eKYC With Consent(2.1)
(Documentation)

1. Overview

- Our page containing the purpose for which eKYC is done will be shown to the aadhaar holder. This ensures that the resident is aware of the use case of his eKYC data.
- According to UIDAI's guidelines, explicit consent has to be taken by KUA to perform eKYC and share data with Sub AUA for the purpose specified.
- Purpose of each transaction has to be communicated during eKYC request initiation by Sub AUA.
- We recommend to integrate the app in preprod environment first and then move to production later by changing our URLs.

2. Process Flow

a. Sub AUA initiates transaction (see eKYC Request Initiation section).
b. If request is valid you will be redirected to Aadhar-Bridge page
    i. When modality is "otp"
        1. Aadhaar resident will generate OTP by clicking on the Generate Otp button.
        2. If OTP is generated successfully, OTP input box will appear.
        3. After entering otp, aadhaar resident has to give his/her consent for performing and sharing the kyc.
    ii. When the modality is "biometric"
        1. Aadhaar resident has to give his/her consent for performing and sharing the kyc.
c. Purpose of performing eKYC will be shown clearly on the same page.
d. On click of submit button, you will be redirected to your successURL or failureURL with status of transaction.
e. On receiving response on your Url, you should immediately trigger eKYC fetch request (See Fetch eKYC data section).

3. eKYC Request Initiation

URL (Method: POST)

- Preprod: *To Be Declared*
- Prod: *To Be Declared*

REQUEST:

- **Headers:**

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

- **Parameters:**

name	value	comments	example
purpose	String, Mandatory	This will appear on our page.	Bank account opening
aadhaarId	String,length=12, Mandatory	12 digit aadhaar number.	999999999999
modality	"otp" or "biometric", Mandatory	Only these both modalities are valid	otp/biometric
channel	"SMS" or "EMAIL" or "BOTH". Mandatory if modality="otp"	Where you want to receive otp from UIDAI.	SMS
requestId	String, Mandatory	Unique transaction identifier for the Sub AUA. This should not be same as aadhaarId since aadhaar number can not be used as primary key for any purpose.	123456789101112
saCode	String, Mandatory	Your Sub AUA code. You can find this in your user portal.	a1b2c3
successUrl	String, Mandatory	This should be a valid url. We will redirect the response on this URL if it was a	https://mydomain.com/success.html

		success.	
failureUrl	String, Mandatory	This should be a valid url. We will redirect response on this URL if it was a failure. This can be same as successUrl.	https://mydomain.com/failure.html
auth-capture-data	Json, Mandatory if modality is biometric	This should be valid json request.	See "**JSON REQUEST FOR BIOMETRIC KYC**" **section**
hash	String, Mandatory	This should be a SHA-256 value of Hash Sequence. See hash generation section	9780cd0d2ce77eef8f64942f54e0281a0e220ff6bbcce0a03df27a2b15575f58

SAMPLE HTML CODE:

```
<form method="post" action="<Url To Be Declared>">
   <input type="hidden" name="saCode" value="<your sa code>" >
   <input type="hidden" name="aadhaarId" value="<aadhaar id>" >
   <input type="hidden" name="requestId" value="<unique request id>" >
   <input type="hidden" name="purpose" value="<purpose of doing e kyc>" >
   <input type="hidden" name="modality" value="otp" >
   <input type="channel" name="channel" value="BOTH" >
   <input type="hidden" name="successUrl" value="<successUrl>" >
   <input type="hidden" name="failureUrl" value="<failureUrl>"  >
   <input type="hidden" name="hash" value="<sha256 hash value>" >
   <input type="hidden" name="auth-capture-data" value="<json value>" >
   <button type="submit">Proceed to KYC</button>
</form >
```

JSON REQUEST FOR BIOMETRIC KYC (auth-capture-data):

```
{
        "consent": "Y",
        "modality": {
                "demographics": false,
                "fp-image": false,
                "fp-minutae": false/true,
                "iris": false/true,
                "otp": false,
                "pin": false
                },
         "pid": {
                "type": "<xml/proto>",
                "value": "<biometric data encrypted by session key>"
         },
          "aadhaar-id": "<aadhaar number>",
          "hmac": "<hmac value from RD service>",
          "session-key": {
                "cert-id": "<expiry date of public certificate in the format of YYYYMMDD>",
                "value": "<session key encrypted by uidai public certificate>"
          },
          "unique-device-code": "<your device code>",
          "dpId":"<from RD service>",
          "rdsId":"<from RD service>",
          "rdsVer":"<from RD service>",
          "dc":"<from RD service>",
          "mi":"<from RD service>",
          "mc":"<from RD service>"
}
```

Note that: While calling RD service to capture biometrics to perform ekyc, you have to pass “wadh” attribute in input to RD service (please refer corresponding RD service documentation from device provider).

We are using

ver = ”2.1”, ra = ”F” for fingerprint or ”I” for iris, rc = “Y”, lr = ”N”, de = ”N”, pfr = “N”

Then “wadh” is calculated as follow:

wadh=SHA-256(ver+ra+rc+lr+de+pfr)

This step is **not** required in case of OTP based eKYC

Additional Error Codes:

- AB-210: Invalid Aadhaar Number
- AB-211: Plan is not active
- AB-212: Plan is expired
- AB-213: Insufficient Api Count
- AB-214: Failure of the first request of the month (after billing) because balance is lower than the monthly minimum(Ensure that balance in your account must be equal or more than the monthly minimum balance **prior to/on your billing date**)
- AB-215: Insufficient balance to perform kyc

RESPONSE:

- Redirected to your successUrl/failureUrl
- If success, we will redirect to

  <successUrl>?hash=<hash>&uuid=<uuid>&requestId=<requestId>&status=success

  Eg:

  https://mydomain.com/success.html?hash=9780cd0d2ce77eef8f64942f54e0281a0e220ff6bbcce0a03df27a2b15575f58&uuid=33adb2ce-b26b-4cef-a485-1f4ac638fa63&requestId=123456789101112&status=success

- If failure,

  <failureUrl>?hash=<hash>&uuid=<uuid>&requestId=<requestId>&status=failure&err=<error-code>

  Eg:

  https://mydomain.com/failure.html?hash=9780cd0d2ce77eef8f64942f54e0281a0e220ff6bbcce0a03df27a2b15575f58&uuid=33adb2ce-b26b-4cef-a485-1f4ac638fa63&requestId=123456789101112&status=failure&err=K-100

#POINTS TO NOTE:

- Uuid supplied to you is required to fetch eKYC Data (See next section).
- successUrl can be same as failureUrl provided you handle it accordingly.
- Https is recommended for Prod environment.
- Once transaction is initiated, it is valid for 10 minutes during which aadhaar resident has to complete his eKYC.

4. Fetch eKYC data

URL:

- Preprod: *To Be Declared*
- Prod: *To Be Declared*

REQUEST:

This will be json request in the format as specified below :

```
{
    "saCode"    : "<Your Sa Code>",
    "uuid"      : "<You got this in response of kyc request initiation>",
    "requestId" : "<Same as sent in kyc request initiation>",
    "aadhaarId" : "<12 digit aadhaar number,same as sent in kyc request initiation >",
    "hash"      : "<sha256 of hash sequence defined for fetch kyc data>"
}
```

KYC RESPONSE:


```
{"kyc":
        {
        "photo":"photo in base 64 format",
        "poi":
                {
                "name":"<name of aadhaar holder>",
                "dob":"<data of birth>",
                "gender":"<M/F>"},
        "poa":
                {
                "co":"<Father name>",
                "house":"<House number>",
                "lm":"<Land mark if any>",
                "lc":"<Colony name if any>",
                "vtc":"<If any>",
                "subdist":"<Sub District if any>",
                "dist":"<District>",
                "state":"<State Name>",
                "country":"<County Name>",
                "pc":"<Pincode>",
                "po":"<Postoffie Name>"},
        },
"aadhaar-id":"<aadhaar id>",
"success":true,
"aadhaar-reference-code":"3e8922e886e44d62a2317fc84b287e1b",
"hash":"<hash supplied by aadhaar bridge>",
"uuid":"<uuid supplied by aadhaar bridge>",
"requestId":"<your request id>"
}
```

#POINTS TO NOTE:

- Note that Kyc Response is same as earlier with addition of hash, uuid and requestId attributes.
- Once you receive response at your successUrl, you should initiate fetch request immediately. The time window is of 10 minutes between successful eKYC with UIDAI and Sub-AUA fetching kyc data.
- Beyond this time limit we will not store any data.
- eKYC data can be fetched once. Multiple calls will not be entertained.

ERROR CODES:

- AB-201: Malformed Json request
- AB-202: Mandatory attribute missing
- AB-203: Invalid Sub-AUA code
- AB-204: Hash mismatch
- AB-205: No Txn found for your request
- AB-207: Txn has expired
- AB-208: Invalid environment(make sure you are fetching data from environment in which eKYC was performed)
- AB-209: Contact us.

5. Hash generation

- It is essential that we have a definitive protocol to verify all the communication between Aadhar-Bridge and Sub AUA.
- So for every request coming to us, you have to supply a hash which we will use as a first step of verification.
- In return all responses will also contain hash supplied by us. You should not entertain any request at your *successUrl* if hash does not match.
- Hash should be calculated this way:
  hash=SHA256(Hash-Sequence)
- Hash Sequence is specified as follows(no space,no commas,no single/double quotes)
  - For _init request: **<saCode>|<aadhaarId>|<requestId>|<salt>**
  - Response at successUrl/failureUrl : **<salt>|<requestId>|<aadhaarId>|<saCode>**
  - eKYC Fetch Request: **<uuid>|<saCode>|<aadhaarId>|<requestId>|<salt>**
  - eKYC Fetch Response: **<salt>|<requestId>|<aadhaarId>|<saCode>|<uuid>**

  Example(For _init request):
  If your
  saCode=*a1b2c3*,
  requestId=1234567890101112,
  aadhaarId=999999999999,
  salt=e1d2c3b4a,
  then
  Hash-Sequence=a1b2c3|999999999999|1234567890101112|e1d2c3b4a
  hash =SHA-256(Hash-Sequence)

- For validation:
  - Receiving end should calculate hash based on request parameters and match it against received hash.
  - If receivedHash=calculatedHash, then only you should proceed with your application logic.
- Salt is the key parameter here. It is known only to Sub-AUA and Aadhar-Bridge. You can generate your salt under your Account -> Sub-AUA Details.
- You can regenerate your salt anytime you want. Only one salt will be active at any point of time for a Sub-AUA.
- Salt is same for Prod and Preprod environment for now.

6. Changes in this version:

1. modality=biometric added.
2. Init request attribute "auth-capture-data" added(Mandatory if modality=biometric)
3. Init request attribute “channel” added(Mandatory if modality=otp)
4. Support for registered device added (Same as auth).
5. UIDAI stopped returning mobile and email in eKYC Response.