

DocuSign Connector

Last Modified on 03/11/2024 4:02 am EDT

Authentication

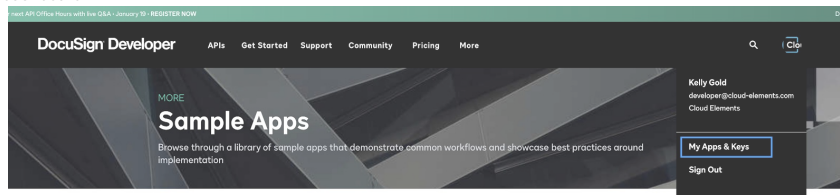
The DocuSign connector uses three authentication workflows - **Oauth2**, **JWT0auth2** and **Oauth2password**. You can authenticate a connector instance using the UI or via API calls. For detailed information on how to authenticate an instance, see our related documentation:

- [Authenticate a Connector Instance \(UI\)](#)
- [Authenticate a Connector Instance \(API\)](#)

All three workflows are discussed in detail further in this article.

Oauth2

1. Sign in to <https://developers.docusign.com/> to procure the API key and API secret from the administrator dashboard.



2. You may need to create an app to get the credentials required for authentication. While creating the app, the redirect URI should be <https://auth.cloudelements.io/oauth>.

You can find more details at [How to get an access token - DocuSign](#).

Required Parameters

To authenticate an instance of the DocuSign connector using Oauth2, you will need the following parameters:

Parameter Name/UI Value	API Value/Key	Description
DocuSign Environment	docusign.environment	The environment on which your DocuSign app is configured
Oauth API Key	oauth.api.key	DocuSign Beta Oauth API key
Oauth API Secret	oauth.api.secret	DocuSign Oauth Secret

Sample Configuration JSON

```
"configuration": {
  "docusign.environment": "demo",
  "authentication.type": "MyUsername",
  "oauth.api.key": "MyUsername",
  "oauth.api.secret": "MyUsername",
}
```

JWT0auth2

To authenticate an instance of the DocuSign connector using **JWT0auth2**, you will need to sign in to <https://developers.docusign.com/> and check for details on <https://developers.docusign.com/platform/auth/jwt/jwt-get-token/>.

Required Parameters

Parameter Name/UI Value	API Value/Key	Description
JWT Base64 Encoded Key	base64Key.pem	JWT Base64 Encoded Key
JWT Issuer	jwt.claim.iss	The integration key, also known as the App Id, of the application
OAuth API Key	oauth.api.key	DocuSign Beta Oauth API key is also an App ID
JWT Subject	jwt.claim.sub	The user Id of the user to be impersonated. For example: https://share.getcloudapp.com/jkuP60LP

Create a JWT Base64 Encoded Key

JWT Base64 Encoded Key is a `Base64` encoded private key. Here is how you can get the value:

1. Create the **private key**.

```
openssl genrsa -des3 -out private.pem  
2048;
```

2. Create the **public key**.

```
openssl rsa -in private.pem -outform PEM -pubout -out  
public.pem;
```

3. Convert the private key to `PKCS8` and `DER` format (for the Java code base to read).

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in private.pem -out private_key_pkcs8.der -  
nocrypt;
```

4. Base64 encode the private key.

```
cat private_key_pkcs8.der | base64 >  
base64Key.pem;
```

5. Create a certificate and sign it with the private key.

```
openssl req -new -x509 -key private.pem -out cacert.pem -days  
1095;
```

6. After running the steps above, you must provide an instance as `public.pem` and `base64Key.pem`

Upload the file from the `public.pem` of the Public key in your application as shown [here](#) and then use the file from `base64Key.pem` as `jwtBase64EncodedKey`.

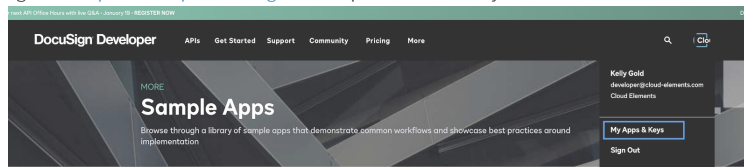
Note: The JWT Grant flow access tokens expire after one hour, with no refresh token. Once the token expires, you must generate a new JWT and exchange it for a new access token.

Sample Configuration JSON

```
"configuration": {  
  "docusign.environment": "demo",  
  "filter.response.nulls": "true",  
  "authentication.type": "jwtOauth2",  
  "synchronous.bulk.notification": "true",  
  "jwt.base64.encoded.key": "xxxxxxxx",  
  "jwt.claim.iss": "xxxxxxxx",  
  "oauth.api.key": "xxxxxxxx",  
  "jwt.claim.sub": "xxxxxxxx"  
}
```

Oauth2 Password

1. Sign in to <https://developers.docusign.com/> to procure the API key from the administrator dashboard.



2. You may need to create an app to get the credentials required for authentication. While creating the app, the redirect URI should be <https://auth.cloudelements.io/oauth>.

You can find more details at [Implicit Grant Authentication - DocuSign](#).

Required Parameters

Parameter Name/UI Value	API Value/Key	Description
Docusign Environment	docusign.environment	The environment on which your DocuSign app is configured
OAuth API Key	oauth.api.key	DocuSign Beta OAuth API key
Username	username	DocuSign Beta OAuth Username
Password	password	DocuSign Beta OAuth Password

Sample Configuration JSON

```

"configuration": {
  "docuSign.environment": "demo",
  "filter.response.nulls": "true",
  "authentication.type": "oauth2password",
  "synchronous.bulk.notification": "true",
  "username": "xxxxxxxxxx",
  "oauth.api.key": "xxxxxxxx",
  "password": "xxxxxxxx"
}

```

Optional Authentication Parameters

These are the optional parameters you will need, to authenticate an connector instance via any of the methods discussed above.

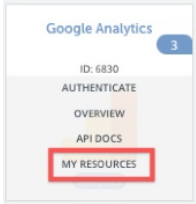
Parameter Name	API Value	Description
Account ID	accountId	Account ID
Webhook connect ID	webhook.connect.id	ID used to establish webhook handshakes with the platform and depends on enabling webhooks
Default select fields	default.select.fields.map	Generic map used to specify default fields for bulk download and GET /all requests. Each key should be the canonical objectName and values may be specified as a comma-delimited string or a list of strings.
Instance Variables	instance.variables	States what the instance variables are
Synchronous Bulk Notification	synchronous.bulk.notification	Enables synchronous bulk callback notification
Filter Null Values from Response	filter.response.nulls	Used to enable/disable filtering of null values from the responses

Connector Fact Sheet and Reference

Authentication and Configuration Parameters

To see all authentication parameters for the DocuSign connector, follow these steps:

1. Sign in to SAP Open Connectors and navigate to Connectors.
2. Hover over the card for , and then click My Resources.



3. In the top navigation toolbar, click Setup.
4. From the **Setup** page, you can view a complete list of the connector's authentication and configuration parameters.

Connector Resources

In order to provide users with normalized APIs across Hubs, SAP Open Connectors resources were mapped to the following Vendor resources.

Platform Placeholder	Platform Method	Platform Resource	Vendor Method	Vendor Resource
brands	GET	/brands	GET	/brands
	GET/{id}	/brands/{id}	GET/{id}	/brands/{id}
custom-tabs	GET	/custom-tabs	GET	/tab-definitions
	GET/{id}	/custom-tabs/{id}	GET/{id}	/tab-definitions/{id}
	GET	/envelopes	GET	/envelopes
	GET/{id}	/envelopes/{id}	GET/{id}	/envelopes/{id}
	POST	/envelopes	POST	/envelopes
	PATCH	/envelopes/{id}	PATCH	/envelopes/{id}
	GET	/envelopes/{id}/attachments	GET	/envelopes/{id}/attachments
	POST	/envelopes/{id}/attachments	POST	/envelopes/{id}/attachment
	PATCH	/envelopes/{id}/attachments/{attachmentId}	PATCH	/envelopes/{id}/attachments/{attachmentId}
	DEL	/envelopes/{id}/attachments/{attachmentId}	DEL	/envelopes/{id}/attachments/{attachmentId}

Platform Placeholder	Platform Method	Platform Resource	Vendor Method	Vendor Resource
	GET/{id}	/envelopes/{id}/attachments/{attachmentId}	GET	/envelopes/{id}/attachments/{attachmentId}
	GET	/envelopes/{id}/comments	GET	/envelopes/{id}/comments/transcript
	POST	/envelopes/{id}/console-url	POST	/view/console
	GET	/envelopes/{id}/custom-fields	GET	/envelopes/{id}/custom-fields
	POST	/envelopes/{id}/custom-fields	POST	/envelopes/{id}/custom-fields
	PATCH	/envelopes/{id}/custom-fields/	PUT	/envelopes/{id}/custom-fields
	DEL	/envelopes/{id}/custom-fields/{customFieldId}	DEL	/envelopes/{id}/custom-fields
	GET	/envelopes/{id}/custom_fields	GET	/envelopes/{id}/custom_fields
	GET	/envelopes/{id}/definitions	GET	/envelopes/{id}/html_definitions
	GET	/envelopes/{id}/documents	GET	/envelopes/{id}/documents
	PUT	/envelopes/{id}/documents	PUT	/envelopes/{id}/documents
	GET	/envelopes/{id}/documents/{documentId}	GET	/envelopes/{id}/documents/{documentId}
	GET	/envelopes/{id}/documents/certificates	GET	/envelopes/{id}/documents/certificate
	PATCH	/envelopes/{id}/documents/{documentId}	PUT	/envelopes/{id}/documents
	DEL	/envelopes/{id}/documents/{documentId}	DEL	/envelopes/{id}/documents
	GET	/envelopes/{id}/documents/{documentId}/definitions	GET	/envelopes/{id}/documents/{documentId}/htr
	GET	/envelopes/{id}/documents/{documentId}/fields	GET	/envelopes/{id}/documents/{documentId}/fiel
	POST	/envelopes/{id}/documents/{documentId}/fields	POST	/envelopes/{id}/documents/{documentId}/fiel
	PATCH	/envelopes/{id}/documents/{documentId}/fields/{fieldName}	PUT	/envelopes/{id}/documents/{documentId}/fiel
	DEL	/envelopes/{id}/documents/{documentId}/fields/{fieldName}	DEL	/envelopes/{id}/documents/{documentId}/fiel
	GET	/envelopes/{id}/documents/{documentId}/tabs	GET	/envelopes/{id}/documents/{documentId}/tat
envelopes	GET	/envelopes/{id}/documents/{documentId}/templates	GET	/envelopes/{id}/documents/{documentId}/ter
	POST	/envelopes/{id}/documents/{documentId}/templates	POST	/envelopes/{id}/documents/{documentId}/ter
	DEL	/envelopes/{id}/documents/{documentId}/templates/{templateId}	DEL	/envelopes/{id}/documents/{documentId}/ter
	POST	/envelopes/{id}/edit-view-url	POST	/envelopes/{id}/views/edit
	GET	/envelopes/{id}/email-settings	GET	/envelopes/{id}/email_settings
	POST	/envelopes/{id}/email-settings	POST	/envelopes/{id}/email_settings
	PUT	/envelopes/{id}/email-settings	PUT	/envelopes/{id}/email_settings
	DEL	/envelopes/{id}/email-settings	DEL	/envelopes/{id}/email_settings
	POST	/envelopes/{id}/envelope-correction-url	POST	/envelopes/{id}/views/correct
	GET	/envelopes/{id}/form-data	GET	/envelopes/{id}/form_data
	POST	/envelopes/{id}/html-preview	POST	/envelopes/{envelopeld}/responsive_html_pr
	GET	/envelopes/{id}/lock	GET	/envelopes/{id}/lock
	POST	/envelopes/{id}/lock	POST	/envelopes/{id}/lock
	PUT	/envelopes/{id}/lock	PUT	/envelopes/{id}/lock
	DEL	/envelopes/{id}/lock	DEL	/envelopes/{id}/lock
	POST	/envelopes/{id}/recipient-views	POST	/envelopes/{envelopeld}/views/recipient
	GET	/envelopes/{id}/recipients	GET	/envelopes/{id}/recipients
	POST	/envelopes/{id}/recipients	POST	/envelopes/{id}/recipients
	PATCH	/envelopes/{id}/recipients/{recipientId}	PUT	/envelopes/{id}/recipients
	PUT	/envelopes/{id}/recipients/document-visibility	PUT	/envelopes/{envelopeld}/recipients/documer
	DEL	/envelopes/{id}/recipients/{recipientId}	DEL	/envelopes/{id}/recipients/{recipientId}
	GET	/envelopes/{id}/recipients/{recipientId}/document-visibility	GET	/envelopes/{id}/recipients/{recipientId}/docur
	PUT	/envelopes/{id}/recipients/{recipientId}/document-visibility	PUT	/envelopes/{envelopeld}/recipients/{recipient
	GET	/envelopes/{id}/recipients/{recipientId}/tabs	GET	/envelopes/{id}/recipients/{recipientId}/tabs
	POST	/envelopes/{id}/recipients/{recipientId}/tabs	POST	/envelopes/{id}/recipients/{recipientId}/tabs
	PATCH	/envelopes/{id}/recipients/{recipientId}/tabs/{tabId}	DEL	/envelopes/{id}/recipients/{recipientsId}/tabs
	PUT	/envelopes/{id}/recipients/{recipientId}/tabs/{tabId}	PUT	/envelopes/{id}/recipients/{recipientsId}/tabs
	POST	/envelopes/{id}/sender-view-url	POST	/envelopes/{id}/views/sender
	GET	/envelopes/{id}/templates	GET	/envelopes/{id}/templates

Platform Placeholder	Platform Method	Platform Resource	Vendor Method	Vendor Resource
	POST	/envelopes/{id}/templates	POST	/envelopes/{id}/templates
groups	GET	/groups	GET	/groups
	GET	/groups/{id}/brands	GET	/groups/{id}/brands
	GET	/groups/{id}/users	GET	/groups/{id}/users
identity-verifications	GET	/identity-verifications	GET	/identity_verification
notary	GET	/notary	GET	/notary
notary-journals	GET	/notary-journals	GET	/notary/journals
notary-jurisdictions	GET	/notary-jurisdictions	GET	/notary/jurisdictions
payment-accounts	GET	/payment-accounts	GET	/payment_gateway_accounts
response	GET	/response	GET	/bulk_send_lists
	GET/{id}	/response/{id}	GET/{id}	/bulk_send_lists/{id}
	POST	/response	GET	/bulk_send_lists
	PATCH	/response/{id}	GET	/bulk_send_lists/{id}
	DEL	/response		/bulk_send_lists/{id}
seals	GET	/seals	GET	/seals
signature-providers	GET	/signature-providers	GET	/signatureProviders
signing-group	GET	/signing-groups	GET	/signing_groups
	GET/{id}	/signing-groups/{id}	GET/{id}	/signing_groups/{id}
	GET	/signing-groups/{id}/users	POST	/signing_groups/{id}/users
tab-settings	GET	/tab-settings	GET	/tab-settings
templates	GET	/templates	GET	/templates
	GET/{id}	/templates/{id}	GET/{id}	/templates/{templatelid}
	GET	/templates/{id}/custom-fields	GET	/templates/{id}/custom_fields
	GET	/templates/{id}/custom-fields	GET	/templates/{id}/custom_fields
	GET/{id}	/templates/{id}/documents/{documentId}	GET/{id}	/templates/{id}/documents/{documentId}
	GET	/templates/{id}/documents/{documentId}/fields	GET	/templates/{id}/documents/{documentId}/fiel
	GET	/templates/{id}/documents/{documentId}/html-definitions	GET	/templates/{id}/documents/{documentId}/htr
	GET	/templates/{id}/documents/{documentId}/html-preview	GET	/templates/{id}/documents/{documentId}/res
	GET	/templates/{id}/documents/{documentId}/tabs	GET	/templates/{id}/documents/{documentId}/tab
	GET	/templates/{id}/html_definitions	GET	/templates/{id}/html_definitions
	GET	/templates/{id}/recipients	GET	/templates/{id}/recipients
	GET	/templates/{id}/recipients/{recipientId}/document_visibility	GET	/templates/{id}/recipients/{recipientId}/docur
	GET	/templates/{id}/recipients/{recipientId}/tabs	GET	/templates/{id}/recipients/{recipientId}/tabs
	POST	/templates/{id}/html-preview	POST	/templates/{id}/responsive_html_preview
	Users	GET	/users	GET
POST		/users	POST	/users
GET		/users/{id}	GET	/users/{id}
GET		/users/{id}/profile	GET	/users/{id}/profile
GET		/users/{id}/profile/image	GET	/users/{id}/profile/image
GET		/users/{id}/settings	GET	/users/{id}/settings
GET		/users/{id}/signature	GET	/users/{id}/signature
GET		/users/{id}/signatures/{signatureId}/{imageType}	GET	/users/{id}/signatures/{signatureId}/{imageTy
userinfo		GET	/userinfo	GET

Events and Supported Resources

The DocuSign connector supports events via polling and webhooks. For detailed information about our Events framework and how to configure a connector instance with events, see our [documentation](#):

- [Authenticate a Connector Instance with Events \(UI\)](#)
- [Authenticate a Connector Instance with Events \(API\)](#)

You can set up polling events for only the `/envelopes` resource of the DocuSign connector:

Add a new tag to the element instance

Event Configuration

Events Enabled

Vendor Event Type

Event Notification Callback URL

Event Notification Basic Authentication Password

Event Notification Basic Authentication UserName

Event Notification Callback Headers

Callback Notification Signature Key

Event Poller Refresh Interval (mins)

The DocuSign connector supports all other events via webhooks.

Event Configuration

Events Enabled

Vendor Event Type

Webhook URL

`https://api.cloud-elements.com/elements/api-v2/events/docusign/MjE4Mzk3NQ==`

Event Notification Callback URL

Event Notification Basic Authentication Password

Event Notification Basic Authentication UserName

Event Notification Callback Headers

Callback Notification Signature Key

Event Poller Refresh Interval (mins)

Event Poller URLs

`envelopes{/hubs/esignature/envelopes?where=from_date=${gmtDate:yyyy-MM-dd'T'HH:mm:ss'Z'}}`

The optional configurations for events via webhooks are as shown below.

Account ID

Webhook connect id

Instance Variables

```
1 | Insert JSON object for variables here
```

Bulk

Bulk provides an option to upload a large number of resources, such as users, to a service provider all at once. You can use bulk to download a .csv or .json file from a large number of records or upload a .csv or .json file to add multiple records.

See [Using Bulk](#) for more information on managing bulk.

users ⓘ Bulk ↑↓

GET `/users` Retrieve list of all Users

POST `/users` Create a User

GET `/users/{id}` Retrieve a specific User by ID

+ Add a new endpoint

Note: When there is no native bulk available from the provider, SAP Open Connectors provides a bulk service for uploading and downloading data from the endpoint. DocuSign connector makes use of Bulk V3. For more information, see our [Bulk documentation](#).

- To make a `POST/bulk/query bulkv2` API call with a query, select * from signing-groups and check the results of a bulk job.

- Retrieve the results of the bulk job in both .csv and .json format. Notice that in .json the created and modified date/times do not show as expected and have an extra \ \ appended for both the fields.
- This exists only in .json, as the created & modified dates/times show up as expected in csv format.

Bulk Details

Bulk Information	Details
Bulk Type	SAP Open Connectors Bulk.
Upsert Support	No
Field Selection Support	Yes. You can specify fields within a resource. For example, <code>select name,id from contacts</code> .
Order By Support	No

Querying

The Open Connectors Query Language (OCNQL) provides a standard way to search across all of our connectors. Rather than having users research how to query each resource at a provider, we've normalized your search experience. SAP Open Connectors translates your queries to the API provider's search syntax. OCNQL represents the "where" portion of a typical query that can include many other components like field selection, sorting, and pagination. For more information on querying, see [Querying with OCNQL](#).

To determine whether where clause is supported for an object, you can make an API call to `object/{objectName}/metadata` while providing the Connector version and check for `"filterable"="true"`.

A few examples for usage of the where clause:

- `/envelopes?where=from_date='2020-07-02T13:41:00Z'`
- `/envelopes?where=from_date='2020-12-16T19:20:30.45+01:00' and envelopeld='d3f1a749-37e6-4a09-bc1c-70802a968f81'`
- `/envelopes?where=from_date='2000-12-16T19:20:30.45+01:00' and status='sent'`

There are several operators that are valid to be used for querying. You can find more details on them [here](#). But each object which has `"filterable"="true"`, can also tell you what operators can be used for that particular filterable.

Connector Extension

Connector Builder gives you the integration tools to extend the current feature set of any of our connectors by adding a resource or method. If your use case requires a resource our connector doesn't currently support, you can easily add it in yourself using Connector Builder. DocuSign is an extendable connector and therefore, you can customize it according to your needs.

For more details, read the [Extend Connectors](#) article which gives a description in detail as to how it can be done.

Discovery APIs

Discovery APIs are usually used in conjunction with a specific authenticated connector instance to:

- Identify data objects in a connector
- Identify data objects and their source (the vendor, a common resource, or SAP Open Connectors library)
- Find metadata, including custom fields, related to an object
- Retrieve the Open API specifications (API docs) of an object

However, Discovery APIs are **not supported** by SAP Open Connectors for the DocuSign connector, as there is no support by the vendor for the same. However, we do support the discovery of objects that are part of our implementation.

Connector API Documentation

The base URL for all API requests is <https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-v2> .

Provider Documentation

- [DocuSign API Documentation](#)