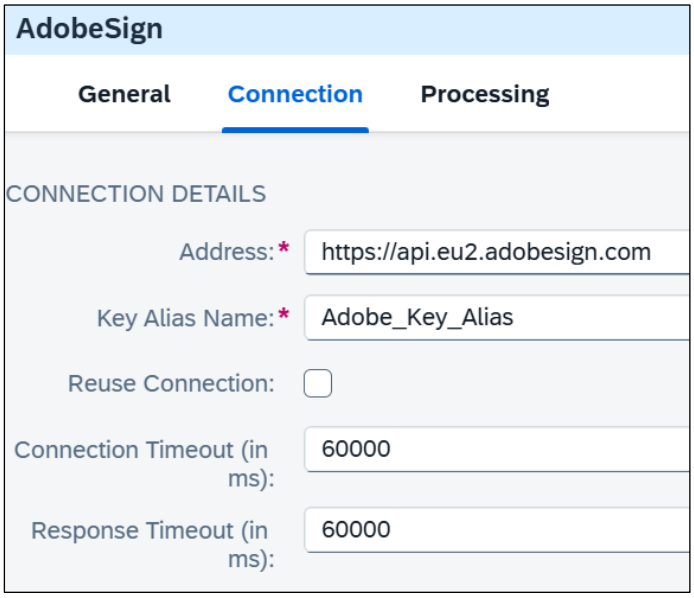
In this section, you will learn how to configure the Adobe Sign receiver adapter. On selecting the Adobe Sign adapter from the list of adapters, you must configure the **Connection** and **Processing** tabs.

4.1.1 Connection

The Connection tab contains connection and authentication parameters for the Adobe Sign adapter.

The Security artifact created in the previous section (Authentication) should be used in the **Connection tab** of the Adapter.

A sample screenshot of the configuration for the Connection tab is given below.

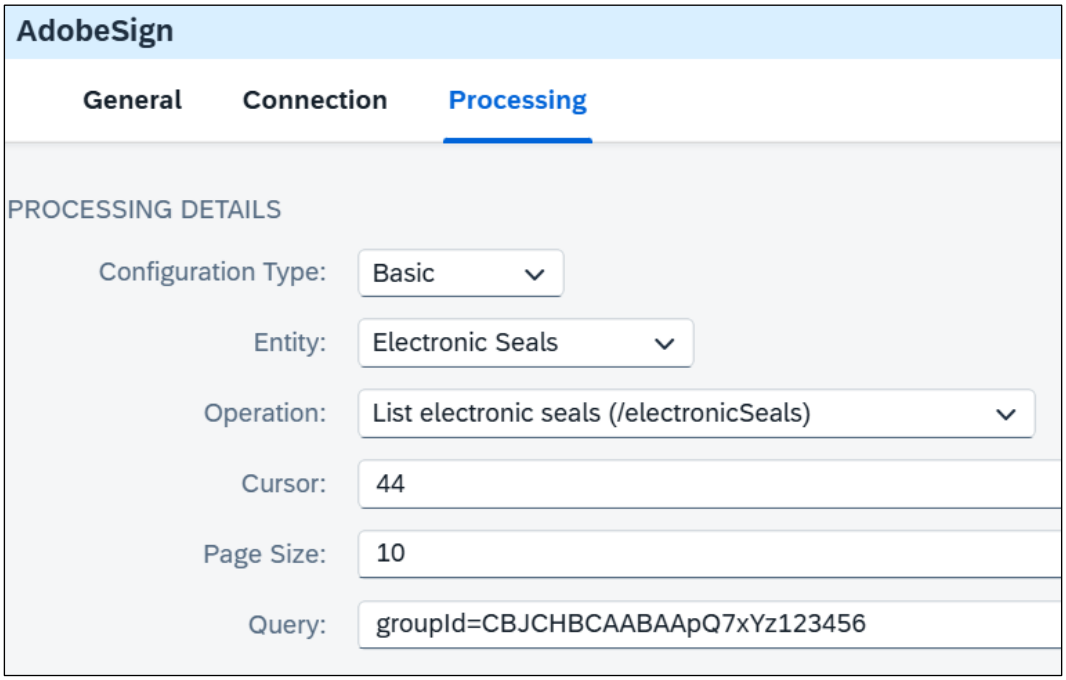


The connection tab contains the following fields:

Parameter	Description
Address	Specify the Adobe Sign Address. Example: `https://api.eu2.adobesign.com`
Key Alias Name	Specify the Secure Parameter Alias that stores the Integration Key to connect to Adobe Sign. For more information, see 3.2.3 Creating Integration Key
Reuse Connection	Enable to reuse the connection.
Connection Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for the connection to be established. Default: `60000`
Response Timeout (in ms)	Specify the maximum waiting time (in milliseconds) for a response message. Default: `60000`

4.1.2 Processing

This section lists the processing tab configurations for Adobe Sign Adapter. The following screenshot shows a sample configuration for List electronic seals operation.



Parameter	Description
Configuration Type	Select the required configuration type: • Select **Basic** to use the dropdowns and parameter text fields. • **Advanced** to specify the relative URL.
Entity (Only available when the **Configuration Type** is **Basic**)	Select the entity based on which the operation will be performed.
Operation (Only available when the **Configuration Type** is **Basic**)	Select the desired operation from the dropdown.
Operation Parameters (Only available when the **Configuration Type** is **Basic**)	Specify the operation parameters as a key-value pair.
Cursor	Specify the cursor value as the starting point for retrieving the items. If not provided, the first page is returned.
Page Size	Specify the maximum number of items to be returned per page. Example: `50`
Request Payload Source (Only available when the **Configuration Type** is **Basic**)	Select an option to specify the source for the request payload: • **UI Configurable** provides user-friendly fields to create the payload automatically. • **Exchange Body** allows you to specify the required structure and values directly as part of the payload.

Parameter	Description
HTTP Method (Only available when the **Configuration Type** is **Advanced**)	Select the required 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: `/api/rest/v6/agreements`
Enable Payload as Request Body (Only available when the **Configuration Type** is **Advanced**)	Select to pass the payload as the request body. In Content Modifier, set header as **content-type = application/json**
Query	Specify the expression containing the query parameter and value. Example: `"param1=value1&param2=value2"` While using space ( ), ensure that you encode the query using `%20`.
File Name	Specify the file name for the uploaded file. Example: `Invoice_Agreement.pdf` Ensure that you enter the file extension.
Mime Type	Specify the content type of the document being uploaded. Example: `application/pdf`

Parameter	Description
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. Note that this value can also be read dynamically using an exchange header or property.
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. Note that this value can also be read dynamically using an exchange header or property.

5. Adobe Sign Operations

This section lists and describes some of the operations supported by the Adobe Sign adapter.

The adapter provides two configuration types: **Basic** and **Advanced**. The **Basic** entity enables you to easily select predefined operations and their corresponding endpoints. The **Advanced** entity offers greater flexibility by allowing users to work with multiple endpoints for the same operation, including endpoints that are not predefined in the adapter, thereby supporting more customised integration scenarios.

5.1 Agreement: Create and send agreement (/agreement)

Create an agreement and send it to the specified recipients for electronic signatures. In the scenario below, we will create and send the agreement.

The operations can be performed in two ways:

1. Basic

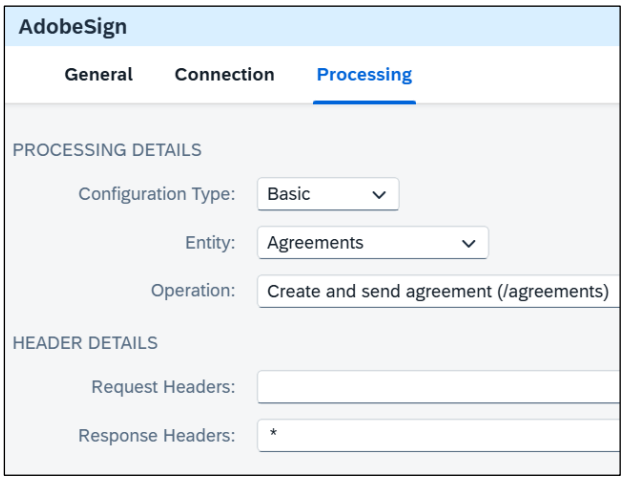
Once the agreement is successfully created, the system returns a unique agreement ID in response.
This agreement ID can be used to track the agreement status, manage reminders, and perform follow-up actions such as retrieving the signed document or cancelling the agreement.

Sample Payload for Create and Send Agreement:

```
{
  "name": "Employment Offer Letter",
  "fileInfos": [
    {
      "libraryDocumentId": "CBJCHBCAABAAabcd1234"
    }
  ],
  "participantSetsInfo": [
    {
      "memberInfos": [
        {
          "email": "recipient@example.com"
        }
      ],
      "order": 1,
```

```
      "role": "SIGNER"
    }
  ],
  "signatureType": "ESIGN",
  "state": "IN_PROCESS",
  "message": "Please review and sign the document."
}
```

A sample screenshot of the configuration to create a ticket is provided below.



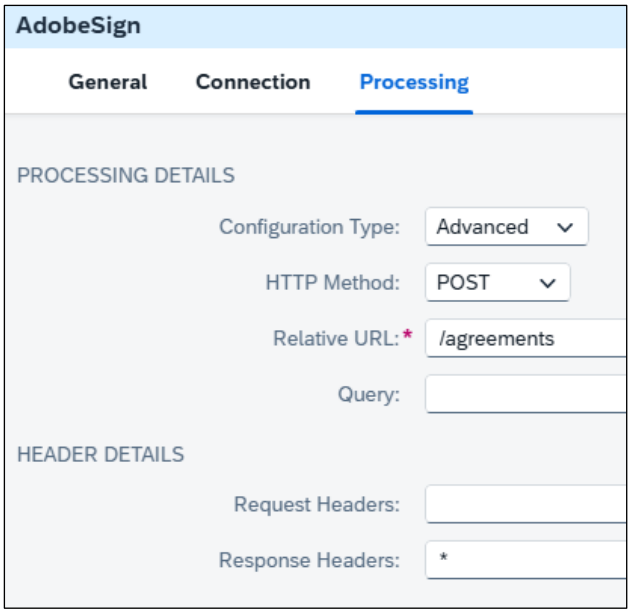
For the complete list of descriptions about the fields below, refer to the Processing tab.

Parameter	Value
Configuration Type	Select **Basic** as the configuration type.
Entity	Select **Agreements** as the entity.
Operation	Select **Create and send agreement (/agreements)** as an operation.

2. Advanced

To use the Advanced Method, follow the instructions below.

A sample screenshot of the configuration is provided below for Create and Send Agreement.



For the complete list of descriptions about the fields below, refer to the Processing tab.

Parameter	Value
Configuration Type	Select **Advanced** as the configuration type.
HTTP Method	Select **POST** as the HTTP Method.
Relative URL	Specify as `/agreements`

Response:

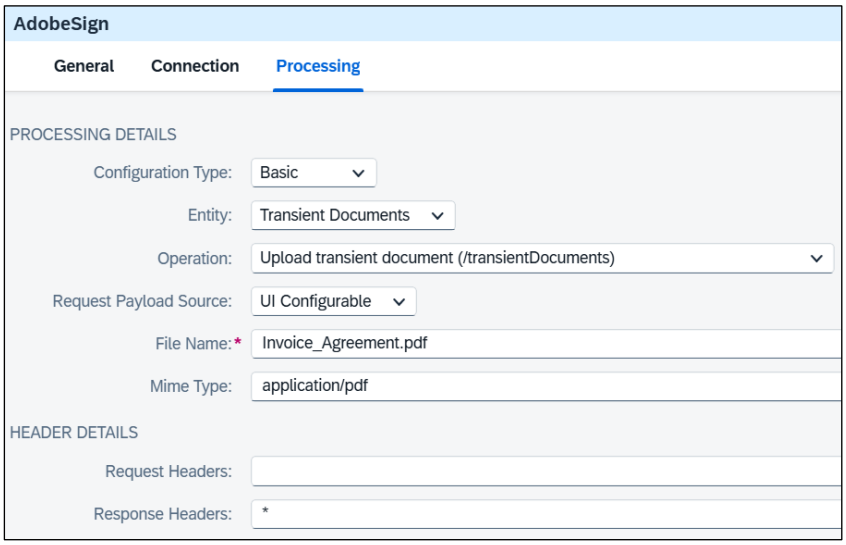
```
{
    "id": "CBJCHBCAABAAJ-BsAgeSfAZ3ohrl8xqpr_Hd01sGwY0X"
}
```

For more information, see Agreement API.

5.2 Transient Documents: Upload transient document (/transientDocuments)

This operation uploads a document as a **transient document**, which is temporarily stored and available for use for up to **7 days** and only one file per request.

The **transient document ID** that is returned can be referenced in subsequent API calls that require the uploaded file.



Response:


```
{
    "transientDocumentId":
        "CBSCTBABDUAAABACAABAA_x3ahsjsRvP4gXp_wElIz_-xq4ri8-Vb9G2bK-8_ksAB7LXbY-Q0T4ft2yHKH1QDXsEnBzz2P4UiYZ35E8XUgKUAKF-H4GXI5-qJFf_8jIqOyj-aYS5gnLCxq_B-h5y1dOO-ZAHa3
        4FOERrMHN3awKGUXMYX45U2FyyYOjIHvUGGsgzC527n1AWHY6FlUbZ6yOJA1gUjPAzS8AXg3KlOaohCsnd7gb1NBXiz1Ea_Zz17vD96r6bBUaSyBY627Zr7svI2emFJ1gTT4mggwtMuoZvbbkscCNg9ZJz8l3Bz
        sA6zj-v9QFpb6ydzv3sVzyYonREGuogB2PIyqTrwcdq4y1KmD8-fGGwX0X_W4daSUpk*"
}
```

For the complete list of descriptions about the fields below, refer to the Processing tab.

Parameter	Value
Configuration Type	Select **Basic** as the configuration type.
Entity	Select **Transient Documents** as the entity.
Operation	Select **Upload transient document (/transientDocuments)** as operation
Request Payload Resource	Select **UI Configurable** as a payload resource.
File Name	Specify the file name for the uploaded file. Example: `Invoice_Agreement.pdf` Ensure that you enter the file extension.
Mime Type	Specify the content type of the document being uploaded. Example: `application/pdf`

For more information, see Transient Documents API.

5.3 MegaSigns: Create MegaSign (/megaSigns)

The **Create MegaSign** operation allows you to create a **MegaSign** agreement to send the same document to multiple recipients in a single request. Each recipient receives an individual agreement instance and can sign independently.
This operation simplifies bulk document distribution and returns a MegaSign ID for tracking and management.

The operation can be performed in two ways:

1. **Basic**

A sample payload is provided to change the description of the ticket.

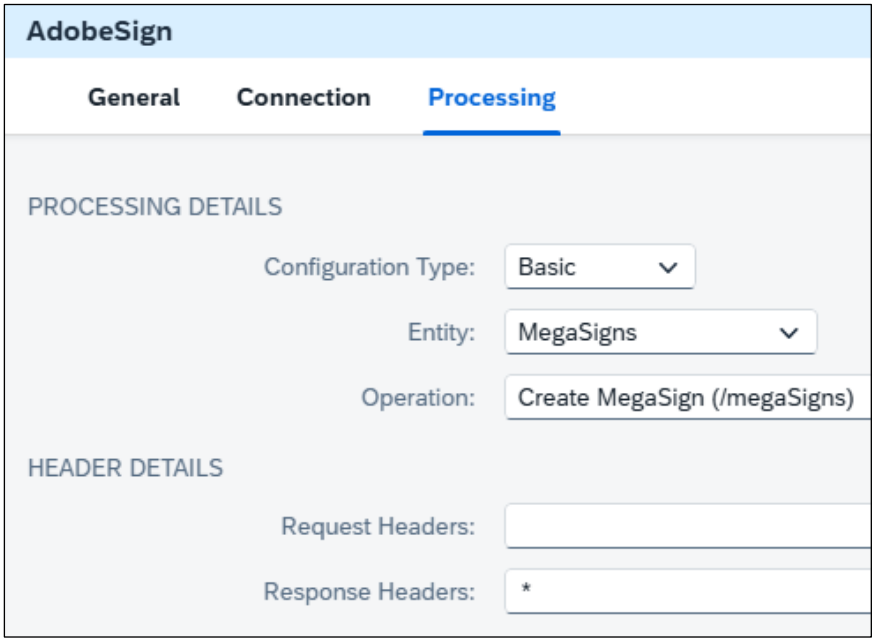
Note: To perform the **megaSign create** operation, you should upload a .csv file containing name and email addresses of recipients as shown in the payload below.

```
Recipient_1:Email,Recipient_1:Name,Recipient_1:Role
Signer1@gmail.com,Signer1,SIGNER
Signer2@gmail.com,Signer2,SIGNER
Signer3@gmail.com,signer3,SIGNER
```

A sample payload is provided to create a MegaSign.

```
{
  "name": "Employee Bulk Signing",
  "fileInfos": [
    {
      "transientDocumentId": "<Id of the PDF need to be sent.>"
    }
  ],
  "childAgreementsInfo": {
    "fileInfo": {
      "transientDocumentId": "<Csv ID of recipient>",
      "fileType": "CSV"
    }
  },
  "signatureType": "ESIGN",
  "state": "IN_PROCESS"
}
```

A sample screenshot of the configuration is provided below.

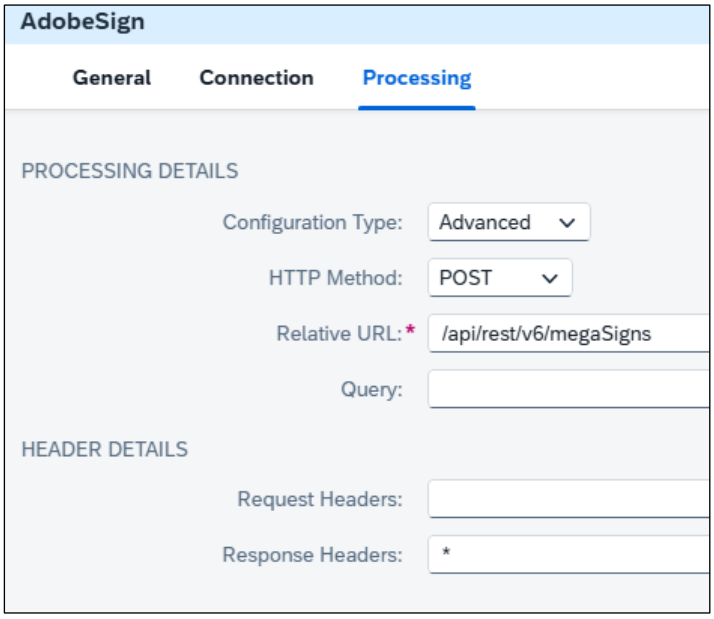


For the complete list of descriptions about the fields below, refer to the Processing tab.

Parameter	Value
Configuration Type	Select **Basic** as the configuration type.
Entity	Select **MegaSigns** as the entity.
Operation	Select **Create MegaSign (/megaSigns)** as an operation.

2. Advanced

A sample screenshot of the configuration is provided below, and the payload can be the same as the basic operation.



For the complete list of descriptions about the fields below, refer to the Processing tab.

Parameter	Value
Configuration Type	Select **Advanced** as the configuration type.
HTTP Method	Select **Post** as the entity.
Relative URL	Specify as `/api/rest/v6/megaSigns`

Response:

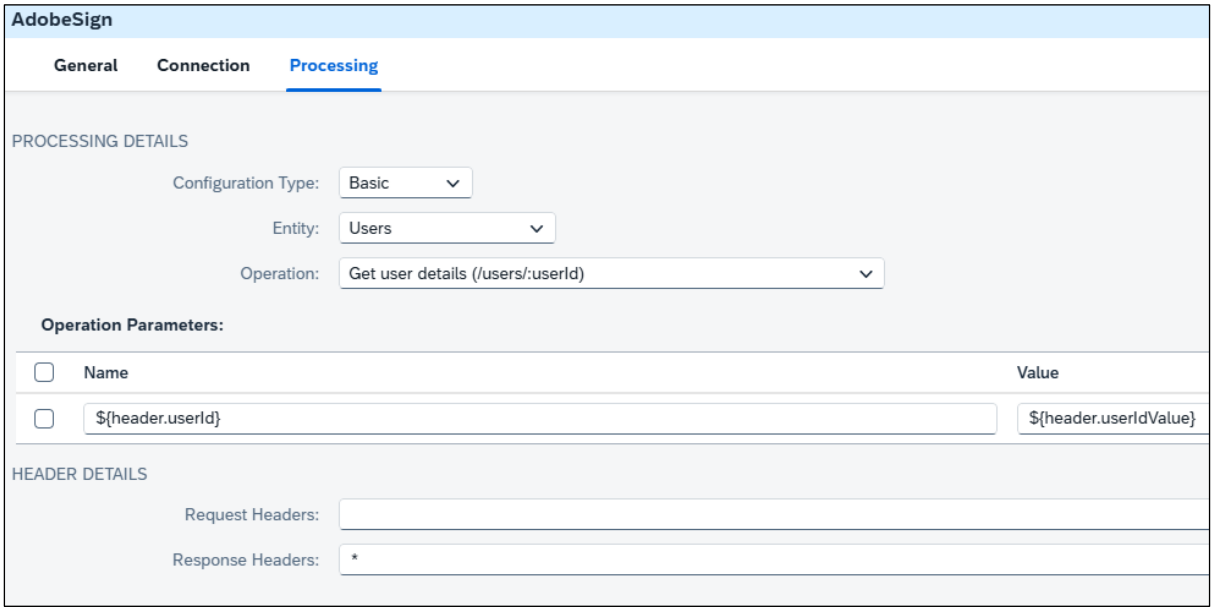
```
{
    "id": "CBJCHBCAABAAal9LLwHzTOaXGj2pdKC7hA09aGZvso39"
}
```


For more information, see Mega Signs.

5.4 User: Get User Details (/users/:userId)

The **Get user details** operation allows you to retrieve detailed information for a specified Adobe Sign user, including profile, account, and status details.

This operation helps applications identify user attributes and manage user-related workflows within Adobe Sign.



For the complete list of descriptions about the fields below, refer to the Processing tab.

Parameter	Value
Configuration Type	Select **Basic** as the configuration type.
Entity	Select **Users** as the entity.
Operation	Select **Get User Details (/users/:userId)** as operation

Parameter	Value
Operation Parameters	Specify the query parameter as a key-value pair. **Name** `${header.userId}` **Value** `${header.UserIdValue}`

Sample Response to retrieve details of the user:


```
{
    "id": "CBJCHBCAABAAOKOiX36b_FfdiZQkoti0qsO0vHQiFE_Y",
    "accountType": "PRO",
    "accountId": "CBJCHBCAABAA9K_MEeNXptK4rQpY2EoixbH70V40MiHv",
    "company": "string",
    "email": "sam_94406@pph.com",
    "phone": "123456789",
    "title": "Mr",
    "initials": "SD",
    "firstName": "Sam_94406",
    "lastName": "deo",
    "createdDate": "2026-02-12T13:04:02Z",
    "isAccountAdmin": false,
    "locale": "en_US",
    "status": "INACTIVE"
}
```


For more information, see User API.

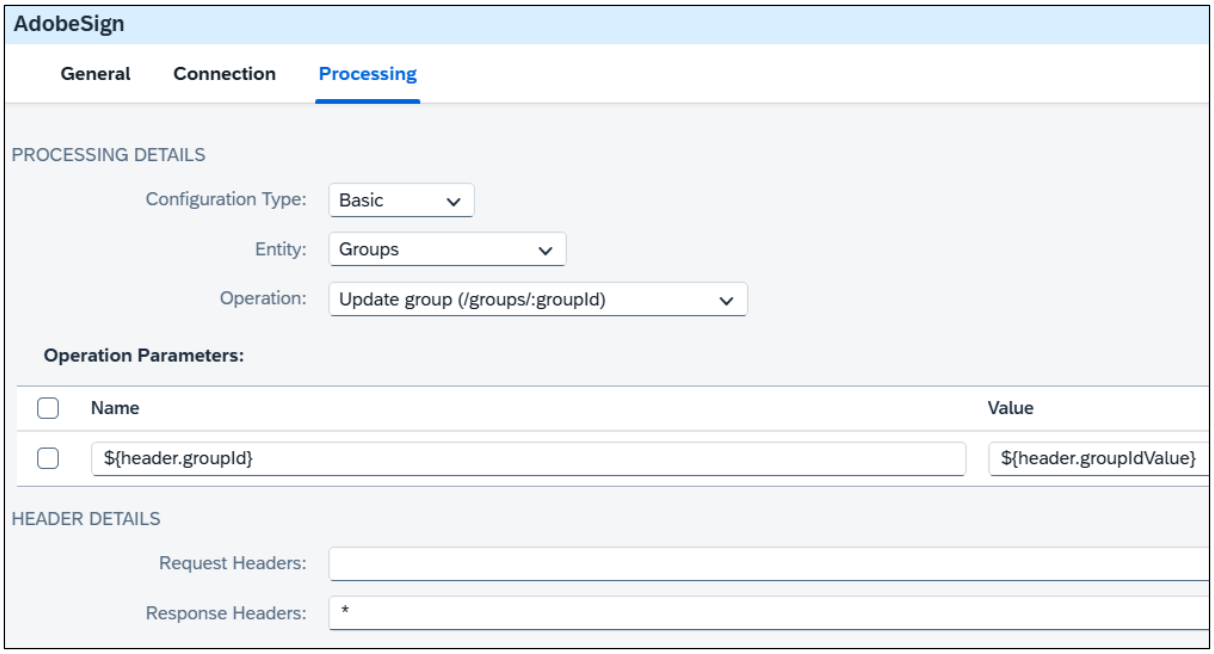
5.5 Group: Update Group (/groups/:groupId)

A **Group** is a logical unit used to organize users and manage their document workflows and control permissions within an account.

The **Updates group** operation allows you to update details of an existing Adobe Sign group identified by the specified **groupId**.

This operation allows you to modify group properties to manage users, permissions, and workflow settings within the group.

A sample screenshot of the configuration is provided below.



A sample payload is provided below.

```
{
  "name": "Ticket Sign-off"
}
```

For the complete list of descriptions about the fields below, refer to the Processing tab.

Parameter	Value
Configuration Type	Select **Basic** as the configuration type.
Entity	Select **Groups** as the entity.
Operation	Select **Update Group (/group/:groupId)** as operation

Sample Response:

```
{
  "name": "Ticket Sign-off Group"
}
```

For more information, see Update Group.

6. Support

This section is used to document additional steps for Adobe Sign.

6.1 Tip

You can monitor, debug, and analyze errors or issues by changing the Log level of your integration flow to Traces. For more information, see Tracing.

6.1.1 Troubleshooting

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

Error Message	Error Reason	Possible Solution
`Reason: Invoke api failed, urlPath: https://api.na1.adobesign.com/api/rest/v6/agreements status:429 response: Too Many Requests errorEntity: {"code":"THROTTLING_TOO_MANY_REQUESTS","retryAfter":62020,"message":"You have reached the limit on the number of agreements you can send at this time. Please try again in 18 hours."]`	API enforces limits and throttling for performance optimization.	For more information, see Transaction Limit.