

eSign API documentation



#95, Rudra Chambers, 3rd Floor, 4th Main Road,
11th Cross, Malleshwaram, Bangalore - 560003
Phone: 080-46682650 www.signdesk.com

Document Number	Last Updated	Version ID	Information Classification
SDES25022020	03-12-2023	V3.1.0	Confidential

Prepared By	Title
Rachita Patro	Product Support Engineer
Approved By	Title
Viveka B C	Tech Team Lead

Disclaimer

This document is for the sole use of the intended recipient(s) and contains confidential and privileged information. Any unauthorized review, use, disclosure, copying, or distribution is strictly prohibited.

Revision History

Revision Number	Version ID	Revision Date	Page No	Revision Description	Author	Approved By
0	1	26/12/2017		Document Created	Rachitha	Vivek
1	1.0.1	12/03/2018		<ul style="list-style-type: none"> 1. Added email notification, type of eSign feature 2. Enhancements in eSign widget appearance 	Rachitha	Vivek
2	2.0.0	19/07/2018	6	<ul style="list-style-type: none"> 1. Initial Draft of eSign 2.1 integration - the title of the document is changed 2. Added a new note 3. Workflow - updated 	Rachitha	Vivek
3	2.0.1	09/08/2018		<ul style="list-style-type: none"> 1. Provision for entering both Aadhaar number and VID is added 	Rachitha	Vivek
4	3.0.1	15/02/2019		<ul style="list-style-type: none"> 1. Facility for Electronic signature added 2. Added signer_mobile parameter 	Rachitha	Vivek
5	3.0.2	16/02/2019		<ul style="list-style-type: none"> 1. Addition of Signer validation 	Rachitha	Vivek
6	3.0.4	30/12/2019		<ul style="list-style-type: none"> 1. Custom parameter functionality added 	Abhijith Koushik B K	Sandesh Acharaya
7	3.0.5	25/02/2020		<ul style="list-style-type: none"> 1. Addition of validations in widget_configuration 	Abhijith Koushik B K	Sandesh Acharaya
8	3.0.6	01/10/2020		<ul style="list-style-type: none"> 1. Addition of latest error codes 	Abhijith Koushik B K	Sandesh Acharaya
9	3.0.7	14/09/2022		<ul style="list-style-type: none"> 1. Addition of supporting_document_required parameter in eSignrequest API and updation of its field option. 	Abhijith Koushik B K	Sandesh Acharaya
10	3.0.8	02/11/2022		<ul style="list-style-type: none"> 1. Updation of latest error codes 	Abhijith Koushik B K	
11	3.0.9	03/12/2023		<ul style="list-style-type: none"> 1. Updation of latest Validations 	Abhijith Koushik B K	
12	3.1.0	01/02/2024	33	Updation of request body and its parameters	Veena G	

Table of Contents

Revision History	2
Introduction	5
About Desk Nine Pvt Ltd	5
SignDesk.com	5
Workflow of the eSign widget	6
eSign Widget APIs	8
eSignRequest API	8
Header	13
Request Parameters	14
Response Parameters	26
JavaScript to call Widget	28
Sample Code to integrate Widget	31
widgetToken API	32
Header	33
Request Parameters	33
Response Parameters	33
getDocketInfo API	34
Header	35
Request Parameters	35
Response Parameters	35
getSignatureStatus API	36
Header	37
Request Parameters	37
Response Parameters	37
updateEsignSignerValidationInputs API	38
Header	39
Request Parameters	39
Response Parameters	41
Resend Invitation API	42
Header	43
Request Parameters	43
Response Parameters	43
updateEsignDocumentInputs API	44
Annexure 1: Error Codes	45
Sample Error Messages	49
Annexure 2: Integration Checklist	52

Introduction

About Desk Nine Pvt Ltd

Desk Nine Pvt. Ltd. is an ISO 27001:2013 and 9001:2015 certified technology company that was started to provide seamless & AI-powered digital documentation solutions. In order to comprehensively solve the problems and challenges caused due to the traditional documentation process, Desk Nine Pvt. Ltd has created an entity to automate & digitize paper-based documentation & manual workflows - SignDesk.com. SignDesk offers contract workflow automation, electronic evidence creation, digital KYC verification & onboarding, and automated recurring payments.

SignDesk.com

SignDesk.com offers a Contract Workflow Automation Solution, developed exclusively to ease paperwork and documentation, and aimed primarily at the financial sector. Our digital documentation product provides end-to-end automated documentation, required for business processes like Customer Onboarding, Loan Disbursal Documentation, Vendor Onboarding, document execution, and many others. Video-based ID verification & onboarding, Aadhaar-based eSign & video signatures, Digitized Stamping, and Automated Recurring Payments via eNACH are the pillars of our Document Workflow Automation module.

Our nation-wide Digital Stamping Solutions help people print legal documents on stamp paper by using our online platform from anywhere. Our eNACH service helps banks and other financial organizations automate their recurring payments digitally, and thereby avoiding a lot of paperwork and saving time. Our legally compliant electronic signatures & innovative video signatures allow remote & instant document execution with seamless electronic evidence creation. Our AI-powered KYC verification solution automates the KYC verification & onboarding process with VCIP & VBIP-enabled digital workflows. We, at SignDesk, offer services that are in compliance with government policies and are legally valid. Our efforts to digitize inefficient paper-based methods have been recognized and awarded, the most recent being the Global Banking & Finance Review's Best Digital Onboarding Product of 2020 in India, Inflection's Best Digital Stamping Solution of 2020, and InnTech's Best AI/ML Product of 2020.

Workflow of the eSign widget

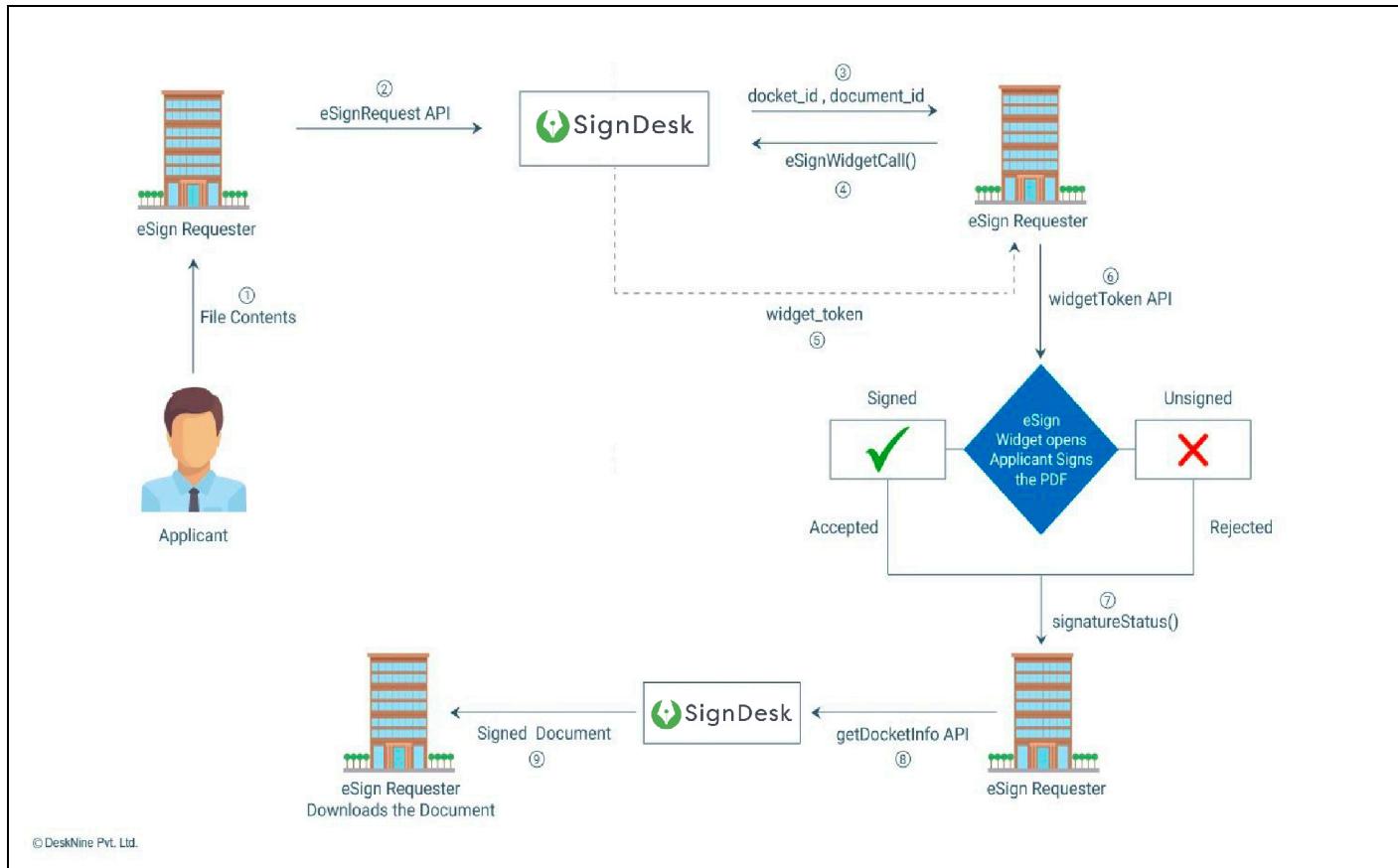


Fig 1: Workflow of the eSign widget

1. Applicant data is fetched by the eSign Requester.
2. The eSign Requester can request an **eSignRequest** API provided by SignDesk to send the applicant's document contents.
Note: eSign Requester is the entity performing eSign transactions for its customers and utilizing the services provided by SignDesk.
3. SignDesk will send docket_id and document_id to the eSign Requester in response to the **eSignRequest** API call.
4. eSign Requester will have to call the JavaScript **eSignWidgetCall()**, using the docket_id and document_id provided by SignDesk to generate the widget_token. (This JavaScript function is explained in the further sections of the document). eSign Requester will get the widget_token in JavaScript **getWidgetToken()**.

5. The `widget_token` is shared with the eSign Requester by SignDesk. After receiving the `widget token`, the eSign Requester will pass this `widget token` to the **`widgetToken`** API provided by SignDesk to capture the `widget token`.
6. Once the **`widgetToken`** API is successful, the **eSign** widget will open.
7. This **eSign** widget will have the applicant's document to be signed. This widget enables the applicant to sign the document using a VID/Aadhaar number through OTP/Biometric.
8. After this, the applicant can sign the document. Once the document is signed, the **eSign** widget will get closed and the signature status will be sent to the eSign Requester by SignDesk through JavaScript **`signatureStatus()`**.
9. After receiving the `signatureStatus()` result, the eSign Requester makes the `getDocketInfo` API call to fetch the signed document from the eSignDesk server.
10. SignDesk sends the applicant's eSigned document to the eSign Requester.

Note:

1. "Applicant" refers to the user/ signer
2. "eSign Requester" refers to the corporate through which the applicant is requesting the eSign widget.

Important Note:

As per new CCA guidelines, please note that for Aadhaar pre-filling in the LD widget, if the Aadhaar number is currently being passed in the eSign 2.0 API parameter, then this feature will not be available in the eSign 2.1 API.

eSign Widget APIs

SignDesk provides three APIs to enable an applicant to create and sign documents online using an eSign widget through the eSign requestor. The APIs are listed below:

1. eSignRequest API
2. widgetToken API
3. getDocketInfo API

Note: The above API calls should be made from the server side, not from the client side.

eSignRequest API

DOCUMENT UPLOAD API	
Feature	Facilitates eSign Requester to upload the document to be eSigned.
HTTP Type	POST
Header	x-parse-rest-api-key x-parse-application-id
Request	{ "reference_id": "xxxx", "mode": "draft", "enable_organisation_seal": "xxxx", "docket_title": "xxxx", "remarks": "xxxx", "enable_email_notification": "xxxx", "docket_description": "xxxx", "description_in_invitation": "xxxx", "final_copy_recipients": "xxxx,xxxx", "notification_email_list": "xxxx,xxxx", "final_copy_phone_number_list": "xxxx;xxxx", "supporting_document_required": "xxxx", "expiry_date": "xxxx", "documents": [{ "reference_doc_id": "xxxx", "content_type": "xxxx", "content": "xxxx", "signature_sequence": "xxxx", },] }

```
"return_url": "xxxx",
"custom_parameter":
{
    "parameter_name": "xxxxxxxx"
},
}],
"callback_file_content": "xxxx",
"supporting_documents":
[{
    "name": "xxxx",
    "content_type": "xxxx",
    "content": "xxxx"
}],
"signers_info":
[{
    "document_to_be_signed": "xxxx",
    "signer_position":
    {
        "appearance": "xxxx"
    },
    "signer_ref_id": "xxxx",
    "signer_email": "xxxx",
    "signer_name": "xxxx",
    "sequence": "xxxx",
    "page_number": "xxxx",
    "esign_type": "xxxx",
    "signer_mobile": "xxxx",
    "signer_remarks": "xxxx",
    "authentication_mode": "xxxx",
    "signer_validation_inputs":
    {
        "year_of_birth": "xxxx",
        "gender": "xxxx",
        "name_as_per_aadhaar": "xxxx",
        "last_four_digits_of_aadhaar": "xxxx"
    },
    {
        "geofencing": true,
        "geofencing_data":
        {
            "latitude": "xxxx",
            "longitude": "xxxx",
            "radius": "xxxx"
        }
    }
}]
```

```
        },
        "signature_type": "xxxx",
        "trigger_esign_request": true,
        "access_type": "xxxx",
        "captureSignerImage": xxxx,
        "group_reference_id": "xxxx",
        "trigger_esign_request_invitation": "xxxx",
        "whatsapp_invitation_mode": "xxxx"
    }],
    "signer_group_info":
    [
        {
            "group_reference_id": "xxxx",
            "mandatory_signer_count": "xxx",
            "group_name": "xxxx"
        }
    ],
    "widget_configuration":
    {
        "widget_title": "xxxx",
        "background_color": "xxxx",
        "button_color": "xxxx",
        "button_font_color": "xxxx",
        "font_family": "xxxx",
        "esign_with_otp_button_text": "xxxx",
        "esign_with_biometric_button_text": "xxxx",
        "user_instruction": "xxxx",
        "remark_field_label": "xxxx",
        "usb_signature_title": "xxxx"
    }
},
"reviewers_info":
[
    {
        "document_to_be_reviewed": "xxxxx",
        "stakeholder_email": "xxxxx",
        "stakeholder_name": "xxxxx",
        "sequence": "xxxxx",
        "approval": "xxxxx",
        "stakeholder_mobile": "xxxxx",
        "reviewer_ref_id": "xxxxx"
    }
],
"editors_info":
[
    {
        "document_to_be_edited": "xxxxx",
        "stakeholder_email": "xxxxx",
        "stakeholder_name": "xxxxx",
        "stakeholder_mobile": "xxxxx"
    }
]
```

	<pre> }] } </pre>
Response	<pre> { "status": "xxxx", "message": "xxxx", "docket_id": "xxxx", "signer_info": [{ "signer_ref_id": "xxxx", "signer_id": "xxxx", "document_id": "xxxx" }, "custom": [{ "document_id": "xxxxxxxx", "parameter_name": "xxxxxx" }]], "api_response_id": "xxxx", "response_time_stamp": "xxxx" } </pre> <p style="text-align: center;">OR</p> <pre> { "status": "xxxx", "error": "xxxx", "error_code": "xxxx", "response_time_stamp": "xxxx" } </pre> <p style="text-align: center;">OR</p> <p>This response is triggered when the editor flow is active.</p> <p>Success Response:</p> <pre> { "status": "success", "message": "xxxxx", "docket_id": "xxxxx", "signer_info": [{ "signer_ref_id": "xxxxx", "signer_id": "xxxx", "document_id": "xxxx", "reference_doc_id": "xxxx", "invitation_link": "xxxx" }, ...] } </pre>

```
"api_response_id": "xxxxx",
"editor_info":
[{
  "editor_id": "xxxxx",
  "document_id": "xxxxx",
  "reference_doc_id": "xxxxx"
}],
  "response_time_stamp": "xxxxx"
}
```

```
{
```

Failed Response:

```
{
  "status": "failed",
  "error_code": "xxxxx",
  "error": "xxxxx",
  "additional_info": {
    "document_to_be_edited": "xxxxx"
  },
  "api_response_id": "xxxxx",
  "response_time_stamp": "xxxxx"
}
```

This response is triggered when the reviewer flow is active.

Success Response:

```
{
  "status": "success",
  "message": "xxxxx",
  "docket_id": "xxxxx",
  "signer_info": [
    {
      "signer_ref_id": "xxxxx",
      "signer_id": "xxxxx",
      "document_id": "xxxxx",
      "reference_doc_id": "xxxxx",
      "invitation_link": "xxxxx"
    }
  ],
  "api_response_id": "xxxxx",
  "reviewer_info": [
    {
      "reviewer_id": "xxxxx",
      "document_id": "xxxxx",
      "reference_doc_id": "xxxxx"
    }
  ]
}
```

	<pre> "reference_doc_id": "xxxxx", "reviewer_ref_id": "xxxx", "approval": "xxxxx" }], "response_time_stamp": "xxxxx" } Failed Response: { "status": "failed", "error_code": "xxxxx", "error": "xxxxx", "additional_info": {}, "api_response_id": "xxxxx", "response_time_stamp": "xxxxx" } </pre>
Content-Type	json in base64 encoded format
File Size	Max 10Mb

Header

The below-mentioned headers should be set in all the API calls. The values will be provided by SignDesk.

x-parse-rest-api-key

x-parse-application-id

Request Parameters

Parameters	Field Option	Description
reference_id	Mandatory	This is a unique ID assigned to every eSign Requester for each API call.
mode	Optional	<p>This returns the initiation link in the response of the primary API if the necessary configuration is set.</p> <p>This is based on configuration.</p>
enable_organisation_seal	Optional	<p>Enabling this parameter will imprint the respective organization seal on the document in the specified coordinates.</p> <p>Validations:</p> <ul style="list-style-type: none">• Data type: Boolean• Possible Values:<ul style="list-style-type: none">○ true○ false
docket_title	Optional	<p>This refers to the title of the document to be signed.</p> <p>Validations:</p> <ul style="list-style-type: none">• Type: Alphanumeric includes space.• Data type: String• Maximum length: 100
remarks	Optional	Any additional information about the document
enable_email_notification	Optional	This takes a boolean value (either true or false). It specifies whether to enable notification regarding the document completion email functionality.

documents	Mandatory	An array containing document information about the file type, file contents in encoded form, etc.
supporting documents	Conditional Mandatory	<p>These are arrays containing the document information that acts as supporting documents to the documents passed.</p> <p>Note:</p> <ul style="list-style-type: none"> • This field is mandatory if the parameter(flag): supporting_document_required is true. • Supporting documents support the following extensions: <ol style="list-style-type: none"> 1) .ppt 2) .msg 3) .pst 4) .eml 5) .ost 6) .pdf 7) .xls 8) .doc 9) .docx 10) .jpg 11) .png 12) .tiff
reference_doc_id	Mandatory	<p>Unique for every uploaded file. Only one reference_id is generated per file.</p> <p>Validations</p> <ul style="list-style-type: none"> • Max Length: 64 • Type: Alphanumeric • Data type: String
content_type	Mandatory	<p>Refers to the file type that the eSign Requester uploads.</p> <p>Possible values: pdf</p>
content	Mandatory	Base64 encoded file content.
signature_sequence	Mandatory	Type of signatures to be done either sequentially or parallelly by the signer(s).

		<p>There are two signature sequence types:</p> <ul style="list-style-type: none"> ● sequential This defines the signer(s) to sign the document sequentially. ● parallel This enables the signer(s) to sign the document in parallel.
custom	Optional	<p>This parameter can be used to pass a custom parameter for custom mapping of the identification of the transaction.</p> <p>The custom parameter name can be passed in this parameter "parameter_name".</p> <p>Default Validations:</p> <ul style="list-style-type: none"> ● Max Length: 256 ● Type: Alphanumeric ● Data type: String ● Max parameter count: 5
docket_description	Optional	<p>This field gives a description of the document within that docket.</p>
description_in_invitation	Optional	<p>It refers to the description that will be included in the email invite sent to the signer.</p> <p>Possible Boolean Values:</p> <ul style="list-style-type: none"> ● true ● false
final_copy_recipients	Optional	<p>This field refers to comma-separated email IDs that will receive a copy of the signed document.</p> <p>For example: "abc@legaldesk.com,mnop@legaldesk.com"</p>
notification_email_list	Optional	<p>This field refers to comma-separated email IDs that will be notified whenever the document is sent for signature.</p> <p>For example: "abc@legaldesk.com,xyz@legaldesk.com"</p>
final_copy_phone_number_list	Optional	<p>This field refers to comma-separated phone numbers to which the final copy will be triggered</p>

		<p>apart from default stakeholders like initiator, signer, etc.</p> <p>For example: "1234567890;0987654321"</p>
expiry_date	optional	<p>Defines the document's expiry date.</p> <p>Data Type: Date</p> <p>Supported Format: DD-MM-YYYY</p>

Note:

These documents can contain an array of documents. Each document has a unique reference_id.

signers_info		Array containing an array of signer's information about each signer signing the document.
document_to_be_signed	Mandatory	<p>Document to be signed by the signer(s). For every document a signer has to sign, there will be a corresponding reference_doc_id.</p> <p>Validations</p> <ul style="list-style-type: none"> ● Max Length: 64 ● Type: Alphanumeric ● Data type: String
trigger_esign_request	Mandatory	<p>If true, the signer will get a sign request email notification to sign the document.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> ● true ● false
signer_position	Mandatory	Position of the signature on the document.
appearance		<p>It states the appearance of the signature. It should be either a string or an object.</p> <p>If string: (either of)</p> <ol style="list-style-type: none"> 1. bottom-left 2. bottom-right 3. top-left 4. top-right

		<p>If array:</p> <ol style="list-style-type: none"> 1. x1 2. y1 3. x2 4. y2 <p>These refer to the coordinate points of the signature and should contain numeric values.</p> <p>Example:</p> <p>1. Single Appearance + Predefined Position</p> <pre>"signer_position":{ "appearance": "top-left" }</pre> <p>2. Multiple Appearance + Predefined Position</p> <pre>"signer_position":{ "appearance": ["top-left", "top-right"] }</pre> <p>3. Single Appearance + Specific Position</p> <pre>"signer_position":{ "appearance": [{"x1":20, "x2":120, "y1":20, "y2":60}] }</pre> <p>4. Multiple Appearance + Specific Position</p> <pre>"signer_position":{ "appearance": [{"x1":20, "x2":120, "y1":20, "y2":60}, {"x1":20, "x2":120, "y1":20, "y2":60}] }</pre>
signer_ref_id	Mandatory	<p>It is a unique reference ID to identify each signer.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Max Length: 64 • Type: Alphanumeric • Data type: String

signer_email	Mandatory	<p>It is the Email address of the signer(s).</p> <p>Validations:</p> <ul style="list-style-type: none"> • Should contain a valid email • Data type: String • Type: Alphanumeric
signer_name	Mandatory	<p>It is the Name of the signer(s).</p> <p>Validations:</p> <ul style="list-style-type: none"> • Type: Alphanumeric and space • Data type: String • Character length: 60 (Recommended: 50)
sequence	Mandatory	<p>Defines the sequence of the signer(s).</p> <p>Note:</p> <ul style="list-style-type: none"> • If it is a sequential signing, the sequence should be in an incremental order. • Need to define the sequence field with numbers such as 1, 2, 3, etc for each signer. • Unless the first applicant signs the document, none of the other signers can sign the document. <p>Validations</p> <ul style="list-style-type: none"> • Max Length: 2 • Type: Numeric • Data type: Integer
page_number	Mandatory	<p>The page number(s) on which the signer(s) wants to sign the document.</p> <p>Note:</p> <ul style="list-style-type: none"> • The page number should not exceed more than the document's total pages. • For signing all pages, the page number should be "all". • For signing the first page, the page number should be 1. • If the signer signs page numbers 1 and 5 page numbers should be "1, 5". • For signing the last page only the page number should be "last". <p>Validations:</p>

		<ul style="list-style-type: none"> • Type: Numeric or string • Data type: Integer
signature_type	Mandatory	<p>This refers to the method of digital signing preferred.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • Aadhaar • DSC • electronic • VBEC • HSM • CloudSign
esign_type	Conditional mandatory	<p>This refers to the type of Aadhaar eSign preferred. It must be passed if signature_type: Aadhaar</p> <p>Possible values:</p> <ul style="list-style-type: none"> • otp • biometric • both • iris • face authentication <p>Note: iris and face authentication is configured on request.</p>
access_type	Optional	<p>This refers to the type of DSC signing preferred.</p> <p>Note: Mandatory only if signature_type: DSC</p> <p>Possible values:</p> <ul style="list-style-type: none"> • login • otp
captureSignerImage	Optional	<p>This defines whether the signer image should be captured while performing the signature.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Data type: Boolean <p>Possible Values:</p> <ul style="list-style-type: none"> • true • false
signer_mobile	Conditional Mandatory	<p>The mobile number of the signer.</p> <p>Note:</p>

		<ul style="list-style-type: none"> • This is mandatory in the case of an electronic signature. • This is mandatory only if the "signature_type": "electronic" <p>Validations:</p> <ul style="list-style-type: none"> • Type: Numeric • Max Length: 10 • Data type: String
signer_remarks	Optional	<p>This refers to the text content to be printed along with the signature.</p> <p>Note: Mandatory if configured</p>
authentication_mode	Conditional mandatory	<p>This defines the mode of authentication used by the signer in case of electronic signature.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> • email • mobile • none • both
trigger_esign_request_invitation	Optional	<p>This refers to the notification to be sent</p> <p>Possible Values:</p> <ul style="list-style-type: none"> • email • mobile
whatsapp_invitation_mode	Optional	<p>This defines whether the signature invitation should be sent through whatsapp or not.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> • true • false

signer_validation_inputs		An array containing an array of information required for validation of the signer.
year_of_birth	Optional	<p>The year of birth of a signer as per Aadhaar has to be provided here.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Max Length: 4 • Type: Numeric

		<ul style="list-style-type: none"> • Data type: String
gender	Optional	<p>The gender of the signer should be entered here</p> <p>Possible Values:</p> <ul style="list-style-type: none"> • M • F • T <p>Validations:</p> <ul style="list-style-type: none"> • Max Length: 1 • Type: Alphabetic • Data type: String
name_as_per_aadhaar	Optional	<p>The Name of the signer as per Aadhaar should be entered here.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Type: Alphabetic • Data type: String
last_four_digits_of_aadhaar	Optional	<p>Consists of the last four digits of Aadhaar number</p> <p>Validations:</p> <ul style="list-style-type: none"> • Data type: String • Type: Numeric • Min and Max Length: 4

Widget_configuration(Optional)	This defines whether any specific configuration is required to the existing eSign widget.	
widget_title	Optional	<p>This refers to the customization in the existing widget title in the eSign widget.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Type: Alphanumeric • Data type: String
background_color	Optional	<p>This refers to the customization in the existing background color in the eSign widget.</p> <p>The color code required can be passed as a value.</p>
button_color	Optional	<p>This refers to the customization of the existing button color in the eSign widget.</p> <p>The color code required can be passed as a value.</p>

button_font_color	Optional	This refers to the customization of the existing button font color in the eSign widget. The color code required can be passed as a value.
font_family	Optional	This refers to the customization of the existing font family in the eSign widget.
esign_with_otp_button_text	Optional	This refers to the customization in the existing button text for OTP signing in the eSign widget. Validations: <ul style="list-style-type: none"> • Type: Alphanumeric • Data type: String
esign_with_biometric_button_text	Optional	This refers to the customization in the existing button text for biometric signing in the eSign widget. Validations: <ul style="list-style-type: none"> • Type: Alphanumeric • Data type: String
user_instruction	Optional	This refers to the customization of the existing user instructions text in the eSign widget.
remark_field_label	Optional	This refers to the customization of the existing label for the remark field in the eSign widget.
usb_signature_title	Optional	This refers to the customization in the existing title for the USB signature field in the eSign widget.
return_url	Optional	Redirection URL to which the SignDesk application will share the data post the successful signing activity. Note: This data will be sent using the POST method. The data includes the following parameters: { "status": "success", "message": "eSign Successful", "document_id": "xxxx", "signer_id": "xxxx" }

supporting_document_required	Optional	<p>This is a flag passed by the client to make "supporting_documents" - a mandatory parameter.</p> <p>Possible Values:</p> <ol style="list-style-type: none"> 1. true 2. false
callback_file_content	Optional	<p>If the client does not want to receive the file content in the callback. The client passes the value in this flag. (Both individual and completion callback)</p> <p>Possible Values:</p> <ol style="list-style-type: none"> 1. true 2. false
group_reference_id	Optional	<p>This is a unique reference ID assigned to the signer group.</p> <p>Note: The group_reference_id entered in the signers_info and signer_group_info should be the same.</p>
mandatory_signer_count	Optional	This defines the count of the minimum number of signers who should sign the document.
group_name	Optional	This defines the name of the group of signers.
editors_info (Optional)		This refers to an array that contains the editor's information.
document_to_be_edited	Mandatory	<p>This refers to the document that has to be edited.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Max Length: 64 • Type: Alphanumeric • Data type: String
stakeholder_email	Mandatory	<p>This refers to the email address of the stakeholder.</p> <p>Validations:</p>

		<ul style="list-style-type: none"> • Should contain a valid email • Data type: String • Type: Alphanumeric
stakeholder_name	Mandatory	<p>This refers to the name of the stakeholder.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Type: Alphanumeric and space. • Data type: String
stakeholder_mobile	Mandatory	<p>This refers to the mobile number of the stakeholder.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Type: Numeric • Max Length: 10 • Data type: String
reviewers_info (Optional)		<p>This refers to an array that contains the editor's information.</p>
document_to_be_reviewed	Mandatory	<p>This refers to the document that has to be reviewed.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Max Length: 64 • Type: Alphanumeric • Data type: String
reviewer_ref_id	Mandatory	<p>It is a unique reference ID to identify each reviewer.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Max Length: 64 • Type: Alphanumeric • Data type: String
approval	Conditional mandatory	<p>It refers to the type of reviewer.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • Pre • Post <p>Note: Its default value is "Pre".</p> <p>It is based on the configuration.</p>

Response Parameters

Parameters	Description
status	Represents "success" or "failure" of the API.
message	It states whether the response is "received" or "failed".
docket_id	Unique for every set of uploaded document(s). One docket_id is assigned for each set of uploaded document(s).
custom	This parameter can be used to pass a custom parameter for custom mapping of the identification of the transaction. The custom parameter name can be passed in this parameter " parameter_name ".
signer_info	Information about the signer(s).
signer_ref_id	It is a unique reference ID provided by the API caller to identify each signer.
signer_id	A unique ID is assigned to each signer by SignDesk to invoke the widget.
document_id	Unique ID assigned to each document.
response_time_stamp	The time at which the response is received after the API request is submitted.
api_response_id	The unique alphanumeric code allocated at the time of initiation of the Transaction
editors_info	This refers to an array that contains the editor's information. Note: Applicable only in template flow.
editor_id	This refers to the unique ID assigned to each editor.

reference_doc_id	This refers to the unique ID for every uploaded file. Only one reference_id is generated per file.
invitation_link	This refers to the link for the document which has to be edited or signed.
reviewer_id	A unique ID is assigned to each reviewer by SignDesk.
reviewer_info	This refers to an array that contains the reviewer's information.
reviewer_ref_id	It is a unique reference ID provided by the API caller to identify each reviewer.
approval	This refers to the type of reviewer who reviews the document. Note: It can be pre-reviewer or post-reviewer.
error	This is the error that corresponds to the failed API request. Refer to Annexure 1 for errors.
error_code	The error code corresponds to the error generated. Refer to Annexure 1 for errors.

Note:

- If the request made is successful: A unique widget_token will be generated to access the eSign widget window for signing the uploaded document(s).
- If the request made has failed: An error code with an error message will be generated.

JavaScript to call Widget

JavaScript code snippet to invoke the Widget

```
var clientData = {  
    document_id: "document_id",  
    signer_id: "signer_id",  
    docket_id:"docket_id",  
    id:"xxxx",  
    widget_width:"xxxxx",  
    widget_height:"xxxxx",  
    getWidgetToken: function(result){  
        // function will return widget token  
    },  
  
    signatureStatus: function(result){  
        // function will return signature status  
    },  
  
    errorLog: function(result){  
        // function will return error  
    }  
}  
// function call to load Signature  
Panel  
eSignWidgetCall(clientData);
```

getWidgetToken()

This callBack() will return the widget_token. This is unique for every widget call. The widget_token has to be captured within 30 seconds to load the widget. To capture this widget_token, use the widgetToken API.

signatureStatus()

This callback () triggers when the applicant signs the document successfully or when the eSign widget closes. This function returns signature status.

Response: "return_kyc_data" : true

```
{  
    "status": "success",  
    "result": "document signed",  
    "document_id": "xxxxxxx",  
    "signer_id": "xxxxxxx",  
    "user_info":  
    {  
        "name": "xxxxxxx",  
        "email": "xxxxxxx"  
    },  
    "api_id": "xxxxxxx",  
    "kyc_data":  
    {  
        "name_as_per_aadhaar": "xxxxxxx ",  
        "last_digits_of_aadhaar": "xxxxxx",  
        "postal_code": "xxxxxxx",  
        "gender": "x",  
        "year_of_birth": "xxxx"  
    }  
}
```

Response : "return_kyc_data" : false

```
{  
    "status": "success",  
    "result": "document signed",  
    "document_id": "xxxxxx",  
    "signer_id": "xxxxxx"  
}  
  
or  
  
{  
    "status": "failed",  
    "result": "document not signed",  
    "document_id": "xxxx",  
    "signer_id": "xxxx"  
}  
  
or
```

```
{
  status: "rejected",
  error: "document not signed",
  error_code: "es_001",
  document_id: "xxxx",
  signer_id:"xxxx",
  validation_error:
  {
    "gender":
    {
      "message": "Gender did not match the details as per Aadhaar records",
      "value_passed": "xxxx",
      "value_as_per_aadhaar": "xxxx"
    },
    "Year_of_birth":
    {
      "message": "Year of Birth did not match the details as per Aadhaar records",
      "value_passed": "xxxx",
      "value_as_per_aadhaar": "xxxx"
    },
    "Name_as_per_aadhaar":
    {
      "message": "The name did not match the details as per Aadhaar records",
      "value_passed": "xxxx",
      "value_as_per_aadhaar": "xxxx",
      "percentage_match": "xxx"
    }
  }
}
```

errorLog Function	This callBack() returns an error if any.
widget_width	Optional To define the width of the eSign widget.
widget_height	Optional To define the height of the eSign widget.
id	x-parse-application-id that is present in the header must be passed as the value.

Note: **widget_height** and **widget_width** are defined in terms of screen occupancy percentage.
Example : 60%

Sample Code to integrate Widget

```
<body>

    <button id="Click" onclick="mycall()">add</button>
    <script type="text/javascript" src="URL"></script>
    <script type="text/javascript">

        function mycall() {

            var clientData = {
                document_id: "document_id",
                signer_id: "signer_id",
                docket_id:"docket_id",
                id:"xxxxx",
                widget_width:"xxxxx",
                widget_height:"xxxxx",
                getWidgetToken:function(result){
                    // function will return widget token
                },
                signatureStatus:function(result){
                    // function will return signature status
                },
                errorLog:function(result){
                    // function will return error
                }
            }
            // function call to load Signature Panel
            esignWidgetCall(clientData);

        </script>
    </body>
```

widgetToken API

Widget Token Request API	
Feature	To capture the widget token
HTTP Type	POST
Header	x-parse-rest-api-key x-parse-application-id
Request	{ "widget_token":"xxxx", "time_stamp":"xxxx", "docket_id":"xxxx" }
Response	{ "status":"xxxx", "widget_token":"xxxx", "response_time_stamp":"xxxx", "docket_id":"xxxx", "error":"NA", "error_code":"NA" } or { "status":"xxxx", "widget_token":"xxxx", "response_time_stamp":"xxxx", "docket_id":"xxxx", "error":"xxxx", "error_code":"xxxx" }
Content-Type	Application/json

Header

The below-mentioned headers should be set in all the API calls. The values will be provided by SignDesk.

x-parse-rest-api-key

x-parse-application-id

Request Parameters

Parameter	Field Option	Description
widget_token		This is generated by the JavaScript getWidgetToken() function.
time_stamp	Optional	Request API timestamp. This is in "YYYY-MM-DDThh:mm:ss" format (derived from ISO 8601). Time zone should not be specified and is automatically defaulted to IST (UTC +5.30).
docket_id	Optional	Unique for every set of uploaded document(s). One docket_id is assigned for each set of uploaded document(s).

Response Parameters

Parameter	Description
widget_token	The widget_token that is sent as the request parameter to the eSignWidgetCall() JavaScript function is fetched as the response parameter.
status	<ul style="list-style-type: none">It represents whether the API is working or not i.e. the success or failure of the API.If the status is "success", then the eSign Widget window opens for the applicant to sign the document.
response_time_stamp	Response API timestamp. This is in "YYYY-MM-DDThh:mm:ss" format (derived from ISO 8601). Time zone should not be specified and is automatically defaulted to IST (UTC +5.30).
error	This is the error that corresponds to the failed API request.
error_code	The error code corresponds to the error mentioned above.

getDocketInfo API

FILE DOWNLOAD API	
Feature	To get the eSigned document using document_id
HTTP Type	POST
Header	x-parse-rest-api-key x-parse-application-id
Request	{ "docket_id": "xxxx", "document_id": "xxxx" }
Response	{ "status": "success", "error": "NA", "error_code": "NA", "message": "docket_info", "docket_info": [{ "content": "<base-64-content>", "content_type": "pdf", "document_id": "xxxx" }] } Note: In a case where the document is deleted from our server the below response is triggered. { "docket_id": "xxxx", "document_id": "xxxx", "error": "xxxx", "error_code": "xxxx" }
Content-Type	Application/json in base64 encoded format

Header

The below-mentioned headers should be set in all the API calls. The values will be provided by SignDesk.

x-parse-rest-api-key

x-parse-application-id

Request Parameters

Parameter	Field Option	Description
docket_id	Conditional mandatory	Unique for every set of uploaded document(s). One docket_id is assigned for each set of uploaded document(s).
document_id	Optional	Unique ID assigned to each uploaded document.

Response Parameters

Parameter	Description
docket_id	Unique for every set of uploaded document(s). One docket_id is assigned for each set of uploaded document(s).
document_id	Unique ID assigned to each document.
content	Contains the content of the uploaded file which is in base64 encoded format.
error	This is the error that corresponds to the failed API request. Refer Annexure 1 for errors.
error_code	The error code corresponds to the error mentioned above. Refer Annexure 1 for errors.

getSignatureStatus API

GET SIGNATURE STATUS API	
Feature	To get the signature status of the signers, of the provided document_id
HTTP Type	POST
Header	x-parse-rest-api-key x-parse-application-id
Request	{ "document_id": "xxxx" }
Response	{ "status": "success", "signer_info": [{"signer_id": "xxxxx", "signature_sequence": "xxxxx", "signed_at": "xxxxx", "status": "xxxxx"}] } OR { "status": "failed", "error": "xxxx", "error_code": "xxxx" }
Content-Type	Application/json in base64 encoded format

Header

The below-mentioned headers should be set in all the API calls. The values will be provided by SignDesk.

x-parse-rest-api-key

x-parse-application-id

Request Parameters

Parameter	Field Option	Description
document_id	Mandatory	Unique ID assigned to each uploaded document.

Response Parameters

Parameter	Description
status	Represents "success" or "failure" of the API.
signer_info	Represents the signer's status and information.
signer_id	This unique ID is assigned to each signer by SignDesk to invoke the widget.
signature_sequence	The sequence of the signers for the document if signature_sequence is sequential.
signed_at	This refers to the timestamp indicating when each signer completed their signature.
status	It represents whether the particular signer has signed the document or not.
error	This is the error that corresponds to the failed API request. Refer Annexure 1 for errors.
error_code	The error code corresponds to the error mentioned above. Refer Annexure 1 for errors.

updateEsignSignerValidationInputs API

updateEsignSignerValidationInputs API	
Feature	Updates input parameters in Signer Validation
HTTP Type	POST
Header	x-parse-rest-api-key x-parse-application-id
Request	<pre>{ "signer_id": "xxxx", "document_id": "xxxx", "signer_validation_inputs": { "year_of_birth": "xxxx", "gender": "xx", "name_as_per_aadhaar": "xxxx", "last_four_digits_of_aadhaar": "xxxx", "state": "xxxxx", "postal_code": "xxxxx" }, "signer_details": { "signer_email": "xxxx", "signer_mobile": "xxxx", "signer_name": "xxxx", "signer_signature_type": "" "access_type": "LOGIN" "payer_email_address": "xxxxx", "payer_mobile_number": "xxxxx", } }</pre>
Response	<p>Type1: Success</p> <pre>{ "status": "success", "error": "NA", "error_code": "NA" }</pre> <p>Type2: Failed</p>

	<pre>{ "status": "failed", "message": "xxxx", "error": "xxxx", "error_code": "xxxx" }</pre>
Content-Type	Application/json

Header

The below-mentioned headers should be set in all the API calls. The values will be provided by SignDesk.

x-parse-rest-api-key

x-parse-application-id

Request Parameters

Parameter	Field Option	Description
signer_id	Mandatory	This is a unique ID assigned to every signer.
document_id	Mandatory	Unique ID assigned to each uploaded document.
signer_validation_inputs	Mandatory	An array containing an array of information required for validation of the signer.
year_of_birth	Mandatory	<p>The year of birth of a signer as per Aadhaar has to be provided here.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Max Length: 4 • Type: Numeric • Data type: String
gender	Mandatory	<p>The gender of the signer should be entered here</p> <p>Possible Values:</p> <ul style="list-style-type: none"> • M • F • T

		<p>Validations:</p> <ul style="list-style-type: none"> • Max Length: 1 • Type: Alphabetic • Data type: String
name_as_per_aadhaar	Mandatory	<p>The Name of the signer as per Aadhaar should be entered here.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Type: Alphabetic • Data type: String
last_four_digits_of_aadhaar	Mandatory	<p>Consists of the last four digits of the Aadhaar number.</p> <p>Validations:</p> <ul style="list-style-type: none"> • Data type: String • Min and Max Length: 4
state	Optional	This indicates the state to which the signer belongs.
postal_code	Optional	This refers to the postal code associated with the signer's address.
signer_details	Mandatory	This refers to the details of the signer of the document.
signer_email	Mandatory	<p>It is the Email address of the signer(s).</p> <p>Validations:</p> <ul style="list-style-type: none"> • Should contain a dot (.) and @ • Data type: String • Type: Alphanumeric
signer_name	Mandatory	<p>It is the Name of the signer(s).</p> <p>Validations:</p> <ul style="list-style-type: none"> • Type: Alphanumeric. Shall include characters, special characters, and space. • Data type: String
signer_signature_type	Mandatory	<p>This refers to the signature type chosen by the signer.</p> <p>Possible Values:</p> <ul style="list-style-type: none"> • DSC • Aadhaar

		<ul style="list-style-type: none"> • Electronic
access_type	Conditional Mandatory	<p>This refers to the type of DSC signing preferred.</p> <p>Note: Mandatory only if signature_type: DSC</p> <p>Possible values:</p> <ul style="list-style-type: none"> • login • otp
payer_email_address	Optional	This refers to the payer's email address.
payer_mobile_number	Optional	This refers to the payer's mobile number.

Response Parameters

Parameter	Description
status	Represents the "success" or "failure" of the API.
message	Represents the message corresponding to the status of the API.
error	This is the error that corresponds to the failed API request. Refer to Annexure 1 for errors.
error_code	The error code corresponds to the error mentioned above. Refer to Annexure 1 for errors.

Resend Invitation API

Resend Invitation API	
Feature	In case the initial signing invitation is not sent, triggering this API resends the invitation
HTTP Type	POST
Header	x-parse-rest-api-key x-parse-application-id
Request	{ "document_id": "xxxx", "docket_id": "xxxx", "stakeholder_id": "xxxx", "reference_id": "{\$timestamp}" }
Response	{ "status": "Success", "error_code": "NA", "message": "Resend Invitation sent successfully", "reference_id": "xxxx", "resendInvitationLink": "xxxx" } OR { "status": "Failed", "error_code": "xxxx", "message": "xxxx", "reference_id": "xxxx" }
Content-Type	Application/json in base64 encoded format

Header

The below-mentioned headers should be set in all the API calls. The values will be provided by SignDesk.

x-parse-rest-api-key

x-parse-application-id

Request Parameters

Parameter	Field Option	Description
stakeholder_id	Mandatory	This is a unique ID assigned to every stakeholder.
reference_id	Mandatory	This is a unique ID assigned to every eSign Requester for each API call.
docket_id	Mandatory	Unique for every set of uploaded document(s). One docket_id is assigned for each set of uploaded document(s).
document_id	Mandatory	Unique ID assigned to each uploaded document.

Response Parameters

Parameter	Description
status	Represents the "success" or "failure" of the API.
message	Defines the status of the Invitation.
reference_id	This is a unique ID assigned to every eSign Requester for each API call.
resendInvitationLink	This parameter refers to the signature invitation link sent to the signers who have not received the initial invitation. (Based on the configuration)
error	This is the error that corresponds to the failed API request. Refer to Annexure 1 for errors.
error_code	The error code corresponds to the error mentioned above. Refer to Annexure 1 for errors.

updateEsignDocumentInputs API

updateEsignDocumentInputs API	
Feature	
HTTP Type	POST
Header	x-parse-rest-api-key x-parse-application-id
Request	<pre>{ "document_id": "xxxx", "reference_id" : "{\$timestamp}", "document_details": { "expiry_date": "xxxx", "notification_email_list": "email address1,email address2", "final_copy_recipients": "email address1,email address2" } }</pre>
Response	<p>Success Response</p> <pre>{ "status": "success", "error_code": "NA", "message": "xxxx", "response_time_stamp": "xxxx", "api_response_id": "xxxx", "reference_id": "xxxx" }</pre> <p>Failure Response</p> <pre>{ "status": "failed", "error_code": "xxxx", "message": "xxxx", "reference_id": "xxxx", "response_time_stamp": "xxxx" }</pre>
Content-Type	Application/json

Annexure 1: Error Codes

Error Code	Status	Error Message
es_101	failed	The required <field_name> was not found.
es_102	failed	The <field_name> value used in the request is incorrect.
es_103	failed	The required <field_name> is not unique.
es_104	failed	The <field_name> does not match the <field_name>.
es_105	failed	The document is already signed.
es_106	failed	Please check your internet connection or try again later.
es_107	failed	User pressed the back button to cancel the process.
es_109	failed	As signature_type value is electronic the esign_type is not applicable.
es_110	failed	As signature_type value is electronic the access_type is not applicable.
es_111	failed	Duplicate <field_name> entries are found in the request.
es_118	failed	The document you are trying to access has been deleted from the system as per the deletion policy.
es_127	failed	The required parameter was not found in the request.
es_128	failed	The document content used in the request is invalid.
es_130	failed	The validation has failed due to the mismatch between Aadhaar data and the Signer input data.(Used in Redirection URL API doc)
es_131	failed	Entered Aadhaar number is not linked with email id and mobile number.
es_132	failed	Something went wrong. Please try again.
es_133	failed	The user has cancelled the process by clicking on the Cancel button.

es_134	failed	Invalid widget token.
es_137	failed	<p>The uploaded PDF has some unneeded signatures. Please upload a PDF without any signatures to proceed further.</p> <p>Password protected and editable PDF files cannot be uploaded.</p>
es_144	failed	The required signer_remarks was not found in the request. (Applicable if it is configured)
es_145	failed	<p>The value of "signer_remarks" is exceeding the maximum character length 15.</p> <p>(Applicable if it is configured)</p>
es_149	failed	The notification_email_list value used in the request is incorrect.
es_151	failed	The document_id present in the request does not match with the x-parse-application-id.
es_152	failed	The signer_authentication_mode used in the request is incorrect.
es_153	failed	The file size of the document_ids exceeds the configured limit.
es_154	failed	The file size of the merged document exceeds the configured limit.
es_155	failed	The Aadhaar Signature was declined either due to network interruption or a delay in the server response.
es_156	failed	The document you are trying to sign is corrupted.
es_157	failed	The signature could not be affixed since the coordinate values used in the request are overlapping.
es_158	failed	The signature could not be affixed to the page since the page_number and appearance parameters are not of the same array length.

es_166	failed	The description_in_invitation value used in the request is incorrect.
es_167	failed	As per the organizational policies, you are not authorized to access this API endpoint from your network location.
es_186	failed	The required <code>docket_id</code> was not found in the request
es_187	failed	The required <code>stakeholder_id</code> was not found in the request
es_188	failed	Resend invitation can be sent only when the status of the document is Invited.
es_189		
es_190	failed	The document expiry date cannot be the current date(T). It should begin from the T+1 date.
es_191	failed	Failed to fetch the document for the <code>document_id</code> "id" passed in the request.
es_192	failed	Failed to fetch the document for the <code>docket_id</code> "id" passed in the request.
es_193	failed	Resend Invitation mail was not triggered due to a technical glitch.
es_194	failed	No stakeholder was found in the system for the <code>stakeholder_id</code> passed in the request.
es_195	failed	The <code>signer_mobile</code> value used in the request already exists with a different signer.
es_196	failed	Failed to delete the Signer.
es_197	failed	The document has been already deleted.
es_198	failed	The required <code>signer_id</code> was not found in the request.
es_199	failed	The required <code>x-parse-application-id</code> was not found in the request.
es_200	failed	The required <code>x-parse-rest-api-key</code> was not found in the request.

es_201	failed	Unexpected error while updating the signer.
es_202	failed	The required signers_info was not found in the request.
es_203	failed	The number of signers has exceeded the Maximum Limit.
es_204	failed	The content was not found in the request.
es_205	failed	The signer details passed in the request are invalid.
es_206	failed	The Signer details do not match with the Certificate details.
es_207	failed	An unexpected error was encountered while validating the signature.
es_267	failed	The group_reference_id passed in the request already exists.
es_268	failed	The group_reference_id passed in the request is not unique.
es_269	failed	The required group_reference_id was not present in the signer_group_info.
es_270	failed	The mandatory_count of signers passed should be less than the total number of group signers.
es_271	failed	The required group_name was not found in the request.
es_272	failed	The value of group_name has exceeded the configured character_length.
es_273	failed	The group_signature feature is not configured for this organization.
es_274	failed	The signature_type:DSC does not support the group_signature feature.
es_275	failed	The document supports only one group of signers.
es_401	failed	The value of payer_email_address passed in the request is invalid.
es_402	failed	The value of payer_mobile_number passed in the request is invalid.
es_403	failed	The payment has already been made. Hence the payer details cannot be modified.
es_404	failed	The payer_details could not be updated due to an unknown exception.
es_405	failed	The pricing has not been configured during onboarding process. Hence the user cannot be added. Please contact SignDesk.

es_406	failed	The initiator and the payer have the same email address in the request. They must be different.
es_407	failed	The initiator and the payer have the same mobile number in the request. They must be different.

Annexure 2: Integration Checklist

S.No.	Integration Requirements
1.	The Aadhaar Number should not be stored.
2.	Electronic Copy of eSigned documents should be made available to the eSign User.
3.	The eSign should be affixed only by the Aadhaar Holder and not by any other party on behalf of the Aadhaar Holder.
4.	The OTP should be entered only by the Aadhaar Number Holder.
5.	The document uploaded should be a permissible document under the purview of the IT Act, 2000.
6.	There can be no changes to software or code as provided by Desk Nine Pvt Ltd, which has been built as per CCA & UIDAI compliance guidelines.
7.	The Stamp duty should be calculated as per the applicable laws of each respective State. Applicable only if eStamp services are consumed.
8.	One OTP can be used for the signature of one single document.