DocuSign
REST API Guide

Version 2

221 Main Street, Suite 1000, San Francisco, CA 94105 | Tel. 866.219.4318 | www.docusign.com | © DocuSign, Inc.

Copyright ©2003-2016 DocuSign, Inc. All rights reserved.

For information about DocuSign trademarks, copyrights and patents refer to the DocuSign Intellectual Property page (https://www.docusign.com/IP) on the DocuSign website. All other trademarks and registered trademarks are the property of their respective holders.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of DocuSign, Inc. Under the law, reproducing includes translating into another language or format. Every effort has been made to ensure that the information in this manual is accurate. DocuSign, Inc. is not responsible for printing or clerical errors. Information in this document is subject to change without notice.

DocuSign REST API Guide, version 2, August 5, 2016

If you have any comments or feedback on our documentation, please send them to us at: Documentation@DocuSign.com.

Summary of changes for this version:

- Updated the Get Shared Access Information and Set Shared Access Information descriptions to included changes for template sharing with users and groups.
- Added a new method to List Signature Providers for DocuSign Standards-Based Signatures. This method is available in the production version of the Signature REST API v2, it is still considered to be in beta and subject to change.
- Updated the Signers Recipient Type with new properties to use DocuSign Standards-Based Signature providers when sending documents for signing. These properties are available in the production version of the Signature REST API v2, but are still considered to be in beta and subject to change

Important: DocuSign is transitioning to a new REST API Reference Guide. The PDF version and old online version will be retired soon, please refer to the new guide.

Table of Contents

Introduction ... 11
Commonly Used Terms ... 11
Basic Process Flow and Steps for Sending with the REST API ... 12
Basic Process Flow for Requesting Status Changes with the REST API ... 12
Request and Response Guidelines ... 14
Request and Response Formats ... 14
Envelope and Recipient Status Codes ... 15
Getting Started ... 16
Integrator Keys ... 16
API Call Rate Limits ... 16
File Size Limitations ... 18
DocuSign Time Track Header ... 18
REST API Versions ... 18
REST API Base URL ... 19
REST API Explorer ... 19
Send On Behalf Of Functionality in the DocuSign REST API ... 19
DocuSign OAuth (Authorization Code Grant or Implicit Grant) ... 21
Configuring an App ... 21
Supported Grant Types ... 24
Making Requests ... 24
Authentication Request ... 24
Initiating the Authorization Code Flow ... 26
Initiating the Implicit Flow (Mobile Apps or Client Applications) ... 28
Access Token Usage ... 29
User Information ... 29
DocuSign OAuth (Resource Owner Credentials or SAML) ... 30
General Usage Notes ... 30
OAuth2 Token Request ... 31
Normal Usage ... 32
OAuth2 and SAML ... 32
Revoking Tokens ... 34
Acting as Other Account Users ... 35
OAuth2 Response Codes ... 37

Basic Scenarios ... 37
Creating an Envelope from a Document ... 38
Creating an Envelope from a Template ... 39
Embedded Sending ... 40
Embedded Signing ... 41
Requesting Envelope Status ... 42
Retrieving Envelope and Documents ... 43
Sending HTTP Requests with cURL ... 44
Setting Up Your Client Application ... 44
Viewing Base Resources with cURL ... 44
Logging in with cURL ... 45
Sending HTTP Requests Using REST API Resources ... 45
Chunked Transfer-Encoding Support ... 47
Example Chunked Transfer-Encoding ... 47
REST API References ... 49
Login ... 67
Change Password ... 68
Create Account ... 69
Create Multiple Accounts ... 86
Delete Account ... 88
Get Account Information ... 89
Get Account Billing Plan ... 90
Update Account Billing Plan ... 94
Get Billing Payment information ... 99
Post a Billing Payment ... 100
Get a List of Billing Invoices ... 102
Get a Billing Invoice ... 103
Get Past Due Invoices ... 105
Get Billing Charges ... 106
Purchase Additional Envelopes ... 109
Get Brand Profile Information ... 111
Upload Brand Profiles ... 112
Delete Brand Profiles ... 113
Checking Status of all Bulk Recipient Batches ... 114
Checking the Status of one Bulk Send Batch ... 115

Delete Captive Recipient Signature 117
Set Up a Connect Configuration 119
Get Connect Configuration Information 122
Update a Connect Configuration 123
Get a Connect Configuration Information 126
Delete a Connect Configuration 128
Republish Connect Information for One Envelope 128
Republish Connect Information for Multiple Envelopes 129
Get the Connect Failure Log 130
Get a Connect Failure Log Entry 132
Clear a Connect Failure Log Entry 134
Get the Connect Log 135
Clear the Connect Log 137
Get One Connect Log Entry 138
Clear One Connect Log Entry 140
Get Disclosure 141
Get List of Account Custom Fields 142
Send an Envelope or Create a Draft Envelope 144
Send an Envelope from a Template 158
Using Bulk Send in the REST API 172
Get Envelope Status Changes 173
Get Individual Envelope Status 178
Send Draft Envelope 180
Void Envelope 180
Modify Draft Envelope Email Subject and Message 181
Purge Documents 182
Add or Modify Envelope Information 184
Get Envelope Purge Document Status 188
Get Envelope Audit Events 190
Add Envelope Custom Fields to an Envelope 191
Get Envelope Custom Field Information 194
Modify Envelope Custom Fields for an Envelope 196
Remove Envelope Custom Fields from an Envelope 198
Add or Modify Documents and Document Fields 200
Remove Documents from a Draft Envelope 202

Get List of Envelope Documents 202
Add a Document to an Envelope 204
Get Document from Envelope 204
Get a Page Image 205
Rotate Page Image 206
Remove a Page 207
Get Custom Document Fields for an Envelope Document 208
Add Custom Document Fields to an Envelope Document 209
Modify Custom Document Fields for an Envelope Document 211
Delete Custom Document Fields from an Envelope Document 214
Get Envelope Certificate 216
Get Envelope Documents and Certificate 217
Add Email Setting Overrides to an Envelope 219
Get Email Setting Overrides for an Envelope 220
Modify Email Setting Overrides for an Envelope 221
Delete Email Setting Overrides for an Envelope 223
Lock an Envelope or Template 224
Get Envelope or Template Lock Information 226
Update Envelope or Template Lock 227
Delete Envelope or Template Lock 229
Get Envelope Notification Information 230
Get Envelope Recipient Status 231
Add Recipients to an Envelope 232
Delete Recipients from an Envelope 234
Modify or Correct and Resend Recipient Information 236
Add or Replace an Envelope Bulk Recipient File 237
Retrieve Envelope Bulk Recipient File 240
Delete Envelope Bulk Recipient File 242
Set Initials Image for Accountless Signer 242
Get Signature Information for Accountless Signer 243
Set Signature Image for Accountless Signer 244
Add Tabs for a Recipient 245
Get Tab Information for a Recipient 246
Delete Tabs for a Recipient 247
Modify Tabs for a Recipient 248

Get List of Templates Used in an Envelope 249
Post Envelope Correction 250
Post Recipient View 252
Post Sender View 256
Post Edit View 257
Get Envelope Status for more than one envelope 257
Get Folder List 259
Get Folder Envelope List 260
Move Envelope 262
Add a New Group 263
Get Group Information 264
Modify Group Information 266
Delete Group Information 267
Get Group Brand ID Information 269
Add Group Brand ID Information 270
Delete Group Brand ID Information 271
Add Users to a Group 272
Get List of Users in a Group 273
Remove Users from a Group 275
Get a List of Permission Profiles 276
Get Recipient Names 277
Get List of Envelopes in Folders 278
Get Account Settings 281
Modify Account Settings 282
Get Shared Access Information 283
Set Shared Access Information 289
List Signature Providers 296
Create a Signing Group 300
Get a List of Signing Groups 303
Update Signing Group Names 304
Delete Signing Groups 306
Get a Signing Group 308
Update a Signing Group 309
Get Signing Group Members 311
Add Signing Group Members 312

Delete Signing Group Members 314
Create a Custom Tab 316
Get List of all Account Tabs 320
Get Custom Tab Information 322
Modify Custom Tab Information 323
Delete a Custom Tab 328
Get List of Templates 329
Post Template 332
Get Template 344
Modify a Template 345
POST Template Edit View 350
Get Custom Document Fields for a Template Document 352
Add Custom Document Fields to a Template Document 353
Modify Custom Document Fields for a Template Document 355
Delete Custom Document Fields from a Template Document 358
Share a Template with a Group 360
Remove Template Sharing for a Group 361
Get Template Recipient Information 362
Add or Replace a Template Bulk Recipient File 364
Retrieve Template Bulk Recipient File 367
Delete Template Bulk Recipient File 368
Get Tab Information for a Template Recipient 369
Get List of Unsupported File Types 370
Add User to Account 371
Get User List 377
Close a User 380
Get User Information 381
Get Cloud Storage Provider Configurations for a User 383
Add Cloud Storage Provider Return Information for a User 385
Delete Cloud Storage Provider Authentications for a User 386
Get One Cloud Storage Provider Configuration for a User 387
Delete One Cloud Storage Provider Authentication for a User 389
Get User Items in a Cloud Storage Provider 390
Get User Items in one Cloud Storage Provider Folder 393
Get Custom User Settings 395

Add or Modify Custom User Settings 397
Delete Custom User Settings 398
Get User Profile 400
Modify User Profile 402
Get User Profile Image 405
Modify User Profile Image 406
Delete User Profile Image 406
Get User Account Settings 407
Modify User Account Settings 408
Get a List of User Signature Definitions 411
Setting User Signature and Initials Images when Creating a Signature 412
Get User Signature Information 414
Modify a User Signature 416
Close a User Signature 418
Get a User Initials Image 418
Set a User Initials Image 419
Delete a User Initials Image 420
Get a User Signature Image 422
Set a User Signature Image 422
Delete a User Signature Image 424
Add a User Social Account 425
Get User Social Accounts 425
Remove a User Social Account 426
Post Authentication View 427
Get Account Provisioning Information 428
Get List of Billing Plans 429
Get Billing Plan Details 430
Enable or Disable API Request Logging 434
Get API Request Logging Settings 435
Get API Request Logging Log Files 436
Get One API Request Logging Log File 437
Clear API Request Logging Logs 437
Revoke an Authorization Token 438
Create an Authorization Token 439
Recipient Types and Parameters 440

Tab Types and Parameters 515
Error Code Information 623
General Error Response Handling 623
Error Response Handling for API calls that Support Multiple Items 624
Error Codes and Associated Messages 624
Appendix 1: DocuSign SOAP API to REST API 646
Appendix 2: Time Zones and Date/Time Format Information 649
Account Settings 649
Time Zone and Date/Time Format Appearance 649
Date/Time Formats for API Calls 650

Introduction

The DocuSign REST API provides you with a powerful, convenient, and simple Web services API for interacting with DocuSign. The REST API uses the same underlying data model and standard objects as those in the DocuSign SOAP-based API. The REST API also has the same limits imposed on it as the SOAP-based API. See the Getting Started section for more information on prerequisites and limits.

If you already use the DocuSign SOAP API, you might be able to leverage some of your existing code and experiences for use with the REST API. See Appendix 1: DocuSign SOAP API to REST API for the relationship between the SOAP and REST information.

This section provides some background information for working with the REST API.

Commonly Used Terms

Definitions of some commonly used DocuSign terms are given below to familiarize you with their use in the DocuSign system.

- **Envelope** - This represents a container used to send documents to recipients. The envelope carries information about the sender and timestamps to indicate the progress of the delivery procedure. It can contain collections of Documents, Tabs and Recipients.
- **Document** - A document that is to be delivered, representing the content to be reviewed and/or signed by a Recipient. Documents have names and are always base64 encoded while in the system.
- **Tab** - This represents a DocuSign Tab (also known as a Field, Tag, or Stick-eTab®) on a document. It is used in several ways.

  First, it is used to indicate to a recipient where a signature or initials are required.

  Second, it is used to include various bits of information in a document in a manner similar to Form Fields or Macros (Example: a tab may automatically fill in the Company Name of a recipient when the document is signed).

  Third, it is used as editable information fields where signers can add data to a document.

- **Recipient** - Someone who receives the envelope and, depending on the settings, can sign (or initial) the documents or can add information where indicated by tabs.
- **Template** – A pre-set envelope with specific documents, set recipient roles, recipient routing order, recipient authentication, signing tabs and information fields. Templates can also contain set signing instructions for the document and signature attachment requests. Templates can be set to allow the sender to make some changes before sending the envelope.

Basic Process Flow and Steps for Sending with the REST API

The general use for the DocuSign service is sending documents for signature and the basic steps are described below:

1. A sender supplies the name and email address of each person they want to sign the document (these are the recipients), adds documents to an envelope and adds tabs to indicate where each person should sign, initial or add other information.

   Note: The sender does not have to add any tabs to a document. Sending without tabs is referred to as using free-form signing. Instead of seeing tabs on a document, the recipient will see the document with a note letting them know they must drag a signature or initial tab onto the document.

2. The sender places the envelope in the DocuSign system and DocuSign notifies each recipient, using the supplied email addresses, that they have been asked to sign a document, and provides a link to the envelope.
3. When a recipient clicks the link, they will see the documents with DocuSign Tabs in the locations specified by the sender, representing where they need to sign, initial or add information to the document.
4. When a recipient has clicked on all of their respective DocuSign Tabs (Stick-eTabs) or added their signature using free-form signing, the document is signed and the envelope is sent to other recipients for their signatures. When all the recipients have signed the documents, the envelope is completed.

Basic Process Flow for Requesting Status Changes with the REST API

You should periodically request the status changes of the envelopes from DocuSign to follow the envelope through the creation, sending and signing process. The basic process flow for requesting status changes is outlined below.

1. Request the envelope status: Use the GET method to request envelope status starting from a specified date and time. Optionally, you can just check for any changes to envelope status or for envelopes that changed to a particular status (this is highly recommended).

   The best practice for recurring status requests is to set the date-time of the request so that each request overlaps the previous request by a short interval, one or two minutes. This limits the size of the response and prevents you from getting information for envelopes that have not changed status. The example that follows shows an instance of this process.

2. Save the date/time of your request: Saving the date/time of your request allows you to format the date-time for recurring requests.

3. DocuSign responds: DocuSign returns information for the envelopes that meet the specified date-time request. The response includes the uri's for the envelope certificate, custom fields, documents, envelope along with the envelope ID, envelope status, and the date-time stamp for the envelope status change for the envelopes. If a query string was added to the original request, only the envelopes that meet those parameters are included in the response.

4. Process the envelope status changes in your system: Use the information in the request response to updated envelope information in your system. This might include triggering requests to retrieve certificates of completion and signed documents.

5. Wait and then repeat the process: The amount of time you wait between requests will depend your business processes, such as how many envelopes your organization sends during a day and how often updates are needed. The best practice for recurring status requests is to request the status no more than once every 15 minutes and only requesting status for those envelope that changed status.

Important: DocuSign imposes API call limit restrictions to prevent flooding of API calls, so your request intervals should be appropriately timed.

Request example: Based on how many envelopes you send, you decide to check for envelope status changes every 15 minutes with a 1 minute overlap to ensure that no changes are missed from the previous request.

Your first request at 12:15 would be:

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
?from_date=5-2-2012+12:15:00+AM&from_to_status=changed
```

Request 2 at 12:30 would be:

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
?from_date=5-2-2012+12:14:00+AM&from_to_status=changed
```

And request 3 at 12:45 would follow a similar pattern:

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
?from_date=5-2-2012+12:29:00+AM&from_to_status=changed
```

Request and Response Guidelines

- Key names in requests and responses use camelcase, starting with a lowercase letter. Examples: “userName”, “envelopeId”.
- Values that represent constants, enums, etc. are lowercase. This allows the client to use the values without worrying about case conversion. The only exceptions to this are error code responses.
  Examples: “status”: “created”.
- Dates are formatted using ISO 8601 format. The REST API assumes that all values passed represent UTC date/times.

  When providing a date/time format for the DocuSign REST API, the preferred formats are:
  - "yyyy-MM-dd | HH:mm"
  - "MMMM d, yyyy | HH:mm"
  - "MMM-dd-yyyy | HH:mm"
  - "dd-MM-yyyy | HH:mm"
- Response codes not in the 200 range include an error details response when possible. The response will contain the following:
  - An “errorCode” field, with an uppercase string constant describing the error.
  - A ”message” field, that describes the error in more detail, if information is available.
- Resource values that are relative to the Base URI will contain “uri” in the key name.
- Resources values that are fully qualified will contain “url” in the key name.
- Base64 content type encoding is supported for documents & images in the following manner:
  - For PUT or POST: Set the header “Content-Transfer-Encoding” to “base64” to indicate that the document bytes and/or image bytes are encoded as base64 and the DocuSign server should decode the stream.
  - For GET: Set the header “Content-Transfer-Encoding” to “base64” to indicate that the document bytes and/or image bytes requested are to be encoded as base64 by DocuSign before sending them in the response. This header will be set in the response if the conversion was performed.

Request and Response Formats

The DocuSign REST API resources can use JSON or XML for the request and response formats. The format used can be set in the file header when using a resource. You can also force the format used for a request/response by adding the format type at the end of a resource call.

Example: To use json for your request/response format when sending an envelope you would append .json at the end of the resource.

https://preview.docusign.net/restapi/v2/accounts/{accountId}/envelopes.json

Envelope and Recipient Status Codes

Knowing the status of envelopes and recipients at any particular time is important for your workflow. When requesting status for envelopes and recipients, DocuSign returns a status code. The possible values and descriptions are provided in this section.

Envelope Status Code Descriptions

The table below provides descriptions of the Envelope Status Codes.

Code	Description
created	The envelope is in a draft state and has not been sent out for signing.
deleted	This is a legacy status and is no longer used.
sent	An email notification with a link to the envelope has been sent to at least one recipient. The envelope remains in this state until all recipients have viewed it at a minimum.
delivered	All recipients have viewed the document(s) in an envelope through the DocuSign signing web site. This is not an email delivery of the documents in an envelope.
signed	The envelope has been signed by all the recipients. This is a temporary state during processing, after which the envelope is automatically moved to Completed status.
completed	The envelope has been completed by all the recipients.
declined	The envelope has been declined for signing by one of the recipients.
voided	The envelope has been voided by the sender.
timedout	This is a legacy status and is no longer used.
authoritativecopy	The envelope is in an Authoritative state. Only “Copy” views of the documents will be shown.
transfercompleted	The envelope has been transferred out of DocuSign to another authority.
template	The envelope is a Template.
correct	The envelope has been opened by the sender for correction. The signing process is stopped for envelopes with this status.

Recipient Status Code Descriptions

The table below provides descriptions of the Recipient Status Codes.

Code	Description
created	The recipient is in a draft state. This is only associated with draft envelopes (envelopes with a Created status).
sent	The recipient has been sent an email notification that it is their turn to sign an envelope.

Code	Description
delivered	The recipient has viewed the document(s) in an envelope through the DocuSign signing web site. This is not an email delivery of the documents in an envelope.
signed	The recipient has completed (signed) all required tags in an envelope. This is a temporary state during processing, after which the recipient is automatically moved to Completed.
declined	The recipient declined to sign the document(s) in the envelope.
completed	The recipient has completed their actions (signing or other required actions if not a signer) for an envelope.
faxpending	The recipient has finished signing and the system is waiting a fax attachment by the recipient before completing their signing step.
autoresponded	The recipient's email system auto-responded (bounced-back) to the email from DocuSign. This status is used in the web console to inform senders about the bounced-back email. This is only used if "Send-on-behalf-of" is turned off for the account.

Getting Started

To get started you should get a DocuSign Integrator Key, review the Basic Scenarios, and review the REST API references.

Integrator Keys

An Integrator Key is a Unique Identifier for each DocuSign integration. It is used (and required) for all API calls (SOAP or REST) to any DocuSign service. Having an Integrator Key lets DocuSign "tag" each API call from all integrations, and helps provide both an additional layer of security and helps DocuSign support our partners. An Integrator Key is REQUIRED for all integrations, and if you want to move to Production (make calls to www.docusign.net) you also must be certified and get that Integrator Key authorized by DocuSign for Production.

All Integrator Keys are used for development first, and as a result, they are all managed (and requested) in DocuSign's DEMO service. If you already have a developer account on demo, log in to request an Integrator Key. If you do not have a Demo Developer Account, go to the DocuSign Developer Center and request a free account.

After you have developed your integration using your Demo Integrator Key, you must get that Integrator Key certified before moving to production. In order to become certified, you must be a member of our Developer Program. With that membership, you get access to all sorts on ongoing support, certification help, and access to production. To join the program, or start the certification process, see the Quick Start Overview and Go Live Overview on the DocuSign Developer Center for more information about the developer program.

API Call Rate Limits

To maintain reliability and stability within our demo and production environments, DocuSign operates with certain API call efficiency guidelines. To ensure effective load balance we continually monitor the API calls to our backend systems and we will contact developers that are putting unnecessary burden on the system.

DocuSign has implemented the following API Call Rate Limits to balance loads on the system:

- The demo environment (demo.docusign.net) is limited to a call rate of 1,000 API calls per hour per account.
- The production environment (www.docusign.net) is limited to a call rate of 1,000 API calls per hour per account.

If the API call rate limit is reached, you will receive an exception for each call until the start of the next hour (this can be up to 60 minutes). The exception message states, “The maximum number of hourly API invocations has been exceeded” (error number 207).

Hourly API usage is tracked from the start of one-hour to start of the next hour.

Customers can monitor the number of API calls their application uses by checking the API Limit Headers that are returned in the response to REST API requests. The headers are returned in both the production and demo environments.

The headers are only returned in the response to REST API requests for calls that are subject to DocuSign API limits. So the API limit headers will not be returned in all the API responses and your code should take this into consideration.

Instances where the API limit headers are not returned include:

- When an error is detected early in an API request. Usually this due to errors in the request body or invalid query string parameters.
- API calls that historically have not been counted as part of the DocuSign legacy API limits strategy.
- Any API calls that do not require user authentication and do not identify a specific account user.

For instances where the API limit headers are not returned, there is no change to the remaining calls available for the account.

The information in the API limit header shows the current API limit for the account (the number of calls that can be made per hour), the number of remaining calls for the current time period, and when the account API limit will reset. The reset time is shown in Unix epoch time.

Example API Limit Headers

```
X-RateLimit-Limit → 1000
X-RateLimit-Remaining → 996
X-RateLimit-Reset → 1425967200
```

For polling compliance DocuSign recommends that you do not exceed 1 status request per unique envelope (GET /accounts/{accountId}/envelopes) per 15 minutes.

There are a number of ways to minimize API impact, such as:

- using bulk operations for requesting status,
- utilizing DocuSign’s event notification feature,
- and refraining from repeatedly requesting information on envelopes that are in terminal state (Completed, Declined or Voided).

If you find your application still requires more than 1,000 calls per hour per account, please contact service@docusign.com for assistance in working on a solution.

If you have any questions, please check our DocuSign Developer Forum on Stack Overflow.

File Size Limitations

DocuSign has the following limitations on files used in envelopes and as attachments:

- DocuSign recommends that you do not add files larger than 25MB to an envelope. Note that, depending on the recipient's internet connection, large files might affect signing performance.
- DocuSign has not imposed a limit on the number of files that can be added to an envelope. However, as with file size, envelopes with a large number of files might affect signing performance.
- For signer-uploaded attachment files, DocuSign supports files sizes up to 25MB for an envelope.
- There is a file size limit of 5MB for attaching completed documents to emails sent by DocuSign to recipients when an envelope is completed. If the completed documents are greater than 5MB, the email still provides a link to the documents on the DocuSign system.

DocuSign Time Track Header

The DocuSign Time Track header is an optional line that can be added to your API request header to return processing times for your requests.

When the X-DocuSignTimeTrack header is included in the API request, DocuSign returns the header in the response with the start-time and the end-time of the associated API request.

For example, when the Time Track header is included in the request:

```
X-DocuSignTimeTrack
```

The response includes the header and times in the response:

```
X-DocuSignTimeTrack: REST0_Start,2016-01-05T22:30:52.2522730Z;REST0_End,2016-01-
05T22:30:52.3182730Z
```

Additionally, you must add your own value on the request and that information is preserved in the response.

For example, you could add the time your application starts a process (such as MySendApp,<start time>) and then check the round-trip network time for an action.

Request:

```
X-DocuSignTimeTrack: MySendApp,2016-01-05T22:30:52.2522666Z
```

Response:

```
X-DocuSignTimeTrack: MySendApp,2016-01-05T22:30:52.2522666Z;REST0_Start,2016-01-
05T22:30:52.2522730Z;REST0_End,2016-01-05T22:30:52.3182730Z
```

REST API Versions

You can determine the REST API versions that are available by checking the following links:

Note: You do not need an integrator key to view the REST API versions and resources.

- DocuSign Production system: https://www.docusign.net/restapi/service_information
- DocuSign Demo system: https://demo.docusign.net/restapi/service_information

To view the base resources available for a version, remove `/service_information` from the above URLs and add the version number for the version you want to view.

> ***Example:*** https://demo.docusign.net/restapi/v2 lists all of the base resources available in version 2 of the REST API on the DocuSign Demo system.

To view descriptions and samples of the service operations for all versions, remove the version number and add /help to the URL.

> ***Example:*** https://demo.docusign.net/restapi/help lists the REST API operations on the DocuSign Demo system with XML and json request and response samples for the active REST API versions.

REST API Base URL

The base URL used for all operations is formatted as follows:

```
https://{server}/restapi/{apiVersion}/
```

The base URL for your account is retrieved by using the `/login_information` uri and is used in future API calls as the base of the request URL.

REST API Explorer

DocuSign has established a website (http://iodocs.docusign.com/) where you can walkthrough and try out the REST API methods. This website is useful when learning the different aspects of the REST API.

Send On Behalf Of Functionality in the DocuSign REST API

The DocuSign *Send On Behalf Of* (SOBO) functionality allows a single user in an account to authenticate for other members of the account. When originally conceived, it was meant for sending fully-defined envelopes only. Since then, most other API functions have been allowed - making this really an "authenticate on behalf of" function instead of just a sending function.

This authentication is especially useful for CRM and other integrations where the API client also maintains an account-with-multiple-members structure similar to DocuSign. In this scenario, having a single user authenticate with DocuSign relieves the API integration from knowing and storing individual user passwords within the client application.

The SOBO user, also known as the authenticating user, authenticates with DocuSign on behalf of an operating user. Once authentication is passed, the operating user's user and member settings are used for all authorization functions.

The SOBO user must have the following userSettings enabled:

- apiAccountWideAccess
- allowSendOnBehalfOf

Note: If user permissions are being set through the Classic DocuSign Experience web application, the listed settings correspond to the Account-Wide Rights and Send On Behalf Of Rights (API) settings.

The DocuSign REST API Send On Behalf Of function uses the <SendOnBehalfOf> node in the X-DocuSign-Authentication header to add a Send On Behalf Of identifier for the operating user.

The Send On Behalf Of identifier can be a properly formatted email address, with an optional semicolon delimited user name, or, if using the Single Sign On (SSO) environment, a User ID (UID) that can be looked up in the DocuSign SSO configuration.

- If the identifier is a properly formatted email address, the system conducts a look-up of the email address and user name (if provided) to see if that user is a member of the account. If membership in the account is not found, an error is returned. The account used for the check is the one associated with the authenticating user's credentials.
- If the identifier is not an email address (the identifier fails the regular expression test for an email), it is assumed that the identifier is a UID. A search is conducted for the UID in the DocuSign system for a userId and the Single Sign On (SSO) customer system. If the UID is found the email address and user name associated with the UID are retrieved. Then the system conducts a look-up of the email address and user name to see if that user is a member of the account. If membership in the account is not found, an error is returned. The account used for the check is the one associated with the authenticating user's credentials.

Functions Not Supported by Send On Behalf Of

The following DocuSign REST API calls do not support SOBO:

- Login (GET /login_information)
- Create an Account (POST /accounts)
- Get Account Provisioning Information (GET /accounts/provisioning)
- Get a list of Billing Plans (GET /billing_plans)
- Get Account Billing Plan (GET /accounts/{accountId}/billing_plan)
- Change User Password or Email (PUT /login_information/password)
- GetRecipientSignature (GET /accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/signature)

All other REST API functions are supported by SOBO authentication.

Send On Behalf Of REST Examples

The Send On Behalf Of identifier is inserted in the <SendOnBehalfOf> node in the "X-DocuSign-Authentication" header. The examples below show Send On Behalf Of authentication using an email address, email address:user name combination and UID. The Send On Behalf Of information is highlighted in the examples.

Example using Email Address

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><SendOnBehalfOf>bob.smith@gmail.com</SendOnBehalfOf><Username>{name}
</Username><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey>
</DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Example using Email Address and User Name

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><SendOnBehalfOf>bob.smith@gmail.com;Bob
Smith</SendOnBehalfOf><Username>{name}</Username><Password>{password}</Password><Integrat
orKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Example using UID

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><SendOnBehalfOf>P134325</SendOnBehalfOf><Username>{name}</Username><
Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCr
edentials>
Accept: application/json
Content-Type: application/json
```

DocuSign OAuth (Authorization Code Grant or Implicit Grant)

DocuSign's OAuth supports the standard grant types for web and mobile applications. This allows third-party integrations to authenticate a user without prompting the user for their password. In addition, this model of authentication allows DocuSign to introduce new authentication features without having to update any third-party integrations.

This new model of authentication will become the standard method of authentication and will eventually supersede the other protocols that DocuSign has in place.

Configuring an App

In order to leverage the DocuSign OAuth protocol, your application must be registered with DocuSign.

Client IDs and Client Secrets

In order to communicate with the DocuSign OAuth service, you will need an integrator key, which will serve as your client id, and, depending on your grant type, a client secret.

To Add an Integrator Key

Note: Because external applications must be certified for use with the DocuSign Production environment, your application must be added to the DocuSign Demo environment before it can be added to Production.

1. Log on to DocuSign Admin

   This can be done from your DocuSign account by selecting the **Go to Admin** link in the account settings drop-down menu.

   Alternately you can directly access DocuSign Admin by going to the administrator log on page and entering your credentials: https://admin.docusign.com.

Note: if you are trying to access the Demo version of DocuSign Admin, use the url: https://admindemo.docusign.com.

2. In DocuSign Admin, click **API and Keys** in the left navigation panel.
3. Click **ADD INTEGRATOR KEY**.

4. Add a description for your app.
5. Add the links to the privacy policy and terms of use for your app.
6. If your app is a client application (e.g. mobile), select **This is a mobile app**.
7. Click **ADD** to save the integrator key information.

   After adding the key, you will need to edit the key information to add redirects and, if needed, client secrets.

Redirect URIs

Redirect URIs must be configured with your client application registration. When provided in the constructed URI, they must be an exact match with the value in your configuration. An app can have multiple redirect URIs configured.

Client Secrets

If your integration is using the authorization code grant type, you need to provision secret keys with DocuSign. An integrator key can have multiple secrets at a time, which allows developers to provision a new secret key without service interruptions.

Secrets should be securely stored within your application. They should never be shared or disclosed publicly. For that reason, the authorization grant type is not suitable for applications which are installed on a user's device (e.g. mobile app or desktop client).

When adding a secret key, be sure to copy the secret to a secure location immediately after you create it. Secret keys are only displayed in plain text when they are first created. After that, for security purposes, DocuSign only shows the last 4 digits of the key.

To Edit an Integrator Key

1. In DocuSign Admin, click **API and Keys** in the left navigation panel.
2. Find the Integrator Key you want to edit and click on the Integrator Key or in the Actions drop down select **Edit**.

   In the Edit dialog you can change the App Description and links, add new redirect URIs and secret keys, and delete redirect URIs and secret keys.

3. To add a new redirect URI, click **ADD URI** and type the URI.

   Repeat this step to add other URIs.
4. To delete a redirect URI, find the URI in the list and click the **X** adjacent to the URI.
5. To add a new Secret Key, click **ADD SECRET KEY** to generate the new secret.

IMPORTANT: Copy the secret key to a secure location immediately after you create it. Secret keys are only displayed in plain text when they are first created. After that, for security purposes, DocuSign only shows the last 4 digits of the key.

6. To delete a secret key, find the key in the list and click the **X** adjacent to the key.
7. Click **SAVE** to save your changes.

Supported Grant Types

The DocuSign OAuth implementation supports the following grant types:

- **Authorization code –** for applications running on a web server
- **Implicit** – for applications running on a desktop or for mobile applications

Making Requests

DocuSign OAuth requests are directed at a particular DocuSign environment (for the Hostname) and Endpoint. The Hostname and Endpoint information is given below and examples of how to use them are in the sections that follow.

DocuSign Hostnames & Endpoints

The Hostnames for the different DocuSign environments are:

Environment	Hostname
Demo	https://account-d.docusign.com
Production	https://account.docusign.com

The Endpoints supported as part of OAuth 2.0 are:

Authentication	/oauth/auth
Token	/oauth/token
User Info	/oauth/userinfo

Authentication Request

The following format can be used to create URLs for authenticating users with the OAuth protocol.

https://{hostname}/oauth/auth

This is considered the start URL of the OAuth protocol. To initiate authentication, the integration should redirect the user to this particular endpoint while specifying the appropriate parameters. Implementers must append the appropriate parameters to their request to properly configure type of authentication.

Authentication Parameters

The following parameters are used to construct the authentication URL.

Name	Required	Value	Description
response_type	Yes	code or token	Determines the response type of the authorization request.

Name	Required	Value	Description
client_id	Yes	The integrator key obtained during application registration	This identifies the client making the request.
redirect_uri	Yes	Pre-registered redirect URIs for this integrator key	The redirect URI must exactly match one of those pre-registered for the integrator key. This determines where to redirect the user after they successfully authenticate.
scope	Yes	all	'all' is the only current supported value
state	No	Any string	Allows for arbitrary state that may be useful to your application. The value in this parameter it is returned with the redirect response
ui_locales	No	Language codes (example: fr-CA fr en)	List of the end-user's preferred languages, represented as a space-separated list of RFC5646 language tag values ordered by preference.

Example Authentication Request

```
GET /oauth/auth?
    response_type=code
    &scope=all
    &client_id=abcd-1231-4asd323
    &state=a39fh23hnf23
    &redirect_uri=https%3A%2F%2Fclient.example.com%2F/callback
Host: account.docusign.com
```

Example Authentication Response

When the authentication completes, the server returns the user to the specified redirect URI with a success or failure. On success, the response has all the appropriate data for the integration to complete the remaining steps to access the API.

The following is an example of the response if the integration is using the authorization code flow.

```
HTTP/1.1 302 Found
Location: https://client.example.com/callback?
    code=ey2dj3nd.AAAA39djasd3.dkn4449d21d
    &state=a39fh23hnf23
```

Error Responses

If authentication did not complete successfully, the user might be redirected to the application with an error. The error response is formatted as follows:

```
HTTP/1.1 302 Found
```

```
Location: https://client.example.com/callback?
    error=invalid_request
    &error_description=Unsupported%20request.
    &state=a39fh23hnf23
```

Error Response Parameters

Name	Required	Description
error	Yes	A well-known error string indicating the cause of the failure.
error_description	No	A human readable string which can be presented directly to the end-user.
state	Only if supplied in request	The OAuth 2.0 state value provided in the request.

Initiating the Authorization Code Flow

The authorization code flow is used when the application has a server component capable of protecting the client secret. In this scenario, the integration will specify code as the response_type.

The server redirects the browser to the OAuth endpoint, which triggers the following request:

```
GET /oauth/auth?
    response_type=code
    &scope=all
    &client_id=abcd-1231-4asd323
    &state=a39fh23hnf23
    &redirect_uri=https%3A%2F%2Fclient.example.com%2F/callback
Host: account.docusign.com
```

Authorization Code Response

Upon successful authentication and authorization, the server redirects the user to the specified redirect URI. On this callback, the request for your callback will contain a code and the original application state. The authorization code can then be exchanged for an access token and refresh token.

```
HTTP/1.1 302 Found
Location: https://client.example.com/callback?
    code=ey2dj3nd.AAAA39djasd3.dkn4449d21d
    &state=a39fh23hnf23
```

Authentication Response Parameters

Name	Required	Description
code	Yes	The OAuth 2.0 authorization code. It is used to request an access token and refresh token.
state	No	The value of the state passed into the original request. If a value is returned, clients must validate it is exactly equal to the value provided in the request.

Token Request

Once an application has an authorization code, it can use the code to make a token request. The response contains the access token and refresh token. The access token is the authorization token used to access resources in the API.

To make a token request, the application would make a form post to the token endpoint with the following parameters.

```
POST /oauth/token
Host: account.docusign.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic <base64 encoded string: {client_id} + ":" +
{client_secret}>

grant_type=authorization_code&code=ey2dj3nd.AAAA39djasd3.dkn4449d21d
```

The application must include their app credentials in the authorization header. This can be achieved by concatenating the client_id with a ":" and the client_secret. The resulting string should be base64 encoded.

Token Request Parameters

Name	Required	Description
grant_type	Yes	When exchanging a code with an access token this value will be "authorization_code"
code	Yes	The authorization code that was received in the authorization code response.

Token Response

If the grant_type and grant are accepted, the token endpoint returns a response similar to the one in the following example. The server issues a new access token and a new refresh token. It's expected that the application will update the refresh token whenever it is reissued so that it retains the most recent refresh token.


```
{
    "access_token":
    "eyJ0eXAiOiJNVCIsImFsZyI6IlJTMjU2Iiwia2lkIjoiZTNlYTVkMmEtMjBlMC00MTkxLWIxNWM
    tZTNhYjJkYTIxZTViIn0.AQUAAAABAAUABwCy-FavV2jSSAgAsnidy91o0kgGAPXvg-
    fhU7JFtZk4EOicKqwCALGa2FDV2gpKtBCS7jEQuXA.alpiE3XAgLTunEmpBsfbk6Lv8CIgJGm3D7
    7EWOCYevGG1bgsXJZuX0pDjRPyMQXqfSI1vzutPJCp9Ckpf9mNUbKYbxavEhioQx2AaZh6uEMwHs
    Yij-FUSpCBhD7SGByjB2giTEGEJDo1vj-YxgSq3OBMb-
    zMBpdgLUqN8VbmEQCLpeUKbgtU4jri1JkCwUQeKN7igEIVK-
    75URPySbFcs3h9jUAmPEKi3NaQiciUFcMXYhXAldDkML8eoRKAGrebfg_Fs686wSfX_ZDz1SyKdr
    z130nB9DZX3NpUlbhmYaPSm7qwT3qgRQJDywMRWQxTr-72xghXHwVzzMuN9-yGhA",
    "token_type": "Bearer",
```

```
    "refresh_token":
    "eyJ0eXAiOiJNVCIsImFsZyI6IlJTMjU2Iiwia2lkIjoiZTNlYTVkMmEtMjBlMC00MTkxLWIxNWM
    tZTNhYjJkYTIxZTViIn0.AQUAAAABAAUABwCy-FavV2jSSAgAsnidy91o0kgGAPXvg-
    fhU7JFtZk4EOicKqwCALGa2FDV2gpKtBCS7jEQuXA.alpiE3XAgLTunEmpBsfbk6Lv8CIgJGm3D7
    7EWOCYevGG1bgsXJZuX0pDjRPyMQXqfSI1vzutPJCp9Ckpf9mNUbKYbxavEhioQx2AaZh6uEMwHs
    Yij-FUSpCBhD7SGByjB2giTEGEJDo1vj-YxgSq3OBMb-
    zMBpdgLUqN8VbmEQCLpeUKbgtU4jri1JkCwUQeKN7igEIVK-
    75URPySbFcs3h9jUAmPEKi3NaQiciUFcMXYhXAldDkML8eoRKAGrebfg_Fs686wSfX_ZDz1SyKdr
    z130nB9DZX3NpUlbhmYaPSm7qwT3qgRQJDywMRWQxTr-72xghXHwVzzMuN9-yGhA",
    "expires_in": 57592
}
```

Using a Refresh Token

Once the access code has been redeemed, the response might contain a refresh token. Refresh tokens have a longer lifetime than access tokens and can be used to obtain a new refresh token.

The following example is a non-normative request to exchange a refresh token for a new set of access and refresh tokens.

```
POST /oauth/token
Host: account.docusign.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic <base64 encoded string: {client_id} + ":" +
{client_secret}>

grant_type=refresh_token&refresh_token=ey4fdd3nd.AAAA3d2ddagq3.akd1243d31d
```

Initiating the Implicit Flow (Mobile Apps or Client Applications)

The implicit flow should be only used by clients that do not have the ability to protect the client secret in a remote server. Typically this happens with a mobile app or a client application.

IMPORTANT: The Integrator Key for apps using implicit flow must have the **This is a mobile app** option selected.

Implicit Flow Request

The following is a sample request to be made by a user-agent. Typically the client will open a browser at the following location. In this scenario, the integration will specify token in the response_type.

```
GET /oauth/auth?
    response_type=token
    &client_id=abcd-1231-4asd323
    &redirect_uri=https%3A%2F%2Fclient.example.com%2F/callback
    &scope=all
    &state=a39fh23hnf23
Host: account.docusign.com
```

Clients implementing the flow should provide users an opportunity to cancel out of the flow, in case the request fails somewhere and it is not possible to redirect the user to the redirect_uri.

Implicit Flow Response

Upon successful authorization, the server redirects the user to the specified redirect_uri. When the request is successful, the response will appear as follows:

```
HTTP/1.1 302 Found
```

```
Location: https://client.example.com/callback#
    access_token=eyA3J3ad.k32jdskd.ann4ds
    &token_type=bearer
    &expires_in=3600
    &state=a39fh23hnf23
```

Implicit Flow Response Parameters

Name	Required	Description
access_token	Yes	The OAuth 2.0 access token.
token_type	Yes	The OAuth 2.0 token type value. This value will be Bearer for most clients.
expires_in	No	Expiration time, in seconds, for the access token.
state	No	The value of the state passed into the original request. If a value is returned, clients must validate it is exactly equal to the value provided in the request.

Access Token Usage

Once your application has an access token it can be used to access resources in the DocuSign API. It will be placed in the Authentication header of any REST API request.

The format in the header is simply:

```
Authorization: Bearer <access token>
```

User Information

The User Info Endpoint is used to request information about the authentication user.

The userinfo API returns their user id, list of accounts in which they have an active membership, and other profile information.

Example UserInfo Request

```
GET /oauth/userinfo HTTP/1.1
Host: account.docusign.com
Authorization: Bearer eyA3J3ad.k32jdskd.ann4ds
```

Example UserInfo Response

```
{
  "sub": "1470ff66-f92e-4e8e-ab81-8c46f140da37",
  "accounts": [
    {
      "account_id": "230546a7-9c55-40ad-8fbf-af205d5494ad",
      "is_default": true,
      "account_name": "jillSmithDev",
      "base uri": "https://demo.docusign.com"
    }
  ],
  "name": "Jill Smith",
  "given_name": "Jill",
  "family_name": "Smith",
```

```
  "email": "jill.smith@example.com"
}
```

IMPORTANT: If an integration is using OAuth with the authorization code or implicit grant type, the userinfo API replaces login_information. If an integration is using this method of OAuth, login_information should never be used.

Constructing a new base URL for the REST API

Integrations can easily use OAuth against any of our APIs. To initiate a REST API call the implementer would construct a base URL with the following fields in the userinfo response.

```
{base_uri} + "/restapi/v2/accounts" + {account_id}
```

DocuSign OAuth (Resource Owner Credentials or SAML)

OAuth2 is an open standard for authorization that can be used through the DocuSign REST API to obtain access tokens. The Resource Owner Password Grant is the only OAuth2 flow supported in the DocuSign REST API. This allows a client application to obtain an access token directly from the REST API by presenting the user name, password and integrator key for the user. After an access token is obtained, it is then used in API calls instead of the user name/password/integrator key combination.

The descriptions in this section assume that a client application is being used for the actions.

General Usage Notes

The following general notes apply to when using the OAuth2 Authentication.

- The OAuth2 client_id is the same as the DocuSign Integrator Key. Integrator Keys can be managed by client application developers in the DocuSign Demo Environment through the DocuSign web console on the Preferences – API page.
- The access_token does not expire, even when the user changes their password. The tokens should be stored and re-used for other calls. As a result, refresh_token is not supported.
- A maximum of 10 access_tokens are supported per userId. This allows multiple instances of a client application to connect to a user. Users can revoke access_tokens through the REST API or through the DocuSign web console as described in the Revoking Tokens section.
- A client application can use the access_token for a user to act as another user in the account if the access_token is associated with a user who has account-wide access and Send on behalf of privileges. This provides functionality that is identical to the "Send On Behalf Of" feature. The X-DocuSign-Act-As-User header is used in conjunction with the Authentication Header for this feature. See the Acting as Other Account Users section for more information.
- A client application can use the access_token for a user who has account-wide access and send on behalf of rights to create an access_token for other users of the account. The access_token created can then be stored in a user record on the client and used by that user to authenticate. The user identified in the access_token can see that a token was issued in the DocuSign web console on the Preferences – Connected Apps page. That user also has rights to revoke the token from the web console.

OAuth2 Token Request

The client application uses the following steps to obtain an access token that can be stored locally with the client. Once authentication is complete, the access token is used instead of the username/password/integrator key combination for all API calls.

The client application normally wants to obtain, and store locally, other information used in API calls including:

- The baseUrl of the DocuSign site where the user's accounts are stored.
- The accountId of the account that the user is using.

The client application should take the following actions to obtain an access token and other information:

1. Prompt the user for email/password.

The client shows the UI for this prompt and is responsible to keep the information confidential and not store it locally.

2. Call the Login_Information API.

The UserName/Password/IntegratorKey are set in the X-DocuSign-Authentication header for this call.

The client uses this API call to obtain the baseUrl, accountId, and userId for the user and stores this information on the client side.

3. Post to Token Endpoint to obtain the access_token.

The baseUrl retrieved in the login_information API is used with the token endpoint. The client user's username, password are used, along with the client application's integrator key (set as "client_id"). The username and password should be urlencoded to encode any special characters (such as '+'). The OAuth2 scope is always set to "api".

The format of the token api call is as follows:

```
POST https://{server}/restapi/{apiVersion}/oauth2/token

Accept: application/json
Content-Type: application/x-www-form-urlencoded
grant_type=password&client_id={IntegratorKey}&username={email}&password={password}&scope
=api
```

The response, if successful, is:

```
{
    "access_token":"{access token}",
    "scope":"api",
    "token_type":"bearer"
}
```

The client application then stores the access_token for future use. The access_token is valid in all future API calls to authenticate this user, until the token is revoked. It is not affected by password changes. The client application is responsible for storing and protecting this token.

Important: The length of the access token is variable. It is normal for the token to be at least 550 characters or longer. Don't make assumptions about the length.

Normal Usage

Once the access_token, baseUrl, accountId and other information has been obtained, other API calls may be made without returning to the token and login_information API's. The only time the client application should need to repeat that sequence (login_information, token) is if other API calls return an error indicating that the client was not authorized.

To use the access_token in REST API calls:

- Set the Authorization header in each call as follows:

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

Authorization:bearer{accessToken}
Accept: application/json
Content-Type: application/json
```

- Do not use the X-DocuSign-Authentication header if the Authorization header is used.

OAuth2 and SAML

Security Assertion Markup Language (SAML) is an XML-based framework for communicating user authentication, entitlement and attribute information. As an extension of the OAuth2 authentication support in the DocuSign REST API, DocuSign supports SAML assertion XML when obtaining an access token.

The OAuth 2.0 SAML bearer assertion flow is similar to a refresh token flow within OAuth. The SAML assertion is POSTed to the OAuth token endpoint, which in turn processes the assertion, and issues an access_token based upon prior approval of the application.

The general steps involved in using the OAuth 2.0 SAML bearer assertion flow are:

1. The developer obtains a DocuSign integrator key.
2. The developer registers an X509 certificate that will be used to sign SAML assertion.
3. The developer creates an application that generates a SAML assertion and signs it with their certificate.
4. The assertion is POSTed to the token endpoint https://{server}/restapi/{apiversion}/oauth2/token.
5. The token endpoint validates the signature using the certificate registered by the developer.
6. The token endpoint validates the Audience, Issuer, Subject, and validity of the assertion.
7. If the assertion is valid and the application has been authorized an Access Token is validated.

The token provided expires after 1 hour and no refresh token is issued.

SAML Assertion

The client must create a valid SAML bearer assertion that conforms to the following rules:

- The Signature must be valid against the stored X509 signing certificate.
- The SAML assertion can use a user's email address or user ID.

  When using an email address in the SAML assertion, set the subject name ID format to email address. For example:

```
<saml:Subject>
  <saml:NameIDFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">{user
emailaddress}</saml:NameID>
</saml:Subject>
```

When using a user Id in the SAML assertion, set the subject name ID format to persistent. For example:

```
<saml:Subject>
  <saml:NameIDFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">{user
id(GUID)}</saml:NameID>
</saml:Subject>
```

- SubjectConfirmation/Method must be “urn:oasis:names:tc:SAML:2.0:cm:bearer”
- SubjectConfirmationData/Recipient must match the token endpoint url.
- SubjectConfirmatinData/NotOnOrAfter must be greater than or equal to the current date and time.
- Audience must match RESTAPI URL.

When requesting the OAuth token, the assertion must use base64url encoding as defined in the Base16, Base32, and Base64 Data Encodings IETF document at the link: https://tools.ietf.org/html/rfc4648

Example SAML Assertion

```
<saml:Assertion Version="2.0" ID="_b4105a9b-5c0a-493a-beac-0759a8b5bed6"
IssueInstant="2012-12-04T19:37:23.143Z"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    <saml:Issuer>{DOCU_Integrator_Key}</saml:Issuer>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
           <SignedInfo>
                  <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
                  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
sha1"/>
                  <Reference URI="#_b4105a9b-5c0a-493a-beac-0759a8b5bed6">
                         <Transforms>
                                <Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                                <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#">
                                       <InclusiveNamespaces PrefixList="#default saml ds
xs xsi" xmlns="http://www.w3.org/2001/10/xml-exc-c14n#" />
                                </Transform>
                         </Transforms>
                         <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
/>
                         <DigestValue>tOxTTQyG2qhfwXnx4ozYFpS/D38=</DigestValue>
                  </Reference>
           </SignedInfo>

    <SignatureValue>QtGWSSUdlzEi26p09etY2PbXfKr72TdM81TK662HvlUuNYeBahintlwvHPhFrJisA17mE
5BqAxcaiBY41+nU3AzujnNzmXpFH9Svrq0kBERjFVWGZVKbm4FKlmUF8fxYgiZIVJ5MY
vaPowglaWfNhyHG2CkSUeNAUIJQSlzuf9I=</SignatureValue>
           <KeyInfo>
                  <X509Data>
                         <X509Certificate>[removed for brevity]</X509Certificate>
                  </X509Data>
           </KeyInfo>
    </Signature>
    <saml:Subject>
```

```
            <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress">{user email address}</saml:NameID>
            <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                  <saml:SubjectConfirmationData NotOnOrAfter="{SAML Assertion Expiration
Date}" Recipient="https://{server}/restapi/{apiversion}/oauth2/token" />
            </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions>
            <saml:AudienceRestriction>
                  <saml:Audience>https://www.docusign.net</saml:Audience>
            </saml:AudienceRestriction>
    </saml:Conditions>
    <saml:AuthnStatement AuthnInstant="2012-12-04T19:37:23.143Z">
            <saml:AuthnContext>
                  <saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:
X509</saml:AuthnContextClassRef>
            </saml:AuthnContext>
    </saml:AuthnStatement>
</saml:Assertion>
```

Revoking Tokens

Both the client application and the user can revoke access_tokens. A revoked access_token is no longer valid and attempts to use it will result in an error response.

A client application should provide a mechanism for revoking access. This could be something like a "Disconnect <myApp> from DocuSign" button in the client application. When selected, the client application would revoke the access_token and remove it from the client application's storage. The client application would then go through the OAuth2 Token Request steps previously described to re-establish authentication.

Revoking tokens through the REST API:

The client application POST's to the revoke endpoint with the content type set to "application/x-www-form-urlencoded" and the body of the request set as "token=<access token to revoke>"

```
POST https://{server}/restapi/{apiVersion}/oauth2/revoke

Accept: application/json
Content-Type: application/x-www-form-urlencoded
token={access token to revoke}
```

200-OK will be returned if the token was revoked.

Revoking tokens through the Web Console:

The DocuSign web console provides a way to revoke tokens for the user identified in the token. The user goes to the Preferences pages and clicks **Connected Apps** under Member Profile. This page shows the applications that are connected to this user with access tokens.

- The user can revoke access by clicking the **Revoke** button on the same line as the application.

> **Note:** An application, as defined by its integrator key, can appear up to 10 times in this list. There is currently no indication of which instance of a given client application will be revoked, other than the date that the client was attached.
>
> For example, if the client application were an iPad app and the user owned two iPads, then both might appear in the list. The user would not know which row in the table is associated with which iPad, unless the user knew the date/time the iPad application created the token.

Acting as Other Account Users

Depending on the account holder, a DocuSign account can support a number of individual users. Some client application integrations, especially those installed in other multi-user environments (such as Salesforce, Dynamics, SugarCRM) have the option to store login information for each of the client application users or to designate a single user as the "authenticating user", who will authenticate for other users in the account. This allows a client application to store less information about individual DocuSign users and also avoid requesting password information for each user of the client application.

A user, known as the authenticating user in this circumstance, that wants to authenticate for other users in the account must have the following DocuSign userSettings enabled:

- apiAccountWideAccess
- allowSendOnBehalfOf

> **Note:** If you are setting user permissions through the DocuSign web console, these correspond to the Account-Wide Rights and Send On Behalf Of Rights (API) settings.

The two methods of for acting as other users are described below:

Acting as Other Account Users

> **Note:** This provides functionality that is identical to the "Send On Behalf Of" feature.

This allows an authenticating user to act as another user in the account. The operations, usually sending an envelope or checking status, are performed as the "operating user," not as the authenticating user. The authenticating user's access_token is used for authentication only, while the

"operating user's" userId is checked for proper authorization for the API method being called. This allows the client application to avoid storing any tokens for individual users.

To act as other account users, the client application specifies the operating user by adding the "X-DocuSign-Act-As-User" header to the request. For example:

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

Authorization: bearer <access token for authenticating user>
X-DocuSign-Act-As-User: bob.smith@gmail.com
Accept: application/json
Content-Type: application/json
```

The syntax email:name is also supported in the <SendOnBehalfOf> node in the "X-DocuSign-Authentication" header.

Obtaining access_tokens for Other Account Users

The authenticating user can also obtain access_tokens for other members in the account. This model would be used if the client application wishes to store individual access tokens for each account user. The advantage of this method is that each user does not need to enter their DocuSign password during a setup sequence. Only the Administrator needs to provide login credentials. The disadvantage of this is that the client application must store and manage access_tokens for each individual account member.

The client application obtains this token by POSTing a request to the token endpoint that contains:

- An Authentication header containing the access_token for the authenticating user.
- A request body that identifies the user in the account that an access_token is being requested for. Note that the password for this user is not specified and should not be requested from the user by the client application.
- The request would be similar to:

```
POST https://{server}/restapi/{apiVersion}/oauth2/token

Authorization: bearer {access_token for authenticating user}
Accept: application/json
Content-Type: application/x-www-form-urlencoded
grant_type=password&client_id={IntegratorKey}&username=bob.smith@example.com&password
=&scope=api
```

- The response returns:

```
200 OK
Content-Length: 605
Cache-Control: private
Content-Type: application/json; charset=utf-8
Date: Mon, 29 Oct 2012 19:36:02 GMT
Server: Microsoft-IIS/7.5

{
       "access_token":"{access token for bob smith}",
       "scope":"api",
       "token_type":"bearer"
}
```

- Once obtained, this access_token should be stored with the individual user's information and protected from unauthorized use, since it will remain valid until revoked. It will be valid across user password changes.

OAuth2 Response Codes

The OAuth2 –related endpoints return 200-OK on success. On failure, they typically return 400-Bad Request, or 401-Unauthorized.

In order to be consistent with OAuth2 documentation, the OAuth2-related endpoints support a different error response format than other DocuSign REST API calls. The body of an error response to the “oauth2/token” and “oauth2/revoke” endpoints have the format:

```
{
    "error":"<error code>",
    "error_description":"<optional description>"
}
```

Oauth2/token Endpoint error codes:

Error code values are all lower-case to match OAuth2 documentation.

Error Code	Error Description
invalid_request	The request was malformed, or contains unsupported parameters
invalid_client	The client authentication failed.
invalid_grant	The provided authorization is invalid.
unauthorized_client	The client application is not allowed to use this grant_type.
unsupported_grant_type	A grant_type other than “password” was used in the request.
invalid_scope	The scope was not set to “api”.

OAuth2/revoke Endpoint error codes

Error code values are all lower-case to match OAuth2 documentation.

Error Code	Error Description
unsupported_token_type	The client tried to revoke an access token on a server not supporting this feature. This error is not supported in DocuSign.
invalid_token	The presented token is invalid.

Basic Scenarios

This section provides some basic scenarios for using the DocuSign REST API. The examples in this section explain basic tasks such as creating envelopes, checking envelope status and retrieving envelopes and documents.

Creating an Envelope from a Document

In this scenario, we are sending a text document for signature to a single recipient. This basic scenario example only uses a portion of the envelope parameters, refer to the Send an Envelope or Create a Draft Envelope section for information about all the envelope parameters.

The basic information needed to create and send an envelope is an email subject, recipient (with name and email address) and a document. This example does not include any tabs, so the recipient will drag their signature into place on the document (this is referred to as free-form signing).

The `/accounts/{accountId}/envelopes` uri is appended to the baseUrl value with the `/login_information` api to send a document for signing.

> **Note:** After using `login_information` to get the baseUrl, the base Url and apiPassword can be stored by the client. The `login_information` call is not needed in subsequent sessions, unless another call fails and the user needs to re-authenticate. Each call performs authentication.

This is a multipart form request. The general envelope information (status, email text and email subject) appears first, followed by the document bytes and recipient information. In this example, a text document is used for clarity, but other document types (pdf, Word, etc.) can be used.

The status value in the envelope is used to set if the envelope should be sent (status = sent) or saved as a draft (status = created).

In this example there is only one document and recipient.

Example Request

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: multipart/form-data; boundary=AAA

--AAA
Content-Type: application/json
Content-Disposition: form-data

{
    “status”:”sent”,
    "emailBlurb":"Test Email Body",
    "emailSubject": "Test Email Subject - EnvelopeDefFull",
    "documents": [{
                  "name": "test1.pdf",
                  "documentId":"1",
                  "order":"1"
    }],
    "recipients": {
           "signers" : [{
                  "email": "m.rosey@thomasind.com",
                  "name": "Mike Rosey",
                  "recipientId":"1"
    }]
}

--AAA
Content-Type: application/pdf
Content-Disposition: file; filename="test1.pdf"; documentid=1
```

```
<documents removed>

--AAA--
```

Response

The response returns the envelope ID, status, and date-time stamp for the envelope status. It also includes the uri for the envelope.

The following example shows the header followed by the response json body.

Example Response

```
201 Created
Content-Length: 175
Cache-Control: private
Content-Type: application/json; charset=utf-8
Date: Tue, 06 Mar 2012 17:22:17 GMT

{
    "envelopeId": "c254d988-f501-42d3-a858-feeb50f361e2",
    "status": "sent",
    "statusDateTime": "2012-03-06T17:22:17.2030000Z",
    "uri": "\/envelopes\/c254d988-f501-42d3-a858-feeb50f361e2"
}
```

Creating an Envelope from a Template

A template is a standard envelope with specific documents, set recipient roles, recipient routing order, recipient authentication, signing tabs and information fields. Templates can also contain set signing instructions for the document and signature attachment requests.

Sending from a template is similar to sending an envelope from a document, but a template definition is used instead of adding a document to the envelope and template roles instead of normal recipients. This basic scenario example only uses a portion of the template parameters, refer to the Send an Envelope from a Template section for information about all the template parameters.

As with sending an envelope with a document, the `/accounts/{accountId}/envelopes` uri is appended to the baseUrl value with the `/login_information` api to send a document for signing.

> **Note:** After using `login_information` to get the baseUrl, the base Url and apiPassword can be stored by the client. The `login_information` call is not needed in subsequent sessions, unless another call fails and the user needs to re-authenticate. Each call performs authentication.

The general envelope information (status, email text and email subject) appears first, followed by the template definition, which contains the unique ID for the template and the recipient roles, which have pre-assigned tags.

The status value in the envelope is used to set if the envelope should be sent (status = sent) or saved as a draft (status = created).

In this example, there are two recipients that are assigned to templateRoles.

Example Request

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

```
Accept: application/json
Content-Type: application/json
Content-Disposition: form-data

{
    "status":"sent",
    "emailBlurb":"Test Email Body (Template)",
    "emailSubject": "Test Email Subject (Template)",
    "templateId":"270D920E-C65A-410D-9640-75D2FBEADAC2",
    "templateRoles":[
           {
                  "email":"mike.rosey@docusign.com",
                  "name":"Mike Rosey",
                  "roleName":"Signer 1"
           },
           {
                  "email":"mike.rosey@docusign.com",
                  "name":"Lara Roseleip",
                  "roleName":"Signer 2"
           }
    ]

}
```

Response

The response returns the envelope ID, status, and date-time stamp for the envelope status. It also includes the uri for the envelope.

The following example shows the header followed by the response json body.

Example Response

```
201 Created
Content-Length: 175
Cache-Control: private
Content-Type: application/json; charset=utf-8
Date: Tue, 06 Mar 2012 17:22:17 GMT

{
    "envelopeId": "c254d988-f501-42d3-a858-feeb50f361e2",
    "status": "sent",
    "statusDateTime": "2012-03-06T17:22:17.2030000Z",
    "uri": "/envelopes/c254d988-f501-42d3-a858-feeb50f361e2"
}
```

Embedded Sending

The embedded sending resource retrieves a url for accessing the tagging and sending page of the DocuSign console. The envelope ID used here is the envelope that is opened in the console view.

To open an envelope in the console for embedded sending, the `/accounts/{accountId}/envelopes/{envelopeId}/views/sender` uri is appended to the base Url value to get the url to open the envelope in the DocuSign console.

After opening the envelope in the console, you can make changes to the envelope and send it.

Example Request

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/views/sender
```

```
X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
{
    "returnUrl":"https://www.docusign.com"
}
```

Response

The response returns the url to access the console.

The following example shows the header followed by the response json body.

Example Response

```
201 Created
Content-Length: 168
Cache-Control: private
Content-Type: application/json; charset=utf-8
Date: Fri, 30 Mar 2012 14:53:16 GMT

{"url":"http:\/\/localhost\/Member\/StartInSession.aspx?StartConsole=1&t=6cafaeaa-cc6b-
496d-812d-421b72f854de&DocuEnvelope=AB52A90E-BDC7-4F59-BFBD-90E32E984EA7&send=1"}
```

Embedded Signing

The embedded signing resource retrieves a url for accessing the recipient view of a DocuSign envelope. This could be used to incorporate the DocuSign signing process directly into your online process flow, without requiring the signer to leave your web page.

To open an envelope embedded signing, the `/accounts/{accountId}/envelopes/{envelopeId}/views/recipient` uri is appended to the base Url value to get the url to open the envelope.

The basic information needed is the envelope ID, recipient information (email, user name), and authentication information.

The authenication information (authenticationMethod and authenticationInstant) are sender-created items that must be included in the request. The authentication information is included in the Cerrtificate of Completion for the envelope.

The return URL provides DocuSign with information about where to direct the sender for certain events, such as completing signing or for a session timeout. See Post Recipient View for more information about the returnURL.

Example Request

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/views/recipient

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
{
    "authenticationMethod":"email",
    "authenticationInstant":"30-03-2012 | 10:05",
```

```
    "email":"mike.rosey@docusign.com",
    "returnUrl":"https://www.docusign.com",
    "userName":"Mike Rosey"
}
```

Response

The response returns the url to access a recipient's view.

The following example shows the header followed by the response json body.

Example Response

```
201 Created
Content-Length: 96
Cache-Control: private
Content-Type: application/json; charset=utf-8
Date: Fri, 30 Mar 2012 14:52:58 GMT

{"url":"http:\/\/localhost\/Member\/StartInSession.aspx?t=d1cf42f2-30b6-499b-ab54-
058fbf438103"}
```

Requesting Envelope Status

This API should be used to periodically determine if the document has been signed and envelope completed by the recipients. DocuSign imposes API call limit restrictions to prevent flooding of API calls, so this API should be called at periodic intervals, for example, once per hour.

To get general envelope status the `/accounts/{accountId}/envelopes` uri is appended to the baseUrl value.

The information returned can be modified by adding query strings to limit the request to check between certain dates and times, or for certain envelopes, or for certain status codes. It is recommended that you use one or more of the query strings in order to limit the size of the response. Unless you are requesting the status for specific envelopes (using envelopeIds), you must add a from_date in the request.

Refer to the Get Envelope Status Changes section for information about the query strings.

Example Request

This example requests any envelopes that changed Completed status since May 2, 2012.

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
?from_date=5-2-2012+12:00:00+AM&from_to_status=changed&status=completed

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the uri's for the envelope certificate, custom fields, documents, envelope along with the envelope ID, envelope status, and the date-time stamp for the envelope status change for the envelopes. If a query string was added to the original request, only the envelopes that meet those parameters are included in the response.

The following example shows the header followed by the response json body.

Example Response

```
200 OK
Content-Length: 1221
Cache-Control: private
Content-Type: application/json; charset=utf-8
Date: Tue, 06 May 2012 17:22:22 GMT
X-AspNet-Version: 4.0.30319

{
    "envelopes": [{
            "certificateUri": "\/envelopes\/fd1afecb-5fc4-4f11-aac4-
0d6fe089105f\/documents\/certificate",
            "customFieldsUri": "\/envelopes\/fd1afecb-5fc4-4f11-aac4-
0d6fe089105f\/custom_fields",
            "documentsCombinedUri": "\/envelopes\/fd1afecb-5fc4-4f11-aac4-
0d6fe089105f\/documents\/combined",
            "documentsUri": "\/envelopes\/fd1afecb-5fc4-4f11-aac4-
0d6fe089105f\/documents",
            "envelopeId": "fd1afecb-5fc4-4f11-aac4-0d6fe089105f",
            "envelopeUri": "\/envelopes\/fd1afecb-5fc4-4f11-aac4-0d6fe089105f",
            "notificationUri": "\/envelopes\/fd1afecb-5fc4-4f11-aac4-
0d6fe089105f\/notification",
            "recipientsUri": "\/envelopes\/fd1afecb-5fc4-4f11-aac4-
0d6fe089105f\/recipients",
            "status": "completed",
            "statusChangedDateTime": "2012-05-23T16:21:09.2830000Z"
    },
    {
            "certificateUri": "\/envelopes\/23306db8-261a-43c5-b078-
ac020113af09\/documents\/certificate",
            "customFieldsUri": "\/envelopes\/23306db8-261a-43c5-b078-
ac020113af09\/custom_fields",
            "documentsCombinedUri": "\/envelopes\/23306db8-261a-43c5-b078-
ac020113af09\/documents\/combined",
            "documentsUri": "\/envelopes\/23306db8-261a-43c5-b078-
ac020113af09\/documents",
            "envelopeId": "23306db8-261a-43c5-b078-ac020113af09",
            "envelopeUri": "\/envelopes\/23306db8-261a-43c5-b078-ac020113af09",
            "notificationUri": "\/envelopes\/23306db8-261a-43c5-b078-
ac020113af09\/notification",
            "recipientsUri": "\/envelopes\/23306db8-261a-43c5-b078-
ac020113af09\/recipients",
            "status": "completed",
            "statusChangedDateTime": "2012-05-23T16:20:44.7830000Z"
    }],
    "resultSetSize": 2
}
```

Retrieving Envelope and Documents

When an envelope reaches the “completed” status, the signed document and associated certificate can be downloaded from the server. The envelope uri information from the envelope status call can be used with the uri part `/documents/combined` appended to download the combined document and certificate in PDF format.

There are two optional flags associated with this action: watermark and certificate.

- watermark adds a watermark to the PDF if the envelope is not complete. It is a Boolean (true/false) setting that is set to true by default.
- certificate adds the certificate to the documents. It is a Boolean (true/false) that is set to false by default.

Example Request

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/combined

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/pdf
Content-Type: application/json
```

Response

The response returns the document and certificate as a PDF document.

The following example shows the header followed by the response json body (the byte stream is not included to limit the size of the example).

Example Response

```
200 OK
Content-Length: 100079
Cache-Control: private
Content-Type: application/pdf
Date: Tue, 06 Mar 2012 17:22:10 GMT

<Bytes of PDF omitted>
```

Sending HTTP Requests with cURL

To interact with the DocuSign REST API, you need to set up your client application (we use cURL) to construct HTTP requests.

Note: The examples in this section have hard returns (\n) added so the text is easier to read. If you copy and paste the examples, you will need to remove the returns in order for the example to function correctly. Additionally, you must replace the DocuSign Credentials with your specific credentials.

Setting Up Your Client Application

The REST API uses HTTP GET and HTTP POST methods to send and receive JSON and XML content, so it is easy to build client applications using the tool or the language of your choice. In this example, we use a command-line tool called cURL to simplify sending and receiving HTTP requests and responses.

cURL is pre-installed on many Linux and Mac systems. Windows users can download a version at: curl.haxx.se/.

When using HTTPS on Windows, ensure that your system meets the cURL requirements for SSL.

Viewing Base Resources with cURL

You can use cURL to request the base resources available for a particular REST version.

The following example requests the base resources for REST version 2.

```
curl -i -k -H "Accept: application/json" https://demo.docusign.net/restapi/v2/
```

The response would appear as:

```
{"resources":[{"name":"accounts","value":"https:\/\/demo.docusign.net\/restapi\/v2\/accou
nts"},{"name":"billing_plans","value":"https:\/\/demo.docusign.net\/restapi\/v2\/billing_
plans"},{"name":"login_information","value":"https:\/\/demo.docusign.net\/restapi\/v2\/lo
gin_information"}]}
```

Logging in with cURL

Before sending any requests, you need to log on to the system to get the account and baseUrl information for future calls. You can log on with json or xml.

cURL json Login

An example cURL Login (login_information) using json would be (you would have to replace the credentials in the example with your actual DocuSign credentials):

```
curl https://demo.docusign.net/restapi/v2/login_information
-i -k -H "Accept: application/json"
-H "Content-Type: application/json"
-H "X-DocuSign-Authentication:
<DocuSignCredentials><Username>account@domain.com</Username><Password>account_password</P
assword><IntegratorKey>your integrator key</IntegratorKey></DocuSignCredentials>"
```

The response returns account information for the account associated with your DocuSign credentials.

```
{"loginAccounts":[{"accountId":"accountId#","baseUrl":"https:\/\/demo.docusign.net\/resta
pi\/v2\/accounts\/accountId#","email":"account@domain.com","isDefault":"true","name":"Use
r Name","siteDescription":"","userId":"userId#","userName":"User Name"}]}
```

cURL XML Login

An example cURL Login (login_information) using xml would be (you would have to replace the credentials in the example with your actual DocuSign credentials):

```
curl https://demo.docusign.net/restapi/v2/login_information.XML
-i -k -H "Accept: application/json"
-H "Content-Type: application/json"
-H "X-DocuSign-Authentication:
<DocuSignCredentials><Username>account@domain.com</Username><Password>account_password</P
assword><IntegratorKey>your integrator key</IntegratorKey></DocuSignCredentials>"
```

The response returns account information for the account associated with your DocuSign credentials.

```
<loginInformation xmlns="http://www.docusign.com/restapi"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
<loginAccounts><loginAccount><accountId>account#</accountId><baseUrl>https://demo.docusig
n.net/restapi/v2/accounts/accountId#</baseUrl><email> account@domain.com
</email><isDefault>true</isDefault><name>User
Name</name><siteDescription/><userId>usedId#</userId><userName>User
Name</userName></loginAccount></loginAccounts></loginInformation>
```

Sending HTTP Requests Using REST API Resources

Your HTTP requests to a REST API resource should contain the following information:

- An HTTP method (GET, POST, PUT, or DELETE)
- Your HTTP Authentication Token ()
- An HTTP ACCEPT header used to indicate the resource format (XML or JSON), or a .json or .xml extension to the URI. The default is JSON.

- The DocuSign REST resource.
- Any JSON or XML files containing information needed for requests, such as updating a record with new information.

The HTTP methods are used to indicate the desired action, such as retrieving information, as well as creating, updating, and deleting records.

- HEAD is used to retrieve resource metadata.
- GET is used to retrieve information, such as basic resource summary information.
- POST is used to create a new object.
- PUT is used to add information or items to an existing object.
- DELETE is used to delete a record.

To access a resource, submit an HTTP request containing a header, method, and resource name.

Example Sending HTTP Requests with cURL

For example, assume that we want to create an envelope from a template. The envelope information is stored in a json-formatted file called envelopeFromTemplate.json.

Note: You should create a template with one signer before trying to send an envelope from a template.

envelopeFromTemplate.json contains the following data:

```
{
    "emailBlurb":"Test Email Body (Template)",
    "emailSubject": "Test Email Subject (Template)",
    "templateId":"270D920E-C65A-410D-9640-75D2FBEADAC2",
    "templateRoles":[
                {
                        "email":"mike.rosey@myDomain.com",
                        "name":"Mike Rosey",
                        "roleName":"Signer 1"
                }
            ]
}
```

The REST Resource for creating an envelope from a template:

`https://demo.docusign.net/restapi/:version/:accountId/envelopes`

:version – REST API version

:accountId – user account id

An example of using the cURL on demo.docusign.net, the request would be:

```
curl -x POST -- upload-file envelopeFromTemplate.json
https://demo.docusign.net/restapi/v2/532/envelopes
-H "Content-Type: application/json"
-H "Accept: application/json"
-H  "X-DocuSign-Authentication:
<DocuSignCredentials><Username>account@domain.com</Username><Password>account
passworld</Password><IntegratorKey>your integrator
key</IntegratorKey></DocuSignCredentials>"
```

An example of the cURL one-line command would be:

```
curl -x POST -- upload-file envelopeFromTemplate.json
https://demo.docusign.net/restapi/v2/532/envelopes
-H "Content-Type: application/json"
-H "Accept: application/json"
-H  "X-DocuSign-Authentication:
<DocuSignCredentials><Username>account@domain.com</Username><Password>account
password</Password><IntegratorKey>your integrator
key</IntegratorKey></DocuSignCredentials>"
```

Chunked Transfer-Encoding Support

Because an API user might not always know the size of a document at the time of transmission, for example a document that is dynamically generated or retrieved from a network, a way to transfer the document as it is received is needed.

The chunked transfer-encoding support allows POST/PUT with Transfer-Encoding: chunked instead of by setting Content-Length header. This allows API users to upload data without knowing size in advance and without pre-buffering.

Note: Because it can be a complicated operation, this is best used from client toolkits that support chunked Transfer-Encoding.

For more information see the HTTP Spec regarding "transfer-encoding: chunked" at:

- http://www.w3.org/Protocols/rfc2616/rfc2616-sec4.html#sec4.4
- http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.6

Example Chunked Transfer-Encoding

In general, chunked requests add HTTP/1.1 to the POST/PUT and a Transfer-Encoding:chunked header. The body of the request is then provided in chunked transfer coding format, where the length of each chunk, listed as a HEX number, precedes the chunk. The last chunk length must be 0.

This section has an example of sending a request to create an envelope via the C#/.Net HttpWebRequest. To do this, the "sendChunked" property is set to true and the AllowWriteStreamBuffering property is set to false. Otherwise the HttpWebRequest is identical to the non-chunked request. The bold sections in the example show the difference from a normal request. Note that chunk lengths are spread throughout the request, corresponding to chunks written by the client to the request stream.

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes HTTP/1.1

X-DocuSign-Authentication: [credentials omitted for brevity]
Accept: application/json
Content-Type: multipart/form-data; boundary=c009ab22-d9ff-4f9d-87ca-3b7a79d092e3
Transfer-Encoding:chunked
Expect: 100-continue
Connection: Keep-Alive

2A

--c009ab22-d9ff-4f9d-87ca-3b7a79d092e3

E
Content-Type:
12
application/json
```

```
15
Content-Disposition:

B
form-data

2

2706
{
    "status" : "sent",
    "emailBlurb":"Test Email Body",
    "emailSubject": "Test Email Subject - EnvelopeDefFull",
    "asynchronous": true,
           "enforceSignerVisibility": true,
    "documents": [{
           "name": "test1.pdf",
           "documentId": 1
    }],
    "recipients": {
           "signers" :
           [
                  {
                         "email": "m.rosey@thomasind.com",
                         "name": "Mike Rosey",
                         "recipientId": "1",
                         "enforceSignerVisibility": true,
                         "tabs": {
                                [tabs ommitted for brevity]
                         }
                  }
           ]
    }
}
2A

--c009ab22-d9ff-4f9d-87ca-3b7a79d092e3

E
Content-Type:
11
application/pdf

15
Content-Disposition:
1C
file; filename="test1.pdf"

1B
Content-Transfer-Encoding:
8
base64

2


3014
[base64 buffer omitted for brevity]
2A

--c009ab22-d9ff-4f9d-87ca-3b7a79d092e3—
0
```

REST API References

This table below lists the supported REST resources, with the Login shown first followed by the other resources in URI alphabetical order, and provides a brief description and link for each resource.

In all cases, the URI for the resource follows the base URI, which is retrieved from the login.

Name	HTTP Method	URI	Description
Login	GET	https://{server}/restapi/{apiVersion}/login_information	Authenticates and provides account login information and a Base Url.
Change Password	PUT	/login_information/password	Changes the password for a user.
Create Account	POST	/accounts	Creates an account for the DocuSign service.
Create Multiple Accounts	POST	/accounts	Creates multiple accounts.
Delete Account	DELETE	/accounts/{accountId}	Delete the specified account.
Get Account Information	GET	/accounts/{accountId}	Returns Account information for specified account.
Get Account Billing Plan	GET	/accounts/{accountId}/billing_plan	Returns billing plan information for the specified account.
Update Account Billing Plan	PUT	/accounts/{accountId}/billing_plan	Updates the billing plan for an account.
Get Billing Plan Information	GET	/accounts/{accountId}/billing_payments	Returns information about one or more payments.
Post a Billing Payment	POST	/accounts/{accountId}/billing_payments	Allows users with account administrator privileges to pay a past due invoice.
Get List of Billing Invoices	GET	/accounts/{accountId}/billing_invoices	Returns a list of invoices for the account.
Get a Billing Invoice	GET	/accounts/{accountId}/billing_invoices/{invoiceId}	Returns the requested invoice.

Name	HTTP Method	URI	Description
Get Past Due Invoices	GET	/accounts/{accountId}/billing_invoices_past_due	Returns any past due invoices for the account and notes if payment can be made through the REST API.
Get Billing Charges	GET	/accounts/{accountId}/billing_charges	Returns the list of recurring and usage charges for the account.
Purchase Additional Envelopes	PUT	/accounts/{accountId}/billing_plan/ purchased_envelopes	Updates purchased envelope information.
Get Brand Profile Information	GET	/accounts/{accountId}/brands	Returns a list of brand profiles associated with the account and the default brand profiles.
Upload Brand Profiles	POST	/accounts/{accountId}/brands	Uploads one or more brand profile files in a zip file for the account.
Delete Brand Profiles	DELETE	/accounts/{accountId}/brands	Deletes one or more brand profiles from an account.
Check Status of all Bulk Recipient Batches	GET	/accounts/{accountId}/bulk_envelopes	Returns status information about all the bulk recipient batches.
Check the Status of a Bulk Send Batch	GET	/accounts/{accountId}/bulk_envelopes/{batchId}	Returns status information about a single bulk recipient batch.
Delete Captive Recipient Signature	DELETE	/accounts/{accountId}/captive_recipients/signature	This deletes the signature for one or more captive recipient records.
Set Up a Connect Configuration	POST	/accounts/{accountId}/connect	Sets up and configures a DocuSign Custom Connect definition.

Name	HTTP Method	URI	Description
Get All Connect Configuration Information	GET	/accounts/{accountId}/connect	Retrieves all the DocuSign Custom Connect definitions for your account.
Update a Connect Configuration	PUT	/accounts/{accountId}/connect	Updates a DocuSign Connect configuration.
Get a Connect Configuration Information	GET	/accounts/{accountId}/connect/{connectId}	Retrieves the information about one DocuSign Connect configuration.
Delete a Connect Configuration	DELETE	/accounts/{accountId}/connect/{connectId}	Deletes one DocuSign Connect configuration.
Republish Connect Information for One Envelope	PUT	/accounts/{accountId}/connect/envelopes/{envelopeId}/retry_queue	Republishes Connect information for a single envelope.
Republish Connect Information for Multiple Envelopes	PUT	/accounts/{accountId}/connect/envelopes/retry_queue	Republishes Connect information for the set of envelopes.
Get the Connect Failure Log	GET	/accounts/{accountId}/connect/failures	Retrieves the Connect Failure Log information.
Get a Connect Failure Log Entry	GET	/accounts/{accountId}/connect/failures/{failureId}	Retrieves the Connect Failure Log information for one entry.
Clear a Connect Failure Log Entry	DELETE	/accounts/{accountId}/connect/failures/{failureId}	Clears (removes) the Connect Failure Log information for one entry.
Get the Connect Log	GET	/accounts/{accountId}/connect/logs	Retrieves a list of connect log entries for your account.
Clear the Connect Log	DELETE	/accounts/{accountId}/connect/logs	Clears (removes) the entries from the Connect Log.
Get One Connect Log Entry	GET	/accounts/{accountId}/connect/logs/{logId}	Retrieves one Connect log entry.

Name	HTTP Method	URI	Description
Clear One Connect Log Entry	DELETE	/accounts/{accountId}/connect/logs/{logId}	Clears (removes) one entry from the Connect Log.
Get Disclosure	GET	/accounts/{accountId}/consumer_disclosure	Returns the Electronic Record and Signature Disclosure, with html formatting, associated with the account
Get List of Account Custom Fields	GET	/accounts/{accountId}/custom_fields	Gets a list of all envelope custom fields associated with the account.
Send an Envelope or send an Envelope from a Template including creating Draft Envelopes	POST	/accounts/{accountId}/envelopes	If “status” is set to “sent”, this creates an envelope and sends it to recipients. If “status” is set to “created”, this creates an envelope in the draft folder. This can use a multipart/form-data format that may include binary files. This can use a template.
Get Envelope Status changes	GET	/accounts/{accountId}/envelopes	Returns envelope status changes for all envelopes.
Get Envelope Status for one envelope	GET	/accounts/{accountId}/envelopes/{envelopeId}	Returns the overall status for a single envelope.
Send Draft Envelope	PUT	/accounts/{accountId}/envelopes/{envelopeId}	Sends a draft envelope by adding “status”:”sent” to request body.

Name	HTTP Method	URI	Description
Void an Envelope	PUT	/accounts/{accountId}/envelopes/{envelopeId}	Voids an in process envelope by adding “status”:”voided” and a reason to “voidReason” to the request body.
Modify Draft Envelope Email Subject and Message	PUT	/accounts/{accountId}/envelopes/{envelopeId}	Allows senders to change the email subject and message for draft envelope.
Purge Documents	PUT	/accounts/{accountId}/envelopes/{envelopeId}	Initiates a request to place envelope documents in a purge queue.
Add or Modify Envelope Information	PUT	/accounts/{accountId}/envelopes/{envelopeId}	Add or modify draft or in-process envelope information.
Get Envelope Purge Document Status	GET	/accounts/{accountId}/envelopes/{envelopeId}	Retrieves the current purge state for an envelope.
Get Envelope Audit Events	GET	/accounts/{accountId}/envelopes/{envelopeId}/audit_events	Returns the events for this envelope.
Add Envelope Custom Fields to an Envelope	POST	/accounts/{accountId}/envelopes/{envelopeId}/custom_fields	Adds envelope custom fields to draft and in-process envelopes.
Get Envelope Custom Field Information	GET	/accounts/{accountId}/envelopes/{envelopeId}/custom_fields	Returns the custom field information for a single envelope.
Modify Envelope Custom Fields for an Envelope	PUT	/accounts/{accountId}/envelopes/{envelopeId}/custom_fields	Modify the envelope custom fields for draft and in-process envelopes.
Remove Envelope Custom Fields from an Envelope	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/custom_fields	Removes envelope custom fields for draft and in-process envelopes.

Name	HTTP Method	URI	Description
Add or Modify Documents and Document Fields	PUT	/accounts/{accountId}/envelopes/{envelopeId}/documents	Add or modify documents in created or sent envelopes.
Remove Documents from a Draft Envelope	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/documents	Removes one or more documents from draft envelope.
Get list of Envelope documents	GET	/accounts/{accountId}/envelopes/{envelopeId}/documents	Returns a list of documents associated with the specified envelope.
Add one Document to an Envelope	PUT	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}	Adds one document to an envelope.
Get document from Envelope	GET	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}	Retrieves a document from the envelope.
Get a Page Image	GET	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/pages/{pageId}/page_image	Returns a page image for display.
Rotate a Page Image	PUT	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/pages/{pageId}/page_image	Rotates a page image for display.
Remove a Page	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/pages/{pageId}	Removes a page from a document based on the page ID.
Get Custom Document Fields for an Envelope Document	GET	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/fields	Retrieves the custom document field information for an existing envelope document.
Add Custom Document Fields to an Envelope Document	POST	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/fields	Adds custom document fields for an existing envelope document.
Modify Custom Document Fields for an Envelope Document	PUT	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/fields	Modifies existing custom document fields for an existing envelope document.

Name	HTTP Method	URI	Description
Delete Custom Document Fields from an Envelope Document	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/fields	Deletes custom document fields for an existing envelope document.
Get Envelope Certificate	GET	/accounts/{accountId}/envelopes/{envelopeId}/documents/certificate	Retrieves a PDF document containing the certificate of completion for the envelope.
Get Envelope documents and Certificate	GET	/accounts/{accountId}/envelopes/{envelopeId}/documents/combined	Retrieves a PDF containing the combined envelope documents and certificate of completion.
Add Email Setting Overrides to an Envelope	POST	/accounts/{accountId}/envelopes/{envelopeId}/email_settings	Adds email override settings, changing the reply to email address or name or the BCC for email archive information, for the envelope.
Get Email Setting Overrides for an Envelope	GET	/accounts/{accountId}/envelopes/{envelopeId}/email_settings	Retrieves email override settings for the envelope.
Modify Email Setting Overrides for an Envelope	PUT	/accounts/{accountId}/envelopes/{envelopeId}/email_settings	Modifies existing email override settings for the envelope.
Delete Email Setting Overrides for an Envelope	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/email_settings	Deletes all existing email override settings.
Lock an Envelope or Template	POST	/accounts/{accountId}/envelopes/{envelopeId}/lock	Locks an envelope or template.

Name	HTTP Method	URI	Description
Get Envelope or Template Lock Information	GET	/accounts/{accountId}/envelopes/{envelopeId}/lock	Returns information about the lock and, if requested by lockedByUser, X-DocuSign-Edit header.
Update Envelope or Template Lock	PUT	/accounts/{accountId}/envelopes/{envelopeId}/lock	Extends lock duration time or update lockedByApp information.
Delete Envelope or Template Lock	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/lock	Removes lock from envelope or template.
Get Envelope Notification Information	GET	/accounts/{accountId}/envelopes/{envelopeId}/notification	Returns the reminder and expiration information for the envelope.
Get Envelope Recipient Status	GET	/accounts/{accountId}/envelopes/{envelopeId}/recipients	Returns the status for all recipients of a single envelope.
Add Recipients to an Envelope	POST	/accounts/{accountId}/envelopes/{envelopeId}/recipients	Adds one or more recipients to an envelope.
Delete Recipients from an Envelope	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/recipients	Removes one or more recipients from an envelope.
Modify or Correct and Resend Recipient Information	PUT	/accounts/{accountId}/envelopes/{envelopeId}/recipients	Modifies a recipient in a draft envelope or to update and resend recipient information for an in process envelope. Can also be used to resend an envelope.
Add or Replace an Envelope Bulk Recipient File	PUT	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/bulk_recipients	Adds or replaces a bulk recipient file to a draft envelope.

Name	HTTP Method	URI	Description
Retrieve Envelope Bulk Recipient File	GET	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/bulk_recipients	Retrieves the bulk recipient file information for an envelope with a bulk recipient.
Delete Envelope Bulk Recipient File	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/bulk_recipients	Removes the bulk recipient file from an envelope. This cannot be used if the envelope has been sent.
Set Initials Image for Accountless Signer	PUT	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/initials_image	Sets a recipient signature initials image.
Get Signature Information for Accountless Signer	GET	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/signature	Gets a recipient's signature information.
Set Signature Image for Accountless Signer	PUT	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/signature_image	Sets a recipient signature image.
Add Tabs for a Recipient	POST	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs	Adds one or more tabs for a recipient.
Get Tab Information for a Recipient	GET	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs	Retrieve tabs information for a recipient.
Delete Tabs for a Recipient	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs	Delete one or more tabs for a recipient.
Modify Tabs for a Recipient	PUT	/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs	Modify one or more tabs for a ricipeint in a draft envelope.
Get a List of Templates used in an envelope	GET	/accounts/{accountId}/envelopes/{envelopeId}/templates	Returns a list of the templates, name and ID, used in an envelope.
Post an Envelope Correction	POST	/accounts/{accountId}/envelopes/{envelopeId}/views/correct	Provides a URL to start the correction view of the DocuSign UI.
Post Recipient View	POST	/accounts/{accountId}/envelopes/{envelopeId}/views/recipient	Provides a URL to start the recipient view of the DocuSign UI.

Name	HTTP Method	URI	Description
Post Sender View	POST	/accounts/{accountId}/envelopes/{envelopeId}/views/sender	Provides a URL to start the sending view of the DocuSign UI.
Post Edit View	POST	/accounts/{accountId}/envelopes/{envelopeId}/views/edit	Provides a URL to start the sending view of the DocuSign UI.
Get Envelope Status for more than one envelope	PUT	/accounts/{accountId}/envelopes/status	Returns envelope status for a selected list of envelopes.
Get Folder List	GET	/accounts/{accountId}/folders	Returns a list of the folders for the account, including the folder hierarchy.
Get Folder Envelope List	GET	/accounts/{accountId}/folders/{folderId}	Returns a list of envelopes in the folder.
Move Envelopes or delete envelopes	PUT	/accounts/{accountId}/folders/{folderId}	Moves envelopes to the listed folder. This can be used to delete envelopes by moving the envelope to the recycle bin folder.
Add a New Group	POST	/accounts/{accountId}/groups	Adds new user groups.
Get Group Information	GET	/accounts/{accountId}/groups	Returns a list of user groups in the account.
Modify Group Information	PUT	/accounts/{accountId}/groups	Modifies user group information.
Delete Group information	DELETE	/accounts/{accountId}/groups	Deletes a user group.
Get Group Brand ID Information	GET	/accounts/{accountId}/groups/{groupId}/brands	Returns information about the brands associated with the requested group.
Add Group Brand ID Information	PUT	/accounts/{accountId}/groups/{groupId}/brands	Adds brand information to a group.

Name	HTTP Method	URI	Description
Delete Group Brand ID Information	DELETE	/accounts/{accountId}/groups/{groupId}/brands	Removes brand information from the requested group.
Add Users to a Group	PUT	/accounts/{accountId}/groups/{groupId}/users	Adds users to a user group.
Get List of Users in a Group	GET	/accounts/{accountId}/groups/{groupId}/users	Returns a list of users in a user group.
Remove Users from a Group	DELETE	/accounts/{accountId}/groups/{groupId}/users	Removes a user from a user groups.
Get a List of Permission Profiles	GET	/accounts/{accountId}/permission_profiles	Returns a list of permission profiles for the account.
Get Recipient Names	GET	/accounts/{accountId}/recipient_names	Returns the recipient names associated with an email address in the account.
Get List of Envelopes in a Folders	GET	/accounts/{accountId}/search_folders/ {search_folder}	Returns a list of envelopes from the specified folder.
Get Account Settings	GET	/accounts/{accountId}/settings	Returns the settings list for specified account
Modify Account Settings	PUT	/accounts/{accountId}/settings	Modifies the setting list for specified account
Get Shared Access Information	GET	/accounts/{accountId}/shared_access	Retrieves shared item status for one or more users and types of items.
Set Shared Access Information	PUT	/accounts/{accountId}/shared_access	Sets the shared access status for one or more users.

Name	HTTP Method	URI	Description
List Signature Providers	GET	/accounts/{accountId}/signatureproviders	Returns a list of signature providers and associated information available to the specified account. Signature providers are used with DocuSign's Standards-Based Signatures.
Create a Signing Group	POST	/accounts/{accountId}/signing_groups	Creates one or more signing groups.
Get a List of Signing Groups	GET	/accounts/{accountId}/signing_groups	Returns a list of all signing groups for the account.
Update Signing Group Names	PUT	/accounts/{accountId}/signing_groups	Updates the name of one or more existing signing groups.
Delete Signing Groups	DELETE	/accounts/{accountId}/signing_groups	Deletes one or more signing groups.
Get a Signing Group	GET	/accounts/{accountId}/signing_groups/{signingGroupId}	Returns information, including group member information, for one signing group.
Update a Signing Group	PUT	/accounts/{accountId}/signing_groups/{signingGroupId}	Updates signing group name and member information.
Get Signing Group Members	GET	/accounts/{accountId}/signing_groups/{signingGroupId}/users	Returns the group member information for one signing group.
Add Signing Group Members	PUT	/accounts/{accountId}/signing_groups/{signingGroupId}/users	Adds one or more new members to a signing group.
Delete Signing Group Members	DELETE	/accounts/{accountId}/signing_groups/{signingGroupId}/users	Removes one or more members from a signing group.

Name	HTTP Method	URI	Description
Create a Custom Tab	POST	/accounts/{accountId}/tab_definitions	Allows you to create a tab with pre-defined properties, such as a text tab with a certain font type and validation pattern.
Get a List of all Account tabs	GET	/accounts/{accountId}/tab_definitions	Retrieves a list of all tabs associated with the account.
Get Custom Tab Information	GET	/accounts/{accountId}/tab_definitions/{customtabid}	Retrieves information about the requested custom tab.
Modify Custom Tab Information	PUT	/accounts/{accountId}/tab_definitions/{customtabid}	Allows you to update information in a custom tab.
Delete a Custom Tab	DELETE	/accounts/{accountId}/tab_definitions/{customtabid}	Deletes a specified custom tab.
Get List of Templates	GET	/accounts/{accountId}/templates	Retrieves the list of templates for this account.
Post a Template	POST	/accounts/{accountId}/templates	Posts a template definition using a multipart request.
Get a Template	GET	/accounts/{accountId}/templates/{templateId}	Retrieves the definition for an existing template.
Modify a Template	PUT	/accounts/{accountId}/templates/{templateId}	Allows users to modify an existing template.
POST Template Edit View	POST	/accounts/{accountId}/templates/{templateId}/views/edit	Provides a URL to the template edit view of the DocuSign UI for the specified template ID.
Get Custom Document Fields for a Template Document	GET	/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields	Retrieves the custom document fields for an existing document in a template.

Name	HTTP Method	URI	Description
Add Custom Document Fields to a Template Document	POST	/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields	Adds custom document fields for an existing template document.
Modify Custom Document Fields for a Template Document	PUT	/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields	Modifies existing custom document fields for an existing template document.
Delete Custom Document Fields from a Template Document	DELETE	/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields	Deletes custom document fields for an existing template document.
Share a Template with a Group	PUT	/accounts/{accountId}/templates/{templateId}/groups	Shares a template with a user group.
Remove Template Sharing for a Group	DELETE	/accounts/{accountId}/templates/{templateId}/groups	Removes template sharing from a user group.
Get Template Recipient Information	GET	/accounts/{accountId}/templates/{templateId}/recipients	This returns the information for all recipients in a single template.
Add or Replace a Template Bulk Recipient File	PUT	/accounts/{accountId}/template/{templateId}/recipients/{recipientId}/bulk_recipients	Adds or replaces a bulk recipient file to a template.
Retrieve Template Bulk Recipient File	GET	/accounts/{accountId}/template/{templateId}/recipients/{recipientId}/bulk_recipients	Retrieves the bulk recipient file information for a template with a bulk recipient.
Delete Template Bulk Recipient File	DELETE	/accounts/{accountId}/template/{templateId}/recipients/{recipientId}/bulk_recipients	Removes the bulk recipient file from a template.
Get Tab Information for a Template Recipient	GET	/accounts/{accountId}/templates/{templateId}/recipients/{recipientId}/tabs	This retrieves information about the tabs associated with a recipient in a template.

Name	HTTP Method	URI	Description
Get List of Unsupported File Types	GET	/accounts/{accountId}/unsupported_file_types	Retrieves a list of file types that are not supported for uploads.
Add User to Account	POST	/accounts/{accountId}/users	Add users to an account
Get User list	GET	/accounts/{accountId}/users	Returns a list of users in the specified account.
Close a User	DELETE	/accounts/{accountId}/users	Closes a user for the account.
Get User Information	GET	/accounts/{accountId}/users/{userId}	Retrieves the user information for specified user.
Get Cloud Storage Provider Configurations for a User	GET	/accounts/{accountId}/users/{userId}/cloud_storage	Retrieves the list of cloud storage providers enabled for the account and the configuration information for the user.
Add Cloud Storage Provider Return Information for a User	POST	/accounts/{accountId}/users/{userId}/cloud_storage	Configures the redirect URL information added to the authentication URL for one or more cloud storage providers for a user.
Delete Cloud Storage Provider Authentications for a User	DELETE	/accounts/{accountId}/users/{userId}/cloud_storage	Removes the user authentication information for one or more cloud storage providers.
Get One Cloud Storage Provider Configuration for a User	GET	/accounts/{accountId}/users/{userId}/cloud_storage/{serviceId}	Retrieves the configuration information for one cloud storage provider for the user.
Delete One Cloud Storage Provider Authentication for a User	DELETE	/accounts/{accountId}/users/{userId}/cloud_storage/{serviceId}	Removes the user authentication information for one cloud storage providers.

Name	HTTP Method	URI	Description
Get User Items in a Cloud Storage Provider	GET	/accounts/{accountId}/users/{userId}/cloud_storage/{serviceId}/folders	Retrieves a list of all items in all folders for the user from the selected cloud storage provider.
Get User Items in one Cloud Storage Provider Folder	GET	/accounts/{accountId}/users/{userId}/cloud_storage/{serviceId}/folders/{folderId}	Retrieves a list of all items in one folder for the user from the selected cloud storage provider.
Get Custom User Settings	GET	/accounts/{accountId}/users/{userId}/custom_settings	Retrieves a list of custom user settings for a single user.
Add or Modify Custom User Settings	PUT	/accounts/{accountId}/users/{userId}/custom_settings	Add or update custom user settings for a single user.
Delete Custom User Settings	DELETE	/accounts/{accountId}/users/{userId}/custom_settings	Deletes the specified custom user settings for a single user.
Get User Profile	GET	/accounts/{accountId}/users/{userId}/profile	Returns the user profile information for the specified user.
Modify User Profile	PUT	/accounts/{accountId}/users/{userId}/profile	Updates the user profile information for the specified user.
Get User Profile image	GET	/accounts/{accountId}/users/{userId}/profile/image	Returns the profile (ID card) image for the specified user.
Modify User Profile image	PUT	/accounts/{accountId}/users/{userId}/profile/image	Updates the profile (ID card) image for the specified user.
Delete User Profile image	DELETE	/accounts/{accountId}/users/{userId}/profile/image	Deletes the profile (ID card) image for the specified user.
Get User Account Settings	GET	/accounts/{accountId}/users/{userId}/settings	Returns settings list for specified user.

Name	HTTP Method	URI	Description
Modify User Account Settings	PUT	/accounts/{accountId}/users/{userId}/settings	Update settings for specified user.
Get a List of User Signature Definitions	GET	/accounts/{accountId}/users/{userId}/signatures	Returns the list of signature definitions for the specified user.
Set User Signature and Initials Images	POST	/accounts/{accountId}/users/{userId}/signatures	This allows user signature and/or initials images to be set when a signature is created.
Get information about a user signature	GET	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}	Returns information about a single signature.
Modify a user signature	PUT	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}	Updates a user signature.
Close a user signature	DELETE	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}	Closes the user signature.
Get a user initials image	GET	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image	Returns a specific initials image.
Set a user initials image	PUT	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image	Saves a specific initials image.
Delete a user initials image	DELETE	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image	Deletes a specific initials image.
Get a user signature image	GET	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/signature_image	Returns a specific signature image.
Set a user signature image	PUT	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/signature_image	Saves a specific signature image.
Delete a user signature image	DELETE	/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/signature_image	Deletes a specific signature image.
Add a User Social Account	PUT	/accounts/{accountId}/users/{userId}/social	Add social account information for a user.
Get User Social Accounts	GET	/accounts/{accountId}/users/{userId}/social	Returns a list of social accounts for a user.
Remove a User Social Account	DELETE	/accounts/{accountId}/users/{userId}/social	Removes a social account for a user.

Name	HTTP Method	URI	Description
Post Authentication View	POST	/accounts/{accountId}/views/console	Provides a URL to start the authentication view of the DocuSign UI
Get Account provisioning information	GET	/accounts/provisioning	Returns Account provisioning information.
Get list of billing plans	GET	/billing_plans	Returns the plans associated with a distributor.
Get billing plan details	GET	/billing_plans/{planId}	Returns details for a specified billing plan.
Enable or Disable API Request Logging	PUT	/diagnostics/settings	Enables or disables API request logging for troubleshooting.
Get API Request Logging Settings	GET	/diagnostics/settings	Retrieves the current API request logging setting for the user and remaining log entries.
Get API Request Logging Log Files	GET	/diagnostics/request_logs	Returns a list of log entries or to download a zip file of the logs.
Get One API Request Logging Log File	GET	/diagnostics/request_logs/{requestLogId}	Returns information for a single log entry.
Clear API Request Logging Logs	DELETE	/diagnostics/request_logs	This clears the request log files.
Revoke an Authorization Token	POST	/oauth2/revoke	Revokes an authorization token
Create an Authorization Token	POST	/oauth2/token	Creates an OAuth2 authorization server token endpoint.

Login

The `/login_information` API is used to determine if a user is authenticated and to choose the account to be used for other operations. Each account associated with the login credentials is listed. The "baseUrl" parameter is used in all future API calls as the base of the request URL. This baseUrl contains the DocuSign server, the API version, and the accountId to be used for the login.

This request uses your DocuSign credentials to retrieve the account information.

URL:

https://{server}/restapi/{apiVersion}/login_information

Optional query strings: api_password={true or false}, include_account_id_guid={true or false}, login_settings={all or none}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query strings can be added to the request.

Name	Reqd?	Type	Description
api_password	No	Boolean	When set to true, shows the account API password in the response.
include_account_id_guid	No	Boolean	When set to true, shows the account ID GUID in the response.
login_settings	No	String	Can be set to all or none. When set to all, all the login settings are returned. When set to none, no login settings are returned.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/login_information

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns account information for the account associated with your DocuSign credentials, along with any optional information included in the request. The baseUrl information for future calls is included in the response.

The following example shows the header followed by the response json body.

Example Response Body

```
200 OK
Content-Length: 557
Cache-Control: private
Content-Type: application/json; charset=utf-8
Date: Tue, 06 Mar 2012 17:22:07 GMT

{
    "apiPassword":"String content",
    "loginAccounts":[{
           "accountId":"String content",
           "baseUrl":"String content",
           "email":"String content",
           "isDefault":"String content",
           "loginAccountSettings":[{
                  "name":"String content",
                  "value":"String content"
           }],
           "loginUserSettings":[{
                  "name":"String content",
                  "value":"String content"
           }],
           "name":"String content",
           "siteDescription":"String content",
           "userId":"String content",
           "userName":"String content"
    }]
}
```

Change Password

This changes the password for a user.

URL:

/login_information/password

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
currentPassword	Yes	String	The user's current password
email	Yes	String	The user's email address for the associated account.
forgottenPasswordInfo	No	String	A complex element that has up to four Question/Answer pairs for forgotten password information.
newPassword	Yes	String	The user's new password.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/login_information/password

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "currentPassword": "String content",
    "email": "String content",
    "forgottenPasswordInfo": {
           "forgottenPasswordAnswer1": "String content",
           "forgottenPasswordAnswer2": "String content",
           "forgottenPasswordAnswer3": "String content",
           "forgottenPasswordAnswer4": "String content",
           "forgottenPasswordQuestion1": "String content",
           "forgottenPasswordQuestion2": "String content",
           "forgottenPasswordQuestion3": "String content",
           "forgottenPasswordQuestion4": "String content"
    },
    "newPassword": "String content"
 }
```

Response

The response returns a success or failure.

Create Account

This creates a new account for using the DocuSign service.

URL:

/accounts

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
accountName	Yes	String	The account name for the new account.
accountSettings	No		The name/value pair information for account settings. These determine the features available for the account. The accountSettings are listed and described below.
addressInformation	No	String	A complex type that contains the following information for the new account (all string content): address1, address2, city, country, fax, phone, postalCode and state. **Note:** If country is US (United States) then State codes are validated for US States. Otherwise, State is treated as a non-validated string and serves the purpose of entering a state/province/region. The maximum characters for the strings are: - address1, address2, city, country and state: 100 characters - postalCode, phone, and fax: 20 characters
creditCardInformation	No	String	A complex type that has information about the credit card used to pay for this account. It included the elements (all string content): cardNumber, cardType, expirationMonth, expirationYear and nameOnCard
distributorCode	Yes	String	The Distributor Code that identifies the billing plan groups and plans for the new account.
distributorPassword	Yes	String	The Distributor Password for the distributorCode.
initialUser	Yes		A complex type with the initial user information for the new account. See Add User To Account for more information on the settings needed for a user.
planInformation currencyCode	Yes		This is the ISO currency code for the account.

Name	Reqd?	Type	Description
planInformation planFeatureSets	No		A complex type that sets the feature sets for the account. It contains the following information (all string content): • currencyFeatureSetPrices - This contains the currencyCode and currencySymbol for the alternate currency values for envelopeFee, fixedFee, seatFee that are configured for this plan feature set. • envelopeFee - An incremental envelope cost for plans with envelope overages (when isEnabled=true). • featureSetId - A unique ID for the feature set. • fixedFee - A one-time fee associated with the plan (when isEnabled=true). • isActive - Determines if the feature set is actively set as part of the plan. • isEnabled - Determines if the feature set is actively enabled as part of the plan. • name - The name of the feature set. • seatFee - An incremental seat cost for seat-based plans (when isEnabled=true).
planInformation planId	Yes	String	The DocuSign Plan ID for the account
planInformation freeTrialDaysOverride	No	String	Reserved for DocuSign use only.

Name	Reqd?	Type	Description
referralInformation	No		A complex type that contains the following information for entering referral and discount information. The following items are included in the referral information (all string content): enableSupport, includedSeats, saleDiscountPercent, saleDiscountAmount, saleDiscountFixedAmount, saleDiscountPeriods, saleDiscountSeatPriceOverride, planStartMonth, referralCode, referrerName, advertisementId, publisherId, shopperId, promoCode, groupMemberId, idType, and industry **Note:** saleDiscountPercent, saleDiscountAmount, saleDiscountFixedAmount, saleDiscountPeriods, and saleDiscountSeatPriceOverride are reserved for DoucSign use only.
socialAccountInformation.	No		A complex type that contains the following information: • email – The users email address. • provider – The social account provider (Facebook, Yahoo, etc.). • socialId • userName – The full user name for the account.

accountSettings:

The accountSettings set the account feature information.

Setting Name	Value	Authorization Required	Description
adoptSigConfig	Boolean	Admin	When true, the Signature Adoption Configuration page is available to account administrators.
allowAccessCodeFormat	Boolean	Admin	When true, the Access Code Format page is available to account administrators.
allowAccountManagementGranular	Boolean	Admin	When true, the Delegated Administration functionality is available to account.

allowBulkSend	Boolean	Admin	When true, the account allows bulk sending of envelopes.
allowCDWithdraw	Boolean	Admin	When true, signers can withdraw their consent to use electronic signatures.
allowConnectSendFinishLater	Boolean	Reserved	Reserved
allowDataDownload	Boolean	Admin	When true, the account can download envelope data.
allowEnvelopeCorrect	Boolean	Admin	When true, the account allows in process envelopes to be corrected.
allowEnvelopePublishReporting	Boolean	Admin	When true, the account can access the Envelope Publish section in Classic DocuSign Experience Account Administration.
allowExpressSignerCertificate	Boolean	Admin	When true, senders are allowed to use the DocuSign Express digital signatures.
allowExternalSignaturePad	Boolean	Admin	When true, an external signature pad can be used for signing. The signature pad type is set by the externalSignaturePadType property.
allowInPerson	Boolean	SysAdmin	When true, the account allows In Person Signing.
allowMarkup	Boolean	Admin	When true, the document markup feature is enabled for the account.
allowMemberTimezone	Boolean	Admin	When true, account users can set their own time zones.
allowMergeFields	Boolean	Admin	When true, the account can use merge fields in conjunction with DocuSign for Salesforce.
allowMultipleSignerAttachments	Boolean	Admin	When true, multiple signer attachments are allowed for an envelope.
allowOfflineSigning	Boolean	Admin	When true, the account can use Offline Signing and envelopes signed using offline signing on mobile devices are synchronized with this account. This option and the inSessionEnabled option must both be enabled (true) for a caller to use offline signing.

allowExpressSignerCertificate	Boolean	Admin	When true, senders are allowed to use the DocuSign Express digital signatures.
allowOpenTrustSignerCertificate	Boolean	Admin	When true, senders are allowed to use the OpenTrust digital signatures.
allowPaymentProcessing	Boolean	Admin	When true, the account can access the Payment Processing set up page.
allowPersonalSignerCertificate	Boolean	Admin	When true, the account can use personal signer certificates.
allowReminders	Boolean	Admin	When true, the reminder and expiration functionality is available when sending envelops.
allowSafeBioPharmaSignerCertificate	Boolean	Admin	When true, senders are allowed to use the SAFE BioPharma digital signatures.
allowSharedTabs	Boolean	Admin	When true, the account allows users to share custom tags (fields). Note: This setting is only shown when getting account settings. It cannot be modified.
allowSignatureStamps	Boolean	Reserved	Reserved
allowSignDocumentFromHomePage	Boolean	Admin	When true, the Sign a Document Now button is available on the Home tab.
allowSignerReassign	Boolean	Admin	When true, the account allows signers to reassign an envelope.
allowSignerReassignOverride	Boolean	Admin	When true, the sender has the option override the default account setting for reassigning recipients.
allowTabOrder	Boolean	Admin	When true, the Tab Order field is available for use when creating tabs.
allowWorkspaceComments	Boolean	Reserved	Reserved
allowWorkspaceCreate	Boolean	Admin	When true, account users can create DocuSign Rooms.
attachCompletedEnvelope	Boolean	Admin	When true, envelope documents are included as a PDF file attachment for signing completed emails.
authenticationCheck	String	Admin	Sets when authentication checks are applied for recipient

			envelope access. This setting only applies to Phone Authentication, SMS Authentication, and Knowledge-Based ID checks. There are two possible options: • **Initial Access**: The authentication check always applies the first time a recipient accesses the documents. Recipients are not asked to authenticate again when they access the documents from the same browser on the same device. If the recipient attempts to access the documents from a different browser or a different device, the recipient must pass authentication again. Once authenticated, that recipient is not challenged again on the new device or browser. The ability for a recipient to skip authentication for documents is limited to documents sent from the same sending account. • **Each Access**: Authentication checks apply each time a recipient attempts to access the envelope. However, you can configure the Authentication Expiration setting to allow recipients to skip authentication when they have recently passed authentication by setting a variable timeframe.
autoNavRule	String	Admin	The auto-navigation rule for the account. Enumeration values are: Off, RequiredFields, RequiredAndBlankFields, AllFields, PageThenRequiredFields, PageThenRequiredAndBlankFi

			elds, PageThenAllFields.
bulkSend	Boolean	Admin	When true, the account allows bulk sending of envelopes.
canSelfBrandSend	Boolean	SysAdmin	When true, account administrators can self-brand their sending console through the DocuSign Console.
canSelfBrandSign	Boolean	SysAdmin	When true, account administrators can self-brand their signing console through the DocuSign Console.
conditionalFieldsEnabled	Boolean	Admin	When true, conditional fields can be used by the account.
dataFieldRegexEnabled	Boolean	Admin	When true, the Regex field is available for tabs with that property.
dataFieldSizeEnabled	Boolean	Admin	When true, the maximum number of characters field is available for tabs with that property.
dataPopulationScope	String	Admin	This sets how data is shared for tabs with the same tabLabel. This applies to Check box, Company, Data Field, Dropdown List, Full Name, Formula, Note and Title tabs. **IMPORTANT:** Changing this setting will affect envelopes that have been sent, but not completed. There are two possible values: • **Document:** Tabs in a document with the same Label populate with the same data. • • **Envelope:** Tabs in all documents in the envelope with the same Label populate with the same data.
disableMobileSending	Boolean	Admin	When true, sending from mobile applications is disabled.
disableMultipleSessions	Boolean	Admin	When true, account users cannot be logged into multiple sessions at once.
disableUploadSignature	Boolean	Admin	When true, signers cannot use the upload signature/initials image option when signing a

			document.
documentConversionRestrictions	String	Admin	Sets the account document upload restriction for non-account administrators. There are three possible values: • **No Restrictions**: there are no restrictions on the type of documents that can be uploaded. • **Allow PDF only**: non-administrators can only upload PDF files. • **No Upload**: Non-administrators cannot upload files.
enableAutoNav	Boolean	SysAdmin or EnableAutoNavByDSAdmin is set	When true, the auto-navigation is enabled for the account.
enableCalculatedFields	Boolean	Admin & AllowExpression is set	When true, this account can use the Calculated Fields feature.
enableDSPro	Boolean	SysAdmin	When true, this account can send and manage envelopes from the DocuSign Desktop Client.
enableEnvelopeStampingByAccountAdmin	Boolean	SysAdmin or account has EnableEnvelopeStampingByDSAdmin set	When true, senders for this account can choose to have the envelope ID stamped in the document margins.
enablePaymentProcessing	Boolean	Admin & AllowPaymentProcessing is set.	When true, Payment Processing is enabled for the account.
enablePowerForm	Boolean	SysAdmin	When true, PowerForm access is enabled for the account.
enablePowerFormDirect	Boolean	Admin	When true, direct PowerForms are enabled for the account.
enableRecipientDomainValidation	Boolean	Admin	When true, validation on recipient email domains for DocuSign Access feature is enabled.
enableRequireSignOnPaper	Boolean	Admin	When true, the account can use the requireSignOnPaper option.

enableReservedDomain	Boolean	SysAdmin	When true, an account administrator can reserve web domain and users.
enableSendToAgent	Boolean	SysAdmin	When true, this account can use the Agent Recipient Type.
enableSendToIntermediary	Boolean	Admin & AllowSendToIntermediary is set	When true, this account can use the Intermediary Recipient Type.
enableSendToManage	Boolean	Admin	When true, this account can use the Editor Recipient Type.
enableSequentialSigningAPI	Boolean	SysAdmin	When true, the account can define the routing order of recipients for envelopes sent using the DocuSign API.
enableSequentialSigningUI	Boolean	SysAdmin	When true, the account can define the routing order of recipients for envelopes sent using the DocuSign console.
enableSignerAttachments	Boolean	Admin	When true, a user can request attachments from a signer.
enableSignOnPaper	Boolean	Admin	When true, a user can allow signers to use the sign on paper option.
enableSignOnPaperOverride	Boolean	Admin	When true, a user can override the default account setting for the sign on paper option.
enableSMSAuthentication	Boolean	Admin	When true, the account can use SMS authentication.
enableTransactionPoint	Boolean	SysAdmin	When true, Transaction Point is enabled for this account.
enableVaulting	Boolean	SysAdmin	When true, this account can use electronic vaulting for documents.
enableWorkspaces	Boolean	Admin	When true, DocuSign Rooms is enabled for the account.
envelopeIntegrationAllowed	String	SysAdmin	Shows the envelope integration rule for the account. Enumeration values are: NotAllowed, Full, IntegrationSendOnly.
envelopeIntegrationEnabled	Boolean	Admin & EnvelopeIntegrationAllowed is set	When true, envelope integration is enabled for the account.

envelopeStamplingDefaultValue	Boolean	(GET only)	When true, envelopes sent by this account automatically have the envelope ID stamped in the margins, unless the sender selects not to have them stamped.
externalSignaturePadType	String	Admin	Sets the type of signature pad that can be used. Enumerated values are: none, topaz, and epadWithIntegriSign.
faxOutEnabled	Boolean	Admin	When true, the account can use the fax out feature.
idCheckExpire	String	Admin	Indicates when a user's authentication expires. Enumeration values are: Always, Never, Variable. Variable uses the value in IDCheckExpireDays.
idCheckExpireDays	Integer	Admin	The number of days before a user's authentication expires. This is only active if the IDCheckExpire value is Variable.
idCheckRequired	String	Admin	Indicates if authentication is required by envelope signers. Enumeration values are: Always, Never, or Optional. Optional means the authentication is determined by the sender.
inPersonIDCheckQuestion	String	Admin	The default question used by the In Person signing host for an In Person signing session.
inSessionEnabled	Boolean	Admin	When true, the account can use In Session (embedded) and offline signing. This option and the allowOfflineSigning option must both be enabled (true) for a caller to use offline signing.
inSessionSuppressEmails	Boolean	Admin	When true, emails are not sent to the embedded recipients on an envelope for the account.
mobileSessionTimeout	String	Admin	Sets the amount of idle activity time, in minutes, before a DocuSign mobile application user is automatically logged off of the system. If the setting is 0, then DocuSign mobile

			application users are never automatically logged off the system. The minimum setting is 1 minute and the maximum setting is 120 minutes. Note this setting only applies to the DocuSign for iOS v2.8.2 or later mobile app.
phoneAuthRecipientMayProvidePhoneNumber	Boolean	Admin	When true, senders can select to allow the recipient to provide a phone number for the Phone Authentication process.
pkiSignDownloadedPDFDocs	String	Admin	The policy for adding a digital certificate to downloaded, printed and emailed documents. Enumeration values are: NoSign, NoSignAllowUserOverride, YesSign
recipientsCanSignOffline	Boolean	Admin	When true, envelopes signed using offline signing on mobile devices are synchronized with this account.
requireDeclineReason	Boolean	Admin	When true, recipients that decline to sign an envelope must provide a reason.
requireSignerCertificateType	String	Admin	Sets which Digital Signature certificate is required when sending envelopes. There are three possible values: • **None:** a Digital Signature certificate is not required. • **DocuSign Express**: signers must use a DocuSign Express certificate. • **OpenTrust**: signers must use an OpenTrust certificate.
rsaVeridAccountName	String	Admin	The RSA account name. **Note:** Modifying this value might inadvertently disrupt your ID Check capability. Ensure you have the correct value before changing this.

rsaVeridPassword	String	Admin	The password used with the RSA account. **Note:** Modifying this value might inadvertently disrupt your ID Check capability. Ensure you have the correct value before changing this.
rsaVeridRuleset	String	Admin	The RSA rule set used with the account. **Note:** Modifying this value might inadvertently disrupt your ID Check capability. Ensure you have the correct value before changing this.
rsaVeridUserId	String	Admin	The user ID for the RSA account. **Note:** Modifying this value might inadvertently disrupt your ID Check capability. Ensure you have the correct value before changing this.
savingCustomTabsEnabled	Boolean	Admin	When true, account users can save custom tabs.
selfSignedRecipientEmail Document	String	Admin	This sets the account setting for how self-signed documents are presented to the email recipients. Enumeration values are: • include_pdf: With this setting a PDF of the completed document is attached to the email • include_link: With this setting a secure link to the self-signed documents is included in the email.
selfSignedRecipientEmailDocume ntRights	Boolean	Admin	When true, account administrators can set the selfSignedRecipientEmailDocu ment option.
selfSignedRecipientEmail DocumentUserOverride	Boolean	Admin	When true the selfSigned RecipientEmailDocument userSetting can be set for an individual user. The userSetting will override the account setting.

selfSignedRecipientEmailDocumentUserOverrideRights	Boolean	Admin	When true, account administrators can set the selfSignedRecipientEmailDocumentOverride option.
senderMustAuthenticateSigning	Boolean	Admin	When true, a sender that is also a recipient of an envelope must follow the authentication requirements for the envelope.
sendToCertifiedDeliveryEnabled	Boolean	Admin	When true, the Certified Deliveries Recipient type can be used by the account.
sessionTimeout	Integer	Admin	The amount of idle activity time, in minutes, before a user is automatically logged off of the system. The minimum setting is 20 minutes and the maximum setting is 120 minutes.
setRecipEmailLang	Boolean	Admin	When true, senders can set the email languages for each recipient.
setRecipSignLang	Boolean	Admin	When true, senders can set the signing language used for each recipient.
sharedCustomTabsEnabled	Boolean	Admin	When true, saved custom tabs can be shared with account users.
signDateFormat	String	Admin	The date/time format applied to Date Signed fields for the account.
signerAttachCertificateToEnvelopePDF	Boolean	AccountAdmin & account is selected in AccountSigningSettings	When true, the Certificate of Completion is included in the envelope documents PDF when it is downloaded.
signerAttachConcat	Boolean	Admin	When true, signer attachments are added to the parent document that the attachment tab is located on, instead of the default behavior that creates a new document in the envelope for every signer attachment.
signerCanCreateAccount	Boolean	AccountAdmin & account is selected in AccountSigningSettings	When true, the signer without a DocuSign account can create a DocuSign account after signing.
signerCanSignOnMobile	Boolean	AccountAdmin	When true, signer can use the

		& account is selected in AccountSigningSettings	DocuSign mobile signing user interface.
signerInSessionUseEnvelopeCompleteEmail	Boolean	Admin	When true, an envelope complete email is sent to an In Session (embedded) or offline signer after DocuSign processes the envelope.
signerLoginRequirements	String	Admin	Sets the Login requirements for the signer. **Important:** If you use Direct PowerForms or captive/embedded signers, the "**Account required** …" settings are bypassed for those signers. If your workflow requires that the signer have an account, you should not use those methods. There are four options: • **Not required to login**: The signer is not required to log on to the system. • **Login required if signer has an account**: If the signer has a DocuSign account, they must log on to sign the document. • **Account required – login to initiate session**: The sender cannot send an envelope to anyone who does not have a DocuSign account. • • **Account required – login for each envelope**: The sender cannot send an envelope to anyone who does not have a DocuSign account and the signer must log on the system for each envelope they will sign.
signerMustHaveAccount	Boolean	AccountAdmin & account is selected in AccountSigningSettings	When true, senders can only send an envelope to a recipient that has a DocuSign account.

signerMustLoginToSign	Boolean	AccountAdmin & account is selected in AccountSigningSettings	When true, an envelope signer must log in to the DocuSign console to sign an envelope.
signerShowSecureFieldInitialValues	Boolean	AccountAdmin & account is selected in AccountSigningSettings	When true, the initial value of all SecureFields is written to the document when sent.
signTimeShowAmPm	Boolean	Admin	When true, AM or PM is shown as part of the time for signDateFormat.
tabDataLabelEnabled	Boolean	Admin	When true, senders can change the default tabLabel for tabs.
tabLockingEnabled	Boolean	Admin	When true, the locked option is available for tabs with that property.
tabTextFormattingEnabled	Boolean	Admin	When true, the formatting options (font type, font size, font color, bold, italic, and underline) are available for tabs with those properties.
universalSignatureOptIn	Boolean	Reserved	Reserved
universalSignatureOptOut	Boolean	Reserved	Reserved
useAccountLevelEmail	Boolean	AccountAdmin & account is selected in AccountSigningSettings	When true, the content of notification emails is determined at the account level.
usesAPI	Boolean	SysAdmin	When true, the account can use the DocuSign API. Not needed if an Integrator Key is used.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

{
    "accountName":"String content",
    "accountSettings":[{
           "name":"String content",
           "value":"String content"
```

```
    }],
    "addressInformation":{
           "address1":"String content",
           "address2":"String content",
           "city":"String content",
           "country":"String content",
           "fax":"String content",
           "phone":"String content",
           "postalCode":"String content",
           "state":"String content"
    },
    "creditCardInformation":{
           "cardNumber":"String content",
           "cardType":"String content",
           "expirationMonth":"String content",
           "expirationYear":"String content",
           "nameOnCard":"String content"
    },
    "distributorCode":"String content",
    "distributorPassword":"String content",
    "initialUser":{
           "activationAccessCode":"String content",
           "email":"String content",
           "enableConnectForUser":"String content",
           "firstName":"String content",
           "forgottenPasswordInfo":{
                  "forgottenPasswordAnswer1":"String content",
                  "forgottenPasswordAnswer2":"String content",
                  "forgottenPasswordAnswer3":"String content",
                  "forgottenPasswordAnswer4":"String content",
                  "forgottenPasswordQuestion1":"String content",
                  "forgottenPasswordQuestion2":"String content",
                  "forgottenPasswordQuestion3":"String content",
                  "forgottenPasswordQuestion4":"String content"
           },
           "groupList": [
                  {
                         "groupId": "string content",
                         "groupName": "string content",
                         "permissionProfileId": "string content",
                         "groupType": "string content"
                  },
                  {
                         "groupId": "string content",
                         "groupName": "string content",
                         "permissionProfileId": "string content",
                         "groupType": "string content"
                  }
           ],
           "lastName":"String content",
           "middleName":"String content",
           "password":"String content",
           "sendActivationOnInvalidLogin":"String content",
           "suffixName":"String content",
           "title":"String content",
           "userName":"String content",
           "userSettings":[{
                  "name":"String content",
                  "value":"String content"
           }]
    },
    "planInformation":{
           "currencyCode":"String content",
           "planFeatureSets":[{
```

```
                    "currencyFeatureSetPrices":[{
                           "currencyCode":"String content",
                           "currencySymbol":"String content",
                           "envelopeFee":"String content",
                           "fixedFee":"String content",
                           "seatFee":"String content"
                    }],
                    "envelopeFee":"String content",
                    "featureSetId":"String content",
                    "fixedFee":"String content",
                    "isActive":"String content",
                    "isEnabled":"String content",
                    "name":"String content",
                    "seatFee":"String content"
             }],
             "planId":"String content",
             "freeTrialDaysOverride":"String content",
     },
     "referralInformation":{
             "advertisementId":"String content",
             "enableSupport":"String content",
             "groupMemberId":"String content",
             "idType":"String content",
             "includedSeats":"String content",
             "industry":"String content",
             "planStartMonth":"String content",
             "promoCode":"String content",
             "publisherId":"String content",
             "referralCode":"String content",
             "referrerName":"String content",
             "saleDiscountPercent":"String content",
             "shopperId":"String content"
     },
     "socialAccountInformation":{
             "email":"String content",
             "provider":"String content",
             "socialId":"String content",
             "userName":"String content"
     }
}
```

Response

The response returns the new account ID, password and the default user information.

The following example shows the response json body.

Example Response Body

```
{
    "accountId":"String content",
    "apiPassword":"String content",
    "baseUrl":"String content",
    "userId":"String content"
}
```

Create Multiple Accounts

This is used to create multiple DocuSign accounts with one call. It uses the same information and formats as the normal Create Account call with the information included within a newAccountRequests element. A maximum of 100 new accounts can be created at one time.

URL:

/accounts

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

See the Create Account parameters for the information needed for creating an account.

Example JSON Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

{
  "newAccountRequests": [
    {
      (Account details ommitted for brevity)
    },
    {
      (Account details ommitted for brevity)
    }
  ]
}
```

Example XML Request Body

Note that the structure of the XML request is slightly different than the json request, in that the new account information is included in a newAccountDefinition element inside the newAccountRequests element.

```
<newAccountsDefinition xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <newAccountRequests>
    <newAccountDefinition>
      (Account details ommitted for brevity)
    </newAccountDefinition>
    <newAccountDefinition>
      (Account details ommitted for brevity)
    </newAccountDefinition>
  </newAccountRequests>
</newAccountsDefinition>
```

Response

The response returns the new account ID, password and the default user information for each newly created account.

> **Note:** A 201 code is returned if the call succeeded, but some of the individual account requests failed. In this case an errorDetails node is added to the new account element that failed to create a new account.

The following examples show the response json and xml bodies.

Example JSON Response Body

```
{
  "newAccounts": [
    {
      "accountId": "string",
      "userId": "string",
      "apiPassword": "string",
      "baseUrl": "string",
      "accountIdGuid": "string"
    },
    {
      "accountId": "string",
      "userId": "string",
      "apiPassword": "string",
      "baseUrl": "string",
      "accountIdGuid": "string"
    }
  ]
}
```

Example XML Response Body

```
<newAccountsSummary xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <newAccounts>
    <newAccountSummary>
      <accountId>string</accountId>
      <accountIdGuid>string</accountIdGuid>
      <apiPassword>string</apiPassword>
      <baseUrl>string</baseUrl>
      <userId>string</userId>
    </newAccountSummary>
    <newAccountSummary>
      <accountId>string</accountId>
      <accountIdGuid>string</accountIdGuid>
      <apiPassword>string</apiPassword>
      <baseUrl>string</baseUrl>
      <userId>string</userId>
    </newAccountSummary>
  </newAccounts>
</newAccountsSummary>
```

Delete Account

This deletes the specified account.

URL:

/accounts/{accountId}

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No parameters are required, only the specified account ID.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response body only returns a success or failure.

Get Account Information

This returns the account information for the specified account.

URL:

/accounts/{accountId}/

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No parameters are required, only the specified account ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}
```

```
X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the account information for the requested account.

The canUpgrade information is a true/false setting that shows if the account can be upgraded through the API.

The following example shows the response json body.

Example Response Body

```
{
    "accountName":"String content",
    "billingPeriodEndDate":"String content",
    "billingPeriodEnvelopesAllowed":"String content",
    "billingPeriodEnvelopesSent":"String content",
    "billingPeriodStartDate":"String content",
    "canUpgrade": "String content",
    "connectPermission":"String content",
    "currencyCode":"String content",
    "currentPlanId":"String content",
    "distributorCode":"String content",
    "docuSignLandingUrl":"String content",
    "forgottenPasswordQuestionCount":"String content",
    "planEndDate":"String content",
    "planName":"String content",
    "planStartDate":"String content",
    "suspensionDate":"String content",
    "suspensionStatus":"String content"
}
```

Get Account Billing Plan

This returns the billing plan information for the specified account, including the current billing plan, successor plans, billing address, and billing credit card.

URL:

/accounts/{accountId}/billing_plan

Optional query strings: include_metadata={true or false}, include_successor_plans={true or false}, include_credit_card_information={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters.

If the optional query include_metadata is true, then the canUpgrade and renewalStatus information is included the response and an array of supportedCountries is added to the billingAddress information.

By default the successor plan and credit card information is included in the response. This information can be excluded from the response by adding the appropriate optional query string with the setting = false.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/billing_plan

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the billing plan information, including the currency code, for the plan. The billingPlan and succesorPlans parameters are the same as those shown in the Get Billing Plan Details reference, the billingAddress and creditCardInformation parameters are the same as those shown in the Update Billing Plan reference.

Note: When credit card number information is shown, a mask is applied to the response so that only the last 4 digits of the card number are visible.

Name	Type	Description
billingAddressIsCreditCardAddress	Boolean	If true, the credit card address information is the same as that returned as the billing address. If false, then the billing address is considered a billing contact address, and the credit card address can be different.

The following example shows the response json body. This example includes the upgrade/renewal information, supportedCountries array (include_metadata=true), and the successor plans and credit card information is not excluded.

Example Response Body

```
{
    "billingPlan": {
           "planId": "string",
           "planName": "string",
           "paymentCycle": "string",
           "paymentMethod": "string",
           "perSeatPrice": "string",
           "otherDiscountPercent": "string",
           "supportIncidentFee": "string",
           "supportPlanFee":"string",
           "includedSeats": "string",
           "enableSupport": "string",
           "currencyCode": "string",
           "canUpgrade": "string",
```

```
            "renewalStatus": "string",
            "seatDiscounts": [
                    {
                            "beginSeatCount": "string",
                            "endSeatCount": "string",
                            "discountPercent": "string"
                    }
            ],
            "planFeatureSets": [
                    {
                            "featureSetId": "string",
                            "isActive": "string",
                            "name": "string",
                            "fixedFee": "string",
                            "envelopeFee": "string",
                            "seatFee": "string",
                            "isEnabled": "string",
                            "currencyFeatureSetPrices": [
                                    {
                                            "currencyCode": "string",
                                            "fixedFee": "string",
                                            "seatFee": "string",
                                            "envelopeFee": "string",
                                            "currencySymbol": "string"
                                    }
                            ]
                    }
            ]
    },
    "successorPlans": [
            {
                    "planName": "string",
                    "paymentCycle": "string",
                    "paymentMethod": "string",
                    "perSeatPrice": "string",
                    "otherDiscountPercent": "string",
                    "supportIncidentFee": "string",
                    "supportPlanFee": "string",
                    "includedSeats": "string",
                    "enableSupport": "string",
                    "planId": "string",
                    "seatDiscounts": [
                            {
                                    "beginSeatCount": "string",
                                    "endSeatCount": "string",
                                    "discountPercent": "string"
                            }
                    ],
                    "planFeatureSets": [
                            {
                                    "featureSetId": "string",
                                    "isActive": "string",
                                    "name": "string",
                                    "fixedFee": "string",
                                    "envelopeFee": "string"
                                    "seatFee": "string",
                                    "isEnabled": "string",
                                    "currencyFeatureSetPrices": [
                                            {
                                                    "currencyCode": "string",
                                                    "fixedFee": "string",
                                                    "seatFee": "string",
                                                    "envelopeFee": "string",
                                                    "currencySymbol": "string"
```

```
                                        }
                                ]
                        }
                ],
                "currencyPlanPrices": [
                        {
                                "currencyCode": "string",
                                "perSeatPrice": "string",
                                "supportIncidentFee": "string",
                                "supportPlanFee": "string",
                                "currencySymbol": "string",
                                "supportedCardTypes": {
                                        "cardTypes": [
                                                "string",
                                                "string"
                                        ]
                                }
                        }
                ]
        }
    ],
    "billingAddress": {
        "address1": "string",
        "address2": "string",
        "city": "string",
        "state": "string",
        "postalCode": "string",
        "phone": "string",
        "fax": "string",
        "country": "string",
        "firstName": "string",
        "lastName": "string",
        "email": "string",
        "supportedCountries": [
                {
                        "isoCode": "string",
                        "provinceValidated": "string",
                        "name": "string",
                        "provinces": [
                                {
                                        "isoCode": "string",
                                        "name": "string"
                                },
                                {
                                        "isoCode": "string",
                                        "name": "string"
                                }
                        ]
                },
                {
                        "isoCode": "string",
                        "provinceValidated": "string",
                        "name": "string",
                        "provinces": [
                                {
                                        "isoCode": "string",
                                        "name": "string"
                                },
                                {
                                        "isoCode": "string",
                                        "name": "string"
                                }
                        ]
                }
```

```
            ]
    },
    "billingAddressIsCreditCardAddress": "string",
    "creditCardInformation": {
            "cardNumber": "string",
            "expirationMonth": "string",
            "expirationYear": "string",
            "nameOnCard": "string",
            "cardType": "string",
            "address": {
                    "street1": "string",
                    "street2": "string",
                    "city": "string",
                    "state": "string",
                    "zip": "string",
                    "zipPlus4": "string",
                    "phone": "string",
                    "fax": "string",
                    "country": "string"
            }
    }
}
```

Update Account Billing Plan

This updates the billing plan information, billing address, and credit card information for the specified account.

URL:

/accounts/{accountId}/billing_plan

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The billingAddressIsCreditCardAddress parameter (found when Getting Account Billing Plan information) determines how the billing address and creditCardInformation address are updated.

- If billingAddressIsCreditCardAddress is ‘true’ then either the billingAddress values or the creditCardInformation address values can be used to update the single address used as billing and credit card address. If the PUT updates both billing and credit card addresses, then the address field values must exactly match or an error is returned. DocuSign recommends that only one of the addresses is updated. .
- If billingAddressIsCreditCardAddress is ‘false’ then the billingAddress is a billing contact address and the credit card address is the current credit card address for billing and can be updated separately.

When updating creditCardInformation, all of the creditCardInformation must be included in the update (including the cardNumber, nameOnCard, expirationYear, expirationMonth, and address), even if only

part of the information is being changed. The only exception is that the expirationMonth and expirationYear may be updated without updating the other information.

Note: When updating account billing plan information to upgrade a plan if creditCardInformation is not included, the system will use the existing credit card information for any charges.

The referralInformation is included so that promotional codes (promoCode) can be used when upgrading plans.

Name	Reqd?	Type	Description
billingAddress	No	String	A complex type that contains the following address information for the account: • address1 – The first address line for the billing address. • address2 – An additional address line for the billing address. • city – The city for the billing address. • state – The state for the billing address, see note below. • postalCode – The postal code for the billing address. • phone – The telephone number for the billing address. • fax – The fax number for the billing address. • country – The country code for the billing address, see note below. • firstName – The first name of a contact person associated with the billing address. • lastName – The last name of a contact person associated with the billing address. • email – The email address associated with the billing address. **Note:** If country is US (United States) then State codes are validated for US States. Otherwise, State is treated as a non-validated string and serves the purpose of entering a state/province/region.

Name	Reqd?	Type	Description
creditCardInformation	No	String	A complex type that has information about the credit card used to pay for this account. It included the elements: • cardNumber – The credit card number. Note that in responses only the last four digits are shown. • expirationMonth – The expiration month shown on the credit card. • expirationYear – The expiration year shown on the credit card. • nameOnCard – The name listed on the credit card. • cardType – The credit card type can be: visa, mastercard, or amex. address – A complex element with the credit card billing address information. This can be the same as billing address and follows the same rules as billingAddress. It contains the following elements: street1, street2, city, state, zip, zipPlus4, phone, fax, and country.
enableSupport	No	Boolean	If true, the plan has support enabled.
includeSeats	No	String	The number of seats included in the plan
saleDiscountPercent, saleDiscountAmount, saleDiscountFixedAmount, saleDiscountPeriods, saleDiscountSeatPriceOverride	No	String	These elements are reserved for DoucSign use only.
renewalStatus	No	String	Sets the renewal status for the account. The acceptable values are: • auto: The account automatically renews. • queued_for_close: Account will be closed at the billingPeriodEndDate. • queued_for_downgrade: Account will be downgraded at the billingPeriodEndDate.
downgradeReason	No	String	An optional element that has the reason for downgrade of account.
planInformation currencyCode	Yes		This is the ISO currency code for the account.

Name	Reqd?	Type	Description
planInformation planFeatureSets	See Desc.		A complex type that sets the feature sets for the account. It contains the following information (all string content): • currencyFeatureSetPrices - This contains the currencyCode and currencySymbol for the alternate currency values for envelopeFee, fixedFee, seatFee that are configured for this plan feature set. • envelopeFee - An incremental envelope cost for plans with envelope overages (when isEnabled=true). • featureSetId - A unique ID for the feature set. • fixedFee - A one-time fee associated with the plan (when isEnabled=true). • isActive - Determines if the feature set is actively set as part of the plan. • isEnabled - Determines if the feature set is actively enabled as part of the plan. • name - The name of the feature set. • seatFee - An incremental seat cost for seat-based plans (when isEnabled=true).
planInformation planId	No	String	The plan ID for the account. It uniquely identifies a plan and is used to set plans in other functions.
planInformation freeTrialDaysOverride	No	String	Reserved for DocuSign use only.
referralInformation	No		A complex type that contains the following information for entering referral and discount information. The following items are included in the referral information (all string content): referralCode, referrerName, advertisementId, publisherId, shopperId, promoCode, groupMemberId, idType, and industry.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/billing_plan

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

{
    "billingAddress":{
           "address1": "string",
           "address2": "string",
           "city": "string",
           "state": "string",
           "postalCode": "string",
           "phone": "string",
           "fax": "string",
           "country": "string",
           "firstName": "string",
           "lastName": "string",
           "email": "string "
           }
    },
    "creditCardInformation":{
           "cardNumber": "string",
           "expirationMonth": "string",
           "expirationYear": "string",
           "nameOnCard": "string",
           "cardType": "string",
           "address": {
                  "street1": "string",
                  "street2": "string",
                  "city": "string",
                  "state": "string",
                  "zip": "string",
                  "zipPlus4": "string",
                  "phone": "string",
                  "fax": "string",
                  "country": "string"
           }
    },
    "enableSupport":"String content",
    "includedSeats":"String content",
    "renewalStatus": "string",
    "downgradeReason": "string",
    "planInformation":{
           "currencyCode":"String content",
           "planFeatureSets":[{
                  "currencyFeatureSetPrices":[{
                         "currencyCode":"String content",
                         "currencySymbol":"String content",
                         "envelopeFee":"String content",
                         "fixedFee":"String content",
                         "seatFee":"String content"
                  }],
                  "envelopeFee":"String content",
                  "featureSetId":"String content",
                  "fixedFee":"String content",
                  "isActive":"String content",
                  "isEnabled":"String content",
                  "name":"String content",
                  "seatFee":"String content"
           }],
           "planId":"String content",
```

```
            "freeTrialDaysOverride":"String content"
    },
    "referralInformation": {
            "referralCode": "string",
            "referrerName": "string ",
            "advertisementId": "string",
            "publisherId": "string",
            "shopperId": "string",
            "promoCode": "string",
            "groupMemberId": "string",
            "idType": "string",
            "industry": "string"
    }
}
```

Response

The response returns a success or failure.

Get Billing Payment information

This returns information about one or more payments. If the from date or to date queries or payment ID are not used, the response returns payment information for the last 365 days. This call can only be used by users with account administrator privileges.

URL:

/accounts/{accountId}/billing_payments

Optional query parameters: from_date={dateTime}, to_date={dateTime}

OR

/accounts/{accountId}/billing_payments/{paymentId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No parameters are required, but the payment ID can be added to the URL to get information about a single payment and the following optional query parameters can be added.

Name	Reqd?	Type	Description
from_date	No	dateTime	The date/time setting that specifies the date/time when the request begins checking for payments made for the account.

Name	Reqd?	Type	Description
to_date	No	dateTime	The date/time setting that specifies the date/time when the request stops checking for payments made for the account.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/billing_payments

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns information for a single payment, if a payment ID was used in the URL, or a list of payments. If the from date or to date queries or payment ID are not used, the response returns payment information for the last 365 days. If the request was for a single payment ID, the nextUri and previousUri elements are not returned.

The following example shows the response json body.

Example Response Body

```
{
    "billingPayments": [
            {
                    "paymentId": "string",
                    "amount": "string",
                    "paymentDate": "string",
                    "paymentNumber": "string",
                    "description": "string"
            },
            {
                    "paymentId": "string",
                    "amount": "string",
                    "paymentDate": "string",
                    "paymentNumber": "string",
                    "description": "string"
            }
    ],
    "nextUri": "string",
    "previousUri": "string"
}
```

Post a Billing Payment

This allows users with account administrator privileges to pay a past due invoice. This call can only be used by users with account administrator privileges.

Note: This can only be used if the paymentAllowed value for a past due invoice is true. This can be determined using Get Past Due Invoices.

URL:

/accounts/{accountId}/billing_payments

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
paymentAmount	Yes	String	The payment amount for the past due invoices. This value must match the pastDueBalance value retrieved using Get Past Due Invoices.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/billing_payments

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "paymentAmount": "string"
}
```

Response

The response returns invoice and payment information for the past due invoices.

The following example shows the response json body.

Example Response Body

```
{
    "billingPayments": [
            {
                    "invoiceId": "string",
                    "paymentId": "string",
                    "amount": "string"
            },
            {
                    "invoiceId": "string",
                    "paymentId": "string",
                    "amount": "string"
            }
    ]
}
```

Get a List of Billing Invoices

This returns a list of invoices for the account. This call can only be used by users with account administrator privileges. If the from date or to date queries are not used, the response returns invoices for the last 365 days.

URL:

/accounts/{accountId}/billing_invoices

Optional query parameters: from_date={dateTime}, to_date={dateTime}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No parameters are required, but the following optional query parameters can be added.

Name	Reqd?	Type	Description
from_date	No	dateTime	The date/time setting that specifies the date/time when the request begins checking for invoices for the account.
to_date	No	dateTime	The date/time setting that specifies the date/time when the request stops checking for invoices for the account.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/billing_invoices

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of invoices for the query timeframe. If the from date or to date queries are not used, the response returns invoices for the last 365 days.

The following example shows the response json body.

Example Response Body

```
{
    "billingInvoices": [
            {
```

```
                "amount": "string",
                "taxableAmount": "string",
                "nonTaxableAmount": "string",
                "balance": "string",
                "dueDate": "string",
                "invoiceId": "string",
                "invoiceNumber": "string",
                "pdfAvailable": "string",
                "invoiceUri": "string",
                "invoiceItems": [
                        {
                                "chargeAmount": "string",
                                "chargeName": "string",
                                "invoiceItemId": "string",
                                "quantity": "string",
                                "unitPrice": "string"
                        },
                        {
                                "chargeAmount": "string",
                                "chargeName": "string",
                                "invoiceItemId": "string",
                                "quantity": "string",
                                "unitPrice": "string"
                        }
                ]
        },
        {
                "amount": "string",
                "taxableAmount": "string",
                "nonTaxableAmount": "string",
                "balance": "string",
                "dueDate": "string",
                "invoiceId": "string",
                "invoiceNumber": "string",
                "pdfAvailable": "string",
                "invoiceUri": "string",
                "invoiceItems": [
                        {
                                "chargeAmount": "string",
                                "chargeName": "string",
                                "invoiceItemId": "string",
                                "quantity": "string",
                                "unitPrice": "string"
                        },
                        {
                                "chargeAmount": "string",
                                "chargeName": "string",
                                "invoiceItemId": "string",
                                "quantity": "string",
                                "unitPrice": "string"
                        }
                ]
        },
    ],
    "nextUri": "string",
    "previousUri": "string"
}
```

Get a Billing Invoice

This returns the requested invoice. This call can only be used by users with account administrator privileges.

URL:

/accounts/{accountId}/billing_invoices/{invoiceId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/billing_invoices/
{invoiceId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the requested invoice information.

Note: If pdfAvailable in the response is true, a PDF version of this invoice can be downloaded. To get the PDF, make the call again and change “Accept:” in the header to “Accept: application/pdf”

The following example shows the response json body.

Example Response Body

```
{
    "amount": "string",
    "taxableAmount": "string",
    "nonTaxableAmount": "string",
    "balance": "string",
    "dueDate": "string",
    "invoiceId": "string",
    "invoiceNumber": "string",
    "pdfAvailable": "string",
    "invoiceUri": "string",
    "invoiceItems": [
           {
                  "chargeAmount": "string",
                  "chargeName": "string",
                  "invoiceItemId": "string",
                  "quantity": "string",
                  "unitPrice": "string"
           },
           {
                  "chargeAmount": "string",
                  "chargeName": "string",
```

```
                    "invoiceItemId": "string",
                    "quantity": "string",
                    "unitPrice": "string"
            }
    ]
}
```

Get Past Due Invoices

This returns any past due invoices for the account and notes if payment can be made through the REST API. This call can only be used by users with account administrator privileges.

URL:

/accounts/{accountId}/billing_invoices_past_due

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/
billing_invoices_past_due

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the past due invoices, the past due balance, and if the payment can be made using Post a Billing Payment.

The following example shows the response json body.

Example Response Body

```
{
    "pastDueBalance": "string",
    "paymentAllowed": "string",
    "billingInvoices": [
            {
                    "amount": "string",
                    "taxableAmount": "string",
                    "nonTaxableAmount": "string",
```

```
                    "balance": "string",
                    "dueDate": "string",
                    "invoiceId": "string",
                    "invoiceNumber": "string",
                    "pdfAvailable": "string",
                    "invoiceUri": "string",
                    "invoiceItems": [
                           {
                                  "chargeAmount": "string",
                                  "chargeName": "string",
                                  "invoiceItemId": "string",
                                  "quantity": "string",
                                  "unitPrice": "string"
                           },
                           {
                                  "chargeAmount": "string",
                                  "chargeName": "string",
                                  "invoiceItemId": "string",
                                  "quantity": "string",
                                  "unitPrice": "string"
                           }
                    ]
            },
            {
                    "amount": "string",
                    "taxableAmount": "string",
                    "nonTaxableAmount": "string",
                    "balance": "string",
                    "dueDate": "string",
                    "invoiceId": "string",
                    "invoiceNumber": "string",
                    "pdfAvailable": "string",
                    "invoiceUri": "string",
                    "invoiceItems": [
                           {
                                  "chargeAmount": "string",
                                  "chargeName": "string",
                                  "invoiceItemId": "string",
                                  "quantity": "string",
                                  "unitPrice": "string"
                           },
                           {
                                  "chargeAmount": "string",
                                  "chargeName": "string",
                                  "invoiceItemId": "string",
                                  "quantity": "string",
                                  "unitPrice": "string"
                           }
                    ]
            }
    ]
}
```

Get Billing Charges

This returns the list of recurring and usage charges for the account. This can be used to determine the charge structure and usage of charge plan items. This call can only be used by users with account administrator privileges.

URL:

/accounts/{accountId}/billing_charges

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/billing_charges

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of charges and information about the charges. Quantities are usually shown as 'unlimited' or an integer. Amounts are shown in the currency set for the account.

The following table provides a description of the different chargeName values. The information will grow as more chargeable items are added to the system.

chargeName	Description
id_check	ID Check charge
in_person_signing	In Person Signing charge
envelopes	Included Sent Envelopes for the account
age_verify	Age verification check
ofac	OFAC Check
id_confirm	ID confirmation check
student_authentication	STAN PIN authentication check
wet_sign_fax	Pages for returning signed documents by fax
attachment_fax	Pages for returning attachments by fax

chargeName	Description
phone_authentication	Phone authentication charge
powerforms	PowerForm envelopes sent
signer_payments	Payment processing charge
outbound_fax	Send by fax charge
bulk_recipient_envelopes	Bulk Recipient Envelopes sent
sms_authentications	SMS authentication charge
saml_authentications	SAML authentication charge
express_signer_certificate	DocuSign Express Certificate charge
personal_signer_certificate	Personal Signer Certificate charge
safe_certificate	SAFE BioPharma Signer Certificate charge
seats	Included active seats charge
open_trust_certificate	OpenTrust Signer Certificate charge

The following example shows the response json body.

Example Response Body

```
{
  "billingChargeItems": [
    {
      "chargeName": "string",
      "chargeType": "string",
      "chargeUnitOfMeasure": "string",
      "allowedQuantity": "string",
      "usedQuantity": "string",
      "includedQuantity": "string",
      "firstEffectiveDate": "string",
      "lastEffectiveDate": "string",
      "unitPrice": "string",
      "blocked": "string",
      "prices": [
        {
          "unitPrice": "string",
          "beginQuantity": "string",
          "endQuantity": "string"
        },
        {
          "unitPrice": "string",
          "beginQuantity": "string",
          "endQuantity": "string"
        }
      ],
      "discounts": [
```

```
        {
          "endQuantity": "string",
          "beginQuantity": "string",
          "discount": "string"
        },
        {
          "endQuantity": "string",
          "beginQuantity": "string",
          "discount": "string"
        }
      ]
    },
    {
      "chargeName": "string",
      "chargeType": "string",
      "chargeUnitOfMeasure": "string",
      "allowedQuantity": "string",
      "usedQuantity": "string",
      "includedQuantity": "string",
      "firstEffectiveDate": "string",
      "lastEffectiveDate": "string",
      "unitPrice": "string",
      "blocked": "string",
      "prices": [
        {
          "unitPrice": "string",
          "beginQuantity": "string",
          "endQuantity": "string"
        },
        {
          "unitPrice": "string",
          "beginQuantity": "string",
          "endQuantity": "string"
        }
      ],
      "discounts": [
        {
          "endQuantity": "string",
          "beginQuantity": "string",
          "discount": "string"
        },
        {
          "endQuantity": "string",
          "beginQuantity": "string",
          "discount": "string"
        }
      ]
    }
  ]
}
```


Purchase Additional Envelopes

IMPORTANT: At this time, this operation is limited to DocuSign internal use only.

This completes the purchase of envelopes for your account. The actual purchase is done as part of an internal workflow interaction with an envelope vendor.

URL:

/accounts/{accountId}/billing_plan/purchased_envelopes

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
amount	Yes	String	The total amount of the purchase.
appName	No	String	The AppName of the client application.
currencyCode	No	String	The CurrencyCode of the purchase. This is based on the ISO 4217 currency code information.
platform	No	String	The Platform of the client application
productId	No	String	The Product ID from the AppStore.
quantity	Yes	String	The quantity of envelopes to add to the account.
receiptData	No	String	The encrypted Base64 encoded receipt data.
storeName	No	String	The name of the AppStore.
transactionId	No	String	The Transaction ID from the AppStore.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/billing_plan/
purchased_envelopes

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

{
    "amount":"String content",
    "appName":"String content",
    "currencyCode":"String content",
    "platform":"String content",
    "productId":"String content",
    "quantity":"String content",
    "receiptData":"String content",
    "storeName":"String content",
    "transactionId":"String content"
}
```

Response

The response returns a success or failure.

Get Brand Profile Information

This returns a list of brand profiles associated with the account and the default brand profiles. The Account Branding feature (accountSettings “canSelfBrandSend” and “canSelfBrandSend” are true) must be enabled for the account to use this.

URL:

/accounts/{accountId}/brands

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/brands

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of brand profiles associated with the account and the default sending and signing brand profile.

The following example shows the response json body.

Example Response Body

```
{
    "brands":[
            {
                    "brandCompany":"String content"
                    "brandId":"String content",
                    "brandName":"String content"
            },
            {
                    "brandCompany":"String content"
                    "brandId":"String content",
                    "brandName":"String content"
            }
    ]
```

```
    "recipientBrandIdDefault":"String content",
    "senderBrandIdDefault":"String content",
}
```

Upload Brand Profiles

This is used to upload one or more brand profile files for the account. The Account Branding feature (accountSettings “canSelfBrandSend” and “canSelfBrandSign” are true) must be enabled for the account to use this.

When uploading brand profile files, if the brandId for a brand profile already exists for the account, an error is returned. If you want to upload a new version of an existing brand profile, you should delete the profile and then upload the newer version.

When brand profile files are being uploaded, they must be combined into one zip file and the Content-Type must be application/zip.

URL:

/accounts/{accountId}/brands

Formats:

Request is application/zip and Response is XML or JSON

HTTP Method:

POST

Parameters:

No additional parameters are required.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/brands

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/zip

<base64bytes brand zip file removed for brevity>
```

Response

The response returns if the upload is successful and the default brand profile for the account and a list of brand profiles added to the account.

The following example shows the response json body.

Example Response Body

```
{
    "brands":[
            {
                    "brandCompany":"String content"
                    "brandId":"String content",
                    "brandName":"String content"
            },
            {
                    "brandCompany":"String content"
                    "brandId":"String content",
                    "brandName":"String content"
            }
    ]
    "recipientBrandIdDefault":"String content",
    "senderBrandIdDefault":"String content",
}
```

Delete Brand Profiles

This deletes one or more brand profiles from an account. The Account Branding feature (accountSettings “canSelfBrandSend” and “canSelfBrandSend” are true) must be enabled for the account to use this.

URL:

/accounts/{accountId}/brands/

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
brandId	Yes	String	The brandId to be deleted from the account.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/brands

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "brands":[
            {"brandId":"String content"},
```

```
            {"brandId":"String content"}
    ]
}
```

Response

The response returns the success (200 – OK) or failure.

Checking Status of all Bulk Recipient Batches

This returns status information about all the bulk recipient batches. A bulk recipient batch is the set of envelopes sent from a single bulk recipient file. The response includes general information about each bulk recipient batch.

URL:

/accounts/{accountId}/bulk_envelopes

Optional query strings: count={integer 1 to 20}, start_position={integer}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required, but the following optional query strings can be added to narrow the response results.

Name	Reqd?	Type	Description
count	No	Integer	The number of results to return. This can be 1 to 20.
start_position	No	Integer	The position of the bulk envelope items in the response. This is used for repeated calls, when the number of bulk envelopes returned is too large for one return. The default value is 0.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/bulk_envelopes

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns information about the envelopes sent with bulk recipient batches, including the batchId, which can be used to retrieve a more detailed status of individual bulk recipient batches.

The following example shows the response json body.

Example Response Body

```
{
  "batchSize": "string",
  "batchId": "string",
  "bulkEnvelopesBatchUri": "string",
  "failed": "string",
  "queued": "string",
  "sent": "string",
  "submittedDate": "string",
  "resultSetSize": "string",
  "startPosition": "string",
  "endPosition": "string",
  "totalSetSize": "string",
  "nextUri": "string",
  "previousUri": "string",
  "bulkEnvelopes": [
    {
      "transactionId": "string",
      "submittedDateTime": "string",
      "envelopeId": "string",
      "envelopeUri": "string",
      "bulkRecipientRow": "string",
      "name": "string",
      "email": "string",
      "bulkStatus": "string"
    }
  ]
}
```

Checking the Status of one Bulk Send Batch

This returns status information about a single bulk recipient batch. A bulk recipient batch is the set of envelopes sent from a single bulk recipient file.

URL:

/accounts/{accountId}/bulk_envelopes/{batchId}

Optional query strings: include={all, failed, queued, processing, sent}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the batchId.

The following optional query strings can be added to narrow the response results.

Name	Reqd?	Type	Description
include	No	String	Sets which entries are included in the response. Multiple entries can be included by using commas in the query string (example: ?include="failed,queued") Valid entries are: • all - Returns all entries. If present, overrides all other query settings. This is the default if no query string is provided. • failed - This only returns entries with a failed status. • queued - This only returns entries with a queued status. • processing – This only returns entries with a processing status. • sent – This only returns entries with a sent status.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}//bulk_envelopes
/{batchId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns status information about a single bulk recipient batch.

The following example shows the response json body.

Example Response Body

```
{
  "bulkEnvelopeStatuses": [
    {
      "batchSize": "string",
      "batchId": "string",
      "bulkEnvelopesBatchUri": "string",
      "failed": "string",
      "queued": "string",
      "sent": "string",
```

```
      "submittedDate": "string",
      "resultSetSize": "string",
      "startPosition": "string",
      "endPosition": "string",
      "totalSetSize": "string",
      "nextUri": "string",
      "previousUri": "string",
      "bulkEnvelopes": [
        {
          "transactionId": "string",
          "submittedDateTime": "string",
          "envelopeId": "string",
          "envelopeUri": "string",
          "bulkRecipientRow": "string",
          "name": "string",
          "email": "string",
          "bulkStatus": "string"
        }
      ]
    }
  ],
  "resultSetSize": "string",
  "startPosition": "string",
  "endPosition": "string",
  "totalSetSize": "string",
  "nextUri": "string",
  "previousUri": "string"
}
```

Delete Captive Recipient Signature

This deletes the signature for one or more captive recipient records. This provides a way to reset the signature associated with a clientUserId so a new signature can be created the next time the clientUserId is used.

URL:

/accounts/{accountId}/captive_recipients/signature

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
Email	Yes	String	The email address associated with the captive recipient.
userName	Yes	String	The user name associated with the captive recipient.

Name	Reqd?	Type	Description
clientUserId	Yes	String	The sender created value associated with the captive recipient.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/
captive_recipients/signature

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "captiveRecipients": [
    {
      "email": "String content",
      "userName": " String content",
      "clientUserId": " String content"
    },
    {
      "email": " String content",
      "userName": " String content",
      "clientUserId": " String content"
    }
  ]
}
```

Response

The response returns success or failure and any error messages.

The following example shows the response json body.

Example Response Body

```
{
  "captiveRecipients": [
    {
      "email": "sample string",
      "userName": "sample string",
      "clientUserId": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    },
    {
      "email": "sample string",
      "userName": "sample string",
      "clientUserId": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    }
  ]
}
```

Set Up a Connect Configuration

This allows you to set up and configure a DocuSign Custom Connect definition for your account. DocuSign Connect enables the sending of real-time data updates to external applications. These updates are generated by user transactions as the envelope progresses through actions to completion. Connect Service provides updated information about the status of these transactions and provides updates that include the actual content of document form fields; however, these updates might or might not include the document itself. Refer to the *DocuSign Connect Service Guide* for more information about Connect.

Note: Connect must be enabled for your account to use this function. This can only be used to set up Custom Connect configurations; it cannot be used to set up Connect configurations for Box, eOriginal or Salesforce.

URL:

/accounts/{accountId}/connect

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
allUsers	No	Boolean	When true, the tracked envelope and recipient events for all users, including users that are added a later time, are sent through Connect.
allowEnvelopePublish	No	Boolean	When true, data is sent to the urlToPublishTo web address. This option can be set to false to stop sending data while maintaining the Connect configuration information.
enableLog	No	Boolean	This turns Connect logging on or off. When true, logging is turned on.
envelopeEvents	No	String	A comma separated list of “Envelope” related events that are tracked through Connect. The possible event values are: Sent, Delivered, Completed, Declined, Voided
includeDocuments	No	Boolean	When true, Connect will send the PDF document along with the update XML.

Name	Reqd?	Type	Description
includeDocumentFields	No	Boolean	When true, the Document Fields associated with envelope documents are included in the data. Document Fields are optional custom name-value pairs added to documents using the API.
includeEnvelopeVoidReason	No	Boolean	When true, Connect will include the voidedReason for voided envelopes.
includeSenderAccount	No	Boolean	When true, Connect will include the sender account as Custom Field in the data.
includeTimeZoneInformation	No	Boolean	When true, Connect will include the envelope time zone information.
name	Yes	String	The name of the Connect configuration. The name helps identify the configuration in the list.
recipientEvents	No	String	A comma separated list of “Recipient” related events that are tracked through Connect. The possible event values are: Sent, Delivered, Completed, Declined, AuthenticationFailed, and AutoResponded.
requireAcknowledgement	No	Boolean	When true and a publication message fails to be acknowledged, the message goes back into the queue and the system will retry delivery after a successful acknowledgement is received. If the delivery fails a second time, the message is not returned to the queue for sending until Connect receives a successful acknowledgement and it has been at least 24 hours since the previous retry. There is a maximum of ten retries Alternately, you can use Republish Connect Information to manually republish the envelope information.
signMessageWithCertificate	No	Boolean	When true, Connect messages are signed with an X509 certificate. This provides support for 2-way SSL in the envelope.
soapNameSpace	No*	String	The namespace of the SOAP interface. * This is required if useSoapInterface is set to true.
urlToPublishTo	Yes	String	This is the web address and name of your listener or Retrieving Service endpoint. You need to include HTTPS:// in the web address.

Name	Reqd?	Type	Description
useSoapInterface	No	Boolean	Set to true if the urlToPublishTo is a SOAP endpoint.
userIds	No*	String	A comma separated list of userIds. This sets the users associated with the tracked envelope and recipient events. When one of the event occurs for a set user, the information is sent through Connect. * If allUsers is set to 'false' then you must provide a list of user id's.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "allUsers": "boolean",
    "allowEnvelopePublish": "boolean",
    "enableLog": "boolean",
    "envelopeEvents": "string content",
    "includeDocuments": "boolean",
    "includeEnvelopeVoidReason": "boolean",
    "includeSenderAccountasCustomField": "boolean",
    "includeTimeZoneInformation": "boolean",
    "name": "string content",
    "recipientEvents": "string content",
    "requiresAcknowledgement": "boolean",
    "signMessageWithX509Certificate": "boolean",
    "soapNamespace": "string content",
    "urlToPublishTo": "string content",
    "useSoapInterface": "boolean",
    "userIds": "string content",
    "includeDocumentFields": "boolean"
}
```

Response

The response returns a success or failure with the Connect configuration information, including a DocuSign generated connectId.

The following example shows the response json body.

Example Response Body

```
{
    "allUsers": "boolean",
    "allowEnvelopePublish": "boolean",
    "connectId":"String content",
    "enableLog": "boolean",
    "envelopeEvents": "string content",
    "includeDocuments": "boolean",
    "includeEnvelopeVoidReason": "boolean",
```

```
    "includeSenderAccountasCustomField": "boolean",
    "includeTimeZoneInformation": "boolean",
    "name": "string content",
    "recipientEvents": "string content",
    "requiresAcknowledgement": "boolean",
    "signMessageWithX509Certificate": "boolean",
    "soapNamespace": "string content",
    "urlToPublishTo": "string content",
    "useSoapInterface": "boolean",
    "userIds": "string content",
    "includeDocumentFields": "boolean"
}
```

Get Connect Configuration Information

This allows you to retrieve all the DocuSign Custom Connect definitions for your account.

Note: Connect must be enabled for your account to use this function. This does not retrieve information for Connect configurations for Box, eOriginal or Salesforce.

URL:

/accounts/{accountId}/connect

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the information for all the Connect configurations for the account and a count of the total number of configurations.

The following example shows the response json body.

Example Response Body

```
{
```

```
    "configurations":[{
           "allUsers":"String content",
           "allowEnvelopePublish":"String content",
           "connectId":"String content",
           "enableLog":"String content",
           "envelopeEvents":"String content",
           "includeCertSoapHeader":"String content",
           "includeDocuments":"String content",
           "includeSenderAccountasCustomField":"String content",
           "includeTimeZoneInformation":"String content",
           "name":"String content",
           "recipientEvents":"String content",
           "requiresAcknowledgement":"String content",
           "signMessageWithX509Certificate":"String content",
           "soapNamespace":"String content",
           "urlToPublishTo":"String content",
           "useSoapInterface":"String content",
           "userIds":"String content",
           "includeDocumentFields": "boolean"
    },
    {
           "allUsers":"String content",
           "allowEnvelopePublish":"String content",
           "connectId":"String content",
           "enableLog":"String content",
           "envelopeEvents":"String content",
           "includeCertSoapHeader":"String content",
           "includeDocuments":"String content",
           "includeSenderAccountasCustomField":"String content",
           "includeTimeZoneInformation":"String content",
           "name":"String content",
           "recipientEvents":"String content",
           "requiresAcknowledgement":"String content",
           "signMessageWithX509Certificate":"String content",
           "soapNamespace":"String content",
           "urlToPublishTo":"String content",
           "useSoapInterface":"String content",
           "userIds":"String content",
           "includeDocumentFields": "boolean"
    }],
    "totalRecords":"String content"
}
```

Update a Connect Configuration

This allows you to update a DocuSign Custom Connect configuration for your account.

Note: Connect must be enabled for your account to use this function. This is cannot be used to update Connect configurations for Box, eOriginal or Salesforce.

URL:

/accounts/{accountId}/connect

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
allUsers	No	Boolean	When true, the tracked envelope and recipient events for all users, including users that are added a later time, are sent through Connect.
allowEnvelopePublish	No	Boolean	When true, data is sent to the urlToPublishTo web address. This option can be set to false to stop sending data while maintaining the Connect configuration information.
connectId	Yes	Integer	The DocuSign generated ID for the Connect configuration.
enableLog	No	Boolean	This turns Connect logging on or off. When true, logging is turned on.
envelopeEvents	No	String	A comma separated list of “Envelope” related events that are tracked through Connect. The possible event values are: Sent, Delivered, Completed, Declined, Voided
includeDocuments	No	Boolean	When true, Connect will send the PDF document along with the update XML.
includeDocumentFields	No	Boolean	When true, the Document Fields associated with envelope documents are included in the data. Document Fields are optional custom name-value pairs added to documents using the API.
includeEnvelopeVoidReason	No	Boolean	When true, Connect will include the voidedReason for voided envelopes.
includeSenderAccount	No	Boolean	When true, Connect will include the sender account as Custom Field in the data.
includeTimeZoneInformation	No	Boolean	When true, Connect will include the envelope time zone information.
name	No	String	The name of the Connect configuration. The name helps identify the configuration in the list.

Name	Reqd?	Type	Description
recipientEvents	No	String	A comma separated list of “Recipient” related events that are tracked through Connect. The possible event values are: Sent, Delivered, Completed, Declined, AuthenticationFailed, and AutoResponded.
requireAcknowledgement	No	Boolean	When true and a publication message fails to be acknowledged, the message goes back into the queue and the system will retry delivery after a successful acknowledgement is received. If the delivery fails a second time, the message is not returned to the queue for sending until Connect receives a successful acknowledgement and it has been at least 24 hours since the previous retry. There is a maximum of ten retries Alternately, you can use Republish Connect Information to manually republish the envelope information.
signMessageWithCertificate	No	Boolean	When true, Connect messages are signed with an X509 certificate. This provides support for 2-way SSL.
soapNameSpace	No*	String	The namespace of the SOAP interface. * This is required if useSoapInterface is set to true.
urlToPublishTo	No	String	This is the web address and name of your listener or Retrieving Service endpoint. You need to include HTTPS:// in the web address.
useSoapInterface	No	Boolean	Set to true if the urlToPublishTo is a SOAP endpoint.
userIds	No*	String	A comma separated list of userIds. This sets the users associated with the tracked envelope and recipient events. When one of the event occurs for a set user, the information is sent through Connect. * If allUsers is set to ‘false’ then you must provide a list of user id’s.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
```

```
Content-Type: application/json

{
    "allUsers": "boolean",
    "allowEnvelopePublish": "boolean",
    "connectId":"String content",
    "enableLog": "boolean",
    "envelopeEvents": "string content",
    "includeDocuments": "boolean",
    "includeEnvelopeVoidReason": "boolean",
    "includeSenderAccountasCustomField": "boolean",
    "includeTimeZoneInformation": "boolean",
    "name": "string content",
    "recipientEvents": "string content",
    "requiresAcknowledgement": "boolean",
    "signMessageWithX509Certificate": "boolean",
    "soapNamespace": "string content",
    "urlToPublishTo": "string content",
    "useSoapInterface": "boolean",
    "userIds": "string content",
    "includeDocumentFields": "boolean"

}
```

Response

The response returns a success or failure with the updated Connect configuration information.

The following example shows the response json body.

Example Response Body

```
{
    "allUsers": "boolean",
    "allowEnvelopePublish": "boolean",
    "connectId":"String content",
    "enableLog": "boolean",
    "envelopeEvents": "string content",
    "includeDocuments": "boolean",
    "includeEnvelopeVoidReason": "boolean",
    "includeSenderAccountasCustomField": "boolean",
    "includeTimeZoneInformation": "boolean",
    "name": "string content",
    "recipientEvents": "string content",
    "requiresAcknowledgement": "boolean",
    "signMessageWithX509Certificate": "boolean",
    "soapNamespace": "string content",
    "urlToPublishTo": "string content",
    "useSoapInterface": "boolean",
    "userIds": "string content",
    "includeDocumentFields": "boolean"
}
```

Get a Connect Configuration Information

This allows you to retrieve the information about one DocuSign Connect configuration.

Note: Connect must be enabled for your account to use this function.

URL:

/accounts/{accountId}/connect/{connectId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/{connectId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the information for the selected Connect configuration.

The following example shows the response json body.

Example Response Body

```
{
    "allUsers":"String content",
    "allowEnvelopePublish":"String content",
    "connectId":"String content",
    "enableLog":"String content",
    "envelopeEvents":"String content",
    "includeCertSoapHeader":"String content",
    "includeDocuments":"String content",
    "includeSenderAccountasCustomField":"String content",
    "includeTimeZoneInformation":"String content",
    "name":"String content",
    "recipientEvents":"String content",
    "requiresAcknowledgement":"String content",
    "signMessageWithX509Certificate":"String content",
    "soapNamespace":"String content",
    "urlToPublishTo":"String content",
    "useSoapInterface":"String content",
    "userIds":"String content",
    "includeDocumentFields": "boolean"
}
```

Delete a Connect Configuration

This allows you to delete one DocuSign Connect configuration.

Note: Connect must be enabled for your account to use this function.

URL:

/accounts/{accountId}/connect/{connectId}

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/
{connectId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a success or failure.

Republish Connect Information for One Envelope

This is used to republish Connect information for a single envelope.

URL:

/accounts/{accountId}/connect/envelopes/{envelopeId}/retry_queue

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

No additional parameters are required.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/envelopes/
{envelopeId}/retry_queue

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a success or failure and the information about the retry queue for the request.

The following example shows the response json body.

Example Response Body

```
{
    "retryQueue":[{
           "configId":"String content",
           "configUrl":"String content",
           "envelopeId":"String content",
           "status":"String content",
           "statusMessage":"String content"
    }]
}
```

Republish Connect Information for Multiple Envelopes

This is used to republish Connect information for the set of envelopes. The primary use is to republish Connect post failures by including envelope IDs for the envelopes that failed to post in the request. The list of envelope IDs that failed to post correctly can be retrieved by getting the failure log.

URL:

/accounts/{accountId}/connect/envelopes/retry_queue

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
envelopeIds	Yes	String	An array list of envelope Id's to be republished.
synchronous	No	Boolean	If true, the system attempts to publish failed posts again and returns the status of the post attempt.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/envelopes/
retry_queue

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "envelopeIds":["String content","String content","String content"],
    "synchronous":"String content"
}
```

Response

The response returns a success or failure and the information about the retry queue for the request.

The following example shows the response json body.

Example Response Body

```
{
    "retryQueue":[{
           "configId":"String content",
           "configUrl":"String content",
           "envelopeId":"String content",
           "status":"String content",
           "statusMessage":"String content"
    }]
}
```

Get the Connect Failure Log

This is used to retrieve the Connect Failure Log information. It can be used to determine which envelopes failed to post, so a republish request can be created.

URL:

/accounts/{accountId}/connect/failures

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/failures

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the Connect failures log and the total number of records in the log.

Name	Type	Description
accountId	String	The account ID associated with the envelope.
configUrl	String	The web address of the listener or Retrieving Service end point for Connect.
connectId	String	The identifier for the Connect configuration that failed. If an account has multiple Connect configurations, this value is used to look up the Connect configuration for the failed post.
created	String	The date and time the entry was created.
email	String	The email that sent the envelope.
envelopeId	String	The envelope ID of the envelope status that failed to post.
error	String	The error that caused the Connect post to fail.
failureId	String	The failure log ID for the failure.
failureUri	String	The uri for the failure.
lastTry	String	The date and time the last attempt to post.
logId	String	The Connect log ID for the entry.
logUri	String	The uri for the log item.
retryCount	Integer	The number of times the Connect post has been retried.
retryUri	String	The uri to retry to publish the Connect failure.
status	String	The new envelope status for the failed Connect post. The possible values are: Any,

		Voided, Created, Deleted, Sent, Delivered, Signed, Completed, Declined, TimedOut, Template, or Processing
subject	String	The envelope subject.
userName	String	The name of the envelope sender.

The following example shows the response json body.

Example Response Body

```
{
    "failures":[{
            "accountId":"String content",
            "configUrl":"String content",
            "connectId":"String content",
            "created":"String content",
            "email":"String content",
            "envelopeId":"String content",
            "error":"String content",
            "failureId":"String content",
            "failureUri":"String content",
            "lastTry":"String content",
            "logId":"String content",
            "logUri":"String content",
            "retryCount":"String content",
            "retryUri":"String content",
            "status":"String content",
            "subject":"String content",
            "userName":"String content"
    },
            "accountId":"String content",
            "configUrl":"String content",
            "connectId":"String content",
            "created":"String content",
            "email":"String content",
            "envelopeId":"String content",
            "error":"String content",
            "failureId":"String content",
            "failureUri":"String content",
            "lastTry":"String content",
            "logId":"String content",
            "logUri":"String content",
            "retryCount":"String content",
            "retryUri":"String content",
            "status":"String content",
            "subject":"String content",
            "userName":"String content"
    }],
    "totalRecords":"String content",
    "type":"String content"
}
```

Get a Connect Failure Log Entry

This is used to retrieve the Connect Failure Log information for one entry.

URL:

/accounts/{accountId}/connect/failures/{failureId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/failures/
{failureId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the Connect failure log information for the failure ID.

Name	Type	Description
accountId	String	The account ID associated with the envelope.
configUrl	String	The web address of the listener or Retrieving Service end point for Connect.
connectId	String	The identifier for the Connect configuration that failed. If an account has multiple Connect configurations, this value is used to look up the Connect configuration for the failed post.
created	String	The date and time the entry was created.
email	String	The email that sent the envelope.
envelopeId	String	The envelope ID of the envelope status that failed to post.
error	String	The error that caused the Connect post to fail.
failureId	String	The failure log ID for the failure.
failureUri	String	The uri for the failure.
lastTry	String	The date and time the last attempt to post.
logId	String	The Connect log ID for the entry.
logUri	String	The uri for the log item.
retryCount	Integer	The number of times the Connect post has been retried.

retryUri	String	The uri to retry to publish the Connect failure.
status	String	The new envelope status for the failed Connect post. The possible values are: Any, Voided, Created, Deleted, Sent, Delivered, Signed, Completed, Declined, TimedOut, Template, or Processing
subject	String	The envelope subject.
userName	String	The name of the envelope sender.

The following example shows the response json body.

Example Response Body

```
{
    "accountId":"String content",
    "configUrl":"String content",
    "connectId":"String content",
    "created":"String content",
    "email":"String content",
    "envelopeId":"String content",
    "error":"String content",
    "failureId":"String content",
    "failureUri":"String content",
    "lastTry":"String content",
    "logId":"String content",
    "logUri":"String content",
    "retryCount":"String content",
    "retryUri":"String content",
    "status":"String content",
    "subject":"String content",
    "userName":"String content"
}
```

Clear a Connect Failure Log Entry

This is used to clear (remove) the Connect Failure Log information for one entry.

URL:

/accounts/{accountId}/connect/failures/{failureId}

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/failures/
{failureId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a success or failure message.

Get the Connect Log

This retrieves a list of connect log entries for your account.

> **Note:** The enableLog setting in the Connect configuration must be set to true to enable logging. If logging is not enabled, then no log entries are recorded.

URL:

/accounts/{accountId}/connect/logs

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/logs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the Connect log information. The response is divided into two sections, one for regular logs and one for Connect failures.

Name	Type	Description
accountId	String	The account ID associated with the envelope.
configUrl	String	The web address of the listener or Retrieving Service end point for Connect.
connectId	String	The identifier for the Connect configuration that failed. If an account has multiple Connect configurations, this value is used to look up the Connect configuration for the failed post.
created	String	The date and time the entry was created.
email	String	The email that sent the envelope.
envelopeId	String	The envelope ID of the envelope status that failed to post.
error	String	The error that caused the Connect post to fail.
failureId	String	The failure log ID for the failure.
failureUri	String	The uri for the failure.
lastTry	String	The date and time the last attempt to post.
logId	String	The Connect log ID for the entry.
logUri	String	The uri for the log item.
retryCount	Integer	The number of times the Connect post has been retried.
retryUri	String	The uri to retry to publish the Connect failure.
status	String	The new envelope status for the failed Connect post. The possible values are: Any, Voided, Created, Deleted, Sent, Delivered, Signed, Completed, Declined, TimedOut, Template, or Processing
subject	String	The envelope subject.
userName	String	The name of the envelope sender.

The following example shows the response json body.

Example Response Body

```
{
  "type": "string",
  "logs": [
    {
      "accountId": "string",
      "envelopeId": "string",
      "subject": "string",
      "created": "string",
      "userName": "string",
      "email": "string",
      "status": "string",
      "lastTry": "string",
```

```
      "retryCount": "string",
      "error": "string",
      "connectId": "string",
      "configUrl": "string",
      "logUri": "string",
      "logId": "string",
      "failureUri": "string",
      "failureId": "string",
      "retryUri": "string",
    }
  ],
  "failures": [
    {
      "accountId": "string",
      "envelopeId": "string",
      "subject": "string",
      "created": "string",
      "userName": "string",
      "email": "string",
      "status": "string",
      "lastTry": "string",
      "retryCount": "string",
      "error": "string",
      "connectId": "string",
      "configUrl": "string",
      "logUri": "string",
      "logId": "string",
      "failureUri": "string",
      "failureId": "string",
      "retryUri": "string",
    }
  ],
  "totalRecords": "string"
}
```

Clear the Connect Log

This is used to clear (remove) the entries from the Connect Log.

URL:

/accounts/{accountId}/connect/logs

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/logs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a success or failure message.

Get One Connect Log Entry

This retrieves one Connect log entry for your account.

> **Note:** The enableLog setting in the Connect configuration must be set to true to enable logging. If logging is not enabled, then no log entries are recorded.

URL:

/accounts/{accountId}/connect/logs/{logId}

Optional query string: additional_info=true

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required, but the following optional query can be added to provide additional information in the response.

Name	Reqd?	Type	Description
additional_info	No	Boolean	When true, the connectDebugLog information is included in the response.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/logs/{logId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the Connect log information for the requested log ID.

Name	Type	Description
accountId	String	The account ID associated with the envelope.
configUrl	String	The web address of the listener or Retrieving Service end point for Connect.
connectId	String	The identifier for the Connect configuration that failed. If an account has multiple Connect configurations, this value is used to look up the Connect configuration for the failed post.
connectDebugLog		A complex element containing information about the Connect configuration, error details, date/time, description and payload. This is only included in the response if the query additional_info=true is used.
created	String	The date and time the entry was created.
email	String	The email that sent the envelope.
envelopeId	String	The envelope ID of the envelope status that failed to post.
error	String	The error that caused the Connect post to fail.
failureId	String	The failure log ID for the failure.
failureUri	String	The uri for the failure.
lastTry	String	The date and time the last attempt to post.
logId	String	The Connect log ID for the entry.
logUri	String	The uri for the log item.
retryCount	Integer	The number of times the Connect post has been retried.
retryUri	String	The uri to retry to publish the Connect failure.
status	String	The new envelope status for the failed Connect post. The possible values are: Any, Voided, Created, Deleted, Sent, Delivered, Signed, Completed, Declined, TimedOut, Template, or Processing
subject	String	The envelope subject.
userName	String	The name of the envelope sender.

The following example shows the response json body.

Example Response Body

```
{
    "accountId":"String content",
    "configUrl":"String content",
    "connectId":"String content",
    "connectDebugLog": [
           {
                  "eventDateTime": "string",
                  "connectConfig": "string",
                  "eventDescription": "string",
                  "payload": "string",
           }
    ],
    "created":"String content",
    "email":"String content",
    "envelopeId":"String content",
    "error":"String content",
    "failureId":"String content",
    "failureUri":"String content",
    "lastTry":"String content",
    "logId":"String content",
    "logUri":"String content",
    "retryCount":"String content",
    "retryUri":"String content",
    "status":"String content",
    "subject":"String content",
    "userName":"String content"
}
```

Clear One Connect Log Entry

This is used to clear (remove) one entry from the Connect Log.

URL:

/accounts/{accountId}/connect/logs/{logId}

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/connect/logs/
{logId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
```

```
Content-Type: application/json
```

Response

The response returns a success or failure message.

Get Disclosure

This returns the Electronic Record and Signature Disclosure, with html formatting, associated with the account. You can use an optional query string to set the language for the disclosure.

URL:

/accounts/{accountId}/consumer_disclosure

Optional query addition: langCode={value}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query can be added to set the language of the disclosure in the response.

Name	Reqd?	Type	Description
langCode	No	String	The simple type enumeration the language used in the response. The supported languages, with the language value shown in parenthesis, are: Arabic (ar), Bulgarian (bg), Czech (cs), Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Croatian (hr), Danish (da), Dutch (nl), English US (en), English UK (en_GB), Estonian (et), Farsi (fa), Finnish (fi), French (fr), French Canada (fr-CA), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Bahasa Indonesia (id), Italian (it), Japanese (ja), Korean (ko), Latvian (lv), Lithuanian (lt), Bahasa Melayu (ms), Norwegian (no), Polish (pl), Portuguese (pt), Portuguese Brazil (pt_BR), Romanian (ro), Russian (ru), Serbian (sr), Slovak (sk), Slovenian (sl), Spanish (es),Spanish Latin America (es_MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). Additionally, the value can be set to ‘browser’ to

Name	Reqd?	Type	Description
			automatically detect the browser language being used by the viewer and display the disclosure in that language.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/consumer_disclosure

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the accountEsignId with the GUID and the esignAgreement with the Electronic Record and Signature Disclosure text. The disclosure text includes the html formatting.

The following example shows the response json body. For brevity, the example only shows a portion of the Electronic Record and Signature Disclosure text.

Example Response Body

```
{
  "accountEsignId": "string",
  "esignAgreement": "\r\n<B>CONSUMER DISCLOSURE</B>\r\n<P></P>\r\nFrom time to time, Rest
Tester Account (we, us or Company) may be required by law to provide to you certain
written notices or disclosures. Described below are the terms and conditions for
providing to you such notices and disclosures electronically through the DocuSign, Inc.
(DocuSign) electronic signing system. Please read the information below carefully and
thoroughly, and if you can access this information electronically to your satisfaction
and agree to these terms and conditions, please confirm your agreement by clicking the "I
agree" button at the bottom of this document.\r\n<P></P>"
}
```

Get List of Account Custom Fields

This retrieves a list of envelope custom fields associated with the account. These fields can be used in the envelopes for your account to record information about the envelope, help search for envelopes and track information. The envelope custom fields are shown in the Envelope Settings section when a user is creating an envelope in the DocuSign member console. The envelope custom fields are not seen by the envelope recipients.

There are two types of envelope custom fields, text and list. A text custom field lets the sender enter the value for the field. The list custom field lets the sender select the value of the field from a premade list.

URL:

/accounts/{accountId}/custom_fields

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/custom_fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of list custom fields and text custom fields associated with the account. .

Name	Type	Description
listCustomFields		
name	String	The name of the Envelope Custom Field
required	String (Boolean)	When true, information must be entered into the custom field to send the envelope.
show	String (Boolean)	When true, the custom field is shown to senders in the DocuSign member console during sending.
listItems	String	The list of values that can be selected by senders. The list values are separated by semi-colons. Example: [one;two;three;four].
textCustomFields		
name	String	The name of the Envelope Custom Field
required	String (Boolean)	When true, information must be entered into the custom field to send the envelope.
show	String (Boolean)	When true, the custom field is shown to senders in the DocuSign member console during sending.

The following example shows the response json body.

Example Response Body

```
{
    "listCustomFields": [{
           "name":"String",
           "required":"String",
           "show":"String",
```

```
            "listItems": ["String","String"]
    },
    {
            "name":"String",
            "required":"String",
            "show":"String",
            "listItems": ["String","String"]
    }],
    "textCustomFields": [{
            "name":"String",
            "required":"String",
            "show":"String",
    },
    {
            "name":"String",
            "required":"String",
            "show":"String",
    }]
}
```

Send an Envelope or Create a Draft Envelope

This creates an envelope and sends it to recipients or saves it as a draft envelope. Setting the status parameter sets if the envelope is sent or if it is saved to the draft envelope folder after the request. This is a multipart/form request.

This resource is also used for sending an envelope from a template; refer to Send an Envelope from a Template for information about sending from a template.

Envelope Event Notification: eventNotification is an optional element that specifies a set of envelope and recipient status codes, a URL and some other options. When the envelope or recipient status changes to one of the specified status codes, a message containing the updated status is sent to the URL.

Note: DocuSign Connect must be enabled to use eventNotification, but Connect does not need to be configured for the account since the configuration is done for each envelope.

Rules for determining the brandId used in an envelope

The following rules are used to determine the brandId used in an envelope:

- If a brandId is specified in the envelope/template and that brandId is available to the account, that brand is used in the envelope.
- If more than one template is used in an envelope and more than one brandId is specified, the first brandId specified is used throughout the envelope.
- In cases where no brand is specified and the sender belongs to a Group; if there is only one brand associated with the Group, then that brand is used in the envelope. Otherwise, the account's default signing brand is used.
- For envelopes that do not meet any of the previous rules, the account's default signing brand is used in the envelope.

Important: The BCC Email address feature is designed to provide a copy of all email communications for external archiving purposes. DocuSign recommends that envelopes sent using the BCC for Email Archive feature, including the BCC Email Override option, include

additional signer authentication options. To send a copy of the envelope to a recipient who does not need to sign, use a Carbon Copies or Certified Deliveries Recipient Type.

URL:

/accounts/{accountId}/envelopes

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

The parameters for sending an envelope are broken down into sections. The parameters used depend on how you are sending the envelope, the recipient types used, and tabs included in the envelope. The parameters listed in this section are: Envelope, Document, Recipient, and Tab.

Envelope Parameters:

Name	Reqd?	Type	Description
accessibility	No	String	An optional element that can only be used if Document Accessibility is enabled for the account. This sets the document reading zones for screen reader applications. **Note:** This information is currently generated from the DocuSign web console by setting the reading zones when creating a template, exporting the reading zone string information, and adding it here.
allowMarkup	No	String	If true, Document Markup is enabled for envelope. Account must have Document Markup enabled to use this.
allowReassign	No	String	If true, the recipient can redirect an envelope to a more appropriate recipient.
allowRecipientRecursion	No	String	When set to true, this enables the Recursive Recipients feature and allows a recipient to appear more than once in the routing order.

Name	Reqd?	Type	Description
asynchronous	No	String	If true, will queue the envelope for processing and the envelope status will have a value of 'Processing'. Additionally, get status calls will return 'Processing' until completed. **Note:** if this is used a transactionId is required for this to work correctly. When the envelope is created, the status is processing and an envelopeId is not returned in the response. To get the envelopeId, use a GET envelope query using the transactionId (example: GET https://{server}/restapi /{apiVersion}/accounts/{accountId} /envelopes?transaction_ids=XXX) or by checking the Connect notification.
authoritativeCopy	No	String	Specifies the Authoritative copy feature. If set to true the Authoritative copy feature is enabled.
autoNavigation	No	String	If true, auto-navigation is enabled for the envelope. The auto-navigation method used is determined by the account setting.
brandId	No	String	This sets the brand profile format used for the envelope. The value in the string is the brandId associated with the profile. Account branding must be enabled for the account to use this option.
emailBlurb	No	String	Optional element. This is the same as the email body. If specified it is included in email body for all envelope recipients. This can be a maximum of 10000 characters.
emailSubject	Yes	String	The subject of the email that will be sent to all recipients. This can be a maximum of 100 characters.
enableWetSign	No	String	If true, the signer is allowed to print the document and sign it on paper.
enforceSignerVisibility	No	String	It true, the signer is required to have a signature or initial tab on the document or that the document has no signers in order to view it. Account must have Document Visibility enabled to use this.
envelopeIdStamping	No	String	If true, Envelope ID Stamping is enabled.

Name	Reqd?	Type	Description
messageLock	No	String	If true, prevents senders from changing the emailBlurb and emailSubject for the envelope. Additionally, this prevents users from making changes to the emailBlurb and emailSubject when correcting envelopes However, if the messageLock node is set to True and the emailSubject is empty, senders and correctors are able to add a subject to the envelope.

notification	No		An optional complex element that specifies the notification options for the envelope. It consists of:
			• useAccountDefaults – When true, the account default notification settings are used for the envelope.
			• reminders – A complex element that specifies reminder settings for the envelope. It consists of:
			  ○ reminderEnabled – When true a reminder message is sent to the recipient.
			  ○ reminderDelay – An interger that sets the number of days after the recipient receives the envelope that reminder emails are sent to the recipient.
			  ○ reminderFrequency – An interger that sets the interval, in days, between reminder emails.
			• expirations – A complex element that specifies the expiration settings for the envelope. It consists of:
			  ○ expireEnabled – When true the envelope expires (is no longer available for signing) in the set number of days. If false, the account default setting is used. If the account does not have an expiration setting, the DocuSign default value of 120 days is used.
			  ○ expireAfter – An integer that sets the number of days the envelope is active.
			  ○ expireWarn – An integer that

Name	Reqd?	Type	Description
			sets the number of days before envelope expiration that an expiration warning email is sent to the recipient. If set to 0 (zero), no warning email is sent.
recipientsLock	No	String	If true, prevents senders from changing, correcting, or deleting the recipient information for the envelope.
signingLocation	No	String	Specifies the physical location where the signing takes place. It can have two enumeration values; InPerson and Online. The default value is Online.
status	Yes	String	Sets envelope status. There are two possible values: sent and created. If status = sent, the envelope is sent to the recipients. If status = created, the envelope is saved as a draft and can be modified and sent later.
transactionId	No	String	An optional element that can be used to identify an envelope. The id is a sender-generated value and is valid in the DocuSign system for 7 days. The transactionId can be used determine if an envelope status (i.e. was created or not) for cases where an internet connection was lost before the envelope status could be returned.
useDisclosure	No	Boolean	When set to 'false', the Electronic Record and Signature Disclosure is not shown to any envelope recipients. When set to 'true', the disclosure is shown to recipients in accordance with the account's Electronic Record and Signature Disclosure frequency setting. If there is no setting for use useDisclosure, then the account's normal disclosure setting is used and the useDisclosure setting is not returned in responses when getting envelope information.

Name	Reqd?	Type	Description
customFields	No		A complex element that can be used to record information about the envelope, help search for envelopes and track information. **Note:** Each custom field used in an envelope must have a unique name. **Important:** If custom fields are set here when sending the envelope, only the fields specified in this section are included in the envelope. This overrides any required account level custom fields. See the section on getting Custom Fields for more information about and descriptions of the custom fields.
documents	No	Document	Complex element contains the details on the documents in the envelope. See the Document parameter section for more information.
recipients	No	Recipient	Specifies the envelope recipients. See the Recipient parameter section for more information.
eventNotification	No	eventNotification	This optional complex element allows a message to be sent a specified URL when the envelope or recipient changes status. It is similar to DocuSign Connect. For example, if an envelope changes from "Sent" to "Delivered", a message containing the updated envelope status and optionally the documents is sent to the URL. When an eventNotification is attached to an envelope using the API, it only applies to the envelope (treating the envelope as the sender). This is different from envelopes created through the console user interface, where the user is treated as the sender. eventNotification consists of: • url – The endpoint where envelope updates are sent. This will accept XML unless “useSoapInterface” is set to true. • loggingEnabled – When set to true, logging is turned on for envelope events on the Web Console Connect page. • requireAcknowledgment – When set to true, the DocuSign Connect service checks that the message was received and retries on failures.

Name	Reqd?	Type	Description
			• useSoapInterface – When set to true, this tells the Connect service that the user's endpoint has implemented a SOAP interface. • soapNameSpace – This lists the namespace in the SOAP listener provided. • includeCertificateWithSoap – When set to true, this tells the Connect service to send the DocuSign signedby certificate as part of the outgoing SOAP xml. This appears in the XML as wsse:BinarySecurityToken. • signMessageWithX509Cert – When set to true, messages are signed with an X509 certificate. This provides support for 2-way SSL in the envelope. • includeDocuments – When set to true, the PDF documents are included in the message along with the updated XML. • includeEnvelopeVoidReason – When set to true, this tells the Connect Service to include the void reason, as entered by the person that voided the envelope, in the message. • includeTimeZone – When set to true, the envelope time zone information is included in the message. • includeSenderAccountAsCustomField – When set to true, the sender account ID is included as a envelope custom field in the data. • includeDocumentFields – When set to true, this tells the Connect Service to include the Document Fields associated with the envelope. Document Fields are optional custom name-value pairs added to documents using the API. • includeCertificateOfCompletion – When set to true, this tells the Connect Service to include the Certificate of Completion with completed envelopes. • envelopeEvents – The list of envelope-level events statuses that will trigger Connect to send updates to the url. It can be a two-part list with:

Name	Reqd?	Type	Description
			○ envelopeEventStatusCode – The envelope status, this can be Sent, Delivered, Signed, Completed, Declined, or Voided. ○ includeDocuments – When set to true, the envelope time zone information is included in the message. • recipientEvents – The list of recipient event statuses that will trigger Connect to send updates to the url. It can be a two-part list with: ○ recipientEventStatusCode – The recipient status, this can be Sent, Delivered, Completed, Declined, AuthenticationFailed, and AutoResponded. ○ includeDocuments – When set to true, the envelope time zone information is included in the message.
emailSettings	No	emailSettings	This optional complex element allows sender to override some envelope email setting information. This can be used to override the Reply To email address and name associated with the envelope and to override the BCC email addresses to which an envelope is sent. When the emailSettings information is used for an envelope, it only applies to that envelope. **IMPORTANT:** The emailSettings information is not returned in the GET for envelope status. Use GET /email_settings to return information about the emailSettings. EmailSettings consists of: • replyEmailAddressOverride – The Reply To email used for the envelope. DocuSign will verify a correct email format is used, but does not verify that the email is active. This can be a maximum of 100 characters. • replyEmailNameOverride – The name associated with the Reply To email address. This can be a maximum of 100 characters.

Name	Reqd?	Type	Description
			• bccEmailAddresses – Only users with canManageAccount setting can use this option. This is an array of up to 5 email addresses the envelope is sent to as a BCC email. email – The BCC email address. DocuSign verifies that the email format is correct, but does not verify that the email is active. This can be a maximum of 100 characters. Using this overrides the BCC for Email Archive information setting for this envelope. Example: if your account has BCC for Email Archive set up for the email address 'archive@mycompany.com' and you send an envelope using the BCC Email Override to send a BCC email to 'salesarchive@mycompany.com', then a copy of the envelope is only sent to the 'salesarchive@mycompany.com' email address.

Documents Parameters:

Name	Reqd?	Type	Description
name	Yes	String	The name of the document. This can be a maximum of 100 characters. If using a file from a cloud storage service, this is the name of the file in the cloud storage service. If the document is encrypted, this is the unencrypted name of the document.
documentId	Yes	String	The unique Id for the document in this specific envelope.
remoteUrl	No	String	The file id from the cloud storage service where the document is located. This information is returned using GET /folders/{folderid}.

Name	Reqd?	Type	Description
order	No	String	A numerical value that sets the order documents are presented in the envelope. This is only applicable when adding documents to a draft envelope. When creating and sending an envelope the documents use the order in which they are listed in the request.
transformPdfFields	No	String	Optional element. When set to true PDF form field data will be transformed into document tab values when the PDF form field name matches the DocuSign custom tab TabLabel. The resulting PDF form data will also be returned in the PDF meta data when requesting the document PDF. See the Transform PDF Fields section for more information about how fields are transformed into DocuSign tabs.
encryptedWithKeyManager	No	String	If true, the document is been already encrypted by the sender for use with the DocuSign Key Manager security appliance.
documentFields	No	documentField	An optional array of name-value strings that allows the sender to provide custom data for a document. This information is returned in the envelope status but otherwise not used by DocuSign. The array of DocumentField contains the elements: name – A string that can be a maximum of 50 characters. value – A string that can be a maximum of 200 characters. **IMPORTANT:** If using xml, the name/value pair is contained in a nameValue element. See Add Custom Document Fields to an Envelope Document for more information about the structure.
pages	No	String	The total number of pages in the document.
fileExtension	No	String	Optional element. File extension type of the document. If the document is non-PDF it will be converted to PDF.

Name	Reqd?	Type	Description
documentBase64	No	String	The document byte stream. This allows putting a base64 version of document bytes into an envelope.
matchboxes	No		Optional element. Only used when uploading and editing templates. Matchboxes are used to define areas in a document for document matching when creating envelopes. A matchbox consists of 5 elements: pageNumber - The document page number that the matchbox should be on. xPosition - The x position of the matchbox on a page. yPosition - The y position of the matchbox on a page. width - The width of the matchbox. height - The height of the matchbox.

Transform PDF Fields

Only the FieldTypes and FieldProperties listed below are extrapolated from the forms:

- FieldTypes that are extrapolated are: CheckBox, DateTime, ListBox, Numeric, Radio, Text, Signature, and Password.
- FieldProperties that are extrapolated are: ReadOnly, Required, MaxLength, Positions, and Initial Data.

Note:

When extrapolating Adobe Digital Signatures, the following Adobe names correspond to DocuSign names:

Adobe name contains DocuSignSignHere or eSignSignHere = DocuSign Signature

Adobe name contains DocuSignSignHereOptional or eSignSignHereOptional = DocuSign Optional Signature

Adobe name contains DocuSignInitialHere or eSignInitialHere = DocuSign Initials

Adobe name contains DocuSignInitialHereOptional or eSignInitialHereOptional = DocuSign Optional Initials

Any other Adobe name will default to DocuSign Signature.

When extrapolating Adobe text fields, the following Adobe names correspond to DocuSign names:

Adobe name contains DocuSignSignHere or eSignSignHere = DocuSign Signature

Adobe name contains DocuSignSignHereOptional or eSignSignHereOptional = DocuSign Optional Signature

Adobe name contains DocuSignInitialHere or eSignInitialHere = DocuSign Initials

Adobe name contains DocuSignInitialHereOptional or eSignInitialHereOptional = DocuSign Optional Initials

Adobe name contains DocuSignEnvelopeID or eSignEnvelopeID = DocuSign EnvelopeID

Adobe name contains DocuSignCompany or eSignCompany = DocuSign Company

*Adobe name contains DocuSignDateSigned or eSignDateSigned = DocuSign DateSigned

Adobe name contains DocuSignTitleor eSignTitle = DocuSign Title

Adobe name contains DocuSignFullNameor eSignFullName = DocuSign FullName

Adobe name contains DocuSignSignerAttachmentOptional or eSignSignerAttachmentOptional = DocuSign Optional Attachment

Any other name will default to a DocuSign data (text) field

Note: DocuSign will not transform PDF form fields that have the text "DocuSignIgnoreTransform" or "eSignIgnoreTransform" as part of the name of the PDF form field.

* Adobe date fields can be transformed to DocuSignDateSigned fields using the same naming scheme.

Recipient Parameters:

There are seven possible recipient types: Agents, Carbon Copies, Certified Deliveries, Editors, In Person Signers, Intermediaries, and Signers. Recipient types share some common parameters, but the exact parameters associated with a recipient depend on the recipient type. Refer to the specific recipient type below for more information.

Agents | Carbon Copies | Certified Deliveries | Editors
In Person Signers | Intermediaries | Signers

Tab Parameters:

The exact parameters associated with a tab depend on the type of tab. Tabs are associated with a specific recipient and are only used by the recipient types In Person Signers and Signers. Refer to the Tab Parameters section for more information.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: multipart/form-data; boundary=AAA

--AAA
Content-Type: application/json
Content-Disposition: form-data

{
    "accessibility":"string content",
    "allowMarkup":"String content",
    "allowReassign":"String content",
    "allowRecipientRecursion":"String content",
    "asynchronous":"String content",
```

```
    "authoritativeCopy":"String content",
    "autoNavigation":"String content",
    "brandId":"String content",
    "emailBlurb":"String content",
    "emailSubject":"String content",
    "enableWetSign":"String content",
    "enforceSignerVisibility":"String content",
    "envelopeIdStamping":"String content",
    "messageLock":"string content",
    "notification":{
           "useAccountDefaults":"String content",
           "reminders":{
                  "reminderEnabled":"String content",
                  "reminderDelay":"String content",
                  "reminderFrequency":"String content"
           },
           "expirations":{
                  "expireEnabled":"String content",
                  "expireAfter":"String content",
                  "expireWarn":"String content"
           }
    },
    "recipientsLock":"string content",
    "signingLocation":"String content",
    "status":"String content",
    "transactionId":"String content",
    "useDisclosure":"String content",
    "customFields":{
           "listCustomFields":[{
                  "name":"String content",
                  "required":"String content",
                  "show":"String content",
                  "value":"String content",
                  "listItems":["String content"]
           }],
           "textCustomFields":[{
                  "name":"String content",
                  "required":"String content",
                  "show":"String content",
                  "value":"String content"
           }]
    },
    "documents": [{
                  "documentId":"1",
                  "name":"String content",
                  "remoteUrl": "string"
                  "documentFields": [{
                         "name": "string",
                         "value": "string"
                         },
                         {
                         "name": "string",
                         "value": "string"
                         }],
                  "encryptedWithKeyManager":"String Content",
                  "transformPdfFields":"String Content",
                  "order":"String Content",
                  "pages":"String Content",
                  "fileExtension":"String Content",
                  "documentBase64":"String Content",
                  "matchboxes": [{
                         "pageNumber": "string",
                         "xPosition": "string",
                         "yPosition": "string",
```

```
                        "width": "string",
                        "height": "string"
                        },
                        {
                        "pageNumber": "string",
                        "xPosition": "string",
                        "yPosition": "string",
                        "width": "string",
                        "height": "string"
                        }],
    }],
    "recipients": {
           "signers" : [
                  {
                        "email":"String content",
                        "name":"String content",
                        "recipientId":"1"
                  }
           ],
    },
    "eventNotification":{
           "url":"String content",
           "loggingEnabled":"String content",
           "requireAcknowledgment":"String content",
           "useSoapInterface":"String content",
           "soapNameSpace" : " String content",
           "includeCertificateWithSoap" : " String content",
           "signMessageWithX509Cert":"String content",
           "includeDocuments":"String content",
           "includeEnvelopeVoidReason": "String content",
           "includeTimeZone": " String content",
           "includeSenderAccountAsCustomField" : " String content",
           "includeDocumentFields": " String content",
           "includeCertificateOfCompletion": " String content"
           "envelopeEvents" : [
                  {
                        "envelopeEventStatusCode" : "sent"
                  },
                  {
                        "envelopeEventStatusCode" : "completed",
                        "includeDocuments" : "true"
                  }
           ],
           "recipientEvents" : [
                  {
                        "recipientEventStatusCode" : "Sent"
                  },
                  {
                        "recipientEventStatusCode" : "completed",
                        "includeDocuments" : "true"
                  }
           ]
    },
    "emailSettings": {
           "replyEmailAddressOverride": "string",
           "replyEmailNameOverride": "string",
           "bccEmailAddresses": [
                  {
                        "email": "string"
                  },
                  {
                        "email": "string"
                  }
           ]
```

```
    }
}

--AAA
Content-Disposition:documentId=1

<bytes of PDF removed>

--AAA--
```

Response

The response returns the envelope ID, status, and date-time stamp for the envelope status. It also includes the uri for the envelope.

The following example shows the response json body.

Example Response Body

```
{
    "envelopeId":"String content",
    "status":"String content",
    "statusDateTime":"String content",
    "uri":"String content"
}
```

Send an Envelope from a Template

Creates an envelope from an existing template and sends it or saves it as a draft envelope. When creating an envelope by using a templateId, the “recipients” structure is used to assign recipients to template roles via the roleName property, override recipient settings that have been specified in the template and set values for tab fields that were defined in the template.

When a template is added or applied to an envelope and the template has a locked email subject and message, that subject and message is used for the envelope and cannot be changed even if another locked template is subsequently added or applied to the envelope. If an email subject or message is entered before adding or applying a locked template, the email subject and message will be overwritten with the email subject and message from the locked template.

Rules for determining the brandId used in an envelope

The following rules are used to determine the brandId used in an envelope:

- If a brandId is specified in the envelope/template and that brandId is available to the account, that brand is used in the envelope.
- If more than one template is used in an envelope and more than one brandId is specified, the first brandId specified is used throughout the envelope.
- In cases where no brand is specified and the sender belongs to a Group; if there is only one brand associated with the Group, then that brand is used in the envelope. Otherwise, the account’s default signing brand is used.
- For envelopes that do not meet any of the previous rules, the account's default signing brand is used in the envelope.

Important: The BCC Email address feature is designed to provide a copy of all email communications for external archiving purposes. DocuSign recommends that envelopes sent

using the BCC for Email Archive feature, including the BCC Email Override option, include additional signer authentication options. To send a copy of the envelope to a recipient who does not need to sign, use a Carbon Copies or Certified Deliveries Recipient Type.

Composite Templates:

This structure can be added to create envelopes from a combination of DocuSign templates and PDF forms. The basic envelope remains the same, while the Composite Template adds new document and template overlays into the envelope. There can be any number of Composite Template structures in the envelope.

Each Composite Template consists of server templates, inline templates, PDF Metadata templates and documents.

- Composite Template ID is an optional element used to identify the composite template. It is used as a reference when adding document object information. If used, the document content-disposition must include the compositeTemplateId to which the document should be added. If compositeTemplateId is not specified in the content-disposition, the document is applied based on the documentId only. If no document object is specified, the composite template inherits the first document.
- Server Templates are server-side templates stored on the DocuSign server. If supplied they are overlaid into the envelope in the order of their Sequence value.
- Inline Templates provide the container to pass inline XML properties. Inline templates allow you to add documents and, for PDF documents, transform the PDF fields into DocuSign tabs. If inline templates are supplied, they are overlaid into the envelope in the order of their Sequence value.
- PDF Metadata Templates provide a container to embed design-time template information into a PDF document. DocuSign uses this information when processing the Envelope. This convention allows the document to carry the signing instructions with it, so that less information needs to be provided at run-time through an inline template or synchronized with an external structure like a server template. PDF Metadata templates are stored in the Metadata layer of a PDF in accordance with Acrobat's XMP specification. DocuSign will only find PDF Metadata templates inside documents passed in the Document object (see below). If supplied the PDF metadata template will be overlaid into the envelope in the order of their Sequence value.
- Document objects are optional structures that provide a container to pass in a document or form. If this object is not passed, the composite template inherits the first document it finds from some other server template or inline template, starting with the lowest *sequence* value (discussed below).

  If there are multiple composite templates, the document content-disposition can include the compositeTemplateId to which the document should be added. Using the compositeTemplateId sets which documents are associated with particular composite templates. An example of this usage is:

```
--5cd3320a-5aac-4453-b3a4-cbb52a4cba5d
Content-Type: application/pdf
Content-Disposition: file; filename="eula.pdf"; documentId=1; compositeTemplateId="1"
Content-Transfer-Encoding: base64
```

  Acrobat form objects are only extrapolated from the document object. DocuSign does not derive Acrobat form properties from server templates or inline templates. To instruct DocuSign to extrapolate objects from the Acrobat form, set transformPdfFields to "true" for the

document. See the Transform PDF Fields section for more information about how fields are transformed into DocuSign tabs.

Sequence

Each type of template has a *sequence* property, which enables the templates to be over-laid in a particular order. This is significant because "last-out" wins in cases of the same property being specified in multiple places in the method schema.

Merge Recipient Roles for Draft Envelopes

When an envelope with multiple templates is sent, the recipients from the templates are merged according to template roles and empty recipients are removed. When creating an envelope with multiple templates, but not sending it (keeping it in a created state), duplicate recipients are not merged, which could cause leave duplicate recipients in the envelope.

To prevent this, the query parameter merge_roles_on_draft should be added when posting a draft envelope (status=created) with multiple templates. Doing this will merge template roles and remove empty recipients.

Note: DocuSign recommends that the merge roles query parameter be used anytime you are creating an envelope with multiple templates and keeping it in draft (created) status.

Template Email Subject Merge Fields

This provides the ability to insert recipient name and email address merge fields into the email subject line when creating or sending from a template.

The merge fields, based on the recipient's roleName, are added to the emailSubject when the template is created or when the template is used to create an envelope. After a template sender adds the name and email information for the recipient and sends the envelope, the recipient information is automatically merged into the appropriate fields in the email subject line.

Both the sender and the recipients will see the information in the email subject line for any emails associated with the template. This provides an easy way for senders to organize their envelope emails without having to open an envelope to check the recipient.

Note: If merging the recipient information into the subject line causes the subject line to exceed 100 characters, then any characters over the 100 character limit are not included in the subject line. For cases where the recipient name or email is expected to be long, you should consider placing the merge field at the start of the email subject.

- To add a recipient's name in the subject line add the following text in the emailSubject when creating the template or when sending an envelope from a template:

  [[<*roleName*>_UserName]]

  Example:

```
  "emailSubject":"[[Signer 1_UserName]], Please sign this NDA",
```

- To add a recipient's email address in the subject line add the following text in the emailSubject when creating the template or when sending an envelope from a template:

  [[<*roleName*>_Email]]

  Example:

```
  "emailSubject":"[[Signer 1_Email]], Please sign this NDA",
```

In both cases the *<roleName>* is the recipient's roleName in the template.

For cases where another recipient (such as an Agent, Editor, or Intermediary recipient) is entering the name and email information for the recipient included in the email subject, then [[*<roleName>*_UserName]] or [[*<roleName>*_Email]] is shown in the email subject.

URL:

/accounts/{accountId}/envelopes

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
accessibility	No	String	An optional element that can only be used if Document Accessibility is enabled for the account. This sets the document reading zones for screen reader applications. **Note:** This information is currently generated from the DocuSign web console by setting the reading zones when creating a template, exporting the reading zone string information, and adding it here.
allowMarkup	No	String	If true, Document Markup is enabled for envelope. Account must have Document Markup enabled to use this.
allowReassign	No	String	If true, the recipient can redirect an envelope to a more appropriate recipient.
allowRecipientRecursion	No	String	When set to true, this enables the Recursive Recipients feature and allows a recipient to appear more than once in the routing order.

Name	Reqd?	Type	Description
asynchronous	No	String	If true, will queue the envelope for processing and the envelope status will have a value of 'Processing'. Additionally, get status calls will return 'Processing' until completed. **Note:** if this is used a transactionId is required for this to work correctly. When the envelope is created, the status is processing and an envelopeId is not returned in the response. To get the envelopeId, use a GET envelope query using the transactionId (example: GET https://{server}/restapi /{apiVersion}/accounts/{accountId} /envelopes?transaction_ids=XXX) or by checking the Connect notification.
authoritativeCopy	No	String	Specifies the Authoritative copy feature. If set to true the Authoritative copy feature is enabled.
autoNavigation	No	String	If true, auto-navigation is enabled for the envelope. The auto-navigation method used is determined by the account setting.
brandId	No	String	This sets the brand profile format used for the envelope. The value in the string is the brandId associated with the profile. Account branding must be enabled for the account to use this option.
emailBlurb	No	String	Optional element. This is the same as the email body. If specified it is included in email body for all envelope recipients. This can be a maximum of 10000 characters.
emailSubject	Yes	String	The subject of the email that will be sent to all recipients. See Template Email Subject Merge Fields for information about adding merge field information to the email subject.
enableWetSign	No	String	If true, the signer is allowed to print the document and sign it on paper.
enforceSignerVisibility	No	String	It true, the signer is required to have a signature or initial tab on the document or that the document has no signers in order to view it. Account must have Document Visibility enabled to use this.
envelopeIdStamping	No	String	If true, Envelope ID Stamping is enabled.

Name	Reqd?	Type	Description
messageLock	No	String	If true, prevents senders from changing the emailBlurb and emailSubject for the envelope. Additionally, this prevents users from making changes to the emailBlurb and emailSubject when correcting envelopes However, if the messageLock node is set to True and the emailSubject is empty, senders and correctors are able to add a subject to the envelope.

notification	No		An optional complex element that specifies the notification options for the envelope. It consists of:
			• useAccountDefaults – When true, the account default notification settings are used for the envelope.
			• reminders – A complex element that specifies reminder settings for the envelope. It consists of:
			  ○ reminderEnabled – When true a reminder message is sent to the recipient.
			  ○ reminderDelay – An interger that sets the number of days after the recipient receives the envelope that reminder emails are sent to the recipient.
			  ○ reminderFrequency – An interger that sets the interval, in days, between reminder emails.
			• expirations – A complex element that specifies the expiration settings for the envelope. It consists of:
			  ○ expireEnabled – When true the envelope expires (is no longer available for signing) in the set number of days. If false, the account default setting is used. If the account does not have an expiration setting, the DocuSign default value of 120 days is used.
			  ○ expireAfter – An integer that sets the number of days the envelope is active.
			  ○ expireWarn – An integer that

Name	Reqd?	Type	Description
			sets the number of days before envelope expiration that an expiration warning email is sent to the recipient. If set to 0 (zero), no warning email is sent.
recipientsLock	No	String	If true, prevents senders from changing, correcting, or deleting the recipient information for the envelope.
signingLocation	No	String	Specifies the physical location where the signing takes place. It can have two enumeration values; InPerson and Online.
status	Yes	String	Sets envelope status. There are two possible values: sent and created. If status = sent, the envelope is sent to the recipients. If status = created, the envelope is saved as a draft and can be modified and sent later.
customFields	No		A complex element that can be used to record information about the envelope, help search for envelopes and track information. **Note:** Each custom field used in an envelope must have a unique name. This is especially important to check if sending an envelope with multiple templates. See the section on getting Custom Fields for more information about and descriptions of the custom fields.
transactionId	No	String	An optional element that can be used to identify an envelope. The id is a sender-generated value and is valid in the DocuSign system for 7 days. The transactionId can be used determine if an envelope status (i.e. was created or not) for cases where an internet connection was lost before the envelope status could be returned.
templateId	Yes	String	Unique identifier of the template.
templateRoles	Yes	Recipients	Specifies the template recipients. Each roleName in the template must have a recipient assigned to it. This is made up elements: • email – The recipient's email address. • name – The recipient's name.

Name	Reqd?	Type	Description
			• roleName – The template roleName associated with the recipient. • clientUserId – Optional, this sets if the signer is This specifies if the recipient is embedded or remote. If the clientUserId is not null then the recipient is embedded. Note that if a ClientUserId is used and the account settings SignerMustHaveAccount or SignerMustLoginToSign are true, an error is generated on sending. • defaultRecipient – Optional, when set to true, this recipient is the default recipient and any tabs generated by the transformPdfFields option are mapped to this recipient. • routingOrder – This specifies the routing order of the recipient in the envelope. • accessCode – This optional element specifies the access code a recipient has to enter to validate the identity. This can be a maximum of 50 characters. • inPersonSignerName – Optional, if the template role is an in person signer, this is the full legal name of the signer. This can be a maximum of 100 characters. • emailNotification – This is an optional complex element that has a role specific emailSubject, emailBody, and language. It follows the same format as the emailNotification node for Recipients. • tabs – This allows the tab values to be specified for matching to tabs in the template.
eventNotification	No	eventNotification	This optional complex element allows a message to be sent a specified URL when the envelope or recipient changes status. It is similar to DocuSign Connect. For example, if an envelope changes from "Sent" to "Delivered", a message containing the updated envelope status and optionally the documents is sent to the URL. When an eventNotification is attached to an envelope using the API, it only applies to the envelope (treating the envelope as the sender). This is different from envelopes created through the console user interface,

Name	Reqd?	Type	Description
			where the user is treated as the sender. eventNotification consists of: • url – The endpoint where envelope updates are sent. This will accept XML unless “useSoapInterface” is set to true. • loggingEnabled – When set to true, logging is turned on for envelope events on the Web Console Connect page. • requireAcknowledgment – When set to true, the DocuSign Connect service checks that the message was received and retries on failures. • useSoapInterface – When set to true, this tells the Connect service that the user’s endpoint has implemented a SOAP interface. • soapNameSpace – This lists the namespace in the SOAP listener provided. • includeCertificateWithSoap – When set to true, this tells the Connect service to send the DocuSign signedby certificate as part of the outgoing SOAP xml. This appears in the XML as wsse:BinarySecurityToken. • signMessageWithX509Cert – When set to true, messages are signed with an X509 certificate. This provides support for 2-way SSL in the envelope. • includeDocuments – When set to true, the PDF documents are included in the message along with the updated XML. • includeEnvelopeVoidReason – When set to true, this tells the Connect Service to include the void reason, as entered by the person that voided the envelope, in the message. • includeTimeZone – When set to true, the envelope time zone information is included in the message. • includeSenderAccountAsCustomField – When set to true, the sender account ID is included as a envelope custom field in the data. • includeDocumentFields – When set to true, this tells the Connect Service to

Name	Reqd?	Type	Description

			include the Document Fields associated with the envelope. Document Fields are optional custom name-value pairs added to documents using the API.
			• includeCertificateOfCompletion – When set to true, this tells the Connect Service to include the Certificate of Completion with completed envelopes.
			• envelopeEvents – The list of envelope-level events statuses that will trigger Connect to send updates to the url. It can be a two-part list with:
			  ○ envelopeEventStatusCode – The envelope status, this can be Sent, Delivered, Completed, Declined, or Voided.
			  ○ includeDocuments – When set to true, the envelope time zone information is included in the message.
			• recipientEvents – The list of recipient event statuses that will trigger Connect to send updates to the url. It can be a two-part list with:
			  ○ recipientEventStatusCode – The recipient status, this can be Sent, Delivered, Completed, Declined, AuthenticationFailed, and AutoResponded.
			  ○ includeDocuments – When set to true, the envelope time zone information is included in the message.

emailSettings	No	emailSettings	This optional complex element allows sender to override some envelope email setting information. This can be used to override the Reply To email address and name associated with the envelope and to override the BCC email addresses to which an envelope is sent. When the emailSettings information is used for an envelope, it only applies to that envelope. **IMPORTANT:** The emailSettings information is not returned in the GET for envelope status. Use GET /email_settings to return information about the

Name	Reqd?	Type	Description
			emailSettings. EmailSettings consists of: • replyEmailAddressOverride – The Reply To email used for the envelope. DocuSign will verify a correct email format is used, but does not verify that the email is active. This can be a maximum of 100 characters. • replyEmailNameOverride – The name associated with the Reply To email address. This can be a maximum of 100 characters. • bccEmailAddresses – Only users with canManageAccount setting can use this option. This is an array of up to 5 email addresses the envelope is sent to as a BCC email. email – The BCC email address. DocuSign verifies that the email format is correct, but does not verify that the email is active. This can be a maximum of 100 characters. Using this overrides the BCC for Email Archive information setting for this envelope. Example: if your account has BCC for Email Archive set up for the email address 'archive@mycompany.com' and you send an envelope using the BCC Email Override to send a BCC email to 'salesarchive@mycompany.com', then a copy of the envelope is only sent to the 'salesarchive@mycompany.com' email address.
compositeTemplates	No		Each compositeTemplate adds a new document and template overlays into the envelope. There can be any number of compositeTemplate structures. Each compositeTemplate consists of: compositeTemplateId - This is an optional element used to identify the composite template. It is used as a reference when adding document information. If used, the document content-disposition must include the compositeTemplateId to which the document should be added. Each compositeTemplateId must be unique within the envelope.

Name	Reqd?	Type	Description
			serverTemplates - 0 or more server-side templates and their position in the overlay. If supplied they are overlaid into the envelope in the order of their Sequence value. inlineTemplates - 0 or more inline templates and their position in the overlay. If supplied they are overlaid into the envelope in the order of their Sequence value. pdfMetaDataTemplate – 0 or 1 embedded template in the PDF meta data. If supplied the PDF meta data template will be overlaid into the envelope in the order of their Sequence value. document - The document is used to specify a different document than the ones in the overlay templates. If there are multiple composite templates, the document content-disposition can include the compositeTemplateId to which the document should be added. The PDF fields from this document will be transformed to DocuSign tags if the TransformPdfFields option is set to true. **Note:** DocuSign will not transform PDF form fields that have the text "DocuSignIgnoreTransform" or "eSignIgnoreTransform" as part of the name of the PDF form field. If pdfMetaDataTemplate is supplied this is the document the template is extracted from. This information is unchanged from the current schema.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
Content-Disposition: form-data

{
    "accessibility":"string content",
    "allowMarkup":"String content",
    "allowReassign":"String content",
```

```
    "allowRecipientRecursion":"String content",
    "asynchronous":"String content",
    "authoritativeCopy":"String content",
    "autoNavigation":"String content",
    "brandId":"String content",
    "emailBlurb":"String content",
    "emailSubject":"String content",
    "enableWetSign":"String content",
    "enforceSignerVisibility":"String content",
    "envelopeIdStamping":"String content",
    "messageLock":"string content",
    "recipientsLock":"string content",
    "signingLocation":"String content",
    "status":"String content",
    "transactionId":"String content",
    "customFields":{
           "listCustomFields":[
                  {
                         "name":"String content",
                         "required":"String content",
                         "show":"String content",
                         "value":"String content",
                         "listItems":[
                                "String content"
                         ]
                  }
           ],
           "textCustomFields":[
                  {
                         "name":"String content",
                         "required":"String content",
                         "show":"String content",
                         "value":"String content"
                  }
           ]
    },
    "templateId":"String content",
    "templateRoles":[
           {
                  "email":"String content",
                  "name":"String content",
                  "roleName":"String content",
                  "inPersonSignerName":"String content",
                  "clientUserId":"String content",
                  "defaultRecipient":"String content",
                  "routingOrder":"String content",
                  "accessCode":"String content",
                  "emailNotification":{
                         "emailSubject":"String content",
                         "emailBody":"String content",
                         "supportedLanguage":"String content"
                  },
                  "tabs": {
                         "textTabs": [
                                {
                                       "tabLabel": "String content",
                                       "name": "String content",
                                       "value": "String content"
                                }
                         ]
                  },
           }
    ],
    "eventNotification":{
```

```
            "url":"String content",
            "loggingEnabled":"String content",
            "requireAcknowledgment":"String content",
            "useSoapInterface":"String content",
            "soapNameSpace" : "http://DocuSignConnectListener",
            "includeCertificateWithSoap" : "true",
            "signMessageWithX509Cert":"String content",
            "includeDocuments":"String content",
            "includeEnvelopeVoidReason": "String content",
            "includeTimeZone": " String content",
            "includeSenderAccountAsCustomField" : " String content",
            "includeDocumentFields": " String content",
            "includeCertificateOfCompletion": " String content"
            "envelopeEvents" : [
                  {
                        "envelopeEventStatusCode" : "sent"
                  },
                  {
                        "envelopeEventStatusCode" : "completed",
                        "includeDocuments" : "true"
                  }
            ],
            "recipientEvents" : [
                  {
                        "recipientEventStatusCode" : "Sent"
                  },
                  {
                        "recipientEventStatusCode" : "completed",
                        "includeDocuments" : "true"
                  }
            ]
      },
      "emailSettings": {
            "replyEmailAddressOverride": "string",
            "replyEmailNameOverride": "string",
            "bccEmailAddresses": [
                  {
                        "email": "string"
                  },
                  {
                        "email": "string"
                  }
            ]
      },
      "compositeTemplates":[
            {
                  "compositeTemplateId":"String content",
                  "serverTemplates":[
                        {
                              "sequence":"String content",
                              "templateId":"String content"
                        },
                        {
                              "sequence":"String content",
                              "templateId":"String content"
                        }
                  ],
                  "inlineTemplates":[
                        {
                              "sequence":"String content",
                              "documents":[
                                    {
                                          "documentId":"String content",
                                          "name":"String content",
```

```
                                                "transformPdfFields":"String"
                                        },
                                        {
                                                "documentId":"String content",
                                                "name":"String content",
                                                "transformPdfFields":"String"
                                        }
                                ]
                        }
                ],
                "pdfMetaDataTemplateSequence":"String content",
                "document" : [
                        {
                                "documentId" : "string",
                                "documentBase64": "string"
                        }
                ]
        }
    ]
}
```

Response

The response returns the envelope ID, status, and date-time stamp for the envelope status. It also includes the uri for the envelope.

The following example shows the response json body.

Example Response Body

```
{
    "envelopeId":"String content",
    "status":"String content",
    "statusDateTime":"String content",
    "uri":"String content"
}
```

Using Bulk Send in the REST API

With the Bulk Send feature, you can easily send the same document to a large number of recipients. You just create a file that contains the list of recipient names and email addresses, start an envelope, select your documents, add the bulk recipient file you created as a recipient, and send the envelope. Once you send an envelope with a bulk recipient file, DocuSign creates a separate envelope for each recipient in the bulk recipient file – eliminating the need to separately create and send an envelope for each signer.

Additionally, you can customize authentication (access code, ID check, phone authentication or social network IDs), add notes and other custom information for each recipient in the list by adding the information to the file.

Bulk Send Limitations:

- Bulk send must be enabled for your account (accountSettings enableBulkRecipient=true) and for the user sending the envelopes (userSettings allowBulkRecipients=true).
- Bulk send can only be used with Signer recipient types and there can only be one bulk recipient in an envelope or template. An envelope or template can have other Signers and recipient types that are added to the envelope or template normally. When a bulk recipient file is added to an

envelope, the single bulk recipient Signer is replaced with all recipients in the bulk recipient file. There can only be one bulk recipient file associated with an envelope when it is sent.

- When an envelope with bulk recipients is sent, the envelopes are added to a bulk recipient queue and sent in a metered fashion. There is a limit of 2,000 envelopes in the bulk recipient queue and an error message is shown to the sender if this limit is reached. If you receive this error, you should wait and resend the envelope at a later time.

> **Note:** If you frequently run into queue limits, please contact your account manage to discuss modifying the queue limits for your account.

General steps to use Bulk Send:

- Create the bulk recipient CSV (Comma Separated Value) file.

  The required and optional information that can be included the file is described in the Add or Replace an Envelope Bulk Recipient File section.

- Create draft envelope with a bulk recipient signer. The envelope can be created from a template that has a bulk recipient.

  A bulk recipient signer is a Signer recipient type where “isBulkRecipient” is set to true. (Note there can only be one bulk recipient signer per envelope).

  Templates and draft envelopes can be saved with a bulk recipient signer, but a bulk recipient file must be uploaded before an envelope with a bulk recipient signer can be sent.

- Add (PUT) the bulk recipient file to the envelope.
- Send the draft envelope.

  Transitioning an envelope with a bulk recipient signer from “created” to “sent” triggers the sending of an envelope for each recipient in the associated bulk recipient file. The original (draft) envelope is discarded. The response returned from sending the envelope included:

  - bulkRecipientsBatchId: This is the batch identifier used to query the status of the entire bulk send operation.
  - bulkRecipientTransactions: This is an array with identifying information about each envelope sent. The information included in this response:
    - transactionId: The ID used to reference the queued envelope transaction.
    - name: The name of the recipient assigned to this envelope transaction.
    - email: The email address of the recipient assigned to this envelope transaction.

Get Envelope Status Changes

This returns envelope status changes for all envelopes. The information returned can be modified by adding query strings to limit the request to check between certain dates and times, or for certain envelopes, or for certain status codes. It is recommended that you use one or more of the query strings in order to limit the size of the response.

> **Important:** Unless you are requesting the status for specific envelopes (using envelopeIds or transactionIds), you must add a from_date in the request.

Request Envelope Status Notes

The REST API GET /envelopes call uses certain filters to find results. In some cases requests are check for “any status change” instead of the just the single status requested. In these cases, more envelopes might be returned by the request than otherwise would be. For example, for a request with the begin date is set to Jan 1st, an end date set to Jan 7th and the status qualifier (from_to_status) set to “Delivered” – the response set might contain envelopes that were created during that time period, but not delivered during the time period.

To avoid unnecessary database queries, the DocuSign system checks requests to ensure that the added filters will not result in a zero-size response before acting on the request. The following table shows the valid envelope statuses (in the Valid Current Statuses column) for the status qualifiers in the request. If the status and status qualifiers in the API request do not contain any of the values shown in the valid current statuses column, then an empty list is returned.

For example, a request with a status qualifier (from_to_status) of “Delivered” and a status of “Created,Sent”, DocuSign will always return an empty list. This is because the request essentially translates to: find the envelopes that were Delivered between the begin and end dates that have a current status of Created or Sent, and since an envelope that has been delivered can never have a status of Created or Sent, a zero-size response would be generated. In this case, DocuSign does not run the request, but just returns the empty list.

Client applications should check that the statuses they are requesting make sense for a given status qualifier.

Status Qualifier (from_to_status)	Effective Status Qualifier	Valid Current Statuses
Any (changed)	StatusChanged	Any, Created, Sent, Delivered, Signed, Completed, Declined, Voided, Deleted
Created	Created	Any, Created, Sent, Delivered, Signed, Completed, Declined, Voided, Deleted
Sent	Sent	Any, Sent, Delivered, Signed, Completed, Declined, Voided, Deleted
Delivered	StatusChanged	Any, Delivered, Signed, Completed, Declined, Voided, Deleted
Signed	StatusChanged	Any, Signed, Completed, Declined, Voided, Deleted
Completed	Completed	Any, Completed, Declined, Voided, Deleted
Declined	StatusChanged	Any, Declined, Deleted
TimedOut - Always return zero results	StatusChanged	Any, Voided, Deleted
Voided	Voided	Any, Voided, Deleted
Deleted	StatusChanged	Any, Deleted

URL:

/accounts/{accountId}/envelopes

Optional query strings: from_date={dateTime}, to_date={dateTime}, status={status code}, from_to_status={changed or any or list of statuses}, envelopeId={envelopeId}, custom_field={envelope custom field name}={envelope custom field value}, transaction_ids={transactionId or request_body}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional parameters can be added to narrow the search results.

Name	Reqd?	Schema Type	Description
from_date	No*	DateTime	The date/time setting that specifies the date/time when the request begins checking for status changes for envelopes in the account. * This is required unless envelopeIds are used.
to_date	No	DateTime	Optional date/time setting that specifies the date/time when the request stops for status changes for envelopes in the account. If not specified, the system uses the time of the call as the to_date.
from_to_status	No	EnvelopeStatusChange	This is the status type checked for in the from_date/to_date period. If "changed" is specified, then envelopes that changed status during the period are found. If for example, "created" is specified, then envelopes created during the period are found. Default is “changed”. Possible values are: Voided, Changed, Created, Deleted, Sent, Delivered, Signed, Completed, Declined, TimedOut and Processing.

Name	Reqd?	Schema Type	Description
status	No	String	The list of current statuses to include in the response. By default, all envelopes found are returned. If values are specified, then of the envelopes found, only those with the current status specified are returned in the results. Possible values are: Voided, Created, Deleted, Sent, Delivered, Signed, Completed, Declined, TimedOut and Processing.
ac_status	No	String	Specifies the Authoritative Copy Status for the envelopes. The possible values are: Unknown, Original, Transferred, AuthoritativeCopy, AuthoritativeCopyExportPending, AuthoritativeCopyExported, DepositPending, Deposited, DepositedEO, or DepositFailed.
envelopeId	No	String	The envelope ID for the envelope.
custom_field	No	String	This specifies the envelope custom field name and value searched for in the envelope information. The value portion of the query can use partial strings by adding '%' (percent sign) around the custom field query value. Example 1: If you have an envelope custom field called "Region" and you want to search for all envelopes where the value is "West" you would use the query: ?custom_field=Region=West. Example 2: To search for envelopes where the ApplicationID custom field has the value or partial value of "DocuSign" in field, the query would be: ?custom_field=ApplicationId=%DocuSign% This would find envelopes where the custom field value is "DocuSign for Salesforce" or "DocuSign envelope."

Name	Reqd?	Schema Type	Description
transactionIds	No	String	If included in the query string, this is a comma separated list of envelope transactionIds. If included in the request_body, this is a list of envelope transactionIds. Note: transactionIds are only valid in the DocuSign system for 7 days.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes?
transaction_ids=request_body

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "transactionIds":[
           "String content",
           "String content"
    ]
}
```

Response

The response returns the uri's for the envelope certificate, custom fields, documents, envelope along with the envelope ID, envelope status, and the date-time stamp for the envelope status change for the envelopes. If a query string was added to the original request, only the envelopes that meet those parameters are included in the response. If transactionIds were part of the query or request body, the envelopeTransactionStatuses with the transactionID, envelopeId and status for the envelope are returned.

The following example shows the response json body.

Example Response Body

```
{
    "envelopes": [{
           "certificateUri": "String content",
           "customFieldsUri": "String content",
           "documentsCombinedUri": "String content",
           "documentsUri": "String content",
           "envelopeId": "String content",
           "envelopeUri": "String content",
           "notificationUri": "String content",
           "recipientsUri": "String content",
           "status": "String content",
           "statusChangedDateTime": "String content"
    },
    {
```

```
            "certificateUri": "String content",
            "customFieldsUri": "String content",
            "documentsCombinedUri": "String content",
            "documentsUri": "String content",
            "envelopeId": "String content",
            "envelopeUri": "String content",
            "notificationUri": "String content",
            "recipientsUri": "String content",
            "status": "String content",
            "statusChangedDateTime": "String content"
    }],
    "resultSetSize": "String content"
}
```

Get Individual Envelope Status

This returns the overall status for a single envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the envelope status information for the requested envelope.

Example Response Body

```
{
  "transactionId": "string",
  "status": "string",
  "documentsUri": "string",
  "recipientsUri": "string",
  "asynchronous": "string",
  "envelopeUri": "string",
```

```
  "emailSubject": "string",
  "emailBlurb": "string",
  "envelopeId": "string",
  "signingLocation": "string",
  "customFieldsUri": "string",
  "authoritativeCopy": "string",
  "notification": {
    "useAccountDefaults": "string",
    "reminders": {
      "reminderEnabled": "string",
      "reminderDelay": "string",
      "reminderFrequency": "string"
    },
    "expirations": {
      "expireEnabled": "string",
      "expireAfter": "string",
      "expireWarn": "string"
    }
  },
  "notificationUri": "string",
  "enforceSignerVisibility": "string",
  "enableWetSign": "string",
  "allowMarkup": "string",
  "allowReassign": "string",
  "createdDateTime": "string",
  "lastModifiedDateTime": "string",
  "deliveredDateTime": "string",
  "sentDateTime": "string",
  "completedDateTime": "string",
  "voidedDateTime": "string",
  "voidedReason": "string",
  "deletedDateTime": "string",
  "declinedDateTime": "string",
  "statusChangedDateTime": "string",
  "documentsCombinedUri": "string",
  "certificateUri": "string",
  "templatesUri": "string",
  "messageLock": "string1",
  "recipientsLock": "string",
  "useDisclosure": "string",
  "emailSettings": {
    "replyEmailAddressOverride": "string",
    "replyEmailNameOverride": "string",
    "bccEmailAddresses": [
      {
        "bccEmailAddressId": "string",
        "email": "string"
      }
    ]
  },
  "purgeState": "string",
  "lockInformation": {
    "lockedByUser": {
      "userName": "string",
      "email": "string",
      "userId": "string",
      "userType": "string",
      "userStatus": "string",
      "uri": "string",
      }
    }
  }
}
```

Send Draft Envelope

This sends a single draft envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}

Add {“status”:”sent”} to the request body to send the envelope.

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The only required parameter is the envelope ID.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "status":"sent"
}
```

Response

The response returns success or failure of the operation.

Void Envelope

This voids a single in-process envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}

Add {“status”:”voided”, “voidedReason”:”void reason”} to the request body to void the envelope.

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The only required parameter is the envelope ID.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "status":"voided",
    "voidedReason":"voided for incorrect recipient"
}
```

Response

The response returns success or failure of the operation.

Modify Draft Envelope Email Subject and Message

This allows senders to change the email subject and message for draft envelope. Changes to the subject or message is not additive, it replaces the current information.

URL:

/accounts/{accountId}/envelopes/{envelopeId}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
emailSubject	Yes*	String	The subject of the email sent to all recipients. This can be a maximum of 100 characters. * The emailSubject for an envelope cannot be blank.
emailBlurb	No	String	Optional element. The email message body text. If specified it is included in email body for all envelope recipients. This can be a maximum of 10000 characters.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "emailSubject":"string",
    "emailBlurb":"string"
}
```

Response

The response returns success or failure of the operation and, for failures, an associated error message.

Purge Documents

This provides the ability to initiate a request to place envelope documents and envelope metadata in a purge queue so that this information is removed from the DocuSign system. The request can only be used for completed envelopes that are not marked as authoritative copy. The requesting user must have permission to purge documents and must be the sender (the requesting user can act as the sender using Send On Behalf Of).

Note: If your account has set up a Document Retention policy by specifying the number of days to retain documents, at the end of the retention period the envelope documents are automatically placed in the purge queue and the warning emails are sent. So setting a Document Retention policy is the same as setting a schedule for purging documents. The Document Retention policy is set in the Classic DocuSign Experience.

When the purge request is initiated the envelope documents, or documents and envelope metadata, are placed in a purge queue for deletion in 14 days. A warning email notification is sent to the sender and all recipients with DocuSign accounts associated with the envelope notifying them that the

envelope documents will be deleted in 14 days and providing a link to the documents. A second email is sent 7 days later with the same message. At the end of the 14-day period, the envelope documents are deleted from the system.

If purgeState="documents_queued" is used in the request, then only the documents are deleted and any corresponding attachments and tabs remain in the DocuSign system. If purgeState= "documents_and_metadata_queued" is used in the request, then the documents, attachments, and tabs are deleted.

URL:

/accounts/{accountId}/envelopes/{envelopeId}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
purgeState	Yes	String	Initiates a purge request. The accepted values are: • documents_queued: Places envelope documents in the purge queue. • documents_and_metadata_queued: Places envelope documents and metadata in the purge queue.

Example JSON Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/
envelopes/{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

{
  "purgeState": "String"
}
```

Response

The response returns success or failure of the operation.

Add or Modify Envelope Information

This lets you add or modify draft or in-process envelope information, including recipients, tabs, custom fields, and notifications, by adding the advanced_update query parameter to the request.

> **Note:** When adding or modifying information for an in-process envelope, DocuSign recommends locking the envelope prior to making any changes. If you send information for a recipient that does not already exist in an envelope, the recipient is added to the envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}

Optional query parameters: advanced_update={true}, resend_envelope={true or false}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The parameters used to modify the envelope information, add or modify recipients, and add or modify tabs are the same as those used in any envelope. See the Recipient Types and Parameters and Tab Types and Parameters sections for more information about recipient and tab parameters.

The resend_envelope flag is only used to resend the in-process envelope.

Name	Reqd?	Type	Description
advanced_update	No	String	When true, envelope information can be added or modified.
resend_envelope	No	String	True or false setting that defaults to false. Setting this to true will resend the envelope to the recipients that have not completed their actions. Note that if automatic reminders are enabled for the envelope, then resending the envelope resets the automatic reminder date and the resend date is used for determining when to send reminder messages.

Example Request Body

The following example shows the request json body.

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}
?advanced_update=true
```

```
X-DocuSign-Authentication:
&lt;DocuSignCredentials&gt;&lt;Username&gt;{name}&lt;/Username&gt;&lt;Password&gt;{passwo
rd}&lt;/Password&gt;&lt;IntegratorKey&gt;{integrator_key}&lt;/IntegratorKey&gt;&lt;/DocuS
ignCredentials&gt;
Accept: application/json
Content-Type: application/json

{
    "allowReassign": "true",
    "brandLock": "true",
    "customFields": {
           "listCustomFields": [
                  {
                         "fieldId": "2429",
                         "name": "custom1List",
                         "value": "orange"
                  },
                  {
                         "fieldId": "2430",
                         "name": "custom2List",
                         "value": "cat"
                  }
           ],
           "textCustomFields": [
                  {
                         "fieldId": "2427",
                         "name": "custom1Text",
                         "value": "newCustom1Text"
                  }
           ]
    },
    "enableWetSign": "true",
    "enforceSignerVisibility": "true",
    "envelopeIdStamping": "false",
    "notification": {
           "expirations": {
                  "expireAfter": "13",
                  "expireEnabled": "true",
                  "expireWarn": "13"
           },
           "reminders": {
                  "reminderDelay": "13",
                  "reminderEnabled": "true",
                  "reminderFrequency": "13"
           },
           "useAccountDefaults": "false"
    },
    "recipients": {
           "signers": [
                  {
                         "recipientId": 1,
                         "tabs": {
                                "signHereTabs": [
                                       {
                                              "documentId": 1,
                                              "pageNumber": 1,
                                              "xPosition": 50,
                                              "yPosition": 50
                                       },
                                       {
                                              "documentId": 1,
                                              "pageNumber": 1,
                                              "xPosition": 100,
                                              "yPosition": 100
```

```
                                        },
                                ]
                        }
                },
                {
                        "recipientId": 2,
                        "tabs": {
                                "signHereTabs": [
                                        {
                                                "documentId": 1,
                                                "pageNumber": 1,
                                                "xPosition": 150,
                                                "yPosition": 150
                                        },
                                        {
                                                "documentId": 1,
                                                "pageNumber": 1,
                                                "xPosition": 200,
                                                "yPosition": \200
                                        },
                                ]
                        }
                },
                {
                        "email": "dcs2.test@anymail.com",
                        "name": "Dee C Smith",
                        "recipientId": 3,
                        "tabs": {
                                "signHereTabs": [
                                        {
                                                "documentId": 1,
                                                "pageNumber": 1,
                                                "xPosition": 250,
                                                "yPosition": 250
                                        },
                                        {
                                                "documentId": 1,
                                                "pageNumber": 1,
                                                "xPosition": 300,
                                                "yPosition": 300
                                        }
                                ]
                        }
                }
        ]
    },
    "signingLocation": "inperson"
}
```

Response

The response returns if the recipient updates were successful and the information that was changed. The following example shows the response json body.

Example Response Body

```
{
    "recipientUpdateResults": [
        {
                "recipientId": "1",
                "errorDetails": {
```

```
                        "errorCode": "SUCCESS",
                        "message": ""
                }
        },
        {
                "recipientId": "2",
                "errorDetails": {
                        "errorCode": "SUCCESS",
                        "message": ""
                }
        },
        {
                "recipientId": "3",
                "errorDetails": {
                        "errorCode": "SUCCESS",
                        "message": ""
                }
        }
    ],
    "tabUpdateResults": {
        "signHereTabs": [
                {
                        "scaleValue": 0.0,
                        "optional": "false",
                        "documentId": "1",
                        "recipientId": "1",
                        "pageNumber": "1",
                        "xPosition": "50",
                        "yPosition": "50",
                        "tabId": "5e92a353-5aa1-416b-8e45-40b6dbf30e69"
                },
                {
                        "scaleValue": 0.0,
                        "optional": "false",
                        "documentId": "1",
                        "recipientId": "1",
                        "pageNumber": "1",
                        "xPosition": "100",
                        "yPosition": "100",
                        "tabId": "f75fb88b-e2eb-4809-aabd-dee039882cd0"
                },
                {
                        "scaleValue": 0.0,
                        "optional": "false",
                        "documentId": "1",
                        "recipientId": "2",
                        "pageNumber": "1",
                        "xPosition": "150",
                        "yPosition": "150",
                        "tabId": "484ece80-4b0f-4001-a3a5-8006beb76c63"
                },
                {
        "scaleValue": 0.0,
        "optional": "false",
        "documentId": "1",
        "recipientId": "2",
        "pageNumber": "1",
        "xPosition": "200",
        "yPosition": "200",
        "tabId": "abd0f9bf-5723-4ac3-aaeb-5f6232268189"
      },
      {
        "scaleValue": 0.0,
        "optional": "false",
```

```
        "documentId": "1",
        "recipientId": "3",
        "pageNumber": "1",
        "xPosition": "250",
        "yPosition": "250",
        "tabId": "1d4b2484-cdfa-4090-98dd-a68699bce7a3"
      },
      {
        "scaleValue": 0.0,
        "optional": "false",
        "documentId": "1",
        "recipientId": "3",
        "pageNumber": "1",
        "xPosition": "300",
        "yPosition": "300",
        "tabId": "5a36985f-1c7d-4cc8-a11c-653137062f40"
      }
    ]
  },
  "textCustomFieldUpdateResults": [
    {
      "fieldId": "2427",
      "name": "custom1Text",
      "show": "false",
      "required": "false",
      "value": "newCustom1Text"
    }
  ],
  "listCustomFieldUpdateResults": [
    {
      "fieldId": "2429",
      "name": "custom1List",
      "show": "false",
      "required": "false",
      "value": "orange"
    },
    {
      "fieldId": "2430",
      "name": "custom2List",
      "show": "false",
      "required": "false",
      "value": "cat"
    }
  ]
}
```


Get Envelope Purge Document Status

This retrieves the current purge state for an envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}
/envelopes/{envelopeId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns envelope information, including the current purge state for the envelope.

Name	Type	Description
purgeState	String	Shows the current purge state for the envelope. The possible values are: • unpurged: There has been no successful request to purge documents. • documents_queued: The envelope documents have been added to the purge queue, but have not been purged. • documents_and_metadata_queued: The envelope documents and metadata have been added to the purge queue, but have not yet been purged. • documents_purged: The envelope documents have been successfully purged. • documents_and_metadata_purged: The envelope documents and metadata have been successfully purged.

The following example shows the response json body.

Example Response Body

```
{
    "transactionId": "string",
    "status": "string",
    "documentsUri": "string",
    "recipientsUri": "string",
    "asynchronous": "string",
    "envelopeUri": "string",
    "emailSubject": "string",
    "emailBlurb": "string",
    "envelopeId": "string",
    "signingLocation": "string",
```

```
    "customFieldsUri": "string",
    "authoritativeCopy": "string",
    "notificationUri": "string ",
    "enforceSignerVisibility": "string",
    "enableWetSign": "string",
    "allowMarkup": "string",
    "allowReassign": "string",
    "createdDateTime": "string",
    "lastModifiedDateTime": "string",
    "deliveredDateTime": "string",
    "sentDateTime": "string",
    "completedDateTime": "string",
    "voidedDateTime": "string",
    "voidedReason": "string",
    "deletedDateTime": "string",
    "declinedDateTime": "string",
    "statusChangedDateTime": "string",
    "documentsCombinedUri": "string",
    "certificateUri": "string",
    "templatesUri": "string",
    "messageLock": "string",
    "recipientsLock": "string",
    "useDisclosure": "string",
    "emailSettings": {
           "replyEmailAddressOverride": "string",
           "replyEmailNameOverride": "string",
           "bccEmailAddresses": [
                  {
                         "bccEmailAddressId": "string",
                         "email": "string"
                  },
                  {
                         "bccEmailAddressId": "string",
                         "email": "string"
                  }
           ]
    },
    "purgeState": "string"
}
```

Get Envelope Audit Events

This returns the events for this envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/audit_events

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/audit_events

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of audit events, in name/value pairs, for the specified envelope. The audit events are listed below.

Name	Value
logTime	The date-time the audit event occurred.
Source	The source of the envelope. Values can be web or API.
UserName	The name of the user associated with the event.
Action	The current action status.
Message	The message associated with the action.
EnvelopeStatus	Status of the envelope. Values can be Voided, Created, Deleted, Sent, Delivered, Signed, Completed, Declined and TimedOut.
ClientIPAddress	The client IP address associated with the action.
Information	Base information associated with the action.
GeoLocation	The geographic location associated with the action.
Language	The language used in the envelope.

The following example shows the response json body.

Example Response Body

```
{
    "auditEvents":[{
           "eventFields":[
                  {"name": "logTime", "value": "2016-02-21T14:44:22.0553110Z"},
                  {"name": "Source", "value": "api"}
           ]
    }]
}
```

Add Envelope Custom Fields to an Envelope

This allows you to add envelope custom fields to draft and in-process envelopes.

Note: Each custom field used in an envelope must have a unique name. Custom Fields added this way are not shown in the Envelope Custom Fields list for the account.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/custom_fields

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

The parameters for an Envelope Custom Field are only required if you are adding that type of custom field.

Name	Reqd?	Type	Description
listCustomFields			
name	Yes	String	The name of the Envelope Custom Field. This can be a maximum of 50 characters.
required	Yes	String (Boolean)	When true, information must be entered into the custom field to send the envelope.
show	Yes	String (Boolean)	When true, the custom field is shown to senders in the DocuSign member console during sending.
value	Yes	String	The selected value for the custom field in this envelope. This can be a maximum of 100 characters.
listItems	Yes	String	The list of values that can be selected by senders. The list values are separated by semicolons. Example: [one,two;three;four"] This can be a maximum of 2048 characters, but each value can only be a maximum of 100 characters.
textCustomFields			
name	Yes	String	The name of the Envelope Custom Field. This can be a maximum of 50 characters.
required	Yes	String (Boolean)	When true, information must be entered into the custom field to send the envelope.
show	Yes	String (Boolean)	When true, the custom field is shown to senders in the DocuSign member console during sending.
value	Yes	String	The value for the custom field in this envelope. This can be a maximum of 100 characters.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/custom_fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "listCustomFields": [{
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String",
           "listItems": ["String";"String"]
    },
    {
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String",
           "listItems": ["String";"String"]
    }],
    "textCustomFields": [{
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String"
    },
    {
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String"
    }]
}
```

Response

The response returns success or failure and any error messages. For all successes, DocuSign will generate a fieldId for the custom envelope field. The fieldId is used when editing or removing the envelope custom field in an envelope.

The following example shows the response json body.

Example Response Body

```
{
    "listCustomFields": [{
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String",
           "listItems": ["String";"String"]
    },
    {
           "fieldId":"String",
           "name":"String",
           "required":"String",
```

```
            "show":"String",
            "value":"String",
            "listItems": ["String";"String"]
    }],
    "textCustomFields": [{
            "fieldId":"String",
            "name":"String",
            "required":"String",
            "show":"String",
            "value":"String"
    },
    {
            "fieldId":"String",
            "name":"String",
            "required":"String",
            "show":"String",
            "value":"String"
    }]
}
```

Get Envelope Custom Field Information

This returns the custom field information for a single envelope. These fields can be used in the envelopes for your account to record information about the envelope, help search for envelopes and track information. The envelope custom fields are shown in the Envelope Settings section when a user is creating an envelope in the DocuSign member console. The envelope custom fields are not seen by the envelope recipients.

There are two types of envelope custom fields, text and list. A text custom field lets the sender enter the value for the field. With a list custom field, the sender selects the value of the field from a pre-made list.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/custom_fields

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/custom_fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
```

```
Content-Type: application/json
```

Response

The response returns information about the custom fields included in the envelope. The response splits the custom fields by list and text types. If there are no custom fields in the envelope, the response returns an empty list.

Name	Type	Description
listCustomFields		
fieldId	String	The DocuSign generated ID number for the Envelope Custom Field.
name	String	The name of the Envelope Custom Field
required	String	When true, information must be entered into the custom field to send the envelope.
show	String	When true, the custom field is shown to senders in the DocuSign member console during sending.
value	String	The selected value for the field.
listItems	String	The list of values that can be selected by senders. The list values are separated by semicolons. Example: [one,two;three;four"]
textCustomFields		
fieldId	String	The DocuSign generated ID number for the Envelope Custom Field.
name	String	The name of the Envelope Custom Field
required	String	When true, information must be entered into the custom field to send the envelope.
show	String	When true, the custom field is shown to senders in the DocuSign member console during sending.
value	String	The entered value for the field.

Example Response Body

```
{
    "listCustomFields":[{
            "fieldId":"String",
            "name":"String content",
            "required":"String content",
            "show":"String content",
            "value":"String content",
            "listItems":["String content"]
    }],
    "textCustomFields":[{
            "fieldId":"String",
            "name":"String content",
            "required":"String content",
            "show":"String content",
```

```
            "value":"String content"
    }]
  }
```

Modify Envelope Custom Fields for an Envelope

This allows you to edit the envelope custom fields for draft and in-process envelopes.

Note: Each custom field used in an envelope must have a unique name.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/custom_fields

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The parameters for an Envelope Custom Field are only required if you are editing that custom field.

Name	Reqd?	Type	Description
listCustomFields			
fieldId	Yes	String	The DocuSign generated ID number for the Envelope Custom Field.
name	Yes	String	The name of the Envelope Custom Field. This can be a maximum of 50 characters.
required	Yes	String (Boolean)	When true, information must be entered into the custom field to send the envelope.
show	Yes	String (Boolean)	When true, the custom field is shown to senders in the DocuSign member console during sending.
value	Yes	String	The selected value for the custom field in this envelope. This can be a maximum of 100 characters.
listItems	Yes	String	The list of values that can be selected by senders. The list values are separated by semicolons. Example: [one,two;three;four"] This can be a maximum of 2048 characters, but each value can only be a maximum of 100 characters.
textCustomFields			

Name	Reqd?	Type	Description
fieldId	Yes	String	The DocuSign generated ID number for the Envelope Custom Field.
name	Yes	String	The name of the Envelope Custom Field. This can be a maximum of 50 characters.
required	Yes	String (Boolean)	When true, information must be entered into the custom field to send the envelope.
show	Yes	String (Boolean)	When true, the custom field is shown to senders in the DocuSign member console during sending.
value	Yes	String	The value for the custom field in this envelope. This can be a maximum of 100 characters.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/custom_fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "listCustomFields": [{
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String",
           "listItems": ["String";"String"]
    },
    {
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String",
           "listItems": ["String";"String"]
    }],
    "textCustomFields": [{
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String"
    },
    {
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String"
    }]
}
```

Response

The response returns success or failure and any error messages. The response also returns the list of envelope custom fields with the updated values.

The following example shows the response json body.

Example Response Body

```
{
    "listCustomFields": [{
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String",
           "listItems": ["String","String"]
    },
    {
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String",
           "listItems": ["String","String"]
    }],
    "textCustomFields": [{
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String"
    },
    {
           "fieldId":"String",
           "name":"String",
           "required":"String",
           "show":"String",
           "value":"String"
    }]
}
```

Remove Envelope Custom Fields from an Envelope

This allows you to remove envelope custom fields for draft and in-process envelopes.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/custom_fields

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
listCustomFields			
fieldId	Yes	String	The DocuSign generated ID number for the Envelope Custom Field.
textCustomFields			
fieldId	Yes	String	The DocuSign generated ID number for the Envelope Custom Field.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/custom_fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "listCustomFields": [{
           "fieldId":"String"
    },
    {
           "fieldId":"String"
    }],
    "textCustomFields": [{
           "fieldId":"String"
    },
    {
           "fieldId":"String"
    }]
}
```

Response

The response returns success or failure and any error messages. The response also returns the list of envelope custom field IDs that were removed from the envelope.

The following example shows the response json body.

Example Response Body

```
{
    "listCustomFields": [{
           "fieldId":"String"
    },
    {
           "fieldId":"String"
    }],
    "textCustomFields": [{
           "fieldId":"String"
    },
    {
           "fieldId":"String"
    }]
```

```
}
```

Add or Modify Documents and Document Fields

This can be used to add or modify documents in created or sent envelopes. Document fields can be added or modified in the same call by adding the apply_document_fields query parameter to the request.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents

Optional query parameter: apply_document_fields={true}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The parameters used to add or modify documents and document fields in an envelope are the same as those used when sending an envelope. See Document parameters. The following optional query parameter can be added to the request.

Name	Reqd?	Type	Description
apply_document_fields	No	Boolean	When true, Document fields can be added or modified while adding or modifying envelope documents.

Example Request Body

The following example shows the request json body.

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
/{envelopeId}/documents?apply_document_fields=true

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: multipart/form-data; boundary=AAA

--AAA
Content-Type: application/json
Content-Disposition: form-data

{
    "documents": [
           {
                  "documentFields": [
```

```
                        {
                                "name": "document category",
                                "value": "added"
                        },
                        {
                                "name": "agent",
                                "value": "Jones"
                        }
                ],
                "documentId": 1,
                "name": "welcomeform.pdf"
            }
    ]
}
--AAA
Content-Type: application/pdf
Content-Disposition: file; filename="welcomeform.pdf"; documentId="1"

%PDF-1.6
%
--- DOC BYTES REMOVED ---
%%EOF
--AAA--
```

Response

The response returns if the updates were successful and the information that was changed.

The following example shows the response json body.

Example Response Body

```
{
    "envelopeId": "03ee1647-f7f8-4e23-9a42-de3bf9d9c7ba",
    "envelopeDocuments": [
            {
                "documentId": "1",
                "name": "welcomeform.pdf",
                "type": "content",
                "uri": "/envelopes/03ee1647-f7f8-4e23-9a42-de3bf9d9c7ba/documents/1",
                "order": "1",
                "containsPdfFormFields": "true",
                "documentFields": [
                        {
                                "name": "document category",
                                "value": "added"
                        },
                        {
                                "name": "agent",
                                "value": "Jones",
                                "errorDetails": {
                                        "errorCode": String,
                                        "message": "String"
                                }
                        }
                ]
            }
    ]
}
```

Remove Documents from a Draft Envelope

This removes one or more documents from an existing draft envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

See Document parameters.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
documents

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

Content-Type: application/json
{
"documents": [{
    "documentId": "String content",
    },
    {
    "documentId": "String content",
    }]
}
```

Response

The response returns the success or failure of each document being added to the envelope and the envelope ID. Failed operations on array elements will add the “errorDetails” structure containing an error code and message. If “errorDetails” is null, then the operation was successful for that item.

Get List of Envelope Documents

This returns a list of documents associated with the specified envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of the documents in the selected envelope, with the document ID, name, type, uri, order, pages, and if the document contains PDF form fields (this is a true/false).

Note: The containPdfFormFields information is only returned for documents that are added to an envelope after the initial envelope creation that have not yet been converted.

The following example shows the response json body.

Example Response Body

```
{
    "envelopeId":"String content",
    "envelopeDocuments":[
           {
                  "documentId":"String content",
                  "name":"String content",
                  "type":"String content",
                  "uri":"String content",
                  "order":"String content",
                  "pages":"String content",
                  "containPdfFormFields":"String content"
           },
           {
                  "documentId":"String content",
                  "name":"String content",
                  "type":"String content",
                  "uri":"String content",
                  "order":"String content",
                  "pages":"String content",
                  "containPdfFormFields":"String content"
           },
    ],
}
```

Add a Document to an Envelope

This adds another document to created or sent envelopes.

> **Note:** When adding or modifying documents for an in-process envelope, DocuSign recommends locking the envelope prior to making any changes.

Refer to Add or Modify Documents and Document Fields for information on adding or modifying multiple documents and document fields with one call.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

See Document parameters.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

Content-Type: application/pdf
Content-Disposition: file; filename="test1.pdf"; documentid=1; fileExtension="pdf"

<Bytes of PDF omitted>
```

Response

The response only returns a success or failure.

Get Document from Envelope

This retrieves the specified document from the envelope. If the account has the Highlight Data Changes feature enabled, there is an option to request that any changes in the envelope be highlighted.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}

Optional additions: encrypted={true}, show_changes={true}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameters are the envelope ID and document ID. The following optional query parameters can be added to the request.

Name	Reqd?	Schema Type	Description
encrypted	No	Boolean	When true the PDF bytes returned in the response are encrypted for all the key managers configured on your DocuSign account. The documents can be decrypted with the KeyManager Decrypt Document API.
show_changes	No	Boolean	Optional. When set to true, any changed fields for the returned PDF are highlighted in yellow and optional signatures or initials outlined in red.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the PDF document as a byte stream. If show_changes was included, any changed fields are highlighted in the PDF.

Get a Page Image

This returns a page image for display.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/pages/{pageId}/page_image

Optional query strings: dpi={dpi for the image}, max_width={maximum width in pixels}, max_height={maximum height in pixels}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following query string parameters can be added to change the page image settings.

Name	Reqd?	Type	Description
dpi	No	String	Sets the dpi for the image.
max_width	No	String	Sets the maximum width for the page image. The dpi is recalculated based on this setting.
max_height	No	String	Sets the maximum height for the page image. The dpi is recalculated based on this setting.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
/{envelopeId}/documents/{documentId}/pages/{pageId}/page_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the requested page image with the requested settings.

Rotate Page Image

This rotates a page image for display. The page image can be rotated to the left or right.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/pages/{pageId}/page_image

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
rotate	Yes	String	Sets the direction the page image is rotated. The possible settings are: left or right

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
/{envelopeId}/documents/{documentId}/pages/{pageId}/page_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "rotate":"String content"
}
```

Response

The response a success or failure.

Remove a Page

This removes a page from a document based on the page ID.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/pages/{pageId}

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
/{envelopeId}/documents/{documentId}/pages/{pageId}
```

```
X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a success or failure.

Get Custom Document Fields for an Envelope Document

This returns the custom document field information for an existing envelope document.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/fields

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}/fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the custom document field name-value pairs for the requested document ID. Examples of the response body for json and XML are shown below.

Example JSON Response Body

```
{
  "documentFields": [
    {
```

```
      "name": "sample string",
      "value": "sample string"
    },
    {
      "name": "sample string",
      "value": "sample string"
    }
  ]
}
```

Example XML Response Body

Note that the structure of the XML response is slightly different than the json response, in that the name/value pairs are included in a nameValue element.

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
    <nameValue>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Add Custom Document Fields to an Envelope Document

This adds custom document fields for an existing envelope document.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/fields

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
documentFields	Yes	documentField	The array of name-value custom data strings to be added to a document. Custom document field information is returned in the status, but otherwise is not used by DocuSign. The array contains the elements:

Name	Reqd?	Type	Description
			name – A string that can be a maximum of 50 characters. value – A string that can be a maximum of 200 characters. **IMPORTANT:** If using xml, the name/value pair is contained in a nameValue element.

Example JSON Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}/fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Example XML Request Body

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Response

The response returns the custom document field name-value pairs added to the document and any errors associated with the fields.

Examples of the response body for json and XML are shown below.

Example JSON Response Body

```
{
  "documentFields": [
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    },
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    }
  ]
}
```

Example XML Response Body

Note that the structure of the XML response is slightly different than the json response, in that the name/value pairs are included in a nameValue element.

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Modify Custom Document Fields for an Envelope Document

This modifies existing custom document fields for an existing envelope document.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/fields

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
documentFields	Yes	documentField	The array of name-value strings that modifies custom data on a document. Custom document field information is returned in the status, but otherwise is not used by DocuSign. The array contains the elements: name – A string that can be a maximum of 50 characters. value – A string that can be a maximum of 200 characters. **IMPORTANT:** If using xml, the name/value pair is contained in a nameValue element.

Example JSON Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}/fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Example XML Request Body

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <name>sample string 1</name>
```

```
      <value>sample string 2</value>
    </nameValue>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Response

The response returns the modified custom document field name-value pairs for the document and any errors associated with the fields.

Examples of the response body for json and XML are shown below.

Example JSON Response Body

```
{
  "documentFields": [
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    },
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    }
  ]
}
```

Example XML Response Body

Note that the structure of the XML response is slightly different than the json response, in that the name/value pairs are included in a nameValue element.

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
    <nameValue>
      <errorDetails>
```

```
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Delete Custom Document Fields from an Envelope Document

This deletes custom document fields for an existing envelope document.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/fields

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
documentFields	Yes	documentField	The array of name-value custom data strings to be deleted from a document. Custom document field information is returned in the status, but otherwise is not used by DocuSign. The array contains the elements: name – A string that can be a maximum of 50 characters. value – A string that can be a maximum of 200 characters. **IMPORTANT:** If using xml, the name/value pair is contained in a nameValue element.

Example JSON Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}/fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
```

```
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Example XML Request Body

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Response

The response returns success or failure and any error messages. The response also returns the list of custom document field name-value pairs that were removed from the document.

Examples of the response body for json and XML are shown below.

Example JSON Response Body

```
{
  "documentFields": [
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    },
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    }
  ]
}
```

Example XML Response Body

Note that the structure of the XML response is slightly different than the json response, in that the name/value pairs are included in a nameValue element.

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Get Envelope Certificate

This retrieves a PDF document containing the certificate of completion for the envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/certificate

Optional additions: language={string}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID. The following optional query string can be added to the request.

Name	Reqd?	Schema Type	Description
language	No	String	Sets the language for the Certificate of Completion in the response. The supported languages, with the language value shown in parenthesis, are: Chinese Simplified (zh_CN), , Chinese Traditional (zh_TW), Dutch (nl), English US (en), French (fr), German (de), Italian (it), Japanese (ja), Korean (ko), Portuguese (pt), Portuguese (Brazil) (pt_BR), Russian (ru), Spanish (es).

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/certificate

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a PDF version of the envelope certificate as a byte stream.

Get Envelope Documents and Certificate

This retrieves a PDF containing the combined content of all documents and the certificate. If the account has the Highlight Data Changes feature enabled, there is an option to request that any changes in the envelope be highlighted.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/combined

Optional additions: certificate={true or false}, encrypted={true or false}, include_metadata={true or false}, language={string}, show_changes={true}, watermark={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID. The following optional query strings can be added to the request.

Name	Reqd?	Schema Type	Description
certificate	No	Boolean	Optional. This option can remove the envelope signing certificate from the download.
encrypted	No	Boolean	When true the PDF bytes returned in the response are encrypted for all the key managers configured on your DocuSign account. The documents can be decrypted with the KeyManager Decrypt Document API.
include_metadata	No	Boolean	When false, metadata that would invalidate a digital signature in the PDF is excluded from the response.
language	No	String	Sets the language for the Certificate of Completion in the response. The supported languages, with the language value shown in parenthesis, are: Chinese Simplified (zh_CN), , Chinese Traditional (zh_TW), Dutch (nl), English US (en), French (fr), German (de), Italian (it), Japanese (ja), Korean (ko), Portuguese (pt), Portuguese (Brazil) (pt_BR), Russian (ru), Spanish (es).
show_changes	No	Boolean	Optional. When set to true, any changed fields for the returned PDF are highlighted in yellow and optional signatures or initials outlined in red.
watermark	No	Boolean	Optional. If the account has the watermark feature enabled and the envelope is not complete, the watermark for the account is added to the PDF documents. This option can remove the watermark.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/documents/combined

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns all the PDFs in an envelope and a PDF version of the envelope certificate as a byte stream. If show_changes was included, any changed fields are highlighted in the PDF.

Add Email Setting Overrides to an Envelope

This adds email override settings, changing the reply to email address or name or the BCC for email archive information, for the envelope. Note that adding email settings will only affect email communications that occur after the addition was made.

Important: The BCC Email address feature is designed to provide a copy of all email communications for external archiving purposes. DocuSign recommends that envelopes sent using the BCC for Email Archive feature, including the BCC Email Override option, include additional signer authentication options. To send a copy of the envelope to a recipient who does not need to sign, use a Carbon Copies or Certified Deliveries Recipient Type.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/email_settings

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
replyEmailAddressOverride	No	String	The Reply To email used for the envelope. DocuSign will verify a correct email format is used, but does not verify that the email is active. This can be a maximum of 100 characters.
replyEmailNameOverride	No	String	The name associated with the Reply To email address. This can be a maximum of 100 characters.
bccEmailAddresses	No	Array of strings	Only users with canManageAccount setting can use this option. This is an array of up to 5 email addresses the envelope is sent to as a BCC email. email – The BCC email address. DocuSign verifies that the email format is correct, but does not verify that the email is active. This can be a maximum of 100 characters. Using this overrides the BCC for Email Archive information setting for this envelope. Example: if your account has BCC for Email Archive set up for the email address 'archive@mycompany.com' and you send an envelope using the BCC Email Override to send a BCC email to

Name	Reqd?	Type	Description
			‘salesarchive@mycompany.com’, then a copy of the envelope is only sent to the ‘salesarchive@mycompany.com’ email address.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/email_settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "replyEmailAddressOverride": "string",
  "replyEmailNameOverride": "string",
  "bccEmailAddresses": [
    {
      "email": "string"
    },
    {
      "email": "string"
    }
  ]
}
```

Response

The response returns a success or failure and the information added to the envelope and the bccEmailAddressId associated with the bccEmailAddresses.

The following example shows the response json body.

Example Response Body

```
{
  "replyEmailAddressOverride": "string",
  "replyEmailNameOverride": "string",
  "bccEmailAddresses": [
    {
      "bccEmailAddressId": "string",
      "email": "string"
    },
    {
      "bccEmailAddressId": "string",
      "email": "string"
    }
  ]
}
```

Get Email Setting Overrides for an Envelope

This retrieves email override settings for the envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/email_settings

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/email_settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the email setting information for the envelope, including the bccEmailAddressId associated with the bccEmailAddresses.

The following example shows the response json body.

Example Response Body

```
{
  "replyEmailAddressOverride": "string",
  "replyEmailNameOverride": "string",
  "bccEmailAddresses": [
    {
      "bccEmailAddressId": "string",
      "email": "string"
    },
    {
      "bccEmailAddressId": "string",
      "email": "string"
    }
  ]
}
```

Modify Email Setting Overrides for an Envelope

This modifies existing email override settings for the envelope. Note that modifying email settings will only affect email communications that occur after the modification was made.

This can also be used to delete an individual email override setting by using an empty string for the value to be deleted.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/email_settings

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
replyEmailAddressOverride	No	String	The Reply To email used for the envelope. DocuSign will verify a correct email format is used, but does not verify that the email is active. This can be a maximum of 100 characters.
replyEmailNameOverride	No	String	The name associated with the Reply To email address. This can be a maximum of 100 characters.
bccEmailAddresses	No	Array of strings	Only users with canManageAccount setting can use this option. This is an array of up to 5 email addresses the envelope is sent to as a BCC email. bccEmailAddressId – This is the DocuSign generated identification for the BCC email address. It is required to change the BCC email address. It can be determined using the GET. email – The BCC email address. DocuSign verifies that the email format is correct, but does not verify that the email is active. This can be a maximum of 100 characters. Using this overrides the BCC for Email Archive information setting for this envelope. Example: if your account has BCC for Email Archive set up for the email address 'archive@mycompany.com' and you send an envelope using the BCC Email Override to send a BCC email to 'salesarchive@mycompany.com', then a copy of the envelope is only sent to the 'salesarchive@mycompany.com' email

Name	Reqd?	Type	Description
			address.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/email_settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "replyEmailAddressOverride": "string",
  "replyEmailNameOverride": "string",
  "bccEmailAddresses": [
    {
      "bccEmailAddressId": "string",
      "email": "string"
    },
    {
      "bccEmailAddressId": "string",
      "email": "string"
    }
  ]
}
```

Response

The response returns a success or failure and the email settings information for the envelope.

The following example shows the response json body.

Example Response Body

```
{
  "replyEmailAddressOverride": "string",
  "replyEmailNameOverride": "string",
  "bccEmailAddresses": [
    {
      "bccEmailAddressId": "string",
      "email": "string"
    },
    {
      "bccEmailAddressId": "string",
      "email": "string"
    }
  ]
}
```

Delete Email Setting Overrides for an Envelope

This deletes all existing email override settings for the envelope. If you want to delete an individual email override setting, use the PUT and set the value to an empty string. Note that deleting email

settings will only affect email communications that occur after the deletion and the normal account email settings are used for future email communications.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/email_settings

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/email_settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a success or failure and any error messages associated with a failure.

Lock an Envelope or Template

This locks the envelope or template, and sets the time until the lock expires, to prevent other users or recipients from accessing and changing the envelope or template.

Note: Users must have envelope locking capability enabled to use this function (userSetting 'canLockEnvelopes' must be true for the user).

URL:

/accounts/{accountId}/envelopes/{envelopeId}/lock

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Note: When locking a template, the templateId is used in place of the envelopeId.

Name	Reqd?	Type	Description
lockDurationInSeconds	No	String	Optional setting to set the time, in seconds, until the lock expires when there is no activity on the envelope. If no value is entered, then the default value of 300 seconds is used. The maximum value is 1,800 seconds. The lock duration can be extended.
lockType	Yes	String	The type of lock being set. The accepted value is: edit
useScratchPad	No	Boolean	Reserved for future use. Optional setting that shows if a scratchpad is used for editing information.
lockedByApp	No	String	Optional setting that provides an easily understood name of the application that is locking the envelope.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/lock

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    “lockDurationInSeconds”: “string”,
    “lockType”: “string”,
    “useScratchPad”: “boolean”,
    “lockedbyApp”: “string”
}
```

Response

The response returns the lock information and a lock token that is used to in the X-DocuSign-Edit header to access and make changes to the locked envelope.

```
{
    "lockedByUser": {
           "userName": "string",
           "email": "string",
           "userId": "string"
    },
    "lockedByApp": "string",
    "lockedUntilDateTime": "dateTime",
    "lockDurationInSeconds": "string",
```

```
    "lockType": "string",
    "useScratchPad": "boolean",
    "token": "string"
}
```

The X-DocuSign-Edit header must be included in all subsequent envelope or template calls to show the user is the owner of the envelope lock.

The header structure for JSON is:

```
X-DocuSign-Edit: {"LockToken":"{token-from-response}", "LockDurationInSeconds":"600"}
```

The header structure for XML is:

```
X-DocuSign-Edit = <DocuSignEdit><LockToken>{token from response}</LockToken>
<LockDurationInSeconds>600</LockDurationInSeconds></DocuSignEdit>
```

Get Envelope or Template Lock Information

This returns general information about the envelope lock.

If the call is made by the lockedByUser and the request has the same integrator key as original, then the X-DocuSign-Edit header and additional lock information is included in the response. This allows users to recover a lost editing session token and the X-DocuSign-Edit header.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/lock

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope or template ID.

Note: When working with a template, the templateId is used in place of the envelopeId.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/lock

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

X-DocuSign-Edit: {"LockToken":"{token-from-response}", "LockDurationInSeconds":"600"}
```

Response

The response returns the lock information. If the call is made by the lockedByUser and the request has the same integrator key as original, then the X-DocuSign-Edit header and additional lock information is included in the response. This allows users to recover a lost editing session token and the X-DocuSign-Edit header.

Example Response (standard)

```
{
       "lockedByUser": {
              "userName": "string",
              "email": "string",
              "userId": "string"
       },
       "lockedByApp": "string",
       "lockedUntilDateTime": "dateTime",
       "lockType": "string"
}
```

Example Response (for lockedByUser)

```
X-DocuSign-Edit: {"token": "token_string", "lockDurationInSeconds": "string"}
{
       "lockedByUser": {
              "userName": "string",
              "email": "string",
              "userId": "string"
       },
       "lockedByApp": "string",
       "lockedUntilDateTime": "dateTime",
       "lockDurationInSeconds": "string",
       "lockType": "string",
       "useScratchPad": "boolean",
       "token": "string"
}
```

Update Envelope or Template Lock

This is used to extend the lock duration time or update the lockedByApp information. The user and integrator key must match the lockByUser user and integrator key information and the X-DocuSign-Edit header must be included or an error will be generated.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/lock

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The only required parameter is the envelope or template ID.

Note: When working with a template, the templateId is used in place of the envelopeId.

Name	Reqd?	Type	Description
lockDurationInSeconds	No	String	Optional setting to set the time, in seconds, until the lock expires when there is no activity on the envelope. If no value is entered, then the default value of 300 seconds is used. The maximum value is 1,800 seconds. The lock duration can be extended.
lockedByApp	No	String	Optional setting that provides an easily understood name of the application that is locking the envelope.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/lock

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integ
ratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

X-DocuSign-Edit: {"LockToken":"{token-from-response}", "LockDurationInSeconds":"600"}

{
       "lockDurationInSeconds": "string",
       "lockedbyApp": "string"
}
```

Response

The response returns the same information as locking the envelope or template.

The following example shows the response json body.

Example Response Body

```
{
    "lockedByUser": {
           "userName": "string",
           "email": "string",
           "userId": "string"
    },
    "lockedByApp": "string",
    "lockedUntilDateTime": "dateTime",
    "lockDurationInSeconds": "string",
    "lockType": "string",
    "useScratchPad": "boolean",
    "token": "string"
```

```
}
```

Delete Envelope or Template Lock

This removes the lock from the envelope or template. The X-DocuSign-Edit header must be included in the request.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/lock

Optional addition: resend_envelope={true or false}

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

The only required parameter is the envelope or template ID, the following optional query string can be added to the request.

Note: When working with a template, the templateId is used in place of the envelopeId.

Name	Reqd?	Type	Description
resend_envelope	No	Boolean	Optional setting, when true the envelope is sent to all recipients in the current routing order. Note that if automatic reminders are enabled for the envelope, then resending the envelope resets the automatic reminder date and the resend date is used for determining when to send reminder messages.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/lock?resend_envelope=true

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

X-DocuSign-Edit: {"LockToken":"{token-from-response}", "LockDurationInSeconds":"600"}
```

Response

The response returns a success or failure.

Get Envelope Notification Information

This returns the reminder and expiration information for the envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/notification

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/notification

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the expiration and reminder settings for the envelope.

The following example shows the response json body.

Example Response Body

```
{
    "expirations": {
           "expireAfter": "String content",
           "expireEnabled": "String content",
           "expireWarn": "String content"
    },
    "reminders": {
           "reminderDelay": "String content",
           "reminderEnabled": "String content",
           "reminderFrequency": "String content"
    }
}
```

Get Envelope Recipient Status

This returns the information and status for all recipients of a single envelope and identifies the current routing order. The current routing order is a number that matches up to the routingOrder for envelope recipients, which shows that the envelope has been sent to a recipient, but the recipient has not completed their actions.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients

Optional query items: include_tabs={true or false}, include_extended={true or false}, include_anchor_tab_locations={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID. There are no required parameters, but the following optional query parameters can be added.

Name	Reqd?	Type	Description
include_tabs	No	Boolean	When true the tab information associated with the recipient is included in the response.
include_extended	No	Boolean	When true the extended properties are included in the response.
include_anchor_tab_locations	No	Boolean	When true and include_tabs=true, all tabs with anchor tab properties are included in the response.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the recipient types and current routing order. The recipient types includes the recipient name, email, ID, recipient type, routing order, tab information, delivery method,

authentication status (including the date/time and result of the authentication) status (including the date/time of the status changes) and, if the recipient status is declined and a reason is required, a decline reason added by the recipient. Additionally, the tab information for Company, Date, Email, Number, SSN, Text, Title, and Zip tabs includes the original value (originalValue) of the tab when it was sent to the recipient.

The following example shows the response json body.

Example Response Body

```
{
    "agents":[]
    "carbonCopies":[],
    "certifiedDeliveries":[],
    "currentRoutingOrder":"String content",
    "editors":[],
    "inPersonSigners":[],
    "intermediaries":[],
    "recipientCount":"String content",
    "signers":[{
           "deliveredDateTime": "String content",
           "recipientAuthenticationStatus":{
                  "(authentication status result)":
                         "eventTimestamp":"String content"
                         "status":"String content"
                  }
           },
           "recipientId": "String content",
           "requireIdLookup": "String content",
           "roleName":"String content",
           "routingOrder": "String content",
           "signedDateTime": "String content",
           "status": "String content",
           "email": "String content",
           "name": "String content",
           "deliveryMethod": "String content"
    }]
}
```

Add Recipients to an Envelope

This is used to add one or more recipeints to an envelope.

> **Note:** For “In Process” envelopes (one that has been sent and is not completed or voided), an email will be sent to a new recipient when they are reached in the routing order. If the new recipient’s routing order is before or the same as the envelope’s next recipient, an email is only sent if the optional “resend_envelope” query string is set to true.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients

Optional query string: resend_envelope={true or false}

Note that if automatic reminders are enabled for the envelope, then resending the envelope resets the automatic reminder date and the resend date is used for determining when to send reminder messages.

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

The parameters used to add recipients are the same as those used in an envelope. See the Recipient Types and Parameters for more information.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

    "signers":[{
           "accessCode":"String content",
           "addAccessCodeToEmail":"String content",
           "clientUserId":"String content",
           "customFields":["String content"],
           "emailNotification":{
                  "emailBody":"String content",
                  "emailSubject":"String content",
                  "supportedLanguage":"String content"
           },
           "idCheckConfigurationName":"String content",
           "inheritEmailNotificationConfiguration":"String content",
           "note":"String content",
           "phoneAuthentication":{
                  "recipMayProvideNumber":"String content",
                  "recordVoicePrint":"String content",
                  "senderProvidedNumbers":["String content"],
                  "validateRecipProvidedNumber":"String content"
           },
           "recipientAttachments":[{
                  "attachmentType":"String content",
                  "label":"String content"
           }],
           "recipientId":"String content",
           "requireIdLookup":"String content",
           "roleName":"String content",
           "routingOrder":"String content",
           "socialAuthentications":[{
                  "authentication":"String content"
           }],
           "email":"String content",
           "name":"String content",
           "defaultRecipient":"String content",
           "signInEachLocation":"String content",
           "signatureInfo":{
                  "fontStyle":"String content",
                  "signatureInitials":"String content",
                  "signatureName":"String content"
```

```
            }
    }]
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed. The response contains a recipient structure similar to that input. If an error occurred during the POST operation for any of the recipients, that recipient will contain an error node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "agents":[{
    }]
    "carbonCopies":[{
    }]
    "certifiedDeliveries":[{
    }]
    "editors":[{
    }]
    "agents":[{
    }]
    "inPersonSigners":[{
           "accessCode":"String content",
           "note":"String content",
           "recipientId":"String content",
           "requireIdLookup":"String content",
           "roleName":"String content",
           "routingOrder":"String content",
           "signerName":"String content",
           "status":"Created",
           "email":"String content",
           "name":"String content"
    }]
    "intermediaries":[{
    }]
    "signers":[{
           "accessCode":"String content",
           "note":"String content",
           "recipientId":"String content",
           "requireIdLookup":"String content",
           "roleName":"String content",
           "routingOrder":"String content",
           "status":"Created",
           "email":"String content",
           "name":"String content"
    }]
}
```

Delete Recipients from an Envelope

This is used to delete one or more recipients from a draft or sent envelope. Recipients to be deleted are listed in the request, with the recipientId being used as the key for deleting recipients.

If the envelope is “In Process” (has been sent and is not completed or voided), recipients that have completed their actions cannot be deleted.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

The only required parameter is recipientId.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "signers":[{
           "recipientId":"String content",
    }]
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed. The response contains a recipient structure similar to that input. If an error occured during the DELETE operation for any of the recipients, that recipient will contain an error node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "agents":[{
    }]
    "carbonCopies":[{
    }]
    "certifiedDeliveries":[{
    }]
    "editors":[{
    }]
    "agents":[{
    }]
    "inPersonSigners":[{
    }]
    "intermediaries":[{
    }]
    "signers":[{
           "recipientId":"String content",
```

```
            "routingOrder":"String content",
            "status":"Deleted",
            "email":"String content",
            "name":"String content"
    }]
}
```

Modify or Correct and Resend Recipient Information

This lets you modify recipients in a draft envelope or correct recipient information for an in process envelope.

For draft envelopes, you can edit: email, userName, routingOrder, faxNumber, deliveryMethod, accessCode and requireIdLookup.

> **Note:** If you send information for a recipient that does not already exist in a draft envelope, the recipient will be added to the envelope (similar to the POST).

Once an envelope has been sent, you can only edit: email, userName, signerName, routingOrder, faxNumber, and deliveryMethod. You can also select to resend an envelope by using the resend_envelope option.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients

Optional addition: resend_envelope {true or false}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The parameters used to modify recipients are the same as those used in an envelope. See the Recipient Types and Parameters for more information. The resend_envelope flag is only used to resend an In Process envelope.

Name	Reqd?	Type	Description
resend_envelope	No	String	True or false setting that defaults to false. Setting this to true will resend the envelope to the recipient. Note that if automatic reminders are enabled for the envelope, then resending the envelope resets the automatic reminder date and the resend date is used for determining when to send reminder messages.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "signers" :
    [
           {
                  "email": "mike.roseleip@docusign.com",
                  "name": "Mike Roseleip",
                  "recipientId": "1"
           }
    ]
}
```

Response

The response returns if the correction was successful.

The following example shows the response json body.

Example Response Body

```
{
    "recipientUpdateResults": [{
           "errorDetails": {
                  "errorCode": "SUCCESS",
                  "message": ""
           },
           "recipientId": "1"
    }]
}
```

Add or Replace an Envelope Bulk Recipient File

This adds or replaces a bulk recipient file to a draft envelope. The Content-Type supported for uploading a bulk recipient file is CSV (text/csv).

The REST API does not support modifying individual rows or values in the bulk recipients file. It only allows the entire file to be added or replaced with a new file.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/bulk_recipients

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The only required parameters are the envelope ID and recipient ID.

The parameters listed below are the supported fields for the csv file being added to the envelope. The first row of the csv file is the header row that must have the field names for the file. Each subsequent row represents a unique recipient with the information for that recipient.

There can be a maximum of 1,000 rows in the bulk recipient csv file.

If the value you are adding has a comma or double-quotation marks (“), the value must be enclosed in double-quotation marks (“). Example: if you have a Text tab with a tabLabel of Group and one of the entries for a recipient is Inside Sales, NE, the would use “Inside Sales, NE” in the Group column for that recipient.

Name	Reqd?	Type	Description
Name	Yes	String	The recipient’s name. This can be a maximum of 50 characters.
Email	Yes	Sting	The recipient’s email address. This can be a maximum of 100 characters.
Note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will also display in the signing UI. This can be a maximum of 1000 characters.
AccessCode	No	String	If a value is provided, the recipient must enter the value as the access code to view and sign the envelope. This can be a maximum of 50 characters and must conform to account’s access code format setting. If blank, but the signer accessCode is specified in the envelope, then that value is used. If blank and the signer accessCode is not specified, then access code is not required.
Identification	No	String	Specifies the authentication check used for the signer. If blank then no authentication check is required for the signer. Only one value can be used in this field. The acceptable values are: • KBA: Enables the normal ID check authentication set up for your account. • Phone: Enables phone authentication. • SMS: enables SMS authentication.

Name	Reqd?	Type	Description
PhoneNumber	No	String	This is only used if the **Identification** field has **Phone** or **SMS**. The value for this field can be a valid telephone number or, if Phone, usersupplied (SMS authentication cannot use a user supplied number). Parenthesis and dashes can be used in the telephone number. If "usersupplied" is used, the signer supplies his or her own telephone number.
{tabLabel}	No	String	This is used to populate recipient tabs with information. This allows each bulk recipient signer to have different values for their associated tabs. Any number of tabLabel columns can be added to the bulk recipient file. The information used in the bulk recipient file header must be the same as the tabLabel for the tab. The values entered in this column are automatically inserted into the corresponding tab for the recipient in the same row. Note that this option cannot be used for tabs that do not have data or that are automatically populated data such as Signature, Full Name, Email Address, Company, Title, and Date Signed tabs.

Example Bulk Recipient CSV File

```
Name,Email,Note,AccessCode,Identification,PhoneNumber,address1
David Jones,david.jones@yahoo.com,Here is the document we discussed.,,ID Check,,123 Main
St
Kevin Smith,kevinmith@yahoo.com,,2243,,,697 My Way
Elisabeth Bozick,elisabeth.bozick@yahoo.com,,,phone,usersupplied,827 1st Ave
```

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}//accounts/{accountId}/envelopes/
{envelopeId}/recipients/{recipientId}/bulk_recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Content-Type: application/json
Content-Disposition: file;filename={file name};fileExtension=.csv
```

Response

The response returns the recipient information for each row of the bulk recipient csv file, the total number of recipients in the file, and the bulkRecipientUri.

Example Response

```
{
  "bulkRecipients": [
    {
      "rowNumber": "string",
      "email": "string",
      "name": "string",
      "note": "string",
      "accessCode": "string",
      "identification": "string",
      "phoneNumber": "string",
      "tabLabels": [
        {
          "name": "string,"
          "value": "string"
        }
      ]
    },
    {
      "rowNumber": "string",
      "email": "string",
      "name": "string",
      "note": "string",
      "accessCode": "string",
      "identification": "string",
      "phoneNumber": "string",
      "tabLabels": [
        {
          "name": "string,
          "value": "string"
        }
      ]
    }
  ],
  "bulkRecipientsCount": "string",
  "bulkRecipientsUri": "string"
}
```

Retrieve Envelope Bulk Recipient File

This is used to retrieve the bulk recipient file information for an envelope with a bulk recipient.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/bulk_recipients

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameters are the envelope ID and recipient ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}//accounts/{accountId}/envelopes/
{envelopeId}/recipients/{recipientId}/bulk_recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integ
ratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the recipient information for each row of the bulk recipient csv file, the total number of recipients in the file, and the bulkRecipientUri.

The following example shows the response json body.

Example Response Body

```
{
  "bulkRecipients": [
    {
      "rowNumber": "string",
      "email": "string",
      "name": "string",
      "note": "string",
      "accessCode": "string",
      "identification": "string",
      "phoneNumber": "string",
      "tabLabels": [
        {
          "name": "string,"
          "value": "string"
        }
      ]
    },
    {
      "rowNumber": "string",
      "email": "string",
      "name": "string",
      "note": "string",
      "accessCode": "string",
      "identification": "string",
      "phoneNumber": "string",
      "tabLabels": [
        {
          "name": "string,
          "value": "string"
        }
      ]
    }
  ],
  "bulkRecipientsCount": "string",
  "bulkRecipientsUri": "string"
}
```

Delete Envelope Bulk Recipient File

This removes the bulk recipient file from an envelope. This cannot be used if the envelope has been sent.

After using this, the bulkRecipientsUri field will not be returned in subsequent GET calls for the envelope, but the recipient will remain as a bulk recipient

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/bulk_recipients

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

The only required parameters are the envelope ID and recipient ID.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}//accounts/{accountId}/envelopes/
{envelopeId}/recipients/{recipientId}/bulk_recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integ
ratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a success or failure.

Set Initials Image for Accountless Signer

This sets the initials image for an accountless signer. The supported image formats for this file are: gif, png, jpeg, and bmp. The file size must be less than 200K.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/initials_image

Note: For the Authentication/Authorization for this call, the credentials must match the sender of the envelope, the recipient must be an accountless signer or inperson signer, the Account has CanSendEnvelope enabled and the SendingUser does not have ExpressSendOnly enabled.

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

No additional parameters are required.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
recipients/{recipientId}/initials_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

Content-Type: image/gif

<image removed>
```

Response

The response returns a success or failure.

Get Signature Information for Accountless Signer

This returns the structure of a single signature with a known signature name.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/signature

Note: For the Authentication/Authorization for this call, the credentials must match the sender of the envelope, the recipient must be an accountless signer or in person signer, the Account has CanSendEnvelope enabled and the SendingUser does not have ExpressSendOnly enabled.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
recipients/{recipientId}/signature

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns information for the request recipient signature. The following example shows the response json body.

Example Response Body

```
{
    "adoptedDateTime":"String content",
    "createdDateTime":"String content",
    "initials150ImageId":"String content",
    "initialsImageUri":"String content",
    "signature150ImageId":"String content",
    "signatureFont":"String content",
    "signatureId":"String content",
    "signatureImageUri":"String content",
    "signatureInitials":"String content",
    "signatureName":"String content",
    "signatureType":"String content"
}
```

Set Signature Image for Accountless Signer

This sets the signature image for an accountless signer. The supported image formats for this file are: gif, png, jpeg, and bmp. The file size must be less than 200K.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/signature_image

Note: For the Authentication/Authorization for this call, the credentials must match the sender of the envelope, the recipient must be an accountless signer or in person signer, the Account has CanSendEnvelope enabled and the SendingUser does not have ExpressSendOnly enabled.

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

No additional parameters are required.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
recipients/{recipientId}/signature_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

Content-Type: image/gif

<image removed>
```

Response

The response returns a success or failure.

Add Tabs for a Recipient

This adds one or more tabs for a recipient. It can be used to add tabs to a draft envelope or to add tabs to an in process envelope.

To add tabs to an in process envelope, the account must have envelope correct enabled, the envelope status must be “Sent” and the recipient status must be “Created” or “Sent.” If these conditions are met, the POST opens the envelope in correct mode, adds the tabs and exits correct mode.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

The parameters used to add tabs are the same as those used in an envelope. See the Tab Types and Parameters section for more information about the tabs.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
recipients/{recipientId}/tabs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
```

```
    "approveTabs":[{
          <Tab information removed>
    }],
    "titleTabs":[{
          <Tab information removed>
    }],
    "signHereTabs":[{
          <Tab information removed>
    }]
}
```

Response

The response returns the success or failure of each tab being added to the envelope and the envelope ID. Failed operations on array elements will add the “errorDetails” structure containing an error code and message. If “errorDetails” is null, then the operation was successful for that item.

Get Tab Information for a Recipient

This retrieves information about the tabs associated with a recipient in a draft envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs

Optional query item: include_anchor_tab_locations={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query parameters can be added.

Name	Reqd?	Type	Description
include_anchor_tab_locations	No	String	When true all tabs with anchor tab properties are included in the response.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
recipients/{recipientId}/tabs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of tabs associated with the recipient. The tab information includes a tabId that can be used when deleting a tab. Additionally for Company, Date, Email, Number, SSN, Text, Title, and Zip tabs the response includes the original value (originalValue) of the tab when it was sent to the recipient. The following example shows the response json body.

Example Response Body

```
{
    "approveTabs":[{
           <Tab information removed>
    }],
    "titleTabs":[{
           <Tab information removed>
    }],
    "signHereTabs":[{
           <Tab information removed>
    }]
}
```

Delete Tabs for a Recipient

This deletes one or more tabs associated with a recipient in a draft envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
tabId	Yes	String	The unique identifier for the tab. The tabId can be retrieved with the GET call.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
recipients/{recipientId}/tabs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "approveTabs":[{
```

```
            "tabId":"String content"
            },
            {
            "tabId":"String content"
            }
    ],
    "titleTabs":[{
            "tabId":"String content"
    }],
    "signHereTabs":[{
            "tabId":"String content"
    }]
}
```

Response

The response returns the success or failure of each document being added to the envelope and the envelope ID. Failed operations on array elements will add the “errorDetails” structure containing an error code and message. If “errorDetails” is null, then the operation was successful for that item.

Modify Tabs for a Recipient

This modifies one or more tabs for a recipient to a draft envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The parameters used to modify tabs are the same as those used in an envelope, but you can only modify existing tabs and the tabId must be included. See the Tab Types and Parameters section for more information about the tabs.

Name	Reqd?	Type	Description
tabId	Yes	String	The unique identifier for the tab. The tabId can be retrieved with the GET call.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
recipients/{recipientId}/tabs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

```
{
    "approveTabs":[{
           <Tab information removed>
    }],
    "titleTabs":[{
           <Tab information removed>
    }],
    "signHereTabs":[{
           <Tab information removed>
    }]
}
```

Response

The response returns the success or failure of each tab being modified and the envelope ID. Failed operations on array elements will add the “errorDetails” structure containing an error code and message. If “errorDetails” is null, then the operation was successful for that item.

Get List of Templates Used in an Envelope

This returns a list of the templates, name and ID, used in an envelope.

Note: This only returns information for server side templates.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/templates

Optional addition: include=matching_applied

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required, but the following optional query string can be added to the request.

Name	Reqd?	Type	Description
include	No	String	The possible values are: • matching_applied – This returns template matching information for the template.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes
/{envelopeId}/templates?include=matching_applied
```

```
X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of the templates used in an envelope, showing the name, ID, and uri for the template, along with template matching information if the optional query parameter is included. The following example shows the response json body with include=matching_applied.

Example Response Body

```
{
    "templates": [
            {
                    "templateId": "string",
                    "name": "string",
                    "documentId": "string",
                    "documentName": "string",
                    "applied": "string",
                    "templateMatch": {
                            "matchPercentage": "string",
                            "documentStartPage": "string",
                            "documentEndPage": "string"
                    },
                    "uri": "string"
            }
    ]
}
```

Post Envelope Correction

This provides a URL to start the correction view of the DocuSign UI.

Important: iFrames should not be used for embedded operations on mobile devices due to screen space issues. For iOS devices DocuSign recommends using a WebView.

URL:

/{accountId}/envelopes/{envelopeId}/views/correct

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
returnUrl	Yes	String	Identifies the return point after correcting the envelope. **Important** – You must include HTTPS:// in the URL or the redirect might be blocked by some browsers. DocuSign returns to the URL and includes an event parameter that can be used to redirect the recipient to another location. The possible event parameters returned are: • send (user corrects and sends the envelope) • save (user saves the envelpe) • cancel (user cancels the transaction.) • error (there is an error when performing the correct or send) • sessionEnd (the session ends before the user completes another action)
suppressNavigation	No	String	Sets whether the window is displayed with or without dressing.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
views/correct

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "returnUrl":"String content",
    "suppressNavigation":"String content"
}
```

Response

The response returns the correction view url.

The following example shows the response json body.

Example Response Body

```
{
    "url":"String content"
}
```

Post Recipient View

This provides a URL to start the recipient view of the DocuSign UI for a sent envelope. This call cannot be used to view draft envelopes, since those envelopes have not been sent.

The X-Frame-Options property specifies whether the browser should render the transmitted resource within a <frame> or an <iframe>. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames. For more information about the X-Frame-Options header, see IEFT RFC 7034.

Important: iFrames should not be used for embedded operations on mobile devices due to screen space issues. For iOS devices DocuSign recommends using a WebView.

An entry is added into the Security Level section of the DocuSign Certificate of Completion that reflects the “securityDomain” – “authenticationMethod” used to verify the user identity.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/views/recipient

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
clientUserId	Yes	String	A sender created value that shows the recipient is embedded (captive). This can be a maximum of 100 characters.
authenticationMethod	Yes	String	A sender created value that indicates the convention used to authenticate the signer. This information is included in the Certificate of Completion.
assertionId	No	String	A unique identifier of the authentication event executed by the client application.
authenticationInstant	No	DateTime	A sender generated value that indicates the date/time that the signer was authenticated.
securityDomain	No	String	The domain to which the user authenticated.
email	Yes*	String	Specifies the email of the recipient. * If a clientUserId is provided, you can use the userId or email and userName to identify the recipient. If a clientUserId is not provided, then email and userName should be used to identify the recipient.

pingUrl	No	String	This option provides a way for the embedded DocuSign signing session to reset the session timer for the client application. This is the client URL pinged by the DocuSign Signing experience to indicate to the client that Signing is active. You must include HTTPS:// in the URL. A GET is executed against the client. Note that the response from the client is ignored by DocuSign. The intent of the ping is for the client to reset the session timer when the request is received.
pingFrequency	No	String	This is the interval, in seconds, between pings to the pingUrl. This is only used if a pingUrl is specified. This can be 60 to 12000. If no value is specified, a default value of 300 is used.
recipientId	No	String	The recipient ID for the recipient.
returnUrl	Yes	String	The URL the recipient is directed to on certain events. **Important** – You must include HTTPS:// in the URL or the redirect might be blocked by some browsers. DocuSign sends returns to the URL and includes an event parameter that can be used to redirect the recipient to another location. The possible event parameters returned are: • cancel (recipient uses the Finish Later option to exit the envelope without completing signing) • decline (recipient declines signing) • exception (there was a system error during signing) • fax_pending (recipient has fax pending) • id_check_faild (recipient failed an ID check) • session_timeout (session times out. An account can control this timeout using the Signer Session Timeout option) • signing_complete (recipient completes signing) • ttl_expired (the Time To Live token for the envelope has expired; these tokens

			expire after being successfully invoked, after 5 minutes, or if the envelope is voided) • viewing_complete (recipient completes viewing an envelope that is in a read-only/terminal state, such as completed, declined, or voided.)
userId	Yes*	String	Specifies the user ID of the recipient. * If a clientUserId is provided, you can use the userId or email and userName to identify the recipient. If a clientUserId is not provided, then email and userName should be used to identify the recipient.
userName	Yes*	String	Specifies the username of the recipient * If a clientUserId is provided, you can use the userId or email and userName to identify the recipient. If a clientUserId is not provided, then email and userName should be used to identify the recipient.
xFrameOptions	No	String	Indicates whether or not a browser should be allowed to render a page in a frame or iframe. There are three accepted values: • deny - The page cannot be displayed in a frame. • same_origin - The page can only be displayed in a frame on the same origin as the page itself. • allow_from - The page can only be displayed in a frame on the origin specified by xFrameOptionsAllowFromUrl.
xFrameOptionsAllowFromUrl	No	String	Sets the origin for display the page in the frame. If xFrameOptions = allow_from, a value must be provided.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/{envelopeId}/
views/recipient

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
```

```
    "clientUserId":"String content",
    "authenticationMethod": "String content",
    "assertionId": "String content",
    "authenticationInstant": "String content",
    "securityDomain": "String content"
    "email":"String content",
    "pingUrl": "String content",
    "pingFrequency":"String content",
    "recipientId": "String content",
    "returnUrl":"String content",
    "userId":"String content",
    "userName":"String content",
    "xFrameOptions":"String content",
    "xFrameOptionsAllowFromUrl":"String content"
}
```

Response

The response returns the recipient view url.

The following example shows the response json body.

Example Response Body

```
{
    "url":"String content"
}
```

Setting Default Signing Language for Embedded Signing

When using embedded signing functionality, you can set the language initially loaded for the embedded signing experience.

This is done by appending a locale parameter (&locale={languageCode}) to the URL provided in the response to a POST recipient view request before loading it in your iframe or web control/view.

Note that even when add the locale parameter there is still an order of precedence for determining the language used in the signing experience:

- If the sender specifies a language for the recipient when the envelope is sent, that language is used.
- If the signing integrator specifies a language in the url locale parameter, that language is used.
- If the signer has a DocuSign account, the user language preference is used.
- For returning signers, the preference saved from a previous signing is used.
- If none of the above conditions are met, the selected browser language is used.

Examples: If the response to a POST recipient view request is:

```
{
  "url": "https://demo.docusign.net/signing/startinsession.aspx?t=d1cf42f2-30b6-499b-
ab54-058fbf438103"
}
```

Then first example would open the signing experience in French, the second would use French (Canada), and the third would use Portuguese (Brazil).

```
https://demo.docusign.net/signing/startinsession.aspx?t=d1cf42f2-30b6-499b-ab54-
058fbf438103&locale=fr
```

```
https://demo.docusign.net/signing/startinsession.aspx?t=d1cf42f2-30b6-499b-ab54-
058fbf438103&locale=fr_CA
```

```
https://demo.docusign.net/signing/startinsession.aspx?t=d1cf42f2-30b6-499b-ab54-
058fbf438103&locale=pt_BR
```

Post Sender View

This provides a URL to start the sending view of the DocuSign UI. This is a one-time use login token that allows the user to be placed into the DocuSign sending view.

Upon sending completion, the user is returned to the return URL provided by the API application.

This should only be used for draft envelopes. If you try to open this view for an envelope which has been already sent or is completed, you are automatically redirected to the returnURL with the ?event=Send appended to the URL.

Note that when using the response URL you can set the default UI page shown to a user by using the send query string parameter on the end of the URL. Setting send=0 starts the user on the prepare page where the sender can add documents and recipients, while setting send=1 starts the user on the tagging page. The setting defaults to the prepare page.

Important: iFrames should not be used for embedded operations on mobile devices due to screen space issues. For iOS devices DocuSign recommends using a WebView.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/views/sender

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
returnUrl	Yes	String	Identifies the return point after sending the envelope. **Important** – You must include HTTPS:// in the URL or the redirect might be blocked by some browsers. DocuSign returns to the URL and includes an event parameter that can be used to redirect the recipient to another location. The possible event parameters returned are: • send (user sends the envelope)

Name	Reqd?	Type	Description
			• save (user saves the envelpe) • cancel (user cancels the sending transaction. No envelopeId is returned in this case.) • error (there is an error when performing the send) • sessionEnd (the sending session ends before the user completes another action)

Example Request Body

```
POST https://{SERVER}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
{envelopeId}/views/sender

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "returnUrl":"String content"
}
```

Response

The response returns the sender view URL.

When using the response URL you can set the default UI page shown to a user by using the send query string parameter on the end of the URL. Setting send=0 starts the user on the prepare page where the sender can add documents and recipients, while setting send=1 starts the user on the tagging page. The setting defaults to the prepare page.

Example: For the URL - `https://docusign.net/Member/StartInSession.aspx?send=1`
The user will start on the tagging page of the sending view.

The following example shows the response json body.

Example Response Body

```
{
    "url":"String content"
}
```

Post Edit View

This is the same as Post Sender View. Refer to that section for more information.

Get Envelope Status for more than one envelope

This returns envelope status for the requested envelopes.

URL:

/accounts/{accountId}/envelopes/status

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
envelopeIds	Yes	String	The envelope IDs for the envelopes being requested.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/envelopes/
status?envelope_ids=request_body

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
{
    "envelopeIds":[
           "String content",
           "String content"
    ],
}
```

Response

The response returns the envelope status information for the requested envelopes.

The following example shows the response json body.

Example Response Body

```
{
    "envelopes": [{
           "certificateUri": "String content",
           "customFieldsUri": "String content",
           "documentsCombinedUri": "String content",
           "documentsUri": "String content",
           "envelopeId": "String content",
           "envelopeUri": "String content",
           "notificationUri": "String content",
           "recipientsUri": "String content",
           "status": "String content",
           "statusChangedDateTime": "String content"
    },
    {
           "certificateUri": "String content",
           "customFieldsUri": "String content",
```

```
            "documentsCombinedUri": "String content",
            "documentsUri": "String content",
            "envelopeId": "String content",
            "envelopeUri": "String content",
            "notificationUri": "String content",
            "recipientsUri": "String content",
            "status": "String content",
            "statusChangedDateTime": "String content"
    }],
    "resultSetSize": "String content"
}
```

Get Folder List

Returns a list of the folders for the account, including the folder hierarchy. There is an option to include the list of template folders and templates.

URL:

/accounts/{accountId}/folders

Optional query string: template={string}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required, but the following optional parameter can be added to the request.

Name	Reqd?	Type	Description
Template	No	String	The two possible values are “include” or “only.” • include - The folder list will return normal folders plus template folders. • only - Only the list of template folders are returned.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/folders

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of folders for the account. The folder list includes the folder ID, the folder name, the name, user ID and email of the folder's owner, the folder type and uri. If the folder is a sub-folder of another folder, the parent folder information is returned. If the template query is included, the template folder structure, along with the templates within each folder, is also returned.

The following example shows the response json body.

Example Response Body

```
{
    "folders": [
            {
                    "ownerUserName": "String content",
                    "ownerEmail": "String content",
                    "ownerUserId": "String content",
                    "type": "String content",
                    "name": "String content",
                    "uri": "String content",
                    "parentFolderId": "String content",
                    "parentFolderUri": "String content",
                    "folderId": "String content",
                    "filter": {
                            "actionRequired": "String content",
                            "expires": "String content",
                            "isTemplate": "String content",
                            "status": "String content",
                            "fromDateTime": "String content",
                            "toDateTime": "String content",
                            "searchTarget": "String content",
                            "searchText": "String content",
                            "folderIds": "String content",
                            "orderBy": "String content",
                            "order": "String content"
                    }
            },
    ]
}
```

Get Folder Envelope List

Returns a list of the envelopes in the specified folder. You can narrow the query by adding some optional items.

URL:

/accounts/{accountId}/folders/{folderId}

Optional query additions: start_position={ integer}, from_date = {date/time}, to_date= {date/time}, search_text={text}, status={envelope status}, owner_name={username}, owner_email={email}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required, but the following optional query strings can be added to narrow the response results.

Name	Reqd?	Type	Description
start_position	No	Integer	The position of the folder items to return. This is used for repeated calls, when the number of envelopes returned is too much for one return (calls return 100 envelopes at a time). The default value is 0.
from_date	No	Date/time	Only return items on or after this date. If no value is provided, the default search is the previous 30 days.
to_date	No	Date/time	Only return items up to this date. If no value is provided, the default search is to the current date.
search_text	No	String	The search text used to search the items of the envelope. The search looks at recipient names and emails, envelope custom fields, sender name, and subject.
status	No	Envelope status	The current status of the envelope. If no value is provided, the default search is all/any status.
owner_name	No	username	The name of the folder owner.
owner_email	No	email	The email of the folder owner.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/folders/{folderId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of envelope in the folder that meet the query information.

The following example shows the response json body.

Example Response Body

```
{
    "resultSetSize": "String content",
    "startPosition": "String content",
    "endPosition": "String content",
```

```
    "totalSetSize": "String content",
    "previousUri": "String content",
    "nextUri": "String content",
    "folderItems": [
           {
                  "ownerName": "String content",
                  "envelopeId": "String content",
                  "envelopeUri": "String content",
                  "status": "String content",
                  "senderName": "String content",
                  "senderEmail": "String content",
                  "createdDateTime": "String content",
                  "sentDateTime": "String content",
                  "completedDateTime": "String content",
                  "subject": "String content",
                  "templateId": "String content",
                  "name": "String content",
                  "shared": "String content",
                  "password": "String content",
                  "description": "String content",
                  "lastModified": "String content",
                  "pageCount": "String content",
                  "uri": "String content",
                  "customFields": [
                         {
                                "fieldId": "String content",
                                "name": "String content",
                                "show": "String content",
                                "required": "String content",
                                "value": "String content",
                         },
                         {
                                "fieldId": "String content",
                                "name": "String content",
                                "show": "String content",
                                "required": "String content",
                                "value": "String content",
                         }
                  ]
           },
    ]
}
```

Move Envelope

This moves an envelope from its current folder to selected folder.

Note: This can be used to delete envelopes by using “recyclebin” as folderId. Placing an in process envelope (envelope status of sent or delivered) in the recycle bin will void the envelope. This can also be used to delete templates by using the templateId in place of the envelopeId and using “recyclebin” as folderId.

URL:

/accounts/{accountId}/folders/{folderId}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
envelopeIds	Yes	String	The envelope ID for the envelope that is being moved.
fromFolderId	No	String	The folder ID the envelope is being moved from.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/folders/{folderId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "envelopeIds": ["String"],
    "fromFolderId": "FolderId"
}
```

Response

The response returns if the move was a success or failure.

Add a New Group

This adds one or more groups for the account.

Groups can be used to help manage users by associating users with a group. A group can be associated with a Permission Profile, which sets the user permissions for users in that group without having to set the userSettings for each user. You are not required to set Permission Profiles for a group, but this makes it easier to manage user permissions for a large number of users. Groups can also be used with template sharing to limit user access to templates.

URL:

/accounts/{accountId}/groups

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
groupName	Yes	String	The name for the new user group.
permissionProfileId	No	String	The ID number of the permission profile that the group is associated with. See Get a List of Permission Profiles for information on retrieving a list of IDs.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "groups":[{
           "groupName":"String content",
           "permissionProfileId":"String content"
    }]
}
```

Response

The response returns if the API execution is successful (201 – Created) or failed. The response contains a group structure similar to the request and includes the groupId assigned to the group. If an error occurred during the POST operation for any of the groups, that group will contain an errorDetails node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "groups":[{
           "errorDetails":{
                  "errorCode":"String content",
                  "message":"String content"
           },
           "groupId":"String content",
           "groupName":"String content",
           "groupType":"String content",
           "permissionProfileId":"String content"
    }]
}
```

Get Group Information

This returns information about groups associated with the account.

URL:

/accounts/{accountId}/groups

Optional query parameters: start_position={integer}, count={integer}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query strings can be added to the request.

Name	Reqd?	Type	Description
start_position	No	Integer	Starting value for the list.
count	No	Integer	Number of records to return. The number must be greater than 1 and less than or equal to 100.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of groups along with the added information about the result set.

Name	Description
resultSetSize	Total number groups returned in the result set.
totalSetSize	Total number groups for the account.
startPosition	Starting position of the current result set.
endPosition	The last position in the result set.
nextUri	Provides the Uri to the next chunk of records based on the request. If the endPosition is the entire results of the search, this is null.
previousUri	Provides the Uri to the previous chunk of records based on the request. If this response is the first response for the search, this is null.

The following example shows the response json body.

Example Response Body

```
{
    "groups":[
            {
                    "groupId":"String content",
                    "groupName":"String content",
                    "groupType":"String content",
                    "permissionProfileId":"String content",
                    "users":[
                            {
                                    "userName": "string",
                                    "email": "string",
                                    "userId": "string",
                                    "userType": "string",
                                    "userStatus": "string",
                                    "uri": "string"
                            },
                            {
                                    "userName": "string",
                                    "email": "string",
                                    "userId": "string",
                                    "userType": "string",
                                    "userStatus": "string",
                                    "uri": "string"
                            },
                    ]
            }
    ],
    "resultSetSize": "String Content",
    "totalSetSize": "String Content",
    "startPosition": "String Content",
    "endPosition": "String Content",
    "nextUri": "String Content",
    "previousUri": "String Content"
}
```

Modify Group Information

This lets you modify a group name and modify, or set, the permission profile for the group.

URL:

/accounts/{accountId}/groups

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
groupId	Yes	String	The group ID number.
groupName	No	String	The new name for the group.
permissionProfileId	No	String	The ID number of the permission profile that the group is associated with. See Get a List of Permission Profiles for information on retrieving a list of IDs.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "groups":[{
           "groupId":"String content",
           "groupName":"String content",
           "permissionProfileId":"String content"
    }]
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed. The response contains a group structure similar to the request. If an error occurred during the PUT operation for any of the groups, that group will contain an errorDetails node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "groups":[{
           "errorDetails":{
                  "errorCode":"String content",
                  "message":"String content"
           },
           "groupId":"String content",
           "groupName":"String content",
           "groupType":"String content",
           "permissionProfileId":"String content"
    }]
}
```

Delete Group Information

This deletes an existing user group.

URL:

/accounts/{accountId}/groups

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
groupId	Yes	String	The group ID number of the group being deleted.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "groups": [
    {
      "groupId": "string"
    },
    {
      "groupId": "string"
    }
  ]
}
```

Response

The response returns a success or failure and information for the deleted groups.

The following example shows the response json body.

Example Response Body

```
{
  "groups": [
    {
      "groupId": "string",
      "groupName": "string",
      "permissionProfileId": "string",
      "groupType": "string",
    },
    {
      "groupId": "string",
```

```
      "groupName": "string",
      "permissionProfileId": "string",
      "groupType": "string",
    }
  ]
}
```

Get Group Brand ID Information

This returns information about the brands associated with the requested group.

URL:

/accounts/{accountId}/groups/{groupId}/brands

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups/{groupId}/
brands

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the brand information (brandId, brandName, and brandCompany) associated with the group.

The following example shows the response json body.

Example Response Body

```
{
    "brands":[{
           "brandId":"String content",
           "brandName":"String content",
           "brandCompany":"String content"
    },
    {
           "brandId":"String content",
```

```
            "brandName":"String content",
            "brandCompany":"String content"
    }]
}
```

Add Group Brand ID Information

This adds brand information to a group.

URL:

/accounts/{accountId}/groups/{groupId}/brands

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
brandId	Yes	String	The brandId of the brand profile being added to the group.
brandName	Yes	String	The brand name associated with the brand profile.
brandCompany	Yes	String	The brand company associated with the brand profile.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups/{groupId}/
brands

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "brands":[{
            "brandId":"String content",
            "brandName":"String content",
            "brandCompany":"String content"
    }]
}
```

Response

The response returns as success or failure, along with the brand information (brandId, brandName, and brandCompany) added to the group.

The following example shows the response json body.

Example Response Body

```
{
    "brands":[{
           "brandId":"String content",
           "brandName":"String content",
           "brandCompany":"String content"
    }]
}
```

Delete Group Brand ID Information

This removes brand information from the requested group.

URL:

/accounts/{accountId}/groups/{groupId}/brands

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
brandId	Yes	String	The brandId of the brand profile being removed from the group.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups/{groupId}/
brands

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "brands":[{
           "brandId":"String content"
    },
           "brandId":"String content"
```

```
    }]
}
```

Response

The response returns as success or failure, along with the brandIds of the brands removed from the group.

The following example shows the response json body.

Example Response Body

```
{
    "brands":[{
            "brandId":"String content"
    },
    {
            "brandId":"String content"
    }]
}
```

Add Users to a Group

This adds one or more users to an existing group.

URL:

/accounts/{accountId}/groups/{groupId}/users

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
userId	Yes	String	The user ID number for a user being added to the group.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups/{groupId}/
users

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

```
{
    "users":[{
            "userId":"String content",
    }]
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed. The response contains a user structure similar to the request and includes the user changes. If an error occurred during the PUT operation for any of the users, that user will contain an errorDetails node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "users":[{
            "errorDetails":{
                    "errorCode":"String content",
                    "message":"String content"
            },
            "uri":"String content",
            "userId":"String content",
            "userName":"String content",
            "userStatus":"String content",
            "userType":"String content"
    }]
 }
```

Get List of Users in a Group

This returns a list of users in a group.

URL:

/accounts/{accountId}/groups/{groupId}/users

Optional query parameters: start_position={integer}, count={integer}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query strings can be added to the request.

Name	Reqd?	Type	Description

Name	Reqd?	Type	Description
start_position	No	Integer	Starting value for the list.
count	No	Integer	Number of records to return. The number must be greater than 1 and less than or equal to 100.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups/{groupId}/
users

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of users in the group along with information about the result set.

Name	Description
resultSetSize	Total number user items returned in the result set.
totalSetSize	Total number user items in the group.
startPosition	Starting position of the current result set.
endPosition	The last position in the result set.
nextUri	Provides the Uri to the next chunk of records based on the request. If the endPosition is the entire results of the search, this is null.
previousUri	Provides the Uri to the previous chunk of records based on the request. If this response is the first response for the search, this is null.

The following example shows the response json body.

Example Response Body

```
{
    "users":[
            {
                    "userName":"String content",
                    "email":"String content",
                    "userId":"String content",
                    "userStatus":"String content",
                    "userType":"String content",
                    "uri":"String content"
            },
            {
                    "userName":"String content",
                    "email":"String content",
                    "userId":"String content",
                    "userStatus":"String content",
                    "userType":"String content",
                    "uri":"String content"
```

```
            }
    ],
    "resultSetSize": "String content",
    "totalSetSize": "String content",
    "startPosition": "String content",
    "endPosition": "String content",
    "nextUri": "String content",
    "previousUri": "String content"
}
```

Remove Users from a Group

This removes one or more users from a group.

URL:

/accounts/{accountId}/groups/{groupId}/users

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
userId	Yes	String	The user ID number for a user being removed from the group.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/groups/{groupId}/
users

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "users":[{
           "userId":"String content",
    }]
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed. The response contains a user structure similar to the request and includes the user changes. If an error occurred during the DELETE operation for any of the users, that user will contain an errorDetails node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "users":[{
           "errorDetails":{
                  "errorCode":"String content",
                  "message":"String content"
           },
           "uri":"String content",
           "userId":"String content",
           "userName":"String content",
           "userStatus":"String content",
           "userType":"String content"
    }]
 }
```

Get a List of Permission Profiles

This retrieves a list of Permission Profiles. Permission Profiles are a standard set of user permissions that can be applied to individual users or users in a Group. This makes it easier to manage user permissions for a large number of users, without having to change permissions on a user-by-user basis.

Currently Permission Profiles can only be created and modified in the DocuSign console.

URL:

/accounts/{accountId}/permission_profiles

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/permission_profiles

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
```

```
Content-Type: application/json
```

Response

The response returns a list of permission profiles for the account.

The following example shows the response json body.

Example Response Body

```
{
    "permissionProfiles":[
            {
                    "permissionProfileId":"String content",
                    "permissionProfileName":"String content"
            },
            {
                    "permissionProfileId":"String content",
                    "permissionProfileName":"String content"
            },
    ]
}
```

Get Recipient Names

This returns a list of recipients that are available for the given email address.

URL:

/accounts/{accountId}/recipient_names

The query string email = {email} is required.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

Name	Reqd?	Type	Description
email	Yes	String	The email address for the user.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/recipient_names?
email={email}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
```

```
Content-Type: application/json
```

Response

The response returns the recipient names associated with the email is used by more than one user.

Example Response Body

```
{
    "multipleUsers": "false",
    "recipientNames": [
           "Resty B. Tester",
           "Resty Tester",
           "Resty Tester III"
    ],
    "reservedRecipientEmail": "false"
}
```

Get List of Envelopes in Folders

This returns a list of envelopes that match the criteria specified in the query.

URL:

/accounts/{accountId}/search_folders/{search_folder}

Optional query additions: start_position={integer}, count={integer}, from_date={date/time}, to_date={date/time}, order_by={string}, order={string}, include_recipients={true/false}, all

Note: If the userId of the user making the call is the same as the userId for any returned recipient, then the userId is added to the returned information for those recipients.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

Name	Reqd?	Type	Description
search_folder	Yes	String	Specifies the envelope group that is searched by the request. These are logical groupings, not actual folder names. Valid values are listed table below.
start_position	No	Integer	Starting value for the list.
count	No	Integer	Number of records to return in the cache. The number must be greater than 1 and less than or equal to 100.

Name	Reqd?	Type	Description
from_date	No	Date/Time	Start of the date range. If no value is provided, the default search is the previous 30 days.
to_date	No	Date/Time	End of the date range.
order_by	No	String	Column used to sort the list. Valid values are listed in the table below.
order	No	String	Direction order used to sort the list. Valid values are: • asc = ascending sort order (a to z) • desc = descending sort order (z to a)
include_recipients	No	True/False	When true, the recipient information is returned in the response.
all	No	N/A	This returns all envelopes that matches the criteria.

search_folder values

Value	Description
Drafts	Drafts Note that Draft envelopes are only held for 30 days before they are removed from the system.
awaiting_my_signature	Envelopes awaiting the users' signature
completed	Envelopes with the status of "completed".
out_for_signature	Sent envelopes that have not been completed.

order_by values

Value	Description
action_required	Awaiting
created	Date/Time envelope was created
completed	Date/Time envelope was completed
sent	Date/Time envelope was sent
signer_list	List of Signers
status	Envelope status
subject	Envelope subject

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/search_folders/
{search_folder}?include_recipients=true
```

```
X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of envelopes that match the specified criteria.

Note: If the userId of the user making the call is the same as the userId for any returned recipient, then the userId is added to the returned information for those recipients.

Value	Description
endPosition	The last row id in the result set.
folderItems	Collection of folder items in the result set. See the table below for more information on the contents of folder items.
nextUri	Provides the Uri to the next chunk of records based on the search request. If the endPosition is the entire results of the search, this is null.
previousUri	Provides the Uri to the previous chunk of records based on the search request. If this response is the first response for the search, this is null.
resultSetSize	Number of folder items returned in the result set.
startPosition	Starting position of the current result set.
totalRows	Total number of rows that meet the search criteria. Each row is for a different envelope.

folderItems

Value	Description
completedDateTime	Date/Time the envelope was completed
createdDateTime	Date/Time envelope was created
envelopeID	Identifier for the envelope
envelopeUri	Uri path for the envelope
expireDateTime	Date/Time envelope is set to expire
folderId	Identifier of the folder where the envelope lives.
folderUri	Path to the folder
ownerName	Name of the envelope owner
recipients	Recipient information and statuses listed by recipient types. This is the same as the information returned by the Get Envelope Recipient Status request.
recipientUri	Uri path of the envelope recipients.
senderName	Name of the envelope sender.
sentDateTime	Date/Time the envelope was sent.

status	Current status of the envelope
subject	Subject of the envelope.

The following example shows the response json body.

Example Response Body


```
{
    "endPosition": "2",
    "folderItems": [{
           "createdDateTime": "2012-07-13T15:57:00.6900000Z",
           "envelopeId": "463d30c4-3z29-417b-9789-010d221a76b7",
           "envelopeUri": "\/envelopes\/463d30c4-3e29-417b-9789-010d221a76b7",
           "expireDateTime": "2012-11-10T21:21:05.2649771Z",
           "folderID": "237f9912-b96b-4d96-8adb-05523d497225",
           "folderUri": "\/folders\/237f9912-b96b-4d96-8adb-05523d497225",
           "ownerName": "Bill Cat",
           "recipients": {
                  "signers": [{
                         "email": "rc.cat@yahoo.com",
                         "name": "Rod Cat",
                         "routingOrder": 1,
                         "status": "sent"
                  },
                  {
                         "email": "dccat@gmail.com",
                         "name": "Don Cat",
                         "routingOrder": 1,
                         "status": "sent"
                  }]
           },
           "senderName": "Bill Cat",
           "sentDateTime": "2012-07-13T15:57:27.8670000Z",
           "status": "sent",
           "subject": "Please DocuSign this document: Agreement.pdf"
    }],
    "nextUri": "accounts/{accountId}/search_folders/{search_name}?start_position=3 "
    "previousUri": "accounts/{accountId}/search_folders/{search_name}?start_position=1 "
    "resultSetSize": "1",
    "startPosition": "2",
    "totalRows": "3"
}
```


Get Account Settings

This returns the account settings information for the specified account

URL:

/accounts/{accountId}/settings

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the account setting name/value information for the specified account. See the accountSettings list for more information about the different account settings.

The following example shows the response json body.

Example Response Body

```
{
    "accountSettings":[{
           "name":"String content",
           "value":"String content"
    }]
}
```

Modify Account Settings

This updates the account settings list for the specified account.

URL:

/accounts/{accountId}/settings

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description

Name	Reqd?	Type	Description
accountSettings		String	The name/value pair information for account settings. These determine the features available for the account. See the accountSettings list for more information about the accountSettings.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "accountSettings":[{
           "name":"String content",
           "value":"String content"
    }]
}
```

Response

The response returns a success or failure.

Get Shared Access Information

This retrieves shared item status for one or more users and types of items.

Users with account administration privileges can retrieve shared access information for all account users. Users without account administrator privileges can only retrieve shared access information for themselves and the returned information is limited to retrieving the status of all members of the account that are sharing their folders to the user. This is equivalent to shared=shared_from.

When making a request, only one item_type, envelopes or templates, can be included in the query parameter.

URL:

/accounts/{accountId}/shared_access

Optional query parameter: userIds={userId,userId}, item_type={envelopes or templates}, envelopeId={envelopeId,envelopeId}, search_text={*string*}, shared={value}, count={integer}, start_position={integer}, templateId={templateId,templateId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query parameter can be added to the request.

Name	Reqd?	Type	Description
userIds	No	String	A comma separated list of userIds for whom the shared item information is being requested.
item_type	No	String	Specifies the type of shared item being requested. Only one item_type can be included in a request. The accepted values are: • envelopes: returns information about envelope sharing between users. • templates: returns information about template sharing between users and groups.
envelopeId	No	String	A comma separated list of envelopeIds which the shared item information is being requested.
templateId	No*	String	A comma separated list of templateIds for which the shared item information is being requested. * - If item_type=templates, then at least one templateId must be provided.
search_text	No	String	This can be used to filter user names in the response. The wild-card '*' (asterisk) can be used around the string.
shared	No	String	Specifies which users should be included in the response. Multiple values can be used in the query by using a comma separated list of shared values. If the requestor does not have account administrator privileges, the shared_to value is used. Requestors that do not have account administrator privileges can only use the shared_to, any other setting will result in an error. The accepted values are: • not_shared: Returns account users that the specified item type is not being shared with and that are not sharing the specified item type with the user. User ✕ (Share) ✕ Account user • shared_to: Returns account users that the

Name	Reqd?	Type	Description
			specified item type is not being shared with and who are sharing the specified item type with the user (only shared to the user). User ◀◀ (Share) Account user • shared_from: Returns account users that the specified item type is being shared with and who are not sharing the specified item type with the user (only shared from the user). User (Share) ▶▶ Account user • shared_to_and_from: Returns account users that the specified item type is being shared with and who are sharing the specified item type with the user. User ◀◀ (Share) ▶▶ Account user
count	No	String	Specifies maximum number of results included in the response. If no value is specified, this defaults to 1000.
start_position	No	String	If the response set exceeds Count, this can be used to specify that the method should return users starting at the specified index. The first index is 0, and should be used in the first GET call. Typically this number is a multiple of Count. If no value is specified, this defaults to be 0.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountid}/shared_access

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response provides information about the result set, an array of the sharedAccess items, and either a success or failure. If the call fails an error code is provided.

A description of the information in the response is given below.

Value	Description
resultSetSize	The number of results returned in this response.
startPosition	The index of the first element in the response set.

Value	Description
endPosition	The index of the last element in the response set.
totalSetSize	The total number of results from the call. This will always be greater than or equal to resultSetSize.
sharedAccess	The shared access information for the users specified in the request. See description below for more information.

sharedAccess Information for envelope requests:

Value	Description
User	A complex element with the information for the user associated with the shared access information. It is made up of the userId, userName, and email information for the user.
envelopes	The shared access information for envelopes user: A complex element with the information for the user associated with the shared status. It is made up of the userId, userName, and email information for the user. shared: Specifies the current sharing status between the requested user and other account user for envelopes. The possible values are: • Not_shared: Envelopes are not being shared with this account user and the account user not sharing the specified item type with the user. User ✕ (Share) ✕ Account user • shared_to: Returns account users that the specified item type is not being shared with and who are sharing the specified item type with the user. User ◀◀ (Share) Account user • shared_from: Returns account users that the specified item type is being shared with and who are not sharing the specified item type with the user. User (Share) ▶▶ Account user • shared_to_and_from: Envelopes are being shared with the account user and account user is sharing envelopes with the user. User ◀◀ (Share) ▶▶ Account user

sharedAccess information for template requests:

Value	Description

Value	Description
templates	The shared access information for templates templateId: The DocuSign generated unique identifier for the template. templateName: The filename for the template. owner: A complex element with the information for the user that currently owns the template. It is made up of the userId, userName, email information, and a uri for the user. sharedUsers: A complex element with the user information for users the template is shared with. It is made up of the userId, userName, and email information for the user. It also includes the shared status information for the template, which will always be shared_to. sharedGroups: A complex element with group information for groups the template is shared with. It is made up of the group information, explained below. It also includes the shared status information for the template, which will always be shared_to. • group: A complex element with the groupId (the DocuSign generated unique identifier for the group), groupName, and groupType.

Examples of the response body for json and XML are shown below.

Example JSON Response Body for envelope requests

```
{
  "resultSetSize": "2",
  "startPosition": "0",
  "endPosition": "1",
  "totalSetSize": "2",
  "sharedAccess": [
    {
      "user": {
        "userName": "John Doe",
        "userId": "7046be14-2898-43ea-ba4e-00c29a60676b",
        "email": "john.doe@docusign.com",
      },
      “envelopes”: [
        {
          "user": {
            "userName": "Estevan Doe",
            "userId": "98ca4b2c-c46f-4802-9bd6-06448cf2abbb ",
            "email": estevan.doe@docusign.com
          },
          "shared": "shared_to"
        },
        {
          "user": {
            "userName": "Jane Doe",
            "userId": "3770d021-0c10-4652-809b-01b669308bb8",
            "email": "jane.doe@docusign.com"
          },
          "shared": "shared_from"
        },
      ]
```

```
    }
  ]
}
```

Example JSON Response Body for template requests

```
{
  "resultSetSize": "1",
  "startPosition": "0",
  "endPosition": "1",
  "totalSetSize": "1",
  "sharedAccess": [
    {
      "templates": [
        {
          "templateId": "270D920E-C65A-410D-9640-75D2FBEADAC2",
          "templateName": "NDA-US",
          "owner": {
            "userName": "John Doe",
            "userId": "7046be14-2898-43ea-ba4e-00c29a60676b",
            "email": "john.doe@docusign.com",
          },
          "sharedUsers": [
            {
              "user": {
                "userName": "Jane Doe",
                "userId": "3770d021-0c10-4652-809b-01b669308bb8",
                "email": "jane.doe@docusign.com"
              },
              "shared": "shared_to"
            }
          ],
          "sharedGroups": [
            {
              "group": {
                "groupId": "3990f021-9a10-4652-809b-01b669308bb8",
                "groupName": "Admin",
                "groupType": "administrators",
              "shared": "shared_to"
            }
          ]
        }
      ]
    }
  ]
}
```

Example XML Response Body for envelope requests

Note that the structure of the XML response is slightly different than the json response, in that the user and envelopes information are included in the memberSharedItems element.

```
<accountSharedAccess xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <accountId>string</accountId>
  <endPosition>string</endPosition>
  <nextUri>string</nextUri>
  <previousUri>string</previousUri>
  <resultSetSize>string</resultSetSize>
  <sharedAccess>
    <memberSharedItems>
      <user>
```

```
        <email>string</email>
        <uri>string</uri>
        <userId>string</userId>
        <userName>string</userName>
      </user>
      <envelopes>
        <sharedItem>
          <shared>string</shared>
          <user>
            <email>string</email>
            <uri>string</uri>
            <userId>string</userId>
            <userName>string</userName>
          </user>
        </sharedItem>
        <sharedItem>
          <shared>string</shared>
          <user>
            <email>string</email>
            <uri>string</uri>
            <userId>string</userId>
            <userName>string</userName>
          </user>
        </sharedItem>
      </envelopes>
    </memberSharedItems>
    <memberSharedItems>
      <user>
        <email>string</email>
        <uri>string</uri>
        <userId>string</userId>
        <userName>string</userName>
      </user>
      <envelopes>
        <sharedItem>
          <shared>string</shared>
          <user>
            <email>string</email>
            <uri>string</uri>
            <userId>string</userId>
            <userName>string</userName>
          </user>
        </sharedItem>
        <sharedItem>
          <shared>string</shared>
          <user>
            <email>string</email>
            <uri>string</uri>
            <userId>string</userId>
            <userName>string</userName>
          </user>
        </sharedItem>
      </envelopes>
    </memberSharedItems>
  </sharedAccess>
  <startPosition>string</startPosition>
  <totalSetSize>string</totalSetSize>
</accountSharedAccess>
```

Set Shared Access Information

This sets the shared access status for one or more users or templates.

When setting user shared access, only users with account administration privileges can set shared access status for envelopes.

When setting template shared access, only users that own a template and have sharing permission or with account administration privileges can set shared access for templates.

Changes to the shared items status are not additive; the change always replaces the current status.

To change template shared access the query parameter item_type = templates must be added to the request. When this is set, the user and envelopes properties are not required.

URL:

/accounts/{accountId}/shared_access

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
user	Yes	String	The userId of the user for which the shared item status is being changed.
envelopes	Yes	String	The shared access information for envelopes. user: The userId of the account user for whom the shared access status is being changed. The status change is relative to the user set by the userId described above. shared: Specifies the change to the sharing status between the user and account user for envelopes. The accepted values are: • not_shared: This removes all sharing between the user and the account user for envelopes. • shared_to: Returns account users that the specified item type is not being shared with and who are sharing the specified item type with the user (only shared to the user). • shared_from: Returns account users that the specified item type is being shared with and who are not sharing the specified item type with the user (only shared from the user).

Name	Reqd?	Type	Description
			• shared_to_and_from: This shares access to and from envelopes for both the user and the account user.
templates	Yes	String	The shared access information for templates. templateId: The DocuSign generated unique identifier for the template. sharedUsers: User information for template sharing changes. • user: An array of userIds of the account users for which template shared access is being changed. sharedGroups: Group information for template sharing changes. Groups can be identified using groupId or groupName. groupId: An array of group Ids for the groups which template shared access is being changed. groupName: An array of group names for the groups which template shared access is being changed. shared: Specifies the change to the sharing status for the template. The accepted values are: • not_shared: This removes shared access to the template for the specified account users or groups. • shared_to: Sets shared access for the template to the specified account users and groups.

Example JSON Request Body for envelope requests

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountid}/shared_access

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "sharedAccess": [
    {
      "user": {
        "userId": "123",
      },
```

```
      “envelopes”: [
        {
          "user": {
            "userId": "1234",
          },
          "shared": "shared_to"
        },
        {
          "user": {
            "userId": "123456",
          },
          "shared": "shared_from"
        }
      ]
    },
    {
      "user": {
        "userId": "3770d021-0c10-4652-809b-01b669308bb8",
      },
      “envelopes”: [
        {
          "user": {
            "userId": "123",
          },
          "shared": "shared_to"
        }
      ]
    },
    {
      "user": {
        "userId": "98ca4b2c-c46f-4802-9bd6-06448cf2abbb",
      },
      “envelopes”: [
        {
          "user": {
            "userId": "3770d021-0c10-4652-809b-01b669308bb8",
          },
          "shared": "shared_to"
        },
        {
          "user": {
            "userId": "29c56d02-ec97-4a2a-b99d-6c5a2e400b8b",
          },
          "shared": "shared_to_and_from"
        }
      ]
    }
  ]
}
```

Example JSON Request Body for template requests

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountid}/shared_access
?item_type=templates

X-DocuSign-Authentication:
&lt;DocuSignCredentials&gt;&lt;Username&gt;{name}&lt;/Username&gt;&lt;Password&gt;{passwo
rd}&lt;/Password&gt;&lt;IntegratorKey&gt;{integrator_key}&lt;/IntegratorKey&gt;&lt;/DocuS
ignCredentials&gt;
Accept: application/json
Content-Type: application/json
{
```

```
  "sharedAccess": [
    {
      "templates": [
        {
          "templateId": "270D920E-C65A-410D-9640-75D2FBEADAC2",
          "sharedUsers": [
            {
              "user": {
                "userId": "3770d021-0c10-4652-809b-01b669308bb8",
                "userId": "936ad868-c597-4140-882b-c0452287e286"
              }
            ],
          "sharedGroups": [
            {
              "group": {
                "groupId": "9990d021-0c15-4561-809x-01b669308bb8",
                "groupId": "9340d065-4g17-6781-935w-87b546308hr8"
              ]
            }
          }
        ],
        "shared": "shared_to"
      }
    }
  ]
}
```

Example XML Request Body for envelope requests

Note that the structure of the XML request is slightly different than the json request, in that the user and envelopes information are included in the memberSharedItems element.

```
<accountSharedAccess xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <sharedAccess>
    <memberSharedItems>
      <envelopes>
        <sharedItem>
          <shared>string</shared>
          <user>
            <userId>string</userId>
          </user>
        </sharedItem>
        <sharedItem>
          <shared>string</shared>
          <user>
            <userId>string</userId>
          </user>
        </sharedItem>
      </envelopes>
      <user>
        <userId>string</userId>
      </user>
    </memberSharedItems>
    <memberSharedItems>
      <envelopes>
        <sharedItem>
          <shared>string</shared>
          <user>
            <userId>string</userId>
          </user>
        </sharedItem>
```

```
      </envelopes>
    </memberSharedItems>
  </sharedAccess>
  <startPosition>string</startPosition>
  <totalSetSize>string</totalSetSize>
</accountSharedAccess>
```

Response

The response returns a success or failure, along with the changed shared access status.

Examples of the response body for json and XML are shown below.

Example JSON Response Body for envelope requests

```
{
  "sharedAccess": [
    {
      "user": {
        "userId": "123",
      },
      "errorDetails": {
        "errorCode": "INVALID_USERID",
        "message": "Invalid Userid."
      }
    },
    {
      "user": {
        "userId": "3770d021-0c10-4652-809b-01b669308bb8",
      },
      "envelopes": [
        {
          "user": {
            "userId": "123",
          },
          "errorDetails": {
            "errorCode": "INVALID_USERID",
            "message": "Invalid Userid."
          }
        }
      ]
    },
    {
      "user": {
        "userName": "Estevan Doe",
        "userId": "98ca4b2c-c46f-4802-9bd6-06448cf2abbb ",
        "email": estevan.doe@docusign.com
      },
      "envelopes": [
        {
          "user": {
            "userName": "Jane Doe",
            "userId": "3770d021-0c10-4652-809b-01b669308bb8",
            "email": "jane.doe@docusign.com"
          },
          "shared": "shared_to"
        },
        {
          "user": {
            "userId": "29c56d02-ec97-4a2a-b99d-6c5a2e400b8b",
          },
          "errorDetails": {
```

```
            "errorCode": "USER_LACKS_MEMBERSHIP",
            "message": "The UserID does not have a valid membership in this Account."
          }
        }
      ]
    }
  ]
```

Example JSON Response Body for template requests

```
{
  "resultSetSize": "1",
  "startPosition": "0",
  "endPosition": "1",
  "totalSetSize": "1",
  "sharedAccess": [
    {
      "templates": [
        {
          "templateId": "270D920E-C65A-410D-9640-75D2FBEADAC2",
          "templateName": "NDA-US",
          "owner": {
            "userName": "John Doe",
            "userId": "7046be14-2898-43ea-ba4e-00c29a60676b",
            "email": "john.doe@docusign.com",
          },
          "sharedUsers": [
            {
              "user": {
                "userName": "Jane Doe",
                "userId": "3770d021-0c10-4652-809b-01b669308bb8",
                "email": "jane.doe@docusign.com"
              },
              "shared": "shared_to"
            }
          ],
          "sharedGroups": [
            {
              "group": {
                "groupId": "3990f021-9a10-4652-809b-01b669308bb8",
                "groupName": "Admin",
                "groupType": "administrators",
              "shared": "shared_to"
            }
          ]
        }
      ]
    }
  ]
}
```

Example XML Response Body for envelope requests

Note that the structure of the XML response is slightly different than the json response, in that the user and envelopes information are included in the memberSharedItems element.

```
<accountSharedAccess xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <accountId>string</accountId>
  <endPosition>string</endPosition>
  <errorDetails>
    <errorCode>string</errorCode>
```

```
    <message>string</message>
  </errorDetails>
  <nextUri>string</nextUri>
  <previousUri>string</previousUri>
  <resultSetSize>string</resultSetSize>
  <sharedAccess>
    <memberSharedItems>
      <errorDetails>
        <errorCode>string</errorCode>
        <message>string</message>
      </errorDetails>
      <user>
        <email>string</email>
        <uri>string</uri>
        <userId>string</userId>
        <userName>string</userName>
      </user>
      <envelopes>
        <sharedItem>
          <errorDetails>
            <errorCode>string</errorCode>
            <message>string</message>
          </errorDetails>
          <shared>string</shared>
          <user>
            <email>string</email>
            <uri>string</uri>
            <userId>string</userId>
            <userName>string</userName>
          </user>
        </sharedItem>
        <sharedItem>
          <shared>string</shared>
          <user>
            <email>string</email>
            <uri>string</uri>
            <userId>string</userId>
            <userName>string</userName>
          </user>
        </sharedItem>
      </envelopes>
    </memberSharedItems>
  </sharedAccess>
  <startPosition>string</startPosition>
  <totalSetSize>string</totalSetSize>
</accountSharedAccess>
```

List Signature Providers

This returns a list of signature providers and associated information available to the specified account. Signature providers are used with DocuSign's Standards-Based Signatures.

Important: While this method is available in the production version of the Signature REST API v2, it is still considered to be in beta and subject to change.

URL:

/accounts/{accountId}/signatureproviders

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters.

Response

The response returns the list of signature providers

Name	Type	Description
signatureProviderId	String	ID associated the signature provider.
signatureProviderName	String	The name of the signature provider used in the system.
signatureProviderDisplayName	String	The signature provider name typically used for display.
Priority	String	An integer for the usage priority order of the signature provider for the account. This can be used to show preferred signature providers. A value of "0" means this is a default provider.
isRequired	Boolean	Indicates if the signature provider is required for the account.
signatureProviderOptionsMetadata		A list of options the signature provider can use to associate a signer with a record in their system. It consists of the following information: • signatureProviderOptionId: The ID of the signature provider option. • signatureProviderOptionName: The system name of the signature provider option. • signatureProviderOptionDisplayName: The signature provider option display name.

Name	Type	Description
signatureProviderRequiredOptions		A list of required options for this signature provider. It consists of the following information: • signerType: The signer type of required signature provider option. • requiredSignatureProviderOptionIds: The required options for the signature provider. The list uses signatureProviderOptionId. If there are multiple options separated by commas, then only one of the options is required (for example, [1,2] means option 1 or 2 is required).

Example JSON Response Layout

```
{
  "signatureProviders": [
    {
      "signatureProviderId": "1",
      "signatureProviderName": "universalsignaturepen_default",
      "signatureProviderDisplayName": "usig default pen",
      "priority": "0",
      "isRequired": "false",
      "signatureProviderOptionsMetadata": [
        {
          "signatureProviderOptionId": "0",
          "signatureProviderOptionName": "onetimepassword",
          "signatureProviderOptionDisplayName": "Password"
        },
        {
          "signatureProviderOptionId": "1",
          "signatureProviderOptionName": "sms",
          "signatureProviderOptionDisplayName": "SMS"
        }
      ]
    },
    {
      "signatureProviderId": "2",
      "signatureProviderName": "universalsignaturepen_opentrustdirect",
      "signatureProviderDisplayName": "EU Advanced",
      "priority": "1",
      "isRequired": "false",
      "signatureProviderOptionsMetadata": [
        {
          "signatureProviderOptionId": "0",
          "signatureProviderOptionName": "onetimepassword",
          "signatureProviderOptionDisplayName": "Password"
        },
        {
          "signatureProviderOptionId": "1",
          "signatureProviderOptionName": "sms",
          "signatureProviderOptionDisplayName": "SMS"
        }
      ],
      "signatureProviderRequiredOptions": [
        {
          "signerType": "signer",
          "requiredSignatureProviderOptionIds": [
```

```
            "0",
            "1"
          ]
        }
      ]
    }
  ]
}
```

Example XML Response Layout

```
&lt;accountSignatureProviders xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi"&gt;
  &lt;signatureProviders&gt;
    &lt;accountSignatureProvider&gt;
      &lt;isRequired&gt;false&lt;/isRequired&gt;
      &lt;priority&gt;0&lt;/priority&gt;
      &lt;signatureProviderDisplayName&gt;usig default
pen&lt;/signatureProviderDisplayName&gt;
      &lt;signatureProviderId&gt;1&lt;/signatureProviderId&gt;
      &lt;signatureProviderName&gt;universalsignaturepen_default&lt;/signatureProviderNam
e&gt;
      &lt;signatureProviderOptionsMetadata&gt;
        &lt;accountSignatureProviderOption&gt;
          &lt;signatureProviderOptionDisplayName&gt;Password&lt;/signatureProviderOptionD
isplayName&gt;
          &lt;signatureProviderOptionId&gt;0&lt;/signatureProviderOptionId&gt;
          &lt;signatureProviderOptionName&gt;onetimepassword&lt;/signatureProviderOptionN
ame&gt;
        &lt;/accountSignatureProviderOption&gt;
        &lt;accountSignatureProviderOption&gt;
          &lt;signatureProviderOptionDisplayName&gt;SMS&lt;/signatureProviderOptionDispla
yName&gt;
          &lt;signatureProviderOptionId&gt;1&lt;/signatureProviderOptionId&gt;
          &lt;signatureProviderOptionName&gt;sms&lt;/signatureProviderOptionName&gt;
        &lt;/accountSignatureProviderOption&gt;
      &lt;/signatureProviderOptionsMetadata&gt;
    &lt;/accountSignatureProvider&gt;
    &lt;accountSignatureProvider&gt;
      &lt;isRequired&gt;false&lt;/isRequired&gt;
      &lt;priority&gt;1&lt;/priority&gt;
      &lt;signatureProviderDisplayName&gt;EU
Advanced&lt;/signatureProviderDisplayName&gt;
      &lt;signatureProviderId&gt;2&lt;/signatureProviderId&gt;
      &lt;signatureProviderName&gt;universalsignaturepen_opentrustdirect&lt;/signaturePro
viderName&gt;
      &lt;signatureProviderOptionsMetadata&gt;
        &lt;accountSignatureProviderOption&gt;
          &lt;signatureProviderOptionDisplayName&gt;Password&lt;/signatureProviderOptionD
isplayName&gt;
          &lt;signatureProviderOptionId&gt;0&lt;/signatureProviderOptionId&gt;
          &lt;signatureProviderOptionName&gt;onetimepassword&lt;/signatureProviderOptionN
ame&gt;
        &lt;/accountSignatureProviderOption&gt;
        &lt;accountSignatureProviderOption&gt;
          &lt;signatureProviderOptionDisplayName&gt;SMS&lt;/signatureProviderOptionDispla
yName&gt;
          &lt;signatureProviderOptionId&gt;1&lt;/signatureProviderOptionId&gt;
          &lt;signatureProviderOptionName&gt;sms&lt;/signatureProviderOptionName&gt;
        &lt;/accountSignatureProviderOption&gt;
      &lt;/signatureProviderOptionsMetadata&gt;
      &lt;signatureProviderRequiredOptions&gt;
          &lt;signatureProviderRequiredOption&gt;
            &lt;requiredSignatureProviderOptionIds&gt;
```

```
              &lt;string&gt;0, 1&lt;/string&gt;
            &lt;/requiredSignatureProviderOptionIds&gt;
            &lt;signerType&gt;signer&lt;/signerType&gt;
          &lt;/signatureProviderRequiredOption&gt;
        &lt;/signatureProviderRequiredOptions&gt;
      &lt;/accountSignatureProvider&gt;
    &lt;/signatureProviders&gt;
&lt;/accountSignatureProviders&gt;
```

Create a Signing Group

This allows you to create one or more signing groups. Multiple signing groups can be created in one call. Only users with account administrator privileges can create signing groups.

An account can have a maximum of 50 signing groups. Each signing group can have a maximum of 50 group members.

Signing groups can be used by any account user.

URL:

/accounts/{accountId}/signing_groups

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
groupName	Yes	String	The display name for the signing group. This can be a maximum of 100 characters.
groupType	Yes	String	Sets the signing group type. The only possible value at this time is: sharedSigningGroup,
users	No	Users	An array of group members for the signing group. It is composed of two elements: • userName – The name for the group member. This can be a maximum of 100 characters. • email – The email address for the group member. This can be a maximum of 100 characters.

Example Request Body

The following example shows the request json body.

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "groups": [
           {
                  "groupName": "string",
                  "groupType": "sharedSigningGroup",
                  "users": [
                         {
                                "userName": "string",
                                "email": "string"
                         },
                         {
                                "userName": "string",
                                "email": "string"
                         }
                  ]
           },
           {
                  "groupName": "string",
                  "groupType": "sharedSigningGroup",
                  "users": [
                         {
                                "userName": "string",
                                "email": "string"
                         },
                         {
                                "userName": "string",
                                "email": "string"
                         }
                  ]
           }
    ]
}
```

The following example shows the request XML body.

```
<signingGroupInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
    <groups>
           <signingGroup>
                  <groupName>string</groupName>
                  <groupType>sharedSigningGroup</groupType>
                  <users>
                         <signingGroupUser>
                                <email>string</email>
                                <userName>string</userName>
                         </signingGroupUser>
                         <signingGroupUser>
                                <email>string</email>
                                <userName>string</userName>
                         </signingGroupUser>
                  </users>
           </signingGroup>
           <signingGroup>
                  <groupName>string</groupName>
```

```
                <groupType>sharedSigningGroup</groupType>
                <users>
                        <signingGroupUser>
                                <email>string</email>
                                <userName>string</userName>
                        </signingGroupUser>
                        <signingGroupUser>
                                <email>string</email>
                                <userName>string</userName>
                        </signingGroupUser>
                </users>
        </signingGroup>
    </groups>
</signingGroupInformation>
```

Response

The response returns a success or failure with any error messages. For successes DocuSign generates a signingGroupId for each group, which is included in the response. The response also includes information about when the group was created and modified, including the account user that created and modified the group.

Example Response Body

The following example shows the response json body.

```
{
    "groups": [
            {
                    "signingGroupId": "string",
                    "groupName": "string",
                    "groupType": "string",
                    "created": "string",
                    "createdBy": "string",
                    "modified": "string",
                    "modifiedBy": "string",
                    "users": [
                            {
                                    "userName": "string",
                                    "email": "string",
                            },
                            {
                                    "userName": "string",
                                    "email": "string",
                            }
                    ],
            }
    ]
}
```

The following example shows the response XML body.

```
  <groups>
          <signingGroup>
                  <created>string</created>
                  <createdBy>string</createdBy>
                  <groupName>string</groupName>
                  <groupType>string</groupType>
                  <modified>string</modified>
                  <modifiedBy>string</modifiedBy>
                  <signingGroupId>string</signingGroupId>
```

```
                <users>
                        <signingGroupUser>
                                <email>string</email>
                                <userName>string</userName>
                        </signingGroupUser>
                </users>
        </signingGroup>
  </groups>
```

Get a List of Signing Groups

This provides a list of all signing groups for the account.

URL:

/accounts/{accountId}/signing_groups

Optional query string: include_users={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query string can be added to the request.

Name	Reqd?	Type	Description
include_users	No	Boolean	When true, the response will include the signing group members.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of shared signing groups for the account. Signing group members are only included in the response if the include_users query parameter was added to the request.

Example Response Body

The following example shows the response json body.

```
{
    "groups": [
            {
                    "signingGroupId": "string",
                    "groupName": "string",
                    "groupType": "string",
                    "created": "string",
                    "createdBy": "string",
                    "modified": "string",
                    "modifiedBy": "string",
            },
            {
                    "signingGroupId": "string",
                    "groupName": "string",
                    "groupType": "string",
                    "created": "string",
                    "createdBy": "string",
                    "modified": "string",
                    "modifiedBy": "string",
            }
    ]
}
```

The following example shows the response XML body.

```
  <groups>
          <signingGroup>
                  <created>string</created>
                  <createdBy>string</createdBy>
                  <groupName>string</groupName>
                  <groupType>string</groupType>
                  <modified>string</modified>
                  <modifiedBy>string</modifiedBy>
                  <signingGroupId>string</signingGroupId>
          </signingGroup>
          <signingGroup>
                  <created>string</created>
                  <createdBy>string</createdBy>
                  <groupName>string</groupName>
                  <groupType>string</groupType>
                  <modified>string</modified>
                  <modifiedBy>string</modifiedBy>
                  <signingGroupId>string</signingGroupId>
          </signingGroup>
  </groups>
```

Update Signing Group Names

This is used to update the name of one or more existing signing groups.

URL:

/accounts/{accountId}/signing_groups

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
signingGroupId	Yes	String	The DocuSign generated ID number for the signing group. This is used to identify the group being changed and cannot be modified.
groupName	Yes	String	The new display name for the signing group. This can be a maximum of 100 characters.

Example Request Body

The following example shows the request json body.

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "groups":[
            {
                    "signingGroupId":"string",
                    "groupName":"string",
            },
            {
                    "signingGroupId":"string",
                    "groupName":"string",
            }
    ]
}
```

The following example shows the request XML body.

```
<signingGroupInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
    <groups>
            <signingGroup>
                    <groupName>string</groupName>
                    <signingGroupId>string</ signingGroupId >
            </signingGroup>
            <signingGroup>
                    <groupName>string</groupName>
                    <signingGroupId>string</signingGroupId >
            </signingGroup>
    </groups>
</signingGroupInformation>
```

Response

The response returns a list of modified signing groups.

Example Response Body

The following example shows the response json body.

```
{
    "groups": [
            {
                    "signingGroupId": "string",
                    "groupName": "string",
                    "groupType": "sample string 3",
                    "created": "string",
                    "createdBy": "string",
                    "modified": "string",
                    "modifiedBy": "string",
            },
            {
                    "signingGroupId": "string",
                    "groupName": "string",
                    "groupType": "sample string 3",
                    "created": "string",
                    "createdBy": "string",
                    "modified": "string",
                    "modifiedBy": "string",
            }
    ]
}
```

The following example shows the response XML body.

```
  <groups>
          <signingGroup>
                  <created>string</created>
                  <createdBy>string</createdBy>
                  <groupName>string</groupName>
                  <groupType>string</groupType>
                  <modified>string</modified>
                  <modifiedBy>string</modifiedBy>
                  <signingGroupId>string</signingGroupId>
          </signingGroup>
          <signingGroup>
                  <created>string</created>
                  <createdBy>string</createdBy>
                  <groupName>string</groupName>
                  <groupType>string</groupType>
                  <modified>string</modified>
                  <modifiedBy>string</modifiedBy>
                  <signingGroupId>string</signingGroupId>
          </signingGroup>
  </groups>
```

Delete Signing Groups

This deletes one or more signing groups.

URL:

/accounts/{accountId}/signing_groups

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
signingGroupId	Yes	String	The DocuSign generated ID number for the signing group.

Example Request Body

The following example shows the request json body.

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "groups":[
            {
                    "signingGroupId":"string"
            },
            {
                    "signingGroupId":"string"
            }
    ]
}
```

The following example shows the request XML body.

```
<signingGroupInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
    <groups>
            <signingGroup>
                    <signingGroupId>string</signingGroupId >
            </signingGroup>
            <signingGroup>
                    <signingGroupId>string</signingGroupId >
            </signingGroup>
    </groups>
</signingGroupInformation>
```

Response

The response returns a success or failure.

Get a Signing Group

This returns information, including group member information, for one signing group.

URL:

/accounts/{accountId}/signing_groups/{signingGroupId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups
/{signingGroupId

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns information about the requested group, including group member information.

Example Response Body

The following example shows the response json body.

```
{
    "groups": [
           {
                  "signingGroupId": "string",
                  "groupName": "string",
                  "groupType": "sample string 3",
                  "created": "string",
                  "createdBy": "string",
                  "modified": "string",
                  "modifiedBy": "string",
                  "users": [
                         {
                                "userName": "string",
                                "email": "string",
                         },
                         {
                                "userName": "string",
                                "email": "string",
```

```
                        }
                ],
            }
        ]
    }
```

The following example shows the response XML body.

```
  <groups>
          <signingGroup>
                <created>string</created>
                <createdBy>string</createdBy>
                <groupName>string</groupName>
                <groupType>string</groupType>
                <modified>string</modified>
                <modifiedBy>string</modifiedBy>
                <signingGroupId>string</signingGroupId>
                <users>
                      <signingGroupUser>
                            <email>string</email>
                            <userName>string</userName>
                      </signingGroupUser>
                </users>
          </signingGroup>
  </groups>
```

Update a Signing Group

This is used to update signing group name and member information. This can also be used to add new members to the signing group. A signing group can have a maximum of 50 members.

URL:

/accounts/{accountId}/signing_groups/{signingGroupId}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
groupName	No	String	The display name for the signing group. This can be a maximum of 100 characters.
groupType	No	String	Sets the signing group type, shared for all account members or private for one account member. The only possible value at this time is: sharedSigningGroup.

Name	Reqd?	Type	Description
users	No	Users	An array of group members for the signing group. It is composed of two elements: • userName – The name for the group member. This can be a maximum of 100 characters. • email – The email address for the group member. This can be a maximum of 100 characters.

Example Request Body

The following example shows the request json body.

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups
/{signingGroupId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "groupName": "string",
    "groupType": "string",
    "users": [
           {
                  "userName": "string",
                  "email": "string"
           }
    ]
}
```

The following example shows the request XML body.

```
<signingGroup xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
    <groupName>string</groupName>
    <groupType>string</groupType>
    <users>
           <signingGroupUser>
                  <email>string</email>
                  <userName>string</userName>
           </signingGroupUser>
    </users>
</signingGroup>
```

Response

The response returns a success or failure along with the updated group information.

Example Response Body

The following example shows the response json body.

```
{
    "signingGroupId": "string",
    "groupName": "string",
    "groupType": "sample string 3",
    "created": "string",
    "createdBy": "string",
    "modified": "string",
    "modifiedBy": "string",
    "users": [
           {
                  "userName": "string",
                  "email": "string",
           },
           {
                  "userName": "string",
                  "email": "string",
           }
    ]
}
```

The following example shows the response XML body.

```
<signingGroup>
    <created>string</created>
    <createdBy>string</createdBy>
    <groupName>string</groupName>
    <groupType>string</groupType>
    <modified>string</modified>
    <modifiedBy>string</modifiedBy>
    <signingGroupId>string</signingGroupId>
    <users>
           <signingGroupUser>
                  <email>string</email>
                  <userName>string</userName>
           </signingGroupUser>
    </users>
</signingGroup>
```

Get Signing Group Members

This returns the group member information for one signing group.

URL:

/accounts/{accountId}/signing_groups/{signingGroupId}/users

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups
/{signingGroupId}/users

X-DocuSign-Authentication:
&lt;DocuSignCredentials&gt;&lt;Username&gt;{name}&lt;/Username&gt;&lt;Password&gt;{passwo
rd}&lt;/Password&gt;&lt;IntegratorKey&gt;{integrator_key}&lt;/IntegratorKey&gt;&lt;/DocuS
ignCredentials&gt;
Accept: application/json
Content-Type: application/json
```

Response

The response returns the group member information for the signing group.

Example Response Body

The following example shows the response json body.

```
{
    "users":[
           {
                  "userName":"string",
                  "email":"string"
           },
           {
                  "userName":"string",
                  "email":"string"
           }
    ]
}
```

The following example shows the response XML body.

```
    <users>
           <signingGroupUser>
                  <email>string</email>
                  <userName>string</userName>
           </signingGroupUser>
           <signingGroupUser>
                  <email>string</email>
                  <userName>string</userName>
           </signingGroupUser>
    </users>
```

Add Signing Group Members

This is used to add one or more new members to a signing group. A signing group can have a maximum of 50 members.

URL:

/accounts/{accountId}/signing_groups/{signingGroupId}/users

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
userName	Yes	String	The name for the new group member. This can be a maximum of 100 characters.
email	Yes	Sting	The email address for the new group member. This can be a maximum of 100 characters.

Example Request Body

The following example shows the request json body.

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups
/{signingGroupId}/users

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "users":[
           {
                  "userName":"string",
                  "email":"string"
           },
           {
                  "userName":"string",
                  "email":"string"
           }
    ]
}
```

The following example shows the request XML body.

```
<signingGroupUsers xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
    <users>
           <signingGroupUser>
                  <email>string</email>
                  <userName>string</userName>
           </signingGroupUser>
           <signingGroupUser>
                  <email>string</email>
                  <userName>string</userName>
           </signingGroupUser>
    </users>
```

```
</signingGroupUsers>
```

Response

The response returns a success or failure along with the information about the added group members.

Example Response Body

The following example shows the response json body.

```
{
    "users":[
            {
                    "userName":"string",
                    "email":"string"
            },
            {
                    "userName":"string",
                    "email":"string"
            }
    ]
}
```

The following example shows the response XML body.

```
    <users>
            <signingGroupUser>
                    <email>string</email>
                    <userName>string</userName>
            </signingGroupUser>
            <signingGroupUser>
                    <email>string</email>
                    <userName>string</userName>
            </signingGroupUser>
    </users>
```

Delete Signing Group Members

This is used to remove one or more members from a signing group.

URL:

/accounts/{accountId}/signing_groups/{signingGroupId}/users

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description

Name	Reqd?	Type	Description
userName	Yes	String	The name of the group member to be deleted.
email	Yes	Sting	The email address of the group member to be deleted.

Example Request Body

The following example shows the request json body.

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/signing_groups
/{signingGroupId}/users

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "users":[
            {
                    "userName":"string",
                    "email":"string"
            },
            {
                    "userName":"string",
                    "email":"string"
            }
    ]
}
```

The following example shows the request XML body.

```
<signingGroupUsers xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
    <users>
            <signingGroupUser>
                    <email>string</email>
                    <userName>string</userName>
            </signingGroupUser>
            <signingGroupUser>
                    <email>string</email>
                    <userName>string</userName>
            </signingGroupUser>
    </users>
</signingGroupUsers>
```

Response

The response returns a success or failure.

Create a Custom Tab

This allows you to create a tab with pre-defined properties, such as a text tab with a certain font type and validation pattern. Users can access the custom tabs when sending documents through the DocuSign web application.

Custom tabs can be created for approve, checkbox, company, date, date signed, decline, email, email address, envelope ID, first name, formula, full name, initial here, last name, list, note, number, radio, sign here, signer attachment, SSN, text, title, and zip tabs.

URL:

/accounts/{accountId}/tab_definitions

Formats:

XML, JSON

HTTP method:

POST

Parameters:

Name	Reqd?	Type	Description
anchor	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
bold	No	Boolean	If true, the information in the tab is bold.
concealValueOnDocument	No	Boolean	If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text tabs and does not affect list tabs, radio buttons, or check boxes.

Name	Reqd?	Type	Description
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text tabs.
editable	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
font	No	String	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, MSGothic, and MSMincho.
fontColor	No	String	The font color used for the information in the tab. Possible values are: black, brightBlue, brightRed, darkGreen, darkRed, gold, green, navyBlue, purple, or white.
fontSize	No	String	The font size used for the information in the tab. Possible values are: size7, size8, size9, size10, size11, size12, size14, size16, size18, size20, size22, size24, size26, size28, size36, size48, or size72.
height	No	String	Height of the tab, in pixels.
includeInEmail	No	Boolean	Only used for note tabs. If true, the text in the note tab is included in the email sent to the assigned recipient.
initialValue	No	String	The initial value of the tab.
italic	No	Boolean	If true, the information in the tab is italic.
items	No	Items	Only used with list tabs. This is a semi-colon separated list used to populate the possible selections for a dropdown list. These are mapped to the list tab – listItems – text information.
locked	No	Boolean	If true, the signer cannot change the data in the tab.
maximumLength	No	String	The maximum length of the tab, in characters.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. Note that radio tabs are not supported with the merge field option. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
name	Yes	String	The name associated with the custom tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
shared	No	Boolean	If true, other account users can use this custom tab.
tabLabel	Yes	String	The label used with the tab. This can be a maximum of 500 characters.
type	Yes	String	Sets the type of custom tab. Possible values are: approve, checkBox, company, date, dateSigned, decline, email, emailAddress, envelopeId, firstName, formula, fullName, initialHere, lastName, list, note, number, radio, signerAttachment, signHere, ssn, text, title, zip5, and zip5Dash4.
underline	No	Boolean	If true, the information in the tab is underlined.
validationMessage	No	String	Message to be displayed to the signer if the validation rule from validationPattern fails. This is optional and if not provided the default DocuSign message will be displayed. This can be a maximum of 250 characters.

Name	Reqd?	Type	Description
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.
width	No	String	Width of the tab.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/tab_definitions

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "anchor": "string",
    "anchorXOffset": "string",
    "anchorYOffset": "string",
    "anchorIgnoreIfNotPresent": "string",
    "anchorUnits": "string",
    "bold": "string",
    "concealValueOnDocument": "string",
    "disableAutoSize": "string",
    "editable": "string",
    "font": "string",
    "fontColor": "string",
    "fontSize": "string",
    "height": "string",
    "includeInEmail": "string",
    "initialValue": "string",
    "italic": "string",
    "items": "string;string;string",
    "locked": "string",
    "maximumLength": "string",
    "mergeField": {
          "configurationType": "string",
          "path": "string",
          "writeBack": "string",
          "allowSenderToEdit": "string"
    },
    "name": "string",
    "required": "string",
    "shared": "string",
    "tabLabel": "string",
    "type": "string",
    "underline": "string",
    "validationMessage": "string",
    "validationPattern": "string",
    "width": "string"
}
```

Response

The response returns a success of failure along with the tab information, including the custom tab ID and information on when the tab was modified and the user that modified the tab. The custom tab ID is used when editing the tab or using the tab.

The following example shows the response json body.

Example Response Body

```
{
    "anchor": "string",
    "anchorXOffset": "string",
    "anchorYOffset": "string",
    "anchorIgnoreIfNotPresent": "string",
    "anchorUnits": "string",
    "bold": "string",
    "concealValueOnDocument": "string",
    "disableAutoSize": "string",
    "editable": "string",
    "font": "string",
    "fontColor": "string",
    "fontSize": "string",
    "height": "string",
    "includeInEmail": "string",
    "initialValue": "string",
    "italic": "string",
    "items": "string;string;string",
    "locked": "string",
    "maximumLength": "string",
    "mergeField": {
           "configurationType": "string",
           "path": "string",
           "writeBack": "string",
           "allowSenderToEdit": "string"
    },
    "name": "string",
    "required": "string",
    "shared": "string",
    "tabLabel": "string",
    "type": "string",
    "underline": "string",
    "validationMessage": "string",
    "validationPattern": "string",
    "width": "string",
    "customTabId": "string",
    "lastModifiedByUserId": "string",
    "lastModifiedByDisplayName": "string",
    "lastModified": "string",
    "createdByUserId": "string",
    "createdByDisplayName": "string"
}
```

Get List of all Account Tabs

This retrieves a list of all tabs associated with the account.

URL:

/accounts/{accountId}/tab_definitions

Optional query string: custom_tab_only={true/false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required, but the following optional queries can be added to the request.

Name	Reqd?	Type	Description
custom_tab_only	No	Boolean	When true, only custom tabs are returned in the response.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/tab_definitions

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns information for all custom tabs associated with the account. If the custom_tab_only=true query was included, only custom tabs are returned.

The following examples show the json response.

Example JSON Response Body

```
{
    "tabs": [
            {
                    "anchor": "string",
                    "anchorXOffset": "string",
                    "anchorYOffset": "string",
                    "anchorIgnoreIfNotPresent": "string",
                    "anchorUnits": "string",
                    "bold": "string",
                    "concealValueOnDocument": "string",
                    "disableAutoSize": "string",
                    "editable": "string",
                    "font": "string",
                    "fontColor": "string",
                    "fontSize": "string",
                    "height": "string",
                    "includeInEmail": "string",
                    "initialValue": "string",
                    "italic": "string",
```

```
                    "items": "string;string;string",
                    "locked": "string",
                    "maximumLength": "string",
                    "mergeField": {
                           "configurationType": "string",
                           "path": "string",
                           "writeBack": "string",
                           "allowSenderToEdit": "string"
                    },
                    "name": "string",
                    "required": "string",
                    "shared": "string",
                    "tabLabel": "string",
                    "type": "string",
                    "underline": "string",
                    "validationMessage": "string",
                    "validationPattern": "string",
                    "width": "string",
                    "customTabId": "string",
                    "lastModifiedByUserId": "string",
                    "lastModifiedByDisplayName": "string",
                    "lastModified": "string",
                    "createdByUserId": "string",
                    "createdByDisplayName": "string"
             }
     ]
}
```

Get Custom Tab Information

This retrieves information about the requested custom tab.

URL:

/accounts/{accountId}/tab_definitions/{customtabid}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/tab_definitions{customtabid}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the requested custom tab.

The following examples show the json and xml responses.

Example JSON Response Body

```
{
    "anchor": "string",
    "anchorXOffset": "string",
    "anchorYOffset": "string",
    "anchorIgnoreIfNotPresent": "string",
    "anchorUnits": "string",
    "bold": "string",
    "concealValueOnDocument": "string",
    "disableAutoSize": "string",
    "editable": "string",
    "font": "string",
    "fontColor": "string",
    "fontSize": "string",
    "height": "string",
    "includeInEmail": "string",
    "initialValue": "string",
    "italic": "string",
    "items": "string;string;string",
    "locked": "string",
    "maximumLength": "string",
    "mergeField": {
           "configurationType": "string",
           "path": "string",
           "writeBack": "string",
           "allowSenderToEdit": "string"
    },
    "name": "string",
    "required": "string",
    "shared": "string",
    "tabLabel": "string",
    "type": "string",
    "underline": "string",
    "validationMessage": "string",
    "validationPattern": "string",
    "width": "string",
    "customTabId": "string",
    "lastModifiedByUserId": "string",
    "lastModifiedByDisplayName": "string",
    "lastModified": "string",
    "createdByUserId": "string",
    "createdByDisplayName": "string"
}
```

Modify Custom Tab Information

This allows you to update information in a custom tab.

URL:

/accounts/{accountId}/tab_definitions/{customtabid}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

There are no required parameters for this request. Only information included in the request is updated in the profile.

Name	Reqd?	Type	Description
anchor	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
bold	No	Boolean	If true, the information in the tab is bold.
concealValueOnDocument	No	Boolean	If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text tabs and does not affect list tabs, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text tabs.
editable	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.

Name	Reqd?	Type	Description
font	No	String	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, MSGothic, and MSMincho.
fontColor	No	String	The font color used for the information in the tab. Possible values are: black, brightBlue, brightRed, darkGreen, darkRed, gold, green, navyBlue, purple, or white.
fontSize	No	String	The font size used for the information in the tab. Possible values are: size7, size8, size9, size10, size11, size12, size14, size16, size18, size20, size22, size24, size26, size28, size36, size48, or size72.
height	No	String	Height of the tab in pixels.
includeInEmail	No	Boolean	Only used for note tabs. If true, the text in the note tab is included in the email sent to the assigned recipient.
initialValue	No	String	The initial value of the tab.
italic	No	Boolean	If true, the information in the tab is italic.
items	No	Items	Only used with list tabs. This is a semi-colon separated list used to populate the possible selections for a dropdown list. These are mapped to the list tab – listItems – text information.
locked	No	Boolean	If true, the signer cannot change the data in the tab.
maximumLength	No	String	The maximum length of the tab, in characters.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. Note that radio tabs are not supported with the merge field option. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
name	Yes	String	The name associated with the custom tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
shared	No	Boolean	If true, other account users can use this custom tab.
tabLabel	Yes	String	The label used with the tab. This can be a maximum of 500 characters.
type	Yes	String	Sets the type of custom tab. Possible values are: approve, checkBox, company, date, dateSigned, decline, email, emailAddress, envelopeId, firstName, formula, fullName, initialHere, lastName, list, note, number, radio, signerAttachment, signHere, ssn, text, title, zip5, and zip5Dash4.
underline	No	Boolean	If true, the information in the tab is underlined.
validationMessage	No	String	Message to be displayed to the signer if the validation rule from validationPattern fails. This is optional and if not provided the default DocuSign message will be displayed. This can be a maximum of 250 characters.

Name	Reqd?	Type	Description
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.
width	No	String	Width of the tab.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/tab_definitions

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "anchor": "string",
    "anchorXOffset": "string",
    "anchorYOffset": "string",
    "anchorIgnoreIfNotPresent": "string",
    "anchorUnits": "string",
    "bold": "string",
    "concealValueOnDocument": "string",
    "disableAutoSize": "string",
    "editable": "string",
    "font": "string",
    "fontColor": "string",
    "fontSize": "string",
    "height": "string",
    "includeInEmail": "string",
    "initialValue": "string",
    "italic": "string",
    "items": "string;string;string",
    "locked": "string",
    "maximumLength": "string",
    "mergeField": {
          "configurationType": "string",
          "path": "string",
          "writeBack": "string",
          "allowSenderToEdit": "string"
    },
    "name": "string",
    "required": "string",
    "shared": "string",
    "tabLabel": "string",
    "type": "string",
    "underline": "string",
    "validationMessage": "string",
    "validationPattern": "string",
    "width": "string"
}
```

Response

The response returns a success of failure along with the modified tab information, including when the tab was modified and the user that modified the tab.

The following example shows the response json body.

Example Response Body

```
{
    "anchor": "string",
    "anchorXOffset": "string",
    "anchorYOffset": "string",
    "anchorIgnoreIfNotPresent": "string",
    "anchorUnits": "string",
    "bold": "string",
    "concealValueOnDocument": "string",
    "disableAutoSize": "string",
    "editable": "string",
    "font": "string",
    "fontColor": "string",
    "fontSize": "string",
    "height": "string",
    "includeInEmail": "string",
    "initialValue": "string",
    "italic": "string",
    "items": "string;string;string",
    "locked": "string",
    "maximumLength": "string",
    "mergeField": {
          "configurationType": "string",
          "path": "string",
          "writeBack": "string",
          "allowSenderToEdit": "string"
    },
    "name": "string",
    "required": "string",
    "shared": "string",
    "tabLabel": "string",
    "type": "string",
    "underline": "string",
    "validationMessage": "string",
    "validationPattern": "string",
    "width": "string",
    "customTabId": "string",
    "lastModifiedByUserId": "string",
    "lastModifiedByDisplayName": "string",
    "lastModified": "string",
    "createdByUserId": "string",
    "createdByDisplayName": "string"
}
```

Delete a Custom Tab

This deletes the specified custom tab.

URLs:

/accounts/{accountId}/tab_definitions/{customtabid}

Formats:

XML, JSON

HTTP method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/tab_definitions{customtabid}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the success or failure.

Get List of Templates

This retrieves a list of templates. For users with administrator privileges, it returns a list of all templates in the account. For non-administrator users, this returns templates created by or shared with the user.

The response can be filtered by adding optional query strings to the request.

URL:

/accounts/{accountId}/templates

Optional query strings: folder={string}, folder_ids={GUID, GUID}, include={string}, count={integer}, start_position={integer}, from_date={date/time}, to_date={date/time}, used_from_date={date/time}, used_to_date={date/time}, search_text={string}, order={string}, order_by={string}, user_filter={string}, shared_by_me={true/false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only information needed for this is the account ID, but the following optional queries can be added to the request.

Name	Reqd?	Type	Description
folder	No	String	The query value can be a folder ID GUID. The response will only return templates in the specified folder.
folder_ids	No		A comma separated list of folder ID GUIDs.
include	No	String	A comma separated list of additional template attributes to include in the response. Valid values are: recipients, folders, documents, custom_fields, and notifications.
count	No	Integer	Number of records to return in the cache.
start_position	No	Integer	The starting index for the first template shown in the response. This must be greater than or equal to 0 (zero).
from_date	No	Date/Time	Start of the search date range. Only returns templates created on or after this date/time. If no value is specified, there is no limit on the earliest date created.
to_date	No	Date/Time	End of the search date range. Only returns templates created up to this date/time. If no value is provided, this defaults to the current date.
used_from_date	No	Date/Time	Start of the search date range. Only returns templates used or edited on or after this date/time. If no value is specified, there is no limit on the earliest date used.
used_to_date	No	Date/Time	End of the search date range. Only returns templates used or edited up to this date/time. If no value is provided, this defaults to the current date.
search_text	No	String	The search text used to search the names of templates.
order	No	String	Sets the direction order used to sort the list. Valid values are: • asc = ascending sort order (a to z) • desc = descending sort order (z to a)

Name	Reqd?	Type	Description
order_by	No	String	Sets the file attribute used to sort the list. Valid values are: • name – template name • modified – date/time template was last modified. • used – date/time the template was last used.
user_filter	No	String	Sets if the templates shown in the response Valid values are: • owned_by_me – only shows templates the user owns. • shared_with_me – only shows templates that are shared with the user. • all – shows all templates owned or shared with the user.
shared_by_me	No	Boolean	If true, the response only includes templates shared by the user. If false, the response only returns template not shared by the user. If not specified, the response is not affected.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

This returns a list of templates for the user that match the query parameters. If the optional folder or folder_id queries are used, only the templates in the specified folders are returned.

Name	Description
resultSetSize	The number of results returned in this response.
startPosition	The index of the first element in the response set.
endPosition	The index of the last element in the response set.
totalSetSize	The total number of results from the call. This will always be greater than or equal to resultSetSize.
nextUri	Provides the Uri to the next chunk of records based on the search request. If the endPosition is the entire results of the search, this is null.

Name	Description
previousUri	Provides the Uri to the previous chunk of records based on the search request. If this response is the first response for the search, this is null.
folders	Folder information for the requested templates.
envelopeTemplates	The list of requested templates.

The following example shows the response json body.

Example Response Body

```
{
    "resultSetSize": "string",
    "startPosition": "string",
    "endPosition": "string",
    "totalSetSize": "string",
    "nextUri": "string",
    "previousUri": "string",
    "folders": [
            {
                    Note: folder definition omitted for brevity
            }
    ],
    "envelopeTemplates": [
            {
                    Note: template definition omitted for brevity
            }
    ]
}
```

Post Template

Saves a template definition using a multipart request.

Template Email Subject Merge Fields

This provides the ability to insert recipient name and email address merge fields into the email subject line when sending from a template.

The merge fields, based on the recipient's roleName, are added to the emailSubject when the template is created or when the template is used to create an envelope. After a template sender adds the name and email information for the recipient and sends the envelope, the recipient information is automatically merged into the appropriate fields in the email subject line.

Both the sender and the recipients will see the information in the email subject line for any emails associated with the template. This provides an easy way for senders to organize their envelope emails without having to open an envelope to check the recipient.

Note: If merging the recipient information into the subject line causes the subject line to exceed 100 characters, then any characters over the 100 character limit are not included in the subject line. For cases where the recipient name or email is expected to be long, you should consider placing the merge field at the start of the email subject.

- To add a recipient's name in the subject line add the following text in the emailSubject when creating the template or when sending an envelope from a template:

  [[<*roleName*>_UserName]]

  Example:

```
  "emailSubject":"[[Signer 1_UserName]], Please sign this NDA",
```

- To add a recipient's email address in the subject line add the following text in the emailSubject when creating the template or when sending an envelope from a template:

  [[<*roleName*>_Email]]

  Example:

```
  "emailSubject":"[[Signer 1_Email]], Please sign this NDA",
```

  In both cases the <*roleName*> is the recipient's roleName in the template.

For cases where another recipient (such as an Agent, Editor, or Intermediary recipient) is entering the name and email information for the recipient included in the email subject, then [[<*roleName*>_UserName]] or [[<*roleName*>_Email]] is shown in the email subject.

URL:

/accounts/{accountId}/templates

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
accessibility	No	String	An optional element that can only be used if Document Accessibility is enabled for the account. This sets the document reading zones for screen reader applications. **Note:** This information is currently generated from the DocuSign web console by setting the reading zones when creating a template, exporting the reading zone string information, and adding it here.
allowMarkup	No	String	If true, Document Markup is enabled for envelope. Account must have Document Markup enabled to use this.

Name	Reqd?	Type	Description
allowReassign	No	String	If true, the recipient can redirect an envelope to a more appropriate recipient.
allowRecipientRecursion	No	String	When set to true, this enables the Recursive Recipients feature and allows a recipient to appear more than once in the routing order.
asynchronous	No	String	If true, will queue the envelope for processing and the envelope status will have a value of 'Processing'. Additionally, get status calls will return 'Processing' until completed. **Note:** if this is used a transactionId is required for this to work correctly. When the envelope is created, the status is processing and an envelopeId is not returned in the response. To get the envelopeId, use a GET envelope query using the transactionId (example: GET https://{server}/restapi /{apiVersion}/accounts/{accountId} /envelopes?transaction_ids=XXX) or by checking the Connect notification.
authoritativeCopy	No	String	Specifies the Authoritative copy feature. If set to true the Authoritative copy feature is enabled.
autoNavigation	No	String	If true, auto-navigation is enabled for the envelope. The auto-navigation method used is determined by the account setting.
brandId	No	String	This sets the brand profile format used for the envelope. The value in the string is the brandId associated with the profile. Account branding must be enabled for the account to use this option.
emailBlurb	No	String	Optional element. This is the same as the email body. If specified it is included in email body for all envelope recipients.
emailSubject	Yes	String	The subject of the email that will be sent to all recipients. See Template Email Subject Merge Fields for information about adding merge field information to the email subject.
enableWetSign	No	String	If true, the signer is allowed to print the document and sign it on paper.

Name	Reqd?	Type	Description
enforceSignerVisibility	No	String	It true, the signer is required to have a signature or initial tab on the document or that the document has no signers in order to view it. Account must have Document Visibility enabled to use this.
envelopeIdStamp	No	String	If true, Envelope ID Stamping is enabled.
eventNotification	No	Event Notification	This optional complex element allows a message to be sent a specified URL when the envelope or recipient changes status. It is similar to DocuSign Connect. For example, if an envelope changes from "Sent" to "Delivered", a message containing the updated envelope status and optionally the documents is sent to the URL. When an eventNotification is attached to an envelope using the API, it only applies to the envelope (treating the envelope as the sender). This is different from envelopes created through the console user interface, where the user is treated as the sender. eventNotification consists of: • url – The endpoint where envelope updates are sent. This will accept XML unless “useSoapInterface” is set to true. • loggingEnabled – When set to true, logging is turned on for envelope events on the Web Console Connect page. • requireAcknowledgment – When set to true, the DocuSign Connect service checks that the message was received and retries on failures. • useSoapInterface – When set to true, this tells the Connect service that the user’s endpoint has implemented a SOAP interface. • soapNameSpace – This lists the namespace in the SOAP listener provided. • includeCertificateWithSoap – When set to true, this tells the Connect service to send the DocuSign signedby certificate as part of the

Name	Reqd?	Type	Description

			outgoing SOAP xml. This appears in the XML as wsse:BinarySecurityToken.
			• signMessageWithX509Cert – When set to true, messages are signed with an X509 certificate. This provides support for 2-way SSL in the envelope.
			• includeDocuments – When set to true, the PDF documents are included in the message along with the updated XML.
			• includeEnvelopeVoidReason – When set to true, this tells the Connect Service to include the void reason, as entered by the person that voided the envelope, in the message.
			• includeTimeZone – When set to true, the envelope time zone information is included in the message.
			• includeSenderAccountAsCustomField – When set to true, the sender account ID is included as a envelope custom field in the data.
			• includeDocumentFields – When set to true, this tells the Connect Service to include the Document Fields associated with the envelope. Document Fields are optional custom name-value pairs added to documents using the API.
			• includeCertificateOfCompletion – When set to true, this tells the Connect Service to include the Certificate of Completion with completed envelopes.
			• envelopeEvents – The list of envelope-level events statuses that will trigger Connect to send updates to the url. It can be a two-part list with:
			  ○ envelopeEventStatusCode – The envelope status, this can be Sent, Delivered, Completed, Declined, or Voided.
			  ○ includeDocuments – When set to true, the envelope time zone

Name	Reqd?	Type	Description

			information is included in the message.
			• recipientEvents – The list of recipient event statuses that will trigger Connect to send updates to the url. It can be a two-part list with:
			  ○ recipientEventStatusCode – The recipient status, this can be Sent, Delivered, Completed, Declined, AuthenticationFailed, and AutoResponded.
			  ○ includeDocuments – When set to true, the envelope time zone information is included in the message.

messageLock	No	String	If true, prevents senders from changing the emailBlurb and emailSubject for the envelope. Additionally, this prevents users from making changes to the emailBlurb and emailSubject when correcting envelopes However, if the messageLock node is set to True and the emailSubject is empty, senders and correctors are able to add a subject to the envelope.
recipientsLock	No	String	If true, prevents senders from changing, correcting, or deleting the recipient information for the envelope.
signingLocation	No	String	Specifies the physical location where the signing takes place. It can have two enumeration values; InPerson and Online.
customFields	No		customFields are part of the json structure, but they are ignored when Posting. They are set in the console.
documents	Yes	Document	Complex element contains the details on the documents in the envelope. See the Document parameter section for more information.

Name	Reqd?	Type	Description
recipients	Yes	Recipient	Specifies the envelope recipients. It uses the same information as a standard envelope recipient, with the following exceptions: • The email and name elements are not required. • The roleName element is required. This can be a maximum of 100 characters. See the Recipient parameter sections for more information.
envelopeTemplateDefinition	Yes	Definition	A complex element with the following information: templateId: Unique identifier of the template. If this is not provided, DocuSign will generate a value. name: Name of the template. This can be a maximum of 100 characters. shared: If true, the template is shared with the Everyone group in the account. If false, the template is only shared with the Administrator group. password: Password, if the template is locked. description: Description of the template. This can be a maximum of 500 characters. pageCount: Number of document pages in the template. folderName: The name of the folder the template is located in. folderId: The ID for the folder. owner: The userName, email, userId, userType, and userStatus for the template owner.
notification	No		An optional complex element that specifies the notification options for the envelope. It consists of: • useAccountDefaults – When true, the account default notification settings are used for the envelope. • reminders – A complex element that specifies reminder settings for the envelope. It consists of:

Name	Reqd?	Type	Description
			○ reminderEnabled – When true a reminder message is sent to the recipient. ○ reminderDelay – An interger that sets the number of days after the recipient receives the envelope that reminder emails are sent to the recipient. ○ reminderFrequency – An interger that sets the interval, in days, between reminder emails. • expirations – A complex element that specifies the expiration settings for the envelope. It consists of: ○ expireEnabled – When true the envelope expires (is no longer available for signing) in the set number of days. If false, the account default setting is used. If the account does not have an expiration setting, the DocuSign default value of 120 days is used. ○ expireAfter – An integer that sets the number of days the envelope is active. ○ expireWarn – An integer that sets the number of days before envelope expiration that an expiration warning email is sent to the recipient. If set to 0 (zero), no warning email is sent.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: multipart/form-data; boundary=AAA

--AAA
Content-Type: application/json
Content-Disposition: form-data

{
```

```
    "envelopeTemplateDefinition": {
           "templateId": "String content",
           "name": "String content",
           "shared": "String content",
           "password": "String content",
           "description": "String content",
           "lastModified": "String content",
           "pageCount": "String content",
           "folderName": "String content",
           "folderId": "String content",
           "owner": {
                  "userName": "String content",
                  "email": "String content",
                  "userId": "String content",
                  "userType": "String content",
                  "userStatus": "String content",
           }
    }
    "allowMarkup":"String content",
    "allowReassign":"String content",
    "allowRecipientRecursion":"String content",
    "asynchronous":"String content",
    "authoritativeCopy":""String content",
    "autoNavigation":"String content",
    "emailBlurb":"String content",
    "emailSubject":"String content",
    "enableWetSign":"String content",
    "enforceSignerVisibility":"String content",
    "signingLocation":"String content",
    "envelopeIdStamping":"String content",
    "eventNotification":{
           "url":"String content",
           "loggingEnabled":"String content",
           "requireAcknowledgment":"String content",
           "useSoapInterface":"String content",
           "soapNameSpace" : " String content",
           "includeCertificateWithSoap" : " String content",
           "signMessageWithX509Cert":"String content",
           "includeDocuments":"String content",
           "includeEnvelopeVoidReason": "String content",
           "includeTimeZone": " String content",
           "includeSenderAccountAsCustomField" : " String content",
           "includeDocumentFields": " String content",
           "includeCertificateOfCompletion": " String content"
           "envelopeEvents" : [
                  {
                         "envelopeEventStatusCode" : "sent"
                  },
                  {
                         "envelopeEventStatusCode" : "completed",
                         "includeDocuments" : "true"
                  }
           ],
           "recipientEvents" : [
                  {
                         "recipientEventStatusCode" : "Sent"
                  },
                  {
                         "recipientEventStatusCode" : "completed",
                         "includeDocuments" : "true"
                  }
           ]
    },
    "customFields":{
```

```
            "listCustomFields":[{
                    "name":"String content",
                    "required":"String content",
                    "show":"String content",
                    "value":"String content",
                    "listItems":["String content"]
            }],
            "textCustomFields":[{
                    "name":"String content",
                    "required":"String content",
                    "show":"String content",
                    "value":"String content",
            }]
    },
    "notification": {
            "useAccountDefaults": "sample string 1",
            "reminders": {
                    "reminderEnabled": "sample string 1",
                    "reminderDelay": "sample string 2",
                    "reminderFrequency": "sample string 3"
            },
            "expirations": {
                    "expireEnabled": "sample string 1",
                    "expireAfter": "sample string 2",
                    "expireWarn": "sample string 3"
            }
    },
    "documents": [{
            "documentId": "1",
            "name": "test1.pdf"
    }],
    "recipients": {
            "signers": [{
                    "accessCode": "",
                    "addAccessCodeToEmail": false,
                    "clientUserId": null,
                    "customFields": null,
                    "emailNotification": null,
                    "idCheckConfigurationName": null,
                    "idCheckInformationInput": null,
                    "inheritEmailNotificationConfiguration": false,
                    "note": "",
                    "phoneAuthentication": null,
                    "recipientAttachments": null,
                    "recipientId": "1",
                    "requireIdLookup": false,
                    "roleName": "Signer 1",
                    "routingOrder": 1,
                    "socialAuthentications": null,
                    "templateAccessCodeRequired": false,
                    "templateLocked": false,
                    "templateRequired": false,
                    "email": "",
                    "name": "",
                    "autoNavigation": false,
                    "defaultRecipient": false,
                    "signatureInfo": null,
                    "tabs": {
                            "approveTabs": null,
                            "checkboxTabs": null,
                            "companyTabs": null,
                            "dateSignedTabs": null,
                            "dateTabs": null,
                            "declineTabs": null,
```

```
                        "emailTabs": null,
                        "envelopeIdTabs": null,
                        "fullNameTabs": null,
                        "initialHereTabs": null,
                        "listTabs": null,
                        "noteTabs": null,
                        "numberTabs": null,
                        "radioGroupTabs": null,
                        "signHereTabs": [{
                               "anchorString": null,
                               "anchorXOffset": null,
                               "anchorYOffset": null,
                               "anchorIgnoreIfNotPresent": null,
                               "anchorUnits": null,
                               "conditionalParentLabel": null,
                               "conditionalParentValue": null,
                               "documentId": "1",
                               "pageNumber": "1",
                               "recipientId": "1",
                               "templateLocked": false,
                               "templateRequired": false,
                               "xPosition": "157",
                               "yPosition": "7",
                               "name": "Sign Here",
                               "optional": false,
                               "scaleValue": 1,
                               "tabLabel": "Signature 1"
                        }],
                        "signerAttachmentTabs": null,
                        "ssnTabs": null,
                        "textTabs": null,
                        "titleTabs": null,
                        "zipTabs": null
                }
        },
        {
                "accessCode": "",
                "addAccessCodeToEmail": false,
                "customFields": null,
                "emailNotification": null,
                "idCheckConfigurationName": null,
                "idCheckInformationInput": null,
                "inheritEmailNotificationConfiguration": false,
                "note": "",
                "phoneAuthentication": null,
                "recipientAttachments": null,
                "recipientCaptiveInfo": null,
                "recipientId": "2",
                "requireIdLookup": false,
                "roleName": "Signer 2",
                "routingOrder": 1,
                "socialAuthentications": null,
                "templateAccessCodeRequired": false,
                "templateLocked": false,
                "templateRequired": false,
                "email": "",
                "name": "",
                "autoNavigation": false,
                "defaultRecipient": false,
                "signatureInfo": null,
                "tabs": {
                        "approveTabs": null,
                        "checkboxTabs": null,
                        "companyTabs": null,
```

```
                        "dateSignedTabs": null,
                        "dateTabs": null,
                        "declineTabs": null,
                        "emailTabs": null,
                        "envelopeIdTabs": null,
                        "fullNameTabs": null,
                        "initialHereTabs": null,
                        "listTabs": null,
                        "noteTabs": null,
                        "numberTabs": null,
                        "radioGroupTabs": null,
                        "signHereTabs": [{
                               "anchorString": null,
                               "anchorXOffset": null,
                               "anchorYOffset": null,
                               "anchorIgnoreIfNotPresent": null,
                               "anchorUnits": null,
                               "conditionalParentLabel": null,
                               "conditionalParentValue": null,
                               "documentId": "1",
                               "pageNumber": "1",
                               "recipientId": "2",
                               "templateLocked": false,
                               "templateRequired": false,
                               "xPosition": "351",
                               "yPosition": "12",
                               "name": "Sign Here",
                               "optional": false,
                               "scaleValue": 1,
                               "tabLabel": "Signature 2"
                        }],
                        "signerAttachmentTabs": null,
                        "ssnTabs": null,
                        "textTabs": null,
                        "titleTabs": null,
                        "zipTabs": null
                  }
            }]
    },
}

--AAA
Content-Type: application/pdf
Content-Disposition: file; filename="test1.pdf"; documentId=1

<PDF Bytes Rmeoved>
```

Response

The response returns the template name, template ID and template uri.

The following example shows the header followed by the response json body.

Example Response Body

```
{
    "name":"String content",
    "templateId":"String content",
    "uri":"String content"
}
```

Get Template

This retrieves the definition of the specified template.

URL:

/accounts/{accountId}/templates/{templateId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the template ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the template definition of the requested template, along with information about when the template last modified and which user modified the template. Refer to Post Template for information about the template parameters.

The following example shows the response json body.

Example Response Body

```
{
    "envelopeTemplateDefinition": {
           "templateId": "String content",
           "name": "String content",
           "shared": "String content",
           "password": "String content",
           "description": "String content",
           "lastModified": "string",
           "lastModifiedBy": {
                  "userName": "string",
                  "email": "string",
                  "userId": "string",
                  "uri": "string"
           },
           "pageCount": "String content",
           "folderName": "String content",
           "folderId": "String content",
           "owner": {
```

```
                    "userName": "String content",
                    "email": "String content",
                    "userId": "String content",
                    "userType": "String content",
                    "userStatus": "String content",
            }
    }
    "allowMarkup":"String content",
    "allowReassign":"String content",
    "allowRecipientRecursion":"String content",
    "asynchronous":"String content",
    "authoritativeCopy":""String content",
    "customFields":{
            "listCustomFields":[{
                    "name":"String content",
                    "required":"String content",
                    "show":"String content",
                    "value":"String content",
                    "listItems":["String content"]
            }],
            "textCustomFields":[{
                    "name":"String content",
                    "required":"String content",
                    "show":"String content",
                    "value":"String content"
            }]
    },
    "documents":[{
            "documentId":"String content",
            "fileExtension":"String content",
            "matchBoxes":[{
            }],
            "name":"String content",
            "password":"String content",
            "uri":"String content"
    }],
    "emailBlurb":"String content",
    "emailSubject":"String content",
    "enableWetSign":"String content",
    "enforceSignerVisibility":"String content",
    "notification":{
            "expirations":{
                    "expireAfter":"String content",
                    "expireEnabled":"String content",
                    "expireWarn":"String content"
            },
            "reminders":{
                    "reminderDelay":"String content",
                    "reminderEnabled":"String content",
                    "reminderFrequency":"String content"
            },
            "useAccountDefaults":"String content"
    },
    "recipients":{
    },
    "signingLocation":"String content",
}
```

Modify a Template

This allows users to modify an existing template.

URL:

/accounts/{accountId}/templates/{templateId}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

This uses the same parameters as creating a new template, but only the parameters being changed are required. Refer to Post Template for information about the template parameters.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

--AAA
Content-Type: application/json
Content-Disposition: form-data

{
    "envelopeTemplateDefinition": {
           "templateId": "String content",
           "name": "String content",
           "shared": "String content",
           "password": "String content",
           "description": "String content",
           "lastModified": "String content",
           "pageCount": "String content",
           "folderName": "String content",
           "folderId": "String content",
           "owner": {
                  "userName": "String content",
                  "email": "String content",
                  "userId": "String content",
                  "userType": "String content",
                  "userStatus": "String content",
           }
    }
    "allowMarkup":"String content",
    "allowReassign":"String content",
    "allowRecipientRecursion":"String content",
    "asynchronous":"String content",
    "authoritativeCopy":""String content",
    "autoNavigation":"String content",
    "emailBlurb":"String content",
    "emailSubject":"String content",
    "enableWetSign":"String content",
    "enforceSignerVisibility":"String content",
```

```
    "signingLocation":"String content",
    "envelopeIdStamping":"String content",
    "eventNotification":{
           "url":"String content",
           "loggingEnabled":"String content",
           "requireAcknowledgment":"String content",
           "useSoapInterface":"String content",
           "soapNameSpace" : "http://DocuSignConnectListener",
           "includeCertificateWithSoap" : "true",
           "envelopeEvents" : [{
                  "envelopeEventStatusCode" : "sent"
           },
           {
                  "envelopeEventStatusCode" : "completed",
                  "includeDocuments" : "true"
           }]
    }],
    "customFields":{
           "listCustomFields":[{
                  "name":"String content",
                  "required":"String content",
                  "show":"String content",
                  "value":"String content",
                  "listItems":["String content"]
           }],
           "textCustomFields":[{
                  "name":"String content",
                  "required":"String content",
                  "show":"String content",
                  "value":"String content",
           }]
    },
    "notification": {
           "useAccountDefaults": "sample string 1",
           "reminders": {
                  "reminderEnabled": "sample string 1",
                  "reminderDelay": "sample string 2",
                  "reminderFrequency": "sample string 3"
           },
           "expirations": {
                  "expireEnabled": "sample string 1",
                  "expireAfter": "sample string 2",
                  "expireWarn": "sample string 3"
           }
    },
    "documents": [{
           "documentId": "1",
           "name": "test1.pdf"
    }],
    "recipients": {
           "signers": [{
                  "accessCode": "",
                  "addAccessCodeToEmail": false,
                  "clientUserId": null,
                  "customFields": null,
                  "emailNotification": null,
                  "idCheckConfigurationName": null,
                  "idCheckInformationInput": null,
                  "inheritEmailNotificationConfiguration": false,
                  "note": "",
                  "phoneAuthentication": null,
                  "recipientAttachments": null,
                  "recipientId": "1",
                  "requireIdLookup": false,
```

```
                "roleName": "Signer 1",
                "routingOrder": 1,
                "socialAuthentications": null,
                "templateAccessCodeRequired": false,
                "templateLocked": false,
                "templateRequired": false,
                "email": "",
                "name": "",
                "autoNavigation": false,
                "defaultRecipient": false,
                "signatureInfo": null,
                "tabs": {
                        "approveTabs": null,
                        "checkboxTabs": null,
                        "companyTabs": null,
                        "dateSignedTabs": null,
                        "dateTabs": null,
                        "declineTabs": null,
                        "emailTabs": null,
                        "envelopeIdTabs": null,
                        "fullNameTabs": null,
                        "initialHereTabs": null,
                        "listTabs": null,
                        "noteTabs": null,
                        "numberTabs": null,
                        "radioGroupTabs": null,
                        "signHereTabs": [{
                                "anchorString": null,
                                "anchorXOffset": null,
                                "anchorYOffset": null,
                                "anchorIgnoreIfNotPresent": null,
                                "anchorUnits": null,
                                "conditionalParentLabel": null,
                                "conditionalParentValue": null,
                                "documentId": "1",
                                "pageNumber": "1",
                                "recipientId": "1",
                                "templateLocked": false,
                                "templateRequired": false,
                                "xPosition": "157",
                                "yPosition": "7",
                                "name": "Sign Here",
                                "optional": false,
                                "scaleValue": 1,
                                "tabLabel": "Signature 1"
                        }],
                        "signerAttachmentTabs": null,
                        "ssnTabs": null,
                        "textTabs": null,
                        "titleTabs": null,
                        "zipTabs": null
                }
        },
        {
                "accessCode": "",
                "addAccessCodeToEmail": false,
                "customFields": null,
                "emailNotification": null,
                "idCheckConfigurationName": null,
                "idCheckInformationInput": null,
                "inheritEmailNotificationConfiguration": false,
                "note": "",
                "phoneAuthentication": null,
                "recipientAttachments": null,
```

```
                    "recipientCaptiveInfo": null,
                    "recipientId": "2",
                    "requireIdLookup": false,
                    "roleName": "Signer 2",
                    "routingOrder": 1,
                    "socialAuthentications": null,
                    "templateAccessCodeRequired": false,
                    "templateLocked": false,
                    "templateRequired": false,
                    "email": "",
                    "name": "",
                    "autoNavigation": false,
                    "defaultRecipient": false,
                    "signatureInfo": null,
                    "tabs": {
                          "approveTabs": null,
                          "checkboxTabs": null,
                          "companyTabs": null,
                          "dateSignedTabs": null,
                          "dateTabs": null,
                          "declineTabs": null,
                          "emailTabs": null,
                          "envelopeIdTabs": null,
                          "fullNameTabs": null,
                          "initialHereTabs": null,
                          "listTabs": null,
                          "noteTabs": null,
                          "numberTabs": null,
                          "radioGroupTabs": null,
                          "signHereTabs": [{
                                "anchorString": null,
                                "anchorXOffset": null,
                                "anchorYOffset": null,
                                "anchorIgnoreIfNotPresent": null,
                                "anchorUnits": null,
                                "conditionalParentLabel": null,
                                "conditionalParentValue": null,
                                "documentId": "1",
                                "pageNumber": "1",
                                "recipientId": "2",
                                "templateLocked": false,
                                "templateRequired": false,
                                "xPosition": "351",
                                "yPosition": "12",
                                "name": "Sign Here",
                                "optional": false,
                                "scaleValue": 1,
                                "tabLabel": "Signature 2"
                          }],
                          "signerAttachmentTabs": null,
                          "ssnTabs": null,
                          "textTabs": null,
                          "titleTabs": null,
                          "zipTabs": null
                    }
            }]
    },
}

--AAA
Content-Type: application/pdf
Content-Disposition: file; filename="test1.pdf"; documentId=1

<PDF Bytes Rmeoved>
```

Response

The response returns success or failure and the template information.

The following example shows the header followed by the response json body.

Example Response Body

```
{
    "name":"String content",
    "templateId":"String content",
    "uri":"String content"
}
```

POST Template Edit View

This provides a URL to the template edit view of the DocuSign UI for the specified template ID. This is a one-time use login token that allows the user to be placed into the DocuSign sending view.

Note that when using the response URL you can set the default UI page shown to a user by using the send query string parameter on the end of the URL. Setting send=0 starts the user on the prepare page where the sender can add documents and recipients, while setting send=1 starts the user on the tagging page. The setting defaults to the prepare page.

Upon completion, the user is returned to the return URL provided by the API application.

Important: iFrames should not be used for embedded operations on mobile devices due to screen space issues. For iOS devices DocuSign recommends using a WebView.

URL:

/accounts/{accountId}/templates/{templateId}/views/edit

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Note: When locking a template, the templateId is used in place of the envelopeId.

Name	Reqd?	Type	Description
returnUrl	No	String	Identifies the return point after editing a template. **Important** – You must include HTTPS:// in the URL or the redirect might be blocked by some browsers. DocuSign returns to the URL and includes an event parameter that can be used to redirect the recipient to another location. The possible event parameters returned are: • save (user saves the template) • cancel (user cancels template editing) • error (there is an error when performing the edit) • sessionEnd (the sending session ends before the user completes another action)

Example Request Body

```
POST https://{SERVER}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/views/edit

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "returnUrl":"String content"
}
```

Response

The response returns the template edit URL.

When using the response URL you can set the default UI page shown to a user by using the send query string parameter on the end of the URL. Setting send=0 starts the user on the prepare page where the sender can add documents and recipients, while setting send=1 starts the user on the tagging page. The setting defaults to the prepare page.

Example: For the URL - `https://docusign.net/Member/StartInSession.aspx?send=1`

The user will start on the tagging page of the template view.

The following example shows the response json body.

```
{
    "url":"string"
}
```

Get Custom Document Fields for a Template Document

This returns the custom document fields for an existing document in a template.

URL:

/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/documents/{documentId}/fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the custom document field name-value pairs for the requested document ID.

Examples of the response body for json and XML are shown below.

Example JSON Response Body

```
{
  "documentFields": [
    {
      "name": "sample string",
      "value": "sample string"
    },
    {
      "name": "sample string",
      "value": "sample string"
    }
  ]
}
```

Example XML Response Body

Note that the structure of the XML response is slightly different than the json response, in that the name/value pairs are included in a nameValue element.

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
    <nameValue>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Add Custom Document Fields to a Template Document

This adds custom document fields for an existing template document.

URL:

/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
documentFields	Yes	documentField	The array of name-value custom data strings to be added to a document. Custom document field information is returned in the status, but otherwise is not used by DocuSign. The array contains the elements: name – A string that can be a maximum of 50 characters. value – A string that can be a maximum of 200 characters. **IMPORTANT:** If using xml, the name/value pair is contained in a nameValue element.

Example Request JSON Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/documents/{documentId}/fields
```

```
X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Example XML Request Body

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Response

The response returns the custom document field name-value pairs added to the document and any errors associated with the fields.

Examples of the response body for json and XML are shown below.

Example JSON Response Body

```
{
  "documentFields": [
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    },
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
```

```
        "message": "sample string"
      }
    }
  ]
}
```

Example XML Response Body

Note that the structure of the XML response is slightly different than the json response, in that the name/value pairs are included in a nameValue element.

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Modify Custom Document Fields for a Template Document

This modifies existing custom document fields for an existing template document.

URL:

/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
documentFields	Yes	documentField	The array of name-value custom data strings to be added to a document. Custom document

Name	Reqd?	Type	Description
			field information is returned in the status, but otherwise is not used by DocuSign. The array contains the elements: name – A string that can be a maximum of 50 characters. value – A string that can be a maximum of 200 characters. **IMPORTANT:** If using xml, the name/value pair is contained in a nameValue element.

Example JSON Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/documents/{documentId}/fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Example XML Request Body

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Response

The response returns the modified custom document field name-value pairs for the document and any errors associated with the fields.

Examples of the response body for json and XML are shown below.

Example JSON Response Body

```
{
  "documentFields": [
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    },
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    }
  ]
}
```

Example XML Response Body

Note that the structure of the XML response is slightly different than the json response, in that the name/value pairs are included in a nameValue element.

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Delete Custom Document Fields from a Template Document

This deletes custom document fields for an existing template document.

URL:

/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
documentFields	Yes	documentField	The array of name-value custom data strings to be added to a document. Custom document field information is returned in the status, but otherwise is not used by DocuSign. The array contains the elements: name – A string that can be a maximum of 50 characters. value – A string that can be a maximum of 200 characters. **IMPORTANT:** If using xml, the name/value pair is contained in a nameValue element.

Example JSON Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/documents/{documentId}/fields

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Example XML Request Body

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
    <nameValue>
      <name>sample string 1</name>
      <value>sample string 2</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Response

The response returns success or failure and any error messages. The response also returns the list of custom document field name-value pairs that were removed from the document.

Examples of the response body for json and XML are shown below.

Example JSON Response Body

```
{
  "documentFields": [
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    },
    {
      "name": "sample string",
      "value": "sample string",
      "errorDetails": {
        "errorCode": "sample string",
        "message": "sample string"
      }
    }
  ]
}
```

Example XML Response Body

Note that the structure of the XML response is slightly different than the json response, in that the name/value pairs are included in a nameValue element.

```
<documentFieldsInformation xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.docusign.com/restapi">
  <documentFields>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
```

```
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
    <nameValue>
      <errorDetails>
        <errorCode>sample string</errorCode>
        <message>sample string</message>
      </errorDetails>
      <name>sample string</name>
      <value>sample string</value>
    </nameValue>
  </documentFields>
</documentFieldsInformation>
```

Share a Template with a Group

This shares a template with a group.

URL:

/accounts/{accountId}/templates/{templateId}/groups

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
groupId	Yes	String	The group ID number.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "groups":[{
           "groupId":"String content",
    }]
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed. The response contains a group structure similar to the request. If an error occurred during the PUT operation for any of the groups, that group will contain an errorDetails node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "groups":[{
           "errorDetails":{
                  "errorCode":"String content",
                  "message":"String content"
           },
           "groupId":"String content",
           "groupName":"String content",
           "groupType":"String content",
           "permissionProfileId":"String content"
    }]
}
```

Remove Template Sharing for a Group

This removes template sharing for a group.

URL:

/accounts/{accountId}/templates/{templateId}/groups

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
groupId	Yes	String	The group ID number.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/groups

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

```
{
    "groups":[{
           "groupId":"String content",
    }]
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed. The response contains a group structure similar to the request. If an error occurred during the DELETE operation for any of the groups, that group will contain an errorDetails node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "groups":[{
           "errorDetails":{
                  "errorCode":"String content",
                  "message":"String content"
           },
           "groupId":"String content",
           "groupName":"String content",
           "groupType":"String content",
           "permissionProfileId":"String content"
    }]
}
```

Get Template Recipient Information

This returns the information for all recipients in a single template.

URL:

/accounts/{accountId}/templates/{templateId}/recipients

Optional query items: include_tabs={true or false}, include_extended={true or false}, include_anchor_tab_locations={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query parameters can be added.

Name	Reqd?	Type	Description
include_tabs	No	Boolean	When true the tab information associated with the recipient is included in the response.
include_extended	No	Boolean	When true the extended properties are included in the response.
include_anchor_tab_locations	No	Boolean	When true and include_tabs=true, all tabs with anchor tab properties are included in the response.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the recipients in the template. The recipient information includes the recipient name, email, ID, recipient type, routing order, tab information (if included in the query), and delivery method.

The following example shows the response json body.

Example Response Body

```
{
    "agents":[]
    "carbonCopies":[],
    "certifiedDeliveries":[],
    "currentRoutingOrder":"String content",
    "editors":[],
    "inPersonSigners":[],
    "intermediaries":[],
    "recipientCount":"String content",
    "signers":[{
           "deliveredDateTime": "String content",
           "recipientAuthenticationStatus":{
                  "(authentication status result)":
                         "eventTimestamp":"String content"
                         "status":"String content"
                  }
           },
           "recipientId": "String content",
           "requireIdLookup": "String content",
           "roleName":"String content",
           "routingOrder": "String content",
           "signedDateTime": "String content",
           "status": "String content",
           "email": "String content",
           "name": "String content",
           "deliveryMethod": "String content"
```

```
    }]
}
```

Add or Replace a Template Bulk Recipient File

This adds or replaces a bulk recipient file to a template. The Content-Type supported for uploading a bulk recipient file is CSV (text/csv).

The REST API does not support modifying individual rows or values in the bulk recipients file. It only allows the entire file to be added or replaced with a new file.

URL:

/accounts/{accountId}/templates/{templateId}/recipients/{recipientId}/bulk_recipients

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The only required parameters are the template ID and recipient ID.

The parameters listed below are the supported fields for the csv file being added to the template. The first row of the csv file is the header row that must have the field names for the file. Each subsequent row represents a unique recipient with the information for that recipient.

There can be a maximum of 1,000 rows in the bulk recipient csv file.

If the value you are adding has a comma or double-quotation marks (“), the value must be enclosed in double-quotation marks (“). Example: if you have a Text tab with a tabLabel of Group and one of the entries for a recipient is Inside Sales, NE, the would use “Inside Sales, NE” in the Group column for that recipient.

Name	Reqd?	Type	Description
Name	Yes	String	The recipient’s name. This can be a maximum of 50 characters.
Email	Yes	Sting	The recipient’s email address. This can be a maximum of 100 characters.
Note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will also display in the signing UI. This can be a maximum of 1000 characters.

Name	Reqd?	Type	Description
AccessCode	No	String	If a value is provided, the recipient must enter the value as the access code to view and sign the envelope. This can be a maximum of 50 characters and must conform to account's access code format setting. If blank, but the signer accessCode is specified in the template, then that value is used. If blank and the signer accessCode is not specified, then access code is not required.
Identification	No	String	Specifies the authentication check used for the signer. If blank then no authentication check is required for the signer. Only one value can be used in this field. The acceptable values are: • KBA: Enables the normal ID check authentication set up for your account. • Phone: Enables phone authentication. • SMS: Enables SMS authentication.
PhoneNumber	No	String	This is only used if the **Identification** field has **Phone** or **SMS**. The value for this field can be a valid telephone number or, if Phone, usersupplied (SMS authentication cannot use a user supplied number). Parenthesis and dashes can be used in the telephone number. If usersupplied is used, the signer supplies his or her own telephone number.
{tabLabel}	No	String	This is used to populate recipient tabs with information. This allows each bulk recipient signer to have different values for their associated tabs. Any number of tabLabel columns can be added to the bulk recipient file. The information used in the bulk recipient file header must be the same as the tabLabel for the tab. The values entered in this column are automatically inserted into the corresponding tab for the recipient in the same row. Note that this option cannot be used for tabs that do not have data or that are automatically populated data such as Signature, Full Name, Email Address, Company, Title, and Date Signed tabs.

Example Bulk Recipient CSV File

```
Name,Email,Note,AccessCode,Identification,PhoneNumber,address1
David Jones,david.jones@yahoo.com,Here is the document we discussed.,,ID Check,,123 Main
St
Kevin Smith,kevinmith@yahoo.com,,2243,,,697 My Way
Elisabeth Bozick,elisabeth.bozick@yahoo.com,,,phone,usersupplied,827 1st Ave
```

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}//accounts/{accountId}/templates/
{templateId}/recipients/{recipientId}/bulk_recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Content-Type: application/json
Content-Disposition: file;filename={file name};fileExtension=.csv
```

Response

The response returns the recipient information for each row of the bulk recipient csv file, the total number of recipients in the file, and the bulkRecipientUri.

Example Response

```
{
  "bulkRecipients": [
    {
      "rowNumber": "string",
      "email": "string",
      "name": "string",
      "note": "string",
      "accessCode": "string",
      "identification": "string",
      "phoneNumber": "string",
      "tabLabels": [
        {
          "name": "string,"
          "value": "string"
        }
      ]
    },
    {
      "rowNumber": "string",
      "email": "string",
      "name": "string",
      "note": "string",
      "accessCode": "string",
      "identification": "string",
      "phoneNumber": "string",
      "tabLabels": [
        {
          "name": "string,
          "value": "string"
        }
      ]
    }
  ],
```

```
  "bulkRecipientsCount": "string",
  "bulkRecipientsUri": "string"
}
```

Retrieve Template Bulk Recipient File

This is used to retrieve the bulk recipient file information for a template with a bulk recipient.

URL:

/accounts/{accountId}/templates/{templateId}/recipients/{recipientId}/bulk_recipients

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameters are the template ID and recipient ID.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}//accounts/{accountId}/templates/
{templateId}/recipients/{recipientId}/bulk_recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integ
ratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the recipient information for each row of the bulk recipient csv file, the total number of recipients in the file, and the bulkRecipientUri.

The following example shows the response json body.

Example Response Body

```
{
  "bulkRecipients": [
    {
      "rowNumber": "string",
      "email": "string",
      "name": "string",
      "note": "string",
      "accessCode": "string",
      "identification": "string",
      "phoneNumber": "string",
      "tabLabels": [
        {
```

```
          "name": "string,"
          "value": "string"
        }
      ]
    },
    {
      "rowNumber": "string",
      "email": "string",
      "name": "string",
      "note": "string",
      "accessCode": "string",
      "identification": "string",
      "phoneNumber": "string",
      "tabLabels": [
        {
          "name": "string,
          "value": "string"
        }
      ]
    }
  ],
  "bulkRecipientsCount": "string",
  "bulkRecipientsUri": "string"
}
```

Delete Template Bulk Recipient File

This removes the bulk recipient file from a template.

After using this, the bulkRecipientsUri field will not be returned in subsequent GET calls for the template, but the recipient will remain as a bulk recipient

URL:

/accounts/{accountId}/templates/{templateId}/recipients/{recipientId}/bulk_recipients

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

The only required parameters are the template ID and recipient ID.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}//accounts/{accountId}/templates/
{templateId}/recipients/{recipientId}/bulk_recipients

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integ
ratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
```

```
Content-Type: application/json
```

Response

The response returns a success or failure.

Get Tab Information for a Template Recipient

This retrieves information about the tabs associated with a recipient in a template.

URL:

/accounts/{accountId}/templates/{templateId}/recipients/{recipientId}/tabs

Optional query item: include_anchor_tab_locations={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query parameters can be added.

Name	Reqd?	Type	Description
include_anchor_tab_locations	No	String	When true all tabs with anchor tab properties are included in the response.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/templates/
{templateId}/recipients/{recipientId}/tabs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of tabs associated with the recipient. The tab information includes a tabId that can be used when deleting a tab.

The following example shows the response json body.

Example Response Body

```
{
    "approveTabs":[{
           <Tab information removed>
    }],
    "titleTabs":[{
           <Tab information removed>
    }],
    "signHereTabs":[{
           <Tab information removed>
    }]
}
```

Get List of Unsupported File Types

Retrieves a list of file types (mime-types and file-extensions) that are not supported for upload through the DocuSign system.

URL:

/accounts/{accountId}/unsupported_file_types

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/unsupported_file_types

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of the file extensions and mime types that are not supported.
The following example shows the response json body.

Example Response Body

```
{
    "fileTypes":[
           {
           "fileExtension":"String content",
```

```
            "mimeType":"String content"
            },
            {
            "fileExtension":"String content",
            "mimeType":"String content"
            }
    ]
}
```

Add User to Account

This adds new users to your account.

URL:

/accounts/{accountId}/users

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
activationAccessCode	No	String	The activation code the new user must enter when activating their account.
email	Yes	String	The user's email address for the associated account. This can be a maximum of 100 characters.
enableConnectForUser	No	String	Sets if the user is enabled for updates from DocuSign Connect. This is a true/false setting.
firstName	No	String	The user's first name. This can be a maximum of 50 characters.
forgottenPasswordInfo	No		A complex element that has up to four Question/Answer pairs for forgotten password information.

Name	Reqd?	Type	Description
groupList	No	String	A list of the group information for groups to add the user to. Group information can be found by using GET group information. Only the groupId is required. The parameters are: • groupId – The DocuSign group ID for the group. • groupName – The name of the group. • permissionProfileId – The ID of the permission profile associated with the group. • groupType – The group type.
lastName	No	String	The user's last name. This can be a maximum of 50 characters.
middleName	No	String	The user's middle name This can be a maximum of 50 characters.
password	No	String	The user's password for the associated account. This can be a maximum of 50 characters.
sendActivationOnInvalidLogin	No	String	Sets if another activation email is sent to the user if the fail a log on before activating their account. This is a true/false setting.
suffixName	No	String	The suffix for the user's name. This can be a maximum of 50 characters.
title	No	String	The user's title. This can be a maximum of 10 characters.
userName	Yes	String	The full user name associated with the account. This can be a maximum of 100 characters.
userSettings	No		The name/value pair information for user settings. These determine the actions that a user can take in the account. The userSettings are listed and described below.

userSettings:

The userSettings determine the actions users can take in the account.

Setting Name	Value	Authorization Required	Description
allowBulkRecipients	Boolean	Admin	When true, this user can use the bulk send functionality.
allowRecipientLanguageSelection	Boolean	Admin	When true, this provides the sender with the option to set the language used in the standard email format for a recipient when creating an envelope.
allowSendOnBehalfOf	Boolean	Admin	When true, this user can send envelopes 'on behalf of' other users through the API.
apiAccountWideAccess	Boolean	Admin	When true, this user can send and manage envelopes for the entire account using the DocuSign API.
canEditSharedAddressBook	String	Admin	This element sets the address book usage and management rights for the user. Enumeration values are: None, UseOnlyShared, UsePrivateAndShared, Share.
canManageAccount	Boolean	Admin & not setting for self	When true, this user can manage account settings, manage user settings, add users, and remove users.
canManageTemplates	String	Admin & not setting for self	This element sets the template usage and management rights for the user. Enumeration values are: None, Use, Create, Share.
canSendAPIRequests	Boolean	Admin & Account:UsesAPI is set	Only needed if Integrator Key is not used. When true, this user can send and manage envelopes using the DocuSign API.
canSendEnvelope	Boolean	Admin & not setting for self	When true, this user can send envelopes though the DocuSign Console.

Setting Name	Value	Authorization Required	Description
enableDSPro	Boolean	SysAdmin	When true, this user can send and manage envelopes from the DocuSign Desktop Client.
enableSequentialSigningAPI	Boolean	SysAdmin	When true, this user can define the routing order of recipients for envelopes sent using the DocuSign API.
enableSequentialSigningUI	Boolean	SysAdmin	When true, this user can define the routing order of recipients while sending documents for signature.
enableSignerAttachments	Boolean	Admin	When true, this user can add requests for attachments from signers while sending documents.
enableSignOnPaperOverride	Boolean	Admin	When true, this user can override the account setting that determines if signers may sign their documents on paper as an option to signing electronically.
enableTransactionPoint	Boolean	SysAdmin	When true, this user can select an envelope from their member console and upload the envelope documents to TransactionPoint.
enableVaulting	Boolean	Admin	When true, this user can use electronic vaulting for documents.

Setting Name	Value	Authorization Required	Description
locale	String	Admin	This sets the default language for the user. The supported languages, with the language value shown in parenthesis are: Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Dutch (nl), English US (en), French (fr), German (de), Italian (it), Japanese (ja), Korean (ko), Portuguese (pt), Portuguese (Brazil) (pt_BR), Russian (ru), Spanish - (es).
powerFormAdmin	Boolean	Admin	When true, this user can create, manage and download the PowerForms documents.
powerFormUser	Boolean	Admin	When true, this user can view and download PowerForms documents.
selfSignedRecipientEmailDocument	String	Admin	This sets the user setting for how self-signed documents are presented to the email recipients. This will override the account setting. This can only be changed if the selfSignedRecipientEmailDocumentUserOverrideaccountSetting is true. Enumeration values are: • include_pdf: With this setting a PDF of the completed document is attached to the email • include_link: With this setting a secure link to the self-signed documents is included in the email.

Setting Name	Value	Authorization Required	Description
vaultingMode	String	Admin	This element sets the electronic vaulting mode for the user. Enumeration values are: None, eStored and electronicOriginal.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/users

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "newUsers":[{
           "activationAccessCode":"String content",
           "email":"String content",
           "enableConnectForUser":"String content",
           "firstName":"String content",
           "forgottenPasswordInfo":{
                  "forgottenPasswordAnswer1":"String content",
                  "forgottenPasswordAnswer2":"String content",
                  "forgottenPasswordAnswer3":"String content",
                  "forgottenPasswordAnswer4":"String content",
                  "forgottenPasswordQuestion1":"String content",
                  "forgottenPasswordQuestion2":"String content",
                  "forgottenPasswordQuestion3":"String content",
                  "forgottenPasswordQuestion4":"String content"
           },
           "groupList": [
                  {
                         "groupId": "string content",
                         "groupName": "string content",
                         "permissionProfileId": "string content",
                         "groupType": "string content"
                  },
                  {
                         "groupId": "string content",
                         "groupName": "string content",
                         "permissionProfileId": "string content",
                         "groupType": "string content"
                  }
           ],
           "lastName":"String content",
           "middleName":"String content",
           "password":"String content",
           "sendActivationOnInvalidLogin":"String content",
           "suffixName":"String content",
           "title":"String content",
           "userName":"String content",
           "userSettings":[{
                  "name":"String content",
                  "value":"String content"
           }]
```

```
    }]
}
```

Response

The response returns the new user's information.

The following example shows the response json body.

Example Response Body

```
{
    "newUsers":[
            {
                    "userId":"String content",
                    "uri":"String content",
                    "apiPassword":"String content",
                    "email":"String content",
                    "userName":"String content",
                    "errorDetails":{
                            "errorCode":"String content",
                            "message":"String content"
                    }
            }
    ]
}
```

Get User List

This returns a list of users for the specified account.

URL:

/accounts/{accountId}/users

Optional query additions: additional_info={true/false}, start_position={integer}, count={integer}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query can be added to provide additional information in the response.

Name	Reqd?	Type	Description
additional_info	No	Boolean	When true, the full list of user information is returned for each user in the account.
start_position	No	Integer	Starting value for the list.

count	No	Integer	Number of records to return. The number must be greater than 1 and less than or equal to 100.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of users for the account along with the information about the result set. If the additional_info query was added to the URL and set to true, the full user information is returned for each user.

Name	Description
resultSetSize	Total number users returned in the result set.
totalSetSize	Total number users in the account.
startPosition	Starting position of the current result set.
endPosition	The last position in the result set.
nextUri	Provides the Uri to the next chunk of records based on the request. If the endPosition is the entire results of the search, this is null.
previousUri	Provides the Uri to the previous chunk of records based on the request. If this response is the first response for the search, this is null.

The following example shows the response json body.

Example Response Body

```
{
    "users": [
            {
                    "userName": "String content",
                    "userId": "String content",
                    "userType": "String content",
                    "isAdmin": "String content",
                    "userStatus": "String content",
                    "uri": "String content",
                    "email": "String content",
                    "password": "String content",
                    "title": "String content",
                    "createdDateTime": "String content",
                    "firstName": "String content",
                    "middleName": "String content",
                    "lastName": "String content",
                    "suffixName": "String content",
```

```
                    "userSettings": [
                           {
                                  "name": "String content",
                                  "value": "String content"
                           }
                    ],
                    "sendActivationOnInvalidLogin": "String content",
                    "activationAccessCode": "String content",
                    "enableConnectForUser": "String content",
                    "forgottenPasswordInfo": {
                           "forgottenPasswordQuestion1": "String content",
                           "forgottenPasswordAnswer1": "String content",
                           "forgottenPasswordQuestion2": "String content",
                           "forgottenPasswordAnswer2": "String content",
                           "forgottenPasswordQuestion3": "String content",
                           "forgottenPasswordAnswer3": "String content",
                           "forgottenPasswordQuestion4": "String content",
                           "forgottenPasswordAnswer4": "String content"
                    },
                    "groupList": [
                           {
                                  "groupId": "String content",
                                  "groupName": "String content",
                                  "permissionProfileId": "String content",
                                  "groupType": "String content"
                           }
                    ],
                    "workAddress": {
                           "address1": "String content",
                           "address2": "String content",
                           "city": "String content",
                           "stateOrProvince": "String content",
                           "postalCode": "String content",
                           "phone": "String content",
                           "fax": "String content",
                           "country": "String content"
                    },
                    "homeAddress": {},
                    "loginStatus": "String content",
                    "passwordExpiration": "String content",
                    "lastLogin": "String content"
                    "customSettings": [
                           {}
                    ],
                    "profileImageUri": "String content",
                    "userProfileLastModifiedDate": " String content",
                    "signatureImageUri": " String content",
                    "initialsImageUri": " String content"

             }
      ],
      "resultSetSize": "String Content",
      "totalSetSize": "String Content",
      "startPosition": "String Content",
      "endPosition": "String Content",
      "nextUri": "String Content",
      "previousUri": "String Content"
}
```

Close a User

This closes one or more user records in the account. Users are never deleted from an account, but closing a user prevents them from using account functions.

URL:

/accounts/{accounted}/users

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
userId	Yes	String	The user ID number for user being removed from the account.
userName	No	String	The user name for the user being removed from the account.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accounted}/users

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "users":[{
           "userId":"String content",
           "userName":"String content"
    }]
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed. The response contains a user structure similar to the request and includes the user changes. If an error occurred during the DELETE operation for any of the users, that user will contain an errorDetails node with an errorCode and message.

The following example shows the response json body.

Example Response Body

```
{
    "users":[{
           "errorDetails":{
                  "errorCode":"String content",
                  "message":"String content"
           },
           "uri":"String content",
           "userId":"String content",
           "userName":"String content",
           "userStatus":"String content",
           "userType":"String content"
    }]
 }
```

Get User Information

This retrieves the user information for specified user.

URL:

/accounts/{accountId}/users/{userId}

Optional query string: additional_info={true/false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query can be added to provide additional information in the response.

Name	Reqd?	Type	Description
additional_info	No	Boolean	When true, the full list of user information is returned for the user. This includes when a user last logged on to the system, when the user's password will expire (if the account has set expiration times), and the user's current password status in the response.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
```

```
Content-Type: application/json
```

Response

The response returns the user information.

If the query string additional_info=true is included, the following user information is included in the response.

Name	Type	Description
passwordExipration	String	Shows the date-time when the user's password will expire. If the account Password Strength Settings – Expiration information is not set, this value is not returned.
lastLogin	String	Shows the date-time when the user last logged on to the system.
loginStatus	String	Shows the current status of the user's password. Possible values are: • password_reset, • password_active, • password_expired, • password_locked, • password_reset_failed,

The following example shows the response json body with the query string additional_info=true.

Example Response Body

```
{
    "userName": "String content",
    "userId": "String content",
    "userType": "String content",
    "isAdmin": "String content",
    "userStatus": "String content",
    "uri": "String content",
    "email": "String content",
    "password": "String content",
    "title": "String content",
    "createdDateTime": "String content",
    "firstName": "String content",
    "middleName": "String content",
    "lastName": "String content",
    "suffixName": "String content",
    "userSettings": [
           {
                  "name": "String content",
                  "value": "String content"
           }
    ],
    "sendActivationOnInvalidLogin": "String content",
    "activationAccessCode": "String content",
    "enableConnectForUser": "String content",
    "forgottenPasswordInfo": {
```

```
            "forgottenPasswordQuestion1": "String content",
            "forgottenPasswordAnswer1": "String content",
            "forgottenPasswordQuestion2": "String content",
            "forgottenPasswordAnswer2": "String content",
            "forgottenPasswordQuestion3": "String content",
            "forgottenPasswordAnswer3": "String content",
            "forgottenPasswordQuestion4": "String content",
            "forgottenPasswordAnswer4": "String content"
    },
    "groupList": [
            {
                    "groupId": "String content",
                    "groupName": "String content",
                    "permissionProfileId": "String content",
                    "groupType": "String content",
            }
    ],
    "workAddress": {
            "address1": "String content",
            "address2": "String content",
            "city": "String content",
            "stateOrProvince": "String content",
            "postalCode": "String content",
            "phone": "String content",
            "fax": "String content",
            "country": "String content"
    },
    "homeAddress": {},
    "loginStatus": "String content",
    "passwordExpiration": "String content",
    "lastLogin": "String content",
    "customSettings": [
            {}
    ]
    "profileImageUri": "String content",
    "userProfileLastModifiedDate": " String content",
    "signatureImageUri": " String content",
    "initialsImageUri": " String content"
}
```

Get Cloud Storage Provider Configurations for a User

This retrieves the list of cloud storage providers enabled for the account and the configuration information for the user.

URL:

/accounts/{accountId}/users/{userId}/cloud_storage

Optional query parameter: redirectUrl={someURL}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query parameter can be added to the request.

Name	Reqd?	Type	Description
redirectUrl	No	String	The URL the user is redirected to after the cloud storage provider authenticates the user. Using this will append the redirectUrl to the authenticationUrl. The redirectUrl is restricted to URLs in the docusign.com or docusign.net domains.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountid}/users/{userid}
/cloud_storage

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response provides a list of the cloud storage providers for the user and shows if the user has passed authentication with provider.

Value	Description
serviceId	The DocuSign generated ID for the cloud storage provider. This information is only included in the response if the user has passed authentication for the cloud storage provider.
service	The service name for the cloud storage provider.
authenticationUrl	The authentication URL used for the cloud storage provider. This information is only included in the response if the user has not passed authentication for the cloud storage provider. If the redirectUrl query string is provided, the returnUrl is appended to the authenticationUrl.

The following example shows the response json body.

Example Response Body

```
{
    "storageProviders": [
            {
```

```
                "serviceId": "4136",
                "service": "DropBox"
            },
            {
                "service": "Box",
                "authenticationUrl": "BoxAuthenticationUrl"
            }
    ]
}
```

Add Cloud Storage Provider Return Information for a User

This configures the redirect URL information added to the authentication URL for one or more cloud storage providers for a user.

URL:

/accounts/{accountId}/users/{userId}/cloud_storage

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
service	Yes	String	The service name for the cloud storage provider.
redirectUrl	Yes	String	The URL the user is redirected to after the cloud storage provider authenticates the user. This value is appended to the authenticationUrl for the cloud storage provider. The redirectUrl is restricted to URLs in the docusign.com or docusign.net domains.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountid}/users/{userid}
/cloud_storage

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "storageProviders": [
```

```
            {
                    "service": "string",
                    "redirectUrl": "string"
            },
            {
                    "service": "string",
                    "redirectUrl": "string"
            }
    ]
}
```

Response

The response returns a success or failure, along with the serviceId and authenticationUrl for the cloud storage providers.

The following example shows the response json body.

Example Response Body

```
{
    "storageProviders": [
            {
                    "serviceId": "4136",
                    "service": "DropBox"
                    "authenticationUrl": "DropBoxAuthenticationUrl"
            },
            {
                    "serviceId": "4137",
                    "service": "Box",
                    "authenticationUrl": "BoxAuthenticationUrl"
            }
    ]
}
```

Delete Cloud Storage Provider Authentications for a User

This removes the user authentication information for one or more cloud storage providers. The next time the user tries to access the cloud storage provider, they must pass normal authentication.

URL:

/accounts/{accountId}/users/{userId}/cloud_storage

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Either service or serviceId must be included. Using “all” in service or serviceId deletes the user authentication information for all cloud storage providers.

Name	Reqd?	Type	Description
serviceId	Yes*	String	The DocuSign generated ID for the cloud storage provider.
service	Yes*	String	The service name for the cloud storage provider.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountid}/users/{userid}
/cloud_storage

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "storageProviders": [
            {
                    "serviceId": "4136"
            },
            {
                    "service": "Box"
            }
    ]
}
```

Response

The response returns the list of cloud storage providers for which authentication has been removed.

The following example shows the response json body.

Example Response Body

```
{
    "storageProviders": [
            {
                    "serviceId": "4136",
                    "service": "DropBox"
            },
            {
                    "serviceId": "4137",
                    "service": "Box",
            }
    ]
}
```

Get One Cloud Storage Provider Configuration for a User

This retrieves the configuration information for one cloud storage provider for the user.

URL:

/accounts/{accountId}/users/{userId}/cloud_storage/{serviceId}

Optional query parameter: redirectUrl={someURL}

Note: The {serviceId} can be either the service name or serviceId.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters. The following optional query parameter can be added to the request.

Name	Reqd?	Type	Description
redirectUrl	No	String	The URL the user is redirected to after the cloud storage provider authenticates the user. Using this will append the redirectUrl to the authenticationUrl. The redirectUrl is restricted to URLs in the docusign.com or docusign.net domains.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountid}/users/{userid}
/cloud_storage/{serviceId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the cloud storage provider information for the user and shows if the user has passed authentication with provider.

Value	Description
serviceId	The DocuSign generated ID for the cloud storage provider. This information is only included in the response if the user has passed authentication for the cloud storage provider.
service	The service name for the cloud storage provider.

Value	Description
authenticationUrl	The authentication URL used for the cloud storage provider. This information is only included in the response if the user has not passed authentication for the cloud storage provider. If the redirectUrl query string is provided, the returnUrl is appended to the authenticationUrl.

The following example shows the response json body.

Example Response Body

```
{
    "storageProviders": [
            {
                    "serviceId": "4136",
                    "service": "DropBox"
            }
    ]
}
```

Delete One Cloud Storage Provider Authentication for a User

This removes the user authentication information for one cloud storage provider. The next time the user tries to access the cloud storage provider, they must pass normal authentication for this cloud storage provider.

URL:

/accounts/{accountId}/users/{userId}/cloud_storage/{serviceId}

Note: The {serviceId} can be either the service name or serviceId.

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Using “all” in service or serviceId deletes the user authentication information for all cloud storage providers.

Name	Reqd?	Type	Description
service	Yes*	String	The DocuSign generated ID for the cloud storage provider.
serviceId	Yes*	String	The service name for the cloud storage provider.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountid}/users/{userid}
/cloud_storage/{serviceId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "storageProviders": [
           {
                  "serviceId": "4136"
           }
    ]
}
```

Response

The response returns the cloud storage provider for which authentication has been removed.

The following example shows the response json body.

Example Response Body

```
{
    "storageProviders": [
           {
                  "serviceId": "4136",
                  "service": "DropBox"
           }
    ]
}
```

Get User Items in a Cloud Storage Provider

This retrieves a list of all items in all folders for the user from the selected cloud storage provider. There is an option to limit the returned items by providing a comma separated list of folder IDs in the request.

URL:

/accounts/{accountId}/users/{userId}/cloud_storage/{serviceId}/folders

Optional query parameters: cloud_storage_folder_path={comma separated list of folder IDs}, start_position={integer}, count={integer}, order={asc/desc}, order_by={modified/name}

Note: The {serviceId} can be either the service name or serviceId.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query parameter can be added to the request.

Name	Reqd?	Type	Description
cloud_storage_folder_path	No	String	A comma separated list of folder IDs included in the request.
start_position	No	String	An optional value that indicates the starting point of the first item included in the response set. It uses a 0-based index. The default setting for this is 0.
count	No	String	An optional value that sets how many items are included in the response. The default setting for this is 25.
order	No	String	An optional value that sets the direction order used to sort the item list. Valid values are: • asc = ascending sort order • desc = descending sort order
order_by	No	String	An optional value that sets the file attribute used to sort the item list. Valid values are: • modified • name

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountid}/users/{userid}
/cloud_storage/{serviceid}/folders

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of all items in all folders with the selected cloud storage provider.

Value	Description
name	The name of the searched location.
id	The cloud storage provider ID for the searched location.

Value	Description
totalSetSize	The total number of items in the searched location.
resultSetSize	The number of items returned in the current request.
startPosition	The starting position of the current result set.
endPosition	The end position of the current result set.
Items	This is a list of the items in the searched location. The list can include the following information: Note: The information presented depends on the item type. Additionally, some cloud storage providers do not provide all the information. • name: The name of the item. • id: The cloud storage provider ID for the item. This is the information used as the remoteUrl when trying to add the file as a document to an envelope. • date: The date the item was last modified. • type: Shows if the item is a file or a folder. • uri: The uri for the item. • size: The size (in bytes) of the item. • supported: A true or false value. When false the file type is included in the list of unsupported file types and cannot be included in a DocuSign envelope. When true, the file type is not included in the list of unsupported file types and might be able to be included in a DocuSign envelope.

The following example shows the response json body.

Example Response Body

```
{
    "name": "All Files and Folders",
    "id": "allfilesandfolders",
    "totalSetSize": "3",
    "resultSetSize": "3",
    "startPosition": "0",
    "endPosition": "2",
    "items": [
            {
                    "name": "nda.pdf",
                    "id": "1911:26490ea3-f725-4cb1-8e86-8275d93a1172:nda.pdf",
                    "date": "2014-04-15T07:00:00.0000000Z",
                    "type": "file",
                    "size": "243300",
                    "supported": "true"
            },
            {
                    "name": "My Folder",
                    "id": "TXklMjBGb2xkZXI",
                    "type": "folder",
                    "uri": "/cloud_storage/194/folders/TXklMjBGb2xkZXI"
            },
```

```
            {
                    "name": "parking.pdf",
                    "id": "1911:26490ea3-f725-4cb1-8e86-8275d93a1712:parking.pdf",
                    "date": "2014-03-12T08:32:00.0000000Z",
                    "type": "file",
                    "size": "233400",
                    "supported": "true"
            }
    ]
}
```

Get User Items in one Cloud Storage Provider Folder

This retrieves a list of all items in one folder for the user from the selected cloud storage provider.

URL:

/accounts/{accountId}/users/{userId}/cloud_storage/{serviceId}/folders/{folderId}

Optional query parameters: start_position={integer}, count={integer}, order={asc/desc}, order_by={modified/name}

Note: The {serviceId} can be either the service name or serviceId. The {folderId} can be a folder ID or “allfilesandfolders” to retrieve all items.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

There are no required parameters, but the following optional query parameter can be added to the request.

Name	Reqd?	Type	Description
start_position	No	String	An optional value that indicates the starting point of the first item included in the response set. It uses a 0-based index. The default setting for this is 0.
count	No	String	An optional value that sets how many items are included in the response. The default setting for this is 25.
order	No	String	An optional value that sets the direction order used to sort the item list. Valid values are: • asc = ascending sort order • desc = descending sort order

Name	Reqd?	Type	Description
order_by	No	String	An optional value that sets the file attribute used to sort the item list. Valid values are: • modified • name

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountid}/users/{userid}
/cloud_storage/{serviceid}/folders/{folderId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a list of all items in all folders with the selected cloud storage provider.

Value	Description
name	The name of the searched location.
id	The cloud storage provider ID for the searched location.
totalSetSize	The total number of items in the searched location.
resultSetSize	The number of items returned in the current request.
startPosition	The starting position of the current result set.
endPosition	The end position of the current result set.

Value	Description
Items	This is a list of the items in the searched location. The list can include the following information: Note: The information presented depends on the item type. Additionally, some cloud storage providers do not provide all the information. • name: The name of the item. • id: The cloud storage provider ID for the item. This is the information used as the remoteUrl when trying to add the file as a document to an envelope. • date: The date the item was last modified. • type: Shows if the item is a file or a folder. • uri: The uri for the item. • size: The size (in bytes) of the item. • supported: A true or false value. When false the file type is included in the list of unsupported file types and cannot be included in a DocuSign envelope. When true, the file type is not included in the list of unsupported file types and might be able to be included in a DocuSign envelope.

The following example shows the response json body.

Example Response Body

```
{
    "name": "My Folder",
    "id": "myfolder",
    "totalSetSize": "1",
    "resultSetSize": "1",
    "startPosition": "0",
    "endPosition": "0",
    "items": [
            {
                    "name": "parking.pdf",
                    "id": "1911:26490ea3-f725-4cb1-8e86-8275d93a1712:parking.pdf",
                    "date": "2014-03-12T08:32:00.0000000Z",
                    "type": "file",
                    "size": "233400",
                    "supported": "true"
            }
    ]
}
```

Get Custom User Settings

This retrieves a list of custom user settings for a single user.

Custom settings are a flexible way to store and retrieve custom user information that can be used in your own system.

Note: Custom user settings are not the same as user account settings.

Getting Grouped Custom User Settings

If the custom user settings you want to retrieve are grouped, you must include the following information in the header, after Content-Type, in the request:

`X-DocuSign-User-Settings-Key:`*`group_name`*

Where the *`group_name`* is your designated name for the group of customer user settings.

If the extra header information is not included, only the custom user settings that were added without a group are retrieved.

URL:

/accounts/{accountId}/users/{userId}/custom_settings

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
custom_settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of the custom name/value pairs for the specified user that are not in a group.

The following example shows the response json body.

Example Response Body

```
{
    "customSettings":[{
           "name":"String content",
           "value":"String content",
    },
    {
           "name":"String content",
           "value":"String content"
    }]
}
```

Add or Modify Custom User Settings

This allows you to add or update custom user settings for a single user.

Note: Custom user settings are not the same as user account settings.

Custom settings are a flexible way to store and retrieve custom user information that can be used in your own system.

Important: There is a limit on the size for all the custom user settings for a single user. All the custom user settings for a single user is limited to 4,000 characters, which includes the xml and json structure for the settings.

Grouping Custom User Settings

In addition to adding the custom user settings, you can group the settings when adding them. Grouping allows you to retrieve only those settings that are in a group, instead of retrieving all the user custom settings.

To group custom user settings, add the following information in the header, after Content-Type:

`X-DocuSign-User-Settings-Key:`*`group_name`*

Where the *`group_name`* is your designated name for the group of customer user settings. Grouping is shown in the Example Request Body below.

When getting or deleting grouped custom user settings, the extra header information must be included.

Grouping custom user settings is not required and if the extra header information is not included, the custom user settings are added normally and can be retrieved or deleted without including the additional header.

URL:

/accounts/{accountId}/users/{userId}/custom_settings

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
customSettings	Yes	String	The name/value pair information for the user custom setting.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
custom_settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

X-DocuSign-User-Settings-Key:group_name

{
    "customSettings":[{
           "name":"String content",
           "value":"String content",
    },
    {
           "name":"String content",
           "value":"String content"
    }]
}
```

Response

The response returns a success or failure message and a list of the added or updated user custom settings.

The following example shows the response json body.

Example Response Body

```
{
    "customSettings":[{
           "name":"String content",
           "value":"String content",
    },
    {
           "name":"String content",
           "value":"String content"
    }]
}
```

Delete Custom User Settings

This deletes the specified custom user settings for a single user.

Deleting Grouped Custom User Settings

If the custom user settings you want to delete are grouped, you must include the following information in the header, after Content-Type, in the request:

`X-DocuSign-User-Settings-Key:`*`group_name`*

Where the *`group_name`* is your designated name for the group of customer user settings.

If the extra header information is not included, only the custom user settings that were added without a group are deleted.

URL:

/accounts/{accountId}/users/{userId}/custom_settings

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
customSettings	Yes	String	The name/value pair information that should be deleted.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
custom_settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "customSettings":[{
           "name":"String content",
           "value":"String content",
    },
    {
           "name":"String content",
           "value":"String content"
    }]
}
```

Response

The response returns a success or failure and the list of the custom name/value pairs that were deleted.

The following example shows the response json body.

Example Response Body

```
{
    "customSettings":[{
           "name":"String content",
           "value":"String content",
    },
    {
           "name":"String content",
           "value":"String content"
    }]
}
```

Get User Profile

This returns user profile information, the privacy settings and personal information (address, phone number, etc.).

URL:

/accounts/{accountId}/users/{userId}/profile

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
profile

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the user profile information from the user's ID card. The User Profile contains the following information.

Name	Description
address	The user's address information. A complex element consisting of the optional elements: address1, address2, city, country, fax, phone, postalCode, stateOrProvince.
authenticationMethods	Shows the authentication methods used by the user.
companyName	The user's Company information.
displayOrganizationInfo	When true, the user's company and title information are shown on the ID card.
displayPersonalinfo	When true, the user's Address and Phone number are shown on the ID card.

displayProfile	When true, the user's ID card can be viewed from signed documents and envelope history.
displayUsageHistory	When true, the user's usage information is shown on the ID card.
title	The user's Title information.
usageHistory	A complex element consisting of: • lastSentDateTime – the date and time the user last sent an envelope. • lastSignedDateTime – the date and time the user last signed an envelope. • sentCount – the number of envelopes the user has sent. • signedCount – the number of enevelopes the user has signed.
userDetails	A complex element with the user name information consisting of: • firstName – The user's first name. • lastName – The user's last name. • middleName – The user's middle name. • suffixName – The suffix for the user's name. • title – The user's title.

The following example shows the response json body.

Example Response Body

```
{
    "address":{
           "address1":"String content",
           "address2":"String content",
           "city":"String content",
           "country":"String content",
           "fax":"String content",
           "phone":"String content",
           "postalCode":"String content",
           "stateOrProvince":"String content"
    },
    "authenticationMethods":[{
           "authenticationType":"String content",
           "lastProvider":"String content",
           "lastTimestamp":"String content",
           "totalCount":2147483647
    }],
    "companyName":"String content",
    "displayOrganizationInfo":"String content",
    "displayPersonalInfo":"String content",
    "displayProfile":"String content",
    "displayUsageHistory":"String content",
    "title":"String content",
    "usageHistory":{
           "lastSentDateTime":"String content",
           "lastSignedDateTime":"String content",
           "sentCount":2147483647,
           "signedCount":2147483647
```

```
    }
    "userDetails":{
            "firstName":"String content",
            "lastName":"String content",
            "middleName":"String content",
            "suffixName":"String content",
            "title":"String content",
    }
}
```

Modify User Profile

This sets the user's detail information, profile information, privacy settings and personal information in the user ID card.

This can also be used to change a user's name by changing the information in the user details (userDetails). When changing a user's name, you can either change the information in the userName OR change the information in firstName, middleName, lastName, suffixName and title. Changes to firstName, middleName, lastName, suffixName and title take precedence over changes to the userName.

URL:

/accounts/{accountId}/users/{userId}/profile

Note: The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
address	No	address	The user's address information. A complex element consisting of the optional elements: address1, address2, city, country, fax, phone, postalCode, stateOrProvince.

Name	Reqd?	Type	Description
authenticationMethods	No	autneticationMethod	These properties cannot be modified in the PUT. Shows the authentication methods used by the user. A complex element consisting of: • authenticationMethodType (PhoneAuth, STAN, ISCheck, OFAC, AccessCode, AgeVerify, or SSOAuth) • lastTimestamp – the data and time the user last used the authentication method. • lastProvider – the last provider that authenticated the user. • totalCount – the number of times used.
companyName	No	String	The user's Company information.
displayOrganizationInfo	No	String	True/False setting. When true, the user's company and title information are shown on the ID card.
displayPersonalinfo	No	String	True/False setting. When true, the user's Address and Phone number are shown on the ID card.
displayProfile	No	String	True/False setting. When true, the user's ID card can be viewed from signed documents and envelope history.
displayUsageHistory	No	String	True/False setting. When true, the user's usage information is shown on the ID card.
title	No	String	The user's Title information.
usageHistory	No	usageHistory	A complex element consisting of: • lastSentDateTime – the date and time the user last sent an envelope. • lastSignedDateTime – the date and time the user last signed an envelope. • sentCount – the number of envelopes the user has sent. • signedCount – the number of envelopes the user has signed.

Name	Reqd?	Type	Description
userDetails	No	userDetails	A complex element with the user name information consisting of: • firstName – The user's first name. This can be a maximum of 50 characters. • lastName – The user's last name. This can be a maximum of 50 characters. • middleName – The user's middle name. This can be a maximum of 50 characters. • suffixName – The suffix for the user's name. This can be a maximum of 50 characters. • title – The user's title. This can be a maximum of 10 characters. • username – The user's full name. This can be a maximum of 100 characters.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
profile

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "address":{
           "address1":"String content",
           "address2":"String content",
           "city":"String content",
           "country":"String content",
           "fax":"String content",
           "phone":"String content",
           "postalCode":"String content",
           "stateOrProvince":"String content"
    },
    "authenticationMethods":[{
           "authenticationType":"String content",
           "lastProvider":"String content",
           "lastTimestamp":"String content",
           "totalCount":2147483647
    }],
    "companyName":"String content",
    "displayOrganizationInfo":"String content",
    "displayPersonalInfo":"String content",
    "displayProfile":"String content",
    "displayUsageHistory":"String content",
    "title":"String content",
```

```
    "usageHistory":{
           "lastSentDateTime":"String content",
           "lastSignedDateTime":"String content",
           "sentCount":2147483647,
           "signedCount":2147483647
    }
    "userDetails":{
           "firstName":"String content",
           "lastName":"String content",
           "middleName":"String content",
           "suffixName":"String content",
           "title":"String content",
           "userName":"String content"
    }
}
```

Response

The response returns success or failure.

Get User Profile Image

This returns the user profile picture. The image is returned in the same format as uploaded.

URL:

/accounts/{accountId}/users/{userId}/profile/image

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
profile/image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

If successful, the response returns a 200 – OK and the user profile image.

Modify User Profile Image

This uploads an image to the user profile. The supported image formats for this file are: gif, png, jpeg, and bmp. The file must be less than 200K. For best viewing results, DocuSign recommends that the image is no more than 79 pixels wide and high.

URL:

/accounts/{accountId}/users/{userId}/profile/image

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

The only item needed is the image.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
profile/image


X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: image/x-ms-bmp

<image removed>
```

Response

The response returns a success (200 – OK) or failure.

Delete User Profile Image

This removes the image from the user profile.

URL:

/accounts/{accountId}/users/{userId}/profile/image

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
profile/image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns a success (200 – OK) or failure.

Get User Account Settings

This returns the account settings list and email notification information for the specified user.

URL:

/accounts/{accountId}/users/{userId}/settings

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
settings


X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the account setting name/value information and the email notification settings for the specified user. See the userSettings list for more information about the different user settings.

The following example shows the response json body.

Example Response Body

```
{
  "userSettings": [
    {
      "name": "string",
      "value": "string",
    }
  ],
  "signerEmailNotifications": {
    "envelopeActivation": " string ",
    "envelopeComplete": " string",
    "carbonCopyNotification": "string",
    "certifiedDeliveryNotification": "string",
    "envelopeDeclined": "string",
    "envelopeVoided": "string",
    "envelopeCorrected": "string",
    "reassignedSigner": "string",
    "purgeDocuments": "string",
    "faxReceived": "string",
    "documentMarkupActivation": "string",
    "agentNotification": "string"
  },
  "senderEmailNotifications": {
    "envelopeComplete": "string",
    "changedSigner": "string",
    "senderEnvelopeDeclined": "string",
    "withdrawnConsent": "string",
    "recipientViewed": "string",
    "deliveryFailed": "string"
  }
}
```

Modify User Account Settings

This updates the account settings list and email notifications for the specified user.

URL:

/accounts/{accountId}/users/{userId}/settings

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
userSettings		String	The name/value pair information for user settings. These determine the features available for the user. See the userSettings list for more information about the different user settings.
signerEmailNotifications	No		An array of email notifications that sets the email the user receives when they are a signer. When the specific email notification is set to true, the user will receive those types of email notifications from DocuSign. **Note:** The user inherits the default account signer email notification settings when the user is created. The email notifications are: • envelopeActivation • envelopeComplete • carbonCopyNotification • certifiedDeliveryNotification • envelopeDeclined • envelopeVoided • envelopeCorrected • reassignedSigner • purgeDocuments • faxReceived • documentMarkupActivation • agentNotification

Name	Reqd?	Type	Description
senderEmailNotifications	No		An array of email notifications that sets the email the user receives when they are a sender. When the specific email notification is set to true, the user will receive those types of email notifications from DocuSign. **Note:** The user inherits the default account sender email notification settings when the user is created. The email notifications are: • envelopeComplete • changedSigner • senderEnvelopeDeclined • withdrawnConsent • recipientViewed • deliveryFailed

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "userSettings": [
    {
      "name": "string",
      "value": "string",
    }
  ],
  "signerEmailNotifications": {
    "envelopeActivation": " string ",
    "envelopeComplete": " string",
    "carbonCopyNotification": "string",
    "certifiedDeliveryNotification": "string",
    "envelopeDeclined": "string",
    "envelopeVoided": "string",
    "envelopeCorrected": "string",
    "reassignedSigner": "string",
    "purgeDocuments": "string",
    "faxReceived": "string",
    "documentMarkupActivation": "string",
    "agentNotification": "string"
  },
  "senderEmailNotifications": {
    "envelopeComplete": "string",
    "changedSigner": "string",
    "senderEnvelopeDeclined": "string",
```

```
    "withdrawnConsent": "string",
    "recipientViewed": "string",
    "deliveryFailed": "string"
  }
}
```

Response

The response returns success or failure.

Get a List of User Signature Definitions

This returns the signature definitions for the specified user.

URL:

/accounts/{accountId}/users/{userId}/signatures

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.
>
> Since the {signatureName} is a string name of a user that likely includes spaces, you might need to URL encode the signatureName before using the URL.
>
> For example: "Bob Smith" to "Bob%20Smith"

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns a list of signatures for the user and shows which signature is the default signature for the user.

Example Response Body

```
{
    "userSignatures":[{
           "signatureId":"String content",
           "adoptedDateTime":"String content",
           "createdDateTime":"String content",
           "initialsImageUri":"String content",
           "signatureFont":"String content",
           "signatureImageUri":"String content",
           "signatureInitials":"String content",
           "signatureName":"String content",
           "signatureType":"String content",
           "signature150ImageId":"String content",
           "initials150ImageId":"String content",
           "isDefault": "String content"

    }]
    [{
           "signatureId":"String content",
           "adoptedDateTime":"String content",
           "createdDateTime":"String content",
           "initialsImageUri":"String content",
           "signatureFont":"String content",
           "signatureImageUri":"String content",
           "signatureInitials":"String content",
           "signatureName":"String content",
           "signatureType":"String content",
           "signature150ImageId":"String content",
           "initials150ImageId":"String content",
           "isDefault": "String content"
    }]
}
```

Setting User Signature and Initials Images when Creating a Signature

This allows user signature and/or initials images to be set when a signature is created. The rules and processes associated with this are:

- If Content-Type is set to application/json, then default behavior for creating a default signature image, based on the name and a DocuSign font, is used.
- If Content-Type is set to multipart/form-data, then the request must contain a first part with the user signature information, followed by parts that contain the images.

  For each Image part, the Content-Disposition header has a “filename” value that is used to map to the “signatureName” and/or “signatureInitials” in the JSON to the image. For example: Content-Disposition: file; filename="Ron Test20121127083900"

  If no matching image (by filename value) is found then the image is not set. One, both or neither of the signature and initials images can be set with this call.

  The Content-Transfer-Encoding: base64 header, set in the header for the part containing the image, can be set to indicate that the images are formatted as base64 instead of as binary.

- If successful, 200-OK is returned, and a JSON structure containing the signature information is provided, note that the signatureId can change with each API POST, PUT or DELETE since the changes to the signature structure cause the current signature to be closed, and a new signature record to be created.

URL:

/accounts/{accountId}/users/{userId}/signatures

Note: The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
signatureFont	No	String	The font type for the signature, if the signature is not drawn. The supported font types are: "7_DocuSign", "1_DocuSign", "6_DocuSign", "8_DocuSign", "3_DocuSign", "Mistral", "4_DocuSign", "2_DocuSign", "5_DocuSign", "Rage Italic"
signatureInitials	No	String	The initials associated with the signature.
signatureName	Yes	String	The signature's name in the system.

Example Request Body

The first example shows the default behavior and the second example shows a multipart request.

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "userSignatures": [{
           "signatureFont": "7_DocuSign",
           "signatureInitials": "R T20121127083847",
           "signatureName": "Ron Test20121127083847",
    }]
}
```

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures
```

```
X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: multipart/form-data; boundary=AAA

-AAA
Content-Type: application/json
Content-Disposition: form-data

{
    "userSignatures":[{
           "signatureFont":"7_DocuSign",
           "signatureInitials":"R T20121127083900",
           "signatureName":"Ron Test20121127083900"
    ]}
}

-AAA
Content-Type: image/gif
Content-Disposition: file;filename="Ron Test20121127083900"
Content-Transer-Encoding: base64

<base64 image bytes omitted>
-AAA
Content-Type: image/gif
Content-Disposition: file;filename="R T20121127083900"
Content-Transer-Encoding: base64

<base64 image bytes omitted>
-AAA
```

Response

The response returns a success or failure and signature information.

Example Response Body

```
{
    "userSignatures":[{
           "adoptedDateTime":"String content",
           "createdDateTime":"String content",
           "initials150ImageId":"String content",
           "initialsImageUri":"String content",
           "signature150ImageId":"String content",
           "signatureFont":"String content",
           "signatureId":"String content",
           "signatureImageUri":"String content",
           "signatureInitials":"String content",
           "signatureName":"String content",
           "signatureType":"String content"
    }]
}
```

Get User Signature Information

This returns the structure of a single signature with a known signature name.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.
>
> The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns information for the request user signature. The following example shows the response json body.

Example Response Body

```
{
    "signatureId":"String content",
    "adoptedDateTime":"String content",
    "createdDateTime":"String content",
    "initialsImageUri":"String content",
    "signatureFont":"String content",
    "signatureImageUri":"String content",
    "signatureInitials":"String content",
    "signatureName":"String content",
    "signatureType":"String content",
    "signature150ImageId":"String content",
    "initials150ImageId":"String content",
    "isDefault": "String content"
}
```

Modify a User Signature

This creates or modifies the signature font and initials for an existing signature. When creating a signature, you would use this resource to create the signature name and then put the signature and initials images into the signature.

Note: This will also create a default signature for the user when one does not exist.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}

Optional query string: close_existing_signature={true or false} when true, this will close the current signature.

Note: The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
signatureFont	No	String	The font type for the signature, if the signature is not drawn. The supported font types are: "7_DocuSign", "1_DocuSign", "6_DocuSign", "8_DocuSign", "3_DocuSign", "Mistral", "4_DocuSign", "2_DocuSign", "5_DocuSign", "Rage Italic"

Name	Reqd?	Type	Description
signatureId	No	String	The signature ID associated with the signature name. The signatureId can be used in the URI in place of the signatureName, and the signatureName in the body will be used. This allows the use of special characters (such as “&”, “<”, “>”) in a signatureName. Note that with each update to signatures, the returned signautreId might change, so the caller will need to trigger off the signatureName to get the new signatureId.
signatureInitials	No	String	The initials associated with the signature.
signatureName	No	String	The user signature name.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "signatureFont":"String content",
    "signatureId":"String content",
    "signatureInitials":"String content",
    "signatureName":"String content"
 }
```

Response

The response returns a success or failure and signature name information.

Example Response Body

```
{
    "signatureId":"String content",
    "adoptedDateTime":"String content",
    "createdDateTime":"String content",
    "initialsImageUri":"String content",
    "signatureFont":"String content",
    "signatureImageUri":"String content",
    "signatureInitials":"String content",
    "signatureName":"String content",
    "signatureType":"String content",
    "signature150ImageId":"String content",
    "initials150ImageId":"String content",
    "isDefault": "String content"
}
```

Close a User Signature

This removes the signature information for the user.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.
>
> The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns a success or failure.

Get a User Initials Image

This returns a specific user initials image. The image is returned in the same format as uploaded. In the request you can specify if the chrome (the added line and identifier around the initial image) is returned with the image.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image

Note: The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Optional request string: include_chrome={true} to include chrome, the default value is false.

Note: Older envelopes might only have chromed images. If getting the non-chromed image fails, try getting the chromed image.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}/initials_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the initials image file.

Set a User Initials Image

This updates the user initials image. The supported image formats for this file are: gif, png, jpeg, and bmp. The file must be less than 200K.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image

Note: The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL

encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

No additional parameters are required.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}/initials_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

Content-Type: image/gif

<image removed>
```

Response

The response returns a success or failure and information about the user initials and signature.

The following example shows the response json body.

```
{
    "adoptedDateTime":"String content",
    "createdDateTime":"String content",
    "initials150ImageId":"String content",
    "initialsImageUri":"String content",
    "signature150ImageId":"String content",
    "signatureFont":"String content",
    "signatureId":"String content",
    "signatureImageUri":"String content",
    "signatureInitials":"String content",
    "signatureName":"String content",
    "signatureType":"String content"
}
```

Delete a User Initials Image

This deletes a specific user initials image.

Note: The user signature image associated with this user is not deleted.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.
>
> The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}/initials_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns a success or failure and information about the user initials and signature.

The following example shows the response json body.

```
{
    "adoptedDateTime":"String content",
    "createdDateTime":"String content",
    "initials150ImageId":"String content",
    "initialsImageUri":"String content",
    "signature150ImageId":"String content",
    "signatureFont":"String content",
    "signatureId":"String content",
    "signatureImageUri":"String content",
    "signatureInitials":"String content",
    "signatureName":"String content",
    "signatureType":"String content"
}
```

Get a User Signature Image

This returns a specific user signature image. The image is returned in the same format as uploaded. In the request you can specify if the chrome (the added line and identifier around the initial image) is returned with the image.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/signature_image

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.
>
> The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Optional request string: include_chrome={true} to include chrome, the default value is false.

> **Note:** Older envelopes might only have chromed images. If getting the non-chromed image fails, try getting the chromed image.

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}/signature_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the signature image as a GIF file.

Set a User Signature Image

This updates the user signature image. The supported image formats for this file are: gif, png, jpeg, and bmp. The file must be less than 200K.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/signature_image

> **Note:** The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.
>
> The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

No additional parameters are required.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}/initials_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

Content-Type: image/gif
```

Response

The response returns a success or failure and information about the user signature and initials.

The following example shows the response json body.

```
{
    "adoptedDateTime":"String content",
    "createdDateTime":"String content",
    "initials150ImageId":"String content",
    "initialsImageUri":"String content",
    "signature150ImageId":"String content",
    "signatureFont":"String content",
    "signatureId":"String content",
    "signatureImageUri":"String content",
    "signatureInitials":"String content",
    "signatureName":"String content",
    "signatureType":"String content"
}
```

Delete a User Signature Image

This deletes a specific user signature image.

Note: The user initials image associated with this user is not deleted by this.

URL:

/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/signature_image

Note: The userId specified in the uri must match the authenticated user's userId and the user must be a member of the account.

The {signatureIdOrName} accepts signature ID or signature name. DocuSign recommends you use signature ID (signatureId), since some names contain characters that don't properly URL encode. If you use the user name, it is likely that the name includes spaces and you might need to URL encode the name before using the URL. For example: "Bob Smith" to "Bob%20Smith"

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
signatures/{signatureIdOrName}/signature_image

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns a success or failure and information about the user signature and initials.

The following example shows the response json body.

```
{
    "adoptedDateTime":"String content",
    "createdDateTime":"String content",
    "initials150ImageId":"String content",
    "initialsImageUri":"String content",
    "signature150ImageId":"String content",
    "signatureFont":"String content",
    "signatureId":"String content",
    "signatureImageUri":"String content",
    "signatureInitials":"String content",
    "signatureName":"String content",
    "signatureType":"String content"
```

```
}
```

Add a User Social Account

This adds a new social account to a user's account.

URL:

/accounts/{accountId}/users/{userId}/social

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
provider	Yes	String	The name of the social account provider (Google, Facebook, Yahoo, etc.).
email	Yes	String	The user's email address for the social account.
userName	Yes	String	The full user name in the social account.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
social

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "provider":"String content",
    "email":"String content",
    "userName":"String content"
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed.

Get User Social Accounts

This returns a list of social accounts linked to a user's account.

URL:

/accounts/{accountId}/users/{userId}/social

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
social

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of social account information for the user and the userId for the user.

The following example shows the response json body.

Example Response Body

```
{
    "socialAccountInformation":[{
           "email":"String content",
           "provider":"String content",
           "socialId":"String content",
           "userName":"String content"
    },
    {
           "email":"String content",
           "provider":"String content",
           "socialId":"String content",
           "userName":"String content"
    }],
    "userId":"String content"
}
```

Remove a User Social Account

This removes a social account from a user's account.

URL:

/accounts/{accountId}/users/{userId}/social

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

Name	Reqd?	Type	Description
provider	Yes	String	The name of the social account provider (Google, Facebook, Yahoo, etc.).
email	Yes	String	The user's email address for the social account.
userName	Yes	String	The full user name in the social account.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/accounts/{accountId}/users/{userId}/
social

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "provider":"String content",
    "email":"String content",
    "userName":"String content"
}
```

Response

The response returns if the API execution is successful (200 – OK) or failed.

Post Authentication View

This provides a URL to start the authentication view of the DocuSign UI.

URL:

/accounts/{accountId}/views/console

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

No parameters are required, but an optional envelopeId and returnUrl can be included in the request.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/accounts/{accountId}/views/console

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{"envelopeId":"10d60706-5d90-4a83-8035-7079cacfc38a",
"returnUrl":"http://www.google.com"}
```

Response

The response returns the authentication view url.

The following example shows the header followed by the response json body.

```
{
    "url":"String content"
}
```

Get Account Provisioning Information

This returns the account provisioning information.

URL:

/accounts/provisioning

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

This request requires a DocuSign Integrator Key and the DocuSign AppToken information. The AppToken is used to determine the account provisioning information that is returned and is provided by the group provisioning the account.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/accounts/provisioning

X-DocuSign-Authentication:
<DocuSignCredentials><IntegratorKey>{integrator_key}</IntegratorKey>
</DocuSignCredentials>
```

```
Accept: application/json
Content-Type: application/json
X-DocuSign-AppToken: {AppToken}
```

Response

The response returns the account provisioning information.

The following example shows the response json body.

Example Response Body

```
{
    "distributorCode":"String content",
    "distributorPassword":"String content",
    "defaultPlanId":"String content",
    "defaultConnectionId":"String content",
    "planPromotionText":"String content",
    "passwordRuleText":"String content",
    "purchaseOrderOrPromAllowed":"String content"
}
```

Get List of Billing Plans

This returns the billing plans associated with a distributor.

URL:

/billing_plans

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/billing_plans
```

Response

The response returns the billing plan information.

The following example shows the response json body.

Example Response Body

```
{
```

```
    "billingPlans":[{
           "currencyPlanPrices":[{
                  "currencyCode":"String content",
                  "currencySymbol":"String content",
                  "perSeatPrice":"String content",
                  "supportIncidentFee":"String content",
                  "supportPlanFee":"String content"
           }],
           "enableSupport":"String content",
           "includedSeats":"String content",
           "otherDiscountPercent":"String content",
           "paymentCycle":"String content",
           "paymentMethod":"String content",
           "perSeatPrice":"String content",
           "planFeatureSets":[{
                  "currencyFeatureSetPrices":[{
                         "currencyCode":"String content",
                         "currencySymbol":"String content",
                         "envelopeFee":"String content",
                         "fixedFee":"String content",
                         "seatFee":"String content"
                  }],
                  "envelopeFee":"String content",
                  "featureSetId":"String content",
                  "fixedFee":"String content",
                  "isActive":"String content",
                  "isEnabled":"String content",
                  "name":"String content",
                  "seatFee":"String content"
           }],
           "planId":"String content",
           "planName":"String content",
           "seatDiscounts":[{
                  "beginSeatCount":"String content",
                  "discountPercent":"String content",
                  "endSeatCount":"String content"
           }],
           "supportIncidentFee":"String content",
           "supportPlanFee":"String content"
    }]
}
```

Get Billing Plan Details

This returns the billing plan details for the specified billing plan ID.

URL:

/billing_plans/{billingPlanId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/billing_plans/{planId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the Billing plan and Successor plan details. The plan details are described below.

Name	Type	Description
currencyPlanPrices		A complex type that contains the alternate currency values for perSeatPrice, supportIncidentFee, supportPlanFee that are configured for this plan. It also includes the alternate currencyCode, currenceySymbol, and a list of accepted credit card types.
enableSupport	Boolean	If true the plan has support enabled.
includedSeats	Integer	The number of seats included in the plan.
otherDiscountPercent	Decimal	Any other percentage discount for the plan.
paymentCycle	String	The payment cycle associated with the plan. The possible values are: Monthly or Annually.
paymentMethod	String	The payment method used with the plan. The possible values are: CreditCard, PurchaseOrder, Premium, or Freemium.
perSeatPrice	Decimal	The per seat price for the plan.
planFeatureSets		A complex type that sets the feature sets for the account. It contains the following information (all string content): • currencyFeatureSetPrices - This contains the currencyCode and currencySymbol for the alternate currency values for envelopeFee, fixedFee, seatFee that are configured for this plan feature set. • envelopeFee - An incremental envelope cost for plans with envelope overages (when isEnabled=true). • featureSetId - A unique ID for the

Name	Type	Description
		feature set. • fixedFee - A one-time fee associated with the plan (when isEnabled=true). • isActive - Determines if the feature set is actively set as part of the plan. • isEnabled - Determines if the feature set is actively enabled as part of the plan. • name - The name of the feature set. • seatFee - An incremental seat cost for seat-based plans (when isEnabled=true).
planId	String	The plan ID for the account. It uniquely identifies a plan and is used to set plans in other functions.
planName	String	The name of the plan.
seatDiscounts	SeatDiscount	A complex type that return any seat discount information. It contains the information: BeginSeatCount, EndSeatCount and SeatDiscountPercent.
supportIncidentFee	Decimal	The support incident fee charged for each support incident.
supportPlanFee	Decimal	The support plan fee charged for this plan.

The following example shows the response json body.

Example Response Body

```
{
    "billingPlan":{
          "currencyPlanPrices":[{
                "currencyCode":"String content",
                "currencySymbol":"String content",
                "perSeatPrice":"String content",
                "supportIncidentFee":"String content",
                "supportPlanFee":"String content"
                "supportedCardTypes": {
                      "cardTypes": [
                            "string",
                            "string"
                      ]
                }
          }],
          "enableSupport":"String content",
          "includedSeats":"String content",
          "otherDiscountPercent":"String content",
          "paymentCycle":"String content",
          "paymentMethod":"String content",
          "perSeatPrice":"String content",
```

```
            "planFeatureSets":[{
                  "currencyFeatureSetPrices":[{
                        "currencyCode":"String content",
                        "currencySymbol":"String content",
                        "envelopeFee":"String content",
                        "fixedFee":"String content",
                        "seatFee":"String content"
                  }],
                  "envelopeFee":"String content",
                  "featureSetId":"String content",
                  "fixedFee":"String content",
                  "isActive":"String content",
                  "isEnabled":"String content",
                  "name":"String content",
                  "seatFee":"String content"
            }],
            "planId":"String content",
            "planName":"String content",
            "seatDiscounts":[{
                  "beginSeatCount":"String content",
                  "discountPercent":"String content",
                  "endSeatCount":"String content"
            }],
            "supportIncidentFee":"String content",
            "supportPlanFee":"String content"
      },
      "successorPlans":[{
            "currencyPlanPrices":[{
                  "currencyCode":"String content",
                  "currencySymbol":"String content",
                  "perSeatPrice":"String content",
                  "supportIncidentFee":"String content",
                  "supportPlanFee":"String content"
                  "supportedCardTypes": {
                        "cardTypes": [
                              "string",
                              "string"
                        ]
                  }
            }],
            "enableSupport":"String content",
            "includedSeats":"String content",
            "otherDiscountPercent":"String content",
            "paymentCycle":"String content",
            "paymentMethod":"String content",
            "perSeatPrice":"String content",
            "planFeatureSets":[{
                  "currencyFeatureSetPrices":[{
                        "currencyCode":"String content",
                        "currencySymbol":"String content",
                        "envelopeFee":"String content",
                        "fixedFee":"String content",
                        "seatFee":"String content"
                  }],
                  "envelopeFee":"String content",
                  "featureSetId":"String content",
                  "fixedFee":"String content",
                  "isActive":"String content",
                  "isEnabled":"String content",
                  "name":"String content",
                  "seatFee":"String content"
            }],
            "planId":"String content",
            "planName":"String content",
```

```
            "seatDiscounts":[{
                   "beginSeatCount":"String content",
                   "discountPercent":"String content",
                   "endSeatCount":"String content"
            }],
            "supportIncidentFee":"String content",
            "supportPlanFee":"String content"
    }]
}
```

Enable or Disable API Request Logging

This enables or disables API request logging for troubleshooting.

When enabled (apiRequestLogging is true), REST API requests and responses for the user are added to a log. There are additional calls to download the log files (individually or as a zip file) and to clear the log by deleting current entries. A log can have up to 50 requests/responses and the current number of log entries can be determined by getting the settings. Logging is automatically disabled when the log limit of 50 is reached.

Private information, such as passwords and integrator key information, which is normally located in the call header is omitted from the request/response log.

Note: API request logging only captures requests from the authenticated user. Any call that does not authenticate the user and resolve a userId isn't logged. Meaning that login_information, NewAccounts or other distributor-credential calls are not logged.

URL:

/diagnostics/settings

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Reqd?	Type	Description
apiRequestLogging	Yes	String	This enables (true) or disables (false) API request logging for the user.

Example Request Body

```
PUT https://{server}/restapi/{apiVersion}/diagnostics/settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
```

```
Content-Type: application/json

{
    "apiRequestLogging": "true"
}
```

Response

The response returns a success or failure, along with the current API request logging settings for the user.

Example Response Body

```
{
    "apiRequestLogging": "true",
    "apiRequestLogMaxEntries": "50",
    "apiRequestLogRemainingEntries": "50"
}
```

Get API Request Logging Settings

This retrieves the current API request logging setting for the user and remaining log entries.

URL:

/diagnostics/settings

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/diagnostics/settings

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the current API request logging setting for the user, along with the maximum log entries and remaining log entries.

Example Response Body

```
{
    "apiRequestLogging": "true",
    "apiRequestLogMaxEntries": "50",
    "apiRequestLogRemainingEntries": "50"
}
```

Get API Request Logging Log Files

This returns a list of log entries or to download a zip file of the logs.

URL:

/diagnostics/request_logs

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required. If the Accept header is set to application/zip, the response will be a zip file containing individual text files, each representing an API request.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/diagnostics/request_logs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

If the Accept header is set to application/zip, the response will be a zip file containing individual text files, each representing an API request.

If the Accept header is set to application/json or application/xml, the response returns list of log entries in either json or XML. An example json response body is shown below.

Example Response Body

```
{
  "apiRequestLogs": [
    {
      "status": "OK",
      "description": "GetAccountInformation",
      "createdDateTime": "2014-06-03T04:19:01.4200000Z",
```

```
      "requestLogId": "90eb155a-9871-4fae-92dd-3aafcfff1d0f"
    },
    {
      "status": "OK",
      "description": "GetAccountInformation",
      "createdDateTime": "2014-06-03T04:19:01.1830000Z",
      "requestLogId": "6762c5fc-c476-4873-8a90-978729ae3b4f"
    }
  ]
}
```

Get One API Request Logging Log File

This returns information for a single log entry.

URL:

/diagnostics/request_logs/{requestLogId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required. The requestLogfId can be retrieved by getting the list of log entries. The Content-Transfer-Encoding header can be set to base64 to retrieve the API request/response as base 64 string. Otherwise the bytes of the request/response are returned.

Example Request Body

```
GET https://{server}/restapi/{apiVersion}/diagnostics/request_logs/{requestLogId}

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
Content-Transfer-Encoding: base64
```

Response

The response returns the bytes of the request/response. If the Content-Transfer-Encoding header was set to base64, the log is returned as base 64 string.

Clear API Request Logging Logs

This clears the request log files.

Note: The full set of logs for this user is deleted with this call. Individual logs may not be deleted.

URL:

/diagnostics/request_logs

Formats:

XML, JSON

HTTP Method:

DELETE

Parameters:

No additional parameters are required.

Example Request Body

```
DELETE https://{server}/restapi/{apiVersion}/diagnostics/request_logs

X-DocuSign-Authentication:
<DocuSignCredentials><Username>{name}</Username><Password>{password}</Password><Integrato
rKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns a success or failure.

Revoke an Authorization Token

This revokes an OAuth2 authorization server token. After this is done, a caller must re-authenticate to restore access.

URL:

/oauth2/revoke

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

The call must have “token_type” and “access_token” for the server token to be revoked in the Authorization.

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/oauth2/revoke

Authorization:<token_type><access_token>

Accept: application/json
Content-Type: application/json
```

Response

The response returns if the API execution is successful (200 – OK) or failed.

Create an Authorization Token

This creates an OAuth2 authorization server token endpoint.

URL:

/oauth2/token

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

The call uses a request with the

Example Request Body

```
POST https://{server}/restapi/{apiVersion}/oauth2/token

Accept: application/json
Content-Type: application/x-www-form-urlencoded
grant_type=password&client_id={integrator_key}&username={name}&password={password}&scope=
api
```

Response

The response returns if the API execution is successful (200 – OK) or failed. If successful, the response will contain the server token information.

Name	Description
access_token	Access token information.

Name	Description
scope	This should always be “api”
token_type	This should always be “bearer”

The following example shows the response json body.

Example Response Body

```
{
    "access_token":"String content",
    "scope":"String content",
    "token_type":"String content"
}
```

Recipient Types and Parameters

This section has the parameter information for the different recipient types. There are seven possible recipient types: Agents, Carbon Copies, Certified Deliveries, Editors, In Person Signers, Intermediaries, and Signers. Recipient types share some common parameters, but the exact parameters associated with a recipient depend on the recipient type. Refer to the specific recipient type below for more information.

Agents Carbon Copies Certified Deliveries Editors
In Person Signers Intermediaries Signers

Agents Recipient Type

This recipient can add name and email information for recipients that appear after the recipient in routing order. This recipient type can only be used if enabled for your account.

An example Agents json layout is shown below:

```
"agents": [{
    "email": "email.name@company.com",
    "name": "recipient name",
    "accessCode": "",
    "addAccessCodeToEmail": false,
    "clientUserId": null,
    "embeddedRecipientStartURL": "string",
    "customFields": {
           "sample string 1",
           "sample string 2"
    },
    "emailNotification": {
           "emailBody":" ",
           "emailSubject":" ",
           "supportedLanguage":" "
    },
    "excludedDocuments": ["2", "4"]
    "idCheckConfigurationName": null,
    "idCheckInformationInput": {
           "addressInformationInput": {
                  "addressInformation": {
                         "street1": "sample string 1",
                         "street2": "sample string 2",
```

```
                        "city": "sample string 3",
                        "state": "sample string 4",
                        "zip": "sample string 5",
                        "zipPlus4": "sample string 6"
                },
                "displayLevelCode": "sample string 1",
                "receiveInResponse": "sample string 2"
        },
        "dobInformationInput": {
                "dateOfBirth": "sample string 1",
                "displayLevelCode": "sample string 2",
                "receiveInResponse": "sample string 3"
        },
        "ssn4InformationInput": {
                "ssn4": "sample string 1",
                "displayLevelCode": "sample string 2",
                "receiveInResponse": "sample string 3"
        },
        "ssn9InformationInput": {
                "ssn9": "sample string 1",
                "displayLevelCode": "sample string 2"
        }
    },
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": {
        "recipMayProvideNumber": "sample string 1",
        "validateRecipProvidedNumber": "sample string 2",
        "recordVoicePrint": "sample string 3",
        "senderProvidedNumbers": [
                "sample string 1",
                "sample string 2"
        ]
    },
    "recipientAttachment": null,
    "recipientId": "1",
    "requireIdLookup": false,
    "roleName": "",
    "routingOrder": 1,
    "samlAuthentication": {
        "samlAssertionAttributes": [
                {
                        "name": "string",
                        "value": "string"
                },
                {
                        "name": "string",
                        "value": "string"
                }
        ]
    },
    "smsAuthentication": {
        "senderProvidedNumbers":[
                "sample string 1",
                "sample string 2"
        ]
    },
    "socialAuthentications": null,
    "templateAccessCodeRequired": false,
    "templateLocked": false,
    "templateRequired": false,
    "canEditRecipientEmails": false,
    "canEditRecipientNames": false,
    "signingGroupId": null
```

```
}],
```

Note for xml there is an additional agent level in the layout.

```
<agents>
    <agent>
            [parameters]
    </agent>
</agents>
```

The Agents recipient type parameters are:

Name	Reqd?	Type	Description
email	Yes	String	Email of the recipient. Notification will be sent to this email id. This can be a maximum of 100 characters.
name	Yes	String	The full legal name of the recipient. This can be a maximum of 100 characaters.
accessCode	No	String	This optional element specifies the access code a recipient has to enter to validate the identity. This can be a maximum of 50 characters.
addAccessCodeToEmail	No	Boolean	This optional attribute indicates that the access code will be added to the email sent to the recipient; this nullifies the Security measure of Access Code on the recipient.
clientUserId	No	String	This specifies if the recipient is embedded or remote. If the clientUserId is not null then the recipient is embedded. Note that if a ClientUserId is used and the account settings SignerMustHaveAccount or SignerMustLoginToSign are true, an error is generated on sending.
embeddedRecipientStartURL	No	String	This is a sender provided valid URL string for redirecting an embedded recipient. When using this option, the embedded recipient still receives an email from DocuSign, just as a remote recipient would, but when the document link in the email is

Name	Reqd?	Type	Description
			clicked the recipient is redirected, through DocuSign, to this URL to complete their actions. When routing to the URL, it is up to the sender's system (the server responding to the URL) to then request a recipient token to launch a signing session. If the value SIGN_AT_DOCUSIGN is used for this node, the recipient will be directed to an embedded signing or viewing process directly at DocuSign. The signing or viewing action is initiated by the DocuSign system and the transaction activity and Certificate of Completion records will reflect this. In all other ways the process is identical to an embedded signing or viewing operation that would be launched by any partner. It is important to remember that in a typical embedded workflow the authentication of an embedded recipient is the responsibility of the sending application and DocuSign expects that senders will follow their own process for establishing the recipient's identity. In this workflow the recipient goes through the sending application before the embedded signing or viewing process in initiated. However, when the sending application sets EmbeddedRecipientStartURL= SIGN_AT_DOCUSIGN, the recipient goes directly to the embedded signing or viewing process bypassing the sending application and any authentication steps the sending application would use. In this case, DocuSign recommends that one of the normal DocuSign authentication features (Access Code, Phone Authentication, SMS

Name	Reqd?	Type	Description
			Authentication, etc.) be used to verify the identity of the recipient. If a clientUserId is NOT provided and a embeddedRecipientStartURL is provided, DocuSign will ignore the redirect URL and launch the standard signing process for the email recipient. Information can be appended to the embeddedRecipientStartURL using merge fields. The available merge fields items are: envelopeId, recipientId, recipientName, recipientEmail, and customFields. The customFields must be part of the recipient or envelope. The merge fields are enclosed in double brackets. *Example:* http://senderHost/[[mergeField1]]/beginSigningSession?[[mergeField2]]&[[mergeField3]]
customFields	No	customField	An optional array of strings that allows the sender to provide custom data about the recipient. This information is returned in the envelope status but otherwise not used by DocuSign. Each customField string can be a maximum of 100 characters.
emailNotification	No	emailNotification	An optional complex type that has information for setting the language for the recipient's email information. It is composed of three elements: • emailBody: a string with the email message sent to the recipient. This can be a maximum of 10000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the simple type enumeration the

Name	Reqd?	Type	Description
			language used. The supported languages, with the language value shown in parenthesis, are: Arabic (ar), Bahasa Indonesia (id), Bahasa Melayu (ms) Bulgarian (bg), Czech (cs), Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Croatian (hr), Danish (da), Dutch (nl), English US (en), English UK (en_GB), Estonian (et), Farsi (fa), Finnish (fi), French (fr), French Canada (fr_CA), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Italian (it), Japanese (ja), Korean (ko), Latvian (lv), Lithuanian (lt), Norwegian (no), Polish (pl), Portuguese (pt), Portuguese Brazil (pt_BR), Romanian (ro),Russian (ru), Serbian (sr), Slovak (sk), Slovenian (sl), Spanish (es),Spanish Latin America (es_MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). **IMPORTANT:** If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.

Name	Reqd?	Type	Description
excludedDocuments	No	Array of Strings	Specifies the documents that are not visible to this recipient. Document Visibility must be enabled for the account and the enforceSignerVisibility property must be set to true for the envelope to use this. When enforceSignerVisibility is enabled, documents with tabs can only be viewed by signers that have a tab on that document. Recipients that have an administrative role (Agent, Editor, or Intermediaries) or informational role (Certified Deliveries or Carbon Copies) can always see all the documents in an envelope, unless they are specifically excluded using this setting when an envelope is sent. Documents that do not have tabs are always visible to all recipients, unless they are specifically excluded using this setting when an envelope is sent.
idCheckConfigurationName	No	String	Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these name can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting. *Example:* Your account has ID Check and SMS Authentication available and in the web console Identify list these appear as 'ID Check $' and 'SMS Auth $'. To use ID check in an envelope, the idCheckConfigurationName should be 'ID Check $'. If you wanted to use SMS, it would be 'SMS Auth $' and you would need to add phone number information to the smsAuthentication node.
idCheckInformationInput	No		This complex element contains input information related to a

Name	Reqd?	Type	Description
			recipient ID check. It can include the following information. addressInformationInput: Used to set recipient address information and consists of: • addressInformation: consists of six elements, with street2 and zipPlus4 being optional. The elements are: street1, street2, city, state, zip, zipPlus4. The maximum number of characters in each element are: street1/street2 = 150 characters, city = 50 characters, state = 2 characters, and zip/zipPlus4 = 20 characters. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. dobInformationInput: Used to set recipient date of birth information and consists of: • dateOfBirth: Specifies the recipient's date, month and year of birth. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn4InformationInput: Used to set

Name	Reqd?	Type	Description
			the last four digits of the recipient's SSN information and consists of: • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn9InformationInput: Used to set the recipient's SSN information. Note that the ssn9 information can never be returned in the response. The ssn9 input consists of: • ssn9: Specifies the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay.
inheritEmailNotificationConfiguration	No	Boolean	Optional element. If true and the recipient creates a DocuSign account after signing, the Manage Account Email Notification settings are used as the default settings for the recipient's account.
note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will display in the signing UI near the upper left corner of the document on the signing screen. This can be a maximum of 1000 characters.

Name	Reqd?	Type	Description
phoneAuthentication	No	RecipientPhone Authentication	Optional element. Contains the elements: • recipMayProvideNumber – Boolean (if true then recipient can use whatever phone number they want) • senderProvidedNumbers – ArrayOfString (a list of phone numbers the recipient may use) • recordVoicePrint – Reserved • validateRecipProvidedNumber – Reserved
recipientAttachment	No	Attachment	Reserved for DocuSign use.
recipientId	No	String	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.
requireIdLookup	No	Boolean	If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate their identity
roleName	No*	String	Optional element. This is the role name associated with the recipient. * - This is required when working with template recipients.
routingOrder	Yes	String	This specifies the routing order of the recipient in the envelope.
samlAuthentication	No	samlAssertionAttributes	Reserved for DocuSign use only.
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: • senderProvidedNumbers – Array with a list of phone numbers the recipient may use for SMS text authentication.
socialAuthentications	No	Boolean	Lists the social ID type that can be used for recipient authentication.

Name	Reqd?	Type	Description
templateAccessCodeRequired	No	Boolean	Optional element. Used only when working with template recipients. If true and TemplateLocked = true, the sender must enter an access code.
templateLocked	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender cannot change any attributes of the recipient.
templateRequired	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender may not remove the recipient.
canEditRecipientEmail	No	Boolean	Optional element only used with recipient types Agents and Editors. If true, the Agent or Editor Recipient associated with this Recipient can change the Recipient's pre-populated Email address. This element is only active if enabled for the account.
canEditRecipientName	No	Boolean	Optional element only used with recipient types Agents and Editors. If true, the Agent or Editor Recipient associated with this recipient can change the recipient's pre-populated name (UserName). This element is only active if enabled for the account.
signingGroupId	No	String	The DocuSign generated Signing Group ID. When used, the email and name parameters are not required for this recipient.

Carbon Copies Recipient Type

Use this action if the recipient should get a copy of the envelope, but the recipient does not need to sign, initial, date or add information to any of the documents. This type of recipient can be placed in any order in the recipient list. The recipient receives a copy of the envelope when the envelope reaches the recipient's order in the process flow and when the envelope is completed.

An example Carbon Copies json layout is shown below:

```
"carbonCopies": [{
```

```
    "email": "email.name@company.com",
    "name": "recipient name",
    "accessCode": "",
    "addAccessCodeToEmail": false,
    "clientUserId": null,
    "embeddedRecipientStartURL": "string",
    "customFields": {
          "sample string 1",
          "sample string 2"
    },
    "emailNotification": {
          "emailBody":" ",
          "emailSubject":" ",
          "supportedLanguage":" "
    },
    "excludedDocuments": ["2", "4"]
    "idCheckConfigurationName": null,
    "idCheckInformationInput": {
          "addressInformationInput": {
                "addressInformation": {
                      "street1": "sample string 1",
                      "street2": "sample string 2",
                      "city": "sample string 3",
                      "state": "sample string 4",
                      "zip": "sample string 5",
                      "zipPlus4": "sample string 6"
                },
                "displayLevelCode": "sample string 1",
                "receiveInResponse": "sample string 2"
          },
          "dobInformationInput": {
                "dateOfBirth": "sample string 1",
                "displayLevelCode": "sample string 2",
                "receiveInResponse": "sample string 3"
          },
          "ssn4InformationInput": {
                "ssn4": "sample string 1",
                "displayLevelCode": "sample string 2",
                "receiveInResponse": "sample string 3"
          },
          "ssn9InformationInput": {
                "ssn9": "sample string 1",
                "displayLevelCode": "sample string 2"
          }
    },
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": {
          "recipMayProvideNumber": "sample string 1",
          "validateRecipProvidedNumber": "sample string 2",
          "recordVoicePrint": "sample string 3",
          "senderProvidedNumbers": [
                "sample string 1",
                "sample string 2"
          ]
    },
    "recipientAttachment": null,
    "recipientId": "1",
    "requireIdLookup": false,
    "roleName": "",
    "routingOrder": 3,
    "samlAuthentication": {
          "samlAssertionAttributes": [
                {
```

```
                        "name": "string",
                        "value": "string"
                },
                {
                        "name": "string",
                        "value": "string"
                }
            ]
    },
    "smsAuthentication": {
            "senderProvidedNumbers":[
                    "sample string 1",
                    "sample string 2"
            ]
    },
    "socialAuthentications": null,
    "templateAccessCodeRequired": false,
    "templateLocked": false,
    "templateRequired": false,
    "signingGroupId": null
}],
```

Note for xml there is an additional carbonCopy level in the layout.

```
<carbonCopies>
    <carbonCoy>
            [parameters]
    </carbonCopy>
</carbonCopies>
```

The Carbon Copies recipient type parameters are:

Name	Reqd?	Type	Description
email	Yes	String	Email of the recipient. Notification will be sent to this email id. This can be a maximum of 100 characters.
name	Yes	String	The full legal name of the recipient. This can be a maximum of 100 characaters.
accessCode	No	String	This optional element specifies the access code a recipient has to enter to validate the identity. This can be a maximum of 50 characters.
addAccessCodeToEmail	No	Boolean	This optional attribute indicates that the access code will be added to the email sent to the recipient; this nullifies the Security measure of Access Code on the recipient.

Name	Reqd?	Type	Description
clientUserId	No	String	This specifies if the recipient is embedded or remote. If the clientUserId is not null then the recipient is embedded. Note that if a ClientUserId is used and the account settings SignerMustHaveAccount or SignerMustLoginToSign are true, an error is generated on sending.
embeddedRecipientStartURL	No	String	This is a sender provided valid URL string for redirecting an embedded recipient. When using this option, the embedded recipient still receives an email from DocuSign, just as a remote recipient would, but when the document link in the email is clicked the recipient is redirected, through DocuSign, to this URL to complete their actions. When routing to the URL, it is up to the sender's system (the server responding to the URL) to then request a recipient token to launch a signing session. If the value SIGN_AT_DOCUSIGN is used for this node, the recipient will be directed to an embedded signing or viewing process directly at DocuSign. The signing or viewing action is initiated by the DocuSign system and the transaction activity and Certificate of Completion records will reflect this. In all other ways the process is identical to an embedded signing or viewing operation that would be launched by any partner. It is important to remember that in a typical embedded workflow the authentication of an embedded recipient is the responsibility of the sending application and DocuSign expects that senders will follow their own process for establishing the recipient's

Name	Reqd?	Type	Description
			identity. In this workflow the recipient goes through the sending application before the embedded signing or viewing process in initiated. However, when the sending application sets EmbeddedRecipientStartURL= SIGN_AT_DOCUSIGN, the recipient goes directly to the embedded signing or viewing process bypassing the sending application and any authentication steps the sending application would use. In this case, DocuSign recommends that one of the normal DocuSign authentication features (Access Code, Phone Authentication, SMS Authentication, etc.) be used to verify the identity of the recipient. If a clientUserId is NOT provided and a embeddedRecipientStartURL is provided, DocuSign will ignore the redirect URL and launch the standard signing process for the email recipient. Information can be appended to the embeddedRecipientStartURL using merge fields. The available merge fields items are: envelopeId, recipientId, recipientName, recipientEmail, and customFields. The customFields must be part of the recipient or envelope. The merge fields are enclosed in double brackets. *Example:* http://senderHost/[[mergeField1]]/beginSigningSession?[[mergeField2]]&[[mergeField3]]

Name	Reqd?	Type	Description
customFields	No	customField	An optional array of strings that allows the sender to provide custom data about the recipient. This information is returned in the envelope status but otherwise not used by DocuSign. Each customField string can be a maximum of 100 characters.
emailNotification	No	emailNotification	An optional complex type that has information for setting the language for the recipient's email information. It is composed of three elements: • emailBody: a string with the email message sent to the recipient. This can be a maximum of 10000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the simple type enumeration the language used. The supported languages, with the language value shown in parenthesis, are: Arabic (ar), Bahasa Indonesia (id), Bahasa Melayu (ms) Bulgarian (bg), Czech (cs), Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Croatian (hr), Danish (da), Dutch (nl), English US (en), English UK (en_GB), Estonian (et), Farsi (fa), Finnish (fi), French (fr), French Canada (fr_CA), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Italian (it), Japanese (ja), Korean (ko), Latvian (lv), Lithuanian (lt), Norwegian (no), Polish (pl), Portuguese (pt), Portuguese Brazil (pt_BR), Romanian (ro),Russian (ru), Serbian (sr), Slovak (sk), Slovenian (sl), Spanish (es),Spanish Latin

Name	Reqd?	Type	Description
			America (es_MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). **IMPORTANT:** If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.
excludedDocuments	No	Array of Strings	Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account and the enforceSignerVisibility property must be set to true for the envelope to use this. When enforceSignerVisibility is enabled, documents with tabs can only be viewed by signers that have a tab on that document. Recipients that have an administrative role (Agent, Editor, or Intermediaries) or informational role (Certified Deliveries or Carbon Copies) can always see all the documents in an envelope, unless they are specifically excluded using this setting when an envelope is sent. Documents that do not have tabs are always visible to all recipients, unless they are specifically excluded using this setting when an envelope is sent.
idCheckConfigurationName	No	String	Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these name can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting. *Example:* Your account has ID Check and SMS Authentication available and in the web console Identify list these appear as 'ID Check $' and 'SMS Auth $'. To

Name	Reqd?	Type	Description
			use ID check in an envelope, the idCheckConfigurationName should be 'ID Check $'. If you wanted to use SMS, it would be 'SMS Auth $' and you would need to add phone number information to the smsAuthentication node.
idCheckInformationInput	No		This complex element contains input information related to a recipient ID check. It can include the following information. addressInformationInput: Used to set recipient address information and consists of: • addressInformation: consists of six elements, with street2 and zipPlus4 being optional. The elements are: street1, street2, city, state, zip, zipPlus4. The maximum number of characters in each element are: street1/street2 = 150 characters, city = 50 characters, state = 2 characters, and zip/zipPlus4 = 20 characters. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. dobInformationInput: Used to set recipient date of birth information and consists of: • dateOfBirth: Specifies the recipient's date, month and year of birth. • displayLevelCode: Specifies the display level for the recipient. Values are:

Name	Reqd?	Type	Description
			ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of: • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn9InformationInput: Used to set the recipient's SSN information. Note that the ssn9 information can never be returned in the response. The ssn9 input consists of: • ssn9: Specifies the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay.
inheritEmailNotificationConfiguration	No	Boolean	Optional element. If true and the recipient creates a DocuSign account after signing, the Manage Account Email Notification settings are used as the default settings for the recipient's account.

Name	Reqd?	Type	Description
note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will display in the signing UI near the upper left corner of the document on the signing screen. This can be a maximum of 1000 characters.
phoneAuthentication	No	RecipientPhone Authentication	Optional element. Contains the elements: • recipMayProvideNumber – Boolean (if true then recipient can use whatever phone number they want) • senderProvidedNumbers – ArrayOfString (a list of phone numbers the recipient may use) • recordVoicePrint – Reserved • validateRecipProvidedNumber – Reserved
recipientAttachment	No	Attachment	Reserved for DocuSign use.
recipientId	Yes	string	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.
requireIdLookup	No	Boolean	If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate their identity
roleName	No*	String	Optional element. This is the role name associated with the recipient. * - This is required when working with template recipients.
routingOrder	Yes	String	This specifies the routing order of the recipient in the envelope.
samlAuthentication	No	samlAssertionAttributes	Reserved for DocuSign use only.

Name	Reqd?	Type	Description
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: • senderProvidedNumbers – Array with a list of phone numbers the recipient may use for SMS text authentication.
socialAuthentications	No	Boolean	Lists the social ID type that can be used for recipient authentication.
templateAccessCodeRequired	No	Boolean	Optional element. Used only when working with template recipients. If true and TemplateLocked = true, the sender must enter an access code.
templateLocked	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender cannot change any attributes of the recipient.
templateRequired	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender may not remove the recipient.
signingGroupId	No	String	The DocuSign generated Signing Group ID. When used, the email and name parameters are not required for this recipient.

Certified Deliveries Recipient Type

Use this action if the recipient must receive the completed documents for the envelope to be completed, but the recipient does not need to sign, initial, date or add information to any of the documents. This recipient type can only be used if enabled for your account.

An example Certified Deliveries json layout is shown below:

```
"certifiedDeliveries": [{
    "email": "email.name@company.com",
    "name": "recipient name",
    "accessCode": "",
    "addAccessCodeToEmail": false,
    "clientUserId": null,
    "embeddedRecipientStartURL": "string",
    "customFields": {
          "sample string 1",
```

```
            "sample string 2"
    },
    "emailNotification": {
            "emailBody":" ",
            "emailSubject":" ",
            "supportedLanguage":" "
    },
    "excludedDocuments": ["2", "4"]
    "idCheckConfigurationName": null,
    "idCheckInformationInput": {
            "addressInformationInput": {
                    "addressInformation": {
                            "street1": "sample string 1",
                            "street2": "sample string 2",
                            "city": "sample string 3",
                            "state": "sample string 4",
                            "zip": "sample string 5",
                            "zipPlus4": "sample string 6"
                    },
                    "displayLevelCode": "sample string 1",
                    "receiveInResponse": "sample string 2"
            },
            "dobInformationInput": {
                    "dateOfBirth": "sample string 1",
                    "displayLevelCode": "sample string 2",
                    "receiveInResponse": "sample string 3"
            },
            "ssn4InformationInput": {
                    "ssn4": "sample string 1",
                    "displayLevelCode": "sample string 2",
                    "receiveInResponse": "sample string 3"
            },
            "ssn9InformationInput": {
                    "ssn9": "sample string 1",
                    "displayLevelCode": "sample string 2"
            }
    },
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": {
            "recipMayProvideNumber": "sample string 1",
            "validateRecipProvidedNumber": "sample string 2",
            "recordVoicePrint": "sample string 3",
            "senderProvidedNumbers": [
                    "sample string 1",
                    "sample string 2"
            ]
    },
    "recipientAttachment": null,
    "recipientId": "1",
    "requireIdLookup": false,
    "roleName": "",
    "routingOrder": 3,
    "samlAuthentication": {
            "samlAssertionAttributes": [
                    {
                            "name": "string",
                            "value": "string"
                    },
                    {
                            "name": "string",
                            "value": "string"
                    }
            ]
```

```
    },
    "smsAuthentication": {
           "senderProvidedNumbers":[
                  "sample string 1",
                  "sample string 2"
           ]
    },
    "socialAuthentications": null,
    "templateAccessCodeRequired": false,
    "templateLocked": false,
    "templateRequired": false,
    "signingGroupId": null
}],
```

Note for xml there is an additional certifiedDelivery level in the layout.

```
<certifiedDeliveries>
    <certifiedDelivery>
           [parameters]
    </certifiedDelivery>
</certifiedDeliveries>
```

The Certified Deliveries recipient type parameters are:

Name	Reqd?	Type	Description
email	Yes	String	Email of the recipient. Notification will be sent to this email id. This can be a maximum of 100 characters.
name	Yes	String	The full legal name of the recipient. This can be a maxium of 100 characaters.
accessCode	No	String	This optional element specifies the access code a recipient has to enter to validate the identity. This can be a maximum of 50 characters.
addAccessCodeToEmail	No	Boolean	This optional attribute indicates that the access code will be added to the email sent to the recipient; this nullifies the Security measure of Access Code on the recipient.

Name	Reqd?	Type	Description
clientUserId	No	String	This specifies if the recipient is embedded or remote. If the clientUserId is not null then the recipient is embedded. Note that if a ClientUserId is used and the account settings SignerMustHaveAccount or SignerMustLoginToSign are true, an error is generated on sending.
embeddedRecipientStartURL	No	String	This is a sender provided valid URL string for redirecting an embedded recipient. When using this option, the embedded recipient still receives an email from DocuSign, just as a remote recipient would, but when the document link in the email is clicked the recipient is redirected, through DocuSign, to this URL to complete their actions. When routing to the URL, it is up to the sender's system (the server responding to the URL) to then request a recipient token to launch a signing session. If the value SIGN_AT_DOCUSIGN is used for this node, the recipient will be directed to an embedded signing or viewing process directly at DocuSign. The signing or viewing action is initiated by the DocuSign system and the transaction activity and Certificate of Completion records will reflect this. In all other ways the process is identical to an embedded signing or viewing operation that would be launched by any partner. It is important to remember that in a typical embedded workflow the authentication of an embedded recipient is the responsibility of the sending application and DocuSign expects that senders will follow their own process for establishing the recipient's

Name	Reqd?	Type	Description
			identity. In this workflow the recipient goes through the sending application before the embedded signing or viewing process in initiated. However, when the sending application sets EmbeddedRecipientStartURL= SIGN_AT_DOCUSIGN, the recipient goes directly to the embedded signing or viewing process bypassing the sending application and any authentication steps the sending application would use. In this case, DocuSign recommends that one of the normal DocuSign authentication features (Access Code, Phone Authentication, SMS Authentication, etc.) be used to verify the identity of the recipient. If a clientUserId is NOT provided and a embeddedRecipientStartURL is provided, DocuSign will ignore the redirect URL and launch the standard signing process for the email recipient. Information can be appended to the embeddedRecipientStartURL using merge fields. The available merge fields items are: envelopeId, recipientId, recipientName, recipientEmail, and customFields. The customFields must be part of the recipient or envelope. The merge fields are enclosed in double brackets. *Example:* http://senderHost/[[mergeField1]]/beginSigningSession?[[mergeField2]]&[[mergeField3]]

Name	Reqd?	Type	Description
customFields	No	customField	An optional array of strings that allows the sender to provide custom data about the recipient. This information is returned in the envelope status but otherwise not used by DocuSign. Each customField string can be a maximum of 100 characters.
emailNotification	No	emailNotification	An optional complex type that has information for setting the language for the recipient's email information. It is composed of three elements: • emailBody: a string with the email message sent to the recipient. This can be a maximum of 10000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the simple type enumeration the language used. The supported languages, with the language value shown in parenthesis, are: Arabic (ar), Bahasa Indonesia (id), Bahasa Melayu (ms) Bulgarian (bg), Czech (cs), Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Croatian (hr), Danish (da), Dutch (nl), English US (en), English UK (en_GB), Estonian (et), Farsi (fa), Finnish (fi), French (fr), French Canada (fr_CA), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Italian (it), Japanese (ja), Korean (ko), Latvian (lv), Lithuanian (lt), Norwegian (no), Polish (pl), Portuguese (pt), Portuguese Brazil (pt_BR), Romanian (ro),Russian (ru), Serbian (sr), Slovak (sk), Slovenian (sl), Spanish (es),Spanish Latin

Name	Reqd?	Type	Description
			America (es_MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). **IMPORTANT:** If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.
excludedDocuments	No	Array of Strings	Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account and the enforceSignerVisibility property must be set to true for the envelope to use this. When enforceSignerVisibility is enabled, documents with tabs can only be viewed by signers that have a tab on that document. Recipients that have an administrative role (Agent, Editor, or Intermediaries) or informational role (Certified Deliveries or Carbon Copies) can always see all the documents in an envelope, unless they are specifically excluded using this setting when an envelope is sent. Documents that do not have tabs are always visible to all recipients, unless they are specifically excluded using this setting when an envelope is sent.
idCheckConfigurationName	No	String	Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these name can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting. *Example:* Your account has ID Check and SMS Authentication available and in the web console Identify list these appear as 'ID Check $' and 'SMS Auth $'. To

Name	Reqd?	Type	Description
			use ID check in an envelope, the idCheckConfigurationName should be 'ID Check $'. If you wanted to use SMS, it would be 'SMS Auth $' and you would need to add phone number information to the smsAuthentication node.
idCheckInformationInput	No		This complex element contains input information related to a recipient ID check. It can include the following information. addressInformationInput: Used to set recipient address information and consists of: • addressInformation: consists of six elements, with street2 and zipPlus4 being optional. The elements are: street1, street2, city, state, zip, zipPlus4. The maximum number of characters in each element are: street1/street2 = 150 characters, city = 50 characters, state = 2 characters, and zip/zipPlus4 = 20 characters. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. dobInformationInput: Used to set recipient date of birth information and consists of: • dateOfBirth: Specifies the recipient's date, month and year of birth. • displayLevelCode: Specifies the display level for the recipient. Values are:

Name	Reqd?	Type	Description
			ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of: • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn9InformationInput: Used to set the recipient's SSN information. Note that the ssn9 information can never be returned in the response. The ssn9 input consists of: • ssn9: Specifies the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay.
inheritEmailNotificationConfiguration	No	Boolean	Optional element. If true and the recipient creates a DocuSign account after signing, the Manage Account Email Notification settings are used as the default settings for the recipient's account.

Name	Reqd?	Type	Description
note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will display in the signing UI near the upper left corner of the document on the signing screen. This can be a maximum of 1000 characters.
phoneAuthentication	No	RecipientPhone Authentication	Optional element. Contains the elements: • recipMayProvideNumber – Boolean (if true then recipient can use whatever phone number they want) • senderProvidedNumbers – ArrayOfString (a list of phone numbers the recipient may use) • recordVoicePrint – Reserved • validateRecipProvidedNumber – Reserved
recipientAttachment	No	Attachment	Reserved for DocuSign use.
recipientId	Yes	String	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.
requireIdLookup	No	Boolean	If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate their identity
roleName	No*	String	Optional element. This is the role name associated with the recipient. * - This is required when working with template recipients.
routingOrder	Yes	String	This specifies the routing order of the recipient in the envelope.
samlAuthentication	No	samlAssertionAttributes	Reserved for DocuSign use only.

Name	Reqd?	Type	Description
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: • senderProvidedNumbers – Array with a list of phone numbers the recipient may use for SMS text authentication.
socialAuthentications	No	Boolean	Lists the social ID type that can be used for recipient authentication.
templateAccessCodeRequired	No	Boolean	Optional element. Used only when working with template recipients. If true and TemplateLocked = true, the sender must enter an access code.
templateLocked	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender cannot change any attributes of the recipient.
templateRequired	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender may not remove the recipient.
signingGroupId	No	String	The DocuSign generated Signing Group ID. When used, the email and name parameters are not required for this recipient.

Editors Recipient Type

This recipient has the same management and access rights for the envelope as the sender and can make changes to the envelope as if they were using the Advanced Correct feature. This recipient can add name and email information, add or change the routing order and set authentication options for the remaining recipients. Additionally, this recipient can edit signature/initial tabs and data fields for the remaining recipients. The recipient must have a DocuSign account to be an editor. This recipient type can only be used if enabled for your account.

An example Editors json layout is shown below:

```
"editors": [{
    "email": "email.name@company.com",
    "name": "recipient name",
    "accessCode": "",
    "addAccessCodeToEmail": false,
```

```
    "clientUserId": null,
    "embeddedRecipientStartURL": "string",
    "customFields": {
           "sample string 1",
           "sample string 2"
    },
    "emailNotification": {
           "emailBody":" ",
           "emailSubject":" ",
           "supportedLanguage":" "
    },
    "excludedDocuments": ["2", "4"]
    "idCheckConfigurationName": null,
    "idCheckInformationInput": {
           "addressInformationInput": {
                  "addressInformation": {
                         "street1": "sample string 1",
                         "street2": "sample string 2",
                         "city": "sample string 3",
                         "state": "sample string 4",
                         "zip": "sample string 5",
                         "zipPlus4": "sample string 6"
                  },
                  "displayLevelCode": "sample string 1",
                  "receiveInResponse": "sample string 2"
           },
           "dobInformationInput": {
                  "dateOfBirth": "sample string 1",
                  "displayLevelCode": "sample string 2",
                  "receiveInResponse": "sample string 3"
           },
           "ssn4InformationInput": {
                  "ssn4": "sample string 1",
                  "displayLevelCode": "sample string 2",
                  "receiveInResponse": "sample string 3"
           },
           "ssn9InformationInput": {
                  "ssn9": "sample string 1",
                  "displayLevelCode": "sample string 2"
           }
    },
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": {
           "recipMayProvideNumber": "sample string 1",
           "validateRecipProvidedNumber": "sample string 2",
           "recordVoicePrint": "sample string 3",
           "senderProvidedNumbers": [
                  "sample string 1",
                  "sample string 2"
           ]
    },
    "recipientAttachment": null,
    "recipientId": "1",
    "requireIdLookup": false,
    "roleName": "",
    "routingOrder": 1,
    "samlAuthentication": {
           "samlAssertionAttributes": [
                  {
                         "name": "string",
                         "value": "string"
                  },
                  {
```

```
                        "name": "string",
                        "value": "string"
                    }
            ]
    },
    "smsAuthentication": {
            "senderProvidedNumbers":[
                    "sample string 1",
                    "sample string 2"
            ]
    },
    "socialAuthentications": null,
    "templateAccessCodeRequired": false,
    "templateLocked": false,
    "templateRequired": false,
    "canEditRecipientEmails": false,
    "canEditRecipientNames": false,
    "signingGroupId": null
}],
```

Note for xml there is an additional editor level in the layout.

```
<editors>
    <editor>
            [parameters]
    </editor>
</editors>
```

The Editors recipient type parameters are:

Name	Reqd?	Type	Description
email	Yes	String	Email of the recipient. Notification will be sent to this email id. This can be a maximum of 100 characters.
name	Yes	String	The full legal name of the recipient. This can be a maxium of 100 characaters.
accessCode	No	String	This optional element specifies the access code a recipient has to enter to validate the identity. This can be a maximum of 50 characters.
addAccessCodeToEmail	No	Boolean	This optional attribute indicates that the access code will be added to the email sent to the recipient; this nullifies the Security measure of Access Code on the recipient.

Name	Reqd?	Type	Description
clientUserId	No	String	This specifies if the recipient is embedded or remote. If the clientUserId is not null then the recipient is embedded. Note that if a ClientUserId is used and the account settings SignerMustHaveAccount or SignerMustLoginToSign are true, an error is generated on sending.
embeddedRecipientStartURL	No	String	This is a sender provided valid URL string for redirecting an embedded recipient. When using this option, the embedded recipient still receives an email from DocuSign, just as a remote recipient would, but when the document link in the email is clicked the recipient is redirected, through DocuSign, to this URL to complete their actions. When routing to the URL, it is up to the sender's system (the server responding to the URL) to then request a recipient token to launch a signing session. If the value SIGN_AT_DOCUSIGN is used for this node, the recipient will be directed to an embedded signing or viewing process directly at DocuSign. The signing or viewing action is initiated by the DocuSign system and the transaction activity and Certificate of Completion records will reflect this. In all other ways the process is identical to an embedded signing or viewing operation that would be launched by any partner. It is important to remember that in a typical embedded workflow the authentication of an embedded recipient is the responsibility of the sending application and DocuSign expects that senders will follow their own process for establishing the recipient's

Name	Reqd?	Type	Description
			identity. In this workflow the recipient goes through the sending application before the embedded signing or viewing process in initiated. However, when the sending application sets EmbeddedRecipientStartURL= SIGN_AT_DOCUSIGN, the recipient goes directly to the embedded signing or viewing process bypassing the sending application and any authentication steps the sending application would use. In this case, DocuSign recommends that one of the normal DocuSign authentication features (Access Code, Phone Authentication, SMS Authentication, etc.) be used to verify the identity of the recipient. If a clientUserId is NOT provided and a embeddedRecipientStartURL is provided, DocuSign will ignore the redirect URL and launch the standard signing process for the email recipient. Information can be appended to the embeddedRecipientStartURL using merge fields. The available merge fields items are: envelopeId, recipientId, recipientName, recipientEmail, and customFields. The customFields must be part of the recipient or envelope. The merge fields are enclosed in double brackets. *Example:* http://senderHost/[[mergeField1]]/beginSigningSession?[[mergeField2]]&[[mergeField3]]

Name	Reqd?	Type	Description
customFields	No	customField	An optional array of strings that allows the sender to provide custom data about the recipient. This information is returned in the envelope status but otherwise not used by DocuSign. Each customField string can be a maximum of 100 characters.
emailNotification	No	emailNotification	An optional complex type that has information for setting the language for the recipient's email information. It is composed of three elements: • emailBody: a string with the email message sent to the recipient. This can be a maximum of 10000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the simple type enumeration the language used. The supported languages, with the language value shown in parenthesis, are: Arabic (ar), Bahasa Indonesia (id), Bahasa Melayu (ms) Bulgarian (bg), Czech (cs), Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Croatian (hr), Danish (da), Dutch (nl), English US (en), English UK (en_GB), Estonian (et), Farsi (fa), Finnish (fi), French (fr), French Canada (fr_CA), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Italian (it), Japanese (ja), Korean (ko), Latvian (lv), Lithuanian (lt), Norwegian (no), Polish (pl), Portuguese (pt), Portuguese Brazil (pt_BR), Romanian (ro),Russian (ru), Serbian (sr), Slovak (sk), Slovenian (sl), Spanish (es),Spanish Latin

Name	Reqd?	Type	Description
			America (es_MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). **IMPORTANT:** If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.
excludedDocuments	No	Array of Strings	Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account and the enforceSignerVisibility property must be set to true for the envelope to use this. When enforceSignerVisibility is enabled, documents with tabs can only be viewed by signers that have a tab on that document. Recipients that have an administrative role (Agent, Editor, or Intermediaries) or informational role (Certified Deliveries or Carbon Copies) can always see all the documents in an envelope, unless they are specifically excluded using this setting when an envelope is sent. Documents that do not have tabs are always visible to all recipients, unless they are specifically excluded using this setting when an envelope is sent.
idCheckConfigurationName	No	String	Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these name can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting. *Example:* Your account has ID Check and SMS Authentication available and in the web console Identify list these appear as 'ID Check $' and 'SMS Auth $'. To

Name	Reqd?	Type	Description
			use ID check in an envelope, the idCheckConfigurationName should be 'ID Check $'. If you wanted to use SMS, it would be 'SMS Auth $' and you would need to add phone number information to the smsAuthentication node.
idCheckInformationInput	No		This complex element contains input information related to a recipient ID check. It can include the following information. addressInformationInput: Used to set recipient address information and consists of: • addressInformation: consists of six elements, with street2 and zipPlus4 being optional. The elements are: street1, street2, city, state, zip, zipPlus4. The maximum number of characters in each element are: street1/street2 = 150 characters, city = 50 characters, state = 2 characters, and zip/zipPlus4 = 20 characters. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. dobInformationInput: Used to set recipient date of birth information and consists of: • dateOfBirth: Specifies the recipient's date, month and year of birth. • displayLevelCode: Specifies the display level for the recipient. Values are:

Name	Reqd?	Type	Description
			ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of: • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn9InformationInput: Used to set the recipient's SSN information. Note that the ssn9 information can never be returned in the response. The ssn9 input consists of: • ssn9: Specifies the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay.
inheritEmailNotificationConfiguration	No	Boolean	Optional element. If true and the recipient creates a DocuSign account after signing, the Manage Account Email Notification settings are used as the default settings for the recipient's account.

Name	Reqd?	Type	Description
note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will display in the signing UI near the upper left corner of the document on the signing screen. This can be a maximum of 1000 characrers.
phoneAuthentication	No	RecipientPhone Authentication	Optional element. Contains the elements: • recipMayProvideNumber – Boolean (if true then recipient can use whatever phone number they want) • senderProvidedNumbers – ArrayOfString (a list of phone numbers the recipient may use) • recordVoicePrint – Reserved • validateRecipProvidedNumber – Reserved
recipientAttachment	No	Attachment	Reserved for DocuSign use.
recipientId	Yes	String	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.
requireIdLookup	No	Boolean	If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate their identity
roleName	No*	String	Optional element. This is the role name associated with the recipient. * - This is required when working with template recipients.
routingOrder	Yes	String	This specifies the routing order of the recipient in the envelope.
samlAuthentication	No	samlAssertionAttributes	Reserved for DocuSign use only.

Name	Reqd?	Type	Description
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: • senderProvidedNumbers – Array with a list of phone numbers the recipient may use for SMS text authentication.
socialAuthentications	No	Boolean	Lists the social ID type that can be used for recipient authentication.
templateAccessCodeRequired	No	Boolean	Optional element. Used only when working with template recipients. If true and TemplateLocked = true, the sender must enter an access code.
templateLocked	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender cannot change any attributes of the recipient.
templateRequired	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender may not remove the recipient.
canEditRecipientEmail	No	Boolean	Optional element only used with recipient types Agents and Editors. If true, the Agent or Editor Recipient associated with this Recipient can change the Recipient's pre-populated Email address. This element is only active if enabled for the account.
canEditRecipientName	No	Boolean	Optional element only used with recipient types Agents and Editors. If true, the Agent or Editor Recipient associated with this recipient can change the recipient's pre-populated name (UserName). This element is only active if enabled for the account.

Name	Reqd?	Type	Description
signingGroupId	No	String	The DocuSign generated Signing Group ID. When used, the email and name parameters are not required for this recipient.

In-Person Signers Recipient Type

Use this action if the signer is in the same physical location as a DocuSign user who will act as a Signing Host for the transaction. The recipient added is the Signing Host and new separate Signer Name field appears after Sign in person is selected. This recipient type can only be used if enabled for your account.

An example In-Person Signers json layout is shown below:

```
"inPersonSigners": [{
    "hostEmail": "signing.host@company.com",
    "hostName": "Mike Host",
    "accessCode": "",
    "addAccessCodeToEmail": false,
    "clientUserId": null,
    "embeddedRecipientStartURL": "string",
    "customFields": {
           "sample string 1",
           "sample string 2"
    },
    "emailNotification": {
           "emailBody":" ",
           "emailSubject":" ",
           "supportedLanguage":" "
    },
    "excludedDocuments": ["2", "4"]
    "idCheckConfigurationName": null,
    "idCheckInformationInput": {
           "addressInformationInput": {
                  "addressInformation": {
                         "street1": "sample string 1",
                         "street2": "sample string 2",
                         "city": "sample string 3",
                         "state": "sample string 4",
                         "zip": "sample string 5",
                         "zipPlus4": "sample string 6"
                  },
                  "displayLevelCode": "sample string 1",
                  "receiveInResponse": "sample string 2"
           },
           "dobInformationInput": {
                  "dateOfBirth": "sample string 1",
                  "displayLevelCode": "sample string 2",
                  "receiveInResponse": "sample string 3"
           },
           "ssn4InformationInput": {
                  "ssn4": "sample string 1",
                  "displayLevelCode": "sample string 2",
                  "receiveInResponse": "sample string 3"
           },
           "ssn9InformationInput": {
                  "ssn9": "sample string 1",
                  "displayLevelCode": "sample string 2"
```

```
            }
    },
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": {
            "recipMayProvideNumber": "sample string 1",
            "validateRecipProvidedNumber": "sample string 2",
            "recordVoicePrint": "sample string 3",
            "senderProvidedNumbers": [
                    "sample string 1",
                    "sample string 2"
            ]
    },
    "recipientAttachment": null,
    "recipientId": "2",
    "requireIdLookup": false,
    "requireSignerCertificate": "string",
    "requireSignOnPaper": "sample string",
    "roleName": "",
    "routingOrder": 1,
    "samlAuthentication": {
            "samlAssertionAttributes": [
                    {
                            "name": "string",
                            "value": "string"
                    },
                    {
                            "name": "string",
                            "value": "string"
                    }
            ]
    },
    "smsAuthentication": {
            "senderProvidedNumbers":[
                    "sample string 1",
                    "sample string 2"
            ]
    },
    "socialAuthentications": null,
    "templateAccessCodeRequired": false,
    "templateLocked": false,
    "templateRequired": false,
    "autoNavigation": false,
    "defaultRecipient": false,
    "signInEachLocation": false,
    "signatureInfo": null,
    "signerEmail": "signer.name@company.com",
    "signerName": "MC Clap Your Hands",
    "tabs": {
            "approveTabs": null,
            "checkboxTabs": null,
            "companyTabs": null,
            "dateSignedTabs": null,
            "dateTabs": null,
            "declineTabs": null,
            "emailTabs": null,
            "envelopeIdTabs": null,
            "fullNameTabs": null,
            "initialHereTabs": null,
            "listTabs": null,
            "noteTabs": null,
            "numberTabs": null,
            "radioGroupTabs": null,
            "signHereTabs": [{
```

```
            "signerAttachmentTabs": null,
            "ssnTabs": null,
            "textTabs": null,
            "titleTabs": null,
            "zipTabs": null
        }
    }],
```

Note for xml there is an additional inPersonSigner level in the layout.

```
<inPersonSigners>
    <inPersonSigner>
            [parameters]
    </inPersonSigner>
</inPersonSigners>
```

The In-Person Signers recipient type parameters are:

Name	Reqd?	Type	Description
hostEmail	Yes	String	Specifies the email for the signing host. This can be a maximum of 100 characters. Required element for In Person Signers recipient Type.
hostName	Yes	String	Specifies the name of the signing host. This can be a maximum of 100 characters. Required element for In Person Signers recipient Type.
accessCode	No	String	This optional element specifies the access code a recipient has to enter to validate the identity. This can be a maximum of 50 characters.
addAccessCodeToEmail	No	Boolean	This optional attribute indicates that the access code will be added to the email sent to the recipient; this nullifies the Security measure of Access Code on the recipient.
clientUserId	No	String	This specifies if the recipient is embedded or remote. If the clientUserId is not null then the recipient is embedded. Note that if a ClientUserId is used and the account settings SignerMustHaveAccount or SignerMustLoginToSign are true, an error is generated on sending.

Name	Reqd?	Type	Description
embeddedRecipientStartURL	No	String	This is a sender provided valid URL string for redirecting an embedded recipient. When using this option, the embedded recipient still receives an email from DocuSign, just as a remote recipient would, but when the document link in the email is clicked the recipient is redirected, through DocuSign, to this URL to complete their actions. When routing to the URL, it is up to the sender's system (the server responding to the URL) to then request a recipient token to launch a signing session. If the value SIGN_AT_DOCUSIGN is used for this node, the recipient will be directed to an embedded signing or viewing process directly at DocuSign. The signing or viewing action is initiated by the DocuSign system and the transaction activity and Certificate of Completion records will reflect this. In all other ways the process is identical to an embedded signing or viewing operation that would be launched by any partner. It is important to remember that in a typical embedded workflow the authentication of an embedded recipient is the responsibility of the sending application and DocuSign expects that senders will follow their own process for establishing the recipient's identity. In this workflow the recipient goes through the sending application before the embedded signing or viewing process in initiated. However, when the sending application sets EmbeddedRecipientStartURL= SIGN_AT_DOCUSIGN, the recipient goes directly to the embedded signing or viewing process bypassing the sending application and any authentication steps the sending application would use. In this case, DocuSign

Name	Reqd?	Type	Description
			recommends that one of the normal DocuSign authentication features (Access Code, Phone Authentication, SMS Authentication, etc.) be used to verify the identity of the recipient. If a clientUserId is NOT provided and a embeddedRecipientStartURL is provided, DocuSign will ignore the redirect URL and launch the standard signing process for the email recipient. Information can be appended to the embeddedRecipientStartURL using merge fields. The available merge fields items are: envelopeId, recipientId, recipientName, recipientEmail, and customFields. The customFields must be part of the recipient or envelope. The merge fields are enclosed in double brackets. *Example:* http://senderHost/[[mergeField1]]/beginSigningSession?[[mergeField2]]&[[mergeField3]]
customFields	No	customField	An optional array of strings that allows the sender to provide custom data about the recipient. This information is returned in the envelope status but otherwise not used by DocuSign. Each customField string can be a maximum of 100 characters.
emailNotification	No	emailNotification	An optional complex type that has information for setting the language for the recipient's email information. It is composed of three elements: • emailBody: a string with the email message sent to the recipient. This can be a maximum of 10000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the simple type enumeration the language

Name	Reqd?	Type	Description
			used. The supported languages, with the language value shown in parenthesis, are: Arabic (ar), Bahasa Indonesia (id), Bahasa Melayu (ms) Bulgarian (bg), Czech (cs), Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Croatian (hr), Danish (da), Dutch (nl), English US (en), English UK (en_GB), Estonian (et), Farsi (fa), Finnish (fi), French (fr), French Canada (fr_CA), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Italian (it), Japanese (ja), Korean (ko), Latvian (lv), Lithuanian (lt), Norwegian (no), Polish (pl), Portuguese (pt), Portuguese Brazil (pt_BR), Romanian (ro),Russian (ru), Serbian (sr), Slovak (sk), Slovenian (sl), Spanish (es),Spanish Latin America (es_MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). **IMPORTANT:** If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.
excludedDocuments	No	String	Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account and the enforceSignerVisibility property must be set to true for the envelope to use this. When enforceSignerVisibility is enabled, documents with tabs can only be viewed by signers that have a tab on that document. Recipients that have an administrative role (Agent, Editor, or Intermediaries) or informational role (Certified Deliveries or Carbon Copies) can always see all the documents in an

Name	Reqd?	Type	Description
			envelope, unless they are specifically excluded using this setting when an envelope is sent. Documents that do not have tabs are always visible to all recipients, unless they are specifically excluded using this setting when an envelope is sent.
idCheckConfigurationName	No	String	Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these name can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting. *Example:* Your account has ID Check and SMS Authentication available and in the web console Identify list these appear as 'ID Check $' and 'SMS Auth $'. To use ID check in an envelope, the idCheckConfigurationName should be 'ID Check $'. If you wanted to use SMS, it would be 'SMS Auth $' and you would need to add phone number information to the smsAuthentication node.
idCheckInformationInput	No		This complex element contains input information related to a recipient ID check. It can include the following information. addressInformationInput: Used to set recipient address information and consists of: • addressInformation: consists of six elements, with street2 and zipPlus4 being optional. The elements are: street1, street2, city, state, zip, zipPlus4. The maximum number of characters in each element are: street1/street2 = 150 characters, city = 50 characters, state = 2 characters, and zip/zipPlus4 =

Name	Reqd?	Type	Description
			20 characters. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. dobInformationInput: Used to set recipient date of birth information and consists of: • dateOfBirth: Specifies the recipient's date, month and year of birth. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of: • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn9InformationInput: Used to set the recipient's SSN information. Note that the ssn9 information can never be returned in the response.

Name	Reqd?	Type	Description
			The ssn9 input consists of: • ssn9: Specifies the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay.
inheritEmailNotificationConfiguration	No	Boolean	Optional element. If true and the recipient creates a DocuSign account after signing, the Manage Account Email Notification settings are used as the default settings for the recipient's account.
note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will display in the signing UI near the upper left corner of the document on the signing screen. This can be a maximum of 1000 characters.
phoneAuthentication	No	Recipient Phone Authentication	Optional element. Contains the elements: • recipMayProvideNumber – Boolean (if true then recipient can use whatever phone number they want) • senderProvidedNumbers – ArrayOfString (a list of phone numbers the recipient may use) • recordVoicePrint – Reserved • validateRecipProvidedNumber – Reserved
recipientAttachment	No	Attachment	Reserved for DocuSign use.
recipientId	Yes	String	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.
requireIdLookup	No	Boolean	If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate their identity

Name	Reqd?	Type	Description
requireSignerCertificate	No	String	Sets the type of signer certificate required for signing. If left blank, no certificate is required. Only one type of certificate can be set for a signer. The possible values are: • docusign_express – Requires a DocuSign Express certificate. • safe – Requires a SAFE-BioPharma certificate. • open_trust – Requires an OpenTrust certificate. **Important:** There are certain rules and restrictions that must be followed when requiring OpenTrust digital signatures. See OpenTrust Rules and Restrictions for more information.
requireSignOnPaper	No	Boolean	When set to 'true', the signer must print, sign, and upload or fax the signed documents to DocuSign.
roleName	No*	String	Optional element. This is the role name associated with the recipient. * - This is required when working with template recipients.
routingOrder	Yes	String	This specifies the routing order of the recipient in the envelope.
samlAuthentication	No	samlAssertion Attributes	Reserved for DocuSign use only.
smsAuthentication	No	senderProvide dNumbers	Optional element. Contains the element: • senderProvidedNumbers – Array with a list of phone numbers the recipient may use for SMS text authentication.
socialAuthentications	No	Boolean	Lists the social ID type that can be used for recipient authentication.
templateAccessCodeRequired	No	Boolean	Optional element. Used only when working with template recipients. If true and TemplateLocked = true, the sender must enter an access code.
templateLocked	No	Boolean	Optional element. Used only when working with template recipients. If

Name	Reqd?	Type	Description
			true, the sender cannot change any attributes of the recipient.
templateRequired	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender may not remove the recipient.
autoNavigation	No	Boolean	Optional element only used with recipient types In Person Signers and Signers. Specifies if the auto navigation setting is on or off for recipient.
defaultRecipient	No	Boolean	If true, this is the default recipient for the envelope. This option is used with the CreateEnvelopeFromTemplatesAn d Forms method.
signInEachLocation	No	Boolean	If set to true and the feature is enabled in the sender's account, then the signing recipient is required to draw signatures and initials at each signature/initial tab (instead of adopting a signature/initial style or only drawing a signature/initial once).
signatureInfo	No	SignatureInfo	Optional element only used with recipient types In Person Signers and Signers. Allows the sender to pre-specify the signature name, signature initials and signature font used in the signature stamp for the recipient.
signerEmail	No	String	Optional element. The email address for an InPersonSigner recipient Type. This can be a maximum of 100 characters.
signerName	Yes	String	The full legal name of a signer for an InPersonSigner recipient Type. This can be a maximum of 100 characters.
tabs	No	Tab	Optional element only used with recipient types In Person Signers and Signers. Specifies the Tabs associated with the recipient. See the Tab Parameters section for

Name	Reqd?	Type	Description
			more information about tabs.

Intermediaries Recipient Type

This recipient can, but is not required to, add name and email information for recipients at the same or subsequent level in the routing order (until subsequent Agents, Editors or Intermediaries recipient types are added). This recipient type can only be used if enabled for your account.

An example Intermediaries json layout is shown below:

```
"intermediaries": [{
    "email": "email.name@company.com",
    "name": "recipient name",
    "accessCode": "",
    "addAccessCodeToEmail": false,
    "clientUserId": null,
    "embeddedRecipientStartURL": "string",
    "customFields": {
           "sample string 1",
           "sample string 2"
    },
    "emailNotification": {
           "emailBody":" ",
           "emailSubject":" ",
           "supportedLanguage":" "
    },
    "excludedDocuments": ["2", "4"]
    "idCheckConfigurationName": null,
    "idCheckInformationInput": {
           "addressInformationInput": {
                  "addressInformation": {
                         "street1": "sample string 1",
                         "street2": "sample string 2",
                         "city": "sample string 3",
                         "state": "sample string 4",
                         "zip": "sample string 5",
                         "zipPlus4": "sample string 6"
                  },
                  "displayLevelCode": "sample string 1",
                  "receiveInResponse": "sample string 2"
           },
           "dobInformationInput": {
                  "dateOfBirth": "sample string 1",
                  "displayLevelCode": "sample string 2",
                  "receiveInResponse": "sample string 3"
           },
           "ssn4InformationInput": {
                  "ssn4": "sample string 1",
                  "displayLevelCode": "sample string 2",
                  "receiveInResponse": "sample string 3"
           },
           "ssn9InformationInput": {
                  "ssn9": "sample string 1",
                  "displayLevelCode": "sample string 2"
           }
    },
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": {
           "recipMayProvideNumber": "sample string 1",
           "validateRecipProvidedNumber": "sample string 2",
```

```
            "recordVoicePrint": "sample string 3",
            "senderProvidedNumbers": [
                   "sample string 1",
                   "sample string 2"
            ]
     },
     "recipientAttachment": null,
     "recipientId": "1",
     "requireIdLookup": false,
     "roleName": "",
     "routingOrder": 1,
     "samlAuthentication": {
            "samlAssertionAttributes": [
                   {
                          "name": "string",
                          "value": "string"
                   },
                   {
                          "name": "string",
                          "value": "string"
                   }
            ]
     },
     "smsAuthentication": {
            "senderProvidedNumbers":[
                   "sample string 1",
                   "sample string 2"
            ]
     },
     "socialAuthentications": null,
     "templateAccessCodeRequired": false,
     "templateLocked": false,
     "templateRequired": false,
     "canEditRecipientEmails": false,
     "canEditRecipientNames": false,
     "signingGroupId": null
}],
```

Note for xml there is an additional intermediary level in the layout.

```
<intermediaries>
    <intermediary>
           [parameters]
    </intermediary>
</intermediaries>
```

The Intermediaries recipient type parameters are:

Name	Reqd?	Type	Description
email	Yes	String	Email of the recipient. Notification will be sent to this email id. This can be a maximum of 100 characters.
name	Yes	String	The full legal name of the recipient. This can be a maximum of 100 characaters.

Name	Reqd?	Type	Description
accessCode	No	String	This optional element specifies the access code a recipient has to enter to validate the identity. This can be a maximum of 50 characters.
addAccessCodeToEmail	No	Boolean	This optional attribute indicates that the access code will be added to the email sent to the recipient; this nullifies the Security measure of Access Code on the recipient.
clientUserId	No	String	This specifies if the recipient is embedded or remote. If the clientUserId is not null then the recipient is embedded. Note that if a ClientUserId is used and the account settings SignerMustHaveAccount or SignerMustLoginToSign are true, an error is generated on sending.
embeddedRecipientStartURL	No	String	This is a sender provided valid URL string for redirecting an embedded recipient. When using this option, the embedded recipient still receives an email from DocuSign, just as a remote recipient would, but when the document link in the email is clicked the recipient is redirected, through DocuSign, to this URL to complete their actions. When routing to the URL, it is up to the sender's system (the server responding to the URL) to then request a recipient token to launch a signing session. If the value SIGN_AT_DOCUSIGN is used for this node, the recipient will be directed to an embedded signing or viewing process directly at DocuSign. The signing or viewing action is initiated by the DocuSign system and the transaction activity and Certificate of Completion records will reflect this. In all other ways the process

Name	Reqd?	Type	Description
			is identical to an embedded signing or viewing operation that would be launched by any partner. It is important to remember that in a typical embedded workflow the authentication of an embedded recipient is the responsibility of the sending application and DocuSign expects that senders will follow their own process for establishing the recipient's identity. In this workflow the recipient goes through the sending application before the embedded signing or viewing process in initiated. However, when the sending application sets EmbeddedRecipientStartURL= SIGN_AT_DOCUSIGN, the recipient goes directly to the embedded signing or viewing process bypassing the sending application and any authentication steps the sending application would use. In this case, DocuSign recommends that one of the normal DocuSign authentication features (Access Code, Phone Authentication, SMS Authentication, etc.) be used to verify the identity of the recipient. If a clientUserId is NOT provided and a embeddedRecipientStartURL is provided, DocuSign will ignore the redirect URL and launch the standard signing process for the email recipient. Information can be appended to the embeddedRecipientStartURL using merge fields. The available merge fields items are: envelopeId, recipientId, recipientName, recipientEmail, and customFields. The customFields must be part of the recipient or envelope. The merge fields are enclosed in double

Name	Reqd?	Type	Description
			brackets. *Example:* http://senderHost/[[mergeField1]]/beginSigningSession?[[mergeField2]]&[[mergeField3]]
customFields	No	customField	An optional array of strings that allows the sender to provide custom data about the recipient. This information is returned in the envelope status but otherwise not used by DocuSign. Each customField string can be a maximum of 100 characters.
emailNotification	No	emailNotification	An optional complex type that has information for setting the language for the recipient's email information. It is composed of three elements: • emailBody: a string with the email message sent to the recipient. This can be a maximum of 10000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the simple type enumeration the language used. The supported languages, with the language value shown in parenthesis, are: Arabic (ar), Bahasa Indonesia (id), Bahasa Melayu (ms) Bulgarian (bg), Czech (cs), Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Croatian (hr), Danish (da), Dutch (nl), English US (en), English UK (en_GB), Estonian (et), Farsi (fa), Finnish (fi), French (fr), French Canada (fr_CA), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Italian (it), Japanese (ja), Korean (ko), Latvian (lv), Lithuanian (lt),

Name	Reqd?	Type	Description
			Norwegian (no), Polish (pl), Portuguese (pt), Portuguese Brazil (pt_BR), Romanian (ro),Russian (ru), Serbian (sr), Slovak (sk), Slovenian (sl), Spanish (es),Spanish Latin America (es_MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). **IMPORTANT:** If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.
excludedDocuments	No	Array of Strings	Specifies the documents that are not visible to this recipient. Document Visibility must be enabled for the account and the enforceSignerVisibility property must be set to true for the envelope to use this. When enforceSignerVisibility is enabled, documents with tabs can only be viewed by signers that have a tab on that document. Recipients that have an administrative role (Agent, Editor, or Intermediaries) or informational role (Certified Deliveries or Carbon Copies) can always see all the documents in an envelope, unless they are specifically excluded using this setting when an envelope is sent. Documents that do not have tabs are always visible to all recipients, unless they are specifically excluded using this setting when an envelope is sent.
idCheckConfigurationName	No	String	Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these name can also be found in the web console sending interface in the Identify list for a recipient). This overrides

Name	Reqd?	Type	Description
			any default authentication setting. *Example:* Your account has ID Check and SMS Authentication available and in the web console Identify list these appear as 'ID Check $' and 'SMS Auth $'. To use ID check in an envelope, the idCheckConfigurationName should be 'ID Check $'. If you wanted to use SMS, it would be 'SMS Auth $' and you would need to add phone number information to the smsAuthentication node.
idCheckInformationInput	No		This complex element contains input information related to a recipient ID check. It can include the following information. addressInformationInput: Used to set recipient address information and consists of: • addressInformation: consists of six elements, with street2 and zipPlus4 being optional. The elements are: street1, street2, city, state, zip, zipPlus4. The maximum number of characters in each element are: street1/street2 = 150 characters, city = 50 characters, state = 2 characters, and zip/zipPlus4 = 20 characters. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. dobInformationInput: Used to set recipient date of birth information and consists of:

Name	Reqd?	Type	Description
			• dateOfBirth: Specifies the recipient's date, month and year of birth. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of: • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn9InformationInput: Used to set the recipient's SSN information. Note that the ssn9 information can never be returned in the response. The ssn9 input consists of: • ssn9: Specifies the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay.

Name	Reqd?	Type	Description
inheritEmailNotificationConfiguration	No	Boolean	Optional element. If true and the recipient creates a DocuSign account after signing, the Manage Account Email Notification settings are used as the default settings for the recipient's account.
note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will display in the signing UI near the upper left corner of the document on the signing screen. This can be a maximum of 1000 characters.
phoneAuthentication	No	Recipient Phone Authentication	Optional element. Contains the elements: • recipMayProvideNumber – Boolean (if true then recipient can use whatever phone number they want) • senderProvidedNumbers – ArrayOfString (a list of phone numbers the recipient may use) • recordVoicePrint – Reserved • validateRecipProvidedNumber – Reserved
recipientAttachment	No	Attachment	Reserved for DocuSign use.
recipientId	Yes	String	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.
requireIdLookup	No	Boolean	If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate their identity
roleName	No*	String	Optional element. This is the role name associated with the recipient. * - This is required when working with template recipients.

Name	Reqd?	Type	Description
routingOrder	Yes	String	This specifies the routing order of the recipient in the envelope.
samlAuthentication	No	samlAssertionAttributes	Reserved for DocuSign use only.
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: • senderProvidedNumbers – Array with a list of phone numbers the recipient may use for SMS text authentication.
socialAuthentications	No	Boolean	Lists the social ID type that can be used for recipient authentication.
templateAccessCodeRequired	No	Boolean	Optional element. Used only when working with template recipients. If true and TemplateLocked = true, the sender must enter an access code.
templateLocked	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender cannot change any attributes of the recipient.
templateRequired	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender may not remove the recipient.
canEditRecipientEmail	No	Boolean	Optional element only used with recipient types Agents and Editors. If true, the Agent or Editor Recipient associated with this Recipient can change the Recipient's pre-populated Email address. This element is only active if enabled for the account.

Name	Reqd?	Type	Description
canEditRecipientName	No	Boolean	Optional element only used with recipient types Agents and Editors. If true, the Agent or Editor Recipient associated with this recipient can change the recipient's pre-populated name (UserName). This element is only active if enabled for the account.
signingGroupId	No	String	The DocuSign generated Signing Group ID. When used, the email and name parameters are not required for this recipient.

Signers Recipient Type

Use this action if your recipient must sign, initial, date or add data to form fields on the documents in the envelope.

Note: The recipientSignatureProviders property can only be used if Standards-Based Signatures are enabled for the account. While this property is available in the production version of the Signature REST API v2, they are still considered to be in beta and subject to change.
The List Signature Providers method can be used to find signature provider information available for an account.

An example Signers json layout is shown below:

```
"signers": [{
    "email": "signer.name@company.com",
    "name": "Mike Signer",
    "isBulkRecipient": "false",
    "bulkRecipientUri": "",
    "accessCode": "",
    "addAccessCodeToEmail": ,
    "clientUserId": ,
    "embeddedRecipientStartURL": "string",
    "customFields": {
           "sample string 1",
           "sample string 2"
    },
    "emailNotification": {
           "emailBody":" ",
           "emailSubject":" ",
           "supportedLanguage":" "
    },
    "excludedDocuments": ["2", "4"]
    "idCheckConfigurationName": ,
    "idCheckInformationInput": {
           "addressInformationInput": {
                  "addressInformation": {
                         "street1": "sample string 1",
                         "street2": "sample string 2",
                         "city": "sample string 3",
                         "state": "sample string 4",
                         "zip": "sample string 5",
```

```
                        "zipPlus4": "sample string 6"
                },
                "displayLevelCode": "sample string 1",
                "receiveInResponse": "sample string 2"
        },
        "dobInformationInput": {
                "dateOfBirth": "sample string 1",
                "displayLevelCode": "sample string 2",
                "receiveInResponse": "sample string 3"
        },
        "ssn4InformationInput": {
                "ssn4": "sample string 1",
                "displayLevelCode": "sample string 2",
                "receiveInResponse": "sample string 3"
        },
        "ssn9InformationInput": {
                "ssn9": "sample string 1",
                "displayLevelCode": "sample string 2"
        }
    },
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": {
        "recipMayProvideNumber": "sample string 1",
        "validateRecipProvidedNumber": "sample string 2",
        "recordVoicePrint": "sample string 3",
        "senderProvidedNumbers": [
                "sample string 1",
                "sample string 2"
        ]
    },
    "recipientAttachment": null,
    "recipientId": "1",
    "requireIdLookup": false,
    "requireSignerCertificate": "string",
    "requireSignOnPaper": "string",
    "roleName": "",
    "routingOrder": 1,
    "samlAuthentication": {
        "samlAssertionAttributes": [
                {
                        "name": "string",
                        "value": "string"
                },
                {
                        "name": "string",
                        "value": "string"
                }
        ]
    },
    "smsAuthentication": {
        "senderProvidedNumbers":[
                "sample string 1",
                "sample string 2"
        ]
    },
    "socialAuthentications": null,
    "templateAccessCodeRequired": false,
    "templateLocked": false,
    "templateRequired": false,
    "autoNavigation": false,
    "defaultRecipient": false,
    "signInEachLocation": false,
    "signatureInfo": null,
```

```
    "tabs": {
           "approveTabs": null,
           "checkboxTabs": null,
           "companyTabs": null,
           "dateSignedTabs": null,
           "dateTabs": null,
           "declineTabs": null,
           "emailTabs": null,
           "envelopeIdTabs": null,
           "fullNameTabs": null,
           "initialHereTabs": null,
           "listTabs": null,
           "noteTabs": null,
           "numberTabs": null,
           "radioGroupTabs": null,
           "signHereTabs": [{
           "signerAttachmentTabs": null,
           "ssnTabs": null,
           "textTabs": null,
           "titleTabs": null,
           "zipTabs": null
    }
    "deliveryMethod":" ",
    "deliveredDateTime":"String Content",
    "signedDateTime":"String Content",
    "offlineAttributes":{
           "deviceName":"String Content",
           "deviceModel":"String Content",
           "gpsLatitude":"String Content",
           "gpsLongitude":"String Content",
           "accountEsignId":"String Content"
    }
    "signingGroupId": null
    "recipientSignatureProviders": [
           {
                  "signatureProviderId": "2",
                  "recipientSignatureProviderOptions": [
                         {
                                "signatureProviderOptionId": "1",
                                "signatureProviderOptionName": "sms",
                                "signatureProviderOptionValue": "+1NNN-NNN-NNNN"
                         }
                  ]
           }
    ]
}],

--[Multipart boundary (example of adding a Recipient Photo)]
Content-Type: image/jpeg
Content-Disposition: inline;recipientId=1
Content-Transfer-Encoding: base64

--[Multipart boundary (example of adding a Tab Image)]
Content-Type: image/gif
Content-Disposition: inline;tabId=1
Content-Transfer-Encoding: base64

--[Multipart boundary]
```

Note for xml there is an additional signer level in the layout.

```
<signers>
    <signer>
```

```
            [parameters]
    </signer>
</signers>
```

The Signers recipient type parameters are:

Name	Reqd?	Type	Description
email	Yes	String	Email of the recipient. Notification will be sent to this email id. This can be a maximum of 100 characters.
name	Yes	String	The full legal name of the recipient. This can be a maximum of 100 characaters.
isBulkRecipient	No	Boolean	When true, this signer is a bulk recipient and the recipient information is contained in a bulk recipient file. Note that when this is true the email and name for the recipient becomes bulk@recipient.com and “Bulk Recipient”. These fields may not be changed for the bulk recipient.
bulkRecipientsUri	No	String	The URL for the bulk recipient file with the bulk recipient information for this envelope. This information is only shown after the bulk recipient file is uploaded (PUT) to the envelope.
accessCode	No	String	This optional element specifies the access code a recipient has to enter to validate the identity. This can be a maximum of 50 characters.
addAccessCodeToEmail	No	Boolean	This optional attribute indicates that the access code will be added to the email sent to the recipient; this nullifies the Security measure of Access Code on the recipient.
clientUserId	No	String	This specifies if the recipient is embedded or remote. If the clientUserId is not null then the recipient is embedded. Note that if a ClientUserId is used and the account settings SignerMustHaveAccount or

Name	Reqd?	Type	Description
			SignerMustLoginToSign are true, an error is generated on sending.
embeddedRecipientStartURL	No	String	This is a sender provided valid URL string for redirecting an embedded recipient. When using this option, the embedded recipient still receives an email from DocuSign, just as a remote recipient would, but when the document link in the email is clicked the recipient is redirected, through DocuSign, to this URL to complete their actions. When routing to the URL, it is up to the sender's system (the server responding to the URL) to then request a recipient token to launch a signing session. If the value SIGN_AT_DOCUSIGN is used for this node, the recipient will be directed to an embedded signing or viewing process directly at DocuSign. The signing or viewing action is initiated by the DocuSign system and the transaction activity and Certificate of Completion records will reflect this. In all other ways the process is identical to an embedded signing or viewing operation that would be launched by any partner. It is important to remember that in a typical embedded workflow the authentication of an embedded recipient is the responsibility of the sending application and DocuSign expects that senders will follow their own process for establishing the recipient's identity. In this workflow the recipient goes through the sending application before the embedded signing or viewing process in initiated. However, when the sending application sets EmbeddedRecipientStartURL=SIGN_AT_DOCUSIGN, the recipient goes directly to the embedded signing or viewing process bypassing the sending application and any authentication steps the

Name	Reqd?	Type	Description
			sending application would use. In this case, DocuSign recommends that one of the normal DocuSign authentication features (Access Code, Phone Authentication, SMS Authentication, etc.) be used to verify the identity of the recipient. If a clientUserId is NOT provided and a embeddedRecipientStartURL is provided, DocuSign will ignore the redirect URL and launch the standard signing process for the email recipient. Information can be appended to the embeddedRecipientStartURL using merge fields. The available merge fields items are: envelopeId, recipientId, recipientName, recipientEmail, and customFields. The customFields must be part of the recipient or envelope. The merge fields are enclosed in double brackets. *Example:* http://senderHost/[[mergeField1]]/beginSigningSession?[[mergeField2]]&[[mergeField3]]
customFields	No	customField	An optional array of strings that allows the sender to provide custom data about the recipient. This information is returned in the envelope status but otherwise not used by DocuSign. Each customField string can be a maximum of 100 characters.
emailNotification	No	emailNotification	An optional complex type that has information for setting the language for the recipient's email information. It is composed of three elements: • emailBody: a string with the email message sent to the recipient. This can be a maximum of 10000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the simple

Name	Reqd?	Type	Description
			type enumeration the language used. The supported languages, with the language value shown in parenthesis, are: Arabic (ar), Bahasa Indonesia (id), Bahasa Melayu (ms) Bulgarian (bg), Czech (cs), Chinese Simplified (zh_CN), Chinese Traditional (zh_TW), Croatian (hr), Danish (da), Dutch (nl), English US (en), English UK (en_GB), Estonian (et), Farsi (fa), Finnish (fi), French (fr), French Canada (fr_CA), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Italian (it), Japanese (ja), Korean (ko), Latvian (lv), Lithuanian (lt), Norwegian (no), Polish (pl), Portuguese (pt), Portuguese Brazil (pt_BR), Romanian (ro),Russian (ru), Serbian (sr), Slovak (sk), Slovenian (sl), Spanish (es),Spanish Latin America (es_MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). **IMPORTANT:** If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.
excludedDocuments	No	Array of Strings	Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account and the enforceSignerVisibility property must be set to true for the envelope to use this. When enforceSignerVisibility is enabled, documents with tabs can only be viewed by signers that have a tab on that document. Recipients that have an administrative role (Agent, Editor, or Intermediaries) or informational role (Certified Deliveries or Carbon Copies) can

Name	Reqd?	Type	Description
			always see all the documents in an envelope, unless they are specifically excluded using this setting when an envelope is sent. Documents that do not have tabs are always visible to all recipients, unless they are specifically excluded using this setting when an envelope is sent.
idCheckConfigurationName	No	String	Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these name can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting. *Example:* Your account has ID Check and SMS Authentication available and in the web console Identify list these appear as 'ID Check $' and 'SMS Auth $'. To use ID check in an envelope, the idCheckConfigurationName should be 'ID Check $'. If you wanted to use SMS, it would be 'SMS Auth $' and you would need to add phone number information to the smsAuthentication node.
idCheckInformationInput	No		This complex element contains input information related to a recipient ID check. It can include the following information. addressInformationInput: Used to set recipient address information and consists of: • addressInformation: consists of six elements, with street2 and zipPlus4 being optional. The elements are: street1, street2, city, state, zip, zipPlus4. The maximum number of characters in each element are: street1/street2 = 150 characters, city = 50 characters, state = 2

Name	Reqd?	Type	Description
			characters, and zip/zipPlus4 = 20 characters. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. dobInformationInput: Used to set recipient date of birth information and consists of: • dateOfBirth: Specifies the recipient's date, month and year of birth. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of: • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay. • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. ssn9InformationInput: Used to set the recipient's SSN information. Note that the ssn9 information can

Name	Reqd?	Type	Description
			never be returned in the response. The ssn9 input consists of: • ssn9: Specifies the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay.
inheritEmailNotificationConfiguration	No	Boolean	Optional element. If true and the recipient creates a DocuSign account after signing, the Manage Account Email Notification settings are used as the default settings for the recipient's account.
note	No	String	A note that will be unique to this recipient. This note will be sent to the recipient via the signing email. This note will display in the signing UI near the upper left corner of the document on the signing screen. This can be a maximum of 1000 characters.
phoneAuthentication	No	Recipient Phone Authentication	Optional element. Contains the elements: • recipMayProvideNumber – Boolean (if true then recipient can use whatever phone number they want) • senderProvidedNumbers – ArrayOfString (a list of phone numbers the recipient may use) • recordVoicePrint – Reserved • validateRecipProvidedNumber – Reserved
recipientAttachment	No	Attachment	Reserved for DocuSign use.
recipientId	Yes	String	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.
requireIdLookup	No	Boolean	If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate

Name	Reqd?	Type	Description
			their identity
requireSignerCertificate	No	String	Sets the type of signer certificate required for signing. If left blank, no certificate is required. Only one type of certificate can be set for a signer. The possible values are: • docusign_express – Requires a DocuSign Express certificate. • safe – Requires a SAFE-BioPharma certificate. • open_trust – Requires an OpenTrust certificate. **Important:** There are certain rules and restrictions that must be followed when requiring OpenTrust digital signatures. See OpenTrust Rules and Restrictions for more information.
requireSignOnPaper	No	Boolean	When set to 'true', the signer must print, sign, and upload or fax the signed documents to DocuSign.
roleName	No*	String	Optional element. This is the role name associated with the recipient. * - This is required when working with template recipients.
routingOrder	Yes	String	This specifies the routing order of the recipient in the envelope.
samlAuthentication	No	samlAssertion Attributes	Reserved for DocuSign use only.
smsAuthentication	No	senderProvide dNumbers	Optional element. Contains the element: • senderProvidedNumbers – Array with a list of phone numbers the recipient may use for SMS text authentication.
socialAuthentications	No	Boolean	Lists the social ID type that can be used for recipient authentication.
templateAccessCodeRequired	No	Boolean	Optional element. Used only when working with template recipients. If true and TemplateLocked = true, the sender must enter an access code.

Name	Reqd?	Type	Description
templateLocked	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender cannot change any attributes of the recipient.
templateRequired	No	Boolean	Optional element. Used only when working with template recipients. If true, the sender may not remove the recipient.
autoNavigation	No	Boolean	Optional element only used with recipient types In Person Signers and Signers. Specifies if the auto navigation setting is on or off for recipient.
defaultRecipient	No	Boolean	If true, this is the default recipient for the envelope.
signInEachLocation	No	Boolean	If set to true and the feature is enabled in the sender's account, then the signing recipient is required to draw signatures and initials at each signature/initial tab (instead of adopting a signature/initial style or only drawing a signature/initial once).
signatureInfo	No	SignatureInfo	Optional element only used with recipient types In Person Signers and Signers. Allows the sender to pre-specify the signature name, signature initials and signature font used in the signature stamp for the recipient.
tabs	No	Tab	Optional element only used with recipient types In Person Signers and Signers. Specifies the Tabs associated with the recipient. See the Tab Parameters section for more information about tabs.
deliveryMethod	No	String	Reserved for DocuSign use.
deliveredDateTime	No	DateTime	Reserved for DocuSign use.
signedDateTime	No*	DateTime	Reserved for DocuSign use.
offlineAttributes	No		Reserved for DocuSign use.
signingGroupId	No	String	The DocuSign generated Signing Group ID.

Name	Reqd?	Type	Description
			When used, the email and name parameters are not required for this recipient.
recipientSignatureProviders	No		This complex element sets the DocuSign Standards-Based Signature provider information for the signer. It is made up of the following properties: • signatureProviderId: The signature provider ID from the DocuSign system. Requests can use the signatureProviderId or signatureProviderName or both. • signatureProviderName: The name of the signature provider from the DocuSign system. Requests can use the signatureProviderId or signatureProviderName or both. • recipientSignatureProviderOptions: A list of signature provider options. If the signature provider requires an option, then this is required. It consists of the following information: signatureProviderOptionId: The ID of the signature provider option. signatureProviderOptionName: The system name of the signature provider option. signatureProviderOptionValue: The value for the signature provider option. This is a required value.

OpenTrust Rules and Restrictions

The DocuSign API enforces certain rules and restrictions when at least one signer or in person signer is required to use an OpenTrust signer certificate (requireSignerCertificate=open_trust).

The following restrictions apply when sending an envelope:

- Sign on Paper, Document Markup, Field Markup, watermarks, and Electronic Notary cannot be enabled for the envelope.

- A SMS Authentication or Access Code authentication check must be included for the OpenTrust signer.
- All recipients must be the only recipient in their routing order.
- The OpenTrust signer must have at least one required Signature or Initials tag in the envelope (the envelope cannot be sent for Free-Form signing). Additionally, the signer cannot have more than 5 OpenTrust Signature tabs in the envelope.
- When a signer in the envelope is required to apply a digital signature, then all subsequent signers that have Signature or Initial tags are required to use a digital signature. Note that other recipient types can be used and subsequent signers can use other tab types.
- Downloading documents individually or as a combined PDF is allowed. However, downloading the combined PDF of all documents might cause the digital signatures to appear invalid due to the process of combining the documents. This is a known issue and each document can be downloaded individually to verify the digital signature.

Other restrictions that apply to envelopes:

- Envelopes cannot be corrected if at least one OpenTrust signer has signed the envelope.

Using Signing Groups in Envelopes and Templates

Signing groups can be used in the REST API by adding the Signing Group ID to the appropriate recipient in an envelope or template.

When a Signing Group ID is used for a recipient, the normally required recipient name and email information is not required. Instead the signing group members are used in place of the normal recipient and any email communications are sent to the signing group members.

There are some restrictions associated with using Signing Groups:

- A Signing Group cannot be used as In Person Signing host.
- When a Signing Group is used and phone authentication is specified for that recipient, then the recipient may provide number option must be enabled (phoneAuthentication – recipMayProvideNumber set to true).
- DocuSign does not recommend that SMS authentication be used with Signing Groups, since the recipient's SMS phone number must be entered before sending and the group members probably aren't sharing the same phone number.
- Signing Groups cannot use the Fax out for Signature feature.
- If a Signing Group is used as an Agent, Editor, or Intermediary recipient and the recipient is not a captive recipient, then Signing Group members that have DocuSign accounts can enter a Signing Group for the role Name for the envelope recipients.

Tab Types and Parameters

Tabs are used to indicate locations in a document where the recipient should take some action. Tabs are associated with a specific recipient and are only used by the recipient types In Person Signers and Signers. There are 24 different tab types.

Tab types share some common parameters, but the exact parameters associated with a tab depend on the tab type. Tabs are associated with a specific recipient and are only used by the recipient types In Person Signers and Signers. Refer to the specific tab type below for more information.

Approve	Checkbox	Company	Date Signed
Date	Decline	Email	Email Address
Envelope ID	First Name	Formula	Full Name
Initial Here	Last Name	List	Note
Number	Radio Group	Sign Here	Signer Attachment
SSN	Text	Title	Zip

Anchoring Tabs

The tab anchoring option allows you to send documents for signature that do not have a fixed layout or format. In these documents, you might not know the absolute location of the tabs when you design your API client application because the tabs must move with text. As an alternative to sending X and Y coordinates for tabs, the DocuSign Service can derive an anchor location for the tab by correlating anchor information to data within the document.

When the DocuSign Service receives a request that contains tabs with anchor information, it searches the document for instances of the anchorString. When found, it places a tab of the specified type for the designated recipient. Tab have a default positioning, but this can be changed using the offset settings for the tab.

When you apply tabs to the document, DocuSign does not remove or replace the anchorString text. You can hide codified anchors by using the same font color as the background of the document. So the anchor can be used by DocuSign processes and it will not be visible on the document.

To use an anchoring option:

1. Identify the location in the document by text string. You can use a pre-existing text string or add a new one.

   For best performance DocuSign recommends using single word anchor strings when possible, especially when there are a large number of pages in the envelope.

   For example, you might want to add a Sign Here tab to the “Borrower’s Signature” lines in a document, but that phrase might occur in places in the document where you don't want to tab to appear. In this case, you could add the text “BorrowerSignHere” in white font color (so that isn't visible in the document) to all the places you want Sign Here tabs to appear and use "BorrowerSignHere" as the anchor string.

2. Reference the anchor through the anchorString element of the tab.

3. Determine the offset from the anchorString location to where the tab should be placed.

   Positive anchorXOffset values move the tab right on the page and positive anchorYoffset values move the tab down the page. The anchorUnits specifies the units used for the offsets and the default unit is pixels.

   For Sign Here and Initial Here tabs the bottom-left of the anchorString is equivalent to position (0,0), and the bottom-left of the tab graphic is placed relative to that.

   For all other tabs the bottom-left of the anchorString is equivalent to position (0,0), and the top-left of the tab graphic is placed relative to that.

   DocuSign does not currently provide tools to derive the offset values. Determination of the proper offset will likely require some trial-and-error.

Rules for working with anchor tags

Note: When anchor tabs are used, all documents in the envelope are searched for the anchorString.

- You set the text of the anchorString. DocuSign tabs are created for each instance of the anchorString within the document, so special care must be taken to establish unique anchorStrings that do not result in unintentional tabs.
- You cannot use the same anchored tab for different recipients for the same document.
- The DocuSign system cannot search for text that is embedded in an image when checking for anchorStrings.
- X or Y offsets supplied for a tab apply to all instances of the tab in the document. To use different offsets at different locations in the document for the same recipient, create multiple, unique anchor tabs.
- If an anchorString Y offset value would force a tab outside of the page boundaries, the tag will be placed at the page boundary. If the X offset value places a tab outside of the page boundaries, the error message "Invalid_User_Offset" is sent. The error message includes the X offset that resulted in the error.
- The system does not support an anchorString embedded in the form of a PDF X-object in the document.
- The system does not re-flow the text that surrounds the anchor tabs. It is the responsibility of the document author to provide sufficient whitespace to contain the potential width of the ultimate tab value.

Tips and Tricks

The following are tips for effective use of anchor tags:

- In order to avoid unintentional conflicts between anchorString text and the text that naturally exists in documents, establish a codified syntax for the anchorString text that is unlikely to appear elsewhere in a document.
- Develop an extensible and consistent syntax that can be used across multiple document types.
- Especially for documents that have variable numbers of tabs or signers, author the source document to include hidden anchor tabs for all potential signers/permutations. Then, control the tabs that are actually placed by including/excluding the anchor tabs in the request. This approach allows a single document to be used for all use cases instead of maintaining separate documents for each scenario.

Automatically Populating Tabs

If you want similar tab types to automatically populate with the same data, you must follow these guidelines:

- Each tabLabel entry must have the characters "*" added in front of the actual tab label. If the "*" characters are not added, then only the first occurrence of the tab is populated.

  Important: When automatically populating tabs, the tabLabel must not contain any spaces.

Example: In the JSON example below, the Text tabs “StudentLastName” and “StudentFirstName” will be auto-populated as specified (“Doe” and “John”) each place they appear throughout the envelope.

```
"tabs": {
       "textTabs": [
       {
              "tabLabel": "\\*StudentLastName",
              "value": "Doe"
       },
       {
              "tabLabel": "\\*StudentFirstName",
              "value": "John"
       }]
}
```

- Each occurence of the tab must have identical properties.

  Example: There are two Text tabs in a document, each with tabLabel = Name, but for the first tab has the property bold = true and the second tab has bold = false. When creating the envelope, only the first tab is populated. In order to automatically populate both occurrences of the "Name" Text tabs, the “bold” property must be set to the same value for both tabs.

Using Custom Tabs in Envelopes and Templates

Custom Tabs can be added to envelopes and templates by setting the customTabId when creating an envelope or template recipient or when adding a new tab for an existing recipient. The custom tab must be added as the correct tab type. For example if the custom tab type is text, it cannot be used as a number tab.

When a customTabId is used, the new tab inherits all the custom tab properties. Required information that is not included in the custom tab, such as document ID and page ID, must be included when adding the tab. If the custom tab does not have anchor settings, the X and Y positions must be included.

After the tab is created, it is treated as any other tab for updating or deleting.

Approve Tab

Place this tag on the document where you want the recipient to approve documents in an envelope without placing a signature or initials on the document. If the recipient clicks the Approve tag during the signing process, the recipient is considered to have signed the document. No information is shown on the document for the approval, but it is recorded as a signature in the envelope history.

An example Approve tab json layout is shown below:

```
"approveTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
```

```
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "300",
    "yPosition": "100",
    "bold": false,
    "font": "lucidaconsole",
    "fontColor": null,
    "fontSize": "size9",
    "italic": false,
    "tabLabel": "Approve 11",
    "underline": false,
    "buttonText": "Approve",
    "height": 18,
    "width": 60,
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional approve level in the layout.

```
<approveTabs>
    <approve>
           [parameters]
    </approve>
</approveTabs>
```

The Approve tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.

Name	Reqd?	Type	Description
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.

Name	Reqd?	Type	Description
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value.
underline	No	Boolean	If true, the information in the tab is underlined.
buttonText	No	String	The text displayed in the tab. Only used in approveTabs and decineTabs.
height	No	String	Height of the tab in pixels.
width	No	String	Width of the tab in pixels.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Checkbox Tab

Place this tag on the document in a location where the recipient can select an option.

An example Checkbox tab json layout is shown below:

```
"checkboxTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
```

```
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "144",
    "yPosition": "155",
    "name": "Checkbox",
    "requireInitialOnSharedChange": false,
    "selected": false,
    "shared": false,
    "tabLabel": "Check Box 14",
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional checkbox level in the layout.

```
<checkboxTabs>
    <checkbox>
           [parameters]
    </checkbox>
</checkboxTabs>
```

The Checkbox tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.

Name	Reqd?	Type	Description
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
name	No	String	Sets the tool tip text for the tab.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.

Name	Reqd?	Type	Description
selected	No	Boolean	If true, the checkbox is selected.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Company Tab

Place this tag on the document where you want the recipient's company name to appear.

Note: When getting information that includes this tab type, the original value of the tab when the associated envelope was sent is included in the response.

An example Company tab json layout is shown below:

```
"companyTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "maxLength": null,
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "394",
    "yPosition": "65",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Company",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": false,
    "name": "Company",
    "required": true,
    "value": "Ro Corp 2",
    "width": 42,
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional company level in the layout.

```
<companyTabs>
    <company>
           [parameters]
    </company>
</companyTabs>
```

The Company tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
maxLength	No	String	Sets the maximum number of characters a signer can enter in the tag.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.

Name	Reqd?	Type	Description
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.

Name	Reqd?	Type	Description
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	Sets the tool tip text for the tab.
Required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Date Signed Tab

Place this tag on the document where you want the date the recipient signed the document to appear.

An example Date Signed tab json layout is shown below:

```
"dateSignedTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
```

```
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "147",
    "yPosition": "103",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Date Signed",
    "underline": false,
    "name": "Date Signed",
    "value": ""
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional dateSigned level in the layout.

```
<dateSignedTabs>
    <dateSigned>
           [parameters]
    </dateSigned>
</dateSignedTabs>
```

The Date Signed tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.

Name	Reqd?	Type	Description
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.

Name	Reqd?	Type	Description
Font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value.
underline	No	Boolean	If true, the information in the tab is underlined.
name	No	String	Sets the tool tip text for the tab.
value	No	String	This element specifies the value of the tab.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Date Tab

Place this tag on the document where you want the recipient to enter a date. Date tabs are single-line fields that allow date information to be entered in any format. The tooltip for this tab recommends entering the date as MM/DD/YYYY, but this is not enforced. The format entered by the signer is retained. If you need a particular date format enforced, DocuSign recommends using a Regex Pattern and Validation Error to enforce the format.

Note: When getting information that includes this tab type, the original value of the tab when the associated envelope was sent is included in the response.

An example Date tab json layout is shown below:

```
"dateTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
```

```
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "maxLength": null,
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Date Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "requireAll": false,
    "requireInitialOnSharedChange": false,
    "shared": false,
    "height": 11,
    "validationMessage": ""
    "validationPattern": ""
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional date level in the layout.

```
<dateTabs>
    <date>
           [parameters]
    </date>
</dateTabs>
```

The Date tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.

Name	Reqd?	Type	Description
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
maxLength	No	String	Sets the maximum number of characters a signer can enter in the tag.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.

Name	Reqd?	Type	Description
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.

Name	Reqd?	Type	Description
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	Sets the tool tip text for the tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab in pixels.
requireAll	No	Boolean	Optional element associated with *shared* element (field markup). When true and *shared* is true, information must be entered in this field to complete the envelope.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
height	No	String	Height of the tab in pixels.

Name	Reqd?	Type	Description
validationMessage	No	String	Message to be displayed to the signer if the validation pattern fails. This is optional and if not provided the default DocuSign message is displayed. This can be a maximum of 250 characters.
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.

Name	Reqd?	Type	Description
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Decline Tab

Place this tag on the document where you want to give the recipient the option of declining an envelope. If the recipient clicks the Decline tag during the signing process, the envelope is voided.

An example Decline tab json layout is shown below:

```
"declineTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "408",
    "yPosition": "105",
    "bold": false,
    "font": "lucidaconsole",
    "fontColor": null,
    "fontSize": "size9",
    "italic": false,
    "tabLabel": "Decline 12",
    "underline": false,
    "buttonText": "Decline",
    "height": 18,
    "width": 60,
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional decline level in the layout.

```
<declineTabs>
    <decline>
           [parameters]
```

```
    </decline>
</declineTabs>
```

The Decline tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.

Name	Reqd?	Type	Description
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.

Name	Reqd?	Type	Description
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value.
underline	No	Boolean	If true, the information in the tab is underlined.
buttonText	No	String	The text displayed in the tab. Only used in approveTabs and decineTabs.
height	No	String	Height of the tab in pixels.
width	No	String	Width of the tab in pixels.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Email Tab

Place this tag on the document where you want the recipient to enter an email. Email tags are single-line fields that accept any characters. The system checks that a valid email format (i.e. xxx@yyy.zzz) is entered in the tag. It uses the same parameters as a Text tab, with the validation message and pattern set for email information.

Note: When getting information that includes this tab type, the original value of the tab when the associated envelope was sent is included in the response.

An example Email tab json layout is shown below:

```
"emailTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "maxLength": null,
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Email Field 3",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "requireAll": false,
    "requireInitialOnSharedChange": false,
    "shared": false,
    "height": 11,
    "validationMessage": ""
    "validationPattern": ""
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional email level in the layout.

```
<emailTabs>
    <email>
           [parameters]
    </email>
```

```
</emailTabs>
```

The Email tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
maxLength	No	String	Sets the maximum number of characters a signer can enter in the tag.

Name	Reqd?	Type	Description
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.

Name	Reqd?	Type	Description
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	Sets the tool tip text for the tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab in pixels.

Name	Reqd?	Type	Description
requireAll	No	Boolean	Optional element associated with *shared* element (field markup). When true and *shared* is true, information must be entered in this field to complete the envelope.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
height	No	String	Height of the tab in pixels.
validationMessage	No	String	Message to be displayed to the signer if the validation rule fails. This is optional and if not provided the default DocuSign message is displayed. This can be a maximum of 250 characters.
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Email Address Tab

Place this tag on a document where you want the recipient's email, as entered in the recipient information, to appear.

An example Email Address tab json layout is shown below:

```
"emailAddressTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
```

```
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "399",
    "yPosition": "35",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Email Address",
    "underline": false,
    "name": " Email Address"
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional emailAddress level in the layout.

```
<emailAddressTabs>
    <emailAddress>
           [parameters]
    </emailAddress>
</emailAddressTabs>
```

The Email Address tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.

Name	Reqd?	Type	Description
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.

Name	Reqd?	Type	Description
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value.
underline	No	Boolean	If true, the information in the tab is underlined.
name	No	String	Sets the tool tip text for the tab.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Envelope ID Tab

Place this tag on the document where you want the envelope ID for to appear. Recipients cannot enter or change the information in this tab, it is for informational purposes only.

An example Envelope ID tab json layout is shown below:

```
"envelopeIdTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "customTabId": null,
    "documentId": "1",
```

```
    "pageNumber": "1",
    "xPosition": "24",
    "yPosition": "153",
    "tabLabel": "Envelope ID,
    "mergeField": {},
}],
```

Note for xml there is an additional envelopeId level in the layout.

```
<envelopeIdTabs>
    <envelopeId>
          [parameters]
    </envelopeId>
</envelopeIdTabs>
```

The Envelope ID tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.

Name	Reqd?	Type	Description
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
tabLabel	Yes	String	The label used with the tab. This can be a maximum of 500 characters.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.

First Name Tab

Place this tag on a document where you want the recipient's first name to appear. This tag takes the recipient's name, as entered in the recipient information, splits it into sections based on spaces and uses the first section as the first name.

An example First Name tab json layout is shown below:

```
"firstNameTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
```

```
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "399",
    "yPosition": "35",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "First Name",
    "underline": false,
    "name": "First Name"
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional firstName level in the layout.

```
<firstNameTabs>
    <firstName>
           [parameters]
    </firstName>
</firstNameTabs>
```

The First Name tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.

Name	Reqd?	Type	Description
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.

Name	Reqd?	Type	Description
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
name	No	String	Sets the tool tip text for the tab.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Formula Tab

This tag is used to add a calculated field to a document. Envelope recipients cannot directly enter information into the tag; the formula tab calculates and displays a new value when changes are made to the reference tag values. The reference tag information and calculation operations are entered in the “formula” element. See the *Using the Calculated Fields Feature* quick start guide or *DocuSign Service User Guide* for more information about formulas.

An example Formula tab json layout is shown below:

```
"formulaTabs": [{
    "formula": " [Line1]+[Line2]+[Tax] ",
    "isPaymentAmount": false
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
```

```
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "roundDecimalPlaces": "2",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Data Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": true,
    "name": "Formula",
    "required": true,
    "value": "",
    "width": 42,
    "height": 11,
    "validationMessage": "",
    "validationPattern": "",
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional formulaTab level in the layout.

```
<formulaTabs>
    <formulaTab>
           [parameters]
    </formulaTab>
</formulaTabs>
```

The Formula tab parameters are:

Name	Reqd?	Type	Description
formula	No	String	The Formula string contains the TabLabel for the reference tabs used in the formula and calculation operators. Each TabLabel must be contained in brackets. This can be a maximum of 2000 characters. **Example:** Three tabs (TabLabels: Line1, Line2, and Tax) need to be added together. The formula string would be: [Line1]+[Line2]+[Tax]

Name	Reqd?	Type	Description
isPaymentAmount	No	Boolean	When true, sets this as a payment tab. Can only be used with Text, Number, Formula or List tabs. The value of the tab must be a number.
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
roundDecimalPlaces	No	String	This sets the number of digits after the decimal point in formula tab. The only accepted values are 0 or 2.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.

Name	Reqd?	Type	Description
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.
locked	No	Boolean	This should always be set to true for formula tabs.
name	No	String	Sets the tool tip text for the tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab in pixels.
height	No	String	Height of the tab in pixels.
validationMessage	No	String	Message to be displayed to the signer if the validation rule fails. This is optional and if not provided the default DocuSign message is displayed. This can be a maximum of 250 characters.
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Full Name Tab

Place this tag on the document where you want the recipient's name to appear.

An example Full Name tab json layout is shown below:

```
"fullNameTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "399",
    "yPosition": "35",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Full Name",
    "underline": false,
    "name": "Full Name",
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional fullName level in the layout.

```
<fullNameTabs>
    <fullName>
           [parameters]
    </fullName>
</fullNameTabs>
```

The Full Name tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.

Name	Reqd?	Type	Description
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.

Name	Reqd?	Type	Description
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
name	No	String	Sets the tool tip text for the tab.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Initial Here Tab

Place this tag to have a recipient place their initials in the document. The “optional” parameter sets if the initials are required or optional.

An example Initial Here tab json layout is shown below:

```
"initialHereTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
```

```
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "249",
    "yPosition": "15",
    "name": "Initial Here",
    "optional": false,
    "scaleValue": 1,
    "tabLabel": "Initial 5"
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional initialHere level in the layout.

```
<initialHereTabs>
    <initialHere>
           [parameters]
    </initalHere>
</initalHereTabs>
```

The Initial Here tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.

Name	Reqd?	Type	Description
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
name	No	String	Sets the tool tip text for the tab.
optional	No	Boolean	When true, the Initial Here tab is optional and the recipient is not required to add initials to complete an envelope.

Name	Reqd?	Type	Description
scaleValue	No	String	Sets the size for Initial Here tab. It can be value from 0.5 to 1.0, where 1.0 represents full size and 0.5 is 50% size. When changing the size the upper left corner of the tab is the origin point for the tab, so changing the size causes the tab to shrink up and to the left. Note that this is different than the scale setting in the Classic DocuSign Experience web application where the tab origin is the lower left corner.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.

Name	Reqd?	Type	Description
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Last Name Tab

Place this tag on a document where you want the recipient's last name to appear. This tag takes the recipient's name, as entered in the recipient information, splits it into sections based on spaces and uses the last section as the last name.

An example Last Name tab json layout is shown below:

```
"lastNameTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "399",
    "yPosition": "35",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Last Name",
    "underline": false,
    "name": "Last Name",
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional lastName level in the layout.

```
<lastNameTabs>
    <lastName>
           [parameters]
    </lastName>
```

```
</lastNameTabs>
```

The Last Name tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.

Name	Reqd?	Type	Description
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.

Name	Reqd?	Type	Description
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
name	No	String	Sets the tool tip text for the tab.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

List Tab

Use this tag to give your recipient a list of options, presented as a drop-down list, from which they can select.

An example List tab json layout is shown below:

```
"listTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "286",
    "yPosition": "163",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Drop Down 16",
    "underline": false,
    "listItems": [{
           "selected": false,
           "text": "Peach",
           "value": "peachValue"
           },
    {
           "selected": true,
           "text": "Pear",
           "value": "PearValue"
    },
    {
           "selected": false,
           "text": "Dog",
           "value": "DogValue"
    }],
    "locked": false,
    "mergeFieldXml": "",
    "required": false,
    "requireAll": false,
    "requireInitialOnSharedChange": false,
    "senderRequired": false,
    "shared": false,

    "value": "Pear",
    "width": 48,
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional list level in the layout.

```
<listTabs>
    <list>
            [parameters]
    </list>
</listTabs>
```

The List tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.

Name	Reqd?	Type	Description
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.

Name	Reqd?	Type	Description
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
listItems	Yes	listItem	Only used with list tabs. These are the possible selections for a dropdown list. Each selection has three parts: • selected: Sets if this is the default selection shown to a signer. Use true/false to show the value is selected or not. Only one selection can be true. • text: The text shown in the dropdown list. • value: The value used when the selected.
locked	No	Boolean	If true, the signer cannot change information in this tab.
required	No	Boolean	If true, the signer is required to select an option for this tab.
requireAll	No	Boolean	Optional element associated with *shared* element (field markup). When true and *shared* is true, information must be entered in this field to complete the envelope.

Name	Reqd?	Type	Description
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.
senderRequired	No	Boolean	When true for a tab in a template, the sender must populate the tab before an envelope can be sent using the template. This value tab can only be changed by modifying (PUT) the template. Tabs with a senderRequired value of true cannot be deleted from an envelope.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
value	No	String	This sets/gets the currently selected list value.
width	No	String	Width of the tab.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.

Name	Reqd?	Type	Description
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Note Tab

Place this tag on the document where you want to place additional information, in the form of a note, on a document for a recipient.

An example Note tab json layout is shown below:

```
"noteTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "name": "Note1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "384",
    "yPosition": "154",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Note 17",
    "underline": false,
    "height": 33,
    "shared": false,
    "value": "This is a test note",
    "width": 66,
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional note level in the layout.

```
<noteTabs>
```

```
    <note>
           [parameters]
    </note>
</noteTabs>
```

The Note tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.

Name	Reqd?	Type	Description
name	No	String	Sets the tool tip text for the tab.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.

Name	Reqd?	Type	Description
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
height	No	String	Height of the tab in pixels.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
value	No	String	This element specifies the value of the tab. This can be a maximum of 2000 characters.
width	No	String	Width of the tab in pixels.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.

Name	Reqd?	Type	Description
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Number Tab

Place this tag on the document where you want the recipient to enter a number. It uses the same parameters as a Text tab, with the validation message and pattern set for number information.

Note: When getting information that includes this tab type, the original value of the tab when the associated envelope was sent is included in the response.

An example Number tab json layout is shown below:

```
"numberTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "maxLength": null,
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Number Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "requireAll": false,
    "requireInitialOnSharedChange": false,
```

```
    "shared": false,
    "height": 11,
    "validationMessage": "",
    "validationPattern": "",
    "isPaymentAmount": false,
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional number level in the layout.

```
<numberTabs>
    <number>
           [parameters]
    </number>
</numberTabs>
```

The Number tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.

Name	Reqd?	Type	Description
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
maxLength	No	String	Sets the maximum number of characters a signer can enter in the tag.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.

Name	Reqd?	Type	Description
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.

Name	Reqd?	Type	Description
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	Sets the tool tip text for the tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab in pixels.
requireAll	No	Boolean	Optional element associated with *shared* element (field markup). When true and *shared* is true, information must be entered in this field to complete the envelope.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
height	No	String	Height of the tab in pixels.
validationMessage	No	String	Message to be displayed to the signer if the validation rule fails. This is optional and if not provided the default DocuSign message is displayed. This can be a maximum of 250 characters.
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.
isPaymentAmount	No	Boolean	When true, sets this as a payment tab. Can only be used with Text, Number, Formula or List tabs. The value of the tab must be a number.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Radio Group Tab

Place this tag on the document in a location where the recipient can select one option from a group of options using a radio button. The radio buttons do not have to be on the same page in a document.

An example Radio Group tab json layout is shown below:

```
"radioGroupTabs": [{
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "groupName": "Radio Group 1",
    "radios": [{
          "anchorString": null,
          "anchorXOffset": null,
          "anchorYOffset": null,
          "anchorIgnoreIfNotPresent": null,
```

```
            "anchorUnits": null,
            "anchorCaseSensitive": null,
            "anchorMatchWholeWord": null,
            "anchorHorizontalAlignment": null,
            "locked": false,
            "pageNumber": "1",
            "required": false,
            "selected": false,
            "value": "Radio",
            "xPosition": "234",
            "yPosition": "159",
            "tabOrder": {}

    },
    {
            "anchorString": null,
            "anchorXOffset": null,
            "anchorYOffset": null,
            "anchorIgnoreIfNotPresent": null,
            "anchorUnits": null,
            "anchorCaseSensitive": null,
            "anchorMatchWholeWord": null,
            "anchorHorizontalAlignment": null,
            "locked": false,
            "pageNumber": "1",
            "required": false,
            "selected": true,
            "value": "Radio",
            "xPosition": "236",
            "yPosition": "190",
            "tabOrder": {}
    }],
    "recipientId": "1",
    "requireInitialOnSharedChange": false,
    "shared": false,
    "templateLocked": false,
    "templateRequired": false,
}],
```

Note for xml there is an additional radioGroup level in the layout.

```
<radioGroupTabs>
    <radioGroup>
            [parameters]
    </radioGroup>
</radioGroupTabs>
```

The Radio Group tab parameters are:

Name	Reqd?	Type	Description
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.

Name	Reqd?	Type	Description
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
groupName	No	String	The group name for radio buttons that are grouped together. This can be a maximum of 500 characters. It works in conjunction with the radios parameter.
radios	No	Radio	This sets the locations and status for radio buttons that are grouped together. The information for each radio button is:: anchorString: Optional anchor text information for the radio button. If it is not used then the other anchor parameters are ignored anchorXOffset: This specifies tab location as XOffset position, using the anchorUnits, from the anchorString. anchorYOffset: This specifies tab location as YOffset position, using the anchorUnits, from the anchorString. anchorIgnoreIfNotPresent: True or false setting. If true, this tab is ignored if anchorString is not found in the document. anchorUnits: Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches. locked: If true, the Signer cannot change the data in the tab. pageNumber: sets the page the radio

Name	Reqd?	Type	Description
			button is located on. required: If true, the signer is required to fill out this tab. selected: Sets if this is radio button is selected. Use true/false to show the value is selected or not. Only one radio button in a group can be selected (true). value: Sets the value for the radio button. The value of each radio button in a group must be unique. xPosition: This indicates the horizontal offset of the radio button on the page. DocuSign uses 72 DPI when determining position. yPosition: This indicates the vertical offset of the radio button on the page. DocuSign uses 72 DPI when determining position. tabOrder: A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field. **Note:** Setting the requireInitialOnSharedChange property to true does not currently require the signer to initial when they modify a shared Radio Group. This is a known issue and DocuSign is working to correct it.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.

Name	Reqd?	Type	Description
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.

Sign Here Tab

Place this tag to have a recipient place their signature in the document. The “optional” parameter sets if the signature is required or optional.

An example Sign Here tab json layout is shown below:

```
"signHereTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "249",
    "yPosition": "15",
    "name": "Sign Here",
    "optional": false,
    "scaleValue": 1,
    "tabLabel": "Signature 3"
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional signHere level in the layout.

```
<signHereTabs>
    <signHere>
           [parameters]
    </signHere>
</signHereTabs>
```

The Sign Here tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.

Name	Reqd?	Type	Description
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
name	No	String	Sets the tool tip text for the tab.
optional	No	Boolean	When true, the Sign Here tab is optional and the recipient is not required to add a signature in the tab to complete an envelope.
scaleValue	No	String	Sets the size for Sign Here tab. It can be value from 0.5 to 1.0, where 1.0 represents full size and 0.5 is 50% size. When changing the size the upper left corner of the tab is the origin point for the tab, so changing the size causes the tab to shrink up and to the left. Note that this is different than the scale setting in the Classic DocuSign Experience web application where the tab origin is the lower left corner.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Signer Attachment Tab

Place this tag on the document when you want the recipient to add supporting documents to an envelope.

An example Signer Attachment tab json layout is shown below:

```
"signerAttachmentTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
```

```
    "customTabId": null,
    "documentId": "1",
    "optional": "false",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "24",
    "yPosition": "153",
    "tabLabel": "Attachment 10",
    "required": true,
    "scaleValue": "1.0",
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional signerAttachment level in the layout.

```
<signerAttachmentTabs>
    <signerAttachment>
           [parameters]
    </signerAttachment>
</signerAttachmentTabs>
```

The Signer Attachment tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.

Name	Reqd?	Type	Description
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
optional	No	Boolean	When set to 'true', the signerAttachmentTab is optional and not required to complete the envelope.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.

Name	Reqd?	Type	Description
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value.
required	No	Boolean	If true, the signer is required to fill out this tab.
scaleValue	No	String	Sets the size for Sign Here tab. It can be value from 0.5 to 1.0, where 1.0 represents full size and 0.5 is 50% size. When changing the size the upper left corner of the tab is the origin point for the tab, so changing the size causes the tab to shrink up and to the left. Note that this is different than the scale setting in the Classic DocuSign Experience web application where the tab origin is the lower left corner.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.

Name	Reqd?	Type	Description
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

SSN Tab

Place this tag on the document where you want the recipient to enter a Social Security Number (SSN). A SSN can be typed with or without dashes. It uses the same parameters as a Text tab, with the validation message and pattern set for SSN information.

Note: When getting information that includes this tab type, the original value of the tab when the associated envelope was sent is included in the response.

An example SSN tab json layout is shown below:

```
"ssnTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "maxLength": null,
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "SSN Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
```

```
    "requireAll": false,
    "requireInitialOnSharedChange": false,
    "shared": false,
    "height": 11,
    "validationMessage": "",
    "validationPattern": "",
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional ssn level in the layout.

```
<ssnTabs>
    <ssn>
            [parameters]
    </ssn>
</ssnTabs>
```

The SSN tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.

Name	Reqd?	Type	Description
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
maxLength	No	String	Sets the maximum number of characters a signer can enter in the tag.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.

Name	Reqd?	Type	Description
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.

Name	Reqd?	Type	Description
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	Sets the tool tip text for the tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab in pixels.
requireAll	No	Boolean	Optional element associated with *shared* element (field markup). When true and *shared* is true, information must be entered in this field to complete the envelope.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
height	No	String	Height of the tab in pixels.
validationMessage	No	String	Message to be displayed to the signer if the validation rule fails. This is optional and if not provided the default DocuSign message is displayed. This can be a maximum of 250 characters.

Name	Reqd?	Type	Description
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Text Tab

This tag is an adaptable field that allows the recipient to enter different text information.

Note: When getting information that includes this tab type, the original value of the tab when the associated envelope was sent is included in the response.

An example Text tab json layout is shown below:

```
"textTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "maxLength": null,
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Data Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "requireAll": false,
    "requireInitialOnSharedChange": false,
    "senderRequired": false,
    "shared": false,
    "validationMessage": "",
    "validationPattern": "",
    "height": 11,
    "isPaymentAmount": false
    "validationPattern": "",
    "validationMessage": "",
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional text level in the layout.

```
<textTabs>
    <text>
          [parameters]
```

```
    </text>
</textTabs>
```

The Text tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.

Name	Reqd?	Type	Description
maxLength	No	String	Sets the maximum number of characters a signer can enter in the tag.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.

Name	Reqd?	Type	Description
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	Sets the tool tip text for the tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab in pixels.
requireAll	No	Boolean	Optional element associated with *shared* element (field markup). When true and *shared* is true, information must be entered in this field to complete the envelope.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.

Name	Reqd?	Type	Description
senderRequired	No	Boolean	When true for a tab in a template, the sender must populate the tab before an envelope can be sent using the template. This value tab can only be changed by modifying (PUT) the template. Tabs with a senderRequired value of true cannot be deleted from an envelope.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
validationMessage	No	String	Message to be displayed to the signer if the validation rule from validationPattern fails. This is optional and if not provided the default DocuSign message will be displayed. This can be a maximum of 250 characters.
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.
height	No	String	Height of the tab in pixels.
isPaymentAmount	No	Boolean	When true, sets this as a payment tab. Can only be used with Text, Number, Formula or List tabs. The value of the tab must be a number.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Title Tab

Place this tag on the document where you want the recipient's title to appear.

Note: When getting information that includes this tab type, the original value of the tab when the associated envelope was sent is included in the response.

An example Title tab json layout is shown below:

```
"titleTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
```

```
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "maxLength": null,
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "30",
    "yPosition": "103",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Title",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": false,
    "name": "Title",
    "required": true,
    "value": "Regular Guy",
    "width": 42,
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional title level in the layout.

```
<titleTabs>
    <title>
           [parameters]
    </title>
</titleTabs>
```

The Title tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.

Name	Reqd?	Type	Description
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
maxLength	No	String	Sets the maximum number of characters a signer can enter in the tag.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.

Name	Reqd?	Type	Description
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	Sets the tool tip text for the tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab.

Name	Reqd?	Type	Description
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Zip Tab

Place this tag on the document where you want the recipient to enter a ZIP code. The ZIP code can be a five numbers or the ZIP+4 format with nine numbers. The zip code can be typed with or without dashes. It uses the same parameters as a Text tab, with the validation message and pattern set for ZIP code information.

Note: When getting information that includes this tab type, the original value of the tab when the associated envelope was sent is included in the response.

An example Zip tab json layout is shown below:

```
"zipTabs": [{
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
```

```
    "anchorUnits": null,
    "anchorCaseSensitive": null,
    "anchorMatchWholeWord": null,
    "anchorHorizontalAlignment": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "customTabId": null,
    "documentId": "1",
    "maxLength": null,
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Zip Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disableAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "requireAll": false,
    "requireInitialOnSharedChange": false,
    "shared": false,
    "height": 11,
    "useDash4": ""
    "validationMessage": "",
    "validationPattern": "",
    "mergeField": {},
    "tabOrder": {}
}],
```

Note for xml there is an additional zip level in the layout.

```
<zipTabs>
    <zip>
          [parameters]
    </zip>
</zipTabs>
```

The Zip tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.
anchorXOffset	No	String	This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.

Name	Reqd?	Type	Description
anchorYOffset	No	String	This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.
anchorIgnoreIfNotPresent	No	String	True or false setting. If true, this tab is ignored if anchorString is not found in the document.
anchorUnits	No	String	Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.
anchorCaseSensitive	No	String	Reserved for DocuSign use only.
anchorMatchWholeWord	No	String	Reserved for DocuSign use only.
anchorHorizontalAlignment	No	String	Reserved for DocuSign use only.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab that controls this tab's visibility.
conditionalParentValue	No	String	Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility. If the parent tab is a Checkbox, Radio button, Optional Signature, or Optional Initial use "on" as the value to show that the parent tab is active.
customTabId	No	String	The DocuSign generated custom tab ID for the custom tab to be applied. This can only be used when adding new tabs for a recipient. When used, the new tab inherits all the custom tab properties.
documentId	Yes	String	This is the document ID number that the tab is placed on. This must refer to an existing Document's ID attribute.
maxLength	No	String	Sets the maximum number of characters a signer can enter in the tag.
pageNumber	Yes	String	This specifies the page number where the tab will be affixed.
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.
templateLocked	No	Boolean	Optional element. Used only when working with template tabs. If true, the attributes of the tab cannot be changed by the sender.

Name	Reqd?	Type	Description
templateRequired	No	Boolean	Optional element. Used only when working with template tabs. If true, the tab cannot be removed by the sender.
xPosition	Yes	String	This indicates the horizontal offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
yPosition	Yes	String	This indicates the vertical offset of the tab on the page, in a coordinate space that has left top corner of the document as origin. DocuSign uses 72 DPI when determining position.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used to display the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, Verdana, OcrA, MSGothic and MSMincho.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue, Purple, or White.
fontSize	No	FontSize	The font size used for the information in the tab. Possible values are: Size7, Size8, Size9, Size10, Size11, Size12, Size14, Size16, Size18, Size20, Size22, Size24, Size26, Size28, Size36, Size48, or Size72.
italic	No	Boolean	If true, the information in the tab is italic.
tabLabel	No	String	The label used for the tab. This can be a maximum of 500 characters. If an empty string is provided for this, an empty sting is used. If no value is provided, the tab type is used as the value. Note: Similar tab types can be set to automatically populate with the same data. See Automatically Populating Tabs for more information.

Name	Reqd?	Type	Description
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	Optional element. If true the field appears normally while the recipient is adding or modifying the information in the field, but the data is not visible (the characters are hidden by asterisks) to any other signer or the sender. When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console. This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.
disableAutoSize	No	Boolean	Disables the auto sizing of single line text boxes in the signing screen when the signer enters data. If disabled users will only be able enter as much data as the text box can hold. By default this is false. This property only affects single line text boxes.
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	Sets the tool tip text for the tab.
required	No	Boolean	If true, the signer is required to fill out this tab.
value	No	String	This element specifies the value of the tab.
width	No	String	Width of the tab in pixels.
requireAll	No	Boolean	Optional element associated with *shared* element (field markup). When true and *shared* is true, information must be entered in this field to complete the envelope.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
height	No	String	Height of the tab in pixels.

Name	Reqd?	Type	Description
useDash4	No	Boolean	When set to true, this tab allows the additional four numbers to be entered in the tab. Zip tabs that do not have this enabled are limited to five numbers.
validationMessage	No	String	Message to be displayed to the signer if the validation rule fails. This is optional and if not provided the default DocuSign message is displayed. This can be a maximum of 250 characters.
validationPattern	No	String	A regular expression that will be validated when data is entered in the field. This is optional and if not provided the default DocuSign validation rules will apply. This can be a maximum of 250 characters. Javascript RegEx object is used for regular expression validation. Regular expressions must be supported by this object to resolve.
mergeField	No		This can only be used it the account has a Connect for DocuSign for Salesforce configuration enabled. A complex element that specifies the relationship between a custom tab and a Salesforce object. It consists of: • configurationType – Sets the merge field type. Currently the only accepted value is Salesforce. • path – Sets the object associated with the custom tab. Currently this is the Salesforce Object. • writeBack – When true information entered in the tab automatically updates the related Salesforce data when an envelope is completed. • allowSenderToEdit – When true the sender can modify the value of the custom tab during the sending process.

Name	Reqd?	Type	Description
tabOrder	No	String	A positive integer that sets the order the tab is navigated to during signing. Tabs on a page are navigated to in ascending Tab order value, starting with the lowest number and moving to the highest. If two or more tabs have the same Tab order number, the normal auto-navigation setting behavior for the envelope is used.

Error Code Information

The DocuSign REST API's return either 200 (OK) or 201 (Created) when an API call successfully runs to completion, and status codes in the 400-500 range for failures. For API's that process a single item, this overall status code determines success/failure. There are special considerations for PUT and POST calls that process multiple items, refer to the Error Response Handling for API calls that Support Multiple Items section for more information on error response handling.

If an API call is not successful, the HTTP response code returned is generally one of the following:

- 400 (BadRequest) – This indicates that a portion of the request made to the API call was not valid or could not be processed in the current context.
- 401 (Unauthorized) – This indicates that the the authentication or authorization required for the API call did not pass.
- 404 (NotFound) – This indicates that the resource being accessed does not exist. This might occur on GET or DELETE requests.
- 415 (UnsupportedMediaType) – This indicates that the data type of some data in the request is not supported.

The following error codes occur with much less frequency:

- 405 (MethodNotAllowed) – This generally indicates that the HTTP Method (GET, PUT, POST or DELETE) is not valid for this resource URI.
- 500 (InternalServerError) – This indicates that an internal server-side error has occurred.

In most cases, a more detailed "errorDetails" structure is returned as the response body when an error occurs. However, depending on the error, it is also possible that the server will return HTML in place of the errorDetails structure or no errorDetails was deemed necessary and it is omitted. The table of Error Codes and Associated Messages lists the possible error codes and messages.

General Error Response Handling

The general flow for response handling is:

```
if (statusCode == 200 || statusCode == 201)
{
    // successful call - but do special processing for multi-item PUT & POST requests
}
else if (statusCode == 404)
```

```
{
    // call was processed but the resource being accessed was not found
    // envelope doesn't exist, user doesn't exist, etc. There may not be any more details
in the errorDetails response.
}
else // other errors
{
    var errorDetails = Json.Decode(response.body);
    if (errorDetails!= null)
    {
           // process errorDetails.errorCode - See table of error codes
           if (errorDetails.message != null)
           {
                  // use errorDetails.message to get more information in textual form.
           }
    }
}
```

Error Response Handling for API calls that Support Multiple Items

The DocuSign REST API supports many calls that operate on multiple items, such as adding one or more users to an account or adding one or more documents to a draft envelope. This is done to decrease the number of web callouts required to perform some common operations. For API calls that support multiple items, the overall HTTP response code will be 200 (OK) or 201 (Created) if the API call has run to completion, which means all items in the request structure were able to be processed. This response code is returned even if there were failures for individual items in the array. The response body contains information about each item processed and the structure for each item contains a non-null errorDetails structure if that item was not successfully processed. The caller is responsible for checking for the errorDetails structure for each item in the array to perform proper error handling.

The general response processing logic for these types of API's is:

```
if (statusCode == 200 || statusCode == 201)
{
    // check for errors with each item processed.
    var itemsArray = Json.Decode(response.body);
    foreach(var item in itemsArray)
    {
           if (item.errorDetails != null)
           {
                  // an error occurred during processing - handle it
                  string errorCode = item.errorDetails.errorCode;
                  string errorMessage = item.errorDetails.message;
                  // perform error handling based on errorCode
           }
           else
           {
                  // item processing was successful
           }
    }
}
else
{
    // statusCode indicated overall error with the request.
}
```

Error Codes and Associated Messages

The following table lists all of the error codes and associated messages. The list is presented in alphabetical order based on the error code.

DocuSign's SOAP and REST APIs use the same error list. The REST API shows the upper-case error code and the message text, while the SOAP API shows the error number and the message text. In many cases additional information is added to the message string.

Error Code	Message (Base)
ACCESS_CODE_IN_WRONG_FORMAT	Access Code provided is not in the correct format.
ACCOUNT_DOES_NOT_EXIST_IN_SYSTEM	The AccountID did not identify an Account in the system.
ACCOUNT_HAS_BEEN_SUSPENDED	The specified Account is suspended
ACCOUNT_IS_NOT_MANAGED_BY_DISTRIBUTOR	The account is not managed by the distributor.
ACCOUNT_LACKS_CUSTOM_TAB_PERMISSION	This Account lacks sufficient permissions to use custom tab.
ACCOUNT_LACKS_PERMISSION_TO_CORRECT_TABS	The account lacks sufficient permissions to correct tabs.
ACCOUNT_LACKS_PERMISSIONS	This Account lacks sufficient permissions.
ACCOUNT_LACKS_PRICING_PLAN	The specified Account lacks an Account Pricing Plan
ACCOUNT_NOT_AUTHORIZED_FOR_ ENVELOPE	This account is not authorized to access the requested envelope.
ACCOUNT_PLAN_DOES_NOT_EIXST_IN_SYSTEM	The Current Account Plan does not exist in the system.
ACCOUNT_PROVISIONER_INSUFFICIENT_SEATS	Insufficient Account Provisioning Distributor Seats.
ACCOUNT_PROVISIONER_INVALID_ADMIN_NRDS	Invalid Admin NRDS ID
ACCOUNT_SPECIFIED_INVALID	The account id provided does not exist, or is improperly formatted.
ACTIVATION_FAILED	The token for a recipient that has failed activation cannot be generated.
ADDRESS_DOES_NOT_EXIST_IN_SYSTEM	The Address Id did not identify an Address in the system.
ADDRESS_UPDATE_FAILED	The address update failed.
ADDRESSBOOK_ACCOUNTNAME_INVALID	The account name provided does not exist or is null or empty.
ADDRESSBOOK_ATTEMPTED_UPDATE_FAILED	Attempting to update the address book failed.
ADDRESSBOOK_CANTSHARE	Invalid address book share permissions.
ADDRESSBOOK_CONTACTID_INVALID	The contact id provided does not exist in the address book, or is improperly formatted.
ADDRESSBOOK_DESIGNATION_INVALID	Address Book Designation is not valid.
ADDRESSBOOK_EMAIL_INVALID	Invalid Email for the address book entry.
ADDRESSBOOK_ID_INVALID	Invalid ID for the address book entry.
ADDRESSBOOK_INVALID_RESULT_STATE	The address book action resulted in an invalid state.
ADDRESSBOOK_INVALID_SEARCH_CRITERIA	The search criteria provided are invalid.

Error Code	Message (Base)
ADDRESSBOOK_INVALID_SHARING_OPTION	Only 'true' or 'false' are valid values for 'shared'.
ADDRESSBOOK_INVALID_UPDATE_CRITERIA	The update criteria provided are invalid.
ADDRESSBOOK_NOTOWNER	Invalid Owner for the address book entry.
ADDRESSBOOK_USERNAME_INVALID	The user name provided was null or empty.
ANCHOR_TAB_STRING_NOT_FOUND	The specified Anchor Tab string was not found in the document.
ANCHOR_TAG_PROCESSING_FAILURE	An Error Occurred during anchor tag processing.
API_DOCGEN_FILE_CREATE	File could not be created.
API_DOCGEN_FILE_NO_VALUE	File does not contain API documentation.
API_DOCGEN_FILE_NOT_EXIST	File does not exist.
API_DOCGEN_FILE_READ	File could not be read.
API_DOCGEN_INVALIDFILE	File is invalid.
APPSTORE_ERROR	App Store Error.
APPSTORE_PAYMENT_METHOD_ERROR	Payment method error.
APPSTORE_PRODUCTID_INVALID	Product Id is Invalid.
APPSTORE_PRODUCTID_MISSING	Product Id not provided
APPSTORE_RECEIPT_DATA_MISSING	Receipt Data not provided
APPSTORE_RECEIPT_MISSING	Receipt not provided
APPSTORE_RECEIPT_VERIFICATION_FAILED	Receipt verification failed.
AUTHORIZATION_HEADER_REQUIRED	Authorization header is required.
AUTHORIZATION_INSUFFICIENT_SCOPE	The request requires higher privileges than provided by the access token.
AUTHORIZATION_INVALID_REQUEST	The authorization request is malformed.
AUTHORIZATION_INVALID_TOKEN	The access token provided is expired, revoked or malformed.
AUTHORIZATION_MAX_ACCESS_TOKENS_EXCEEDED	Maximum number of access tokens exceeded.
AUTHORIZATION_REVOKED	Authorization has been revoked.
AUTHORIZATION_REVOKED_FAILED	Unable to revoke access.
BCC_EMAIL_ADDRESS_ID_NOT_FOUND	BCC Email Address Id not found.
BILLING_CHANGES_PENDING	Billing changes are pending.
BILLING_INVOICES_NOT_RETRIEVED	Billing Invoices could not be retrieved.
BILLING_NOT_CONFIGURED	Billing not configured in the system.
BILLING_PERIOD_DOES_NOT_EXIST_IN_SYSTEM	The Current Account Billing Period does not exist in the system.
BILLING_PLAN_ERROR	The billing plan could not be changed.
BILLING_PLAN_PREVIEW_ERROR	The billing plan could not be previewed.
BILLING_SETUP_DOES_NOT_EXIST_IN_SYSTEM	The Billing setup is not found in the system.
BILLING_SYSTEM_LOGIN_FAILED	Login to billing system failed.
BRAND_CREATE_FAILED	Brands could not be created.

Error Code	Message (Base)
BRAND_DELETE_FAILED	Brands could not be deleted.
BRAND_LOCKED	Brand is locked and cannot be reset.
BRAND_UPDATE_FAILED	One or more brands could not be updated.
BULK_ENVELOPE_ACCOUNT_LACKS_PERMISSIONS	This Account lacks bulk send permissions.
BULK_ENVELOPE_BLANK_CSV_HEADER_FIELD	CSV file contains blank header fields.
BULK_ENVELOPE_DUPLICATE_RECIPIENTS	Bulk Recipient duplicates found.
BULK_ENVELOPE_RECIPIENT_NOT_BULK_RECIPIENT	Recipient is not a bulk recipient.
BULK_ENVELOPE_EMPTY_VALUE	Invalid empty value.
BULK_ENVELOPE_FIELD_NOT_SUPPORTED	Field not supported for bulk recipient.
BULK_ENVELOPE_FILE_DELETE_FAILED	Fail to delete bulk send file.
BULK_ENVELOPE_HAS_NO_BULK_RECIPIENTS	Bulk recipient envelope doesn't have bulk recipients.
BULK_ENVELOPE_INVALID_AUTH_ID	Invalid authentication identifier.
BULK_ENVELOPE_INVALID_BATCHID	Batch Id is invalid.
BULK_ENVELOPE_INVALID_CHARACTERS	Invalid characters.
BULK_ENVELOPE_INVALID_CSV_LIST	Invalid bulk recipient CSV list.
BULK_ENVELOPE_INVALID_CSV_ROW	Invalid line in bulk recipient CSV list.
BULK_ENVELOPE_INVALID_EMAIL	Invalid Email.
BULK_ENVELOPE_INVALID_PERMISSION_PHONE_AUTH	Recipient Phone Authentication not enabled for account.
BULK_ENVELOPE_INVALID_PERMISSION_SAML_AUTH	Recipient SAML Authentication not enabled for account.
BULK_ENVELOPE_INVALID_PERMISSION_SMS_AUTH	Recipient SMS Authentication not enabled for account.
BULK_ENVELOPE_INVALID_PERMISSION_SOCIAL_AUTH	Recipient Social Authentication not enabled for account.
BULK_ENVELOPE_INVALID_PHONE	Invalid phone number.
BULK_ENVELOPE_MAX_ENVELOPE_EXCEEDED	Maximum number of bulk envelope exceeded.
BULK_ENVELOPE_MAX_RECIPIENTS_EXCEEDED	Maximum number of bulk recipients was exceeded.
BULK_ENVELOPE_MAX_VALUE_LENGTH_EXCEEDED	Maximum length for a value was exceeded.
BULK_ENVELOPE_MUST_HAVE_ACCOUNT	Bulk Recipient must have an account.
BULK_ENVELOPE_NOT_DRAFT_OR_TEMPLATE	The requested envelope is not a draft or template.
BULK_ENVELOPE_ONLY_ONE_BULK_RECIPIENT_ALLOWED	Bulk envelope only allows one bulk recipient.
BULK_ENVELOPE_RECIPIENT_NOT_BULK_RECIPIENT	Recipient is not a bulk recipient.
BULK_ENVELOPE_REQUIRED_FIELD_MISSING	Required field is missing.

Error Code	Message (Base)
BULK_ENVELOPE_SEND_FAILED	Fail to send bulk envelopes.
BULK_ENVELOPE_SMS_AUTH_PHONE_INVALID	SMS authentication phone number cannot be usersupplied.
BULK_ENVELOPE_SMS_AUTH_PHONE_MISSING	SMS authentication phone number cannot be empty.
BULK_ENVELOPE_TEMPLATE_IS_TIED_TO_POWERFORM	Template that is tied to a powerform is not allowed to bulk send.
BULK_ENVELOPE_USER_LACKS_PERMISSIONS	This User lacks bulk send permissions.
CACHE_INVALID_INIT_STRING	The initialization string was null or empty.
CACHE_OBJECT_SIZE_EXCEEDED	The cache object size has been exceeded.
CANNOT_ADD_DOCUMENT	Cannot add the document for the envelope.
CANNOT_ALLOW_MARKUP	Account does not have permission to set Allow Markup.
CANNOT_ASSIGN_MANAGERS_SAME_ROUTING_ORDER	Multiple Send to Manage Roles (Agent, Editor or Intermediary) cannot be assigned same routing order.
CANNOT_ASSIGN_TAB_TO_MANAGERROLE	Send to Manage Role (Agent, Editor or Intermediary) cannot have tabs assigned.
CANNOT_DELETE_FROM_RECYCLEBIN	Envelopes may not be deleted from the RecycleBin
CANNOT_DELETE_PREDEFINED_FOLDERS	You cannot delete predefined system folders.
CANNOT_EXCLUDE_DOCUMENT	This document cannot be excluded for this recipient.
CANNOT_SEND_TO_AGENT	Account does not have permission to send to Agent recipient type.
CANNOT_SEND_TO_CERTIFIED_DELIVERY	Account does not have permission to send to Certified Delivery recipient type.
CANNOT_SEND_TO_EDITOR	Account does not have permission to send to Editor recipient type.
CANNOT_SEND_TO_INPERSONSIGNER	Account does not have permission to send to InPersonSigner recipient type.
CANNOT_SEND_TO_INTERMEDIARY	Account does not have permission to send to Intermediary recipient type.
CAPTIVE_CARBON_COPY_RECIPIENT_NOT_SUPPORTED	A carbon copy recipient has been specified as captive. This operation is not supported.
CAPTIVE_IN_PERSON_SIGNER_RECIPIENT_NOT_SUPPORTED	Captive Recipient cannot be of type In Person Signer.
CAPTIVE_RECIPIENT_NOT_FOUND	The specified captive recipient was not found.
CAPTIVE_RECIPIENT_TOKEN_CLIENTURLS_NULL	Invalid RequestRecipientTokenClientURLs value. The clientUrls argument is null.
CONDITIONALTAB_HAS_INVALID_PARENT	A conditional tab references an invalid parent. Parent label must match another tab. Only one parent allowed. Signature tabs may not be parent tabs.

Error Code	Message (Base)
CONNECT_CANNOT_CHANGE_TYPE	Connect configuration cannot change type after creation.
CONNECT_ERROR	Connect Error.
CONSUMER_DISCLOSURE_ACCEPTANCE_REQUIRED	Consumer disclosure acceptance is required.
CORRECTED_EMAIL_IS_RESERVED	The corrected email(s) are reserved.
CORRECTION_HAS_A_BLANK_USERNAME	The specified envelope correction has a blank username.
CORRECTION_HAS_A_USERNAME_WITH_INVALID_CHARACTERS	The specified envelope correction has a username with invalid characters.
CORRECTION_HAS_DUPLICATE_RECIPIENTS	The specified envelope corrections have duplicate recipients.
CORRECTION_RESULTS_IN_DUPLICATE_RECIPIENTS	The specified envelope corrections result in duplicate recipients.
CREDIT_CARD_DOES_NOT_EXIST_IN_SYSTEM	The Credit Card does not exist in the system.
CREDIT_CARD_EXPIRED	Credit Card is expired.
CREDIT_CARD_INVALID_ADDRESS	Credit Card has invalid address.
CREDIT_CARD_NOT_ALLOWED_FOR_CURRENCY	The Credit Card is not allowed for the specified currency.
CREDIT_CARD_NOT_AUTHORIZED	Credit Card not authorized.
CREDIT_CARD_UPDATE_FAILED	Credit Card update failed.
CUSTOM_FIELD_ALREADY_EXISTS	Custom Field already exists.
CUSTOM_SETTING_NOT_FOUND	The specified setting was not found.
CUSTOM_SETTINGS_MAX_EXCEEDED	The maximum length for custom settings was exceeded
CUSTOM_SETTINGS_NOT_FOUND	Custom settings were not found.
CUSTOM_TAB_APICREDENTIAL_IS_NULL	The API Credential is null.
CUSTOM_TAB_CREATE_FAILED	Custom Tab create failed.
CUSTOM_TAB_DELETE_FAILED	Custom Tab delete failed.
CUSTOM_TAB_IS_NULL	The Custom Tab is null.
CUSTOM_TAB_UPDATE_FAILED	Custom Tab update failed.
CUSTOMTAB_IS_INCOMPLETE	A Custom Tab is not Complete. A Custom Tab requires both a Name and a TabLabel.
DECLINE_TO_SIGN_USER_IS_NOT_RECIPIENT	User does not match recipient.
DELETED_DRAFTS_CAN_ONLY_BE_MOVED_TO_DRAFTS	Deleted draft envelopes can only be moved back to the drafts folder.
DISCOUNT_FAILED	There is an error selecting the specified discount.
DOCUMENT_DOES_NOT_EXIST	The document specified was not found.
DOCUMENT_NAME_NOT_SPECIFIED	Document name not specified.
DOCUMENT_PURGE_FAILED	Error adding documents to the purge queue.

Error Code	Message (Base)
DOCUMENT_UPDATE_FAILED	The document could not be added/updated.
DOCUMENT_UPDATE_NOT_ALLOWED	Document update is not allowed.
DOCUMENT_UPLOAD_NOT_ALLOWED	Document upload not allowed.
DOCUMENT_UPLOAD_ONLY_PDF_ALLOWED	Only PDF document upload allowed.
DOCUMENT_UPLOAD_PDF_NOT_ALLOWED	PDF document upload not allowed.
DOCUMENTFIELDNAMETOLONG	Document field name to long.
DOCUMENTFIELDVALUETOLONG	Document field value to long.
DOMAIN_CHANGE_FAILED	The email domain change was not initiated.
DOMAIN_CHANGE_INTERRUPTED	The domain change was not successful.
DOMAIN_CHANGE_PARTIALLY_APPLIED	The domain change may have been partially successful.
DRAFT_ENVELOPES_CAN_ONLY_BE_MOVED_TO_RECYCLEBIN	Draft envelopes can only be moved to the RecycleBin.
DUPLICATE_BCC_EMAIL_ADDRESS	Duplicate BCC email address.
DUPLICATE_BCC_EMAIL_ADDRESS_ID	Duplicate BCC Email Address Id specified.
EDIT_LOCK_ENVELOPE_ALREADY_LOCKED	The envelope is already locked.
EDIT_LOCK_ENVELOPE_LOCKED	The envelope is locked.
EDIT_LOCK_ENVELOPE_NOT_LOCKED	The envelope is not locked.
EDIT_LOCK_ERROR	Edit lock error.
EDIT_LOCK_FAILED_TO_ACQUIRE_LOCK	Failed to acquire the lock.
EDIT_LOCK_FAILED_TO_UPDATE_LOCK	Failed to update the lock.
EDIT_LOCK_INVALID_LOCK_DURATION	Invalid lock duration value.
EDIT_LOCK_INVALID_LOCK_TYPE	Invalid lock type.
EDIT_LOCK_INVALID_LOCKEDBYAPP	Invalid LockedByApp value.
EDIT_LOCK_INVALID_STATE_FOR_LOCK	Failed to acquire the lock because the envelope status was not 'created', 'sent' or 'delivered'.
EDIT_LOCK_NOT_ENABLED	Locking for edit is not enabled.
EDIT_LOCK_NOT_LOCK_OWNER	The user is not the owner of the lock.
EDIT_LOCK_TEMPLATE_ALREADY_LOCKED	The template is already locked.
EDIT_LOCK_TEMPLATE_LOCKED	The template is locked.
EDIT_LOCK_TEMPLATE_NOT_LOCKED	The template is not locked.
EDIT_SCRATCHPAD_CREATE_ERROR	Error creating the scratchpad.
EDIT_SCRATCHPAD_NOT_ENABLED	Scratchpad not enabled.
EDIT_SCRATCHPAD_SAVE_ERROR	Error saving the scratchpad changes to the database.
EDITOR_MUST_HAVE_ACCOUNT	The recipient Editor must be an existing DocuSign account member.
EMAIL_IS_RESERVED	The specified email(s) are reserved.
EMAIL_NOT_ACTIVE	The email address does not specify an active user.
EMAIL_NOT_SHARED	The email address is not shared.

Error Code	Message (Base)
EMAIL_SETTINGS_ALREADY_EXISTS	This email settings object already exists.
EMAIL_SETTINGS_NOT_FOUND	No email settings were found.
ENVELOPE_AC_EXPORT_COMPLETED	Envelope already exported.
ENVELOPE_AC_EXPORT_INVALID_STATUS	Envelope is not in the correct export status.
ENVELOPE_AC_EXPORT_INVALID_TRANSID	Invalid transaction ID.
ENVELOPE_AC_EXPORT_NOT_AC_COPY	Envelope not authoritative copy.
ENVELOPE_ALLOWANCE_EXCEEDED	The envelope allowance for the account has been exceeded.
ENVELOPE_ALREADY_SENT	Envelope has already been sent.
ENVELOPE_CANNOT_CORRECT_INVALID_STATE	Only envelopes in the 'Sent' or 'Delivered' states may be corrected.
ENVELOPE_CANNOT_TRANSFER_INVALID_ACSTATUS	Envelopes with AC Status - Deposit Pending or Deposited or Deposited As Authoritative Copy or Deposit Failed cannot be transferred.
ENVELOPE_CANNOT_TRANSFER_INVALID_STATE	The specified envelope is not in a terminal state and cannot be transferred.
ENVELOPE_CANNOT_USE_TEMPLATEID_FOR_ENVELOPE_OPERATION	Cannot use a template id for an envelope operation.
ENVELOPE_CANNOT_VOID_INVALID_STATE	Only envelopes in the 'Sent' or 'Delivered' states may be voided.
ENVELOPE_CUSTOM_FIELD_MISSING	A required envelope custom field is missing.
ENVELOPE_DECLINE_TO_SIGN_INVALID_STATUS	The envelope is no longer in a status to be signed.
ENVELOPE_DOES_NOT_EXIST	The envelope specified either does not exist or you have no rights to it.
ENVELOPE_HAS_BEEN_VOIDED	The envelope you are attempting to access has been voided by the sender. For more information, please contact the sender:
ENVELOPE_HAS_DUPLICATE_RECIPIENTS	The specified envelope has duplicate recipients.
ENVELOPE_HAS_INSESSION_RECIPIENTS	The specified envelope has In-Session recipients.
ENVELOPE_HAS_NO_RECIPIENTS	The specified envelope has no recipients.
ENVELOPE_INVALID_STATUS	Invalid envelope status.
ENVELOPE_IS_INCOMPLETE	The Envelope is not Complete. A Complete Envelope Requires Documents, Recipients, Tabs, and a Subject Line.
ENVELOPE_NOT_DRAFT	The requested envelope is not a draft.
ENVELOPE_NOT_ENABLED_WETSIGN	Sign on paper has not been enabled for this envelope.
ENVELOPE_NOT_IN_FOLDER	The envelope does not exist in the folder
ENVELOPE_ONLYTHREECUSTOMFIELDS_ALLOWED	Only three custom fields allowed per envelope.

Error Code	Message (Base)
ENVELOPE_TRANSFEREE_ALREADY_OWNS_ENVELOPE	The specified transfer recipient of the envelope is already the owner of the envelope.
EXISTING_SALESFORCE_CONNECT	This is already any existing Salesforce Connect integration.
EXTERNAL_DOC_SERVICE_ERROR	An error occurred in the external doc service.
EXTERNAL_DOC_SERVICE_NOT_AUTHENTICATED	External doc service not authenticated.
FAILED_EMAIL_SENDING	Email sending failed.
FAILED_SCHEMA_VALIDATION	Schema validation Failed. Additional Errors follow.
FAILED_TO_CONNECT_TO_SERVICE	Unable to connect to the service.
FAILED_TO_CREATE_REQUEST	Failed to create the request for the operation.
FAILED_TO_CREATE_RESPONSE	Failed to create the response for the operation.
FAX_BACK_MUST_BE_SIGNER	Fax Recipient Token must be of type Signer.
FAX_ERROR_DOMAIN_INVALID	The domain name specified was invalid. This field is required.
FAX_ERROR_INVALID_ENVELOPE_ID	The fax entry could not be saved.
FAX_ERROR_NO_MATCHING_FAX_RECORD	A corresponding fax status could not be found.
FAX_ERROR_OCCURRED	Fax error occurred.
FAX_ERROR_POST_INVALID	The POST request for submitting a fax was invalid or not present.
FAX_ERROR_SAVE_FAX_FAILED	The envelope fax id provided is invalid (zero or less).
FAX_ERROR_STATUS_REQUEST_INVALID	The fax error request was null or empty.
FAX_FAILED_TO_SEND	The fax was attempted, but failed to send.
FAX_FORMAT_INVALID	The fax could not be read as a PDF.
FAX_INTEGRATOR_NOT_AUTHORIZED	The fax provider is not authorized to send or receive faxes.
FAX_RECIPIENT_CORRECT_WAS_SENT	Fax Recipient cannot be corrected after sent.
FAX_RECIPIENT_ROUTING	Fax Recipient must be the only recipient at a routing order.
FAX_RECIPIENT_TYPE_NOT_SUPPORTED	Fax Recipient Type must be a signer.
FAX_START_PENDING_FAILED	Fax Pending Failed.
FAX_USAGE_NOT_RECORDED	Fax usage could not be recorded.
FILTER_END_GREATER_THAN_MAX_ALLOWED	Filter End Date is greater than maximum value allowed.
FILTER_END_LESS_THEN_MIN_ALLOWED	Filter End Date is less than minimum value allowed.
FILTER_START_EXCEEDS_END	Filter Start Date cannot be greater than Filter End Date.

Error Code	Message (Base)
FILTER_START_GREATER_THAN_MAX_ALLOWED	Filter Start Date is greater than maximum value allowed.
FILTER_START_LESS_THAN_MIN_ALLOWED	Filter Start Date is less than minimum value allowed.
FOLDER_ID_DEFINED_DURING_ADD	The folder id must not be defined when adding a folder.
FOLDER_ID_NOT_FOUND	The folder id was not found.
FOLDER_NAME_NOT_DEFINED	The name was not specified for the folder.
FOLDER_NOT_FOUND	The specified folder id was not found for the user.
FOLDER_UPDATE_FAILED	The folder could not be created/updated.
FORMAT_CONVERSION_ERROR	The data could not be converted.
FORMAT_CONVERSION_ERROR_INVALID_FILE_TYPE	Invalid file type chosen for conversion.
FORMULA_CIRCULAR_REFERENCE	Recipient Calculated Fields have a circular reference.
GET_PAGE_IMAGE_FAILED	Get page image failed.
HOURLY_APIINVOCATION_LIMIT_EXCEEDED	The maximum number of hourly API invocations has been exceeded.
IN_PERSON_SIGNER_NAME_CANNOT_BE_BLANK	For In Person Signer type, the Recipient Signer Name cannot be blank.
IN_PERSON_SIGNING_HOST_MUST_BE_VALID_USER	The In Person Signing Host must be a valid and active DocuSign user.
INVALID_ACCESSIBILITY_FORMAT	Accessibility format is invalid.
INVALID_ACCOUNT_ROLE_SETTINGS	The settings for the account role are invalid.
INVALID_ADDRESS_COUNTRY	The value for 'country' is not supported.
INVALID_ADDRESS_STATE	The value for 'state' is not supported.
INVALID_ANCHOR_TAB_STRING	The Anchor Tab String Format is invalid.
INVALID_API_VERSION	The API Version specified is not valid.
INVALID_APPTOKEN	AppToken is invalid or distributor not found for AppToken
INVALID_BILLING_PLAN_RETIRED	The billing plan is retired and cannot be used.
INVALID_BILLING_PLAN_SUCCESSOR	Invalid Successor Billing Plan.
INVALID_BRAND	Invalid Brand for this Account.
INVALID_BRAND_ID	The brand identifier is unknown or invalid.
INVALID_BRAND_LOGO	The branding logo is unsupported or invalid.
INVALID_BRAND_LOGOTYPE	The type of logo is unrecognized.
INVALID_CAPTIVE_RECIPIENT_OPERATION	Invalid operation on captive recipient.
INVALID_CHARACTERS_PROVIDED	Invalid characters detected
INVALID_CLIENTUSERID	Invalid value specified for clientUserId.
INVALID_CONNECT_ID	Invalid connect id.
INVALID_CONNECT_ID	Invalid Connect ID.

Error Code	Message (Base)
INVALID_CONNECT_LOGID	Invalid connect log id.
INVALID_CONTENT_TRANSFER_ENCODING	Invalid 'content transfer encoding' for envelope.
INVALID_CONTENT_TYPE	Content Type specified is not supported.
INVALID_COUNTRY	The Country is not valid in the system.
INVALID_CREDIT_CARD_EXPIRATION	Credit Card Expiration is not valid.
INVALID_CREDIT_CARD_EXPIRATION_YEAR	Credit Card Expiration Year is not valid.
INVALID_CREDIT_CARD_INFORMATION	The credit card information supplied was not valid.
INVALID_CREDIT_CARD_NUMBER	Credit Card Number is not valid.
INVALID_CREDIT_CARD_TYPE	The Credit Card Type is not valid in the system.
INVALID_CSS_COLOR	The CSS color is not regognized as a valid and supported W3C-compliant format.
INVALID_CURRENCY	The currency is not valid in the system.
INVALID_CURRENCY_REQUEST	The currency type cannot be changed.
INVALID_CUSTOM_FIELD_ID	Invalid Custom Field Id.
INVALID_CUSTOM_SETTING_NAME	Invalid Custom Setting Name.
INVALID_CUSTOM_TAB_PROPERTY	Custom Tab Property is invalid.
INVALID_CUSTOM_TAB_TYPE	Custom Tab Type is invalid.
INVALID_CUSTOM_TAB_USER_ID	Custom Tab User Id is invalid.
INVALID_DISTRIBUTOR	The Distributor specified is not valid.
INVALID_DOCUMENT_ID	The DocumentId provided is invalid.
INVALID_DOCUMENT_ORDER	Invalid Document Order.
INVALID_DOCUMENT_SEQUENCE	Document sequence is invalid.
INVALID_EMAIL_ADDRESS	Invalid email address.
INVALID_EMAIL_ADDRESS_FOR_RECIPIENT	The email address for the recipient is invalid. The recipient Id follows.
INVALID_EMAIL_ADDRESS_FOR_SENDER	The email address for the sender is invalid.
INVALID_ENVELOPE_EMAILBLURB	Invalid Envelope Email Blurb.
INVALID_ENVELOPE_SUBJECT	Invalid Envelope Subject.
INVALID_EXPIRATION_REMINDER_DATA	Invalid envelope expiration and/or reminder information provided.
INVALID_FAX_RECIPIENT_SIGNERCERTREQUIREMENT	Invalid request for Fax token with a PKI signer certificate requirement.
INVALID_FAXNUMBER	Fax Number is invalid.
INVALID_FOLDER_FILTER	Invalid value specified for the folder filter.
INVALID_FOLDER_ID	A folder ID is missing or invalid.
INVALID_FOLDER_TYPE	The folder type specified is not valid.
INVALID_GROUP_ID	The MemberGroupId provided is invalid.
INVALID_GROUP_NAME	The group name is invalid.

Error Code	Message (Base)
INVALID_INCLUDED_SEATS_VALUE	Included seats must be greater than zero and not less than the minimum included seats.
INVALID_INVOICE_ID	Invoice Id is not valid
INVALID_LIST_VALUE	Value specified is not part of the list item values.
INVALID_LOGIN_STATUS	Invalid login status.
INVALID_MEMBER_GROUP_ID	Invalid Member Group ID
INVALID_MERGEUSERID	Invalid MergeUserId.
INVALID_MULTI_PART_REQUEST	An error was found while parsing the multipart request.
INVALID_OAUTH2_SAML_ASSERTION	Invalid Grant
INVALID_PAGE_TRANSACTIONID	The Page TransactionId is invalid.
INVALID_PARENT_FOLDER_ID	Invalid parent folder id.
INVALID_PASSWORD	The password is invalid.
INVALID_PASSWORD_CHALLENGE	Invalid forgotten password challenge.
INVALID_PAYMENT_ID	The Payment Id is not valid
INVALID_PERMISSION_PROFILE_ID	The SettingsTemplateId provided is invalid.
INVALID_PERMISSION_PROFILE_NAME	The permission profile name is invalid or already in use.
INVALID_PLAN_CURRENCY	The plan is not offered in the requested currency.
INVALID_PLAN_FEATURESET	The feature set is not configured for the requested plan.
INVALID_PLANID	The PlanId specified is not valid.
INVALID_PUT_RECIPIENT_SIGNATURE_ATTRIBUTE	Invalid attribute specified for updating recipient signature.
INVALID_RECIPIENT_EMAILBLURB	Invalid Recipient Email Blurb.
INVALID_RECIPIENT_ID	A recipient ID is missing or invalid.
INVALID_RECIPIENT_SETTING_VALUE	Recipient setting value is not an allowed value.
INVALID_RECIPIENT_STATUS_FOR_CORRECT	Invalid recipient status for correct.
INVALID_RECIPIENT_STATUS_FOR_DELETE	Invalid recipient status for delete.
INVALID_RECIPIENT_SUBJECT	Invalid Recipient Subject.
INVALID_RELATED_FIELD	Invalid related field sequence.
INVALID_REQUEST_BODY	The request body is missing or improperly formatted.
INVALID_REQUEST_PARAMETER	The request contained at least one invalid parameter.
INVALID_REQUEST_RESPONSE_OVERRIDE_FORMAT	The URL override format is not supported. Use supported formats such as .json or .xml.
INVALID_SALESFORCE_CREDENTIALS	Invalid Salesforce credentials.
INVALID_SESSION_TIMEOUT_VALUE	The Session Timeout value is not valid.
INVALID_SIGNATURE_STATUS	Invalid status specified for signature.

Error Code	Message (Base)
INVALID_SOCIAL_ACCOUNT	Social account information provided is invalid.
INVALID_SOCIAL_INFORMATION	The Social Provider information specified is not valid.
INVALID_SOCIAL_PROVIDER	Social Provider is not valid.
INVALID_SOCIAL_USER_NAME	Social Provider User Name is not valid.
INVALID_STATE	The State/Province is not valid in the system.
INVALID_TAB_FORMULA	The Tab formula is not valid.
INVALID_TAB_OPERATION	The Tab specified is not valid for the requested operation.
INVALID_TABID	Invalid TabId specified.
INVALID_TOKEN	Invalid_Token.
INVALID_TOKEN_FORMAT	The security token format does not conform to expected schema.
INVALID_URI	The URI is not valid.
INVALID_USER_OFFSET	The specified User Offset exceeds the page boundaries.
INVALID_USERCUSTOMTABID	UserCustomTabId is invalid.
INVALID_USERID	Invalid UserId.
INVALID_USERNAME	The user name is invalid.
INVALID_USERNAME_FOR_RECIPIENT	The user name for the recipient is invalid.
INVALID_WATERMARK	The watermark is invalid.
INVALID_WATERMARK_PREVIEW	There is an error generating Watermark preview.
INVALID_WORKSPACE_NAME	Workspace name is invalid.
INVALIDAUTHENTICATIONSETUP	Authentication is not setup correctly for the recipient.
INVOICE_DOES_NOT_EXIST_IN_SYSTEM	The InvoiceID did not identify an Invoice in the system.
INVOICE_PDF_NOT_AVAILABLE	The Invoice PDF is not available.
JANRAIN_REQUEST_ERROR	Janrain request Error.
LENGTH_EXCEEDED_ADDRESSLINE1	Address Line 1 length exceeded.
LENGTH_EXCEEDED_ADDRESSLINE2	Address Line 2 length exceeded.
LENGTH_EXCEEDED_CITY	City length exceeded.
LENGTH_EXCEEDED_COUNTRY	Country length exceeded.
LENGTH_EXCEEDED_CREDIT_CARD_NAME	Credit Card Cardholder Name length exceeded.
LENGTH_EXCEEDED_EMAIL	Email length exceeded.
LENGTH_EXCEEDED_FAX	Fax length exceeded.
LENGTH_EXCEEDED_FIRSTNAME	First Name length exceeded.
LENGTH_EXCEEDED_LASTNAME	Last Name length exceeded.
LENGTH_EXCEEDED_PHONE	Phone length exceeded.

Error Code	Message (Base)
LENGTH_EXCEEDED_POSTALCODE	Postal Code length exceeded.
LENGTH_EXCEEDED_RECIPIENT_CUSTOMFIELD1	CustomField1 maximum length Exceeded.
LENGTH_EXCEEDED_RECIPIENT_CUSTOMFIELD2	CustomField2 Maximum length Exceeded.
LENGTH_EXCEEDED_RECIPIENT_CUSTOMFIELD3	CustomField3 maximum length Exceeded.
LENGTH_EXCEEDED_RECIPIENT_EXTERNALUSERNAME	ExternalUserName maximum length Exceeded.
LENGTH_EXCEEDED_STATE	State/Province length exceeded.
MAX_ACCOUNTS_EXCEEDED	The maximum number of accounts was exceeded.
MAX_BCC_ADDRESSES_EXCEEDED	The maximum number of BCC email addresses has been exceeded.
MAX_CONNECT_CUSTOM_CONFIGURATION_EXCEEDED	Maximum number of connect custom configuration exceeded.
MAX_MEMBERS_EXCEEDED	The maximum number of members for the account has been exceeded.
MAX_SALE_DISCOUNT_PERIODS_EXCEEDED	The maximum sale discount period has been exceeded.
MAX_TABS_EXCEEDED	The maximum number of tabs was exceeded.
MESSAGE_LOCKED	Message is locked.
METHOD_NOT_ALLOWED	The HTTP method specified is not allowed with this resource.
METHOD_NOT_IMPLEMENTED	The method specified is not implemented.
MOBILE_NOTIFIER_CONFIG_FAILED	Mobile notifier configuration failed.
MULTIPLESATABSONDOCUMENT	More than one signer attachment tab is placed on a signer attachment document for a recipient.
MUST_DELETE_SUBFOLDERS	You must delete all sub-folders first before you can delete this folder.
MUST_ENABLE_SIGNONPAPER	EnableWetSign must be set on the envelope.
MUST_MOVE_ENVELOPES_BEFORE_DELETE	You must move all of the envelopes before you can delete this folder.
NAR_NOT_ENABLED_FOR_USER	NAR not enabled for user.
NAR_UNABLE_TO_VALIDATE_CREDENTIALS	NAR not able to validate NRDs ID and NRDs Last Name.
NAR_USER_NOT_ACTIVE	NAR user not active.
NO_DOCUMENT_RECEIVED	The document element did not contain the encoded document, or there is a problem with the encoding.
NO_SECURITY_APPLIANCES_CONFIGURED	No Security Appliances configured on account.

Error Code	Message (Base)
NOT_ELIGIBLE_FOR_RENEWAL_CANCELLATION	Not eligible for renewal cancellation.
NOT_SUPPORTED	Functionality not supported.
OAUTH_ERROR	An OAuth2 error occurred.
OFFLINE_SIGNING_CLIENTUSERID_REQUIRED	ClientUserId is required for offline signer.
OFFLINE_SIGNING_DATETIME_INVALID	Invalid value specified for offline DateTime.
OFFLINE_SIGNING_DELIVERY_METHOD_MUST_BE_SIGNER	DeliveryMethod of offline is only allowed for signer recipients.
OFFLINE_SIGNING_DOCUMENTS_REQUIRED	Documents are required for offline signing.
OFFLINE_SIGNING_DUPLICATE_RECIPIENTS	Must have distinct email and username for all offline recipients.
OFFLINE_SIGNING_INVALID_DRAFT_ENVELOPE	Draft envelopes are not allowed for offline signing.
OFFLINE_SIGNING_INVALID_SIGNER_ROUTING_ORDER	Invalid offline signer routing order.
OFFLINE_SIGNING_INVALID_TAB_TYPE	Invalid tab type specified for offline signing.
OFFLINE_SIGNING_NOT_ALLOWED	Offline signing not allowed.
OFFLINE_SIGNING_SIGNEDDATETIME_REQUIRED	'signedDateTime' must be specified for offline recipients.
OFFLINE_SIGNING_SIGNER_TABS_REQUIRED	Offline signer must have at least one signature or initials tab.
OFFLINE_SIGNING_TAB_IMAGE_REQUIRED	Signature image not provided for offline signature or initials tab.
ONESIGNALLSIGN_NOT_SATISFIED	Freeform signing is not allowed for your account because it conflicts with other settings, please place signing tabs for each signer.
PAGE_IMAGE_NOT_FOUND	The page image specified was not found.
PAGE_IMAGE_NOT_RETRIEVED	The page image is not available.
PARTNER_APP_CREATION_NOT_ALLOWED	Partner application creation not allowed in this environment.
PARTNER_APP_NAME_REQUIRED	Partner application Name is required.
PARTNER_APP_REDIRECT_URI_INVALID	Partner application Redirect URI is invalid.
PARTNER_AUTHENTICATION_FAILED	The specified Integrator Key was not found or is disabled.
PARTNER_INVALID_CLIENT_ID	Invalid Client Id provided.
PARTNER_NO_AUTHORITY_FOR_SENDER	The sender specified is not part of a group that the Partner is authorized to send for.
PAYMENT_DOES_NOT_EXIST_IN_SYSTEM	The Payment does not exist in the system.
PAYMENT_NOT_VALID	The Payment is not valid.
PDF_INACCESSIBLE	PDF URL Inaccessible
PDF_NOT_FOUND	PDF was not found.
PERMISSION_PROFILE_ID_DOES_NOT_MATCH_PERMISSION_PROFILE_NAME	Either permission profile Id or permission profile name is invalid is invalid.

Error Code	Message (Base)
PLAN_ITEM_NOT_ENABLED	A requested plan item is not enabled for this account.
PLAN_START_DATE_DOES_NOT_EXIST_IN_SYSTEM	The Current Account Plan Start Date does not exist in the system.
POWERFORMS_DIGITALCERTS_FREE_FORM_TABS_NOT_ALLOWED	Signers that are required to use a digital certificate must have at least one required, non-conditional Signature or Initials tab.
POWERFORMS_DIGITALCERTS_MARKUP_NOT_ALLOWED	Document markup is not allowed because a digital certificate is required for a signer.
POWERFORMS_DIGITALCERTS_MULTIPLE_RECIPIENTS_ROUTING_ORDER	Signers that are required to use a digital certificate must be the only recipient in a routing order. Please edit the routing order or remove the digital certificate requirement.
POWERFORMS_DIGITALCERTS_SHARED_TABS_NOT_ALLOWED	Shared tags are not allowed because a digital certificate is required for a signer.
POWERFORMS_DIRECT_NOT_ENABLED	Direct signing mode is not enabled for PowerForms.
POWERFORMS_INCOMPLETE_RECIPIENT	Recipient UserName, Email or Role not set.
POWERFORMS_INVALID_ENVELOPEID_FOR_RECIPIENT	Invalid envelope for recipient.
POWERFORMS_NOT_ADMIN	User lacks permission, not a PowerForms admin.
POWERFORMS_NOT_ENABLED	PowerForms is not enabled on the account.
POWERFORMS_NOT_USER	User lacks permission, not a PowerForms user.
POWERFORMS_POWERFORMID_MISMATCH	PowerFormId mismatch.
POWERFORMS_POWERFORMID_NOT_FOUND	The PowerForm specified was not found
POWERFORMS_POWERFORMID_REQUIRED	PowerFormId is required.
POWERFORMS_RECIPIENT_DENIED_DOCUMENTS	Recipient denied access to documents.
POWERFORMS_SIGNINGMODE_REQUIRED	Signing mode is required.
POWERFORMS_TEMPLATE_FIRST_RECIPIENT	Invalid template, the first recipient in the routing order must be a role recipient.
POWERFORMS_TEMPLATEID_REQUIRED	TemplateId is required.
POWERFORMS_USESREMAINING_REQUIRED	Uses remaining is required when max use is enabled.
PURCHASED_ENVELOPES_NOT_ADDED	Purchased envelopes not added.
RECIPIENT_ALREADY_EXISTS_IN_ENVELOPE	This recipientId already exists.
RECIPIENT_AUTHENTICATION_REQUIRED	Recipient has not passed authentication steps.
RECIPIENT_CORRECT_SUCCESS_DECLINE_TO_SIGN_FAILED	Recipient correct succeed, but Decline to Sign failed.
RECIPIENT_DELETE_FAILED	The recipient could not be deleted.

Error Code	Message (Base)
RECIPIENT_HAS_FAILED_SECURITY_CHECK	The requested user has failed the security check for the requested envelope. Correct the envelope to continue.
RECIPIENT_HAS_INVALID_ROUTINGORDER	The specified recipient has invalid Routing Order.
RECIPIENT_MUST_BE_VALID_USER	Account settings indicate the recipient must be an active DocuSign user.
RECIPIENT_NO_ACCESS	Recipient does not have permission to access document.
RECIPIENT_NOT_ACCOUNTLESS	Recipient must not have an active account.
RECIPIENT_NOT_CAPTIVE	The specified recipient is not a captive recipient. This operation requires a captive recipient.
RECIPIENT_NOT_FOUND_FOR_CORRECTION	The specified User is not a recipient of the specified envelope.
RECIPIENT_NOT_IN_SEQUENCE	The token for an out of sequence recipient cannot be generated.
RECIPIENT_NOT_SIGNER	Recipient is not a signer type.
RECIPIENT_UPDATE_FAILED	The recipient could not be updated.
RECIPIENTS_LOCKED	Recipients are locked and cannot be modified.
RECIPIENTS_NOT_PROVIDED	No recipients were found in the request.
REDUNDANT_API_DOCUMENT_ACCESS	Redundant API document access violation
REGULAREXPRESSION_IS_INVALID	The regular expression provided is not valid.
REPORT_DATA_INVALID	Report data is invalid.
REPORT_DATE_INVALID	No date value was provided or the date provided was null.
REPORT_DOES_NOT_EXIST	The Enterprise Report being requested does not exist.
REPORT_DUPLICATE_NAME_EXISTS	You have already saved a report by this name.
REPORT_EXECUTION_FAILED	The Enterprise Report failed to execute. See additional information for more details.
REPORT_MAX_SAVED_REPORTS_EXCEEDED	You have reached the maximum number of saved reports. Please deactivate one and try again.
REPORT_MAX_SCHEDULED_EXCEEDED	You have reached the maximum number of scheduled reports. Please deactivate one and try again.
REPORT_MISSING_REQUIRED_PARAMETER	Report is missing a required parameter.
REPORT_MULTIPLE_MATCHES_FOUND	The Enterprise Report could not be executed because more than one report match the query/filter provided.

Error Code	Message (Base)
REPORT_QUERY_INVALID	No Enterprise Report data could be returned because one or more query/filter parameters was invalid.
REPORT_QUERY_PARAMETER_TYPES_INCONSISTENT	Query parameter types are not consistent.
REPORT_QUERY_PARAMETERS_EXCEED_EXPECTED	Too many query parameters specified.
REPORT_REQUIRED_PARAMETER_NOT_SET	A parameter is not set for the report.
REPORT_SCHEDULE_ALREADY_EXISTS	Report schedule already exists.
REPORT_SCHEDULE_CREATION_FAILED	Report schedule could not be created.
REPORT_SCHEDULE_INVALID_OPTION	Invalid option for report schedule.
REPORT_SCHEDULE_SCHEDULE_FAILED	Report schedule failed.
REPORT_SUBSCRIPTION_FAILED	Report subscription failed.
REPORT_TIMEZONE_INVALID	The timezone offset specified is invalid.
REPORT_XML_INVALID	Report Xml is invalid.
REQUIRE_ALL_SHARED_TAB_NOT_ENABLED	Allow Shared Tabs is not enabled on account.
REQUIRE_ALL_SHARED_TAB_NOT_SET	Shared must be set to 'true' when setting 'Require All' to 'true'.
REQUIRED_ADDRESS	The address is required.
REQUIRED_ADDRESSLINE1	Address Line 1 is required.
REQUIRED_CITY	City is required.
REQUIRED_COUNTRY	Country is required.
REQUIRED_CREDIT_CARD_NAME	Credit Card Cardholder Name is required.
REQUIRED_EMAIL	Email is required.
REQUIRED_FIRSTNAME	First Name is required.
REQUIRED_LASTNAME	Last Name is required.
REQUIRED_POSTALCODE	Postal Code is required.
REQUIRED_SECURITY_CHECK_URL_MISSING	A security check was specified for the recipient but the appropriate callback URL was not provided.
REQUIRED_STATE	State/Province is required.
REQUIRED_TAB_INCOMPLETE	A Required field is incomplete.
RESOURCE_NOT_FOUND	The URL provided does not resolve to a resource.
ROLE_DOES_NOT_EXIST	The role does not exist.
SAML_AUTHENTICATION_NOT_ALLOWED	SAML Authentication not allowed.
SATABONNONSADOCUMENT	A signer attachment tab is placed on a non-signer attachment document (a document without an AttachmentDescription).
SENDER_DOES_NOT_EXIST_IN_SYSTEM	The UserName and Email did not identify a sender in the system.

Error Code	Message (Base)
SENDER_REQUIRED_FIELD_CANNOT_BE_DELETED	Sender required fields cannot be deleted from envelope.
SENDER_REQUIRED_FIELD_NOT_ALLOWED	Property 'sender required' can only be set/modified on 'Text' or 'Drop Down' tags on a template.
SENDER_REQUIRED_FIELD_NOT_SATISFIED	Not all 'sender required fields' have been populated.
SHARED_TEMPLATE_NOT_CREATOR	Shared Templates can only be moved by the creator of the template.
SHARED_VIEW_LINK_NOT_ALLOWED	Shared view links not allowed.
SIGNATURE_NOT_FOUND	The signature specified was not found.
SIGNER_CERTIFICATE_PROVIDER_NOT_SET	PKI Certificate Provider is not set.
SIGNER_CERTIFICATE_RECIPIENT_TYPE_INVALID	RequireSignerCertificate can only be used for Recipient Type Signer and InPersonSigner
SIGNER_CERTIFICATE_REQUIRED_TAB_ERROR	RequireSignerCertificate specified but no required SignHere or InitialHere tabs are present.
SIGNER_CERTIFICATE_ROUTING_ORDER_ERROR	When RequireSignerCertificate is specified, there must not be another recipient at the same routing order.
SIGNER_CERTIFICATE_SETTINGS_CONFLICT	RequireSignerCertificate conflicts with another envelope setting.
SIGNER_CERTIFICATE_TYPE_NOT_ENABLED	Signer certificate type not enabled for account.
SIGNER_CERTIFICATE_TYPE_NOT_SUPPORTED	Signer certificate type not supported.
SIGNER_RECIPIENT_HAS_NO_SIGNABLE_TABS	One or more Signer type recipients have not been assigned any signable tabs.
SIGNING_GROUP_CAPTIVE_RECIPIENT_NOT_ALLOWED	Captive recipient not allowed to be a signing group.
SIGNING_GROUP_FAX_DELIVERY_NOT_ALLOWED	Fax delivery is not allowed with signing groups.
SIGNING_GROUP_INVALID	Invalid signing group supplied.
SIGNING_GROUP_MAX_REACHED	Maximim signing groups reached.
SIGNING_GROUP_MAX_USERS_REACHED	Maximum users in a signing group reached.
SIGNING_GROUP_SMS_AUTH_NOT_ALLOWED	SMS authentication is not allowed with signing groups.
SMS_AUTHENTICATION_NOT_ALLOWED	SMS Authentication not allowed.
SOBO_USER_NOT_ALLOWED	A 'Send on Behalf Of' user is not permitted for this operation.
SOCIAL_LOGIN_NOT_ENABLED	Social login not enabled on account.
SOCIAL_PROVIDER_ALREADY_MAPPED	This social provider account is already mapped in the system.
SUBSCRIPTION_ERROR	There is an error with the subscription.

Error Code	Message (Base)
SUBSCRIPTION_FAILED	The subscription failed.
SUBSCRIPTION_NOT_FOUND	The subscription was not found.
TAB_MERGEFIELD_VALUE_NOT_RETRIEVED	A Mergefield value for a Tab could not be retrieved.
TAB_NOT_FOUND	Tab specified was not found.
TAB_OUT_OF_BOUNDS	Tab is placed off of the page.
TAB_PAGE_NUMBER_NOT_SPECIFIED	Page number not specified in tab element.
TAB_PAGENUMBER_IS_NOT_IN_DOCUMENT	The pagenumber specified in the tab element is not in the document that the tab refers to.
TAB_REFERS_TO_MISSING_DOCUMENT	The DocumentId specified in the tab element does not refer to a document in this envelope.
TAB_REFERS_TO_MISSING_RECIPIENT	The RecipientId specified in the tab element does not refer to a recipient of this envelope.
TEMPLATE_ADDITIONALTABS_ERROR	Unable to add AdditionalTabs from Templates to Envelope.
TEMPLATE_AUTHENTICATION_FAILED	Not authorized to change template.
TEMPLATE_CANNOT_ADD_DOCUMENT	Cannot add the document for the template.
TEMPLATE_CANNOT_USE_ENVELOPEID_FOR_TEMPLATE_OPERATION	Cannot use an envelope id for a template operation.
TEMPLATE_DOCUMENTID_MAPPING_INVALID	The document ID has already been used.
TEMPLATE_ID_INVALID	Invalid template ID.
TEMPLATE_MERGE_INVALID_ROUTE_ORDER	The routing order of the recipients is not valid.
TEMPLATE_NOT_PROVIDED	Template was not provided.
TEMPLATE_OVERRIDE_ENVELOPEINFORMATION_ERROR	Unable to override Envelope data from EnvelopeInformation.
TEMPLATE_OVERRIDE_RECIPIENTDATA_ERROR	Unable to override recipient data from Recipients to Envelope.
TEMPLATE_PASSWORD_REQUIRED	This user lacks sufficient permissions to access this resource. A password is required to edit this template.
TEMPLATE_RECIPIENTID_FOR_ROLE_NOTFOUND	Recipient ID not found in the Recipient list for the role.
TEMPLATE_REQUIRED_RECIP_NOT_SATISFIED	Required recipient in the template has not been provided.
TEMPLATE_ROLESPECIFIED_DOES_NOT_EXIST	Role does not exist in the template.
TEMPLATE_TO_ENVELOPE_ERROR	Unable to transform Template and data to Envelope.
TEMPLATE_UNABLE_TO_FLATTEN_PDF	Unable to flatting PDF for the Template.
TEMPLATE_UNABLE_TO_LOAD	Unable to load template.
TEMPLATE_UNABLE_TO_LOAD_FIELDDATA	Unable to load field data from PDF.
TEMPLATE_UNABLE_TO_PROCESS_FIELDDATA	Unable to process field data for the Template.

Error Code	Message (Base)
TEMPLATE_UPLOAD_NOT_ALLOWED	Template upload not allowed.
TRANSACTIONID_DOES_NOT_EXIST	The transaction ID does not exist.
TRANSACTIONID_REDUNDANT	The transaction ID has already been submitted.
TRANSACTIONID_UNABLE_TO_SUBMIT	Unable to submit the transaction for processing.
UNABLE_TO_CLONE_ENVELOPE	Unable to clone the specified envelope.
UNABLE_TO_CONVERT_DOCUMENT	System was unable to convert this document to a PDF.
UNABLE_TO_DELETE_DOCUMENT	Unable to delete document.
UNABLE_TO_LOAD_DOCUMENT	Unable to load the document.
UNABLE_TO_PURGE_DOCUMENTS	Unable to purge documents.
UNKNOWN_ENVELOPE_RECIPIENT	The recipient you have identified is not a valid recipient of the specified envelope.
UNSPECIFIED_ERROR	An Error Occurred.
UNSPECIFIED_ERROR_IN_BILLING_SYSTEM	An error occurred in the billing system.
UNSPECIFIED_ERROR_PROCESSING_PAYMENT	A payment processing error occurred in the billing system.
USER_ALREADY_EXISTS_IN_ACCOUNT	Username and email combination already exists for this account.
USER_AUTHENTICATION_FAILED	One or both of Username and Password are invalid.
USER_AWAITING_ACTIVATION	This user name and email are awaiting activation.
USER_DOES_NOT_BELONG_TO_SPECIFIED_ACCOUNT	The specified User is not a member of the specified Account.
USER_DOES_NOT_EXIST_IN_SYSTEM	The UserID did not identify a User in the system.
USER_LACKS_MEMBERSHIP	The UserID does not have a valid membership in this Account.
USER_LACKS_PERMISSIONS	This User lacks sufficient permissions.
USER_LACKS_RECIPIENTEMAILNOTIFICATION_PERMISSION	Account or user does not have permission to set recipient email notifications.
USER_MERGE_FAILED	The user merge failed
USER_NOT_ACCOUNT_ADMIN	User is not an account administrator.
USER_NOT_ENVELOPE_OWNER	This user lacks sufficient permissions to access this resource. The user is not the owner of the envelope.
USER_NOT_ENVELOPE_SENDER	This user is not the sender of the envelope. Only the sender of the envelope may perform the requested operation.

Error Code	Message (Base)
USER_NOT_ENVELOPE_SENDER_OR_RECIPIENT	This user is not the sender or a recipient of the envelope. Only the sender or a recipient of the envelope may perform the requested operation.
USER_NOT_FOLDER_OWNER	The user does not own the folder
USER_NOT_FOUND	No User was found for given criteria.
USER_NOT_RECIPIENT	User is not a recipient.
USER_NOT_TEMPLATE_OWNER	This user lacks sufficient permissions to access this resource. The user is not the owner of the template.
USERNAME_IS_NOT_ALLOWED	The specified username(s) and email(s) are not in our system and the email does not allow new users.
USERS_IN_PERMISSION_PROFILE	There are users still associated to this permission profile.
WATERMARK_DISABLED	Watermark is disabled.
WATERMARK_UPDATE_FAILED	Watermark could not be updated.
WORKSPACE_DOES_NOT_EXIST	The workspace does not exist or permission denied.

Appendix 1: DocuSign SOAP API to REST API

If you already use the DocuSign API you might be able to leverage some of your existing code and experiences for use with the REST API. The following table shows how DocuSign SOAP API methods related to REST URIs.

DocuSign SOAP	REST URI and HTTP Method
AddMembersToAccount (AcctMgmt)	/accounts/{accounted}/users POST
ChangeAccountPricePlan (AcctMgmt)	/accounts/{accountsId}/billing_plan PUT
ChangePassword (AcctMgmt)	/login_information/password PUT
CheckAccountMember (AcctMgmt)	/accounts/{accountId}/users?email=<email> GET
CloseSignature (AcctMgmt)	/accounts/{accountId}/users/{userId}/signatures/{signatureName} DELETE
CorrectAndResendEnvelope (dsapi)	/accounts/{accountId}/envelopes/{envelopeId}/recipients PUT
CreateAndSendEnvelope (dsapi)	/accounts/{accountId}/envelopes POST
CreateEnvelope (dsapi)	/accounts/{accountId}/envelopes POST
CreateEnvelopeFromTemplates (dsapi)	/accounts/{accountId}/envelopes POST
DeleteEnvelopes (dsapi)	/accounts/{accounted}folders/{folderId} Use recyclebin as folderId PUT
GetAccountDistributorCode (AcctMgmt)	/accounts/{accountId} GET
GetAccountInformation (AcctMgmt)	/accounts/{accountId} GET
GetAccountSettings (AcctMgmt)	/accounts/{accountId}/settings GET
GetAccountSettingsList (dsapi)	/accounts/{accountId}/settings GET
GetAuthetication Token (Cred, dsapi)	/accounts/{accountId}/views/console POST
GetFolderItems (dsapi)	/accounts/{accounted}folders/{folderId} GET
GetFolderList	/accounts/{accounted}folders GET
GetMemberSettings (AcctMgmt)	/accounts/{accountId}/users/{userId}/settings GET

DocuSign SOAP	REST URI and HTTP Method
GetPlanGroupInformation (AcctMgmt)	/billing_plans GET
GetPlanPricingInformation (AcctMgmt)	/billing_plans/{planId} GET
GetProvisioningInformation (AcctMgmt)	/accounts/provisioning GET
GetRecipientList (dsapi)	/accounts/{accountId}/recipient_names?email=<email address> GET
GetSuccessorPlanInformation (AcctMgmt)	/accounts/{accounted}/billing_plan GET
GetUserProfile (AcctMgmt)	/accounts/{accountId}/users/{userId}/profile GET
GetUserProfileImage (AcctMgmt)	/accounts/{accountId}/users/{userId}/profile/image GET
Login (Cred)	/login_information GET
MoveEnvelopes (dsapi)	/accounts/{accounted}folders/{folderId} PUT
NewAccount (AcctMgmt)	/accounts POST
NewSocialAccount (AcctMgmt)	/accounts POST
RequestCertificate (dsapi)	/{accountId}/envelopes/{envelopeId}/documents/certificate GET
RequestCorrectToken (Cred)	/accounts/{accountId}/envelopes/{envelopeId}/views/correct POST
RequestEnvelope (dsapi)	/accounts/{accountId}/envelopes/{envelopeId} GET
RequestPDF (dsapi)	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId} GET
RequestPDFNoWaterMark (dsapi)	/accounts/{accountId}/envelopes/{envelopeId}/documents/combined watermark=false GET
RequestPDFWithCert (dsapi)	/accounts/{accountId}/envelopes/{envelopeId}/documents/combined GET
RequestRecipientToken (dsapi)	/accounts/{accountId}/envelopes/{envelopeId}/views/recipient POST
RequestSenderToken (Cred)	/accounts/{accountId}/envelopes/{envelopeId}/views/sender POST
RequestStatus (dsapi)	/accounts/{accountId}/envelopes/{envelopeId}/recipients GET
RequestStatus (dsapi)	/accounts/{accountId}/envelopes/{envelopeId} GET

DocuSign SOAP	REST URI and HTTP Method
RequestStatusChanges (dsapi)	/accounts/{accountId}/envelopes
RequestStatusCodes (dsapi)	GET
RequestTemplate (dsapi)	/accounts/{accountId}/templates/{templateId}
	GET
RequestTemplates (dsapi)	/accounts/{accountId}/templates
	GET
SaveTemplate (dsapi)	/accounts/{accountId}/templates
	POST
SendEnvelope (dsapi)	/accounts/{accountId}/envelopes/{envelopeId}/status
	Add {“status”:”sent”} in request body.
	PUT
SetSignatureImage (AcctMgmt)	/accounts/{accountId}/users/{userId}/signatures/{signatureName}
	PUT
SetUserProfile (AcctMgmt)	/accounts/{accountId}/users/{userId}/profile
	PUT
SetUserProfileImage (AcctMgmt)	/accounts/{accountId}/users/{userId}/profile/image
	PUT
UpdateAccountSettings (AcctMgmt)	/accounts/{accountId}/settings
	PUT
UpdateMemberSettings (AcctMgmt)	/accounts/{accountId}/users/{userId}/settings
	PUT
VoidEnvelope (daspi)	/accounts/{accountId}/envelopes/{envelopeId}
	Add {“status”:”voided”, “statusReason”:”voided reason”} in request body.
	PUT

Appendix 2: Time Zones and Date/Time Format Information

This appendix provides information on how time zone and date/time format information is set for an account and how it appears in the DocuSign web console, Certificate of Completion, and API responses. The appendix also has information on the expected date/time formats for API requests.

Account Settings

There are a number of account settings that determine the time zone and date/time format used for an account. These options are set in the DocuSign web console Preferences – Features; some can be set through the API account settings.

- **Default Account Time Zone:** This option selects the default time zone to be used in the User Interface display. A member of the account can set their own time zone preference to override the account default if the Allow account member to set their own unique time zone option (see below) is enabled.
- **Time Zone Used For API:** This option selects the default time zone used for DocuSign SOAP API operations.
- **Format for Date Signed:** This option selects the date format applied to the Date Signed tags.
- **Format for Time Signed:** This option selects the time format applied to the Date Signed tags. You can also select to **Include AM/PM designation** for the time format.
- **Allow account member to set their own unique time zone:** When selected, account members can set their own personal time zone, which can be different that the default account time zone.

If the **Allow account member to set their own unique time zone** option is selected, there are two options the account members can set by a user:

- **My local time zone:** This option selects the default time zone to be used in the account member's DocuSign web console.
- **Format my local date and time:** This option selects the date/time format used in the account member's DocuSign web console and the format used when the account member signs envelopes.

Time Zone and Date/Time Format Appearance

How the time zone and date/time format appears depends on what is being viewed, who is viewing the information, and the settings for the account.

Signing user interface and PDF documents: The time zone and date/time format information shown in Date tags and form data in the signing user interface and PDF documents depends on the signer:

- If the signer has a DocuSign account, then the signer's time zone and the sender's date/time format settings are used for the information. If the sender's account does not allow users to set their own date/time format, then the date/time format setting for the account is used.
- If the signer does not have a DocuSign account, then the sender's time zone and date/time format settings are used. If the sender's account does not allow users to set their own time zones or date/time format, then the account settings are used.

Certificate of Completion: The sender's time zone and date/time format settings are used. If the sender's account does not allow users to set their own time zone and date/time format, then the account time zone and date/time format settings are used.

Envelope History: When viewing History through the web console or signing user interface, the time zone and date/time format used depends on the viewer:

- If the sender is viewing the History, the sender's time zone and date/time format settings are used. If the sender's account does not allow users to set their own time zones or date/time format, then the account settings are used.
- If the viewer has a DocuSign account that allows users to set their own time zone and date/time format, then the viewer's time zone and date/time format settings are used.
- If the viewer has a DocuSign account that does not allow users to set their own time zone and date/time format, then the account time zone and date/time format settings are used.
- If the viewer does not have a DocuSign account, then the sender's time zone and date/time format are used. If the sender's account does not allow users to set their own time zones or date/time format, then the account settings are used.

REST API: When getting information from the REST API, all non-PDF time zone responses are returned in ISO 8601 date/time format using GMT as the time zone. PDF items retrieved through the REST API, such as a Certificate of Completion or documents with date tags, use Certificate of Completion and PDF documents settings previously described.

SOAP API: When getting information from the SOAP API, all non-PDF items use the time zone set by the **Time Zone Used For API** option. PDF items retrieved through the SOAP API, such as a Certificate of Completion or documents with date tags, use Certificate of Completion and PDF documents settings previously described.

Date/Time Formats for API Calls

All DocuSign SOAP and REST API requests must use ISO 8601 date/time formats. The REST API assumes that all values passed represent UTC date/times.

When providing a date/time format for the DocuSign REST API, the preferred formats are:

- "yyyy-MM-dd | HH:mm"
- "MMMM d, yyyy | HH:mm"
- "MMM-dd-yyyy | HH:mm"
- "dd-MM-yyyy | HH:mm"