



The Global Standard for eSignature



DocuSign REST API Guide

Version 2

Copyright ©2003-2013 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, December 6, 2013

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:

- Added information about using [Certificate-Based Authentication](#) with the REST API.
- Updated [Login](#) information to include optional request parameters.
- Changed envelope [emailBlurb](#) limit from 2,000 characters to 10,000 characters.
- Added pages, fileExtension, documentBase64, and matchboxes parameters to the envelope [document parameter](#).
- Added containPdfFormFields parameters to the [GET List of Envelope Documents](#) response.
- Added ownerUserId parameter to the [GET Folder Lists](#) response
- Added optional notifications, envelopeTemplateDefinition-folderId, and envelopeTeplateDefinition-owner parameters to [POST Template](#) request and [GET Template](#) response.
- Added an optional query parameter that will close a user signature when [modifying-a-user signature](#).
- Added optional locked and required parameters to the [Radio Group Tab](#).

Table of Contents

Introduction.....	9
Commonly Used Terms	9
Basic Process Flow and Steps for Sending with the REST API	10
Basic Process Flow for Requesting Status Changes with the REST API	10
Request and Response Guidelines	12
Request and Response Formats.....	13
Envelope and Recipient Status Codes.....	13
Getting Started	14
Integrator Keys	14
API Call Rate Limits	15
File Size Limitations	15
REST API Versions.....	16
REST API Base URL	16
REST API Explorer	16
Send On Behalf Of Functionality in the DocuSign REST API.....	16
OAuth2 Authentication Support in DocuSign REST API	18
General Usage Notes	18
OAuth2 Token Request	18
Normal Usage.....	19
OAuth2 and SAML.....	20
Revoking Tokens	22
Acting as Other Account Users	23
OAuth2 Response Codes	25
Certificate-Based Authentication.....	26
Certificate-Based Authentication Requests and Responses.....	26
Basic Scenarios.....	28
Creating an Envelope from a Document	28
Creating an Envelope from a Template	29
Embedded Sending	31
Embedded Signing	31
Requesting Envelope Status.....	32
Retrieving Envelope and Documents	33
Sending HTTP Requests with cURL	34

Setting Up Your Client Application	34
Viewing Base Resources with cURL	35
Logging in with cURL	35
Sending HTTP Requests Using REST API Resources	36
Chunked Transfer-Encoding Support.....	37
Example Chunked Transfer-Encoding	37
REST API References.....	39
Login.....	50
Change Password	52
Create Account.....	53
Delete Account	62
Get Account Information	62
Get Account Billing Plan	63
Update Account Billing Plan.....	66
Purchase Additional Envelopes	68
Get Brand Profile Information.....	69
Upload Brand Profiles	70
Delete Brand Profiles.....	72
Delete Captive Recipient Signature	72
Set Up a Connect Configuration.....	74
Get Connect Configuration Information.....	77
Update a Connect Configuration	79
Get a Connect Configuration Information.....	82
Delete a Connect Configuration	83
Republish Connect Information for One Envelope	83
Republish Connect Information for Multiple Envelopes	84
Get the Connect Failure Log	85
Get a Connect Failure Log Entry	88
Clear a Connect Failure Log Entry	90
Get the Connect Log	91
Clear the Connect Log	92
Get One Connect Log Entry	93
Clear One Connect Log Entry	94
Get Consumer Disclosure.....	94
Get List of Account Custom Fields	96

Send an Envelope or Create a Draft Envelope	98
Send an Envelope from a Template	109
Get Envelope Status Changes	120
Get Individual Envelope Status	124
Send Draft Envelope	125
Void Envelope	126
Get Envelope Audit Events	127
Add Envelope Custom Fields to an Envelope	128
Get Envelope Custom Field Information	130
Modify Envelope Custom Fields for an Envelope	132
Remove Envelope Custom Fields from an Envelope	135
Add Documents to a Draft Envelope	136
Remove Documents from a Draft Envelope	137
Get List of Envelope Documents	138
Add a Document to a Draft Envelope	139
Get Document from Envelope	140
Get a Page Image	141
Rotate Page Image	142
Remove a Page	142
Get Custom Document Fields for an Envelope Document	143
Add Custom Document Fields to an Envelope Document	144
Modify Custom Document Fields for an Envelope Document	146
Delete Custom Document Fields from an Envelope Document	147
Get Envelope Certificate	149
Get Envelope Documents and Certificate	149
Get Envelope Notification Information	150
Get Envelope Recipient Status	151
Add Recipients to an Envelope	153
Delete Recipients from an Envelope	155
Modify or Correct and Resend Recipient Information	156
Set Initials Image for Accountless Signer	157
Get Signature Information for Accountless Signer	158
Set Signature Image for Accountless Signer	159
Add Tabs for a Recipient	160
Get Tab Information for a Recipient	161

Delete Tabs for a Recipient.....	162
Modify Tabs for a Recipient	163
Get List of Templates Used in an Envelope	164
Post Envelope Correction	165
Post Recipient View.....	166
Post Sender View	168
Get Envelope Status for more than one envelope	169
Get Folder List	170
Get Folder Envelope List	172
Move Envelope	174
Add a New Group	175
Get Group Information	176
Modify Group Information.....	177
Get Group Brand ID Information	178
Add Group Brand ID Information.....	179
Delete Group Brand ID Information.....	180
Add Users to a Group	181
Get List of Users in a Group.....	182
Remove Users from a Group	183
Get a List of Permission Profiles	185
Get Recipient Names.....	186
Get List of Envelopes in Folders	186
Get Account Settings	190
Modify Account Settings	191
Get List of Templates.....	192
Post Template.....	193
Get Template	202
Get Custom Document Fields for a Template Document	204
Add Custom Document Fields to a Template Document.....	205
Modify Custom Document Fields for a Template Document.....	206
Delete Custom Document Fields from a Template Document.....	208
Share a Template with a Group	210
Remove Template Sharing for a Group.....	211
Get List of Unsupported File Types	212
Add User to Account.....	213

Get User List.....	217
Close a User.....	218
Get User Information.....	220
Get Custom User Settings	221
Add or Modify Custom User Settings	222
Delete Custom User Settings.....	224
Get User Profile	225
Modify User Profile.....	228
Get User Profile Image	231
Modify User Profile Image.....	232
Delete User Profile Image	232
Get User Account Settings.....	233
Modify User Account Settings	234
Get a List of User Signature Definitions	235
Setting User Signature and Initials Images when Creating a Signature.....	236
Get User Signature Information	239
Modify a User Signature	240
Close a User Signature	242
Get a User Initials Image	242
Set a User Initials Image.....	243
Delete a User Initials Image	244
Get a User Signature Image	246
Set a User Signature Image.....	246
Delete a User Signature Image.....	248
Add a User Social Account	249
Get User Social Accounts	249
Remove a User Social Account	250
Post Authentication View	251
Get Account Provisioning Information	252
Get List of Billing Plans	253
Get Billing Plan Details	254
Revoke an Authorization Token	257
Create an Authorization Token	258
Recipient Types and Parameters.....	259
Tab Types and Parameters.....	322

Error Code Information	391
General Error Response Handling	391
Error Response Handling for API calls that Support Multiple Items	392
Error Codes and Associated Messages	392
Appendix 1: DocuSign SOAP API to REST API.....	404
Appendix 2: Time Zones and Date/Time Format Information	407
Account Settings.....	407
Time Zone and Date/Time Format Appearance	407
Date/Time Formats for API Calls.....	408

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

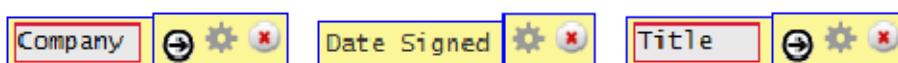
Definitions of some commonly used DocuSign terms are given below to familiarize you with their use in the DocuSign system.

- 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 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).



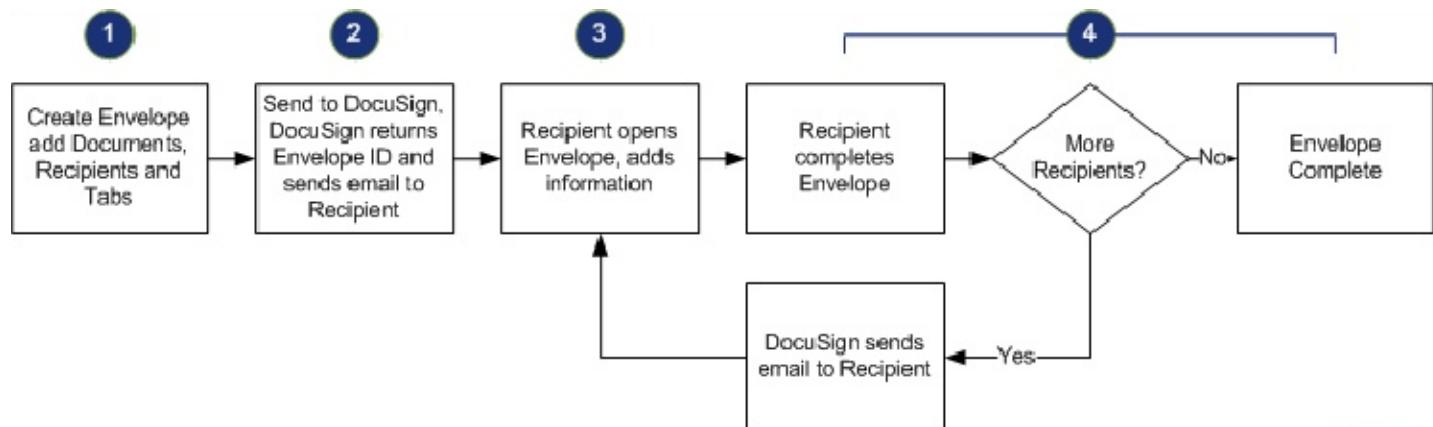
Third, it is used as editable information fields where signers can add data to a document.



- 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:



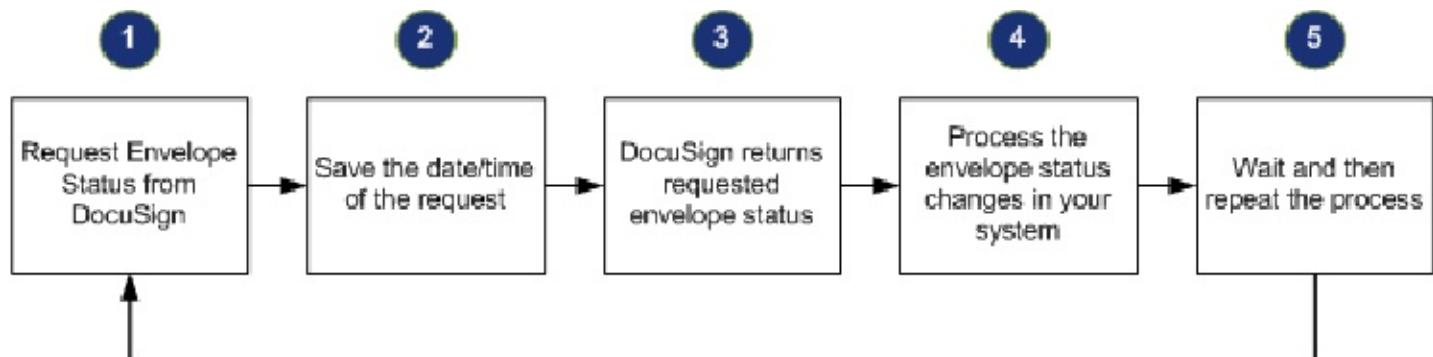
1. A sender supplies the name and email address of each person they want to sign the document (these are the recipients), adds documents to an envelope and adds tabs to indicate where each person should sign, initial or add other information.

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.



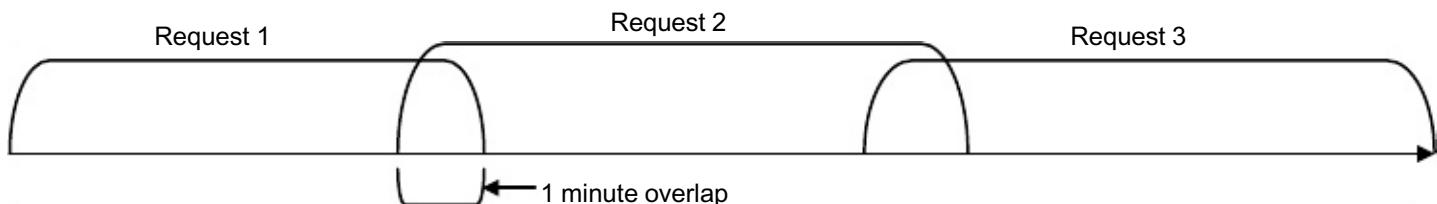
1. Request the envelope status: Use the GET method to request envelope status starting from a specified date and time. Optionally, you can just check for any changes to envelope status or for envelopes that changed to a particular status (this is highly recommended).

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 on 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 envelopes 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]/rest/api/[api Version]/accounts/[accountId]/envelopes
?fromDate=5-2-2012+12:15:00+AM&fromToStatus=changed
```

Request 2 at 12:30 would be:

```
GET https://[server]/rest/api/[api Version]/accounts/[accountId]/envelopes
?fromDate=5-2-2012+12:14:00+AM&fromToStatus=changed
```

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

```
GET https://[server]/rest/api/[api Version]/accounts/[accountId]/envelopes  
?fromDate=5-2-2012+12:29:00AM&fromToStatus=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 following date formats are accepted in the DocuSign REST API:
 - "M/d/yyyy"
 - "MMMM d, yyyy"
 - "dd-MM-yy"
 - "dd-MM-yyyy"
 - "MMM-dd-yyyy"
 - "MMMM d, yyyy | HH:mm"
 - "dd-MM-yy | HH:mm"
 - "dd-MM-yyyy | HH:mm"
 - "MMM-dd-yyyy | HH:mm"
 - "yyyy-MM-dd"
 - "yyyy-MM-dd | 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/v1/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.
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 of ongoing

support, certification help, and access to production. To join the program, or start the certification process, refer to the [certification steps](#) in the DocuSign Community.

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.

Additionally, 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 [Developer Forum in the DocuSign Community](#).

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.

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_info or /version 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 / {api Version}/`

The base URL for your account is retrieved by using the / login_info or /version 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/) (<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 functionality permits automated sending through the DocuSign API by an integrated sender on behalf of another sender. The Send On Behalf Of functionality is only available to accounts that use the DocuSign API to send envelopes and can be enabled for an account member by a DocuSign Customer Administrator through the DocuSign console or by contacting your DocuSign Account Manager.

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.

This allows an authenticating user to act as another user in the account, so the client application does not have to track the passwords for individual users. Instead, only the authenticating user's

information is used. The operations, usually sending an envelope or checking status, are performed as the “operating user,” not as the authenticating user. The authenticating user’s information is used for authentication only, while the “operating user’s” userId is checked for proper authorization for the API method being called.

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.

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]/rest/api/{api Version}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authorization:
<DocuSignCredentials><SendOnBehalfOf>bob.smith@mail.com</SendOnBehalfOf><User name>[name]</User name><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]/rest/api/{api Version}/accounts/{accountId}/envelopes/
{envelopeId}

X-DocuSign-Authorization:
<DocuSignCredentials><SendOnBehalfOf>bob.smith@mail.com Bob
Smith</SendOnBehalfOf><User name>[name]</User name><Password>[password]</Password><IntegratorKey>[integrator_key]</IntegratorKey>
</DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Example using UID

```
GET https://[server]/rest/api/{api Version}/accounts/{accountId}/envelopes/
{envelopeId}
```

```
X-DocuSign-Authorization:  
<DocuSignCredential><SendOnBehalfOf>P134325</SendOnBehalfOf><UserName>{name}</UserName>  
Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCr  
edential>  
Accept: application/json  
Content-Type: application/json
```

OAuth2 Authentication Support in DocuSign REST API

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. As a result, refresh_token is not supported.
- Up to 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.
- 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 / {api Version} / oauth2 / token
```

Accept: application/json

Content-Type: application/x-www-form-urlencoded

grant_type=password&client_id={Integrator Key}&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/{{api Version}}/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
    email address}}</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.
- SubjectConfirmationData/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 at the link:
<http://tools.ietf.org/html/rfc4648#page-7>

Example SAML Assertion

```
<saml:Assertion Version="2.0" ID="b4105a9b-5c0a-493a-beac-0759a8b5bed6"
IssuedInstant="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#sha1"/>
      <Reference URI="#_b4105a9b-5c0a-493a-beac-0759a8b5bed6">
        <Transforms>
          <Transform
            <XPath>/</XPath>
            <Transform
              <XPath>/</XPath>
              <Transform
                <XPath>/</XPath>
                <Transform
                  <XPath>/</XPath>
                  <Transform
                    <XPath>/</XPath>
                    <Transform
                      <XPath>/</XPath>
                      <Transform
                        <XPath>/</XPath>
                        <Transform
                          <XPath>/</XPath>
                          <Transform
                            <XPath>/</XPath>
                            <Transform
                              <XPath>/</XPath>
                              <Transform
                                <XPath>/</XPath>
                                <Transform
                                  <XPath>/</XPath>
                                  <Transform
                                    <XPath>/</XPath>
                                    <Transform
                                      <XPath>/</XPath>
                                      <Transform
                                        <XPath>/</XPath>
                                        <Transform
                                          <XPath>/</XPath>
                                          <Transform
                                            <XPath>/</XPath>
                                            <Transform
                                              <XPath>/</XPath>
                                              <Transform
                                                <XPath>/</XPath>
                                                <Transform
                                                  <XPath>/</XPath>
                                                  <Transform
                                                    <XPath>/</XPath>
                                                    <Transform
                                                      <XPath>/</XPath>
                                                      <Transform
                                                        <XPath>/</XPath>
                                                        <Transform
                                                          <XPath>/</XPath>
                                                          <Transform
                                                            <XPath>/</XPath>
                                                            <Transform
                                                              <XPath>/</XPath>
                                                              <Transform
                                                                <XPath>/</XPath>
                                                                <Transform
                                                                  <XPath>/</XPath>
                                                                  <Transform
                                                                    <XPath>/</XPath>
                                                                    <Transform
                                                                      <XPath>/</XPath>
                                                                      <Transform
                                                                        <XPath>/</XPath>
                                                                        <Transform
                                                                          <XPath>/</XPath>
                                                                          <Transform
                                                                            <XPath>/</XPath>
                                                                            <Transform
                                                                              <XPath>/</XPath>
                                                                              <Transform
                                                                                <XPath>/</XPath>
                                                                                <Transform
                                                                                  <XPath>/</XPath>
                                                                                  <Transform
                                                                                    <XPath>/</XPath>
                                                                                    <Transform
                                                                                      <XPath>/</XPath>
                                                                                      <Transform
                                                                                        <XPath>/</XPath>
                                                                                        <Transform
              </Transform>
            </Transform>
          </Transform>
        </Transform>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>0xTTQyG2qhf wXnx4ozYFpS/D38=</DigestValue>
    </SignedInfo>
    <SignatureValue>Si gnat ureVal ue>Qt GM6SUDl zEi 26p09etY2PbXf Kr 72TdM81TK662HvI UuNYeBahi ntI wHPhFrJ i sA17nE
5BqAxcai BY41+nU3Azuj nNznrXpFH9Svr q0kBERj FVNGZVKbn4FKI mUF8f xYgi ZI VJ 5M
vaPowgl aW NhyHG2OkSueNAUI J QSI zuf 9I =</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="{SAMLCAssertionExpirationDate}" Recipient="https://[server]/restapi/[api version]/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>
```

Token Request

When submitting the token request, the grant type should be: urn:ietf:params:oauth:grant-type:saml2-bearer and the SAML bear assertion must use base64url encoding as defined at the link:

<http://tools.ietf.org/html/rfc4648#page-7>

Example Request

```
POST https://[server]/restapi/v2/oauth2/token
```

```
Accept: application/json
Content-Type: application/x-www-form-urlencoded
grant_type=urn:ietf:params:oauth:grant-type:saml2-bearer&assertion=[base64url encoded SAML Assertion XML]
```

If successful, DocuSign returns an access token that can be stored for future use.

Example Response

```
{
  "access_token": "Access_Token",
  "scope": "api",
  "expires_in": "3600",
  "token_type": "bearer"
}
```

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/{api Version}/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.

You have granted access to the following applications to access your DocuSign account. Picking "Revoke" will prevent the associated application from accessing your account until you grant access to the application in the future.

Connected Application	Date Created
Test App 2	10/29/2012 12:36:02 PM PT

Done

- 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 user_id 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}/rest/api/{api Version}/accounts/{accountId}/envelopes/
{envelopeId}

Authorization: bearer <access token for authenticating user>
X-DocuSign-Act-As-User: bob.smith@mail.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}/rest/api/{api Version}/oauth2/token

Authorization: bearer {access_token for authenticating user}
Accept: application/json
Content-Type: application/x-www-form-urlencoded
grant_type=password&client_id={Integrator Key}&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.

Certificate-Based Authentication

Certificate-based authentication is an optional way to provide additional security checks for REST API calls.

When this option is enabled, it requires that REST API calls are signed and validates the call against the certificate specified by the header containing the certificate thumbprint. The REST API looks up the PublicKeyXML information stored in the DocuSign system based on the thumbprint header and Account ID and uses the information to validate the signature against the request body. If the validation information matches, the REST API confirms that the message body was not tampered with during transmission. If the information is not sent or does not validate, then a '401 Unauthorized' message is returned. DocuSign signs responses with DocuSign's certificate.

The client's PublicKey information is stored in DocuSign's signing certificates table. The information must be provided to DocuSign by the client.

To enable this option and add the client PublicKey information to the DocuSign's signing certificates table, you must coordinate with your Account Manager and, in some circumstance, DocuSign Professional Services.

Certificate-Based Authentication Requests and Responses

After certificate-based authentication is enabled and you make requests to DocuSign, all your REST API calls must be signed with your certificate information in the "X-DocuSign-Signature" and the certificate thumbprint in the "X-DocuSign-Thumbprint" in the request header.

The REST API looks up the PublicKeyXML information based on the thumbprint header and your Account ID and validates the signature against the request body.

If the validation information matches, the REST API confirms that the message body was not tampered with during transmission and DocuSign sends a response with DocuSign's certificate information in the "X-DocuSign-Signature" in the response header.

Certificate-Based Authentication Examples

The following example is a request to send an envelope with certificate-based authentication.

```
POST http://localhost/restapi/v2/accounts/{accountId}/envelopes
X-DocuSign-Authorization:
<DocuSignCredentails><User name>docu.user@docsign.com</User name><Password>docuuser
</Password><IntegratorKey>DOCUSER-ad1d9282-4f95-4a22-af6d-4287d3b21507</IntegratorKey>
</DocuSignCredentails>

X-DocuSign-Signature:
KEYUSk1P612XWg1PE6TyE4udKRq2ZRe67zEhMWrGWWrOhcPkJ8VDrOGXETkQWlMXeq7w1A25b1Wldc+BEKaX4w6fTT9Nc0+UpOFy+EwVNekFKyB/tfGkzu9GYpbhQtjDwH77FfCM1bva/j2+fsKK7zidP+psdoHryGu4LfP4ON3F/CxOLQOGNi jY2*xAK1KQES11WRg6WkT7zi2dUB1Ejch7SJGCFYXG7uVdrEdj3h3kU05RbA10+1MSYv2B3Sh2As+fBzfDlsTHhWxKd/MkpJOhJ2qjbVj28pWPaiJwPA5Ook+Ox4key2WHsmOsTzqeD2716dNji juAToWbYpVPuqq==

X-DocuSign-Thumbprint: 3E571CB172041F02B1F1A4894092FDE79B43731B

Accept: application/json
Content-Type: multipart/form-data; boundary=637627d1-581c-4d9e-8430-d3a86f571355
--637627d1-581c-4d9e-8430-d3a86f571355
Content-Type: application/json
Content-Disposition: form-data

{
```

```

"documents": [
  {
    "documentId": "1",
    "name": "test2.pdf"
  }
],
"status": "sent",
"emailSubject": "Blah Get Envelope_20130925094933",
"recipients": {
  "signers": [
    {
      "tabs": {
        "dateSignedTabs": [
          {
            "value": "default",
            "documentId": "1",
            "recipientId": "1",
            "pageNumber": "1",
            "xPosition": "100",
            "yPosition": "100",
            "tabId": "1"
          }
        ],
        "name": "Mr. Don L. Bits",
        "email": "dbits@mail.com",
        "recipientId": "1",
        "customFields": [
          "CustomField1",
          "CustomField2"
        ],
        "routingOrder": "1"
      }
    }
  ]
}
}

-- 637627d1-581c-4d9e-8430-d3a86f571355
Content-Type: application/pdf
Content-Disposition: file; filename="document.pdf"; documentId=1
Content-Transfer-Encoding: base64

<<docbytes>
-- 637627d1-581c-4d9e-8430-d3a86f571355—

```

The following example shows the response to the previous request.

```

201 Created

Pragma: no-cache

X-DocuSign-Signature:
UY+byH+ynuemEPj3cbgogf4vws2Y6cGz6ekfP1V8ajjFAy4R1rsQwpZkHEXAr/+nCP0Ht+CkOR/IIQJxWYanP8NydB3Enc8L5oM0kFDy0nvL1VlojPupgbWHaXVVKW4pX3+qU+fDQAjIi7+3ps+gxuTbUwHvgntFVF/NWE3/NMu5z1Wj4Amk/Rf5LQ4r8I0tmeKB+6B1XypgrKwkWA29r84UzBqURPgcXstR7D1Dwmcb50ILIdSA/eH0V32jGRA5tYC1r0gicoZqk4rhBeQw3zJk7hf07iOUx2sjpgKGzNB7pVN/19XFUBJCY3JPxsCI0BR07ulLaipQNhcJs17mA==

Content-Length: 173
Cache-Control: no-cache
Content-Type: application/json; charset=utf-8

```

```
Date: Wed, 25 Sep 2013 16:49:56 GMT
Expires: -1

{
  "envelopeId": "8b07f822-a2d4-413f-a7ea-b1d262e2379b",
  "uri": "/envelopes/8b07f822-a2d4-413f-a7ea-b1d262e2379b",
  "statusDate": "2013-09-25T16:49:55.200Z",
}
"status": "sent"
```

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 /logInInfornation api to send a document for signing.

Note: After using logInInformation to get the baseUrl, the base Url and apiPassword can be stored by the client. The logInInformation 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/{api Version}/accounts/{accountId}/envelopes
X-DocuSign-Authentication:
<DocuSignCredentia ls><User name>{name}</User name><User name>{passwor d}</User name><Passwor d>{password}</Passwor d><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
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 - EnvelopeDefinition",
"documents": [ {
    "name": "test1.pdf",
    "documentId": "1"
}],
"recipients": {
    "signers": [ {
        "email": "mroseve@homeland.com",
        "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-fee50f361e2",
    "status": "sent",
    "statusDateTime": "2012-03-06T17:22:17.203000Z",
    "uri": "/envelopes/c254d988-f501-42d3-a858-fee50f361e2"
}

```

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_info or login api to send a document for signing.

Note: After using login_info or login to get the baseUrl, the base Url and apiPassword can be stored by the client. The login_info or login 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/[api Version]/accounts/[accountId]/envelopes

X-DocuSign-Authorization:<User name>{name}</User name><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey><DocuSignCredentials>
Accept: application/json
Content-Type: application/json
Content-Disposition: form-data

{
    "status": "sent",
    "emailSubject": "Test Email Body (Template)",
    "emailSubject": "Test Email Subject (Template)",
    "templateId": "270D920E-C65A-410D-9640-75D2FBEADAC2",
    "templateRoles": [
        {
            "email": "mike.roseyn@docsign.com",
            "name": "Mike Rosey",
            "roleName": "Signer 1"
        },
        {
            "email": "mike.roseyn@docsign.com",
            "name": "Lara Roselyn",
            "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-f0eb50f361e2",
    "status": "sent",
    "statusDateTime": "2012-03-06T17:22:17.203000Z",
    "uri": "/envelopes/c254d988-f501-42d3-a858-f0eb50f361e2"
}
```

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/ {account id}/envelopes/ {envelope id}/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/{api Version}/accounts/{account id}/envelopes/
{envelope id}/views/sender

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><Password>{password}</Password><IntegratorKey>{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&=6cafaea-cc6b-496d-812d-421b7f854de&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/ {account id}/envelopes/ {envelope id}/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 and recipient information (email, user name).

Example Request

```
POST https://{server}/restapi/{api Version}/accounts/{account id}/envelopes/
{envelope id}/views/recipient
```

```
X-DocuSign-Authorization:
<DocuSignCredentails><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentails>
Accept: application/json
Content-Type: application/json
{
    "authenticationMethod": "email",
    "email": "mike.rosey@doctussoft.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
Cache-Control: private
Content-Type: application/json; charset=utf-8
Date: Fri, 30 Mar 2012 14:52:58 GMT
{"url": "http://localhost/Member/StartSession.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 base URL.

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 envelopelds), 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}/rest/api/{api Version}/accounts/{accountId}/envelopes
?fromDate=5-2-2012+12:00:00+AM&fromStatus=changed&status=completed

X-DocuSign-Authorization:
<DocuSignCredentails><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentails>
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\/customfields",
      "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": "2012-05-23T16:21:09.283000Z"
    },
    {
      "certificateUri": "\/envelopes\/23306db8-261a-43c5-b078-ac020113af09\/documents\/certificate",
      "customFieldsUri": "\/envelopes\/23306db8-261a-43c5-b078-ac020113af09\/customfields",
      "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": "2012-05-23T16:20:44.783000Z"
    }
  ],
  "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/[api Version]/accounts/[accountId]/envelopes/[envelopeId]/documents/combined

X-DocuSign-Authorization:<DocuSignCredentials><User name>[name]</User name><Password>[password]</Password><Password>[integratorKey]</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/accounts"}, {"name": "billing_plans", "value": "https://demo.docusign.net/restapi/v2/billing_plans"}, {"name": "login_information", "value": "https://demo.docusign.net/restapi/v2/login_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-Authorization:
<DocuSignCredentials><User name>account@domain.com</User name><Password>account_password</Password><IntegratorKey>your_integrator_key</IntegratorKey></DocuSignCredentials>"
```

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

```
{"loginAccounts": [{"account_id": "account_id#", "base_url": "https://demo.docusign.net/restapi/v2/accounts", "account_email": "account@domain.com", "is_default": true, "name": "User Name", "site_description": "", "user_id": "user_id#", "user_name": "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-Authorization:
<DocuSignCredentials><User name>account@domain.com</User name><Password>account_password</Password><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_id#</accountId><baseUrl>https://demo.docusign.net/restapi/v2/accounts/<accountId>/baseUrl</baseUrl><email>account@domain.com</email><i><isDefault>true</isDefault></i><name>User Name</name></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:

```
{  
    "envelopeBody": "Test Email Body (Template)",  
    "envelopeSubject": "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-Authorization:<DocuSignCredentials><User name><account@domain.com><User name><Password><account password><Password><Integrator Key><your integrator key><Integrator Key><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-Authorization:<DocuSignCredentials><User name><account@domain.com><User name><Password><account password><Password><Integrator Key><your integrator key><Integrator Key><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/[api Version]/accounts/[accountId]/envelopes HTTP/1.1
X-DocuSign-Authorization: [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

{

```
"envelopeId": "sent",  
"emailSubject": "Test Email Subject - EnvelopeDefinition",  
"asynchronous": true,  
    "enforceSignerVisibility": true,  
"documents": [ {  
    "name": "test1.pdf",  
    "documentId": 1  
}],  
"recipients": {  
    "signers": [  
        {  
            "email": "mroseyc@homeland.com",  
            "name": "Mike Rosey",  
            "envelopeRecipientVisibility": true,  
            "tabs": {  
                "tabs omitted 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
Delete Account	DELETE	/accounts/{accountId}	DocuSign service. 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	PUT	/accounts/{accountId}/billing_plan	Updates the billing plan for an
Plan Purchase Additional Envelopes	PUT	/accounts/{accountId}/billing_plan/purchased_envelopes	account. Updates purchased envelope information.

Name	HTTP Method	URI	Description
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.
Delete Captive Signature	DELETE	/accounts/{accountId}/captive_recipients/signature	This deletes the signature for one or more captive recipient records; it is primarily used for testing.
Set Up a Connect Configuration	POST	/accounts/{accountId}/connect	Sets up and configures a DocuSign Custom Connect definition.
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.

Name	HTTP Method	URI	Description
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	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.
Clear One Connect Log Entry	DELETE	/accounts/{accountId}/connect/logs/{logId}	Clears (removes) one entry from the Connect Log.
Get Consumer Disclosure	GET	/accounts/{accountId}/consumer_disclosure	Returns the consumer 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.

Name	HTTP Method	URI	Description
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
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.
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.
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.

Name	HTTP Method	URI	Description
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.
Add Documents to a Draft Envelope	PUT	/accounts/{accountId}/envelopes/{envelopeId}/documents	Adds one or more documents to draft envelope.
Remove Documents from a Draft Envelope	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/documents	Removes 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 Draft Envelope	PUT	/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}	Adds one document to draft 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.

Name	HTTP Method	URI	Description
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.
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.
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.
Remove Recipients from an Envelope	DELETE	/accounts/{accountId}/envelopes/{envelopeId}/recipients	Removes one or more recipients from an envelope.

Name	HTTP Method	URI	Description
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.
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 recipient 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 envelope.
Get Envelope Status for more than one envelope	PUT	/accounts/{accountId}/envelopes/status	DocuSign UI. 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.
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.
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.

Name	HTTP Method	URI	Description
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 List of Templates	GET	/accounts/{accountId}/templates	Retrieves the list of templates for this
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.
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.
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.

Name	HTTP Method	URI	Description
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 Share for a Group	DELETE	/accounts/{accountId}/templates/{templateId}/groups	Removes template sharing from a group.
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 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.

Name	HTTP Method	URI	Description
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.
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.

Name	HTTP Method	URI	Description
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.
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.
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 endpoint 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/{api Version}/login_information
X-DocuSign-Authorization:
<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></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

{
  "api Password": "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"
    }]
  }]
}
  
```

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.

Name	Reqd?	Type	Description
newPassword	Yes	String	The user's new password.

Example Request Body

```
PUT https://[server]/rest/api/[api Version]/login_information/password

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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.

Name	Reqd?	Type	Description
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	No	String	The Distributor Code that identifies the billing plan groups and plans for the new account.
distributorPassword	No	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		<p>A complex type that sets the feature sets for the account. It contains the following information (all string content):</p> <ul style="list-style-type: none"> • 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 DocuSign Plan ID for the account
referralInformation	No		<p>A complex type that contains the following information (all string content): EnableSupport, IncludedSeats, ReferrerName, ReferralCode, AdvertisementID, PublisherID, ShopperID, PromoCode, GroupMemberID, IdType, and Industry.</p>
socialAccountInformation.	No		<p>A complex type that contains the following information:</p> <ul style="list-style-type: none"> • 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
allowInPerson	Boolean	SysAdmin	When true, the account allows In Person Signing.
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.
allowSignerReassign	Boolean	Admin	When true, the account allows signers to reassign an envelope.
attachCompletedEnvelope	Boolean	Admin	When true, envelope documents are included as a PDF file attachment for signing completed emails.
autoNavRule	String	Admin	The auto-navigation rule for the account. Enumeration values are: Off, RequiredFields, RequiredAndBlankFields, AllFields, PageThenRequiredFields, PageThenRequiredAndBlankFields, PageThenAllFields.
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.
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 & AllowPayment Processing is set.	When true, Payment Processing is enabled for the account.
enablePowerForm	Boolean	SysAdmin	When true, PowerForm access is enabled for the account.
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.
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.
envelopeIntegrationAllowed	String	SysAdmin	Shows the envelope integration rule for the account. Enumeration values are: NotAllowed, Full,
envelopeIntegrationEnabled	Boolean	Admin & EnvelopeIntegrationAllowed is set	IntegrationSendOnly. When true, envelope integration is enabled for the account.
envelopeStampingDefaultValue	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.
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.
pkiSignDownloadedPDFDocs	String	Admin	The policy for adding a digital certificate to downloaded, printed and emailed documents. Enumeration values are:

			NoSign, NoSignAllowUserOverride, YesSign
sessionTimeout	Integer	Admin	The amount of idle activity time, in minutes, before a user is automatically logged out of the system. The minimum setting is 20 minutes and the maximum setting is 120 minutes.
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.
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 & account is selected in AccountSigningSettings	When true, signer can use the 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.
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.
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]/rest/api/[api Version]/accounts/
```

X-DocuSign-Authorization:
<DocuSignCredentails><User name>{name}</User name><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentails>

```
{
    "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"
    },
    "disclaimerPage": "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": {
        "group1": "String content",
        "group2": "String content"
    },
    "lastName": "String content",
    "middleName": "String content",
    "password": "String content",
    "sendActivationEmail": "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"
    }
},
"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"  
}
```

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/[api Version]/accounts/{accountId}  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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}/rest/api/{api Version}/accounts/{accountId}
```

X-DocuSign-Authorization:

```
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the account information for the requested account.

The following example shows the response json body.

Example Response Body

```
{  
    "accountName": "String content",  
    "billingPeriodEndDate": "String content",  
    "billingPeriodOpenBalowed": "String content",  
    "billingPeriodOpenSent": "String content",  
    "billingPeriodStartDate": "String content",  
    "connectPermission": "String content",  
    "currencyCode": "String content",  
    "currentPlanId": "String content",  
    "districtOrCode": "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.

URL:

/accounts/{accountId}/billing_plans

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/rest/api/{api Version}/accounts/{accountId}/billing_plan  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the billing plan information, including the currency code, for the plan. The parameters are the same as those shown in the [Get Billing Plan Details reference](#).

The following example shows the response json body.

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",  
        "seatDiscounts": [  
            {  
                "beginSeatCount": "string",  
                "endSeatCount": "string",  
                "discountPercent": "string"  
            }  
        ],  
        "planFeatureSets": [  
            {  
                "featureSetId": "string",  
                "isActive": "string",  
                "name": "string",  
                "fixedFee": "string",  
                "envelopeFee": "string",  
                "seatFee": "string",  
                "isEnabled": "string",  
                "currencyFeatureSetPrices": [  
                    {  
                        "id": "string",  
                        "label": "string",  
                        "order": 0,  
                        "amount": "string",  
                        "currency": "string",  
                        "isDefault": "string"  
                    }  
                ]  
            }  
        ]  
    }  
}
```

```
        "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"
        }
    ]
}
]
```

Update Account Billing Plan

This updates the billing plan for the specified account.

URL:

/accounts/{accountId}/billing_plan

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Read?	Type	Description
billingAddress	No	String	A complex type that contains the following information for the new account: 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
creditCardInformation	No	String	state/province/region. A complex type that has information about the credit card used to pay for this account. It included the elements: cardNumber, cardType, expirationMonth, expirationYear and nameOnCard
enableSupport	No	Boolean	If true, the plan has support enabled.
includeSeats	No	String	The number of seats included in the plan
planInformation currencyCode	Yes		This is the ISO currency code for the account.

Name	Reqd?	Type	Description
planInformation planFeatureSets	See Desc.		<p>A complex type that sets the feature sets for the account. It contains the following information (all string content):</p> <ul style="list-style-type: none"> 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.

Example Request Body

```
PUT https://{server}/rest/api/{api Version}/accounts/{accountId}/billing_plan

X-DocuSign-Authorization:<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>

{
    "billingAddress": {
        "address1": "String content",
        "address2": "String content",
        "city": "String content",
        "country": "String content",
        "fax": "String content",
        "phone": "String content",
        "postal Code": "String content",
        "state": "String content"
    },
    "creditCardInformation": {
        "cardNumber": "String content",
        "cardType": "String content"
    }
}
```

```

        "cardType": "String content",
        "expirationMonth": "String content",
        "expirationYear": "String content",
        "nameOnCard": "String content"
    },
    "enableSupport": "String content",
    "includedSeats": "String content",
    "planInformation": {
        "featureSets": [ {
            "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"
    }
}
}

```

Response

The response returns a success or failure.

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.

Name	Reqd?	Type	Description
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}/rest/api/{api Version}/accounts/{accountId}/billing_plan/purchased_envelopes

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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]/rest/api/[api Version]/accounts/[accountId]/brands  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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 “canSelfBrandSend” 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/[api Version]/accounts/{accountId}/brands
X-DocuSign-Authorization:<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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"
    }
  ],
  "recipientBrandDefault": "String content",
  "senderBrandDefault": "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/[api Version]/accounts/{accountId}/brands
X-DocuSign-Authorization:<DocuSignCredentia ls><User name>[name]</User name><Pass word>[pass word]</Pass word><Integrato r Key>[integrator_key]</IntegratorKey><DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json

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

Response

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

Delete Captive Recipient Signature

This deletes the signature for one or more captive recipient records; it is primarily used for testing. 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.
user_name	Yes	String	The user name associated with the captive recipient.
clientUserId	Yes	String	The sender created value associated with the captive recipient.

Example Request Body

```
DELETE https://[server]/restapi/[api Version]/accounts/{accountId}/
captive_recipients/signature

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Passw ord>{password}</Passw ord></nt egrat o
r Key></IntegratorKey></IntegratorKey></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json

{
  "captiveRecipients": [
    {
      "email": "String content",
      "user_name": "String content",
      "clientUserId": "String content"
    },
    {
      "email": "String content",
      "user_name": "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

```
{  
  "capti veRecipients": [  
    {  
      "email": "sample string",  
      "userName": "sample string",  
      "envelopeId": "sample string",  
      "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, Signed, Completed, Declined, Voided
includeDocuments	No	Boolean	When true, Connect will send the PDF document along with the update XML.
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 this Connect configuration.
recipientEvents	No	String	A comma separated list of "Recipient" related events that are tracked through Connect. The possible event values are: Sent, Delivery Failed, Delivered, Signed/Completed, Declined, AuthenticationFailure

Name	Reqd?	Type	Description
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.
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/[api Version]/accounts/[accountId]/connect
X-DocuSign-Authorization:<DocuSignCredentails><UserName>[name]</UserName><User name>[name]</User name><Password>[password]</Password><Password>[password]</Password><IntegratorKey>[integrator_key]</IntegratorKey><IntegratorKey>[integrator_key]</IntegratorKey><DocuSignCredentails>
Accept: application/json
Content-Type: application/json

{
  "allUsers": "boolean",
  "allowEnvelopePublish": "boolean",
  "enableLog": "boolean",
  "envelopeEvents": "string",
  "includeDocuments": "boolean",
  "includeEnvelopeVoidReason": "boolean",
}

```

```

    "includeSenderAccountAsCustomField": "boolean",
    "includeTimeZoneInformation": "boolean",
    "name": "string content",
    "recipientEvents": "string content",
    "requiresAcknowledgment": "boolean",
    "signMessageWithX509Certificate": "boolean",
    "soapNamespace": "string content",
    "useToPublishTo": "boolean",
    "userIds": "string content"
}

```

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",
  "requiresAcknowledgment": "boolean",
  "signMessageWithX509Certificate": "boolean",
  "soapNamespace": "string content",
  "useToPublishTo": "string content",
  "userIds": "string content"
}

```

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]/rest/api/[api Version]/accounts/{accountId}/connect
X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
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",
      "allowEnvelopePublishing": "String content",
      "connectId": "String content",
      "enableLog": "String content",
      "envelopeEvents": "String content",
      "includeCustomSoapHeaders": "String content",
      "includeSenderAccountAsCustomField": "String content",
      "includeTimeZoneInformation": "String content",
      "name": "String content",
      "recipientEvents": "String content",
      "requireAcknowledgment": "String content",
      "signMessageWithX509Certificate": "String content",
      "soapNamespace": "String content",
      "urlToPublishTo": "String content",
      "useSoapInterface": "String content",
      "users": "String content"
    },
    {
      "allUsers": "String content",
      "allowEnvelopePublishing": "String content",
      "enableLog": "String content",
      "envelopeEvents": "String content",
      "includeCustomSoapHeader": "String content",
      "includeDocuments": "String content",
      "includeSenderAccountAsCustomField": "String content",
      "includeTimeZoneInformation": "String content",
      "name": "String content",
      "recipientEvents": "String content",
      "requireAcknowledgment": "String content",
      "signMessageWithX509Certificate": "String content"
    }
  ]
}
```

```

    "soapNamespace": "String content",
    "urlToPublishTo": "String content",
    "useSoapInterface": "String content",
    "userIds": "String content"
},
"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, Signed, Completed, Declined, Voided

Name	Reqd?	Type	Description
includeDocuments	No	Boolean	When true, Connect will send the PDF document along with the update XML.
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 this Connect configuration.
recipientEvents	No	String	A comma separated list of “Recipient” related events that are tracked through Connect. The possible event values are: Sent, Delivery Failed, Delivered, Signed/Completed, Declined, AuthenticationFailure
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.

Name	Reqd?	Type	Description
userIds	No*	String	A comma separated list of user Ids. 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]/rest/api/[api Version]/accounts/{accountId}/connect
X-DocuSign-Authentication:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integratorKey}</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",
    "requiresAcknowledgment": "boolean",
    "signMessageWithX509Certificate": "boolean",
    "soapNamespace": "string content",
    "urlToPublishTo": "string content",
    "useSoapInterface": "boolean",
    "userIds": "string content"
}
```

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",
    "requiresAcknowledgment": "boolean",
}
```

```

    "signMessageWithX509Certificate": "boolean",
    "soapNamespace": "string content",
    "urlToPublishTo": "string content",
    "useSoapInterface": "boolean",
    "userIds": "string content"
}

```

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}/rest/api/{api Version}/accounts/{accountId}/connect/{connectId}

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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",
  "includeCertificateSoapHeader": "String content",
  "includeDocuments": "String content",
  "includeSenderAccountAsCustomField": "String content",
  "isArchivingEnabled": "String content"
}
```

```
"includeTemplateInformation": "String content",
"name": "String content",
"recipientEvents": "String content",
"requireAcknowledgment": "String content",
"signMessageWithX509Certificate": "String content",
"soapNamespace": "String content",
"urlToPublishTo": "String content",
"useSoapInterface": "String content",
}
```

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}/{api Version}/accounts/{accountId}/connect/
{connectId}

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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]/rest/api/[api Version]/accounts/{accountId}/connect/envelopes/[envelopeId]/retry_queue
X-DocuSign-Authorization:<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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
envelopelds	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]/rest/api/[api Version]/accounts/[accountId]/connect/envelopes/retry_queue

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integrator_key]</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "envelopelds": [ "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/{api Version}/accounts/{accountId}/connect/failures

X-DocuSign-Authorization:
<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Password>{password}</Password>
<Integrator Key>{integrator_key}</Integrator Key></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.
connectDebugLog		A complex element containing information about the Connect configuration, error details, date/time, description and payload.
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",
      "connectDebugLog": [
        {
          "connectConfig": "String content",
          "errorDetails": {
            "errorCode": "String content",
            "message": "String content"
          },
          "eventDateTime": "String content",
          "eventDescription": "String content",
          "payload": "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",
      "connectDebugLog": [
        {
          "connectConfig": "String content",
          "errorDetails": {
            "errorCode": "String content",
            "message": "String content"
          }
        },
        "eventDateTime": "String content"
      ]
    }
  ]
}
```

```

        "eventDescription": "String content",
        "payload": "String content"
    ],
    "connectId": "String content",
    "created": "String content",
    "email": "String content",
    "envelopeId": "String content",
    "error": "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}/rest/api/{api Version}/accounts/{accountId}/connect/failures/
{failureId}
X-DocuSign-Authorization:
<DocuSignCredentials><username>{name}</username><password>{password}</password><integratorKey>{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.
connectDebugLog		A complex element containing information about the Connect configuration, error details, date/time, description and payload.
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",
      "connectDebugLog": [
        {
          "connectConfig": "String content",
          "failure": {
            "account": "String content"
          }
        }
      ],
      "envelopeId": "String content"
    }
  ]
}
```

```

        "errorDetails": {
            "errorCode": "String content",
            "message": "String content"
        },
        "eventDateTime": "String content",
        "eventDescription": "String content",
        "payload": "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/{api Version}/accounts/{accountId}/connect/failures/
{failureId}

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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 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}/rest/api/{api Version}/accounts/{accountId}/connect/logs

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Int egrator Key>{integrator_key}</Int egrator Key></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the Connect log information.

Name	Type	Description
accountId	String	The account ID associated with the envelope.
created	String	The date and time the entry was created.
envelopeId	String	The envelope ID of the envelope status that failed to post.
logId	String	The Connect log ID for the entry.
logUri	String	The uri for the log item.
subject	String	The envelope subject.

The following example shows the response json body.

Example Response Body

```
{
  "logs": [
    {
      "accountId": "String content",
      "created": "String content",
      "logId": "String content",
      "logUri": "String content",
      "subject": "String content"
    },
    {
      "accountId": "String content",
      "created": "String content",
      "envelopeId": "String content",
      "logId": "String content",
      "logUri": "String content",
      "subject": "String content"
    }
  ],
  "totalRecords": "String content",
  "type": "String content"
}
```

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/{api Version}/accounts/{accountId}/connect/logs

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
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 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}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://[server]/rest/api/[api Version]/accounts/{accountId}/connect/logs/{logId}

X-DocuSign-Authorization:
<DocuSignCredentia><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia>
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.
created	String	The date and time the entry was created.
envelopeId	String	The envelope ID of the envelope status that failed to post.
logId	String	The Connect log ID for the entry.
logUri	String	The uri for the log item.
subject	String	The envelope subject.

The following example shows the response json body.

Example Response Body

```
{  
  "logs": [ {  
    "accountId": "String content",  
    "created": "String content",  
    "logId": "String content",  
    "logUri": "String content",  
    "subject": "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/{api Version}/accounts/{accountId}/connect/logs/  
{logId}  
  
X-DocuSign-Authorization:  
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Int egrator  
Key>{integrator_key}</Int egrator Key></DocuSignCredentia ls>  
Accept: application/json  
Content-Type: application/json
```

Response

The response returns a success or failure message.

Get Consumer Disclosure

This returns the consumer disclosure, with html formatting, associated with the account. You can use an optional query string to set the language for the consumer 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 consumer 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 automatically detect the browser language being used by the viewer and display the consumer disclosure in that language.

Example Request Body

```
GET https://[server]/rest/api/[api Version]/accounts/{accountId}/consumer_disclosure
X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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 consumer disclosure text. The consumer disclosure text includes the html formatting.

The following example shows the response json body. For brevity, the example only shows a portion of the consumer disclosure text.

Example Response Body

```
{  
  "accountEsignd": "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}/rest/api/{api Version}/accounts/{accountId}/custom_fields  
X-DocuSign-Authorization:  
<DocuSignCredentia>{username}</username><password>{password}</password><integratorKey>{integrator_key}</integratorKey><DocuSignCredentia>
```

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 entered with quotations, separated by commas. 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.

This resource can also be used with offline signing client applications, see the [Offline Signing section](#) for more information about offline signing.

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.

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.
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
authoritativeCopy	No	String	completed. 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.

Name	Reqd?	Type	Description
enableWetSign	No	String	If true, the signer is allowed to print the document and sign it on paper.
enforceSignerVisibility	No	String	If 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.
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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> ◦ reminderEnabled – When true a reminder message is sent to the recipient. ◦ reminderDelay – An integer that sets the number of days after the recipient receives the envelope that reminder emails are sent to the recipient. ◦ reminderFrequency – An integer that sets the interval, in days, between reminder emails. • expirations – A complex element that specifies the expiration settings for the envelope. It consists of: <ul style="list-style-type: none"> ◦ expireEnabled – When true the envelope expires (is no longer available for signing) in the set number of days.

Name	Reqd?	Type	Description
			<ul style="list-style-type: none"> o expireAfter – An integer that sets the number of days the envelope is active. o 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.
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	<p>Sets envelope status. There are two possible values: sent and created.</p> <p>If status = sent, the envelope is sent to the recipients.</p> <p>If status = created, the envelope is saved as a draft and can be modified and sent later.</p>
transactionId	No	String	<p>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.</p> <p>It is recommended that a transactionId is used for offline signing to ensure that an envelope is not sent multiple times.</p> <p>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.</p>
customFields	No		<p>A complex element that can be used to store information about the envelope, help search for envelopes and track information.</p> <p>See the section on getting Custom Fields for more information about and descriptions of the custom fields.</p>

Name	Reqd?	Type	Description
documents	No	Document	<p>Complex element contains the details on the documents in the envelope.</p> <p>See the Document parameter section for more information.</p>
recipients	No	Recipient	<p>Specifies the envelope recipients.</p> <p>See the Recipient parameter section for more information.</p>
eventNotification	No	eventNotification	<p>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.</p> <p>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.</p> <p>eventNotification consists of:</p> <ul style="list-style-type: none"> • 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.

Name	Reqd?	Type	Description
			<ul style="list-style-type: none"> • 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. • 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. • 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: <ul style="list-style-type: none"> o envelopeEventStatus – The envelope status, this can be Sent, Delivered, Signed, Completed, Declined, or Voided. • 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: <ul style="list-style-type: none"> o recipientEventStatus – The recipient status, this can be AuthenticationFailed, AutoResponded, Completed, Declined, Delivered or Sent.

Documents Parameters:

Name	Reqd?	Type	Description
name	Yes	String	The name of the document. This can be a maximum of 100 characters. 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.

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.
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.
documentBase64	No	String	The document byte stream. This allows putting a base64 version of document bytes into an envelope.

Name	Reqd?	Type	Description
matchboxes	No		<p>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:</p> <ul style="list-style-type: none"> 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 DocuSignTitle or eSignTitle = DocuSign Title

Adobe name contains DocuSignFullName or eSignFullName = DocuSign FullName

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](#)

[In Person Signers](#)

[Carbon Copies](#)

[Intermediaries](#)

[Certified Deliveries](#)

[Signers](#)

[Editors](#)

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/[api Version]/accounts/[accountId]/envelopes
```

X-DocuSign-Authorization:

```
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><integratorKey>[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",  
"allowReassign": "String content",  
"allowRecipientRecursion": "String content",  
"asynchronous": "String content",  
"authoritativeCopy": "String content",  
"autoNavigation": "String content",  
"brandId": "String content",  
"emailBlurb": "String content",  
"emailSubject": "String content",  
"enableSignature": "String content",  
"enforceSignerVisibility": "String content",
```

```
"envelopeOpelDetails": "String content",
"messageLock": "string content",
"notifications": {
    "useAccountDefaults": "String content",
    "reminders": [ {
        "reminderEnabled": "String content",
        "reminderDelay": "String content",
        "reminderFrequency": "String content"
    }],
    "expirations": [ {
        "expirationEnabled": "String content",
        "expirationAfter": "String content",
        "expirationWarn": "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"
    }]
},
"documents": [ {
    "documentId": "1",
    "name": "String content",
    "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",
        "logInEnabled": "String content",
        "requireAcknowledgment": "String content",
        "useSoapInterface": "String content",
        "soapNamespace": "http://DocuSignConnectListener",
        "includeCertificateWithSoap": "true",
        "signMessageWithX509Cert": "String content",
        "includeDocuments": "String content",
        "includeTimeZone": "http://DocuSignConnectListener",
        "includeSenderAccountAsCustomField": "true",
        "envelopeEvents": [ {
            "envelopeEventStatusCode": "sent"
        },
        {
            "envelopeEventStatusCode": "completed",
            "includeDocuments": "true"
        }],
        "recipientEvents": [ {
            "recipientEventStatusCode": "Sent"
        },
        {
            "recipientEventStatusCode": "completed",
            "includeDocuments": "true"
        }]
    }
}
-- 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"
}
```

Offline Signing

Offline signing provides a way for customers to use DocuSign to obtain recipient signatures, even when the customer is not connected to the Internet. To use Offline Signing, the customer using a client application to create an envelope on an Internet-capable device, such as a smart phone or tablet, that is not connected to the Internet and has a recipient fill out and sign (or initial) the envelope documents. When the device later connects to the Internet, the client application uploads the envelope information to DocuSign where it is processed.

When processing, the events in envelope audit trail are marked with the description “Offline” as appropriate. The Certificate of Completion will include a separate Offline section that will have the offline signing view event and complete event date/time information, the recipient photo (if included), and any custom information (such as the offlineAttributes) for the signer.

Offline Signing Settings and Restrictions

The following settings and restrictions apply to offline signing use cases.

- Offline signing is not allowed when sending from templates.
- All offline signers must be in routing order 1 and must have at least one required signature or initials tag. Other recipients can be included in the envelope.
- Offline signing supports the following tab types:
Sign Here, Initial Here, Date Signed, Full Name, Company, Title, Text and Checkbox
- When sending an offline signing envelope, the envelope “status” must be set to “sent” (“status”=“created” is not allowed and will cause an error).
- When sending an offline signing envelope, the following [Signers Recipient type](#) settings are required: clientUserId, deliveryMethod (must be set to “offline”) and signedDateTime. An image of the signer’s signature/initials must be included for each signature or initials tab. The image is included as part of the multipart section. Optionally, you can include a deliveredDateTime parameter, but if this parameter is not included DocuSign will set it to the signedDateTime value.

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.

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.

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.

- 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 meta data template will be overlaid into the envelope in the order of their Sequence value.
- Document objects are optional structure 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).

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.

URL:

/accounts/{accountId}/envelopes

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Read?	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.
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.
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.

Name	Reqd?	Type	Description
enableWetSign	No	String	If true, the signer is allowed to print the document and sign it on paper.
enforceSignerVisibility	No	String	If 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.
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.

Name	Reqd?	Type	Description
notification	No		<p>An optional complex element that specifies the notification options for the envelope. It consists of:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> ◦ 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: <ul style="list-style-type: none"> ◦ expireEnabled – When true the envelope expires (is no longer available for signing) in the set number of days. ◦ 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.
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.

Name	Reqd?	Type	Description
status	Yes	String	<p>Sets envelope status. There are two possible values: sent and created.</p> <p>If status = sent, the envelope is sent to the recipients.</p> <p>If status = created, the envelope is saved as a draft and can be modified and sent later.</p>
customFields	No		<p>A complex element that can be used to record information about the envelope, help search for envelopes and track information.</p> <p>See the section on getting Custom Fields for more information about and descriptions of the custom fields.</p>
transactionId	No	String	<p>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.</p> <p>It is recommended that a transactionId is used for offline signing to ensure that an envelope is not sent multiple times.</p> <p>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.</p>
templateId	Yes	String	Unique identifier of the template.
templateRoles	Yes	Recipients	<p>Specifies the template recipients. Each roleName in the template must have a recipient assigned to it. This is made up elements:</p> <ul style="list-style-type: none"> • email – The recipient's email address. • Name – The recipient's name. • roleName – The template roleName associated with the recipient. • clientUserId – Optional, this sets if the signer is embedded or remote. If the clientUserId is not null then the recipient is embedded. • 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.

Name	Reqd?	Type	Description
			<ul style="list-style-type: none"> 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. 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	<p>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.</p> <p>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.</p> <p>eventNotification consists of:</p> <ul style="list-style-type: none"> 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

Name	Reqd?	Type	Description
			<p>send the DocuSign signedby certificate as part of the outgoing SOAP xml. This appears in the XML as wsse:BinarySecurityToken.</p> <ul style="list-style-type: none"> • 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. • 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. • 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: <ul style="list-style-type: none"> ◦ envelopeEventStatus – The envelope status, this can be Sent, Delivered, Signed, Completed, Declined, or Voided. • 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: <ul style="list-style-type: none"> ◦ recipientEventStatus – The recipient status, this can be AuthenticationFailed, AutoResponded, Completed, Declined, Delivered or Sent.
compositeTemplates	No		<p>Each CompositeTemplate adds a new document and template overlays into the envelope. There can be any number of CompositeTemplate structures. Each CompositeTemplate consists of:</p> <p>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.</p> <p>InlineTemplates - 0 or more inline templates and their position in the overlay. If supplied</p>

Name	Reqd?	Type	Description
			<p>they are overlaid into the envelope in the order of their Sequence value.</p> <p>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.</p> <p>Document - 0 or 1 document. The document is used to specify a different document than the ones in the overlay templates. The PDF fields from this document will be transformed to DocuSign Secure Fields if the TransformPdfFields option is set to true.</p> <p>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.</p> <p>If PDFMetaDataTemplate is supplied this is the document the template is extracted from. This information is unchanged from the current schema.</p>

Example Request Body

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

X-DocuSign-Authorization:<User name>{name}</User name><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
Content-Disposition: form-data

{
  "accessibility": "string content",
  "allowMarkup": "String content",
  "allowReassign": "String content",
  "allowRecipientSelection": "String content",
  "asynchronousCopy": "String content",
  "autoNavigation": "String content",
  "brandId": "String content",
  "emailBlurb": "String content",
  "emailSubject": "String content",
  "enableSignature": "String content",
  "enforceSignerVisibility": "String content",
  "envelopeStamping": "String content",
  "messageLock": "string content",
  "recipientLock": "string content",
  "signingLocation": "String content",
  "status": "string content"
}
```

```

    "status": "String content",
    "transactionId": "String content",
    "customFields": [
        "listCustomFields": [
            {
                "name": "String content",
                "required": "String content",
                "shape": "String content",
                "listItems": [
                    "String content"
                ]
            }
        ],
        "textCustomFields": [
            {
                "name": "String content",
                "required": "String content",
                "show": "String content",
                "value": "String content"
            }
        ]
    ],
    "template": "String content",
    "templateRoles": [
        {
            "email": "String content",
            "name": "String content",
            "roleName": "String content",
            "clientUserId": "String content",
            "defaultRecipient": "String content",
            "emailNotification": {
                "emailSubject": "String content",
                "emailBody": "String content",
                "supportLanguage": "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",
                "includeTimeZone": "http://DocuSignConnect Listener",
                "includeSenderAccountAsCustomField": "true",
                "envelopeEvents": [
                    {
                        "envelopeEventStatusCode": "sent"
                    },
                    {
                        "envelopeEventStatusCode": "completed",
                        "includeDocuments": "true"
                    }
                ]
            }
        }
    ]
}

```

```
        ],
        "recipientEvents": [
            {
                "recipientEventStatusCode": "Sent"
            },
            {
                "recipientEventStatusCode": "completed",
            }
        ],
        "includeDocuments": "true"
    ]
},
"compositeTemplates": [
    {
        "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": 1
            }
        ]
    }
]
```

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",
    "statusDate": "String content",
    "uri": "String content"
}

```

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.

Getting envelope status using transactionIds is useful for offline signing situations where it can be used determine if an envelope was created or not, for the cases where a network connection was lost, before the envelope status could be returned.

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

Status Qualifier (from_to_status)	Effective Status Qualifier	Valid Current Statuses
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}, envelopId={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.

Name	Reqd?	Schema Type	Description
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.
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.

Name	Reqd?	Schema Type	Description
custom_field	No	String	<p>This specifies the envelope custom field name and value searched for in the envelope information.</p> <p>The value portion of the query can use partial strings by adding '%' (percent sign) around the custom field query value.</p> <p>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.</p> <p>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."</p>
transactionIds	No	String	<p>If included in the query string, this is a comma separated list of envelope transactionIds.</p> <p>If included in the request_body, this is a list of envelope transactionIds.</p> <p>Note: transactionIds are only valid in the DocuSign system for 7 days.</p>

Example Request Body

```
GET https://[server]/rest/api/[api Version]/accounts/{accountId}/envelopes?
transaction_ids=request_body

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Int egrator Key>{integrator_key}</Int egrator Key></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json

{
  "transaction_ids": [
    "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",  
    "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]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><egratorKey>{integrator_key}</egratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the envelope status information for the requested envelope.

Example Response Body

```
{
  "allowReassig": "String content",
  "certificateUri": "String content",
  "createdDateTime": "String content",
  "customFieldsUri": "String content",
  "documentsCombinedUri": "String content",
  "documentsUri": "String content",
  "emailBlurb": "String content",
  "emailSubject": "String content",
  "emailSignature": "String content",
  "envelopeId": "String content",
  "envelopeUri": "String content",
  "notificationUri": "String content",
  "recipientsUri": "String content",
  "status": "String content",
  "statusChangedDateTime": "String content"
}
```

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}/rest/api/{api Version}/accounts/{accountId}/envelopes/
{envelopeId}
X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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}/rest/api/{api Version}/accounts/{accountId}/envelopes/
{envelopeId}
X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{ }
```

```

    "status": "voiced",
    "voicedReason": "voiced for incorrect recipient"
}

```

Response

The response returns success or failure of the operation.

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]/rest/api/[api Version]/accounts/{account Id}/envelopes/
{envelopeId}/audit_events

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>[name]</User name><Pass word>[password]</Pass word><Int egrator
r Key>[integrator_key]</Integrator Key></DocuSignCredentia ls>
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 that sent the envelope.
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": "String content",
          "value": "String content"
        }
      ]
    }
}
```

Add Envelope Custom Fields to an Envelope

This allows you to add envelope custom fields to draft and in-process envelopes.

Note: Custom Fields added with this 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
required	Yes	String (Boolean)	When true, information must be entered into the custom field to send the envelope.

Name	Reqd?	Type	Description
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.
listItems	Yes	String	The list of values that can be selected by senders. The list values are entered with quotations, separated by commas. Example: ["one", "two", "three", "four"]
textCustomFields			
name	Yes	String	The name of the Envelope Custom Field
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.

Example Request Body

```
POST https://[server]/restapi/[api Version]/accounts/[accountId]/envelopes/
{envelopeId}/customFields

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>[name]</User name><Pass word>[password]</Pass word><Int egrat o r Key>[integrator_key]</IntegratorKey></DocuSignCredentia ls>

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"
  }]
}
```

```
        "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]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/custom_fields
X-DocuSign-Authentication:<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><Password>[password]</Password><IntegratorKey>[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 entered with quotations, separated by commas. 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" ]
    },
    {
      "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.

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	Reqd?	Type	Description
name	Yes	String	The name of the Envelope Custom Field
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 new value for the custom field in this envelope.
listItems	Yes	String	The list of values that can be selected by senders. The list values are entered with quotations, separated by commas. Example: ["one", "two", "three", "four"]
textCustomFields			
fieldId	Yes	String	The DocuSign generated ID number for the Envelope Custom Field.
name	Yes	String	The name of the Envelope Custom Field
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 new value for the custom field in this envelope.

Example Request Body

```

PUT https://{server}/rest/api/{api Version}/accounts/{account Id}/envelopes/
{envelope Id}/customFields

X-DocuSign-Authorization:<DocuSignCredentials>{name}</User name><password>{password}</password><integratorKey>{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",
      "listItems": [ "String", "String" ]
    }
  ]
}

```

```
        "show": "String",
        "value": "String",
        "listItems": ["String", "String"]
    }],
    "textCustomFields": [
        {
            "fieldId": "String",
            "name": "String",
            "isRequired": "String",
            "value": "String"
        },
        {
            "fieldId": "String",
            "name": "String",
            "isRequired": "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",
            "isRequired": "String",
            "show": "String",
            "value": "String",
            "listItems": ["String", "String"]
        },
        {
            "fieldId": "String",
            "name": "String",
            "isRequired": "String",
            "show": "String",
            "value": "String",
            "listItems": ["String", "String"]
        }
    ],
    "textCustomFields": [
        {
            "fieldId": "String",
            "name": "String",
            "isRequired": "String",
            "show": "String",
            "value": "String"
        },
        {
            "fieldId": "String",
            "name": "String",
            "isRequired": "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	Read?	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/[api Version]/accounts/{accountId}/envelopes/
{envelopeId}/custom_fields

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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"  
    },  
    "listCustomFields": [ {  
        "fieldId": "String"  
    },  
    {  
        "fieldId": "String"  
    }]  
}
```

Add Documents to a Draft Envelope

This can be used to add one or more documents to an existing draft envelope or change the order of documents in a draft envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

See [Document parameters](#).

Example Request Body

```
PUT https://[server]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/  
documents  
  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integrator_key]</IntegratorKey></DocuSignCredentials>  
  
Content-Type: multipart/form-data; boundary=AAA  
  
--AAA  
Content-Type: application/json
```

```
Content-Disposition: form-data
{
  "documents": [ {
    "documentId": "String content",
    "name": "String content",
    "order": "String content"
  },
    {
    "documentId": "String content",
    "name": "String content",
    "order": "String content"
  }
]
--AAA
Content-Type: application/pdf
Content-Disposition: file; filename="String content"; documentId=10
--AAA
Content-Type: application/pdf
Content-Disposition: file; filename=" String content"; documentId=40
--AAA-
```

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.

Remove Documents from a Draft Envelope

This removes one or more document 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}/{api Version}/accounts/{accountId}/envelopes/{envelopeId}/
documents
```

```
X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/documents

X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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).

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",
      "content": "String content"
    },
    {
      "documentId": "String content",
      "name": "String content",
      "type": "String content",
      "uri": "String content",
      "order": "String content",
      "pages": "String content",
      "content": "String content"
    },
  ],
}
```

Add a Document to a Draft Envelope

This adds one document to an existing draft envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

See [Document parameters](#).

Example Request Body

```
PUT https://[server]/rest/api/[api Version]/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}
```

```
X-DocuSign-Authorization:<DocuSignCredentia ls><User name>[name]</User name><Pass word>[password]</Pass word><Integrator Key>[integrator_key]</Integrator Key></DocuSignCredentia ls>
```

```
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: show_changes={true}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameters are the envelope ID and document ID.

Name	Reqd?	Schema Type	Description
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/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
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.
max_height	No	String	The dpi is recalculated based on this setting. Sets the maximum height for the page image. The dpi is recalculated based on this setting.

Example Request Body

```
GET https://[server]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/pages/{pageId}/page_image
X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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}/rest/api/{api Version}/accounts/{accountId}/envelopes/{envelopeId}/documents/{documentId}/pages/{pageId}/page_image

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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/{api Version}/accounts/{accountId}/envelopes  
/{envelopeId}/documents/{documentId}/pages/{pageId}  
  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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/{api Version}/accounts/{accountId}/envelopes/  
{envelopeId}/documents/{documentId}/fields  
  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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. The following example shows the response json body.

Example Response Body

```
{  
  "documentFields": [  
    {  
      "name": "sample string",  
      "value": "sample string"  
    },  
    {  
      "name": "sample string",  
      "value": "sample string"  
    }  
  ]  
}
```

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	Read2	Type	Description
------	-------	------	-------------

Name	Reqd?	Type	Description
documentFields	Yes	documentField	<p>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:</p> <p>Name – A string that can be a maximum of 50 characters.</p> <p>Value – A string that can be a maximum of 200 characters.</p>

Example Request Body

```
POST https://[server]/restapi/[api Version]/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}/fields

X-DocuSign-Authentication:<{name}></User name><{password}></Password><{integratorKey}></Integrator Key><{DocuSignCreds}>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Response

The response returns the custom document field name-value pairs added to the document and any errors associated with the fields.

The following example shows the response json body.

Example 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"
        }
    ]
}

```

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	<p>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:</p> <p>Name – A string that can be a maximum of 50 characters.</p> <p>Value – A string that can be a maximum of 200 characters.</p>

Example Request Body

```

PUT https://[server]/rest/api/[api Version]/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}/fields

X-DocuSign-Authorization:<DocuSignCredentails><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentails>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample_string_1",
      "value": "sample_string_2",
    }
  ]
}

```

```
},
{
  "name": "sample string 1",
  "value": "sample string 2",
}
]
```

Response

The response returns the modified custom document field name-value pairs for the document and any errors associated with the fields.

The following example shows the response json body.

Example 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"
      }
    }
  ]
}
```

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
------	-------	------	-------------

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.

Example Request Body

```
DELETE https://{server}/restapi/{api Version}/accounts/{accountId}/envelopes/
{envelopeId}/documents/{documentId}/fields

X-DocuSign-Authentication<{name}></User name><{password}></Password><{integrator_key}></Integrator Key><{DocuSignCredent i als}>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

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.

The following example shows the response json body.

Example Response Body

```
{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Get Envelope Certificate

This retrieves a PDF document containing the certificate of completion for the envelope.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/documents/certificate

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID.

Example Request Body

```
GET https://{server}/rest/api/{api Version}/accounts/{account Id}/envelopes/{envelopeId}/documents/certificate

X-DocuSign-Authorization:<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></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}, show_changes={true}, watermark={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID.

Name	Reqd?	Schema Type	Description
certificate	No	Boolean	Optional. This option can remove the envelope signing certificate from the download.
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/{api Version}/accounts/{accountId}/envelopes/{envelopeId}/documents/combined

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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.

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]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/notification  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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 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}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only required parameter is the envelope ID. If the optional query include_extended is set to true, the extended properties are returned.

Example Request Body

```
GET https://[server]/rest/api/[api Version]/accounts/{account ID}/envelopes/
{envelope ID}/recipients

X-DocuSign-Authentication:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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, 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.

The following example shows the response json body.

Example Response Body

```
{
  "agents": [],
  "carbonCopies": [],
  "currentRoutingOrder": "String content",
  "editors": [],
  "inPersonSigners": [],
  "intermediaries": [],
  "recipientCount": "String content",
  "signers": [
    {
      "deliveredDate": "String content",
      "recipientAuthenticationStatus": {
        "authenticationStatusResult": {
          "eventTimestamp": "String content",
          "status": "String content"
        }
      },
      "recipientLookupString": "String content",
      "roleName": "String content",
      "routingOrder": "String content",
      "signedDate": "String content",
      "status": "String content",
      "email": "String content",
      "name": "String content"
    }
  ]
}
```

Add Recipients to an Envelope

This is used to add one or more recipients 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}`

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/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/recipients

X-DocuSign-Authorization:<DocuSignCredentia><User name>{name}</User name><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentia>
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",
        "supportLanguage": "String content"
    },
    "idCheckConfigurationName": "String content",
    "inheritingEmailNotificationConfiguration": "String content",
    "note": "String content",
    "phoneAuthentication": {
        "recipientMayProvideNumber": "String content",
        "recordVoicePrint": "String content",
        "senderProvidedNumbers": ["String content"],
        "validateRecipientProvidedNumber": "String content"
    },
    "recipientAttachments": [ {
        "attachment": "String content"
    }]
}]
```

```

        "attachmentType": "String content",
        "label": "String content"
    ],
    "recipientId": "String content",
    "requireRelLookup": "String content",
    "roleName": "String content",
    "routingOrder": "String content",
    "socialAuthentications": ["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",
    "requireRelLookup": "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",
    "requireRelLookup": "String content",
    "roleName": "String content",
    "status": "Created"
  }
]
}
```

```

        "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 an envelope. The recipients that need 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/{api Version}/accounts/{accountId}/envelopes/
{envelopeId}/recipients

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><egratorKey>{integrator_key}</egratorKey></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 occurred 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": "Delivered",  
        "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	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.

Example Request Body

```
PUT https://[server]/rest/api/[api Version]/accounts/{account Id}/envelopes/
{envelope Id}/recipients

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>[name]</User name><Password>[password]</Password><Integrator Key>[integrator_key]</Integrator Key></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json

{
  "signers": [
    {
      "email": "mke.roseliep@docsign.com",
      "name": "Mke Roseliep",
      "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"
    }
  ]
}
```

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 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]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/initials_image

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><egratorKey>{integrator_key}</egratorKey></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]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/signature
```

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>

Response

The response returns information for the request recipient signature. The following example shows

Example Response Body

```
{  
    "adoptedDateTime": "String content",  
    "createdDateTime": "String content",  
    "initialSignatureId": "String content",  
    "initialSignatureUri": "String content",  
    "signature150ImageId": "String content",  
    "signatureFont": "String content",  
    "signatureId": "String content",  
    "signatureImageUri": "String content",  
    "signatureInitials": "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]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/signatureImage  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integratorKey]</IntegratorKey></DocuSignCredentials>
Accept:[application/json]
Content-Type: application/json

{
    "approveTabs": [ {
        <TabInformation removed>
    }],
    "titleTabs": [ {
        <TabInformation removed>
    }],
    "signHereTabs": [ {
        <TabInformation 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

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://[server]/restapi/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integratorKey]</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>
  }]
}
```

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}/{api Version}/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs

X-DocuSign-Authorization:<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
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	Read?	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]/rest/api/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/recipients/{recipientId}/tabs
```

```

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
    "approveTabs": [
        <TabInformation removed>
    ],
    "titleTabs": [
        <TabInformation removed>
    ],
    "signHereTabs": [
        <TabInformation 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

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```

GET https://[server]/rest/api/[api Version]/accounts/{accountId}/envelopes
/{envelopeId}/templates

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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.

The following example shows the response json body.

Example Response Body

```
{
  "templates": [
    {
      "name": "String content",
      "templateId": "String content",
      "uri": "String content"
    }
  ]
}
```

Post Envelope Correction

This provides a URL to start the correction view of the DocuSign UI.

Important: DocuSign recommends that you do not use iFrames for embedded operations on mobile devices.

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. The URL will have an event passed to it as a query parameter. The parameter will be named "event", see "In-session sending events" below for description of the valid events. The DocuSign Envelope Id will also be returned in the "envelopeId" parameter.
suppressNavigation	No	String	Sets whether the window is displayed with or without dressing.

Example Request Body

```
POST https://[server]/restapi/[api Version]/accounts/{accountId}/envelopes/{envelopeId}/views/correct

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><ntegratorKey>[integratorKey]</ntegratorKey></DocuSignCredentials>
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.

Important: DocuSign recommends that you do not use iFrames for embedded operations on mobile devices.

URL:

/accounts/{accountId}/envelopes/{envelopeId}/views/recipient

Formats:

XML, JSON

HTTP Method:

POST

Parameters:

Name	Reqd?	Type	Description
authenticationMethod	Yes	String	The convention used to authenticate the end-user.

clientUserId	Yes	String	A sender created value that shows the recipient is embedded (captive). This can be a maximum of 100 characters.
email	Yes*	String	Specifies the email of the recipient. * You can use either email and userName or userId to identify the recipient.
returnUrl		String	The URL the recipient is directed to on certain events. 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: <ul style="list-style-type: none">• cancel (recipient cancels signing)• decline (recipient declines signing)• exception (exception occurs)• fax_pending (recipient has fax pending)• id_check_failed (recipient failed an ID check)• session_timeout (session times out)• signing_complete (recipient completes signing)• ttl_complete (the TTL expires)• viewing_complete (recipient completes viewing the envelope)
userId	Yes*	String	Specifies the user ID of the recipient. * You can use with userId or email and userName to identify the recipient. If userId is used and a clientUserId is provided, the userId must match a recipientId (which can be retrieved with a GET recipients call) for the envelope. If userId is used and a clientUserId is not provided, the userId must match the userId of the authenticating user.
userName	Yes*	String	Specifies the username of the recipient * You can use either email and userName or userId to identify the recipient.

Example Request Body

```
GET https://[server]/rest/api/[api Version]/accounts/[accountId]/envelopes/[envelopeId]/view/recipient
X-DocuSign-Authorization:<DocuSignCredentials><User name>[name]</User name><Password>[password]</Password><Integrator Key>[integrator_key]</Integrator Key></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

```
{  
    "authenticationMethod": "String content",  
    "clientUserId": "String content",  
    "email": "String content",  
    "returnUrl": "String content",  
    "userId": "String content"  
}  
"userName": "String content"
```

Response

The response returns the recipient view url.

The following example shows the response json body.

Example Response Body

```
{  
    "url": "String content"  
}
```

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.

Important: DocuSign recommends that you do not use iFrames for embedded operations on mobile devices.

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.

Example Request Body

```
POST https:// {SERVER}/restapi/ {api Version}/accounts/ {accountId}/envelopes/  
{envelopeId}/views/sender
```

```

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
} "returnUrl": "String content"

```

Response

The response returns the sender view url.

The following example shows the response json body.

Example Response Body

```
{
  "url": "String content"
}
```

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]/rest/api/[api Version]/accounts/{accountId}/envelopes/
status?envelope_ids=request_body
```

```

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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",
      "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	<p>The two possible values are “include” or “only.”</p> <ul style="list-style-type: none"> • 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]/rest/api/[api Version]/accounts/[accountId]/folders
X-DocuSign-Authorization:<DocuSignCredentails><User name>[name]</User name><Password>[password]</Password><IntegratorKey>[integrator_key]</IntegratorKey></DocuSignCredentails>
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",
        "fromDate": "String content",
        "toDate": "String content",
        "searchTarget": "String content",
        "searchText": "String content"
      }
    }
  ]
}
```

```

        "folders": [
            {
                "id": "string",
                "name": "string",
                "order": "String content",
                "orderBy": "String content",
                "parentFolderId": "string"
            }
        ],
        "total": 1
    }
}

```

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
<code>start_position</code>	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.
<code>from_date</code>	No	Date/time	Only return items on or after this date. If no value is provided, the default search is the previous 30 days.
<code>to_date</code>	No	Date/time	Only return items up to this date. If no value is provided, the default search is to the current date.
<code>search_text</code>	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.
<code>status</code>	No	Envelope status	The current status of the envelope. If no value is provided, the default search is all/any status.

Name	Reqd?	Type	Description
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/[api Version]/accounts/{accountId}/folders/{folderId}

X-DocuSign-Authorization:<DocuSignCredentails><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentails>
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",
  "folders": [
    {
      "ownerName": "String content",
      "envelopeId": "String content",
      "status": "String content",
      "senderName": "String content",
      "senderEmail": "String content",
      "createDate": "String content",
      "sentDate": "String content",
      "completedDate": "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",
      "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.

URL:

/accounts/{accountId}/folders/{folderId}

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

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/[api Version]/accounts/{accountId}/folders/{folderId}

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>[name]</User name><Pass word>[pass word]</Pass word><Int egrator Key>[integrator_key]</Int egrator Key></DocuSignCredentia ls>

Accept: application/json
Content-Type: application/json

{
  "envelopes": [ "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 permissionProfileId	Yes No	String String	The name for the new user group. 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/[api Version]/accounts/{accountId}/groups
X-DocuSign-Authorization:<DocuSignCredentia><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia>
Accept: application/json
Content-Type: application/json

{
  "groups": [ {
    "groupName": "String content",
    "permissionsProfileId": "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",  
    "permissionsProfile": "String content"  
  }]  
}
```

Get Group Information

This retrieves information about groups associated with the account.

URL:

/accounts/{accountId}/groups

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/{api Version}/accounts/{accountId}/groups  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>  
Accept: application/json  
Content-Type: application/json
```

Response

The response returns the information for the groups associated with the account.

The following example shows the response json body.

Example Response Body

```
{
  "groups": [ {
    "groupId": "String content",
    "groupName": "String content",
    "permissionsProfileId": "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	Read?	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/{api Version}/accounts/{accountId}/groups
X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "groups": [ {
    "groupId": "String content",
    "groupName": "String content",
    "permissionsProfileId": "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",
    "permissionProfileId": "String content"
  }]
}

```

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}/rest/api/{api Version}/accounts/{accountId}/groups/{groupId}/
brands

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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]/rest/api/[api Version]/accounts/{accountId}/groups/{groupId}/brands

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integratorKey]</IntegratorKey></DocuSignCredentials>
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]/rest/api/[api Version]/accounts/{accountId}/groups/{groupId}/brands

X-DocuSign-Authentication:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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	Read2	Type	Description
userId	Yes	String	The user ID number for a user being added to the group.

Example Request Body

```
PUT https://[server]/rest/api/[api Version]/accounts/[accountId]/groups/[groupId]/users

X-DocuSign-Authorization:<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></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 retrieves a list of users in a group.

URL:

/accounts/{accountId}/groups/{groupId}/users

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{api Version}/accounts/{accountId}/groups/{groupId}/users

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><Password>{password}</Password>
IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the information for the users in the group.

The following example shows the response json body.

Example Response Body

```
{
  "users": [
    {
      "uri": "String content",
      "userId": "String content",
      "userName": "String content",
      "userStatus": "String content",
      "userType": "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/[api Version]/accounts/[accountId]/groups/[groupId]/users
X-DocuSign-Authentication:<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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 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}/rest/api/{api Version}/accounts/{account Id}/permission_profiles  
X-DocuSign-Authorization:  
<DocuSignCredentia ls><User name>{name}</User name><Passw ord>{password}</Passw ord><Integrato r Key>{integrator_key}</IntegratorKey></DocuSignCredentia ls>  
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}/rest/api/{api Version}/accounts/{accountId}/recipient_names?  
email={email}

X-DocuSign-Authorization:  
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{pass word}</Pass word><Int egrat o r Key>{integrator_key}</IntegratorKey></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the recipient names associated with the email if used by more than one user.

Example Response Body

```
{
  "multipl eUsers": "f al se",
  "recipients": [
    "Resty B. Tester",
    "Resty Tester",
    "Resty Tester III"
  ],
  "reservedRecipientEmail": "f al se"
}
```

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.
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: <ul style="list-style-type: none"> • 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}/rest/api/{api Version}/accounts/{accountId}/search_folders/
{search_folder}?include_recipients=true

X-DocuSign-Authorization:
<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></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.

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": "2012-07-13T15:57:00.690000Z",
      "envelopeID": "463d30c4-3e29-417b-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@mail.com",
        }
      }
    }
  ]
}
```

```

        "name": "Don Cat",
        "routingOrder": 1,
        "status": "sent"
    }]
},
"senderName": "Bill Cat",
"sentDateTime": "2012-07-13T15:57:27.867000Z",
"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/{api Version}/accounts/{accountId}/settings
X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Int egrator Key>{integrator_key}</Int egrator Key></DocuSignCredentia ls>
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
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}/rest/api/{api Version}/accounts/{accountId}/settings

X-DocuSign-Authorization:<DocuSignCredentia ls><User name>{name}</User name><Pass word>{pass word}</Pass word><Integrator Key>{integrator _key}</Integrator Key></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json

{
  "accountSettings": [ {
    "name": "String content",
    "value": "String content"
  }]
}
```

Response

The response returns a success or failure.

Get List of Templates

Retrieves the list of templates for the specified account. The request can be limited to a specific folder.

URL:

/accounts/{accountId}/templates

Optional query string: folder={string}, includeSubFolders={true or false}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

The only information needed for this is the account ID, but the following optional parameters can be added to the request.

Name	Reqd?	Type	Description
Folder	No	String	The query value can be a folder name or folder ID. The response will only return templates in the specified folder.
includeSubFolders	No	Boolean	When true, the response will include all subfolders.

Example Request Body

```
GET https://[server]/rest/api/[api Version]/accounts/{accountId}/templates
X-DocuSign-Authorization:
<DocuSignCredentia><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integrator_key]</IntegratorKey></DocuSignCredentia>
Accept: application/json
Content-Type: application/json
```

Response

This returns a list of templates for the account with the following information. If the optional folder query is used, only the templates in the specified folder are returned.

Name	Type	Description
description	String	Description of the template.

Name	Type	Description
folderName	String	The name of the folder where the template is located.
folderUri	String	The uri for accessing the folder.
lastModified	String	Last time the template was modified.
name	String	Name of the template.
pageCount	number	Number of document pages in the template.
password	String	Password, if the template is locked. NOTE: This is only returned if the requestor is the creator of the template.
shared	String	If true, the template is shared with the Everyone group in the account. If false, the template is only shared with the Administrator group.
templateId	String	DocuSign Unique identifier for the template.
uri	String	The uri for accessing the template.

The following example shows the response json body.

Example Response Body

```
{
  "envelopeTemplates": [
    {
      "description": "String content",
      "folderName": "String content",
      "folderUri": "String content",
      "lastModified": "String content",
      "name": "String content",
      "pageCount": 2147483647,
      "password": "String content",
      "shared": "String content",
      "templateId": "String content",
      "uri": "String content"
    }
  ]
}
```

Post Template

Saves a template definition using a multipart request.

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.
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 EnvelopeStatus will have a value of 'Processing'. Use SynchEnvelope to determine when the queued envelope has been processed. Additionally, RequestStatus calls will return 'Processing' until completed.
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.

Name	Reqd?	Type	Description
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.
enableWetSign	No	String	If true, the signer is allowed to print the document and sign it on paper.
enforceSignerVisibility	No	String	If 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	An optional complex element consisting of: <ul style="list-style-type: none"> • 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

Name	Reqd?	Type	Description
			<p>wsse:BinarySecurityToken.</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> ◦ envelopeEventStatus – The envelope status, this can be Sent, Delivered, Signed, Completed, Declined, or Voided. ◦ includeDocuments – When set to true, the documents in the envelope are included in the exported XML.
messageLock	No	String	<p>If true, prevents senders from changing the emailBlurb and emailSubject for the envelope.</p> <p>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.</p>
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	<p>Complex element contains the details on the documents in the envelope. See the Document parameter section for more information.</p>

Name	Reqd?	Type	Description
recipients	Yes	Recipient	<p>Specifies the envelope recipients. It uses the same information as a standard envelope recipient, with the following exceptions:</p> <ul style="list-style-type: none"> • The email and name elements are not required. • The roleName element is required. This can be a maximum of 100 characters. <p>See the Recipient parameter sections for more information.</p>
envelopeTemplateDefinition	Yes	Definition	<p>A complex element with the following information:</p> <p>templateId: Unique identifier of the template.</p> <p>name: Name of the template. This can be a maximum of 100 characters.</p> <p>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.</p> <p>password: Password, if the template is locked.</p> <p>description: Description of the template. This can be a maximum of 500 characters.</p> <p>pageCount: Number of document pages in the template.</p> <p>folderName: The name of the folder the template is located in.</p> <p>folderId: The ID for the folder.</p> <p>owner: The userName, email, userId, userType, and userStatus for the template owner.</p>
notification	No		<p>An optional complex element that specifies the notification options for the envelope. It consists of:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> ◦ reminderEnabled – When true

Name	Reqd?	Type	Description
			<p>a reminder message is sent to the recipient.</p> <ul style="list-style-type: none"> o reminderDelay – An integer that sets the number of days after the recipient receives the envelope that reminder emails are sent to the recipient. o reminderFrequency – An integer that sets the interval, in days, between reminder emails. • expirations – A complex element that specifies the expiration settings for the envelope. It consists of: <ul style="list-style-type: none"> o expireEnabled – When true the envelope expires (is no longer available for signing) in the set number of days. o expireAfter – An integer that sets the number of days the envelope is active. o 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/[api Version]/accounts/[accountId]/templates

X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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",
    "status": "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",
"enableSignature": "String content",
"enforceSignatureVisibility": "String content",
"signingLocation": "String content",
"envelopeStamping": "String content",
"eventNotification": {
    "url": "String content",
    "logInEnabled": "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,  
        "customFields": null,  
        "emailNotification": null,  
        "idCheckConfigurationName": null,  
        "idCheckInformationInput": null,  
        "inheritedEmailNotificationConfiguration": false,  
        "note": "",  
        "phoneAuthentication": null,  
        "recipientAttachments": null,  
        "recipientId": "1",  
        "requireRelLookup": false,  
        "roleName": "Signer 1",  
        "routingOrder": 1,  
        "socialAuthentications": null,  
        "tempalteAccessCodeRequired": 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, null,  
            "envelopeTabs": 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,
    "inheritedEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": null,
    "recipientAttachments": null,
    "recipientCaption": null,
    "recipientId": "2",
    "requireRelLookup": false,
    "fileName": "Signature_2",
    "fileNameOrder": 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,
        "declineTabs": null,
        "emailTabs": null,
        "envelopeTabs": null,
        "fullNameTabs": null,
        "initialHereTabs": null,
        "listTabs": null,
        "noteTabs": null,
        "numberTabs": null,
        "radioGroupTabs": null,
        "signHereTabs": [
            {
                "anchorString": null,
                "anchorXOffset": null,
                "anchorYOffset": null,
                "anchorUrgencyNotPresent": 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,
    "zipTabs": null
}
},
}

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

<PDF Bytes Removed>

```

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/{api Version}/accounts/{accountId}/templates/{templateId}
```

```
X-DocuSign-Authorization:
<DocuSignCredentails><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentails>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the template definition of the requested 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 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",
  "allowResign": "String content",
  "allowRecipientResign": "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"
  }]
}
```

```

}],
"envelopeBody": "String content",
"envelopeSubject": "String content",
"enableSignature": "String content",
"enforceSignerVisibility": "String content",
"notification": {
    "expirations": {
        "expireAfter": "String content",
        "expireWarn": "String content"
    },
    "reminders": {
        "reminderDelay": "String content",
        "reminderEnabled": "String content",
        "reminderFrequency": "String content"
    },
    "useAccountDefaults": "String content"
},
"recipients": {
},
"signingLocation": "String content",
}
}

```

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}/rest/api/{api Version}/accounts/{accountId}/templates/
{templateId}/documents/{documentId}/fields
X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><egratorKey>{integrator_key}</egratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

```

Response

The response returns the custom document field name-value pairs for the requested document ID.

The following example shows the response json body.

Example Response Body

```
{  
  "documentFields": [  
    {  
      "name": "sample string",  
      "value": "sample string"  
    },  
    {  
      "name": "sample string",  
      "value": "sample string"  
    }  
  ]  
}
```

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	<p>The array of name-value custom data strings to be added to a document. Custom document field information is returned with template information and added to envelopes created with the template, but otherwise is not used by DocuSign. The array contains one or more elements:</p> <p>Name – A string that can be a maximum of 50 characters.</p> <p>Value – A string that can be a maximum of 200 characters.</p>

Example Request Body

```
POST https://[server]/restapi/[api Version]/accounts/{accountId}/templates/
{templateId}/documents/{documentId}/fields

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integratorKey]</IntegratorKey>
```

Content-Type: application/json

```
{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Response

The response returns the custom document field name-value pairs added to the document and any errors associated with the fields.

The following example shows the response json body.

Example 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"
      }
    }
  ]
}
```

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	<p>The array of name-value custom data strings to be modified in a document. Custom document field information is returned with template information and added to envelopes created with the template, but otherwise is not used by DocuSign. The array contains the elements:</p> <p>Name – A string that can be a maximum of 50 characters.</p> <p>Value – A string that can be a maximum of 200 characters.</p>

Example Request Body

```
PUT https://[server]/rest/api/[api Version]/accounts/{accountId}/templates/{templateId}/documents/{documentId}/fields

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Passw ord>{password}</Passw ord><Integrato r Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json

{
  "documentFields": [
    {
      "name": "sample string 1",
      "value": "sample string 2",
    },
    {
      "name": "sample string 1",
      "value": "sample string 2",
    }
  ]
}
```

Response

The response returns the modified custom document field name-value pairs for the document and any errors associated with the fields.

The following example shows the response json body.

Example 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"  
      }  
    }  
  ]  
}
```

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
------	-------	------	-------------

Name	Reqd?	Type	Description
documentFields	Yes	documentField	<p>The array of name-value custom data strings to be deleted from a document. Custom document field information is returned with template information and added to envelopes created with the template, but otherwise is not used by DocuSign. The array contains the elements:</p> <p>Name – A string that can be a maximum of 50 characters.</p> <p>Value – A string that can be a maximum of 200 characters.</p>

Example Request Body

```
DELETE https://{server}/restapi/{api Version}/accounts/{accountId}/templates/
{templateId}/documents/{documentId}/fields

X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integratorKey}</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",
    }
  ]
}
```

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.

The following example shows the response json body.

Example Response Body

```
{
  "documentFields": [
    {
      "name": "sample string",
      "value": "sample string"
    },
    {
      "name": "sample string",
      "value": "sample string"
    }
  ]
}
```

{}

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]/rest/api/[api Version]/accounts/{accountId}/templates/{templateId}/groups
```

X-DocuSign-Authentication: <User name>{name}</User name><Password>{password}</Password><IntegratorKey>{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"
    },
    "groupI d": "String content",
    "groupNa me": "String content",
    "groupType": "String content",
    "permissi onProfil e": "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}/{api Version}/accounts/{accountId}/templates/
{templateId}/groups

X-DocuSign-Authorization:
<DocuSignCredenti al s><User name>{name}</User name><Passw ord>{passw ord}</Passw ord><Integr ato r Key>{integrator _key}</Integrator Key></DocuSignCredenti al s>
Accept: application/json
Content-Type: application/json

{
    "groups": [
        {
            "groupI d": "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"
    }
  }]
}
```

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/{api Version}/accounts/{accountId}/unsupported_file_types
X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{pass word}</Pass word><Int egrat o r Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
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": [
    {
      "fileType": "String content"
    }
  ]
}
```

```

        "fileExtension": "String content",
        "imeType": "String content"
    },
    {
        "fileExtension": "String content",
        "imeType": "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.
groupList	No	String	A list of groupIds for groups to add the user to.
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.

Name	Reqd?	Type	Description
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 they 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
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.

Setting Name	Value	Authorization Required	Description
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 through the DocuSign Console.
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.

Setting Name	Value	Authorization Required	Description
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.
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.
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/[api Version]/accounts/[accountId]/users
X-DocuSign-Authorization:<DocuSignCredentails><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentails>
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",
        "groupList": "String content"
    },
    "lastName": "String content",
    "middleName": "String content",
    "password": "String content",
    "sendActivationEmailIdLogin": "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": [ {
        "apiPassword": "String content",
        "email": "String content",
        "errorDetails": {
            "errorCode": "String content",
            "message": "String content"
        },
        "uri": "String content",
        "userId": "String content",
        "userName": "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}

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.

Example Request Body

```
GET https://{server}/rest/api/{api Version}/accounts/{account Id}/users
```

```
X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><Password>{password}</Password><IntegratorKey>{integrator key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the list of users for the specified account. If the additional_info query was added to the URL and set to true, the full user information is returned for each user.

The following example shows the response json body.

Example Response Body

```
{
  "users": [ {
    "uri": "String content",
    "userId": "String content",
    "userName": "String content",
    "userStatus": "String content",
    "userType": "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/[api Version]/accounts/{account ed}/users

X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://{server}/restapi/{api Version}/accounts/{accountId}/users/{userId}

X-DocuSign-Authorization:<DocuSignCredentails><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentails>
Accept: application/json
Content-Type: application/json
```

Response

The response returns the user information.

The following example shows the response json body.

Example Response Body

```
{
  "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": [
    {
      "errorDetails": {
        "errorCode": "String content",
        "message": "String content"
      }
    },
    "groupId": "String content",
    "isPrimary": true
  ],
  "language": "en-US",
  "lastName": "String content",
  "middleName": "String content",
  "name": "String content",
  "notes": "String content",
  "profile": "String content",
  "status": "String content",
  "timeZone": "String content",
  "username": "String content"
}
```

```

        "groupName": "String content",
        "groupType": "String content",
        "permissionProfileId": "String content"
    }],
    "homeAddress": {
        "address1": "String content",
        "address2": "String content",
        "city": "String content",
        "fax": "String content",
        "phone": "String content",
        "postalCode": "String content",
        "stateOrProvince": "String content"
    },
    "lastName": "String content",
    "middleName": "String content",
    "sendActivationEmail": "String content",
    "suffixName": "String content",
    "title": "String content",
    "userName": "String content",
    "userSettings": [ {
        "errorDetails": {
            "errorCode": "String content",
            "message": "String content"
        },
        "name": "String content",
        "value": "String content"
    }],
    "workAddress": {
        "address1": "String content",
        "address2": "String content",
        "city": "String content",
        "country": "String content",
        "fax": "String content",
        "phone": "String content",
        "postalCode": "String content",
        "stateOrProvince": "String content"
    }
}
}

```

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/{api Version}/accounts/{accountId}/users/{userId}/custom_settings
X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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]/rest/api/[api Version]/accounts/{accountId}/users/{userId}/
custom_settings

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>[name]</User name><Pass word>[password]</Pass word><Integrator
Key>[integrator_key]</Integrator Key></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json

X-DocuSign-User-Settings-Key: group_name
```

```
{  
  "customSettings": [ {  
    "name": "String content",  
    "value": "String content",  
  },  
  {  
    "name": "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	Read?	Type	Description
customSettings	Yes	String	The name/value pair information that should be deleted.

Example Request Body

```
DELETE https://[server]/restapi/[api Version]/accounts/{accountId}/users/{userId}/
customSettings

X-DocuSign-Authentication:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><Password>{password}</Password><IntegratorKey>{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/{api Version}/accounts/{accountId}/users/{userId}/profile
X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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:

	<ul style="list-style-type: none"> • 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.
userDetails	<p>A complex element with the user name information consisting of:</p> <ul style="list-style-type: none"> • 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",
  "displayOrganization": "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	<p>These properties cannot be modified in the PUT. Shows the authentication methods used by the user. A complex element consisting of:</p> <ul style="list-style-type: none"> • authenticationMethodType (PhoneAuth, STAN, ISCheck, OFAC, AccessCode, AgeVerify, or SSOAuth) • lastTimestamp – the date 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
displayUsageHistory	No	String	history. 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	<p>A complex element consisting of:</p> <ul style="list-style-type: none"> • 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	<p>A complex element with the user name information consisting of:</p> <ul style="list-style-type: none"> • <code>firstName</code> – The user's first name. This can be a maximum of 50 characters. • <code>lastName</code> – The user's last name. This can be a maximum of 50 characters. • <code>middleName</code> – The user's middle name. This can be a maximum of 50 characters. • <code>suffixName</code> – The suffix for the user's name. This can be a maximum of 50 characters. • <code>title</code> – The user's title. This can be a maximum of 10 characters. • <code>username</code> – The user's full name. This can be a maximum of 100 characters.

Example Request Body

```

PUT https://{server}/rest/api/{api Version}/accounts/{account Id}/users/{user Id}/profile

X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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": "StringContent",  
    "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]/rest/api/[api Version]/accounts/{accountId}/users/{userId}/profile/image  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><integratorKey>{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]/rest/api/[api Version]/accounts/{accountId}/users/{userId}/profile/image

X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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/[api Version]/accounts/{accountId}/users/{userId}/profile/image  
X-DocuSign-Authorization:<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentials>
```

Response

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

Get User Account Settings

This returns the account settings list 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/[api Version]/accounts/{accountId}/users/{userId}/settings
```

X-DocuSign-Authorization:

```
[key]<[IntegrationKey]>[UserAgent]<[Base64EncodedPassword]>password<[ntegrato]
```

Response

The response returns the account setting name/value information 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_content",  
         "value": "String content"}  
    ]  
}
```

Modify User Account Settings

This updates the account settings list for the specified user.

URL:

/accounts/{accountId}/users/{userId}/settings

Formats:

XML, JSON

HTTP Method:

PUT

Parameters:

Name	Read?	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.

Example Request Body

```
PUT https://{server}/rest/api/{api Version}/accounts/{accountId}/users/{userId}/
settings

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integratorKey}</IntegratorKey>
Content-Type: application/json

{
    "userSettings": [
        {
            "name": "String content",
            "value": "String content"
        }
    ]
}
```

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}/rest/api/{api Version}/accounts/{accountId}/users/{userId}/
signatures
```

X-DocuSign-Authorization:

```
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
```

Response

The response returns a list of signatures for the user.

Example Response Body

```
{
  "userSignatures": [
    {
      "adoptedDateTime": "String content",
      "createdDateTime": "String content",
      "initialsImageUri": "String content",
      "signatureFont": "String content",
      "signatureImageUri": "String content",
      "signatureInitials": "String content",
      "signatureName": "String content",
      "signatureType": "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"
    }
  ]
}
```

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	<u>The font type for the signature, if the signature is not drawn.</u> <u>The supported font types are:</u> "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/[api Version]/accounts/{accountId}/users/{userId}/signatures
X-DocuSign-Authorization:<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredentials>
Accept: application/json
Content-Type: application/json

{
  "userSignatures": [
    {
      "signatureFont": "7_DocuSign",
      "signatureInitials": "RT20121127083847",
      "signatureName": "Ron Test 20121127083847"
    }
  ]
}
```

```

POST https://{server}/restapi/{api Version}/accounts/{accountId}/users/{userId}/
signatures

X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{integratorKey}</IntegratorKey><Accept>[image/png]</Accept><DocuSignCredentials>
Content-Type: multipart/form-data; boundary=AAA

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

{
  "userSignatures": [
    "signatureFont": "DocuSign",
    "signatureInitials": "RT20121127083900",
    "signatureName": "Ron Test 20121127083900"
  ]
}

- AAA
Content-Type: image/gif
Content-Disposition: file;filename="Ron Test 20121127083900"
Content-Transfer-Encoding: base64

<base64 image bytes omitted>
- AAA
Content-Type: image/gif
Content-Disposition: file;filename="R T20121127083900"
Content-Transfer-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",
      "initials150Image": "String content",
      "initials150SignatureUri": "String content",
      "signature150Image": "String content",
      "signatureFont": "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]/rest/api/[api Version]/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}  
X-DocuSign-Authorization:<DocuSignCredential><User name>{name}</User name><Password>{password}</Password><ntegratorKey>{integrator_key}</IntegratorKey></DocuSignCredential>
```

Response

The response returns information for the request user signature. The following example shows the response json body.

Example Response Body

```
{  
    "adoptedDateTime": "String content",  
    "initialImageUri": "String content",  
    "signatureFont": "String content",  
    "signatureImageUri": "String content",  
    "signatureInitials": "String content",  
    "signatureName": "String content",  
    "signatureType": "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	<p>The font type for the signature, if the signature is not drawn.</p> <p>The supported font types are:</p> <p>"7_DocuSign", "1_DocuSign", "6_DocuSign", "8_DocuSign", "3_DocuSign", "Mistral", "4_DocuSign", "2_DocuSign", "5_DocuSign", "Rage Italic"</p>

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 signatureId 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]/rest/api/[api Version]/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}

X-DocuSign-Authorization:<User name>{name}</User name><Password>{password}</Password><IntegratorKey>{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

```
{
  "adoptedDateTime": "String content",
  "createdDateTime": "String content",
  "initials150ImageId": "String content",
  "initials150ImageUri": "String content",
  "signature150ImageId": "String content",
  "signatureFont": "String content",
  "signatureId": "String content",
  "signatureImageUri": "String content",
  "signatureInitials": "String content",
  "signatureName": "String content",
  "signatureType": "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}}/rest/api/{{api Version}}/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}  
X-DocuSign-Authorization:  
<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></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}/rest/api/{api Version}/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
```

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]/rest/api/[api Version]/accounts/[account Id]/users/[userId]/signatures/[signatureIdOrName]/initials_image  
X-DocuSign-Authentication:  
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[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",  
    "initialsImageId": "String content",  
    "initialsImageUri": "String content",  
    "signatureImageId": "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/[api Version]/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image
X-DocuSign-Authorization:<DocuSignCredential><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredential>
```

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",
  "initialsImageId": "String content",
  "initialsImageUri": "String content",
  "signatureImageId": "String content",
  "signatureFont": "String content",
  "signatureId": "String content",
  "signatureImageUrl": "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}/rest/api/{api Version}/accounts/{account Id}/users/{user Id}/signatures/{signatureIdOrName}/signature_image

X-DocuSign-Authorization:
<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></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]/rest/api/[api Version]/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/initials_image

X-DocuSign-Authorization:<DocuSignCredentia ls><User name>{name}</User name><Pass word>{password}</Pass word><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>

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",  
    "initialsImageId": "String content",  
    "initialsImageUri": "String content",  
    "signatureImageId": "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/{api Version}/accounts/{accountId}/users/{userId}/signatures/{signatureIdOrName}/signature_image
X-DocuSign-Authorization:<DocuSignCredentials><User name>{name}</User name><Password>{password}</Password><Integrator Key>{integrator_key}</Integrator Key></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}/rest/api/{api Version}/accounts/{accountId}/users/{userId}/social

X-DocuSign-Authorization:
<DocuSignCredentia ls><User name>{name}</User name><Pass word>{pass word}</Pass word><Integrator Key>{integrator_key}</Integrator Key></DocuSignCredentia ls>
Accept: application/json
Content-Type: application/json

{
    "provider": "String content",
    "email": "String content",
    "user Name": "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/{api Version}/accounts/{accountId}/users/{userId}/social

X-DocuSign-Authorization:<User name>{name}</User name><Password>{password}</Password><IntegratorKey>{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/{{api Version}}/accounts/{{accountId}}/users/{{userId}}/social
X-DocuSign-Authorization:
<DocuSignCredentials><UserName>{{name}}</UserName><Password>{{password}}</Password><IntegratorKey>{{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 can be included in the request.

Example Request Body

```
POST https://[server]/restapi/[api Version]/accounts/[accountId]/views/console  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integrator_key]</IntegratorKey></DocuSignCredentials>  
Accept: application/json  
Content-Type: application/json
```

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:

No parameters are required.

Example Request Body

```
GET https://[server]/restapi/[api Version]/accounts/provisioning  
X-DocuSign-Authorization:  
<DocuSignCredentials><UserName>[name]</UserName><Password>[password]</Password><IntegratorKey>[integrator_key]</IntegratorKey></DocuSignCredentials>
```

Response

The response returns the account provisioning information.

The following example shows the response json body.

Example Response Body

```
{  
    "defaultSpanningOrder": "String content",  
    "distributorCode": "String content",  
    "distributorPassword": "String content",  
    "passwordRuleText": "String content",  
    "planPromotionText": "String content",  
    "purchaseOrderOrPromotion": "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/[api Version]/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"
    }],
    "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/{planId}

Formats:

XML, JSON

HTTP Method:

GET

Parameters:

No additional parameters are required.

Example Request Body

```
GET https://[server]/restapi/[api Version]/billing_plans/{planId}
```

X-DocuSign-Authorization:

```
<DocuSignCredentials><UserName>{name}</UserName><Password>{password}</Password><IntegratorKey>{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 and currencySymbol.
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		<p>A complex type that sets the feature sets for the account. It contains the following information (all string content):</p> <ul style="list-style-type: none"> • 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

Name	Type	Description
		<p>set is actively enabled as part of the plan.</p> <ul style="list-style-type: none"> • name - The name of the feature set. • seatFee A planned seat cost for 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	<p>A complex type that return any seat discount information.</p> <p>It contains the information: BeginSeatCount, EndSeatCount and SeatDiscountPercent.</p>
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

```
{
  "billablePlan": {
    "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"
    },
    "successorPlans": [
        {
            "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",
                    "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"
        }
    ]
}
```

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/{api Version}/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/{api Version}/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.
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"
}
```

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](#)
[In Person Signers](#)

[Carbon Copies](#)
[Intermediaries](#)

[Certified Deliveries](#)
[Signers](#)

[Editors](#)

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,
  "customFields": {
    "sample string 1",
    "sample string 2"
  },
  "emailNotification": {
    "emailBody": " ",
    "emailSubject": " ",
    "supportedLanguage": " "
  }
}]
```

```
},
"excludedDocuments": [ "2", "4" ],
"idCheckConfiguration": null,
"idCheckInformationInput": {
    "addressInformationInput": {
        "addressInformation": {
            "street1": "sample string 1",
            "street2": "sample string 2",
            "state": "sample string 4",
            "zip": "sample string 5",
            "zipPlus4": "sample string 6"
        },
        "displayLevelCode": "sample string 1",
        "receiveResponse": "sample string 2"
    },
    "dobInformationInput": {
        "dateOfBirth": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveResponse": "sample string 3"
    },
    "ssn4InformationInput": {
        "ssn4": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveResponse": "sample string 3"
    },
    "ssn9InformationInput": {
        "ssn9": "sample string 1",
        "displayLevelCode": "sample string 2"
    }
},
"inheritEmailNotificationConfiguration": false,
"note": "",
"phoneAuthentication": null,
"recipientAttachment": null,
"recipientId": "1",
"requireRelLookup": false,
"roleName": "",
"routeOrder": 1,
"sanAutentication": {
    "sanAssertionAttributes": [
        {
            "name": "string",
            "value": "string"
        },
        {
            "name": "string",
            "value": "string"
        }
    ]
},
"signAuthentication": {
    "senderProvidedNumbers": [
        "sample string 1",
        "sample string 2"
    ]
},
"socialAuthentications": null,
"templateAccessCodeRequired": false,
"templateLocked": false,
"templateRequired": false,
"canEditRecipientEmails": false,
"canEditRecipientNames": false
}].
```

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 characters.
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.
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:

Name	Reqd?	Type	Description
			<ul style="list-style-type: none">• emailBody: a string with the email message sent to the recipient. This can be a maximum of 2000 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 America (es-MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi). <p>IMPORTANT: If this is</p>

Name	Reqd ?	Type	Description
			<p>enabled for one recipient, it overrides the Envelope Subject and EmailBlurb.</p> <p>Also, you must enable emailNotification for all recipients.</p>
excludeDocuments	No	String	<p>Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account to use this.</p>
idCheckConfigurationName	No	String	<p>Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these names can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting.</p> <p>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 you would need to add phone number information to the smsAuthentication node.</p>
idCheckInformationInput	No		<p>This complex element contains input information related to a recipient ID check. It can include the following information.</p> <p>addressInformationInput: Used to set recipient address</p>

Name	Reqd ?	Type	Description
			<p>information and consists of:</p> <ul style="list-style-type: none"> • 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. <p>dobInformationInput: Used to set recipient date of birth information and consists of:</p> <ul style="list-style-type: none"> • 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

Name	Reqd?	Type	Description
			<p>information needs to be returned in the response.</p> <p>ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of:</p> <ul style="list-style-type: none"> • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. <p>Values are: ReadOnly, Editable, or DoNotDisplay.</p> <ul style="list-style-type: none"> • receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. <p>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:</p> <ul style="list-style-type: none"> • ssn9: Specifies the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. <p>Values are: ReadOnly, Editable, or DoNotDisplay.</p>

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	RecipientPhone Authentication	Optional element. Contains the elements: RecipMayProvideNumbers – Boolean (if true then recipient can use whatever phone number they want) SenderProvideNumbers – ArrayOfString (a list of phone numbers the recipient may use) RecordVoicePrint – Reserved ValidateRecipProvideNumber – Reserved
recipientAttachment	No	Attachment	It would be possible to send attachments with recipients. This complex element contains data in base64Binary. It also contains label and type for the attachment.
recipientId	No	String	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.

Name	Reqd?	Type	Description
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	Optional element, account must be set up to use SSO to use this. Contains the name/value pair information for the SAML assertion attributes: name – The name of the SAML assertion attribute. value – The value associated with the named SAML assertion attribute.
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: senderProvideNumbers – 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.
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.

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,
  "customFields": {
    "sample string 1",
    "sample string 2"
  }
},
```

```
"emailNotification": {
    "emailBody": " ",
    "emailSubject": " ",
    "supportedLanguage": " "
},
"excludedDocuments": [ "2", "4" ],
"idCheckConfigurationName": null,
"idCheckInformationInput": {
    "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",
    "receiveResponse": "sample string 2"
},
"dobInformationInput": {
    "dateOfBirth": "sample string 1",
    "displayLevelCode": "sample string 2"
},
"ssn4InformationInput": {
    "ssn4": "sample string 1",
    "displayLevelCode": "sample string 2",
    "receiveResponse": "sample string 3"
},
"ssn9InformationInput": {
    "ssn9": "sample string 1",
    "displayLevelCode": "sample string 2"
},
},
"inheritEmailNotificationConfiguration": false,
"note": "",
"phoneAuthentication": null,
"recipientAttachment": null,
"recipientId": "1",
"requireRelLookup": false,
"roleName": "",
"routingOrder": 3,
"sanAutentication": {
    "assertionAttributes": [
        {
            "name": "string",
            "value": "string"
        },
        {
            "name": "string",
            "value": "string"
        }
    ]
},
"snsAutentication": {
    "senderProvidedNumbers": [
        "sample string 1",
        "sample string 2"
    ]
},
"socialAuthentications": null,
"templateAccessCodeRequired": false,
"templateLocked": false,
```

```
"templateId": false,
}],
```

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 characters.
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.
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.

Name	Reqd?	Type	Description
			<p>It is composed of three elements:</p> <ul style="list-style-type: none">• emailBody: a string with the email message sent to the recipient. This can be a maximum of 2000 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 America (es-MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian

Name	Reqd?	Type	Description
			(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.
excludeDocuments	No	String	Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account to use this.
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 names 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 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.

Name	Reqd ?	Type	Description
			<p>addressInformationInput: Used to set recipient address information and consists of:</p> <ul style="list-style-type: none"> • 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. <p>dobInformationInput: Used to set recipient date of birth information and consists of:</p> <ul style="list-style-type: none"> • dateOfBirth: Specifies the recipient's date of birth. • displayLevelCode: Specifies the display level for the recipient. Values are: ReadOnly, Editable, or DoNotDisplay.

Name	Reqd?	Type	Description
			<ul style="list-style-type: none"> receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. <p>ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of:</p> <ul style="list-style-type: none"> 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. <p>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:</p> <ul style="list-style-type: none"> 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	RecipientPhone Authentication	Optional element. Contains the elements: RecipMayProvideNumbers – Boolean (if true then recipient can use whatever phone number they want) SenderProvideNumbers – ArrayOfString (a list of phone numbers the recipient may use) RecordVoicePrint – Reserved ValidateRecipProvideNumber – Reserved
recipientAttachment	No	Attachment	It would be possible to send attachments with recipients. This complex element contains data in base64Binary. It also contains label and type for the attachment.
recipientId	Yes	string	Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.

Name	Reqd?	Type	Description
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	Optional element, account must be set up to use SSO to use this. Contains the name/value pair information for the SAML assertion attributes: name – The name of the SAML assertion attribute. value – The value associated with the named SAML assertion attribute.
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: senderProvideNumbers – 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.

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,
    "customFields": {
        "sample string 1",
        "sample string 2"
    },
    "emailNotification": {
        "emailBody": " ",
        "emailSubject": " ",
        "supportedLanguage": " "
    },
    "excludedDocuments": [ "2", "4" ],
    "idCheckConfigurationName": null,
    "idCheckInformation": {
        "addressInformation": {
            "addressInformation": {
                "street1": "sample string 1",
                "street2": "sample string 2",
                "city": "sample string 3",
                "state": "sample string 4",
                "zip": "sample string 5"
            }
        },
        "displayLevelCode": "sample string 1",
        "receiveResponse": "sample string 2"
    },
    "dobInformation": {
        "dateOfBirth": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveResponse": "sample string 3"
    },
    "ssn4Information": {
        "ssn4": "sample string 4"
    }
}]
```

```

        "ssn4": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveEmailResponse": "sample string 3"
    },
    "ssn9InformationInput": {
        "ssn9": "sample string 1",
        "displayLevelCode": "sample string 2"
    }
},
"inheritEmailNotificationConfiguration": false,
"note": "",
"phoneAuthentication": null,
"recipientAttachment": null,
"recipientId": "1",
"requireRelLookup": false,
"roleName": "",
"routingOrder": 3,
"sanAutentication": {
    "sanAssertionAttributes": [
        {
            "name": "string",
            "value": "string"
        },
        {
            "name": "string",
            "value": "string"
        }
    ]
},
"snsAuthentication": {
    "senderProvidedNumbers": [
        "sample string 1",
        "sample string 2"
    ]
},
"socialAuthentication": null,
"templateAccessCodeRequired": false,
"templateLocked": false,
} ],
}

```

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 maximum of 100 characters.

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.
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: <ul style="list-style-type: none"> • emailBody: a string with the email message sent to the recipient. This can be a maximum of 2000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the

Name	Reqd ?	Type	Description
			<p>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 America (es-MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi).</p> <p>IMPORTANT: If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.</p>
excludeDocuments	No	String	Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account to use this.

Name	Reqd ?	Type	Description
idCheckConfigurationName	No	String	<p>Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these names can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting.</p> <p>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 you would need to add phone number information to the smsAuthentication node.</p>
idCheckInformationInput	No		<p>This complex element contains input information related to a recipient ID check. It can include the following information.</p> <p>addressInformationInput: Used to set recipient address information and consists of:</p> <ul style="list-style-type: none"> • 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:

Name	Reqd ?	Type	Description
			<p>street1/street2 = 150 characters, city = 50 characters, state = 2 characters and zip/zipPlus4 = 20 characters.</p> <ul style="list-style-type: none"> • 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. <p>dobInformationInput: Used to set recipient date of birth information and consists of:</p> <ul style="list-style-type: none"> • 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. <p>ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of:</p> <ul style="list-style-type: none"> • ssn4: Specifies the last four digits of the recipient's SSN.

Name	Reqd?	Type	Description
			<ul style="list-style-type: none"> • 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. <p>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:</p> <ul style="list-style-type: none"> • 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	<p>Optional element. Contains the elements:</p> <ul style="list-style-type: none"> RecipMayProvideNumbers – Boolean (if true then recipient can use whatever phone number they want) SenderProvideNumbers – ArrayOfString (a list of phone numbers the recipient may use) RecordVoicePrint – Reserved ValidateRecipProvideNumber – Reserved
recipientAttachment	No	Attachment	<p>It would be possible to send attachments with recipients. This complex element contains data in base64Binary. It also contains label and type for the attachment.</p>
recipientId	Yes	String	<p>Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.</p>
requireIdLookup	No	Boolean	<p>If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate their identity</p>
roleName	No*	String	<p>Optional element. This is the role name associated with the recipient.</p> <p>* - This is required when working with template recipients.</p>
routingOrder	Yes	String	<p>This specifies the routing order of the recipient in the envelope</p>

Name	Reqd ?	Type	Description
samlAuthentication	No	samlAssertionAttributes	Optional element, account must be set up to use SSO to use this. Contains the name/value pair information for the SAML assertion attributes: name – The name of the SAML assertion attribute. value – The value associated with the named SAML assertion attribute.
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: senderProvideNumbers – 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.

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",
    "addAccessCodeToEmail": false,
    "clientUserId": null,
    "customFields": {
        "sample string 1",
        "sample string 2"
    },
    "emailNotification": {
        "emailBody": " ",
        "emailSubject": " ",
        "supportedLanguage": " "
    },
    "excludedDocuments": ["2", "4"],
    "idCheckConfigurationName": null,
    "idCheckInformationInput": {
        "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",
        "receivedResponse": "sample string 2"
    },
    "dobInformationInput": {
        "dateOfBirth": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receivedResponse": "sample string 3"
    },
    "ssn4InformationInput": {
        "ssn4": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receivedResponse": "sample string 3"
    },
    "ssn9InformationInput": {
        "ssn9": "sample string 1",
        "displayLevelCode": "sample string 2"
    }
},
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": null,
    "recipientAttachment": null,
    "recipientId": "1",
    "requireIdLookup": false,
    "roleName": "",
    "routingOrder": 1,
    "sanAutentication": {
        "sanAssertionsAttributes": [
            {
                "name": "string",
                "value": "string"
            }
        ]
    }
}
```

```

        {
            "name": "string",
            "value": "string"
        }
    ],
    "snsAuthentication": {
        "senderProvidedNumbers": [
            "sample string 1",
            "sample string 2"
        ]
    },
    "socialAuthentications": null,
    "templateAccessCodeRequired": false,
    "templateLocked": false,
    "templateRequired": false,
    "canEditRecipientEmails": false,
    "canEditRecipientNames": false
},
]

```

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 maximum of 100 characters.
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.

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	<p>An optional complex type that has information for setting the language for the recipient's email information.</p> <p>This is composed of three elements:</p> <ul style="list-style-type: none"> • emailBody: a string with the email message sent to the recipient. This can be a maximum of 2000 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 of 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).

Name	Reqd?	Type	Description
			<p>(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).</p> <p>IMPORTANT: If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.</p>
excludeDocuments	No	String	<p>Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the</p>
idCheckConfigurationName	No	String	<p>account to use this. Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these names can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting.</p> <p>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</p>

Name	Reqd ?	Type	Description
			should be 'ID Check \$'. If you wanted to use SMS, it would be 'SMS Auth \$' and you would need to add you would need to add phone number information to the smsAuthentication node.
idCheckInformationInput	No		<p>This complex element contains input information related to a recipient ID check. It can include the following information.</p> <p>addressInformationInput: Used to set recipient address information and consists of:</p> <ul style="list-style-type: none"> • 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.

Name	Reqd?	Type	Description
			<p>dobInformationInput: Used to set recipient date of birth information and consists of:</p> <ul style="list-style-type: none"> • 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. <p>ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of:</p> <ul style="list-style-type: none"> • 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. <p>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:</p>

Name	Reqd ?	Type	Description
			<ul style="list-style-type: none"> • ssn9: Specifies the recipient's SSN. • DisplayLevel ^{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	RecipientPhone Authentication	Optional element. Contains the elements: RecipMayProvideNumbers – Boolean (if true then recipient can use whatever phone number they want) SenderProvideNumbers – ArrayOfString (a list of phone numbers the recipient may use) RecordVoicePrint – Reserved ValidateRecipProvideNumber – Reserved

Name	Reqd?	Type	Description
recipientAttachment	No	Attachment	It would be possible to send attachments with recipients. This complex element contains data in base64Binary. It also contains label and type for the attachment.
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	Optional element, account must be set up to use SSO to use this. Contains the name/value pair information for the SAML assertion attributes: name – The name of the SAML assertion attribute. value – The value associated with the named SAML assertion attribute.
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: senderProvideNumbers – Array with a list of phone numbers the recipient may use for SMS text authentication.

Name	Reqd ?	Type	Description
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.

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,
    "useDefault": null,
        "sampleString 1",
        "sampleString 2"
},
    "emailNotification": {
        "emailBody": "",
        "emailSubject": "",
        "supportedLanguage": ""
    },
    "excludedDocuments": [ "2", "4" ],
    "idCheckConfiguration": null,
    "idCheckInformation": {
        "addressInformation": {
            "street1": "sample string 1",
            "city": "sample string 3",
            "state": "sample string 4",
            "zip": "sample string 5",
            "zipPlus4": "sample string 6"
        },
        "displayLevelCode": "sample string 1",
        "receiveResponse": "sample string 2"
    },
    "dobInformation": {
        "dateOfBirth": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveResponse": "sample string 3"
    },
    "ssn4Information": {
        "ssn4": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveResponse": "sample string 3"
    },
    "ssn9Information": {
        "ssn9": "sample string 1",
        "displayLevelCode": "sample string 2"
    }
},
    "inheritEmailNotificationConfiguration": false,
    "note": "",
    "phoneAuthentication": null,
    "recipientAttachment": null,
    "recipientId": "2",
    "requireEmailLookup": false,
    "routingOrder": 1,
    "sanAutentication": {
        "assertionAttributes": [
            {
                "name": "string",
                "value": "string"
            },
            {
                "name": "string",
                "value": "string"
            }
        ]
    }
}
```

```
        },
        ],
        "smstAuthentification": {
            "senderProvidedNumbers": [
                "sample string 1",
                "sample string 2"
            ],
            "socialAuthentications": null,
            "tempalteAccessCodeRequired": false,
            "templateLocked": false,
            "templateRequired": false,
            "autoNavigation": false,
            "defaultRecipient": false,
            "sgnlnEachLocation": false,
            "signatureInfo": null,
            "signerName": "MC Clap Your Hands",
            "tabs": {
                "approveTabs": null,
                "checkboxTabs": null,
                "companyTabs": null,
                "dateSignedTabs": null,
                "declineTabs": null,
                "emailTabs": null,
                "envelopeTabs": null,
                "fullNamesTabs": null,
                "initialHereTabs": null,
                "listTabs": null,
                "noteTabs": null,
                "numberTabs": null,
                "radioGroupTabs": null,
                "signHereTabs": [
                    "signerAttachmentTabs": null,
                    "ssnTabs": null,
                    "textTabs": null,
                    "tipTabs": null
                ]
            }
        }
    }
}
```

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

Name	Reqd?	Type	Description
			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.
customFields	No	customField	An optional array of strings that allows the sender to provide custom data about the recipient. <small>This information is returned in the envelope status but otherwise not used by DocuSign. Each customField string can be a maximum of 100 characters.</small>
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: <ul style="list-style-type: none"> • emailBody: a string with the email message sent to the recipient. This can be a maximum of 2000 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),

Name	Reqd?	Type	Description
			<p>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).</p> <p>IMPORTANT: If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.</p>
excludeDocuments	No	String	<p>Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account to use this.</p>
idCheckConfigurationName	No	String	<p>Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these names can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting.</p> <p>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 you would need to add phone number information to the smsAuthentication node.</p>
idCheckInformationInput	No		<p>This complex element contains input information related to a recipient ID check. It can include</p>

Name	Reqd?	Type	Description
			<p>the following information.</p> <p>addressInformationInput: Used to set recipient address information and consists of:</p> <ul style="list-style-type: none"> • 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, 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. <p>dobInformationInput: Used to set recipient date of birth information and consists of:</p> <ul style="list-style-type: none"> • 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.

Name	Reqd?	Type	Description
			<p>ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of:</p> <ul style="list-style-type: none"> • 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. <p>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:</p> <ul style="list-style-type: none"> • 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	Recipient Phone Authentication	Optional element. Contains the elements: RecipMayProvideNumbers – Boolean (if true then recipient can use whatever phone number they want) SenderProvideNumbers – ArrayOfString (a list of phone numbers the recipient may use) RecordVoicePrint – Reserved ValidateRecipProvideNumber – Reserved
recipientAttachment	No	Attachment	It would be possible to send attachments with recipients. This parameter element contains data in base64Binary. It also contains label and type for the attachment.
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
roleName	No*	String	their identity 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	Optional element, account must be set up to use SSO to use this. Contains the name/value pair information for the SAML assertion attributes: name – The name of the SAML assertion attribute. value – The value associated with the named SAML assertion attribute.
smsAuthentication	No	senderProvide dNumbers	Optional element. Contains the element:

Name	Reqd?	Type	Description
			senderProvideNumbers – 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 <u>attributes of the recipient</u> .
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 CreateEnvelopeFromTemplatesAnd 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.

Name	Reqd?	Type	Description
signerName	No	SignerName	Optional element only used with recipient types In Person Signers and Signers. 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 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,
    "clientId": null,
    "customFields": {
        "sample string 1",
        "sample string 2"
    },
    "emailNotification": {
        "emailBody": " ",
        "emailSubject": " ",
        "supportedLanguage": " "
    },
    "excludedDocuments": [ "2", "4" ],
    "idCheckConfigurationName": null,
    "idCheckInformation": {
        "addressInformation": {
            "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",
            "receiveResponse": "sample string 2"
        },
        "dobInformation": {
            "dateOfBirth": "sample string 1",
            "displayLevelCode": "sample string 2",
            "receiveResponse": "sample string 3"
        },
        "ssnInformation": {

```

```

        "ssn4": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveEmailResponse": "sample string 3"
    },
    "ssn9InformationInput": {
        "ssn9": "sample string 1",
        "displayLevelCode": "sample string 2"
    },
}
"inheritEmailNotificationConfiguration": false,
"note": "",
"phoneAuthentication": null,
"recipientAttachment": null,
"recipientId": "1",
"requireRelLookup": false,
"roleName": "",
"routingOrder": 1,
"sanAutentication": {
    "sanAssertionAttributes": [
        {
            "name": "string",
            "value": "string"
        },
        {
            "name": "string",
            "value": "string"
        }
    ]
},
"sanAuthentication": {
    "senderProvidedNumbers": [
        "sample string 1",
        "sample string 2"
    ]
},
"socialAuthentication": null,
"templateAccessCodeRequired": false,
"templateLocked": false,
"canEditRecipientEmails": false,
"canEditRecipientNames": false
} ],
}

```

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 characters.

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.
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: <ul style="list-style-type: none"> • emailBody: a string with the email message sent to the recipient. This can be a maximum of 2000 characters. • emailSubject: a string with the subject of the email sent to the recipient. This can be a maximum of 100 characters. • supportedLanguage: the

Name	Reqd ?	Type	Description
			<p>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 America (es-MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi).</p> <p>IMPORTANT: If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.</p>
excludeDocuments	No	String	Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account to use this.

Name	Reqd ?	Type	Description
idCheckConfigurationName	No	String	<p>Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these names can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting.</p> <p>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 you would need to add phone number information to the smsAuthentication node.</p>
idCheckInformationInput	No		<p>This complex element contains input information related to a recipient ID check. It can include the following information.</p> <p>addressInformationInput: Used to set recipient address information and consists of:</p> <ul style="list-style-type: none"> • 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:

Name	Reqd ?	Type	Description
			<p>street1/street2 = 150 characters, city = 50 characters, state = 2 characters and zip/zipPlus4 = 20 characters.</p> <ul style="list-style-type: none"> • 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. <p>dobInformationInput: Used to set recipient date of birth information and consists of:</p> <ul style="list-style-type: none"> • 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. <p>ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of:</p> <ul style="list-style-type: none"> • ssn4: Specifies the last four digits of the recipient's SSN.

Name	Reqd?	Type	Description
			<ul style="list-style-type: none"> • 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. <p>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:</p> <ul style="list-style-type: none"> • 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	<p>Optional element. Contains the elements:</p> <ul style="list-style-type: none"> RecipMayProvideNumbers – Boolean (if true then recipient can use whatever phone number they want) SenderProvideNumbers – ArrayOfString (a list of phone numbers the recipient may use) RecordVoicePrint – Reserved ValidateRecipProvideNumber – Reserved
recipientAttachment	No	Attachment	<p>It would be possible to send attachments with recipients. This complex element contains data in base64Binary. It also contains label and type for the attachment.</p>
recipientId	Yes	String	<p>Unique for the recipient. It is used by the tab element to indicate which recipient is to sign the Document.</p>
requireIdLookup	No	Boolean	<p>If set true then the recipient is required to use the specified ID check method (including Phone and SMS authentication) to validate their identity</p>
roleName	No*	String	<p>Optional element. This is the role name associated with the recipient.</p> <p>* - This is required when working with template recipients.</p>
routingOrder	Yes	String	<p>This specifies the routing order of the recipient in the envelope</p>



samlAuthentication	No	samlAssertionAttributes	Optional element, account must be set up to use SSO to use this. Contains the name/value pair information for the SAML assertion attributes: name – The name of the SAML assertion attribute. value – The value associated with the named SAML assertion attribute.
smsAuthentication	No	senderProvidedNumbers	Optional element. Contains the element: senderProvideNumbers – 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.

Name	Reqd?	Type	Description
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.

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.

An example Signers json layout is shown below:

```
"signers": [ {
    "email": "signer.name@company.com",
    "name": "Mike Signer",
    "accessCode": "",
    "addAccessCodeToEmail": ,
    "clientUserId": ,
    "customFields": {
        "sample string 1",
        "sample string 2"
    },
    "emailNotification": {
        "emailBody": " ",
        "emailSubject": " ",
        "supportedLanguage": " "
    },
    "excludedDocuments": [ "2", "4" ]
}, idCheckConfiguration: ,
{idCheckInformation: {
    "addressInformation": {
        "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",
        "receiveNResponse": "sample string 2"
    },
    "dobInfo": {
        "dateOfBirth": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveNResponse": "sample string 3"
    },
    "ssn4Info": {
        "ssn4": "sample string 1",
        "displayLevelCode": "sample string 2",
        "receiveNResponse": "sample string 3"
    },
    "ssn9Info": {
        "ssn9": "sample string 1",
        "displayLevelCode": "sample string 2"
    }
},
"inheritEmailNotificationConfiguration": false,
"note": "",
"phoneAuthentication": null,
"recipientAttachment": null,
"recipientDokup": "1",
"roleName": "",
"routingOrder": 1,
"sanhAuthentication": {
    "sanhAssertionAttributes": [
        {
            "name": "string",
            "value": "string"
        },
        {
            "name": "string",
            "value": "string"
        }
    ]
},
$nsAuthentication": {
    "senderProvidedNumbers": [
        "sample string 1",
        "sample string 2"
    ]
},
"socialAuthentications": null,
"tenantAccessCodeRequired": false,
"tenantLocked": false,
"tenantRequired": false,
"autoNavigation": false,
"defaultRecipient": false,
"signEachLocation": false,
"signatureInfo": null,
"tabs": {
    "approveTabs": null,
    "checkboxTabs": null,
    "companyTabs": null,
    "dateSignedTabs": null,
    "dateTabs": null,
    "declineTabs": null,
    "emailTabs": null,
    "envelopeTabs": null,
    "fullNamesTabs": null,
    "initialHereTabs": null,
    "listTabs": null
}
```

```

    "not eTabs": null,
    "numberTabs": null,
    "radi oGroupTabs": null,
    "si gnHereTabs": [
      "si gnerAttachmentTabs": null,
      "ssnTabs": null,
      "textTabs": null,
      "ti tleTabs": null
    ],
  }
  "de i veryMet hod": "off l i ne",
  "de i ver edDateTi me": "String Content",
  "si gnedDateTi me": "String Content",
  "off l i neAttri butes": {
    "devi ceName": "String Content",
    "devi ceMbdel": "String Content",
    "gpsLat i tude": "String Content",
    "gpsLongi tude": "String Content",
    "account Esi gnl d": "String Content"
  }
},
],
}

```

-- [Multi part boundary (example of adding a Recipient Photo)]
Content-Type: image/jpeg
Content-Di spositi on: inline; recipi entId=1
Content-Transfer- Encodi ng: base64

-- [Multi part boundary (example of adding a Tab Image)]
Content-Type: image/gif
Content-Di spositi on: inline; tabl d=1
Content-Transfer- Encodi ng: base64

-- [Multi part boundary]

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 characters.
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.

Name	Reqd?	Type	Description
			If the clientUserId is not null then the recipient is embedded. * This is required if the signer is an offline signer.
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: <ul style="list-style-type: none"> • emailBody: a string with the email message sent to the recipient. This can be a maximum of 2000 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),

Name	Reqd?	Type	Description
			<p>Slovak (sk), Slovenian (sl), Spanish (es), Spanish Latin America (es-MX), Swedish (sv), Thai (th), Turkish (tr), Ukrainian (uk) and Vietnamese (vi).</p> <p>IMPORTANT: If this is enabled for one recipient, it overrides the Envelope Subject and EmailBlurb. Also, you must enable emailNotification for all recipients.</p>
excludeDocuments	No	String	<p>Specifies the documents that <u>are not</u> visible to this recipient. Document Visibility must be enabled for the account to use this.</p>
idCheckConfigurationName	No	String	<p>Specifies authentication check by name. The names used here must be the same as the authentication type names used by the account (these names can also be found in the web console sending interface in the Identify list for a recipient). This overrides any default authentication setting.</p> <p>Example: Your account has ID Check and SMS Authentication available and in the web console</p> <p>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 you would need to add phone number information to the smsAuthentication node.</p>
idCheckInformationInput	No		<p>This complex element contains input information related to a recipient ID check. It can include the following information.</p> <p>addressInformationInput: Used to set recipient address information and consists of:</p> <ul style="list-style-type: none"> • addressInformation: consists of six elements, with street2 and zipPlus4

Name	Reqd?	Type	Description
			<p>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.</p> <ul style="list-style-type: none"> • 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. <p>dobInformationInput: Used to set recipient date of birth information and consists of:</p> <ul style="list-style-type: none"> • 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. <p>ssn4InformationInput: Used to set the last four digits of the recipient's SSN information and consists of:</p> <ul style="list-style-type: none"> • ssn4: Specifies the last four digits of the recipient's SSN. • displayLevelCode: Specifies the display level for the recipient. Values are:

Name	Reqd?	Type	Description
			<p>ReadOnly, Editable, or DoNotDisplay.</p> <ul style="list-style-type: none"> receiveInResponse: A Boolean element that specifies if the information needs to be returned in the response. <p>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:</p> <ul style="list-style-type: none"> 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: RecipMayProvideNumbers – Boolean (if true then recipient can use whatever phone number they want) SenderProvideNumbers – ArrayOfString (a list of phone numbers the recipient may use) RecordVoicePrint – Reserved ValidateRecipProvideNumber –

Name	Reqd?	Type	Description
			Reserved
recipientAttachment	No	Attachment	It would be possible to send attachments with recipients. This complex element contains data in base64Binary. It also contains label and type for the attachment.
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	samlAssertion Attributes	Optional element, account must be set up to use SSO to use this. Contains the name/value pair information for the SAML assertion attributes: name – The name of the SAML assertion attribute. value – The value associated with the named SAML assertion attribute.
smsAuthentication	No	senderProvide dNumbers	Optional element. Contains the element: senderProvideNumbers – 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 used with the CreateEnvelopeFromTemplatesAndForms 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.
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	Sets if the signer is an offline signer. The only accepted value is: offline. * This is required if the signer is an offline signer.

Name	Reqd?	Type	Description
deliveredDateTime	No	DateTime	<p>Optional item if the signer is an offline signer. If this is not included, DocuSign will set it to the signedDateTime value. The date/time setting that specifies when the envelope was delivered to the recipient. The “sent” date for the recipient is set to the same value. Dates are formatted using ISO 8601 format. It is recommended that you include a time value. If no time value is included, the time will default to Pacific Time.</p>
signedDateTime	No*	DateTime	<p>The date/time setting that specifies when the recipient signed the envelope. Dates are formatted using ISO 8601 format. It is recommended that you include a time value. If no time value is included, the time will default to Pacific Time.</p> <p>* This is required if the signer is an offline signer.</p>
offlineAttributes	No		<p>Optional structure that contains attributes about the device used and location for an offline signer. You can include none, some or all of the attributes. This information is added to the Offline Signing section of the Certificate of Completion. The possible attributes are:</p> <ul style="list-style-type: none"> <u>deviceName</u>: A string with information about the type of device used for offline signing. <u>deviceModel</u>: A string with information about the model of the device used for offline signing. <u>gpsLatitude</u>: A string with the latitude of the device location at the time of signing. <u>gpsLongitude</u>: A string with the longitude of the device location at the time of signing. <u>accountEsignId</u>: A string with GUID of the account. This can be retrieved with the Get Consumer

Name	Reqd?	Type	Description
			Disclosure call.

Offline Signing Multipart section information:

Name	Reqd?	Type	Description
Recipient Photo	No	Image	<p>An optional multipart section that associates an offline signer image with the recipient, as set by the recipientId. The photo image is placed in the Offline Signing section of the Certificate of Completion.</p> <p>After the multipart boundary, add the content information:</p> <pre>Content-Type: image/{image file type} Content-Disposition: inline; recipientId={recipientId} Content-Transfer-Encoding: {encoding type}</pre>
Tab Image	No*	Image	<p>A multipart section with the signature or initial image that associates an offline signer's signature or initials with a signature or image tag, as set by the tabId. The image is placed in the appropriate tab location on the documents.</p> <p>After the multipart boundary, add the content information:</p> <pre>Content-Type: image/{image file type} Content-Disposition: inline; tabId={tabId} Content-Transfer-Encoding: {encoding type}</pre> <p>* This is required for offline signers that use signature or initial images.</p>

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 positions are established by setting offsets 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 example, "Borrower's Signature" might already exist in the document. If not, you could add the text string, "BorrowersSignHere".

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. 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. Positive anchorXOffset values move the tab right on the page and positive anchorYOffset values move the tab up on the page. The anchorUnits specifies the units used for the offsets.

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

- 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.
- The X and Y offset values must not extend a tab beyond the boundaries of the page. If the offset do extend beyond the page boundaries, the error message "Invalid_User_Offset" is sent. The error message includes the X or Y offset that resulted in the exception.
- 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/permuations. 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.

Example: In the JSON example below, the Text tabs "Student Last Name" and "Student First Name" will be auto-populated as specified ("Doe" and "John") each place they appear throughout the envelope.

```
"t abs": {
    "t ext Tabs": [
        {
            "t abLabel ": "\*St udent  L ast  N ame",
            "val ue": "Doe"
        },
        {
            "t abLabel ": "\*St udent  F irst  N ame",
            "val ue": "J ohn"
        }
    ]
}
```

- Each occurrence 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.

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:

```
"app roveTabs": [ {
    "anch orString": null,
    "anch orXOff set": null,
```

```

"anchorYOffset": null,
"anchorIgnoreIfNotPresent": null,
"anchorUnits": null,
"conditionalParentLabel": null,
"conditionalParentValue": null,
"documentId": "1",
"pageNumber": "1",

"recipientId": "1" false,
"templateLocked": false,
"templateRequired": false,
"xPosition": "300",
"yPosition": "100",
"bold": false,
"font": "Lucida Console",
"fontColor": null,
"fontSize": "size9",
"italic": false,
"tabLabel": "Approve 11",
"underline": false,
"buttonText": "Approve",
"height": 18,
"width": 60
},
]
}

```

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.
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	<p>Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility.</p> <p>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.</p>
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, Arial Narrow, Calibri, Courier New, Garamond, Georgia, Helvetica, Lucida Console, Tahoma, Times New Roman, Trebuchet, and Verdana.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, Bright Blue, Bright Red, Dark Green, Dark Red, Gold, Green, Navy Blue, 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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
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 declineTabs.
height	No	String	Height of the tab.
width	No	String	Width of the tab.

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,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "documentId": "1",
    "pageNumber": "1",
    "recipientId": "1",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "144",
    "yPosition": "155",
    "mergeField": null,
    "name": "Checkbox",
    "requireInitialOnSharedChange": false,
    "selected": false,
    "shared": false,
    "tabLabel": "Check Box 14"
}],
```

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.
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.
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.

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.
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.
mergeFieldXml	No	String	Reserved for future use.
requireInitialOnSharedChange	No	Boolean	Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.

Company Tab

Place this tag on the document where you want the recipient's company name to appear.

An example Company tab json layout is shown below:

```
"companyTabs": [ {
    "anchorString": null,
    "anchorXOffset": null,
    "anchorYOffset": null,
    "anchorIgnoreIfNotPresent": null,
    "anchorUnits": null,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,
    "documentId": "1",
    "pageNumber": "11",
    "templateId": "11",
    "templateLocked": false,
    "templateRequired": false,
    "xPosition": "394",
    "yPosition": "65",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Company"
}]
```

```

    "underline": false,
    "concealValueOnDocument": false,
    "disabled": false,
    "locked": false,
    "name": "Company",
    "required": true,
    "value": "Ro Corp 2",
    "width": 42
} ],

```

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.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab
conditionalParentValue	No	String	that controls this tab's visibility. 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.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
concealValueOnDocument	No	Boolean	<p>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.</p> <p>When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console.</p> <p>This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.</p>
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	The names associated with the tab type.
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.

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,
  "conditionalParentLabel": null,
  "conditionalParentValue": 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",
    }],
}
]

```

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.
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.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
name	No	String	The names associated with the tab type.
value	No	String	This element specifies the value of the tab.

Date Tab

Place this tag on the document where you want the recipient to enter a date. A date can be entered in any format and the date will be reformatted to MM/DD/YYYY. It uses the same parameters as a Text tab, with the validation message and pattern set for date information.

An example Date tab json layout is shown below:

```
"dateTabs": [ {
    "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": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Date Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disabledAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "mergeFieldXml": "",
    "requirementType": false,
    "shared": false,
    "height": 11,
} ],
```

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.
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.
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.
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.

Name	Reqd?	Type	Description
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
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	The names associated with the tab type.
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.
mergeFieldXml	No	String	Reserved for future use.
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.

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,
  "conditionalParentLabel": null,
  "conditionalParentValue": null,
  "documentId": "1",
  "pageNumber": "1",
  "recipientId": "1",
  "templateLocked": false,
  "templateRequired": false,
  "xPosition": "408",
  "yPosition": "105",
  "bold": false,
  "font": "Lucida Sans",
  "fontColor": null,
  "fontSize": "size9",
}
```

```

    "italic": false,
    "tabLabel": "Decline 12",
    "underline": false,
    "buttonText": "Decline",
    "height": 18,
    "width": 60
},
],

```

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.
conditionalParentLabel	No	String	Optional element. For conditional fields this is the TabLabel of the parent tab
conditionalParentValue	No	String	that controls this tab's visibility. 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.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
buttonText	No	String	The text displayed in the tab. Only used in approveTabs and declineTabs.
height	No	String	Height of the tab.
width	No	String	Width of the tab.

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.

An example Email tab json layout is shown below:

```
"emailTabs": [ {
    "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": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "Email Field 3",
    "underline": false,
    "concealValueOnDocument": false,
    "disabledAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "mergeFieldXml": "",
    "requirementType": false,
    "shared": false,
    "height": 11
}],
```

The Email tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.

Name	Reqd?	Type	Description
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.
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.
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.

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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
concealValueOnDocument	No	Boolean	<p>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.</p> <p>When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console.</p> <p>This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.</p>
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	The names associated with the tab type.
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.
mergeFieldXml	No	String	Reserved for future use.
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.

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,
"conditionalParentLabel": null,
"conditionalParentValue": null,

"documentId": "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"
},

```

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.
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
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 placed.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue,
fontSize	No	FontSize	Purple, or White. 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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the tab's value to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.
name	No	String	The names associated with the tab type.

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,
  "documentId": "1",
  "pageNumber": "1",
  "xPosition": "24",
  "yPosition": "153",
  "tabLabel": "Envelope ID"
}],
```

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.

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 annexed.
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.
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.
tabLabel	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.

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,
    "conditionalParentLabel": null,
    "conditionalParentValue": 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"
}],
```

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.
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.
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.

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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.
name	No	String	The names associated with the tab type.

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,
  "anchorXOffset": null,
  "anchorYOffset": null,
  "anchorIgnoreIfNotPresent": null,
  "anchorUnits": null,
  "documentId": "1",
  "pageNumber": "1",
  "recipientId": "1",
  "templateLocked": false,
  "templateRequired": false,
  "xPosition": "24",
  "yPosition": "153",
  "bold": false,
  "font": "arial",
  "fontColor": "#000000",
  "italic": false,
  "tabLabel": "Data Field 13",
  "underline": false,
  "concealValueOnDocument": false,
  "disableAutoSize": false,
  "locked": true,
  "name": "Formula",
  "required": true,
  "value": "",
  "width": 42,
  "mergeFieldXml": "",
  "height": 11,
} ],
```

The Formula tab parameters are:

Name	Reqd?	Type	Description
formula	No	String	<p>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.</p> <p>Example: Three tabs (TabLabels: Line1, Line2, and Tax) need to be added together. The formula string would be:</p> <p style="text-align: center;">[Line1]+[Line2]+[Tax]</p>

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.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.

Name	Reqd?	Type	Description
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	<p>The label used with the tab. This can be a maximum of 500 characters.</p> <p>Notes:</p> <p>Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.</p>
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	<p>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.</p> <p>When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console.</p> <p>This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.</p>
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	This should always be set to true for formula tabs.
name	No	String	The names associated with the tab type.
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.
mergeFieldXml	No	String	Reserved for future use.
height	No	String	Height of the tab.

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,
  "conditionalParentLabel": null,
  "conditionalParentValue": null,
  "documentId": "1",
  "pageNumber": "1",
  "recipientId": "1",
  "templateLocked": false,
  "templateRequired": false,
  "xPosition": "35",
  "bold": false,
  "font": "arial",
  "fontColor": null,
  "fontSize": null,
  "italic": false,
  "tabLabel": "Full Name",
  "underline": false,
  "name": "Full Name"
}],
```

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.

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.
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.
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.
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.

Name	Reqd?	Type	Description
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.
name	No	String	The names associated with the tab type.

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",
  "conditionalParentLabel": "null",
  "conditionalParentValue": "null",
  "documentId": "1",
  "pageNumber": "1",
  "recipientId": "1",
  "templateLocked": false,
```

```

    "template": false,
    "xPosition": "249",
    "yPosition": "15",
    "name": "Initial Here",
    "optional": false,
    "scaleValue": 1,
    "tabLabel": "Initial 5"
  },
],
},

```

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.
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.
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.
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.
name	No	String	The names associated with the tab type.
optional	No	Boolean	When true, the Initial Here tab is optional and the recipient is not required to add initials to complete an envelope.
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.
tabLabel	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.

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": [{}]
```

```

    "templaterequired": false,
    "xPosition": "399",
    "yPosition": "35",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "tabLabel": "false>Last Name",
    "underline": false,
    "name": "Last Name"
},
]

```

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.
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.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
name	No	String	The names associated with the tab type.

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,
    "conditionalParentLabel": null,
    "conditionalParentValue": 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"
    }
],
"mergeFieldId": "",
"requireInitialOnSharedChange": false,
"shared": false,
"value": "Pear",
"width": 48
}],
```

The List tab parameters are:

Name	Reqd?	Type	Description
anchorString	No	String	Specifies string searched for to place the tab in the document.

Name	Reqd?	Type	Description
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.
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.
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.

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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
listItems	Yes	listItem	<p>Only used with list tabs. These are the possible selections for a dropdown list. Each selection has three parts:</p> <ul style="list-style-type: none"> • 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.
mergeFieldXml	No	String	Reserved for future use.
requireInitialOnSharedChange	No	Boolean	<p>Optional element for field markup. When set to true it requires the signer to initial when they modify a shared field.</p>
shared	No	Boolean	<p>Optional element for field markup. When set to true, enables field markup for the field.</p>
Value	No	String	This sets/gets the currently selected list value.
width	No	String	Width of the tab.

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,
    "conditionalParentLabel": null,
    "conditionalParentValue": null,

    "documentId": "1",
    "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
},
]

```

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.
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.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
height	No	String	Height of the tab.
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.

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.

An example Number tab json layout is shown below:

```
"numberTabs": [ {
  "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": "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,
  "mergeFieldXml": "",
  "requireRehitOnSharedChange": false,
  "shared": false,
  "height": 11,
}],
```

The Number tab parameters are:

Name	Reqd?	Type	Description
------	-------	------	-------------

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.
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.
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.

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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	<p>The label used with the tab. This can be a maximum of 500 characters.</p> <p>Notes:</p> <p>Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.</p>
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
concealValueOnDocument	No	Boolean	<p>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.</p> <p>When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console.</p> <p>This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.</p>
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	The names associated with the tab type.
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.
mergeFieldXml	No	String	Reserved for future use.
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.

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,
"documentId": "1",
"groupName": "Radio Group 1",
"radios": [
    {
        "anchorString": null,
        "anchorXOffset": null,
        "anchorYOffsetNotPresent": null,
        "anchorUnits": null,
        "locked": false,
        "pageNumber": "1",
        "required": false,
        "selected": false,
        "value": "Radio",
        "xPosition": "234",
        "yPosition": "159"
    },
    {
        "anchorString": null,
        "anchorXOffset": null,
        "anchorYOffset": null,
        "anchorYOffsetNotPresent": null,
        "locked": false,
        "pageNumber": "1",
        "required": false,
        "selected": true,
        "value": "Radio",
        "xPosition": "236",
        "yPosition": "190"
    }
],
"recipientId": "1",
"requireInitialOnSharedChange": false,
"shared": false,
"templateLocked": false,
"templateRequired": false
]
}

```

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.
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.
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
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	<p>This sets the locations and status for radio buttons that are grouped together. The information for each radio button is::</p> <p>anchorString: Optional anchor text information for the radio button. If it is not used then the other anchor parameters are ignored</p> <p>anchorXOffset: This specifies tab location as XOffset position, using the anchorUnits, from the anchorString.</p> <p>anchorYOffset: This specifies tab location as YOffset position, using the anchorUnits, from the anchorString.</p> <p>anchorIgnoreIfNotPresent: True or false setting. If true, this tab is ignored if anchorString is not found in the document.</p> <p>anchorUnits: Specifies units of the X and Y offset. Units could be pixels, mms, cms or inches.</p> <p>locked: If true, the Signer cannot change the data in the tab.</p> <p>pageNumber: sets the page the radio button is located on.</p> <p>required: If true, the signer is required to fill out this tab.</p> <p>selected: Sets if this is radio button is selected. Use true/false to show the value is selected or not. Only one radio button can be true.</p> <p>value: if an X, this is the selected radio button. Only one radio button in a group can be selected.</p> <p>xPosition: This indicates the horizontal offset of the radio button on the page.</p> <p>yPosition: This indicates the vertical offset of the radio button on the page.</p>
recipientId	Yes	String	This specifies the recipient associated with the tab. Must refer to an existing recipient's ID attribute.

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.
shared	No	Boolean	Optional element for field markup. When set to true, enables field markup for the field.
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,
  "conditionalParentLabel": null,
  "conditionalParentValue": 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"
},
],
]
```

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.

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.
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.
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.
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.

Name	Reqd?	Type	Description
name	No	String	The names associated with the tab type.
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.
tabLabel	Yes	String	<p>The label used with the tab. This can be a maximum of 500 characters.</p> <p>Notes:</p> <p>Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.</p>

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,

  "xPosition": "24",
  "yPosition": "153",
  "tabLabel": "Attachment 10",
  "required": true
}],
```

The Signer Attachment tab parameters are:

Name	Read?	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.
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.
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.
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.

Name	Reqd?	Type	Description
tabLabel	Yes	String	<p>The label used with the tab. This can be a maximum of 500 characters.</p> <p>Notes:</p> <p>Making a tab's TabLabel the same will cause the all the fields to update when the user enters data.</p>
required	No	Boolean	If true, the signer is required to fill out this tab.

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.

An example SSN tab json layout is shown below:

```
"ssnTabs": [ {
    "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": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "italic": false,
    "tabLabel": "SSN Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disabledAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "mergeFieldXml": "",
    "recurringInitialSharedChange": false,
    "shared": false,
    "height": 11,
} ],
```

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.
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.
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.

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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	<p>The label used with the tab. This can be a maximum of 500 characters.</p> <p>Notes:</p> <p>Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.</p>
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
concealValueOnDocument	No	Boolean	<p>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.</p> <p>When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console.</p> <p>This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.</p>
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	The names associated with the tab type.
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.
mergeFieldXml	No	String	Reserved for future use.
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.

Text Tab

This tag is an adaptable field that allows the recipient to enter different text information.

An example Text tab json layout is shown below:

```
"textTabs": [ {  
    "anchorString": null,
```

```

"anchorXOffset": null,
"anchorYOffset": null,
"anchorIgnoreIfNotPresent": null,
"anchorUnits": null,
"conditionalParentLabel": null,
"conditionalParentValue": null,
"documentId": "1",
"pageNumber": "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,
"required": true,
"value": "",
"width": 42,
"mergeFieldXml": "",
"requireInitialOnSharedChange": false,
"shared": false,
"valiadationMessage": "",
"validationPattern": "",
"height": 11,
"isPaymentAmount": false
}],
```

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.
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	<p>Optional element. For conditional fields this is the Value of the parent tab that controls this tab's visibility.</p> <p>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.</p>
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
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 required	No No	String Boolean	The names associated with the tab type. 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.
mergeFieldXml	No	String	Reserved for future use.

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.
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.
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.

Title Tab

Place this tag on the document where you want the recipient's title to appear.

An example Title tab json layout is shown below:

```
"titleTabs": [ {
  "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": "30",
  "yPosition": "103",
  "bold": false,
  "font": "arial",
  "fontColor": null,
  "fontSize": null,
  "italic": false,
```

```

    "tabLabel": "Title",
    "underline": false,
    "concealValueOnDocument": false,
    "disabledAutoSize": false,
    "locked": false,
    "name": "Title",
    "required": true,
    "tabType": "Text"
  },
]
}
],

```

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.
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.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
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	Yes	String	The label used with the tab. This can be a maximum of 500 characters. Notes: Making a tab's TabLabel the same will cause the all like fields to update when the user enters data.
underline	No	Boolean	If true, the information in the tab is underlined.

Name	Reqd?	Type	Description
concealValueOnDocument	No	Boolean	<p>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.</p> <p>When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console.</p> <p>This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.</p>
disableAutoSize	No	Boolean	<p>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.</p>
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	The names associated with the tab type.
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.

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.

An example Zip tab json layout is shown below:

```
"zipTabs": [ {
    "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": "24",
    "yPosition": "153",
    "bold": false,
    "font": "arial",
    "fontColor": null,
    "fontSize": null,
    "tabLabel": "Zip Field 13",
    "underline": false,
    "concealValueOnDocument": false,
    "disabledAutoSize": false,
    "locked": false,
    "name": "Text",
    "required": true,
    "value": "",
    "width": 42,
    "mergeFieldXml": "",
    "relativeOnSharedChange": false,
    "shared": false,
    "height": 11,
},
]
}
  
```

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.
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.
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
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 placed.
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.
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.
bold	No	Boolean	If true, the information in the tab is bold.
font	No	Font	The font type used for the information in the tab. Possible values are: Arial, ArialNarrow, Calibri, CourierNew, Garamond, Georgia, Helvetica, LucidaConsole, Tahoma, TimesNewRoman, Trebuchet, and Verdana.
fontColor	No	FontColor	The font color used for the information in the tab. Possible values are: Black, BrightBlue, BrightRed, DarkGreen, DarkRed, Gold, Green, NavyBlue,
fontSize	No	FontSize	Purple, or White. 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	Yes	String	<p>The label used with the tab. This can be a maximum of 500 characters.</p> <p>Notes:</p> <p>Making a tab's TabLabel the same will cause the tab-like fields to update when the user enters data.</p>
underline	No	Boolean	If true, the information in the tab is underlined.
concealValueOnDocument	No	Boolean	<p>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.</p> <p>When an envelope is completed the information is available to the sender through the Form Data link in the DocuSign Console.</p> <p>This setting applies only to text boxes and does not affect list boxes, radio buttons, or check boxes.</p>
disableAutoSize	No	Boolean	<p>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.</p>
locked	No	Boolean	If true, the Signer cannot change the data in the tab.
name	No	String	The names associated with the tab type.
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.
mergeFieldXml	No	String	Reserved for future use.
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.

Name	Reqd?	Type	Description
height	No	String	Height of the tab.

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 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 (`errorDetails.errorCode`) and associated messages (`errorDetails.message`). Note that in many cases, additional information is added to the message string.

Error Code	Message (Base)
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

Error Code	Message (Base)
ACCOUNT_LACKS_PERMISSIONS	This Account lacks sufficient permissions.
ACCOUNT_LACKS_PRICING_PLAN	The specified Account lacks an Account Pricing Plan
ACCOUNT_NOTAUTHORIZEDFOR_ENVELOPE	This account is not authorized to access the requested envelope.
ACTIVATION_FAILED	The token for a recipient that has failed activation cannot be generated.
ADDRESSBOOK_CANTSHARE	Invalid address book share permissions.
ADDRESSBOOK_EMAIL_INVALID	Invalid Email for the address book entry.
ADDRESSBOOK_ID_INVALID	Invalid ID for the address book entry.
ADDRESSBOOK_NOTOWNER	Invalid Owner for the address book entry.
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.
BILLING_PLAN_ERROR	The billing plan could not be changed.
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_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

Error Code	Message (Base)
	Editor 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.
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.
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.
CUSTOMTAB_IS_INCOMPLETE	A Custom Tab is not Complete. A Custom Tab requires both a Name and a TabLabel.
DOCUMENT_DOES_NOT_EXIST	The document specified was not found.
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.
ENVELOPE_AC_EXPORT_COMPLETED	Envelope already exported.

Error Code	Message (Base)
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_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_VOID_INVALID_STATE	Only envelopes in the 'Sent' or 'Delivered' states may be voided.
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_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

Error Code	Message (Base)
ENVELOPE_ONLYTHREECUSTOMFIELDS_ALLOWED	Only three custom fields allowed per envelope.
ENVELOPE_TRANSFEREE_ALREADY_OWNS_ENVELOPE	The specified transfer recipient of the envelope is already the owner of the envelope.
FAILED_EMAIL_SENDING	Email sending failed.
FAILED_SCHEMA_VALIDATION	Schema validation Failed. Additional Errors follow.
FAX_BACK_MUST_BE_SIGNER	Fax Recipient Token must be of type Signer.
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.
FORMAT_CONVERSION_ERROR	The data could not be converted.
FORMULA_CIRCULAR_REFERENCE	Recipient Calculated Fields have a circular reference.
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_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_APPTOKEN	AppToken is invalid or distributor not found for AppToken
INVALID_BILLING_PLAN_RETIRE	The billing plan is retired and cannot be used.
INVALID_BILLING_PLAN_SUCCESSOR	Invalid Successor Billing Plan.

Error Code	Message (Base)
INVALID_CAPTIVE_RECIPIENT_OPERATION	Invalid operation on captive recipient.
INVALID_CHARACTERS_PROVIDED	Invalid characters detected
INVALID_CONTENT_TYPE	Content Type specified is not supported.
INVALID_CONTENT_TRANSFER_ENCODING	Invalid 'content transfer encoding' for envelope.
INVALID_CREDIT_CARD_INFORMATION	The credit card information supplied was not valid.
INVALID_DISTRIBUTOR	The Distributor specified is not valid.
INVALID_DOCUMENT_ID	The DocumentId provided 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_EXPIRATION_RemINDER_DATA	Invalid envelope expiration and/or reminder information provided.
INVALID_FAXNUMBER	Fax Number is invalid.
INVALID_FOLDER_TYPE	The folder type specified is not valid.
INVALID_MULTI_PART_REQUEST	An error was found while parsing the multipart request.
INVALID_PASSWORD	The password is invalid.
INVALID_PASSWORD_CHALLENGE	Invalid forgotten password challenge.
INVALID_PLANID	The PlanId specified is not valid.
INVALID_RECIPIENT_ID	A recipient ID is missing or invalid.
INVALID_RECIPIENT_STATUS_FOR_DELETE	Invalid recipient status for delete.
INVALID_REQUEST_BODY	The request body is missing or improperly formatted.
INVALID_REQUEST_PARAMETER	The request contained at least one invalid

Error Code	Message (Base)
	parameter.
INVALID_REQUEST_RESPONSE_OVERRIDE_FORMAT	The URL override format is not supported. Use supported formats such as .json or .xml.
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_TAB_FORMULA	The Tab formula is not valid.
INVALID_TAB_OPERATION	The Tab specified is not valid for the requested operation.
INVALID_TOKEN_FORMAT	The security token format does not conform to expected schema.
INVALID_USER_OFFSET	The specified User Offset exceeds the page boundaries.
INVALID_USERID	Invalid UserId.
INVALID_USERNAME	The user name is invalid.
INVALID_USERNAME_FOR_RECIPIENT	The user name for the recipient is invalid.
INVALIDAUTHENTICATIONSETUP	Authentication is not setup correctly for the recipient.
JANRAIN_REQUEST_ERROR	Janrain request Error.
MAX_MEMBERS_EXCEEDED	The maximum number of members for the account has been exceeded.
MAX_TABS_EXCEEDED	The maximum number of tabs was exceeded.
METHOD_NOT_ALLOWED	The HTTP method specified is not allowed with this resource.
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.

Error Code	Message (Base)
MUST_ENABLE_SIGNONPAPER	EnableWetSign must be set on the envelope.
NO_DOCUMENT RECEIVED	The document element did not contain the encoded document, or there is a problem with the encoding.
NOSATABONSADOCUMENT	No signer attachment tabs found on a signer attachment document.
NOT_SUPPORTED	Functionality not supported.
ONESIGNALSIGN NOT_SATISFIED	Freeform signing is not allowed for your account because it conflicts with other settings, please place signing tabs for each signer.
PARTNER_NO_AUTHORITY_FOR_SENDER	The sender specified is not part of a group that the Partner is authorized to send for.
PDF_INACCESSIBLE	PDF URL Inaccessible
PLAN_ITEM_NOT_ENABLED	A requested plan item is not enabled for this account.
PURCHASED_ENVELOPES_NOT_ADDED	Purchased envelopes not added.
RECIPIENT_ALREADY_EXISTS_IN_ENVELOPE	This recipientId already exists.
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_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.

Error Code	Message (Base)
RECIPIENT_NOT_SIGNER	Recipient is not a signer type.
RECIPIENT_UPDATE_FAILED	The recipient could not be updated.
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.
REQUIRED_SECURITY_CHECK_URL_MISSING	A security check was specified for the recipient but the appropriate callback URL was not provided.
RESOURCE_NOT_FOUND	The URL provided does not resolve to a resource.
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.
SIGNATURE_NOT_FOUND	The signature specified was not found.
SIGNER_RECIPIENT_HAS_NO_SIGNABLE_TABS	One or more Signer type recipients have not been assigned any signable tabs.
SOBO_USER_NOT_ALLOWED	A 'Send on Behalf Of' user is not permitted for this operation.
SOCIAL_PROVIDER_ALREADY_MAPPED	This social provider account is already mapped in the system.
TAB_MERGEFIELD_VALUE_NOT_RETRIEVED	A Mergefield value for a Tab could not be retrieved.
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.

Error Code	Message (Base)
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_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_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)
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_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.
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_NOT_ACCOUNT_ADMIN	User is not an account administrator.
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
USERNAME_IS_NOT_ALLOWED	The specified username(s) and email(s) are not in our system and the email does not allow new users.

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 URLs.

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} <i>Use recyclebin as folderId</i> PUT
GetAccountDistributorCode (AcctMgmt)	/accounts/{accountId} GET
GetAccountInformation (AcctMgmt)	/accounts/{accountId} GET
GetAccountSettings (AcctMgmt)	/accounts/{accountId}/settings GET
GetAccountSettingsList (dsapi)	/accounts/{accountId}/settings GET
GetAuthenticationToken (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

DocuSign SOAP	REST URI and HTTP Method
RequestStatus (dsapi)	/accounts/{accountId}/envelopes/{envelopeId} GET
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 than 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.

~~The Time Zone used for API options: 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 following date/time formats are accepted in the DocuSign APIs:

- "M/d/yyyy"
- "MMMM d, yyyy"
- "dd-MM-yy"
- "dd-MM-yyyy"
- "MM-M-d-yyyy"
- "MMMM d, yyyy | HH:mm"
- "dd-MM-yy | HH:mm"
- "dd-MM-yyyy | HH:mm"
- "MMM-dd-yyyy | HH:mm"
- "yyyy-MM-dd"
- "yyyy-MM-dd | HH:mm"