JioSign API Specification

V1.6

Jio Platforms Ltd., Reliance Corporate Park
Thane Belapur Road, Ghansoli-400 701

DISCLAIMER

The information in this document is subject to change without notice and should not be construed as commitment by Jio Platforms Ltd. assumes no responsibility or makes no warranties for any errors that may appear in this document and disclaims any implied warranty of merchantability or fitness for a particular purpose

1 Change History

Ver	Date	Author	Remarks
Draft	10-08-2021	JioSign Team	Draft Document
0.1	18-08-2021	JioSign Team	1. Gateway API Details 2. Error Codes updated in separate excel file. 3. Added endpoints for api access.
0.2	19-08-2021	JioSign Team	1. All the Integration Url should have 8443 port in the base url. 2. Correction in http method and minor input and output changes of APIs.
0.3	24-08-2021	JioSign Team	1. Added steps needed to decrypt the session token downloaded from JioSign Portal.
0.4	08-09-2021	JioSign Team	1. Added Mac Id is header of request.
0.5	30-09-2021	JioSign Team	1. Changed the jiosign portal url 2. Modified new UI images.
0.6	21-03-2022	JioSign Team	1. Added new field in Audit Trail API output. a. trails.identifier 2. Removed following attributes Audit Trail API output. a. trails.userPhone b. trails.userEmail
0.6.1	17-05-2022	JioSign Team	1. Added new API for download Audit Trail PDF. 2. Modified output of Get Audit Trail API to return OTP entered by user in meta-attribute and few additional attributes.
0.7	01-06-2022	JioSign Team	1. Modified Document – Create api a. Added new “doc-owner” attribute in request body. 2. Modified Document – Save Data api a. Notification array will be mandatory in input b. Added new notification types – 6/7/13 c. Added enable attribute for notification types 3. Added Document Sign/Decline workflow api. 4. Modified Document – Get Data a. Added creatorId attribute in the response. 5. Added Get Participant Status api to provide list of participants with signing status in a document. 6. Added Get Card Details api to provide card details. 7. Added Delete Participant api. 8. Added Document finalization workflow API. 9. Added appendix section for certificate and card coordinates references.
0.8	13-09-2022	JioSign Team	Releasing Version 1.1 for RP APIs. Added below features. 1) Document owner can upload supporting documents 2) User can upload signature/initial images and provide name to be used in document signing. Below is the Change history for Minor Version upgrade: 1. Modified Document – Save Data api a. Settings array will be optional in input b. Added new settings type – 22 c. Added enable attribute for settings type 2. Added Document – Save Supporting document api 3. Modified Document – Get Data api a. Added docType in response. 4. Updated Sign/Decline api workflow to upload signature/initial image and name.
0.9	16-12-2022	JioSign Team	Releasing below features in Version 1.1:

			1. Modified Document – Save Data api a. grpSettings array will be optional in input b. Added new grpSettings type – 24 c. Added enable attribute for grpSettings type 2. Modified Document – Get participants status API a. Added randomUuid in response. 3. Added Document – Save callback url api
1.0	05-01-2022	JioSign Team	Releasing below features in API Version 1.1: 1. Updated the algorithm for token decryption. 2. Modified Document – Save Callback URL a. Added retry mechanism 3. Modified Document – Save Data api a. grpSettings enable value changed to 0/1 4. Modified Document - Save CallBack Url a. Updated the endpoint from /document/callback to /callback
1.1	09-02-2023	JioSign Team	Releasing below features in API Version 1.1: 1. Modified Document – Save Data api a. Added “Document Signer” as a new signature method with assurance level as 6 b. Added a card Type=10, this card type is to be selected for the signature method as “Document Signer”. 2. Added Bulk Sign Workflow in section 5.2.6. 3. Modified Document - Save CallBack Url api a. Updated the endpoint from /document/callback to /callback b. Updated the request body to support the callback url registration for the bulk sign process.
1.2	09-03-2023	JioSign Team	Releasing below features in API Version 1.1: 1. Modified Document – Save Data api a. Added “signOrderType” for Grouped Signing Order. b. Removed card limit of 20 for ‘e-Signature’ signature method.
1.3	12-04-2023	JioSign Team	Releasing below features in API Version 1.1: 1. Modifies Document – Bulk Sign Status API a. Added “status” filter as non-mandatory field. 2. Changed tab name from “API token” to “Integerations” in profile page.
1.4	27-07-2023	JioSign Team	Releasing below features in API Version 1.1: Increased the email length limit from 50 chars to 320 chars.
1.5	07-11-2023	JioSign Team	Releasing below features in API version 1.1: 1. Bulk DSC Sign Workflow API’s 2. Multicard support for DSC Signing
1.6	05-01-2024	JioSign Team	Releasing below features in API version 1.1: 1. Modified Document - Save Data API: a. Added attributes ‘autoLocateCards’, ‘participantTag’ & ‘cTag’ in the request payload to support automatic card placement feature for signature cards based on text-tags

Table of Contents

2 API Endpoints

Environment	TYPE	Endpoints
Integration	API	**rpapi-sit.jiosign.jio.com:8443**
Production/Live	API	Will be shared once integration testing is done by RP (Relying Partners a.k.a API Consuming Systems).
JioSign Portal Integration	Portal	**https://demo.jiosign.jiolabs.com/**
JioSign Portal Live	Portal	Will be shared once integration testing is done by RP (Relying Partners).

3 API Constants

Category	Code	Meaning	Description
Participant Access Level	1	Signer	Participant who can sign the document. Will receive All Signing Collected, Automated Reminder notifications if opted in.
	2	Viewer	Participant can only view the document. Will receive final signing notifications if opted in.
Participant Identifier Type	1	Email	Type of contact is email for given participant while document data save.
	2	Mobile	Type of contact is phone for given participant while document data save.
Card on Page (for signature card placement)	1	Only on one page of document.	Only on the page number which is passed in the request. Only on one page.
	2	On All the page of document.	Card will be applied to all the page on same coordinates. Coordinates can’t be different on different page.
Card Type	1	Initial	Initial card, user’s initial from JioSign system will be picked and applied on this card location.
	2	Signature	Signature card, user’s signature from JioSign system will be picked and applied on this card location.
	10	Document Signer	Document Signer card, If the signature method is “Document Signer” then only this card has to be added.
Notification Type	1	Document Finalization Notification	Once all the participant signs this notification is initiated. **It can be set for Viewer, Signer access types.**
	2	Each Participant Signing Notification.	Document owner can set this notification to receive notification for each participant signing. **It can be set for Document Owner.**

			Document owner is the user who is owning the document. It can't be changed.
	5	Automated Signing Reminder Notification	Based on defined frequency automated notification reminder is sent to participants. **It can be set for Signer access types.** **Frequency Unit and Frequency Value is mandatory for this notification type.**
	6	Document Decline Notification	JioSign Default notification, which is sent to all participants, if any signatories has declined to sign the document. Now RPs can enable/disable this notification by passing input in the request. **Mandatory for Document Owner, Document Creator, and All Participants.**
	7	Invitation Notification	JioSign Default notification, which is sent to all the participants, who has been invited to sign/view the document. Now RPs can enable/disable this notification by passing input in the request. **Mandatory for All Participants.**
	13	Document Upload Notification	JioSign Default notification, which is sent to the document owner/creator when document is uploaded in JioSign system. Now RPs can enable/disable this notification by passing input in the request. **Mandatory for Document Owner and Creator.**
Frequency Units	1	Days	Daily automated notification
Assurance Level	2	Virtual Signature	This is an alternate signature method for E-Signature. This assurance level can't be combined with other two methods defined below. If virtual signature assurance level set in request than all the participants should have same method set otherwise API returns error.
	3	DSC Token	Multiple cards per participant is allowed. Combination of 3 and 4 is allowed in request for participants.
	4	E-Signature	Multiple cards per participant is allowed. Combination of 3 and 4 is allowed in request for participants
	6	Document Signer	Multiple cards per participant is allowed. Combination of 2, 3, 4 and 6 is allowed in request for participants. This signature method can only be used by Document owner or Document creator to sign the documents.

			This signature method can be used to sign the documents by generating access-token from JioSign Portal.
Document Envelop (Group) Status	1	Active	Document Upload is done and waiting for participants signatures.
	2	Delete	Document is marked as deleted and won't be accessible to anyone.
	3	InProgress	Document upload is in progress. Document will not be visible to any participant other than document owner and document creator.
	4	Complete	All participants have signed the document and document is locked for further signing or modifications.
Audit Events EventId	1	Created Document or Group	Events captured for group create.
	2	Document Download	Once participant downloads the file.
	3	Signed Document	Captured on signature events by participants.
	5	Added as Signatory	On addition of participants for given document.
	9	Deleted participant	On deletion of participants for given document.
	10	Added as Viewer	On addition of participants as viewer for given document.
	11	Decline to Sign	Once participant has declined to sign.
	14	Viewed Document	On participant viewing document in portal.
Document User Type	1	Document Owner	Owner for document.
	2	Document Participant	Participant of document.
	3	Document Creator	Document Creator who uploads the document.
Participants Status for given Document	1	Ready for signing	Signing Invitation is shared with Participant. They can sign document.
	2	Signing Complete	Participant has completed document signing.
	3	Deleted by Participant	Participant has deleted document from his Login, document will be visible to another participant.
	4	Draft	Document is in draft status; Document Creator has not completed all the steps.
	5	Sign Waiting	Document Owner/Creator has set signing order, and user signing turn has not come. This status will change to ready to sign once previous participant signs the document in signing order.
	6	Document Declined	Participant has declined document.
Signing API Authentication Type	2	Email	If participant is signing using email.
	3	Mobile	If participant is signing using phone.
Signing Action values			
	101	Signing	Signing the document
	102	Decline	Decline to sign the document.
Settings Type	22	Show Supporting Documents	This is participant level settings to show them supporting documents in JioSign portal. Document Owner/Creator can disable this setting. By default, it is enabled.

Group Settings Type	24	Show Signing Comments	This is group level settings to show the signing comments option to participants. Document Owner/Creator can disable this setting. By default, it is enabled.
Group Settings Enable	0	Disable	Value 0 is for disabling the signing Comment
	1	Enable	Value 1 is for enabling the signing Comment
signOrderType	2	Group Signing order and all signatories must sign	DO/DC can assign the same signing order to multiple participants. For the document to get completed, all the signatories must sign. For e.g. Participant 1,2 are assigned to signing order 1, Participant 3,4,5 to signing order 2 and so on. All these participants must sign the document.
	3	Grouped Signing order, one signatory from each group must sign	DO/DC can assign the same signing order to multiple participants. For the document to get completed, only one participant from each signing order must sign. For e.g. Participant 1,2 are assigned to signing order 1, if participant 2 has signed the document first then participant 1 will be become viewer.
authType	2	Email	Email Id
	3	Mobile Number	Participant Mobile Number
Source	3	DSC Sign	Sign Using DSC
Action	1	Sign Document	Sign Document
	3	Hash Generation	Hash generation request
	4	Decline Document	Decline the document
autoLocateCards **(*Note : If parameter is not found in request by default it will be 0)**	1	Enable	Document will be searched for text tags to auto-place cards.
	0	Disable	Document will not be searched for text tags to auto-place cards **(i.e. normal flow will happen.)**
cTag **(*Note : If parameter is not found in request by default it will be 0)**	1	Enable	Card that needs to be auto-located based upon the text tags added in the document. **(*Note: If autoLocateCards = 1 then cTag should be 1.)**
	0	Disable	All the required card details are provided and no automatic card placement is required. **(*Note: The card structure will remain as a normal card parameter)**

4 Gateway Authorization API

4.1 API Gateway Session

API provides authorization token, which is required for opening the channel to initiate System to System call.

This is the first step to be executed before starting consumption of JioSign API described latter in document. Token is valid for stipulated time if it expires then RP system needs to generate new authorization token.

4.1.1.1 API Endpoints

Http Method	URL	Note
POST	https://{FQDN}/token	Refer {FQDN} in EndPoint section above.

4.1.1.2 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	grant_type	String	18	Y	client_credentials	Require to generate token based on client and secret.

4.1.1.3 Output

Success Output:

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	access_token	String	36	Y	e9b3f25d-cae7-3d98-bd38-1f15e792f990	Access Token to be used for JioSign API calls
2.	scope	String	28	Y	am_application_scope default	Access Scope.
3.	token_type	String	6	Y	Bearer	Token type
4.	expires_in	Number	NA	Y	3600	Value is in Second.

Sample Error Outputs

```
Token API Error Structure:
{
    "error_description": "Client Authentication failed.”,
    "error": "invalid_client"
}
API Invocation Error Structure:

<ams:fault xmlns:ams="http://wso2.org/apimanager/security">
    <ams:code>Code</ams:code>
    <ams:message>Message</ams:message>
    <ams:description>Description</ams:description>
</ams:fault>
```

4.1.1.4 API Request Headers

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	Content-Type	String	NA	Y	application/x-www-form-urlencoded	Content Type of api , will always be application/x-www-form-urlencoded
2.	Content-Length	Integer		Y	29	Content length value should always be 29.

3.	Authorization	String		Y	Basic <value>	<Base64-encoded-client_key:client_secret> Client key and Secret will shared during the system onboarding process.

5 JioSign API

JioSign RP (Relying Partners) API is having two main categories 1) Session Management 2) Document Management. These APIs are described below in separate sections.

5.1 Session Management API's

These are sets of API which is provides features like validating user and generating valid token for JioSign API calls.

Note: Authorized user needs to login to JioSign portal with their Business Account registered email-id/phone and download encrypted token from their profile section. Refer Appendix B for certificate creation.

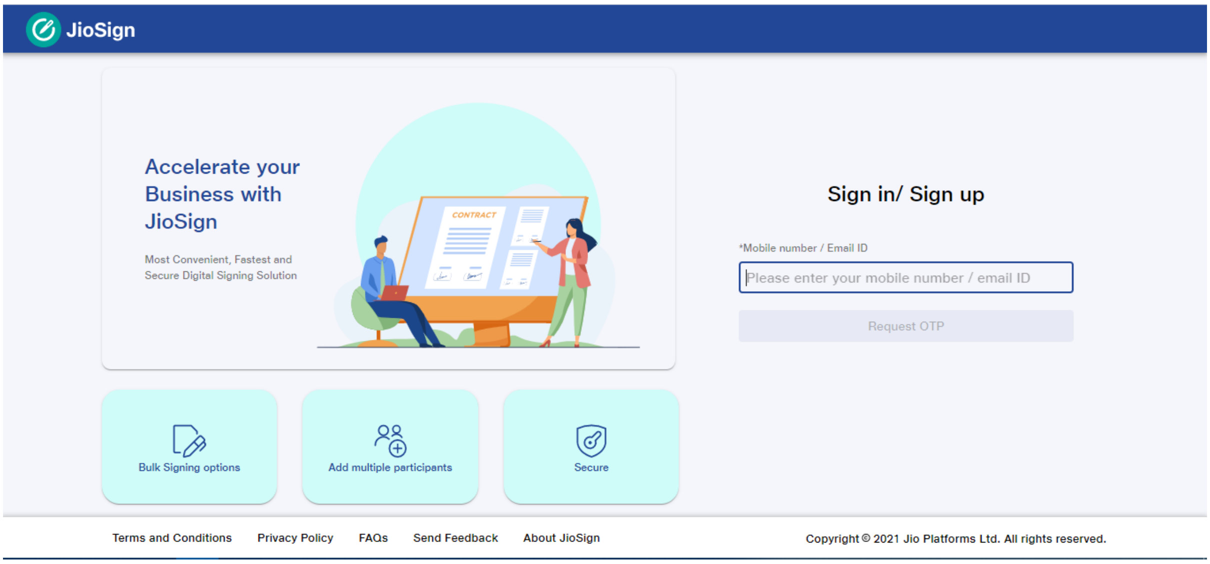
5.1.1 Session – Getting API Token

Authorized user needs to login to JioSign portal (EndPoint) with their Business Account registered email-id/phone and download token file from their profile.

Note: Downloaded token data is encrypted with public key which was shared during Business Account on boarding. RPs would need to use respective private key pair to decrypt the data.

Below steps will guide RPs on how to get the token from JioSign portal and retrieve the tokens from encrypted data.

1) Before login, please ensure that account which is getting used for login is already linked to Business Account and API feature is enabled. Otherwise, you will not find below mentioned functions in portal. If API feature is not visible, then please contact JioSign Team.
2) Login to JioSign Portal.



JioSign
Accelerate your Business with JioSign
Most Convenient, Fastest and Secure Digital Signing Solution
CONTRACT
Bulk Signing options
Add multiple participants
Secure
Terms and Conditions
Privacy Policy
FAQs
Send Feedback
About JioSign
Copyright © 2021 Jio Platforms Ltd. All rights reserved.

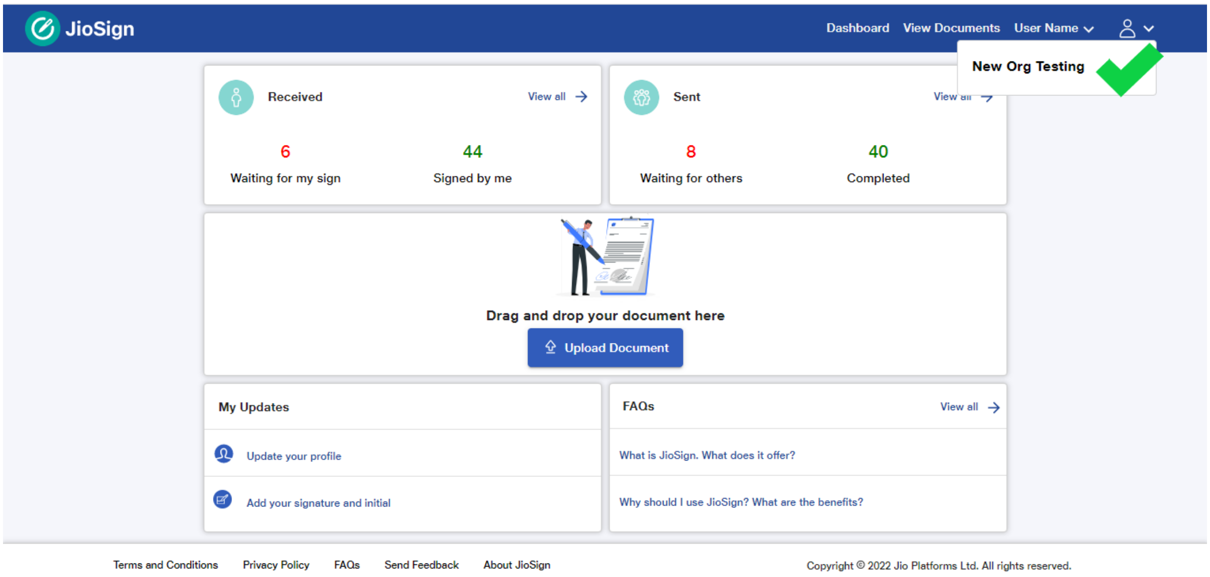
Sign in/ Sign up

*Mobile number / Email ID

Please enter your mobile number / email ID

Request OTP

3) After login to JioSign portal, from the top right menu switch the profile to Associated Business Account.

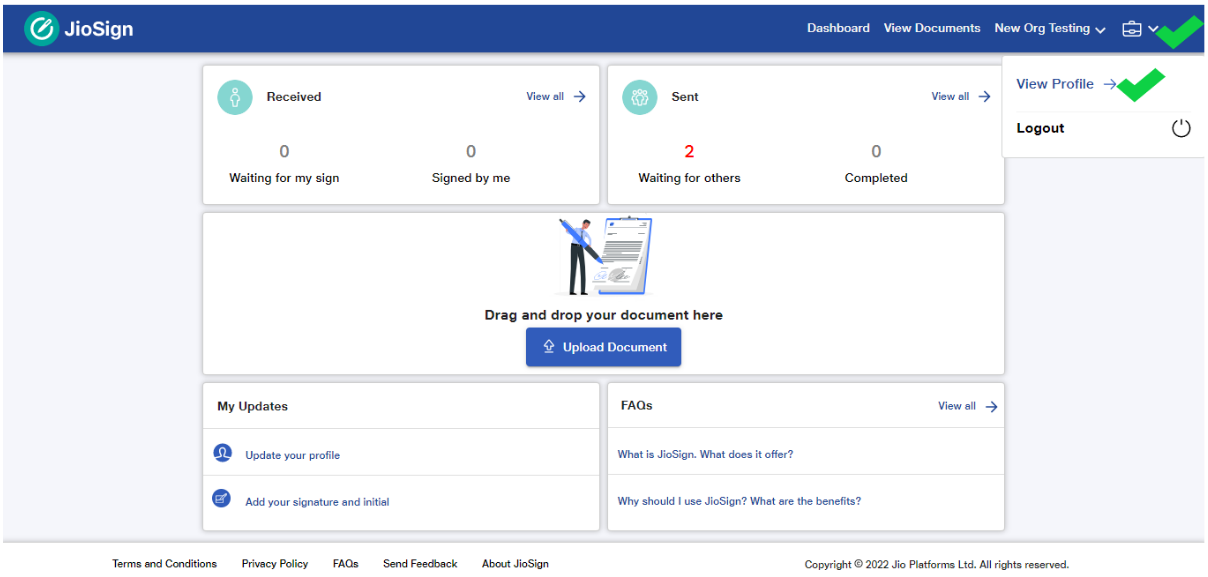


JioSign
Dashboard
View Documents
User Name
New Org Testing
Drag and drop your document here
Upload Document
My Updates
Update your profile
Add your signature and initial
FAQs
View all
What is JioSign. What does it offer?
Why should I use JioSign? What are the benefits?
Terms and Conditions
Privacy Policy
FAQs
Send Feedback
About JioSign
Copyright © 2022 Jio Platforms Ltd. All rights reserved.

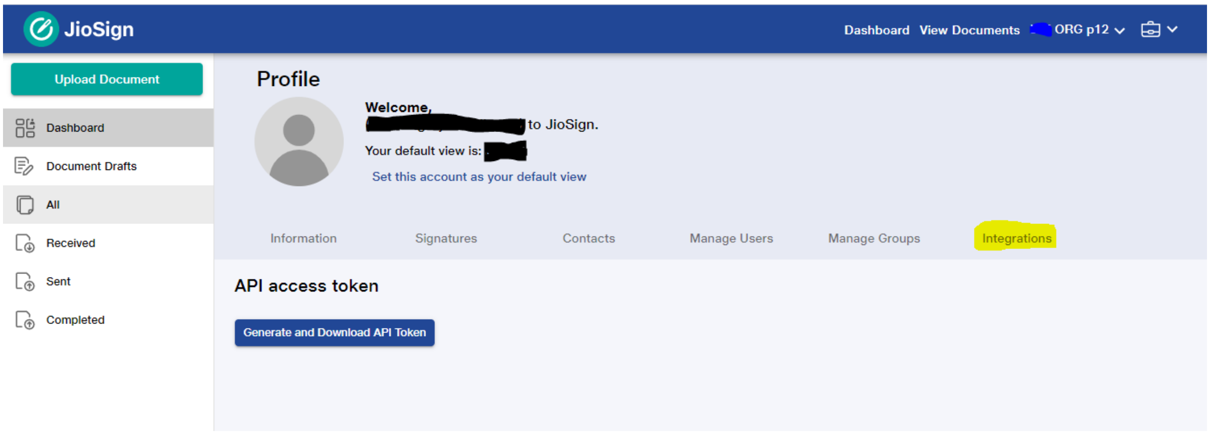
Received	
6	44
Waiting for my sign	Signed by me

Sent	
8	40
Waiting for others	Completed

4) Click on **Profile** link on menu. Check below screen with green tick marks.



5) In Profile page, click on the “**Integrations**” tab. You can download token using “Generate and Download API Token” button. It will download file containing token data. The format of file is json.



6) Json file details.
   a. Attributes and Description
      i. Key – This attribute has key, using which “data” attribute value is encrypted.
         “Key” value is encrypted with **public key** which was shared during initial onboarding. You would need private key pair to decrypt this value.
      ii. Data – This attribute has encrypted data which contains API Session Token (token) and Refresh Token (rtoken). It is encrypted using decrypted “Key” value specified above.

7) Follow below steps to get the data from the json file.
   a. Step1: Getting the key for data decryption
      i. Base64 Decode the value present in “key” attribute.
      ii. Use Private Key pair of Public key which was shared as part of initial Business Account onboarding.

iii. Decrypt the value inside “key” attribute using the private key using "RSA/ECB/PKCS1Padding".

iv. Output of step (iii) will be decrypted key, which needs to be used for decrypting “data” attribute value. Follow below Step2 for decrypting data.

b. Step2: Getting the token and rtoken value

i. Base64 Decode the value present in “data” attribute.

Decrypt the value inside “data” attribute using decrypted key received in Step1 → (iv) using “AES/ECB/PKCS5Padding”.

c. Attributes and Description

i. **token**: JioSign API Session token which needs to be used in consuming API’s e.g. Create Document Save Document Data etc.
Validity is 24 hours. RP needs to refresh the token before it expires.

ii. **rtoken**: Refresh Token to be used for extending session token validity. Session token can only be extended before their expiry, please ensure that you call it before Session token expires.
Validity is 24 hours. RP needs to use valid refresh token to get new token.

5.1.2 API Request Headers

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	Ip-Address	String	11	Y	192.222.112.122	IP Address of the user system
2.	Content-Type	String	NA	Y	application/json	Content Type of api , will be application/json
3.	Txn	String	16	Y	1123-3223-4333-2344	RPs needs to generate transaction id send in the request. Uniqueness of txn should be RPs system, JioSign does not have any validation of uniqueness.
4.	Token	String	NA	Y		Valid token received during session creation.
5.	Rtoken	String	NA	CM		**Only required for refresh token api.**
6.	Authorization	String	NA	Y	Bearer <value>	Token created during Gateway Authorization. Check Section Gateway Authorization Section

5.1.3 Session - Refresh Token

Provides new JWT token based on issued refresh token and valid token.

5.1.3.1 Details

- API will take the refresh token (rtoken) and token (token) in header.
- Both Token should not have expired, otherwise API will return error.

- If all condition is satisfied, then API will issue following in response.
  - token – New Session token, to be used in API call
  - rtoken – New Refresh token to be used in next refresh token call.

5.1.3.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/session/v1.1/token	Refer {FQDN} in EndPoint section above.	application/json

5.1.3.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	NA					

5.1.3.4 Output

Output1:

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	Errcode	String	11	Y	JDSH-LO-100	Error code of request.
2.	Message	String	20	Y	message	Message describing the reason of failure.

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	Token	String	NA	Y		JWT token will be for API calls.
2.	Rtoken	String	NA	Y		Refresh token to be used for refreshing new JWT.

5.1.4 Session – Logout

5.1.4.1 Details

- Used for session invalidation in JioSign System.
- Session invalidation can be only done for token which are not expired.
- Token is passed as part of request header.
- If expired token is sent API will return error back.

5.1.4.2 API Endpoints

Http Method	URL	Note	Output Format

DELETE	https://{FQDN}/session/v1.1/logout	Refer {FQDN} in EndPoint section above.	application/json

5.1.4.3 Input

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	NA					

5.1.4.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	message	Message from API

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	50	Y	message	Message from API

5.2 Document Management API's

Below are the set of APIs which can be used for document related operation, like create document, create document data, delete, get, get original file, get signed file, document signing, bulk sign, Get Participants, Get Cards.

5.2.1 API Request Headers

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	Ip-Address	String	11	Y	192.222.112.122	IP Address of the user system
2.	Content-Type	String	NA	Y	application/json	Content Type of api, will be • application/json • multipart/form-data
3.	Token	String	NA	Y		JWT token created and shared in JioSign Portal or refreshed using Session Management API.
4.	Txn	String	16	Y	1123-3223-4333-2344	RPs needs to generate transaction id send in the request. Uniqueness of txn should be RPs system, JioSign does not have any validation of uniqueness.
5.	Mac-Id	String		N	01:23:45:67:89:AB	Mac Id of the calling system.

6.	Authorization	String		Y	Bearer <value>	Token created during Gateway Authorization. Check Section Gateway Authorization Section
7.	action-token	String	NA	CM		**Mandatory for following steps in document signing API.** **1. Document Signing Initiate Status** **2. Document Signing Verify OTP** **3. Document Signing** **4. Document Signing Status** **4. Document Finalization Status**
8.	access-token	String	NA	CM		**Mandatory for following steps in Bulk signing API.** **1. Bulk Sign Initiate** **2. Bulk Sign Status**

5.2.2 Document – Create Envelop

API takes care of creation of document envelop. Document Envelop is top level entity which will wrap all the attribute like file, participants (signatories), signing cards for participants, and participant's notifications.

API returns envelop unique id in response called groupId, which will be used by RPs to query data back from JioSign System.

5.2.2.1 Details

- A valid session needs to be there for the API consumption.
- API will create document envelop in JioSign system.
- User who is creating the document can add the document owner, who will be owning that document and managing it, in the request in case if the document owner is different than document creator. If the user who is uploading the document is going to manage the document, then there is no need to set the document owner in the request.
- File, Participants, Cards needs to be created against this document envelop Id (groupId).
- Integrating system needs to store this id for further operation on Document like getting document/deleting/Signing document etc.

5.2.2.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document	Refer {FQDN} in EndPoint section above.	application/json

5.2.2.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	name	String	200	Y	RentalAgg	Name of the group to be created. Allowed Characters:

						- _ . & a-z A-Z 0-9 (), space
2.	doc-owner	String	320	N	abcd@ril.com/9193923113	Document owner's email or phone number. If doc-owner is not provided in the request, then document creator will be the document owner.

5.2.2.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	message	Message from API

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	50	Y	Created successfully.	Message from API
2.	groupId	String	50	Y	1234	Unique groupId/document envelop Id created.
3.	txn	String	16	N	1123-3223-4333-2344	Unique transaction id for each request. It is not mandatory, it may be present and may not be present

`

5.2.3 Document - Save Data

API is used for saving document related additional data in JioSign system.

5.2.3.1 Details

- API requires valid session.
- Session creation can be created in JioSign portal.
- Response will return the groupId (document envelop id), after saving all the required data which is sent in API input.
- Content Type: **multipart/form-data**

5.2.3.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/data	Refer {FQDN} in EndPoint section above.	application/json

5.2.3.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	group	Object				
2.	group.groupId	String	50	Y	12312-1233-123-12	Id of the document envelop/groupId created with Document Create API.
3.	group.message	String	500	N	Please sign or text.	This message will be part of signing invitation mail to all participants. Allowed Characters: a-z A-Z 0-9 . , ; : ! ? ( ) { } @ * / # % ^ = | ~ _
4.	file	MultipartFile	30 MB	Y	NA	Single file selected by user, Maximum limit of 30 MB.
5.	group.grpSettings	Object[]		N		Group level settings
6.	grpSettings.type	int		N	24	Defines what type of settings owner/creator wants to set for group. Check API Constant section for code description. Settings Type=24 is to “Show Signing Comments”. By default, this setting is enabled, i.e., participants of that group will get an option to provide their signing comments while signing the document. Document Owner or Document Creator can disable this setting by sending enable=0 for that group.
7.	grpSettings.enable	int		N	0/1	Set enable to 1 if the grpSettings type needs to be enabled for the group, else set it to 0. By default, this grpSettings type is enabled.
8.	group.signOrderType	int	2	CM	2/3	Defines what type of signing order owner/creator wants to set for the participants. signOrderType is required only if grouped signing order is defined. If there is no signing order or the signing order is strict, i.e., only single participant can be assigned against a signing order and all the participants must sign with their

						respective sign order, then there is no need to pass signOrderType. Check API Constant section for code description for grouped signing order.
9.	group.participants	Object[]		Y		Participant Object which holds all the participants which has been added by document owner.
10.	group.autoLocateCards	int		N	0/1	Should be passed as 1 if user wants to do auto card placement. Check API Constant section for information related to allowed values.
11.	participants.assuranceLevel	String	50	Y	2 3,4 3 4 6 2,3,4,6	Selected signing method for participant by document owner. Check API Constant section for code description. Important Note: 2 – Alternate Signature method for E-Signature. 3 – Multiple cards per participant is allowed. 4 – Multiple cards per participant is allowed. 6 -- Multiple cards per participant is allowed. This signature method can only be used by Document owner or Document creator to sign the documents.
12.	participants.idValue	String	320	Y	abcd@ril.com/9193923113	Participant's email or phone number.
13.	participants.idType	Int	2	Y	1/2	Type of value selected in "idValue". Check API Constant section for code description.
14.	participants.access	Int	2	Y	1 / 2	Participant access level in document signing process. If participant is going to sign or viewer. Check API Constant section for code description.
15.	participants .signOrder	Int	3	Y	1,2,3,4 -1- no sign order	Signing order for a given participant. **If no sign order, then send -1.** **For Grouped signing Order, check signOrderType.**

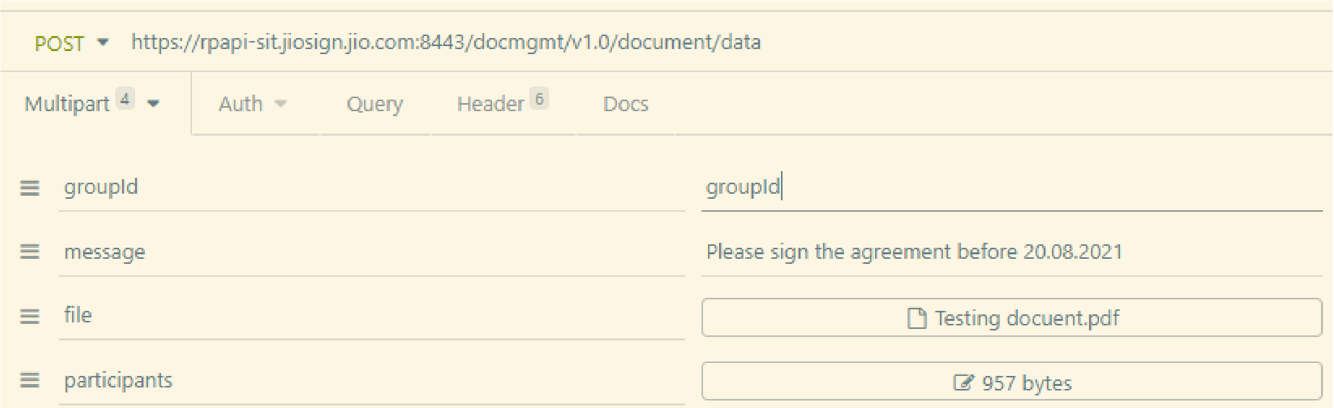
16.	participants.notifications	Object[]		Y		Notification Setting for participant.
17.	participants.participantTag	Integer		CM	1,2,3,4,5…	When autoLocateCards is enabled and there are multiple participants this field is mandatory. It should be unique to differentiate participants and auto locate cards to respective participants based upon the text-tags in document. Ex. **{{Signature-P<participantTag>}}** In case of single participants it is not mandatory.
18.	notifications.frequency	Int	1	CM	1	Reminder Frequency, value unit is captured in freq_unit, whether the value is 1 (Day) **Mandatory if notification type is 5.**
19.	notifications.freq_unit	Int	1	CM	1/2/3/4	Frequency Unit defines the frequency value given in point#12, Check API Constant section for details. **Mandatory if notification type is 5**.
20.	notifications.notification_type	Int	2	Y	1/2/5/6/7/13	Defines what type of notification owner wants to set for participant. Check API Constant section for details. RPs can now control default notifications which JioSign system sends, if RPs want to manage notification on their own then they can disable these default notifications. Default Notification Type 6, 7 and 13 are mandatory for following cases: **6 - Mandatory for Document Owner, Document Creator, and All Participants.** **7 - Mandatory for All Participants.** **13 – Mandatory for Document Owner and Document Creator.** **Allowed Notification role wise**: **Signer Role**: • Automated Signing reminder. • All Signing Reminder **Viewer Role**: • All Signing Reminder **Owner Role**: (Additional) • Each Sign Reminders

21.	notications.enable	Int	1	Y	0/1	Set enable to 1 for the notification type which has to be sent. If enable is set to 0 then notification will not be sent from JioSign.
22.	participants.settings	Object[]		N		Participant level settings.
23.	settings.type	Int	2	N	22	Defines what type of settings owner wants to set for participant. Check API Constant section for code description. Settings Type=22 is to “Show Supporting Documents”. By default, this setting is enabled, i.e., participants will see the supporting documents in JS portal uploaded by owner. Document Owner or Document Creator can disable this setting by sending enable=0 for that participant.
24.	settings.enable	Int	1	CM	0/1	Set enable to 1 if the setting type needs to be enabled for the participant else set it to 0.
25.	participants.cards	Object[]		Y		Stores information related to signature cards and their coordinates for each participant.
26.	cards.totalPage	Int		CM	1	Total number of pages in the document.
27.	cards.cardX	Float		CM	123.2	X coordinate of the signature card placement. Refer Appendix A
28.	cards.cardY	Float		CM	12.1	Y coordinate of the signature card placement. Refer Appendix A
29.	cards.cardH	Float		CM	23	Height of signature card.
30.	cards.cardW	Float		CM	2	Width of signature card
31.	cards.unit	String	50	CM	px	Unit of cardX, cardY, cardH, cardW fields. **Default is px.**
32.	cards.cardColor	String	50	Y	#FRSRWY	Colour of the card which needs to be used for given user. **Hexadecimal code.**
33.	cards.cardType	Int	2	CM	1 / 2/ 10	Type of card which is placed in UI. Card can be of these type. Check API Constant section for details.
34.	cards.cardPageNo	Int		CM	1/2/3/4	Page Number on which card has been placed.
35.	cards.cardOnPage	Int		CM	1/2	If card need to be added on all page or only on the value

						mentioned on cardPageNo attribute. Check API Constant for more details.

36.	cards.cTag	int		CM	0/1	Check API Constant for information related to allowed values.
						For a card that needs to be automatically placed on document cTag should be 1, if cTag is 1 then that card cannot be placed on every page and cardOnPage value will be made 1 from backend.
						
						Sample card for cTag = 1
						`{`
						`  "cTag": 1,`
						`  "cardColor": "#ff0000"`
						`},`
						Sample card for cTag = 0
						`{`
						`  "cTag": 0,`
						`  "cardType": 1,`
						`  "cardX": 288.5,`
						`  "cardY": 497,`
						`  "cardH": 44,`
						`  "cardW": 105,`
						`  "totalPage": 2,`
						`  "cardPageNo": 1,`
						`  "cardOnPage": 1,`
						`  "cardColor": "#f2d130"`
						`}`

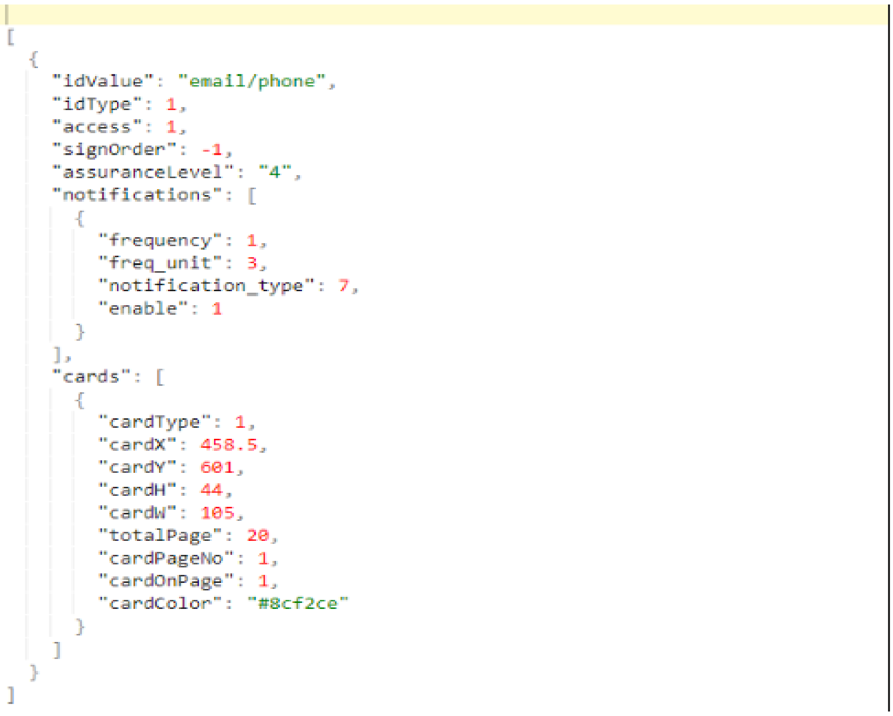
Sample Input from API Client tool:



POST https://rpapi-sit.jiosign.jio.com:8443/docmgmt/v1.0/document/data

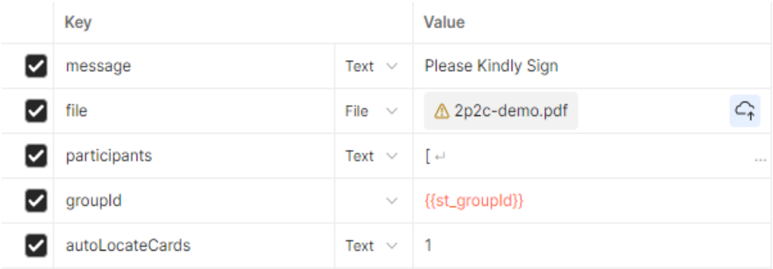
groupId	groupId
message	Please sign the agreement before 20.08.2021
file	Testing docuent.pdf
participants	957 bytes

“participants” section data



```
[
  {
    "idValue": "email/phone",
    "idType": 1,
    "access": 1,
    "signOrder": -1,
    "assuranceLevel": "4",
    "notifications": [
      {
        "frequency": 1,
        "freq_unit": 3,
        "notification_type": 7,
        "enable": 1
      }
    ],
    "cards": [
      {
        "cardType": 1,
        "cardX": 458.5,
        "cardY": 601,
        "cardH": 44,
        "cardW": 105,
        "totalPage": 20,
        "cardPageNo": 1,
        "cardOnPage": 1,
        "cardColor": "#8cf2ce"
      }
    ]
  }
]
```

Sample Input for automatic card placement:



	Key		Value
☑	message	Text	Please Kindly Sign
☑	file	File	⚠ 2p2c-demo.pdf
☑	participants	Text	[↵
☑	groupId		{{st_groupId}}
☑	autoLocateCards	Text	1

Participants section :

```
[
 {
  "idValue": "email/phone",
  "idType": 1,
  "access": 1,
  "signOrder": -1,
  "participantTag": 1,
  "assuranceLevel": "4",
  "notifications": [
   {
    "frequency": 1,
    "freq_unit": 3,
    "notification_type": 1,
    "enable": 1
   }
  ],
  "cards": [
   {
    "cTag": 1,
    "cardColor": "#f2d130"
   }
```

```
        ]
      }
    ]
```

5.2.3.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request.
2.	message	String	50	Y	message	Message describing the reason of failure.

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	50	Y	User created successfully.	Message describing the reason of failure.
2.	groupId	String	50	Y	1234	document envelop/groupId for which document data has been added.
3.	txn	String	16	N	1123-3223-4333-2344	Unique transaction id for each request. It is not mandatory, it may be present and may not be present

5.2.4 Document - Save Supporting Document

API is used for saving supporting document as reference to the signing document in JioSign system.

5.2.4.1 Details

- API requires valid session.
- Session creation can be created in JioSign portal.
- API will take groupId (document envelop id) in the request and return the documentId for the supporting document.
- Below are the validations on supporting documents:
  1. Number of supporting documents allowed: 20
  2. Maximum size limit on supporting document file: 30 MB
  3. Maximum size limit for supporting documents in a group: 30 MB
- Content Type: **multipart/form-data**

5.2.4.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/document	Refer {FQDN} in EndPoint section above.	application/json

5.2.4.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	Y	12312-1233-123-12	Id of the document envelop/groupId created with Document Create API.
2.	docType	Int	1	Y	2	Send docType=2 for supporting documents
3	file	MultipartFile	30 MB	Y	NA	Single file selected by user, Maximum limit of 30 MB.

5.2.4.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request.
2.	message	String	50	Y	message	Message describing the reason of failure.

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	50	Y	User created successfully.	Success message from API
2.	documentId	String	50	Y	12123-1233-123-15	Supporting document Id.

5.2.5 Document – Sign/Decline Workflow

Document Signing/Decline workflow steps is described below.

Workflow Steps	API Name	Description
Step 1	Sign/Decline Initiate	API initiates Signing, initiate OTP to participant who is signing. Click on link to jump to the section.
Steps2	Sign/Decline Initiate Status	API tells the status of Initiate Call, action-token from Step1 should be passed.
Step 3	Sign/Decline VerifyOTP	API validates entered OTP, action-token from Step1 should be passed.
Step 4	Sign/Decline	API initiates signing/declining, action-token from Step3 should be passed.
Step 5	Sign/Decline Status	API tells the status of document Sign call, action-token from previous Step3 should be passed.

Important Points:

- A valid RP session needs to be there for the API consumption.
- Workflow sequence should be followed otherwise API will fail.
- action-token in the response plays critical role, any new action-token received in the response of steps, should be passed in next step request header under action-token header. Check section 5.2.1 for API request headers.
- action-token is only valid for specific document Sign or Decline request.

- RPs will have to register the callback url for the groupid (envelop id) so that they can be notified on the registered callback url after any participant has signed or declined the document. After RPs receive the notification from JioSign system, they can check the Document – Status api and Document - Get Participants Status api to know the document and participant status. Refer section 5.2.18.
- If the RPs have not registered the callback url then they will have to continue polling Document – Status api and Document - Get Participants Status api to know the document and participant status.

5.2.5.1 Document – Sign/Decline Initiate

5.2.5.1.1 Details

- API validates the input data and sends One Time Password (OTP) to identifier (Email/Phone) received in input.
- API will validate below fields:
  1. Terms and Condition field should be "Y" in the input request.
  2. Identifier passed in the input request should be valid.
  3. Authentication Type passed in the input request should be valid.
  4. Group Id passed in the input request should be valid.
  5. Action passed in the input request should be valid.
  6. If action is to sign a document, then assuranceLevel should be valid.
- New action-token will be returned in the response.

5.2.5.1.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/sign/initiate	Refer {FQDN} in EndPoint section above.	application/json

5.2.5.1.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1	identifier	String	320	M	email/phone	Email or phone of user who is signing or declining the document. User will be receiving OTP on same identifier.
2	authType	Int	2	M	2/3	Authentication Type defines kind of identifier, participant is using to validate his authenticity. Check API Constant for Authentication Type.
3	groupId	String	50	M	1234-2091-2912-2938	document envelop id/groupId which participant wants to sign or decline.
4	message	String	200	N	Message	Message during signing or decline document. This message will be sent to all signers if notification is enabled.
5	tandc	String	2	M	Y/N	User Agreement for Jiosign terms and condition.

						RPs needs to ask user select terms and condition and if required route user to JioSign portal for terms and condition.
6	assuranceLevel	Int	2	CM	2/4	Assurance Level of Signing. assuranceLevel is mandatory for action 101. For action value 102 it is not used. Check API Constant for details.
7	action	Int	3	M	101/102	Action which the user wants to take on the document. Check API Constant for details.

5.2.5.1.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1	status	String	20	Y	INITIATED	Status of Document Sign/Decline Initiated Status. Check details section of API for different status.
2	action-token	String	NA	Y	Action Token	Temporary action token issued for the user who is signing/declining the document.

5.2.5.2 Document – Sign/Decline Initiate Status

5.2.5.2.1 Details

- API provides the status of Previous Workflow Step, which is Signing Initiate.
- RPs needs to send action-token received in Step1.
- Responds back with 200 OK, with following **status** value.

Status Value	Description
INITIATED	• Request is initiated to sanitize the input data. • RPs needs to continue Polling to know status of API.
RECEIVED	• Data sanitization has been completed. • Process is preparing to notify user with OTP. • RPs needs to continue Polling.
RUNNING	• OTP is sent to user. • RPs needs to stop Polling.

	• RPs needs to ask user to enter OTP received on email/phone and initiate next step.
COMPLETE	• Process is completed with error. • RPs need not poll after this status it gets in the response. • There will be "**message**" attribute which will have "**error or error message**" value. If the value is not "OK", it means that API is not successful. • **error**: Means OTP generation has failed. RPs can show error message to the user and ask them to reinitiate the OTP process. • **RPs needs to start the process from the beginning if this status is received during polling API call**.

5.2.5.2.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/sign/initiate/status	Refer {FQDN} in EndPoint section above.	application/json

5.2.5.2.3 Input

No parameter required.

5.2.5.2.4 Output

Output1: Error, in case API runs into technical error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	status	String	20	Y	INITIATED RECEIVED RUNNING	Status of Document Sign/Decline Initiated Status. Check details section of API for different status.

5.2.5.3 Document – Sign/Decline VerifyOTP

5.2.5.3.1 Details

- RPs needs to send received OTP in request body "code" field only in the first request.
- RPs need to send the action-token in request header which have been received in the response from Step1.
- API will return error response if it is submitted more than once.
- RPs need to continue polling till the **COMPLETE** status is received.

- After the OTP Validation, API will validate signature/initial images and name for the user who is signing. These details are required for signing the document.
- API will return the message if these details are missing from user's profile in case of signing request. RPs need to show this message to the user and ask user to provide these details in Step4 to initiate signing.
- If user's signature/initial images and name are already present in the JioSign system, API will return the message to initiate signing at Step4.
- New action-token will be returned in the response. RPs need to send this token in Step4.
- Responds back with 200 OK, with following **status** value, description of various status in the response:

Status Value	Description
RUNNING	• OTP validation is initiated in JioSign System. • RPs needs to continue Polling to know status.
COMPLETE	• OTP Validation is complete. • RPs can start executing the Next Workflow Step4.

5.2.5.3.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/sign/verifyotp	Refer {FQDN} in EndPoint section above.	application/json

5.2.5.3.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	code	String	15	M	1228	OTP received by signing participant on the email/phone.

5.2.5.3.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	status	String	20	Y	RUNNING COMPLETE	Status of OTP Validation. Check details section of API for different status.
2.	message	String	50	Y		Message from API

3.	action-token	String	NA	Y	Action Token	Temporary action token issued for the user who is signing/declining the document.

5.2.5.4 Document – Sign/Decline

5.2.5.4.1 Details

- API will be initiating signing/declining process.
- RPs need to send the action-token in request header which have been received in the response from Step3.
- API will validate name and signature/initial images of the user who is signing the document. These details are mandatory for signing the document.
- API will overwrite the existing name and signature/initial images if provided in the request. Otherwise it will use the existing details present in the JS system to sign the document.
- For document decline request, API will not validate these details and initiate the decline request.
- Content Type: **multipart/form-data**
- Below are validations for signature and initial images:
    1. Maximum Limit for Image size = 30 KB
    2. Maximum limit for Image Dimensions: 300X100
    3. Supported Image formats: PNG, JPG, JPEG
- Responds back with 200 OK, with following **status** value, description of various status in the response:

Status Value	Description
RUNNING	• Signing/Decline request is processing in JioSign System. • RPs can start executing the Next Workflow Step5.

5.2.5.4.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/sign	Refer {FQDN} in EndPoint section above.	application/json

5.2.5.4.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	Name	String	100	CM		Name of the user to be used for document signing. Mandatory for signing request if name does not exist in the JioSign System.
2.	signatureImg	PNG, JPEG, JPG	30KB	CM		Signature image of the user to be used for document signing.

						Mandatory for signing request if Signature image does not exist in the JioSign System. Maximum Limit for Image size is 30 KB Maximum Limit for Image Width is 300 pixels Maximum Limit for Image Height is 100 pixels
3.	initialImg	Supported Image formats are PNG, JPEG, JPG	30KB	CM		Initial Image of the user to be used for document signing. Mandatory for signing request if Initial image does not exist in the JioSign System. Maximum Limit for Image size is 30 KB Maximum Limit for Image Width is 300 pixels Maximum Limit for Image Height is 100 pixels

5.2.5.4.4 Output

Output1: Error, in case API runs into technical error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	Errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	Message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	Status	String	20	Y	RUNNING	Status of Document Sign/Decline Initiated Status. Check details section of API for different status.

5.2.5.5 Document – Sign/Decline Status

5.2.5.5.1 Details

- API provides the status of Previous Workflow Step4.
- RPs needs to continue polling till the **COMPLETE** status is received.
- RPs needs to send action-token received in Step3.
- Responds back with 200 OK, with following **status** value.

Status Value	Description
SIGNING/DECLINING	• Document Signing/Declining is initiated. • RPs needs to continue Polling to know status.

SIGNED	• Document is signed, but still process is not complete, final steps are being done by API. • RPs needs to continue Polling to know final status.
COMPLETE	• Process is completed. • RPs stops poll after this status it gets in the response. • Check "**message**" attribute, which will have "**ok**" or "**error or error message**" value. If the value is not "ok", it means that API is not successful. • **ok**: Means signing/decline is Success. • **error**: Means signing/decline is failed. RPs needs to show error message to the user and ask them to reinitiate the process. • **RPs needs to start the process from the beginning if "Error" status is received during polling API call**.

5.2.5.5.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/sign/status	Refer {FQDN} in EndPoint section above.	application/json

5.2.5.5.3 Input

No parameter required.

5.2.5.5.4 Output

Output1: Error, in case API runs into technical error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
3.	Errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
4.	Message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
2.	Message	String	20	Y	Error OK	Message from API. If status value is COMPLETE and message is OK, means signing/declining is successful. Error means process failed.
3.	signatureId	String	50	N	aa0b10f-63a6-4ec7-b2f9-8d96848e96e	Signature Id will be sent in response only when the action is for document signing and status value COMPLETE and signing is successful. When the action is for document decline then signatureId will not be sent.

4.	Status	String	20	Y	SIGNING/DECLINING COMPLETE	Status of Signing/Decline Request. Check the details section of API for more information.

5.2.6 Document – Bulk Sign Workflow

Document Bulk Signing workflow steps is described below.

Workflow Steps	API Name	Description
Step 1	Bulk Sign Initiate	API initiates Bulk Signing, access-token generated from JioSign Portal should be passed.
Step 2	Bulk Sign Status	API tells the status of Initiate Call, access-token generated from JioSign Portal should be passed.

Important Points:

- A valid RP session needs to be there for the API consumption.
- Workflow sequence should be followed otherwise API will fail.
- access-token in the request plays critical role. RPs need to generate access-token from JioSign portal (profile section) and send the access-token in step1 and step2 request headers.
- Bulk Sign Workflow APIs can only be used to sign the documents with signature methods as “Document Signer” or “E-Signature” or “Virtual Signature”.
- If the access-token is generated for the “Document Signer” signature method, then send that access-token in the request header of the API when signing documents. Check section 5.2.1 for API request headers.
- If the access-token is generated for the “E-Signature” or “Virtual Signature” signature method, then send that access-token in the request header of the API when signing documents.
- access-token is valid only for that RP account.
- RPs will have to register the callback url for the suiteid, which is returned in response of step1, so that they can be notified on the registered callback url after the bulk sign process is completed. After RPs receive the notification from JioSign system, they can check the bulk sign status api (step2) to know the final status. Refer section 5.2.18.
- In case RPs have not registered the callback url for the suite id, then they will have to poll the bulk sign status api (step2) till they get the final status.

5.2.6.1 Document – Bulk Sign Initiate

5.2.6.1.1 Details

- API validates the input data and returns success message along with suiteId in response.
- API will validate below fields:
    1. Terms and Condition field should be “Y” in the input request.
    2. Group Ids passed in the input request should be valid.
    3. Action passed in the input request should be valid.
    4. AssuranceLevel should be 2 (Virtual Signature), 4 (E-Signature) or 6 (Document Signer).
    5. Access-token in the request headers should be valid and active.
    6. Maximum Number of documents (envelop / group ids) that can be sent in bulk sign request is 500.

5.2.6.1.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/sign/bulk	Refer {FQDN} in EndPoint section above.	application/json

5.2.6.1.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1	grpIds	String[]	500	Y		List of GroupIds. Group id is the id of the document (envelop id). Maximum number of documents to be sent for signing is 500
2	Message	String	200	N	Message	Message during signing or decline document. This message will be sent to all signers if notification is enabled.
3	Tandc	String	2	M	Y/N	User Agreement for Jiosign terms and condition. RPs needs to ask user select terms and condition and if required route user to JioSign portal for terms and condition.
4	assuranceLevel	Int	2	M	2/4/6	Assurance Level of Signing. Check API Constant for details.
5	Action	Int	3	M	101	Action which the user wants to take on the document. Only allowed action is Signing. Check API Constant for details.

5.2.6.1.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1	Errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2	Message	String	50	Y	message	Message from API
3	grpIds	String[]	500	N		List of invalid group ids (envelop ids) sent in the bulk sign request.

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1	Message	String	50	Y	Success	Success message from API

2	suiteId	String	50	Y	1234	Unique suiteId for bulk sign. This suiteid can be sent in step2 to get the signing status.

5.2.6.2 Document – Bulk Sign Status

5.2.6.2.1 Details

- API validates the suiteId sent in the request and returns the status of the suiteId in response.
- RPs needs to continue polling till the **COMPLETE** status is received.
- RPs needs to send access-token which is generated from JioSign Portal (profile section) in request headers.
- Responds back with 200 OK, with following **status** value.

Status Value	Description
INITIATED	• Bulk Document Signing is initiated. • RPs needs to continue Polling to know status.
COMPLETE	• Bulk Document Signing process is complete. RPs can stop polling after receiving this status.

5.2.6.2.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/sign/bulk/status	Refer {FQDN} in EndPoint section above.	application/json

5.2.6.2.3 Input

#	Request Parameter	Type	Length	Mandatory	Sample	Description
1	suiteId	String	50	Y	12312-1233-123-12	Unique Id received in response of step1.
2	groupId	String	50	N	12312-1233-123-12	Unique Id of the document (envelop Id). In case RPs need to know the status of specific document then they can send group Id along with suite Id.
3	Status	String		N	COMPLETE ERROR	This filter can be used to fetch all the completed records or all the failed records.

5.2.6.2.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2	message	String	50	Y	message	Message from API

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1	suiteId	String	50	Y	12312-1233-123-12	Unique Id for bulk sign.
2	userId	String	50	Y	JID-	User Id who has initiated bulk sign process.
3	status	String		Y	INITIATED COMPLETE	Status of the bulk sign process.
4	items	Object[]				Items Object which holds the list of group Ids.
5	items.suiteItemId	String	50	Y	12312-1233-123-12	Unique Id for the document processed through bulk sign.
6	items.groupId	String	50	Y	1234	document envelop/groupId
7	items.message	String		Y	[{ "errcode": "SD-006", "message": "You have already signed this document." }]	Message contains the reason for the error occurred in document signing process.
8	items.status	String		Y	COMPLETE ERROR	If the status is COMPLETE, then the document has been signed successfully. In case the status is ERROR then check the message field to know the exact reason.

5.2.7 Document- Bulk DSC Sign/Decline WorkFlow

Prerequisite:

- The signatory must possess a dongle issued by the Certificate Authority (CA) authority. This dongle is essential to initiate bulk signing.
- Jiosigner, the signing application, must be invoked within the Signatory's system to enable the signing process.
- An action token will always be generated against the default account of the signatory. It is crucial to ensure that the document to be signed is present in the default account of the signatory.
- It is always recommended that the groupId passed in the signing request must be validated from the Eligible group Id's for Signatory API.

- For API calls, the action-token received in response in previous step must be passed as requests headers in the next step.
- For websockets calls, the action-token must be passed in the request body as token.

Workflow Steps	API Name	Description
Step 1	Login initiate	This API will initiate an OTP for a signatory on identifier (mobile number or email). As a response, it will return an action-token with the status as 'INITIATED.' This action-token should be passed in step 2.
Step 2 (This step is optional) Mandatory when users are signing documents for the first time, else it can be skipped.	Login verify/polling	The API provides information about the status of the OTP initiation call, and it requires the action-token from Step 1 to be passed as part of the headers. Once the status is received as "RUNNING", RPs should pass the OTP for verification, and continue polling again (without otp in request body) until status is "COMPLETE" (RP's will get action-token in response). If the user has already been authorised using OTP for the first time, then this step can be skipped, and RP's can use action-token received in Step1 to proceed further.
Step 3	Jiosigner Version	It will return the correct version of Jiosigner. The response from this request needs to be passed in Step 4 for the Websocket call.
Step 4 (Websocket call to JioSigner)	Action 1 - SIGN_INITIALISE	The INITIALISE request is sent to JioSigner, and in response, it provides the "action 2-SIGN_INITIALISE" response. It's important to note that the action-token should be passed as "token" in the request for "action 1" itself, and this token can be retrieved from either Step 1 or Step 2, depending on the specific context and user's signing history. For users signing first time, since OTP steps are required therefore pass action-token from step2 else pass action-token from step1 from next time onwards.
Step 5 (websocket call to Jiosigner)	Action 3 - SIGN_PKEY	The request for SIGN_PKEY Details is made, and upon clicking the "Sign" button on JioSigner, it returns the "Action 4-SIGN_PKEY" response. This response should be passed in the request of next document hash api call.
Step 6	DSC Hash Initiate	The API initiates the request for the document hash, and you should pass the action-token obtained from either Step 1 or Step 2. The details from the Step 4 response should be included in the request payload when making this API call. For users signing first time, since OTP steps are required therefore pass action-token from step2 else pass action-token from step1 from next time onwards.

		The action-token received in this api response should be passed as "action-token" in the request headers of the Step7 api call.
Step 7	DSC Hash status	To initiate the API polling request, RPs should continuously poll the API until a "HASHED" message is received. RPs need to pass the action-token received in response from Step 6. This will help to track the progress of the task until it reaches the "HASHED" message.
Step 8 (Websocket call to Jiosigner)	Action-5 Request for signed Hash	Request for SIGN_Document in action 5. It will return the signed hash. RPs should pass the document hash generated in the previous request (Step 7) as a requirement for this action.
Step 9 Same API as Step 6, difference in Payload	DSC SIGN Initiate	The API initiates the request for signing. RPs should pass the action-token in request headers obtained in response from Step 5. The response from Step 8, which contains the signed hash, should be included in the payload when making the API call.
Step 10 Same API as Step 7, difference in action-token	DSC Sign status	For the API polling request, RPs should continuously poll the API until they receive a "COMPLETE" status. RPs need to pass the action-token in request headers obtained in response from Step 9. This step will help to track the progress of the task until it reaches the "COMPLETE" status. This will complete the signing workflow.
Profile API (optional)	Profile Name Update	To update the profile name for the signing user. RPs need to pass the action-token from either Step 1 or Step 2, depending on the context and user's signing history.
Group details DSC (Optional)	Retrieve eligible group IDs for signatory.	The API returns the group IDs that are eligible for DSC signing against a specific user, and these group IDs are sent by the RP token account. RPs need to pass the action-token from either Step 1 or Step 2, depending on the context and user's signing history.

Important Points:

- If user is signing for the first-time using DSC as the signature method, then both Step1 and Step2 must be followed to authorise the user. From next time, RPs can skip Step2 for this user and can directly get the action-token from Step1 and proceed ahead.
- If Status is “INITIATED” from step 1, then RPs need to invoke step 2 and complete verify OTP process.
- If Status is “COMPLETE” from step 1 then RPs can skip Step2 and proceed ahead. In this case, RPs will receive action-token directly from Step1.
- Jiosigner can be downloaded from https://jiosign.com/jiosigner/win/64/JioSigner.exe
- Process of Signing documents using DSC method in bulk:
- Step1/Step2/ Step3/Step4 -> these steps need to be called only once for generating the session and initiating jiosigner desktop application which are pre-requisites for bulk DSC sign.
- RPs need to store the response of Step5 (action 3) from Jiosigner (pkey request) and execute rest of the steps in a loop for the groupIds to be signed. Please checkout Flow Diagram in Appendix E.
- Profile Name must be present for the user who is signing the document, else RPs need to update the profile name of the signatory by calling Profile Name Update API.

5.2.7.1 Login Initiate -Step1

5.2.7.1.1 Details

- API validates the input data and sends One Time Password (OTP) to identifier (Email/Phone) received in input.
- For the users signing using DSC for the first Time, RPs need to complete both Step1 and Step2 to complete OTP initiation and verification process.
- From next time onwards, RPs will directly get the action-token for that user, using which RPs can proceed ahead skipping Step2.
- If Status is “INITIATED”, RPs need to invoke Step 2 for OTP Verification.
- If status is “COMPLETE”, RPs will directly get the action-token and can proceed ahead skipping Step2.
- The expiration time of the action-token is 4 hours. RPs need to make sure to generate a new action-token on the expiry of the previous token.
- API will validate below fields:
    1. Terms and Condition field should be “Y” in the input request.
    2. Identifier passed in the input request should be valid and should be Valid Signatory.
    3. Authentication Type passed in the input request should be valid.
    4. Assurance Level should be valid (For DSC it should be 3).
- New action-token will be returned in the response.

Status Value	Description
INITIATED	• For the user signing first time user, consent is not stored in JS system. RP’s must verify OTP in Step2.
COMPLETE	• Consent is already stored; RP’s can directly proceed for signing

5.2.7.1.2 API Endpoints

Http Method	URL	Note	Output Format

POST	https://{FQDN}/docmgmt/v1.1/document/login/initiate	Refer {FQDN} in EndPoint section above.	application/json

5.2.7.1.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1	identifier	String	320	M	email/phone	Email or phone of user who is signing or declining the document. Signatory will be receiving OTP on same identifier.
2	authType	Int	2	M	2/3	Authentication Type defines kind of identifier, participant is using to validate his authenticity. Check API Constant for Authentication Type.
3	tandc	String	2	M	Y/N	User Agreement for Jiosign terms and condition. RPs needs to ask user select terms and condition and if required route user to JioSign portal for terms and condition.
4	asslvl	Int	2	CM	3	Assurance Level of Signing. For DSC only 3 is allowed Check API Constant for details.

Sample Payload:

```
{
 "identifier": "abc@ril.com",
 "authType": 2,
 "tandc": "Y",
 "asslvl": 3
}
```

5.2.7.1.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					

1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output: Success

In Case of First time Signing:

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	status	String	20	Y	INITIATED	Initiate the OTP status request
2.	action-token	String	NA	Y	Action Token	Temporary action token issued for the user who is signing/declining the document. Expiration time is 4 hours.

Sample Response:

```
{
    "status": "INITIATED",
    "action-token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.-VxxK2ODy-
ktiVD5Ubo0OeUv9H_BV-TfNCf4GKUexHZ9IQLB96-CqHZTyQxi5YBJ6RQzqa5GsPcweuiWcS4cVhTs_Y6s-
UJ8KYvc4sFj8kjhltXOWLneQA"
}
```

If consent is already stored for signing:

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	status	String	20	Y	COMPLETE	Consent already present return back token
2.	action-token	String	NA	Y	Action Token	action token issued for the user who is signing/declining the document.

Sample Response:

```
{
    "status": "COMPLETE",
    "action-token": " pxZHmohnhwf4su96pXpBjon6I-
yF1Z5YLKgr6NfNk2Wu7YjAJsEiiG37BUplifITqqNgHCjpgM8f7JUNJoV_zVG0bJ2ntPuzhD6Q2k2SNfcAr
JjTbcXNWdymLnpGhG0RffuPImPmXvGsk55gSaSvXKsdZ1JOMJqYUogIZ-
_8QhwRDbimjoIyrIGhkTWGhrUqJhMvJ6qugFrFWVbBhsrpx86rYeyxAUhoDSRwNC-
KhMCuGThwF0gV81fAULTFGCFA"
```

}

5.2.7.2 Login Verify/Polling- Step2

5.2.7.2.1 Details

- This API call checks the status of OTP initiation and also verifies it.
- RPs need to start polling with the code ='' ”, and once the status is 'RUNNING', RPs need to send the received OTP in the request body with 'code' and then continue polling again.
- RPs need to send the action-token in request header which have been received in the response from Step1.
- RPs need to continue polling till the **COMPLETE** status is received.
- This API will return status code as 400, if profile name, of the user signing the document, is not present, RPs can update the profile name using Profile Name Update API.
- New action-token will be returned in the response. RPs need to send this token in further steps.
- Responds back with 200 OK, with following **status** value, description of various status in the response.

Status Value	Description
RUNNING	• Once Running received, please submit the OTP and continue polling again
COMPLETE	• OTP Validation is complete. • RPs can start executing Step3

5.2.7.2.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/login/verify	Refer {FQDN} in EndPoint section above.	application/json

5.2.7.2.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1	code	String	320	M		Email or phone of user who is signing or declining the document. User will be receiving OTP on same identifier.

5.2.7.2.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	status	String	20	Y	Completed	Status completed stop polling
2.	action-token	String	NA	Y	Action Token	action token issued for the user who is signing/declining the document.

5.2.7.3 JioSigner Version - Step3

The API response will be passed to web socket call with action 1.

Http Method	URL	Note	Output Format
GET	https://jiosign.com/loginapi/commonsworker/v1.0/notifications/category/1/status/1/filter/1	Refer {FQDN} in EndPoint section above	application/json

Step4 and Step5 are web socket calls.

5.2.7.4 DSC Hash Initiate (Document Hash) – Step6

This API is for document hash process Initiation, and the action value for the payload will be 3.

5.2.7.4.1 Details

- Prechecks:
    - Signatory should have valid action-token and access to the document/group.
    - API will start the process of signing only if valid token and access is present, otherwise error will be sent.
    - API will validate if **the group is locked by any other participant. In case it is locked, error will be thrown.**
    - User selects the type of signing source which should be 3 for DSC and action should be 3 in payload for hash generation.
    - Validates if user has already signed the document.
    - **Any of the precheck validation fails error will be thrown back to caller.**
- Post checks:
    - **When RPs initiate signing process for a user for a specific group, JS System will lock the document and it will not be accessible to any other participant.**
    - Generates unique transaction Id for transaction tracking for sign request.
    - Source=3 →
        - If action → 3
            - API prepares data specific to DSC based request for Hash Calculation.
        - Check details flow in sequence diagram in Appendix E.
    - After the signing process completes, document will be unlocked and will be accessible by any other participant.
    - For Decline request (action=4), RP's will directly get the "COMPLETE" response, no need of any polling API's
    - In case of any unknown error, application will be unlocking the group.

- The expiration time for the action-token generated in this API is 30 minutes.

5.2.7.4.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/dsc/sign/initiate	Refer {FQDN} in EndPoint section above.	application/json

5.2.7.4.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1	source	Int	2	Y	3	User selected source for Signing of document. 3– DSC Token
2	groupId	String	50	Y	1234-2091-2912-2938	GroupID which user wants to sign.
3	action	Int	2	Y	1 /3/4	Action value to decide if document needs finalization or not. 1 – Document Sign request (Default) 3 – Document Hash Generation Request 4 – Document Decline request
4	pkey	String		Y		Public key of dongle in user's system who is initiating the sign request. Encoded value needs to be passed. **Mandatory**
5	cchain	String		Y		Certificate chain of dongle in user's system based on selection from frontend. Encoded value needs to be passed. **Mandatory**
	message	String	200	N	Comment to be passed	Comments while signing / declining to be added here

Sample Payload:

```
{
    "source": 3,
    "groupId": "0a90618a-8b4c-19ba-818b-4c8c72f50016",
    "action": 3,
    "pkey": "tOeF/+AvnW/Wb4bHYuUq2SOxF7OFhu+PxtZkRQPGziOD/V/rjOwD2F1WfTK7X6M+CX2n8r
/mHnU/k01ICgOq+FbpD64EN10JK6qpIn2DsbN39x0YmfuaBLDwUflF3/IpMcjcLFJLGEWdouQlYQKGj3b39
msmX0roNC9NJjP/dBbpA7EvxIvrbt2haa",
    "cchain": " dBbpA7EvxIvrbt2haaHXwm /lvRzOTZ9uKx5MMkD1QpPWxD"

}
```

5.2.7.4.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	status	String	20	Y	INITIATED	Initiate the Sign hash request
2.	action-token	String	NA	Y	Action Token	Temporary action token issued for the user who is signing/declining the document. Expiration time is 30 minutes.

Sample Response:

```
{
    "status": "INITIATED",
    "action-token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.
Aof7h3ljWZLNm361TjgM9NFNy_ti_B63_OjKEMEpWLANdv8HPwawGC0FbKAk2Oy9pBzj-
GkKvrYvIlZz0yhXTsJhfqUlVif7tE3HPunriyMLjXMfwwMMWDJRI2PaF16owFHrIBbDGTkxa2q06hFWsrpY
ZXHmbJLgu5cdLXZ1YGtkDwvB8CEy7BVuA2T-VxxK2ODy-ktiVD5Ubo0OeUv9H_BV-
TfNCf4GKUexHZ9IQLB96-CqHZTyQxi5YBJ6RQzqa5GsPcweuiWcS4cVhTs_Y6s-
UJ8KYvc4sFj8kjhltXOWLneQ"
}
```

5.2.7.5 DSC Hash Status (Document Hash polling) – Step7

This API can be used for checking the Sign status of step 6 (Document Hash Initiate).

5.2.7.5.1 Details

- RPs need to pass action-token in header received in api response from Step 6.
- Responds back with 200 OK along with following **status** value.
  - INITIATED – Request is initiated.
  - RECEIVED – **JS System** has started the hash generation process for that document.
  - RUNNING –RP’s Frontend System need to continue Polling till message received is “HASHED”.

- Important: If the message ='HASHED' is received, stop polling the API. RPs will receive the document hash in the response. If the message='RECEIVED,' the hash generation process is in progress, so continue polling the API.

5.2.7.5.2 Method Type: GET

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/dsc/sign/status	Refer {FQDN} in EndPoint section above.	application/json

5.2.7.5.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	NA					

5.2.7.5.4 Output

Output1:

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	20	Y	message	Message from API

Output2: With Status=RUNNING and message=HASHED

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	20	Y	message	Message from API
2.	status	String	20	Y	RUNNING	Status of the transaction, using this field Frontend will decide to poll for status or not. Frontend should stop polling once the status changes to RUNNING and message as HASHED, otherwise it needs to keep polling for status check.
3.	docs	[]				Contains list of all documents which needs to be signed and its hash.
4.	docs.id	String		Y		Document Id for the document which needs to be signed.
5.	docs.h	String		Y		Hash of the document which needs to be signed.

Sample Output:

```
{
    "message": "HASHED",
    "status": "RUNNING",
    "docs": [
        {
            "id": "0af4259b-8b61-125c-818b-9f662f860888",
            "h": "+38J1DhU8lHOybrHxCwRHpioPyNoMONwhKPdMLES1oE="
        }
    ]
}
```

Output3: With Received

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	20	Y	message	Message from API
2.	status	String	20	Y	Received	Keep pooling API document hash request in progress.

Step8 is a web socket call.

5.2.7.6 DSC Sign Initiate (action=1 in payload) – Step9

This API will initiate a signed hash generation request, and the action token from Step 6 needs to be passed.

5.2.7.6.1 Details

- Prechecks:
    - RPs should make sure that the user(signatory) have valid session and access to the document/group.
    - API starts signing only if valid action-token and access is present, otherwise error will be thrown.
    - API will validate if **the group is locked by any other participant. In case it is locked, error will be thrown.**
    - RPs will input the type of signing source as 3 for DSC.
    - API validates if user has already signed the document.
    - **Any of the precheck validation fails error will be thrown back.**
- Post checks:
    - **When user initiates signing process for group, System will lock the document signing for another signer of document.**
    - Generates unique transaction Id for transaction tracking for sign request.
    - Source=3 →

        If action → 1

            - API prepares data specific to DSC based request for signing the documents.

        - Check details flow in in sequence diagram.
    - After the signing process completes, document will be unlocked and will be accessible by any other participant.
    - Incase of any unknown error, API will unlock group.
    - The expiration time for the action-token generated in this API is 30 minutes.

5.2.7.6.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/dsc/sign/initiate	Refer {FQDN} in EndPoint section above.	application/json

5.2.7.6.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	source	Int	2	Y	3	3-DSC Sign
2.	groupId	String	50	Y	1234-2091-2912-2938	Group ID which user wants to sign.
3.	action	Int	2	Y	1/3/4	Action value to decide if document needs finalization or not. 1 – Document Sign request (Default) 3 – Document Hash Generation Request 4 -- Decline request
4.	pkey	String		Y		Public key of the dongle of user's system who is initiating the sign request. Encoded value needs to be passed.
5.	cchain	String		Y		Certificate chain of dongle in user's system based on selection from frontend. Encoded value needs to be passed.
6.	docs	[]				Contains list of all document which needs to be signed and its hash.
7.	docs.id	String		Y		GroupId to be signed
8.	docs.sh	String		Y		Signed hash of the document received from jiosigner
9.	message	String	200	N	Comment to be passed	Comments while signing / declining to be added here

Sample Payload:

```
{
    "source": 3,
    "groupId": "0a90618a-8b47-1ce7-818b-47d5f2300126",
    "action": 1,
    "pkey": "MIIFYDCCBEigAwIBAgIUSyxY1+ /
+PxtZkRQPGziOD/V/rjOwD2F1WfTK7X6M+CX2n8r/mHnU/k01ICgOq+FbpD64EN10JK6qpIn2DsbN39x0Ym
fuaBLDwUflF3/IpMcjcLFJLGEWdouQlYQKGj3b39msmX0roNC9NJjP/dBbpA7EvxIvrbt2haaHXwm",
    "cchain": "MIIFYDCCBEigAwIBAgIUSyxY1+U1dov74g1Khm0rcLaWHwcwDQYJKoZIhvcNAQELBQAw
LjERMA8GA1UEAwwIS1BMREVWQ0ExDDAKBgNVBAoMA0pQTDELMAkGA1UEBhMCSU4wHhcNMjIwODIzMDgwMjI
xWhcNMjUwODIyMDgwMjIwWjBdMRkwFwYDVQQDDBBKaW9TaWduIERTQyBUZXN0MRAwDgYDVQQLDAdUZXN0IE
9VMQwwCgYDVQQKDANKUEwxEzARBgNBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtbpjOGTaMmIZ55hioE
BcNEcVUrlkSBXUjdoW97PIn0vFvJOGulQI463nELeqzzQvYiHAI2529+m6ChjyzPFlmlRrSEqdFI84R9vYI
u4G05rlPRAUEBiRUGKJxo4pJ4+bUO+T1YPsEB+e2QxGGS+jEj2PJHlQvJN41fJ0mqiC6Rh8+eNwlhY30zIw
0gsRWs/zEcUPP9nRzYuCUPpA+5",
    "docs": [
        {
            "id": "0a90618a-8b47-1ce7-818b-47bd568c010f",
            "sh": "MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFADCABgkqhkiG9w0B
BwEAAKCAMIIFYDCCBEigAwIBAgIUSyxY1+U1dov74g1Khm0rcLaWHwcwDQYJKoZIhvcNAQELBQAwLjERMA8
GA1UEAwwISlBMREVWQ0ExDDAKBgNVBAoMA0pQTDELMAkGA1UEBhMCSU4wHhcNMjIwODIzMDgwMjIxWhcNMj
UwODIyMDgwMjIwWjBdMRkwFwYDVQQDDBBKaW9TaWduIERTQyBUZXN0MRAwDgYDVQQLDAdUZXN0IE9VMQwwC
gYDVQQKDANKUEwxEzARBgNVBAgMCk1haGFyYXN0cmExCzAJBgNVBAYTAklOMIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEAtbpjOGTaMmIZ55hioEBcNEcVUrlkSBXUjdoW97PIn0vFvJOGulQI463nELeqzzQ
vYiHAI2529+m6ChjyzPFlmlRrSEqdFI84R9vYIu4G05rlPRAUEBiRUGKJxo4pJ4+bUO+T1YPsEB+e2QxGGS
+jEj2PJHlQvJN41fJ0mqiC6Rh8+eNwlhY30zIw0gsRWs/zEcUPP9nRzYuCUPpA+ebpNkKKtS25AVZE7TaVa
Cq3JrmmKbYzmOdiWOn6xoENSGFaP9DU/ixQArpvOAILa6NiWUIDrZZ2MgXRiLYcY+no8/2SSPFi5vNt0fs4
T4dYhl5z+7TbAR2Y+6oSKUgi2cOwSQIDAQABo4ICRTCCAkEwDAYDVR0TAQH/BAIwADAfBgNVHSMEGDAWgBT
1tt2QMziMKaR5DEgOHuRfLUN9TDBZBggrBgEFBQcBAQRNMEswSQYIKwYB "
        }
    ],
    "message": ""
}
```

5.2.7.6.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	status	String	20	Y	INITIATED	Initiate the OTP status request
2.	action-token	String	NA	Y	Action Token	Temporary action token issued for the user who is signing/declining the document. Expiration time is 30 mins.

Sample Response:

```
{
    "message": "INITIATED",
    "action-token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.
_tdY9zPIhTZpZX3iBOQuoVkRS_8pmfuXAePPFwDxjvJZa1EsBw_4pi7F-
fNf4n6UTg4tYtp9K1qu_BOQGbdeecrVJqdYihI2RweWEcn6iMukPAjoCY-_eXIJ9PTb6Nz7vgR4z4hDnEG-
dkGKGT-
ygkyAG6hZHr7wglftUq7FJbbUYSFfqV4Q9NN1RAJvi0e_3fZzmVgjj_siV31JD3LKZ9f6XtQi4skGGMVVp9
jWS2MviLwUmadoAt9rV-
wAYmUPixMg47cpZUQPaq2PzJfF3SYPCk4n9FNiIYCgPfoKZUizC970_2TfYGnPg"
}
```

5.2.7.7 DSC Sign Status – Step10

This is a polling API for DSC Sign API (Step9), action-token should be passed from Step 9.

5.2.7.7.1 Details

- Responds back with 200 OK along with following **status** value.
  - INITIATED – Request is initiated.
  - RECEIVED – JS system has started signing the document.
  - RUNNING – RP Frontend need to continue Polling.
  - COMPLETE - Request is completed, RP can stop polling.

- IMP: message=”OK” and status=”COMPLETE” then stop polling

5.2.7.7.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/dsc/sign/status	Refer {FQDN} in EndPoint section above.	application/json

5.2.7.7.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	NA					

5.2.7.7.4 Output

Output1:

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	status	String	20	Y	COMPLETE	Status of the transaction, using this field RP Frontend will decide to poll for status or not. RP Frontend should stop polling once the status changes to COMPETE, otherwise it needs to keep polling for status check.

3.	token	String		Y		Document Signing is complete, RPs can ignore this token and proceed for signing the next document by using the action-token received in Step1/Step2.
4.	message	String	20	Y	message	Message from API

Output2: With Status=COMPLETE and message=OK

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	20	Y	message	Message from API
2.	status	String	20	Y	COMPLETE	Status of the transaction, using this field RP Frontend will decide to poll for status or not. RP Frontend should stop polling once the status changes to COMPETE, otherwise it needs to keep polling for status check.
3.	token	String		Y		Document Signing is complete, RPs can ignore this token and proceed for signing the next document by using the action-token received in Step1/Step2.
4	signatureId	String		Y		Signature ID generated for the signing participant

Sample Response:

```
{
    "message": "OK",
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJqc2xnbiI6Imh0dHBzOi8vamlvc2ln
bi5jb20vYXBpL290cC9sb2dpbiIsIm5ldG4iOiJNSkU0VmxuY0pnM0hIWi9EM0pjVTNnUUlITHM9IiwibGl
kIjoiRU1MLTAxZWQ3Nzk4LWM5YTMtMTI1MC1iOTMxLWJiM2FiMjgxYWJiNyIsImlzcyI6Ikppb1NpZ24uY2
9tIiwibHQiOiIyIiwic2Vzc2lvbklkIjoiZU5GbFpFOCtYekFJWVFsaFkxQWhHNEpTUTZvPSIsInRmYSI6I
lkiLCJ1aWQiOiJjMGE4NjE2ZS04NmEzLTEzMjUtODE4Ni1hNjM0NzI5ZDAxMjkiLCJsdGlkIjoiYzBhODYx
NTAtOGI5Ni0xZjkwLTgxOGItZWI0MDM4OWUyNThiIiwiYXNzTHZsIjoiMiIsInIiOiIzMCIsImF0IjoiMSI
sInRhYyI6IlkiLCJ0cmNpZCI6ImMwYTg2MTUwLTg0ZjItMWE2MS04MTg0LWY0ZDY1ODhmMDA2YiIsImZkbG
d0IjoiaHR0cHM6Ly9sb2dpbi5qaW9jb25uZWN0LmNvbS9vYW0vc2VydmVyL2xvZ291dD9lbmRfdXJsPUpJT
1NJR05fQ09NX1NBTUxfSU5URyIsImZkbGduIjoiaHR0cHM6Ly9sb2dpbi5qaW9jb25uZWN0LmNvbS9vYW1m
ZWQvaWRwL2luaXRpYXRlc3NvP3Byb3ZpZGVyaWQ9SklPU01lHTl9DT01fU0FNTF9JTlRHIiwianNsZ3QiOiJ
odHRwczovL2ppb3NpZ24uY29tL3VzZXIvbG9nb3V0IiwiZXhwIjoxNzAwNDczMzI2LCJpYXQiOjE3MDA0NT
g5MjYsImFpZCI6ImMwYTg2MTZlLTg2YTMtMTMyNS04MTg2LWE2MzQ2YjhiMDEyNyJ9.yzPQgnyjw7734HrU
DeTIgAQZIoPO_M05osWOPOpX4ysm1E-w1uSnwQQj7BTfHJacwgwGm2ZKod6-qCCj-
nve45he6PqkfBShCDaCpRXXAQD8B12eRY6NnY1i2p8iozROIwjlooLwxzYkfq0uxz5L-
JdebAlMDsRBvEy8dMutTqLczXB5uSAqSqC5iQPAv69DYg_E3ZvwMq5LeIkRhE8yaO9aJJSMze9QTUbTo94h
c9UIPRltjqVxVMWxDcuNWmh_AjVr0yi1Yx0R1N9Pq2fttV05csGyg6HdGUhLy3VTz0D6k6KidFtn6tSmGTf
XEVH5ozKSNHA04-jyUg7zkMwxiQ",
    "status": "COMPLETE",
    "signatureId": "86396cdf-8b5f-4657-a8c6-38ae3aac4035"
}
```

Output2: With Received

#	Parameter	Type	Length	Mandatory	Sample	Description
3.	message	String	20	Y	message	Message from API
4.	Status	String	20	Y	Received	Keep pooling API document hash request in progress.

5.2.7.8 Web Socket calls from RP System to Jiosigner

These are the calls which will be happening between RP System to JioSigner for different actions.

5.2.7.8.1 Jiosigner websocket based Action=1-SIGN_INITILISE

RPs need to invoke a request from their system to JioSigner and pass the action-token from Step 1 or Step 2 based on consent status. It will return an action=2, which is the sign_initialize response.

Request:

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	action	Int	2	Y	1	
2.	pkey	String		CM	Pass empty	Public key of the dongle of user's system who is initiating the sign request. Encoded value needs to be passed.
3.	cchain	String		CM	Pass empty	Certificate chain of dongle in user's system based on selection from frontend. Encoded value needs to be passed.
4.	errMsg	String		CM	Pass empty	If any errmsg will be passed here
5.	token	String		M		Action Token received in earlier request
6.	uri	String		CM	https://jiosign.com/	Url of a portal
7.	docs	[]				Contains list of all document which needs to be signed and its hash.
8.	docs.id	String		M		group Id for the document which needs to be signed.
9.	docs.hash	String		CM	Pass Empty	
10.	docs.signedhash	String		CM	Pass empty	Signed hash of the document received from jiosigner
11.	luuri	String	200	M		This uri is used to check the version upgrade of jiosigner

Sample message:

```
{
    "action": 1,
    "pkey": "",
    "cchain": "",
    "errmsg": "",
    "token": ".NcoW35wtf9zzpot1xMpUUj6mnvOqWfX7izoXUaOSUFhupHWTPNBhOkhpdzi9HhX84WU0
3skiuzEZFxOoYtlhuTlSKF2TAjqLkuotGNrhw0x4848uaL2bYhK7AgV1ApEvvO7NHY-
7nMOY7h6GxpONskr6WBj1_9znalBc-
BmAOiFRShJVXUnRmGTM_6ksNrd84adBZCgZCOOo37BoX8asz6eXWZGzzq3ltSo81S1a6Xewr67tn0u4ARqy
YqjjEGeTzt4h1-xi-CtraTL-wnrdAbPS-
zS3ghsshoVxJ299gACCjPFobhd3LNsOSaIHULnVedbWOPj22AqLESrle5MJyg",
    "uri": "https://jiosign.com/",
    "docs": [
        {
            "id": "0af4629f-8b37-1b1d-818b-4777f92f04a1",
            "hash": "",
            "signedhash": ""
        }
    ],
    "luuri": "https://jiosign.com/"

}
```

Return Response: Action 2- SIGN_INITILISE_Response

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	action	Int	2	Y	1	
2.	pkey	String		CM		Public key of the dongle of user's system who is initiating the sign request. Encoded value needs to be passed. **Mandatory**
3.	cchain	String		CM		Certificate chain of dongle in user's system based on selection from frontend. Encoded value needs to be passed. **Mandatory**
4.	errMsg	String		CM		If any errmsg will be passed here
5.	token	String		M		Action Token received in earlier request
6.	uri	String		CM		Url of a portal
7.	docs	[]				Contains list of all document which needs to be signed and its hash.
8.	docs.id	String		M		group Id for the document which needs to be signed.
	docs.hash	String		CM		
9.	docs.signedhash	String		CM		Signed hash of the document. Which is received from jiosigner

10.	luuri	String	200	M		This uri is used to check the version upgrade of jiosigner

Sample Action 2 response:

```
{
    "action": 2,
    "pkey": "",
    "cchain": "",
    "errmsg": "",
    "token": ".NcoW35wtf9zzpot1xMpUUj6mnvOqWfX7izoXUaOSUFhupHWTPNBhOkhpdzi9HhX84WU0
3skiuzEZFxOoYtlhuTlSKF2TAjqLkuotGNrhw0x4848uaL2bYhK7AgV1ApEvvO7NHY-
7nMOY7h6GxpONskr6WBj1_9znalBc-
BmAOiFRShJVXUnRmGTM_6ksNrd84adBZCgZCOOo37BoX8asz6eXWZGzzq3ltSo81S1a6Xewr67tn0u4ARqy
YqjjEGeTzt4h1-xi-CtraTL-wnrdAbPS-
zS3ghsshoVxJ299gACCjPFobhd3LNsOSaIHULnVedbWOPj22AqLESrle5MJyg",
    "uri": "https://jiosign.com/",
    "docs": [
        {
            "id": "0af4629f-8b37-1b1d-818b-4777f92f04a1",
            "hash": "",
            "signedhash": ""
        }
    ],
    "luuri": "https://jiosign.com/"

}
```

5.2.7.8.2 Jiosigner websocket based Action=3-SIGN_PKEY

Request the private key (action=3) and receive a response with action=4, including private key and certificate chain details. Click the 'Sign' button in JioSigner, and it will return action 4..

Request:

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	action	Int	2	Y	1	
2.	pkey	String		CM	Pass empty	Public key of the dongle of user's system who is initiating the sign request. Encoded value needs to be passed. **Mandatory**
3.	cchain	String		CM	Pass empty	Certificate chain of dongle in user's system based on selection from frontend. Encoded value needs to be passed. **Mandatory**
4.	errMsg	String		CM	Pass empty	If any errmsg will be passed here
5.	token	String		M		Action Token received in earlier request

6.	uri	String		CM	https://jiosign.com/	Url of a portal
7.	docs	[]				Contains list of all document which needs to be signed and its hash.
8.	docs.id	String		M	Pass empty	group Id for the document which needs to be signed.
9.	docs.hash	String		CM		
10.	docs.signedhash	String		CM	Pass empty	Signed hash of the document received from jiosigner
11.	luuri	String	200	M		This uri is used to check the version upgrade of jiosigner

Sample :

```
{
    "action": 3,
    "pkey": "",
    "cchain": "",
    "errmsg": "",
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJqc2xnbiI6Imh0dHBzOi8vamlvc2ln
bi5qaW8uY29tL2FwaS9zYW1sL2xvZ2luIiwibmV0biI6IjJMaVV4dUZ6dFlQK0ZmTm5vOXRMU1dFUks3VT0
iLCJsaWQiOiJFTUwtMDFlZGI3NmMtODg1NS0xNWM4LWI4OTYtZTg0NjcxMjY5ODc1IiwiaXNzIjoiSmlvU2
lnbi5jb20iLCJsdCI6IjIiLCJzZXNzaW9uSWQiOiJYQVB1eHI0MFg2WnFBRFZ4SVllcVBJdmx0eWc9Iiwid
GZhIjoiWSIsInVpZCI6IjBhZjQzODlmLTg5Y2UtMTNlZS04MThhLTU3NzU3YTAwMDlkZSIsImx0aWQiOiIw
YWY0NjI5OC04YjM2LTE3MjEtODE4Yi00NzVlOTM0MzAwZWQiLCJhc3NMdmwiOiIyIiwiciI6IjMwIiwiYXQ
iOiIxIiwiaWRlbiI6IkNoYW5kcmFrYW50MS5HQHJpbC5jb20iLCJ0YWMiOiJZIiwidHJjaWQiOiIwYWY0Nj
JiOC04NjZmLTEzN2ItODE4Ni05NmQ2MmM3YjA0OWIiLCJmZGxndCI6Imh0dHBzOi8vc2l0b2hzLmppby5jb
206NDQ0My9vYW0vc2VydmVyL2xvZ291dD9lbmRfdXJsPUpJT1NJR05fU0FNTF9JTlRHX0RFViIsImZkbGdu
IjoiaHR0cHM6Ly9zaXRvaHMuamlvLmNvbTo0NDQzL29hbWZlZC9pZHAvaW5pdGlhdGVzc28_cHJvdmlkZXJ
pZD1KSU9TSUdOX1NBTUxfSU5URl9ERVYiLCJuYW0iOiJDaGFuZHJha2FudDEuR0ByaWwuY29tIiwianNsZ3
QiOiJodHRwczovL2ppb3NpZ24uamlvLmNvbS91c2VyL2xvZ291dCIsImV4cCI6MTY5NzcyMzg1NSwiaWF0I
joxNjk3NzA5NDU1LCJhaWQiOiIwYWY0NjJhOC04ODA1LTE2ZDAtODE4OC1mMWEwNmY0OTEzZTgifQ.NcoW3
5wtf9zzpot1xMpUUj6mnvOqWfX7izoXUaOSUFhupHWTPNBhOkhpdzi9HhX84WU03skiuzEZFxOoYtlhuTlS
KF2TAjqLkuotGNrhw0x4848uaL2bYhK7AgV1ApEvvO7NHY-7nMOY7h6GxpONskr6WBj1_9znalBc-
BmAOiFRShJVXUnRmGTM_6ksNrd84adBZCgZCOOo37BoX8asz6eXWZGzzq3ltSo81S1a6Xewr67tn0u4ARqy
YqjjEGeTzt4h1-xi-CtraTL-wnrdAbPS-
zS3ghsshoVxJ299gACCjPFobhd3LNsOSaIHULnVedbWOPj22AqLESrle5MJyg",
    "uri": "https://jiosign.com/",
    "docs": [
        {
            "id": "0af4629f-8b37-1b1d-818b-4777f92f04a1",
            "hash": "",
            "signedhash": ""
        }
    ],
    "luuri": "https://jiosign.com/"

}
```

Response: Action 4-SIGN_PKEY_RESPONSE

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	action	Int	2	Y	1	
2.	pkey	String		CM		Public key of the dongle of user's system who is initiating the sign request. Encoded value needs to be passed. **Mandatory**
3.	cchain	String		CM		Certificate chain of dongle in user's system based on selection from frontend. Encoded value needs to be passed. **Mandatory**
4.	errMsg	String		CM		If any errmsg will be passed here
5.	token	String		M		Action Token received in earlier request
6.	uri	String		CM		Url of a portal
7.	docs	[]				Contains list of all document which needs to be signed and its hash.
8.	docs.id	String		M		group Id for the document which needs to be signed.
9.	docs.hash	String		CM		
10.	docs.signedhash	String		CM		Signed hash of the document. Which is received from jiosigner
11.	luuri	String	200	M		This uri is used to check the version upgrade of jiosigner

Sample:

```
{
    "action": 4,
    "pkey": "MIIFYDCCBEigAwIBAgIUSyxY1+
+bUO+T1YPsEB+e2QxGGS+jEj2PJHlQvJN41fJ0mqiC6Rh8+eNwlhY30zIw0gsRWs/zEcUPP9nRzYuCUPpA+
ebpNkKKtS25AVZE7TaVaCq3JrmmKbYzmOdiWOn6xoENSGFaP9DU/ixQArpvOAILa6NiWUIDrZZ2MgXRiLYc
Y+no8/ /dBbpA7EvxIvrbt2haaHXwm",
    "cchain": " /P/W87UwcS+
/gSFyyyCbCiKBQfedEn/IFEhM/TuGaWsA0IFi5GNPZXfXCSQYYx79XwjYp47jr8gsHpGjX3mXNosTyu8D7W
CCxnKCp9vbu0SkajThHn2S/AAmubGS/aj695dGwvloqqPFwrqI4DxCg08444pvkUan85xwGS32k9t7w9qSk
WEw9IxfNKHKmJIp3GxRUtRl2lmU0Vq2Zg2NjH9shbfnXVU1f7/lvRzOTZ9uKx5MMkD1QpPWxDd5",
    "docs": [
        {
            "id": "0af4629f-8b37-1b1d-818b-4777f92f04a1",
            "hash": "",
            "signedhash": ""
        }
    ],
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJsaWQiOiJFTUwtMDFlZGI3NmMtODg1
NS0xNWM4LWI4OTYtZTg0NjcxMjY5ODc1IiwiaXNzIjoiSmlvU2lnbi5jb20iLCJzZXNzaW9uSWQiOiJKL2F
RV2tPV08xbE9jU3JGeGVrQU8wN0VlcFE9IiwidGZhIjoiWSIsInJwVHJhbnNObyI6ImJtdzJOM0ZaVlc1al
RTOTVWWGRuVlZCSlVraDJPRkZPTkdOQlBRIiwiYXNzTHZsIjoiMyIsInVpZCI6IjBhZjQzODlmLTg5Y2UtM
TNlZS04MThhLTU3NzU3YTAwMDlkZSIsInIiOiIzMCIsImF0IjoiMSIsImlkZW4iOiJjaGFuZHJha2FudDEu
Z0ByaWwuY29tIiwidGFjIjoiWSIsIm5hbSI6IkNoYW5kcmFrYW50IiwiZXhwIjoxNjk4NzI5NDQ1LCJpYXQ
iOjE2OTg3Mjc2NDUsImFpZCI6IjBhZjQ2MmE4LTg4MDUtMTZkMC04MTg4LWYxYTA2ZjQ5MTNlOCJ9.XDb0M
XV3xVahWC3SAQRL8HFjCH4C4Ob4mtmPwp160HLNSUlSvRt7w- ",
    "uri": "https://jiosign.com/",
    "errmsg": "",
    "luuri": "https://jiosign.com/"

}
```

5.2.7.8.3 Jiosigner websocket based Action=5 SIGN_DOCUMENT

Request the signed hash (action 5) and receive action 6 (the signed hash) in response. You need to pass the document's hash here, and the action token from Step 6 should also be included as a token.

Request:

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	action	Int	2	Y	15	
2.	pkey	String		CM	Pass Value received from action 4	Public key of user who is initiating the sign request. Encoded value needs to be passed.
3.	cchain	String		CM	Pass Value received from action 4	Certificate chain based on user selection from frontend. Encoded value needs to be passed.
4.	errMsg	String		CM	Pass empty	If any errmsg will be passed here
5.	token	String		M	Action-token	Action Token received in earlier request
6.	uri	String		CM	https://jiosign.com/	Url of a portal
7.	docs	[]				Contains list of all document which needs to be signed and its hash.
8.	docs.id	String		M		Id received in Hash request response . Step 6
9.	docs.hash	String		CM		Hash value received in Step 6 needs be passed here
10.	luuri	String	200	M		This uri is used to check the version upgrade of jiosigner

Sample:

```
{
    "action": 5,
    "pkey": "MIIFYDCCBEigAwIBAgIUSyxY1+
+2+o/byo0tv/AVl+VQvcuIwDgYDVR0PAQH/BAQDAgO4MA0GCSqGSIb3DQEBCwUAA4IBAQAbtruHUCscSJEC
sG8WWfRKP6mulbUojmdzkRWNG0fl+bkxgJ2fI9Crobq4SiQBkNUhmePlFdI3iWH2IdPeNxwRi3tZwAfoffw
wR7sU//0ZLbEvRGmJtifNAtpzBQtLbhocQPI9l2U+Dkgjfwz7gJbCU4tOeF/+AvnW/Wb4bHYuUq2SOxF7OF
hu+PxtZkRQPGziOD/V/rjOwD2F1WfTK7X6M+CX2n8r/mHnU/k01ICgOq+FbpD64EN10JK6qpIn2DsbN39x0
YmfuaBLDwUflF3/IpMcjcLFJLGEWdouQlYQKGj3b39msmX0roNC9NJjP/dBbpA7EvxIvrbt2haaHXwm",
    "cchain": "MIIFYDCCBEigAwIBAgIUSyxY1+
+no8/2SSPFi5vNt0fs4T4dYhl5z+7TbAR2Y+6oSKUgi2cOwSQIDAQABo4ICRTCCAkEwDAYDVR0TAQH/BAIw
ADAfBgNVHSMEGDAWgBT1tt2QMziMKaR5DEgOHuRfLUN9TDBZBggrBgEFBQcBAQRNMEswSQYIKwYBBQUHMAG
GPWh0dHA6Ly9kZXYuZWpiY2EuamlvbGFicy5jb206ODA4MC9lamJjYS9wdWJsaWN3ZWIvc3RhdHVzL29jc3
AwgYkGA1UdLgSBgTB/MH2ge6B5hndodHRwOi8vZGV2LmVqYmNhLmppb2xhYnMuY29tOjgwODAvZWpiY2Evc
HVibGljd2ViL3dlYmRpc3QvY2VydGRpc3Q/ /P/W87UwcS+
/AAmubGS/aj695dGwvloqqPFwrqI4DxCg08444pvkUan85xwGS32k9t7w9qSkWEw9IxfNKHKmJIp3GxRUtR
l2lmU0Vq2Zg2NjH9shbfnXVU1f7/lvRzOTZ9uKx5MMkD1QpPWxDd5",
    "errmsg": "",
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9..h_B6eX4TcjqA_gKWb8wD3wwxaXqZC3P
rvdPomSsXe8gGIGuGX3SCaGK2CkuTJEumhGgiVF1aezMnxz3B7hBNXGVypskE1h8ko41dEdw9lTuEkfHNr3
RG-VmL3VT0bmFTwqRz9itTxJ7xZbCjF5Op-JJ42dcP_zzAHKhfvufgmoenFpk0utMCqhOrWk9Xi-
euyTdEgYrgPh9iEcmr81QG4ORSS2CS5XynTu3dPSx7edRmcu5wLL54YdCJ22FEHgAh8axOO_kUmSw1_GNh_
N068sn5tKJiKQDreOKHFCoVewBpFSgonwIbQukudwS2q-EBF3TFaJYWEAuqyC9-u0e1kg",
    "uri": "https://jiosign.com/",
    "docs": [
        {
            "id": "0af4629f-8b37-1b1d-818b-4777f92f04a1",
            "hash": "FmYteofZ79RFgdY1nJ/Zt5OMkR/o1WGw7QKpYWtdXkM="
        }
    ],
    "luuri": https://jiosign.com/

}
```

Response: Action 6 SIGN_DOCUMENT_RESPONSE

#	Parameter	Type	Length	Mandatory	Sample	Description
1	action	Int	2	Y		
2	pkey	String		CM		Public key of user who is initiating the sign request. Encoded value needs to be passed. **Mandatory**
3	cchain	String		CM		Certificate chain based on user selection from frontend. Encoded value needs to be passed. **Mandatory**
4	errMsg	String		CM		If any errmsg will be passed here

5	token	String		M		Action Token received in earlier request
6	uri	String		CM	https://jiosign.com/	Url of a portal
8	docs	[]				Contains list of all document which needs to be signed and its hash.
9	docs.id	String		M		group Id for the document which needs to be signed.
10	docs.hash	String		CM		
11	docs.signedhash	String		CM		Signed hash of the document. Which is received from jiosigner
12	luuri	String	200	M		This uri is used to check the version upgrade of jiosigner

Sample:

```
{
    "action": 6,
    "pkey": "MIIFYDCCBEigAwIBAgIUSyxY1+
/+AvnW/Wb4bHYuUq2SOxF7OFhu+PxtZkRQPGziOD/V/rjOwD2F1WfTK7X6M+CX2n8r/mHnU/k01ICgOq+Fb
pD64EN10JK6qpIn2DsbN39x0YmfuaBLDwUflF3/IpMcjcLFJLGEWdouQlYQKGj3b39msmX0roNC9NJjP/dB
bpA7EvxIvrbt2haaHXwm",
    "cchain": "MIIFYDCCBEigAwIBAgIUSyxY1+
ifNAtpzBQtLbhocQPI9l2U+Dkgjfwz7gJbCU4tOeF/+AvnW/Wb4bHYuUq2SOxF7OFhu+PxtZkRQPGziOD/V
/rjOwD2F1WfTK7X6M+CX2n8r/mHnU/k01ICgOq+FbpD64EN10JK6qpIn2DsbN39x0YmfuaBLDwUflF3/IpM
cjcLFJLGEWdouQlYQKGj3b39msmX0roNC9NJjP/dBbpA7EvxIvrbt2haaHXwmMIIDTTCCAjWgAwIBAgIULg
W/P/W87UwcS+ziDrNLGsqIKWMwDQYJKoZIhvcNAQELBQAwLjERMA8GA1UEAwwISlBMREVWQ0ExDDAKBgNVB
AoMA0pQTDELMAkGA1UEBhMCSU4wHhcNMjExMDA3MDQzMTU5WhcNMzExMDA1MDQzMTU4WjAuMREwDwYDVQQD
DAhKUExERVZDQTEMMAoGA1UECgwDSlBMMQswCQYDVQQGEwJJTjCCASIwDQYJKoZIhvcNAQEBBQADggEPADC
CAQoCggEBAM5CguS/lffo+//ctuWCIZe+QOXVuQy1KOneMhLjFkmS9+ngpdTSLy3TB58tz8iGgBtA6QURAN
Y/N2uWD660g4rV0VByQPx0AWzQ7nygXOU0G/ySHoYZvWOsFVgwb5CnqmBdQBuKL5yJ9AxXxyA4HXFR6ksQt
0dk/ilvXrAiI2yvdmcpPf4F5gFMj+RWmLg57TDWbPXY6m7/biQLa/UgD/Ed67Z5m7QUAzqjh84tt8Xa9ulv
OCw9snicZfsGK4fpUIOUXA5BU0exspqprOzqhudQhYAB6n8cc76pVSR5irhIikoKaBTkP+DXmLsOL6r/sVk
MoftgCR2Fi7l6mZD/ZGECAwEAAaNjMGEwDwYDVR0TAQH/BAUwAwEB/zAfBgNVHSMEGDAWgBT1tt2QMziMKa
R5DEgOHuRfLUN9TDAdBgNVHQ4EFgQU9bbdkDM4jCmkeQxIDh7kXy1DfUwwDgYDVR0PAQH/BAQDAgGGMA0GC
SqGSIb3DQEBCwUAA4IBAQDJhgGzHX0AQm9KdWozRSGQDZD+yJj+pvqfGKlFbg4UYXaSIh+HqwQMpXHvix0W
9OVebT1BQkM1TFIayWnzaCxQ/3x41K303dRA7lO/gSFyyyCbCiKBQfedEn/IFEhM/TuGaWsA0IFi5GNPZXf
XCSQYYx79XwjYp47jr8gsHpGjX3mXNosTyu8D7WCCxnKCp9vbu0SkajThHn2S/AAmubGS/aj695dGwvloqq
PFwrqI4DxCg08444pvkUan85xwGS32k9t7w9qSkWEw9IxfNKHKmJIp3GxRUtRl2lmU0Vq2Zg2NjH9shbfnX
VU1f7/lvRzOTZ9uKx5MMkD1QpPWxDd5",
    "docs": [
        {
            "id": "0af4629f-8b37-1b1d-818b-4777f92f04a1",
            "hash": "FmYteofZ79RFgdY1nJ/Zt5OMkR/o1WGw7QKpYWtdXkM=",
            "signedhash": "MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFADCABgkq
hkiG9w0BBwEAAKCAMIIFYDCCBEigAwIBAgIUSyxY1+U1dov74g1Khm0rcLaWHwcwDQYJKoZIhvcNAQELBQA
wLjERMA8GA1UEAwwISlBMREVWQ0ExDDAKBgNVBAoMA0pQTDELMAkGA1UEBhMCSU4wHhcNMjIwODIzMDgwMj
IxWhcNMjUwODIyMDgwMjIwWjBdMRkwFwYDVQQDDBBKaW9TaWduIERTQyBUZXN0MRAwDgYDVQQLDAdUZXN0I
E9VMQwwCgYDVQQKDANKUEwxEzARBgNVBAgMCk1haGFyYXN0cmExCzAJBgNVBAYTAklOMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtbpjOGTaMmIZ55hioEBcNEcVUrlkSBXUjdoW97PIn0vFvJOGulQI463
nELeqzzQvYiHAI2529+m6ChjyzPFlmlRrSEqdFI84R9vYIu4G05rlPRAUEBiRUGKJxo4pJ4+bUO+T1YPsEB
9usPVu8f1wWFRyoep5jbAFuJzel5xdXd4h2q+iWeF4+VSIRebYHicHRKfNKNDqIJBVYadTVi78OILRLzQdA
Xq61GdOIA8//Aq3YEGXk+OGGFHmwD4/hvbY6qAIl9ImgzWz/8Q9Y/TAE8JcFuq0LE/1Hk324hEkXfV9xXe0
hgQyESWD+YFx/IoEFmkFYiToNLNSkcfymPu/2OXzBuDZALYC7/R/0S8hJgNIFye/+XjpL2T+zqUJps3y9yw
IMVm2AX7PCDV1/wBkA4C4/3RI/9x7yrNc1d5IjAR2RGNPFIgpMaanja+x+krDZ4JVxQd555795RsjKKUGSQ
AAAAAAAA="
        }
    ],
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9. -u0e1kg",
    "uri": "https://jiosign.com/",
    "errmsg": "",
    "luuri": "https://jiosign.com/"

}
```

5.2.7.8.4 Jiosigner websocket based Action=9 CLOSE_JIOSIGNER

Call to close jiosigner.

Sample :

```
{
    "action": 9,
    "pkey": "",
    "cchain": "",
    "errmsg": "",
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJsaWQiOiJFTUwtMDFlZDNkOTctZDNi
Yi0xYTcwLWI0ZWUtZDNiNWVhOTU3MjIyIiwiaXNzIjoiSmlvU2lnbi5jb20iLCJzZXNzaW9uSWQiOiJrclU
rc2NqWkxiaUVIUTVMRGVYTnRBbitVcTg9IiwidGZhIjoiWSIsInJwVHJhbnNObyI6Ik9WcHdWa3g1VUdaNm
FuWlFlVFV6VWs1a1RtaHlkM2hrVGxaM1BRIiwiYXNzTHZsIjoiMyIsInVpZCI6IjBhZjQ2MmJlLTg0YTAtM
TJiNS04MTg0LWM3YTMzZmNmMDMzZCIsInIiOiIzMCIsImF0IjoiMSIsImlkZW4iOiJuYXZ5YS5rb25kYWtp
bmRpQHJpbC5jb20iLCJ0YWMiOiJZIiwibmFtIjoiUmVkZHkiLCJleHAiOjE3MDA0NjAwMTEsImlhdCI6MTc
wMDQ1ODIxMSwiYWlkIjoiMGFmNDYyYmUtODRhMC0xMmI1LTgxODQtYzdhMzM4ZGEwMzNiIn0.YLMHXUjTKv
H0qZEJA1L8PavGbT8lrf1QhaPXRgdESS3TELZSYck-
tpmOGVAsQevhWTUMOaa5BhXZ6osuNP5aMHVzJzQXho9OGdrWE5CmaH_LkrBvt6FSkJgRdDWbuXZgatZdRDC
ITBbDyaV8Dl1uj2b-
_HNMVhvbmCN9BeEZeaAJDUek0r0uWfLsI2l0Wh61FONp0zDwyPYGmzs1DKpubl169QBpIekSU9Hmj-
kEy9m7uNnHeZw22Rul9ZlanrDT6u8tq6mX2BCQvaErzAhvUZ3IpqPc2_lcLrNwbbAbFFMcB37nNXW78XifD
p5HSP4H9MLdLlMEYTI8M5q85iQG_g",
    "uri": "https://jiosign.com/",
    "docs": [
        {
            "id": "0af4629f-8b37-1b1d-818b-4777f92f04a1",
            "hash": "",
            "signedhash": ""
        }
    ],
    "luuri": "https://jiosign.com/"

}
```

Response:

Jiosigner will be closed.

Request:

#	Parameter	Type	Length	Mandatory	Sample	Description
1	action	Int	2	Y	9	

5.2.7.8.5 Jiosigner websocket based Action=10 Unauthorized access

If Token is Invalid or any other error message related to authorization, It will throw this error message, RP's must handle these error messages

Response:

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	action	Int	2	Y	10	
2.	pkey	String		CM	Pass empty	Public key of user who is initiating the sign request. Encoded value needs to be passed. **Mandatory**
3.	cchain	String		CM	Pass empty	Certificate chain based on user selection from frontend. Encoded value needs to be passed. **Mandatory**
4.	errMsg	String		CM	Pass empty	Error message will be appear here
5.	token	String		M		Action Token received in earlier request
6.	uri	String		CM	https://jiosign.com/	Url of a portal
7.	docs	[]				Contains list of all document which needs to be signed and its hash.
8.	docs.id	String		M	Pass empty	group Id for the document which needs to be signed.
9.	docs.hash	String		CM		
10.	docs.signedhash	String		CM	Pass empty	Signed hash of the document. Which is received from jiosigner
11.	luuri	String	200	M		This uri is used to check the version upgrade of jiosigner

Sample Error response:

```
{
    "action": 10,
    "errmsg": " Unauthorized access, please pass the correct values."
}
```

5.2.7.8.6 Jiosigner websocket based Action=8 Error message from Jiosigner

If Jiosigner closed by cross button, this action will be trigged this error handling must be done on RP side. Its a Jiosigner Response . Sample format below.

Response:

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	action	Int	2	Y	8	
2.	pkey	String		CM	Pass empty	Public key of user who is initiating the sign request. Encoded value needs to be passed. **Mandatory**
3.	cchain	String		CM	Pass empty	Certificate chain based on user selection from frontend. Encoded value needs to be passed. **Mandatory**
4.	errMsg	String		CM	Pass empty	If any errmsg will be passed here
5.	token	String		M		Action Token received in earlier request
6.	uri	String		CM	https://jiosign.com/	Url of a portal
7.	docs	[]				Contains list of all document which needs to be signed and its hash.
8.	docs.id	String		M	Pass empty	group Id for the document which needs to be signed.
9.	docs.hash	String		CM		
10.	docs.signedhash	String		CM	Pass empty	Signed hash of the document. Which is received from jiosigner
11.	luuri	String	200	M		This uri is used to check the version upgrade of jiosigner

Sample Error response:

```
{
    "action": 8,
    "pkey": "",
    "cchain": "",
    "docs": [
        {
            "id": "0af4629f-8b37-1b1d-818b-4777f92f04a1",
            "hash": "",
            "signedhash": ""
        }
    ],
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.
fbUoUAoKcNvCiTpo_3YutDbRbJ3_eJAl44KvmnYSbA",
    "uri": "https://jiosign.com/",
    "errmsg": "You have cancelled document signing",
    "luuri": "https://jiosign.com/"
```

}

5.2.7.9 Profile Name Update API for DSC

5.2.7.9.1 Details

- API will update the Profile Name of the user signing the document by DSC.
- API needs valid user session therefore RPs need to send the action-token in request header which have been received in the response from Step1/Step2.
- If profile name already exits, it will update the name with new value.
- Pass asslvl=3 for profile Update.

5.2.7.9.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/sign	Refer {FQDN} in EndPoint section above.	application/json

5.2.7.9.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	name	String	100	M		Name of the user to be used for document signing. Mandatory for signing request if name does not exist in the JioSign System.
2.	assLvl	String	2	M	3	DSC Update profile Name(Must pe passed 3)

5.2.7.9.4 Output

Output1: Error, in case API runs into technical error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	100	Y	Profile name updated successfully	It will return success response

5.2.7.10 Get Group details for DSC

API will return all the documents which are pending for the user's signature to be signed by DSC as signature method. RPs can fetch all such documents for the user from this API and then loop in for Bulk DSC signing.

5.2.7.10.1 Details

API will return all group Ids related to DSC, waiting for user's signature.

RPs should be an owner or creator for all these documents.

5.2.7.10.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/document/pages	Refer {FQDN} in EndPoint section above.	application/json

5.2.7.10.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	limit	int		Y	500	No of documents to be return(max limit 500)
2.	offset	int		Y	1	Page number
3.	sort	Object{}		CM		
4.	sort.by	Int		Y	1,2,3	Order By value 1- Create timestamp 2- Group name 3- Status 4- Last signed Date
5.	sort.order	int		Y	1	Order ascending, descending 1- Ascending 2- Descending
6.	search	Object{}		CM		
7.	search.groupName	String	200	CM	ABC	Name of the group
8.	search.groupIds	List		CM	groupId	Comma separated groupIds.
9.	search.status	Int		CM	1	1-ALL
10.	search.filter	Int		CM	8	Filter for different search operations 8=waiting for DSC signature

11.	search.allDoc	String		Y	1	This will fetch all the documents which are pending to be signed in the user’s account. RPs need to pass this value as 1 to fetch the documents associated to that account for the user.

Sample Request :

```
{
    "limit": "10",
    "offset": 1,
    "sort": {
        "by": "1",
        "order": "2"
    },
    "search": {
        "groupName": "sam",
        "groupIds": [“c0934”],
        "status": "1",
        "filter": 8,
        "allDoc": "1"
    }
}
```

5.2.7.10.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	message	Message from API

Output2: Data is present.

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	groups	Object[]				
2.	group.groupId	String	50	Y	12345	Selected group id.
3.	group.name	String	200	Y	RentalAgg	Name of group
4.	group.ownerId	String	200	Y	OwnerId for user	Owner of the document
5.	group.creatorId	String	200	Y	CreaterId for user	CreaterId of the document
6.	group.cdate	String		Y	yyyy-mm-dd HH:MM:SS	Document added/created date.

7.	group.status	Int[]	2	Y	1/2/3	Shows the signature status for group. 8-Waiting for my Signature 2-Waiting for Participant Signature 3-Signed 4-Waiting for all Signature 5-Previous signatory not signed document.waiting for their sign 7-Declined by participant 6-Declined by me 9-Signed By Me
8.	group.partCt	Int	2	Y	5	Contains participant count for given group include viewer and signer.
9.	group.access	Int	2	Y	1 /2	Participant Access to the document. 1 – Signer 2 - Viewer
10.	group.grpStatus	Int	2	Y	1/2/3	Status of the group/bundle.
11.	group.sgndCt	Int	2	Y	2	Count of participants who have signed.
12.	group.sgnrCt	Int	2	Y	2	Count of participants who are signer for the document.
13.	group. vwrCt	Int	2	Y	2	Count of participants who are viewer for the document.
14.	group. decCt	Int	2	Y	2	Count of participants who decline the document.
15.	group.asslvl	Int	2	Y	3	Asslvl for signatory
16.	group.authHandleId	String	200	Y		AuthHandleId of signatory
17.	group.accountId	String	200	Y		AccountId for Signatory
18.	group.lastSignedTime	String	200	Y		LastSignedDate timestamp

5.2.8 Document – Status Check

API takes care of giving the status of Document. Can be used for polling API for further actions by RP.

5.2.8.1 Details

- A valid session needs to be there for the API consumption, Session creation can be done using JioSign portal.
- API return groupId and current status.

5.2.8.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/data/status/{groupId}	Refer {FQDN} in EndPoint section above.	application/json

5.2.8.3 Input

#	Request Path Parameter	Type	Length	Mandatory	Sample	Description
1.	{groupId}	String	50	Y	1123-3223-4333-2344	document envelop id/groupId for which status needs to be checked.

5.2.8.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	message	Message from API

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	Y	1234	Unique document envelop (groupId) sent in request.
2.	status	Number	2	Y	1/2/3/4	Status of Document Envelop (Group). Check the API Constant for Group Status values and description.

5.2.9 Document - Get Data

API returns document information based on groupId sent as parameter.

5.2.9.1 Details

- Valid session token needs to be sent in request header, else API will return error.
- groupId needs to be passed as request parameter.
- Provides document and related information in the response.

5.2.9.2 API Endpoints

Http Method	URL	Note	Output Format

GET	https://{FQDN}/docmgmt/v1.1/document?groupId=<data>	Refer {FQDN} in EndPoint section above.	application/json

5.2.9.3 Input

#	Url Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	Y	1234	document envelop id/groupId for which data needs to be returned.

5.2.9.4 Output

Output1: In-case of any error or not data Scenario

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request.
2.	message	String	50	Y	message	Message describing the reason of failure.

Output2: Data is present

#	Parameter	Type	Length	Mandatory	Sample	Description
1.		Object				
2.	groupId	String	50	Y	1234	document envelop id/groupId of selected group.
3.	groupName	String	200	Y	RentalAgg	Name of group
4.	orgId	String	50	N	128	Organization Id associated with group
5.	createTime	String		Y	yyyy-mm-dd HH:MM:SS	Document added/created timestamp.
6.	updateTime	String		Y	yyyy-mm-dd HH:MM:SS	Document updated timestamp.
7.	deadline	String		Y	yyyy-mm-dd HH:MM:SS	Document deadline.
8.	accessibility	Int	2	Y		Will always be 0.
9.	lock	Int		N		Locked status of the document.
10.	lockedBy	String	30	N		Lockedby user Id.
11.	status	Int	2	Y	1/2/3	Status of the document.
12.	message	String	500	N	Message	User entered message to be used in notification mail.
13.	ownerId	String	50	Y	123-123--3123	Owner user Id.
14.	creatorId	String	50	Y	12-123-23-12	Creator User Id
15.	docs	Object[]	NA	Y	NA	Main object which holds list of files. Attributes are mentioned below.
16.	docs.documentId	String	50	Y	12343	Document Id of document.
17.	docs.documentName	String	200	Y	Rental	Name of the document

18.	docs.documentMimeType	String	100	Y	Pdf	Type of the document
19.	docs.documentSize	String	10	Y	18681	Size of the document In KB
20.	docs.createdTimestamp	String	10	Y	25-03-2020 HH:MM:SS	Document Created date.
21.	docs.docType	Int	2	N	2	docType=2 will be for Supporting documents.

5.2.10 Document - Get Participants Status

This api will return list of participants along with their signing status for the document. RPs can use this API to poll and check if any of the participants has signed the document.

5.2.10.1 Details

- RPs should have valid session and access to the document/group.
- API starts only if valid session and access is present, otherwise error will be sent.
- API will return participant status along with their signing status.
- This API return all the signatory regardless of their signing status.

5.2.10.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/participants/status?groupId={grpId}	Refer {FQDN} in EndPoint section above.	application/json

5.2.10.3 Input

#	Request Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	Y	1234-2091-2912-2938	document envelop id/groupId for which participants status are to be fetched

5.2.10.4 Output

Output1: In case of any error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	message	Message from API

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
		Object[]				
1.	groupId	String	50	Y	12111	Document Envelop Id (GroupId).

2.	userId	String	50	Y	JID-	User Id who is participant in Document.
3.	identifier	String	320	Y	email@email.com	Email or phone number of the participant on which invitation was sent.
4.	identifierType	Int	5	Y	1/2	Participants Identifier Type. Check API Constant for Identifier Type details.
5.	assuranceLevel	Int	2	Y	1	Assurance level of record. Check API Constant for Assurance Level details.
6.	access	Int	2	Y	1,2	Access of the participant in document signing request.
7.	accessId	String	50	Y	1235-1929-1920-9291	Unique Id for given participant for given document.
8.	status	Int	2	Y	1,2,3,4,5	Status of the group access for participants. This field talks about the participant status in document signing. Check API Constant for Sign Status.
9.	signOrder	Int		N	1,2,3,4	If signing order is defined, then this field gives the order value for participant.
10.	message	String	200	CM	Comment to be passed	Comments while signing / declining. This field will be in response only if comments are added.
11.	authHandleId	String	50	Y	1e1e-12e-12e	Auth Handle Id, specific to each Participant. Unique Id for each participant.
12.	accountId	String	50	Y	1e1e-12e-12e	Business Account Id which is associated to participant.
13.	createTime	String	15	Y	yyyy-mm-dd HH:MM:SS	Created time stamp
14.	updateTime	String	15	N	2022-05-06 11:56:06	Updated time stamp
15.	randomUuid	String	50	Y	K3l3SDA1M0poUUMzK1d0eFZZd0dmbW1IdXZrPQ	Random unique identification string for a participant and will vary for each document

5.2.11 Document - Get Card Details

API returns the document cards detail.

5.2.11.1 Details

- Valid session token needs to be sent in request header, else API will return error.
- This api will provide the cards detail of all the participants in a document.

5.2.11.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/cards	Refer {FQDN} in EndPoint section above.	application/json

5.2.11.3 Input

#	URL Parameter	Type	Length	Mandatory	Sample	Description
1.	accessIds	String[]		CM		Accepts multiple value comma separated. Maximum of 36 id in one call.
2.	groupId	String	50	M		document envelop id/groupId for which card details need to be fetched

5.2.11.4 Output

Output1: In-case of any error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request.
2.	message	String	50	Y	message	Message from API

Output1: Success

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.		Object[]				
2.	cardId	String	50	Y	920902	Primary Key
3.	accessId	String	50	Y	1122-2120112	Access Id/Participant to which card information is associated.
4.	documentId	String	50	Y	812899-22-2-22	Document Id on which cards is placed.
5.	totalPage	Int		Y	1	Total number of pages in the document.
6.	cardX	Float		Y	123.2	X coordinate of the card on document.
7.	cardY	Float		Y	12.1	Y coordinate of the card on document.
8.	cardH	Float		Y	23	Height of card.
9.	cardW	Float		Y	2	Width of card
10.	Unit	String	50	N	px	Unit of x, y, w, h values.

11.	cardColor	String	50	N	#FRSRWY	Color of the card which was given for participant.
12.	cardType	String	50	Y	Initial/Signature	Type of card which is placed in UI. Check API Constant for details.
13.	cardAngle	String	50	N	0	Angle of the card if user has rotated. Currently this is always 0 as JioSign does not support card rotation.
14.	cardPageNo	Int		Y	1,2	Page on which card has been placed.
15.	cardOnPage	Int		Y	1,2	If card need to be added on all page or only cardPageNo.

Output3: No Data

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	50	Y	message	Message from API

5.2.12 Document - Delete Participant

This api will delete the participant for a given groupId and accessId. AccessId for the participant can be retrieved from *Get Participant Status* API.

5.2.12.1 Details

- RPs should have valid session and access to the document/group.
- API will check for below validations:
    1. Participant to be deleted should not have signed the document.
    2. Participant to be deleted cannot be Document Owner or Document Creator.
    3. Document should not have been finalized.

5.2.12.2 API Endpoints

Http Method	URL	Note	Output Format
DELETE	https://{FQDN}/docmgmt/v1.1/document/participant?groupId={grpId}&accessId={accessId}	Refer {FQDN} in EndPoint section above.	application/json

5.2.12.3 Input

#	Request Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	Y	1234-2091-2912-2938	document envelop id/groupId for which participants details are to be fetched

2.	accessId	String	50	Y	1234-2091-2912-2938	Unique Access Id for given participant for given document

5.2.12.4 Output

Output1: In case of any error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	message	Message from API

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	50	Y	Participant deleted successfully.	Message from API

5.2.13 Document - Finalization Workflow

Document Finalization workflow steps is described below.

Workflow Steps	API Name	Description
Step 1	Finalize Initiate	API initiates document finalization. Click on link to jump to the section.
Step 2	Finalize Initiate Status	API tells the status of Initiate Call.

Important Points:

- A valid RP session needs to be there for the API consumption.
- Workflow sequence should be followed otherwise API will fail.

5.2.13.1 Document – Finalization Initiate

5.2.13.1.1 Details

- API will initiate document finalization.
- API will have below validations
  1. The document should not have been finalized.
  2. There should not be any pending participants to sign the document.
  3. User who is finalizing the document should have proper access.
- At this step data validation occurs on the document which is to be finalized. API might return data validation errors at this stage.
- After receiving response from this API call, RPs needs to poll and check finalization status using Step2 (Document Finalization Status).

5.2.13.1.2 API Endpoints

Http Method	URL	Note	Output Format

PUT	https://{FQDN}/docmgmt/v1.1/document/finalization	Refer {FQDN} in EndPoint section above.	application/json

5.2.13.1.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	CM	1234-2091-2912-2938	Document Envelop Id (groupId) which RPs wants to finalize.

5.2.13.1.4 Output

Output1: Errors

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	status	String	20	Y	INITIATED	Status of Document Finalization Initiated Status. Check details section of API for different status.
2.	action-token	String	NA	Y	Action Token	Temporary action token issued document finalization. To be passed in finalization workflow status api call.

5.2.13.2 Document – Finalization Status

5.2.13.2.1 Details

- Polling API to know the status of document finalization.
- RPs should stop polling once they receive “**COMPLETE**” status.
- API returns 200 OK, with below status value in response. RPs need to handle status value to complete the entire process.

Status Value	Description
INITIATED	• Document Finalizing is initiated. • RPs needs to continue Polling to know status.
RECEIVED	• Data sanitization has been completed. • RPs needs to continue Polling to know status.
RUNNING	• Document finalization is happening. • RPs needs to continue Polling to know status.

SIGNED	• Document is digitally signed, but still process is not complete, final steps are being done by API. • RPs needs to continue Polling to know final status.
COMPLETE	• Process is completed. • RPs stops poll after this status it gets in the response. • Check "**message**" attribute, which will have "**ok**" or "**error/error message**" value. If the value is not "ok", it means that API is not successful. • **ok**: Means signing/decline is Success. • **error**: Means signing/decline is failed. RPs needs to show error message to the user and ask them to reinitiate the process. • **RPs needs to start the process from the beginning if "Error" status is received during polling API call**.

5.2.13.2.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/finalization/status	Refer {FQDN} in EndPoint section above.	application/json

5.2.13.2.3 Input

No Parameters.

5.2.13.2.4 Output

Output1: Errors

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	20	Y	Error OK	Message from API. If status value is COMPLETE and message is OK, means signing is successful.
2.	status	String	20	Y	FINALIZING COMPLETE	Status of Finalization Request Check the details section of API for more information.

5.2.14 Document – Delete Document

API takes care of deleting of group/document.

5.2.14.1 Details

- A valid session needs to be there for the API consumption.
- API will delete group/document in JioSign system.
- Only document owner can delete the document.
- Signed or finalized documents can also be deleted.

5.2.14.2 API Endpoints

Http Method	URL	Note	Output Format
DELETE	https://{FQDN}/docmgmt/v1.1/document/data/{groupId}	Refer {FQDN} in EndPoint section above.	application/json

5.2.14.3 Input

#	URL Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	200	Y	12312-123-12312-321	document envelop id/groupId which needs to be deleted.

5.2.14.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	50	Y	Document deleted successfully.	Message from API

5.2.15 Document - Get Original File

API returns the original uploaded file.

5.2.15.1 Details

- API will download the uploaded files in JioSign system and sends back binary stream content.
- API checks access of the user before, incase of no access API sends error back.
- API can download supporting documents uploaded by owner as reference for the signing document.

5.2.15.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/original/file?groupId=<val>&docId=<val>	Refer {FQDN} in EndPoint section above.	application/pdf -in case of file gets downloaded.

			application/json – In case of Error.

5.2.15.3 Input

#	Url Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	Y	1111	Selected document envelop id/groupId.
2.	docId	String	50	Y	1222	Selected document id.

5.2.15.4 Output

Output1: In-case of any error or not data Scenario

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request.
2.	message	String	50	Y	message	Message describing the reason of failure.

Output2: Data is present

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	fileData	Binary		Y		File content

5.2.16 Document - Get Signed File

API takes care of downloading the file which is signed by at least one participant.

5.2.16.1 Details

- API will validate proper access for the user, otherwise error will be returned.
- Ensure valid session token is shared in header.
- If no participant has signed the document, then API will return 404.

5.2.16.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/signed/file?groupId=<val>	Refer {FQDN} in EndPoint section above.	application/pdf -in case of file gets downloaded. application/json – In case of Error.

5.2.16.3 Input

#	URL Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	Y	1233	Selected document envelop id/groupId.

5.2.16.4 Output

Output1: In-case of any error or not data Scenario

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request.
2.	message	String	50	Y	Message	Message describing the reason of failure.

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	filedata	binary		Y	NA	File content of signed document.

5.2.17 Document – Get Audit Trails

API provides audit trail information for the given group/document.

5.2.17.1 Details

- A valid session needs to be there for the API consumption.
- Limited eventIds can be queried, check API Constant section for allowed event ids.

5.2.17.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/audits?groupId=<value>&evntIds=<val>	Refer {FQDN} in EndPoint section above.	application/json

5.2.17.3 Input

#	URL Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	200	Y	123-123-123-123	document envelop id/groupId for which audit event needs to retrieve.
2.	evntIds	String		Y	Comma separated event numbers	Events which needs to be queried. Allowed values are mentioned in API Constant section.

5.2.17.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	Message	Message from API

Output2: Data is present

#	Parameter	Type	Length	Mandatory	Sample	Description
1.		Object				
2.	groupId	String	50	Y	11111-123-123-123	document envelop id/groupId of selected group.
3.	groupName	String	50	Y	Name of document	Name of group
4.	trails	Object[]	NA	Y	NA	Main group which holds list of audit trails.
5.	trails.userId	String	30	Y		User Id who has taken the action.
6.	trails.userName	String	50	N		If username is not there, then it needs to be phone number of the user.
7.	trails.identifier	String	320	N		User Email Id
8.	trails.groupEventId	String	50	Y	1728-2910-1928-1929	Trail Id
9.	trails.eventId	int	2	Y	1,2,3,4…	Action value taken by user. Check below for details.
10.	trails.eventDesc	String	100	Y	Document Created	Action value taken by user, the value will be transformed before sending the data to UI.
11.	trails.event_timestamp	String	15	Y	yyyy-mm-dd hh:mm	Event Created date.
12.	trails.documentId	String	50	N	12e1-12-e12e-12e	Document Id for associated file. Can be null.
13.	trails.accountId	String	50	Y	e32e-2er-223-e3	Users Account Id.
14.	trails.authHandleId	String	50	Y	223-e23-e23-e23	User Login Id.
15.	trails.meta	Object {}				Stores data related to action taken by user, it is dynamic structure. e.g. "ipAddress": "12.22.32.22", "inputTxn": "dff5699a-fc96-419d-be82-dbaed79dc0bf", "signId": "1d9b2a94-4215-43af-9e5e-e2b218af7ee8", “so”:”123133”, "sigIds": ["0af438a8-7a7a-1ae6-817b-3b2d9f480005", "0af4388a-797a-1f12-8179-93a579d10010"]

5.2.18 Document - Get Audit Trail File

API takes care of downloading the audit trail pdf file.

5.2.18.1 Details

- API will validate proper access for the user, otherwise error will be returned.
- Ensure valid session token is shared in header.

5.2.18.2 API Endpoints

Http Method	URL	Note	Output Format
GET	https://{FQDN}/docmgmt/v1.1/document/audit/file?groupId=<val>	Refer {FQDN} in EndPoint section above.	application/pdf -in case of file gets downloaded. application/json – In case of Error.

5.2.18.3 Input

#	URL Parameter	Type	Length	Mandatory	Sample	Description
1.	groupId	String	50	Y	1233	Selected document envelop id/groupId.

5.2.18.4 Output

Output1: In-case of any error or not data Scenario

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request.
2.	message	String	50	Y	message	Message describing the reason of failure.

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	filedata	binary		Y	NA	File content for audit events for given groupId.

5.2.19 Document – Save CallBack Url

API registers callback url for an envelop (groupId). RPs need to register a web service (callback listener) that is available on the public internet.

RPs will be notified on this callback url for any event updates happening in the corresponding groupId. JS will send an HTTPS GET call on the registered callback url. RPs can then check the status api to know the group status updates.

Refer Appendix D for the sequence diagram of callback workflow.

5.2.19.1 Details

- A valid session needs to be there for the API consumption.

- API returns success message along with groupId in response.
- API will register callback url for envelop(groupid) in JioSign system.
- Document owner and document creator are allowed to register the callback url.
- Once the document has been finalized, RPs cannot register the callback url for corresponding groupId.
- If any participant signs or declines the document, JioSign will notify these events to RPs on the registered callback url after which RPs can check the envelop status updates from document status and participants status check api.
- If the registered callback url is returning an error or is not responding, JS system will retry to notify on the registered callback url.
- Retry mechanism: If message delivery for a given envelop fails, retries will be attempted 3 times in the following sequence:

  5 minutes later

  15 minutes later

  30 minutes

5.2.19.2 API Endpoints

Http Method	URL	Note	Output Format
POST	https://{FQDN}/docmgmt/v1.1/callback	Refer {FQDN} in EndPoint section above.	application/json

5.2.19.3 Input

#	Request Body Parameter	Type	Length	Mandatory	Sample	Description
1.	id	String	50	Y	Fmwejin-323ncslnck2-32m3k	GroupId or the suiteId for which the callback url is being registered.
2.	type	Int	2	Y	1/ 2	Type refers to the type of Id for which callback url is being registered. type=1 is for registering callback url for groupid (envelop id). type=2 is for registering callback url for suiteid (Bulk Sign Workflow).
3.	url	String	500	Y	https://abc.com/	Callback url to be registered in JioSign system. Make sure to provide different url for registering different id since RP will be notified on this url for any changes in the groupid/suiteid.

5.2.19.4 Output

Output1: Error

#	Parameter	Type	Length	Mandatory	Sample	Description
	Array[]					
1.	errcode	String	11	Y	JDSH-LO-100	Error code of request. Check error code for structure details.
2.	message	String	50	Y	message	Message from API

Output2: Success

#	Parameter	Type	Length	Mandatory	Sample	Description
1.	message	String	50	Y	Success	Success message from API
2.	groupId	String	50	Y	1234	Unique groupId for the document.

5.3 Error Code and Description

Check the **JioSign Error codes.pdf** file shared.

Error Codes are divided into two sections.

Section1 → Gateway Errors: These errors will be returned while consuming system has issues between their system and Gateway layer.

- Expired token / wrong credential / Missing mandatory parameters.

Section2 → JioSign API Errors: These errors will be returned during the Document Management or Session Management API consumptions. Few examples of these error are:

- Refreshing expired token
- Missing out mandatory API input parameters
- Consuming JioSign API with invalid/expired token

5.4 Appendix

5.4.1 Appendix A

PDF Page Coordinates (page size, placement, etc.)

Important Note: This information's are only for reference. These portals and document are not managed by JioSign Team. Also, RPs need to consider scenarios for different page size, and different rotations (horizontal/vertical) of pdf files. If the PDF structure is dynamic, then identifying coordinate statically as shown below will not be helpful. RPs need to handle this scenario programmatically.

5.4.1.1 Reference 1:

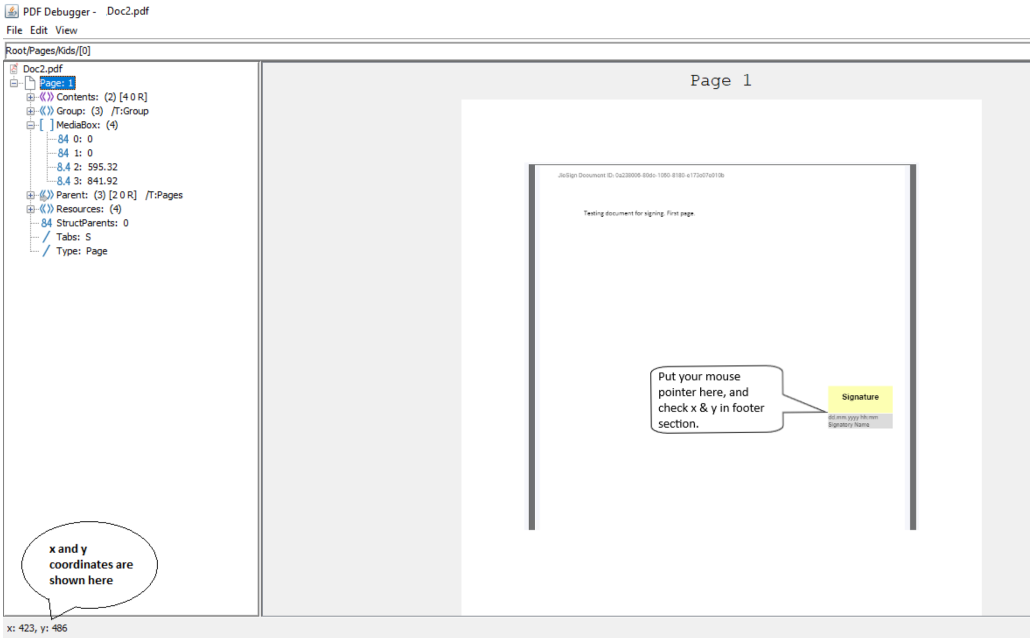
Identifying coordinates using PDF Box jar tool. Apache PDFBox PDFDebugger 3.0.* displays PDF coordinates in the status bar.

Download it from here: https://pdfbox.apache.org/download.cgi

Download the pdfbox-app-3.0.*.jar available under command line tools on above link. Then run the below command with the required file. “**java -jar pdfbox-app-3.0.0-alpha3.jar debug "Doc2.pdf"**”.

You would be able to see the coordinates in footer section by hovering the mouse on pdf page. You can select a particular page from left hand side and corresponding page will be displayed on right.

Note: pdf box displays coordinates from lower left of the page so if you want to extract some text using these coordinates you need to subtract the y axis from the total height and then use it. In case of below example, you will have to use x:47 y:(841.92-486) =355.92



Root/Pages/Kids/[0]

```
Doc2.pdf
  Page: 1
    Contents:  (2) [4 0 R]
    Group:  (3)  /T:Group
    MediaBox:  (4)
      0:  0
      1:  0
      2:  595.32
      3:  841.92
    Parent:  (3) [2 0 R]   /T:Pages
    Resources:  (4)
    StructParents:  0
    Tabs:  S
    Type:  Page
```

Put your mouse pointer here, and check x & y in footer section.

x and y coordinates are shown here

x: 423, y: 486

5.4.1.2 Reference 2:

URL: https://www.pdfscripting.com/public/PDF-Page-Coordinates.cfm

5.4.1.3 Reference 3:

URL: https://www.verypdf.com/wordpress/201308/how-to-get-the-x-and-y-coordinates-of-a-point-in-a-pdf%EF%BC%9F-38274.html

5.4.1.4 Reference 4:

URL: https://coderwall.com/p/3h8nog/view-real-time-pdf-coordinates

5.4.1.5 Reference 5:

URL: http://www.quickpdflibrary.com/faq/display-x-and-y-coordinates-in-adobe-reader.php

5.4.2 Appendix B

This section talks about how to create Self signed certificate. This information’s are only for reference.

Note: Self Signed Certificate is not recommended for Production System. For production RPs should procure Authorized CA signed certificate.

5.4.2.1 Reference 1:

Creating Certificate using openssl. The value used in below command is for reference only, should be changed based on RPs.

Follow below steps to create a Self Signed Certificate.

- Create a new file rpname.cnf, with below content. Here we are defining details to be used in certificate.

```
[req]
default_bits = 2048
prompt = no
encrypt_key = no
default_md = sha256
distinguished_name = dn
req_extensions = req_ext
[dn]
CN = www.company.com
emailAddress = yourmailid
O = MyCompany
OU = MyDivision
L = SomeCity
ST = MH
C = IN
[req_ext]
subjectAltName = @alt_names
[alt_names]
DNS.1 = www.company.net
DNS.2 = company.com
DNS.3 = company.net
```

- Execute the below command for creating self-signed certificate using data from above file.

Create Public and Private certificate run below command
`openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout key.pem -out public_cert.pem -config rpname.cnf`
Verify Public certificate run below command
`openssl x509 -in public_cert.pem -noout -text`
key.pem - This is the private key which needs to be kept secret and should not be shared with anyone. public_cert.pem - This is the public certificate which needs to be shared with JioSign Team. 730 - validity in days, can be decided by RPs

5.4.2.2 Reference 2:

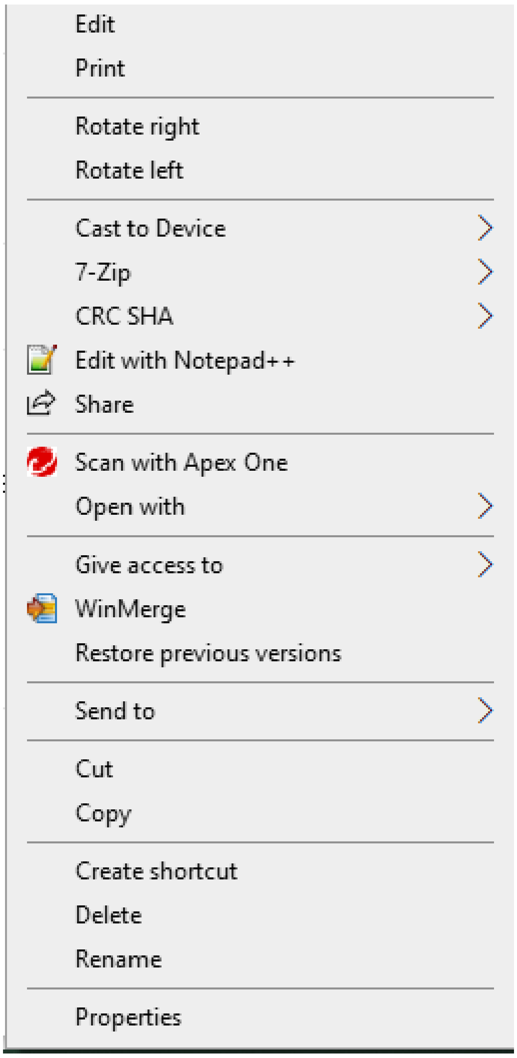
Creating Self Signed Certificate in Windows use following guide present in URL.

https://medium.com/the-new-control-plane/generating-self-signed-certificates-on-windows-7812a600c2d8

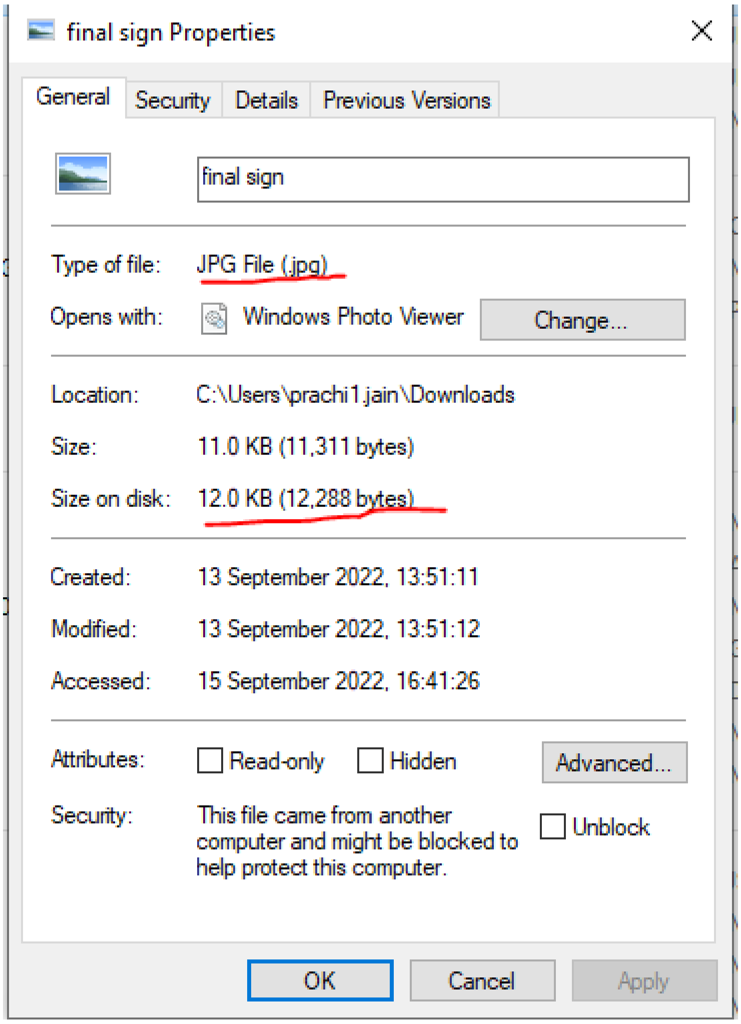
5.4.3 Appendix C

Finding Image size, format and dimensions.

1) Right click on the image and click on properties as should below.



2) The image format and size will be mentioned as shown below.



final sign

Type of file: JPG File (.jpg)

Opens with: Windows Photo Viewer

Location: C:\Users\prachi1.jain\Downloads

Size: 11.0 KB (11,311 bytes)

Size on disk: 12.0 KB (12,288 bytes)

Created: 13 September 2022, 13:51:11

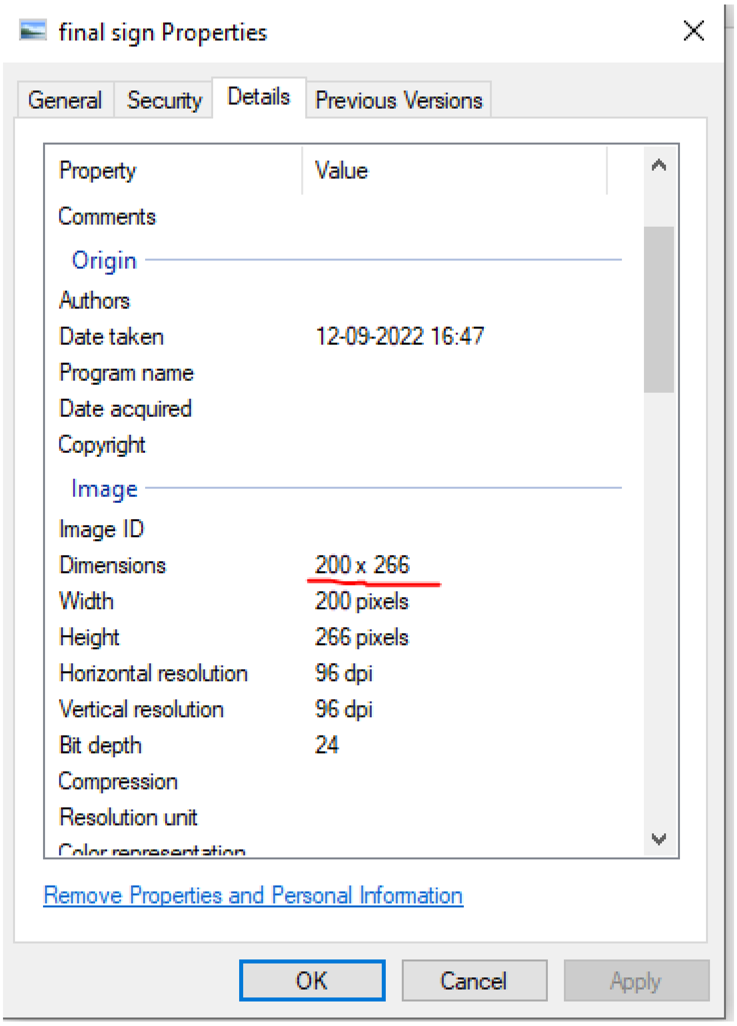
Modified: 13 September 2022, 13:51:12

Accessed: 15 September 2022, 16:41:26

Attributes: □ Read-only □ Hidden

Security: This file came from another computer and might be blocked to help protect this computer. □ Unblock

3) Click on the details tab to view the image dimensions.

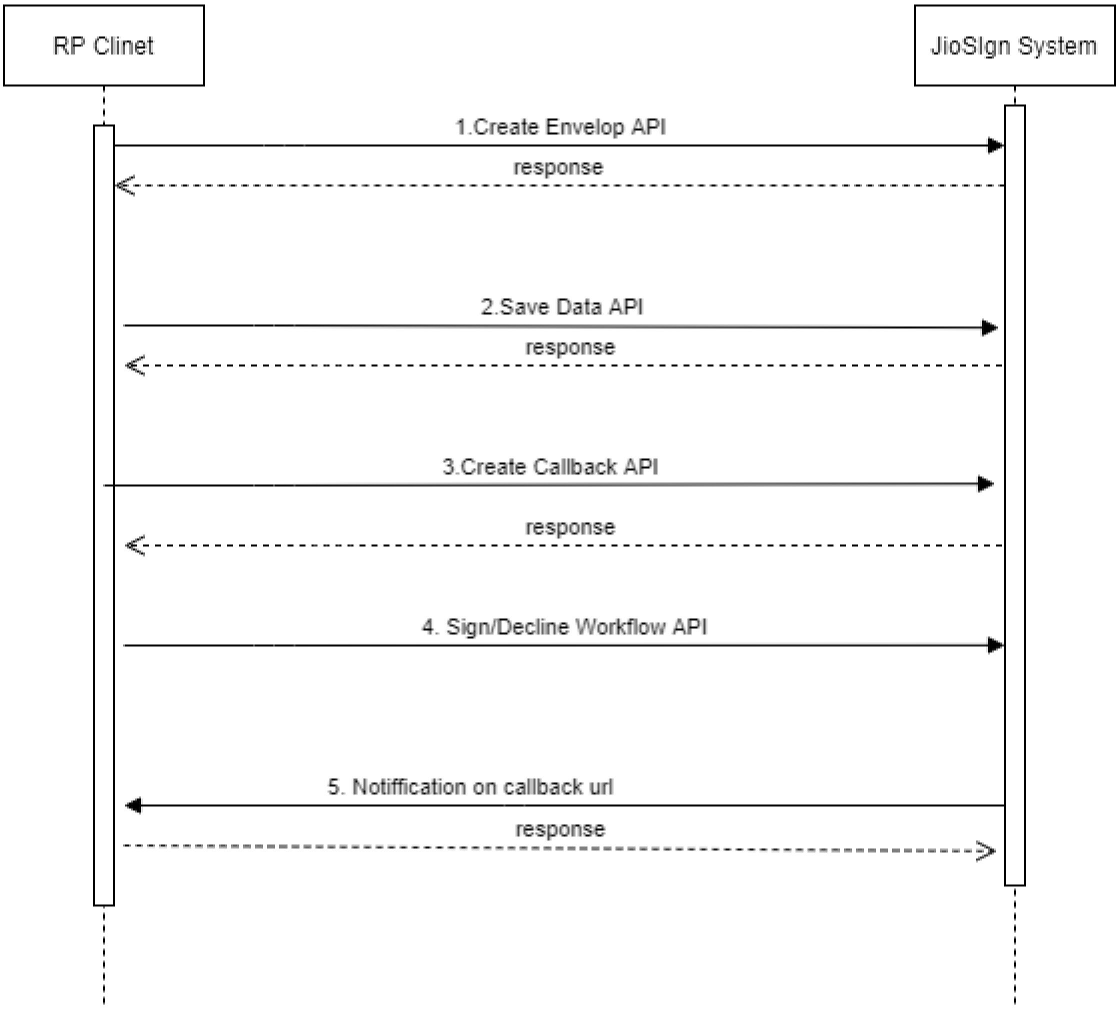


final sign Properties

Property	Value
Comments	
Origin	
Authors	
Date taken	12-09-2022 16:47
Program name	
Date acquired	
Copyright	
Image	
Image ID	
Dimensions	200 x 266
Width	200 pixels
Height	266 pixels
Horizontal resolution	96 dpi
Vertical resolution	96 dpi
Bit depth	24
Compression	
Resolution unit	
Color representation	

5.4.4 Appendix D

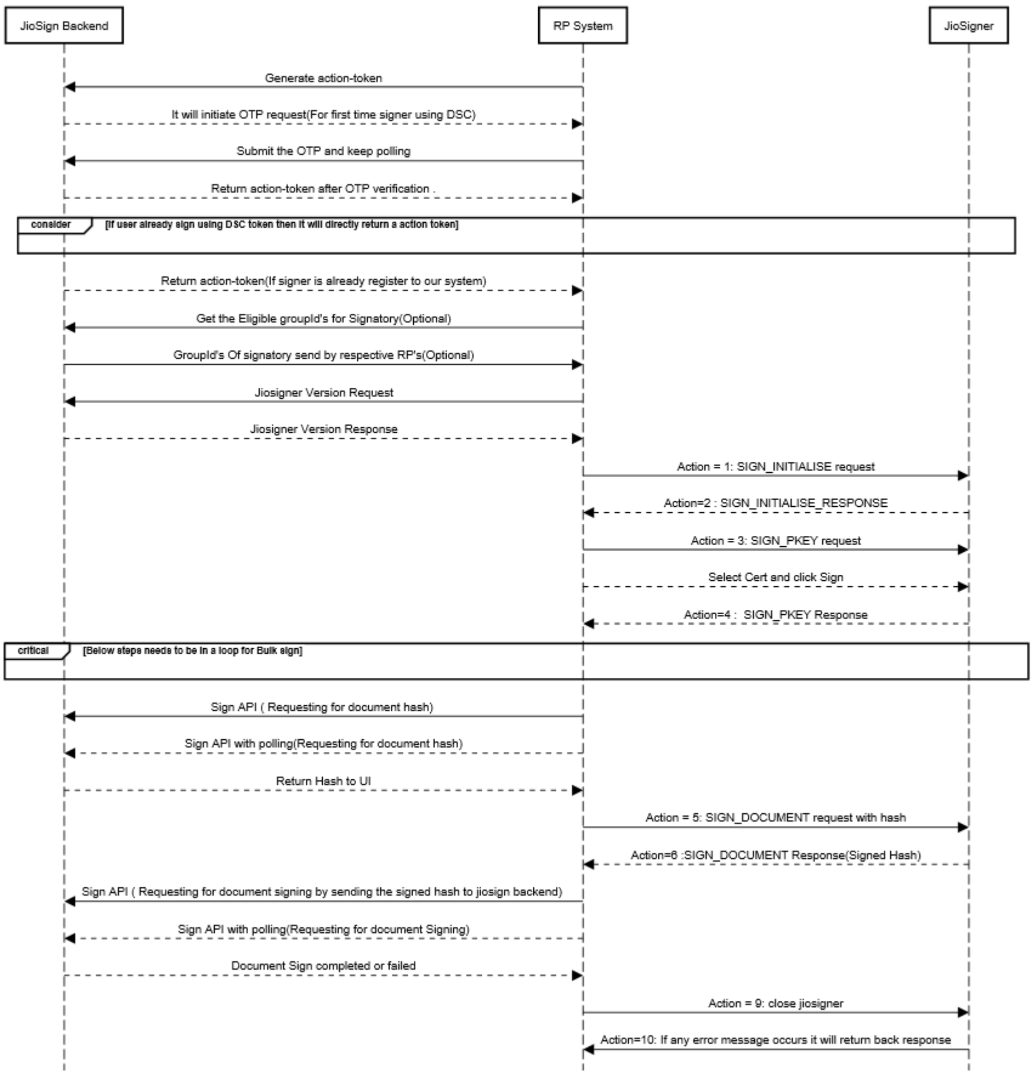
Sequence Diagram of callback workflow.



5.4.5 Appendix E

Sequence Diagram of Bulk DSC Sign workflow.

Bulk DSC Sign Flow


JioSign Backend
RP System
JioSigner
Generate action-token
It will initiate OTP request(For first time signer using DSC)
Submit the OTP and keep polling
Return action-token after OTP verification .
consider
[If user already sign using DSC token then it will directly return a action token]
Return action-token(If signer is already register to our system)
Get the Eligible groupId's for Signatory(Optional)
GroupId's Of signatory send by respective RP's(Optional)
Jiosigner Version Request
Jiosigner Version Response
Action = 1: SIGN_INITIALISE request
Action=2 : SIGN_INITIALISE_RESPONSE
Action = 3: SIGN_PKEY request
Select Cert and click Sign
Action=4 : SIGN_PKEY Response
critical
[Below steps needs to be in a loop for Bulk sign]
Sign API (Requesting for document hash)
Sign API with polling(Requesting for document hash)
Return Hash to UI
Action = 5: SIGN_DOCUMENT request with hash
Action=6 :SIGN_DOCUMENT Response(Signed Hash)
Sign API (Requesting for document signing by sending the signed hash to jiosign backend)
Sign API with polling(Requesting for document Signing)
Document Sign completed or failed
Action = 9: close jiosigner
Action=10: If any error message occurs it will return back response