Fayda Platform API Specification

Policy: A Policy is a document in Fayda IDMS which dictates various actions between the partner and Fayda system. Policies for various partners may differ based on various use cases.

BETA VERSION

Authentication Service

Functional Features

Identity verification services

- Yes/no confirms identity claimed
- Works with Virtual ID

Supports multiple levels of confidence and trust

- Single and multi-factor authentication
- Supports OTP, demographic, biometric factors

e-KYC based authentication

- Returns selected non biometric attributes excluding ID
- Policy based sharing of data

Authorization/Consent

- OTP based consent mechanism linked to transaction being authorized or consented to

Partner Ecosystem

- API Key and Policy driven usage

Technical Features

- Trust and security validations are performed on the request
  - Registered Devices, Authorized Partners
- Uses a third party SDK for biometrics comparison
- Call to authentication is a single request, OTP generation is a prior step
- UIN/VID based authentication requests can be made
  - Recommended configuration is VID only
- Easy to plug in validations such as liveness detection
- Supports L0/L1 SBI Specifications for clients: Refer to SBI specification
- Domain feature and transaction feature for inter-op and authorization support

Rules for using FAYDA's Authentication API

- The authentication request should have a defined set of parameters as mentioned in the API specification
- The authentication request should have signature of the request in the header signed by the authentication partner.
- The biometric data should be sent in as a JWT token where the payload base64URL encoded and the signature is signed using the device key. More details of the biometric data block is available in the SBI specification document
- The request should be sent to the authentication server, within a set time period in the configurations (i.e. time period between the current time stamp and the request time stamp is <= time period set in configurations).

Users of Authentication service

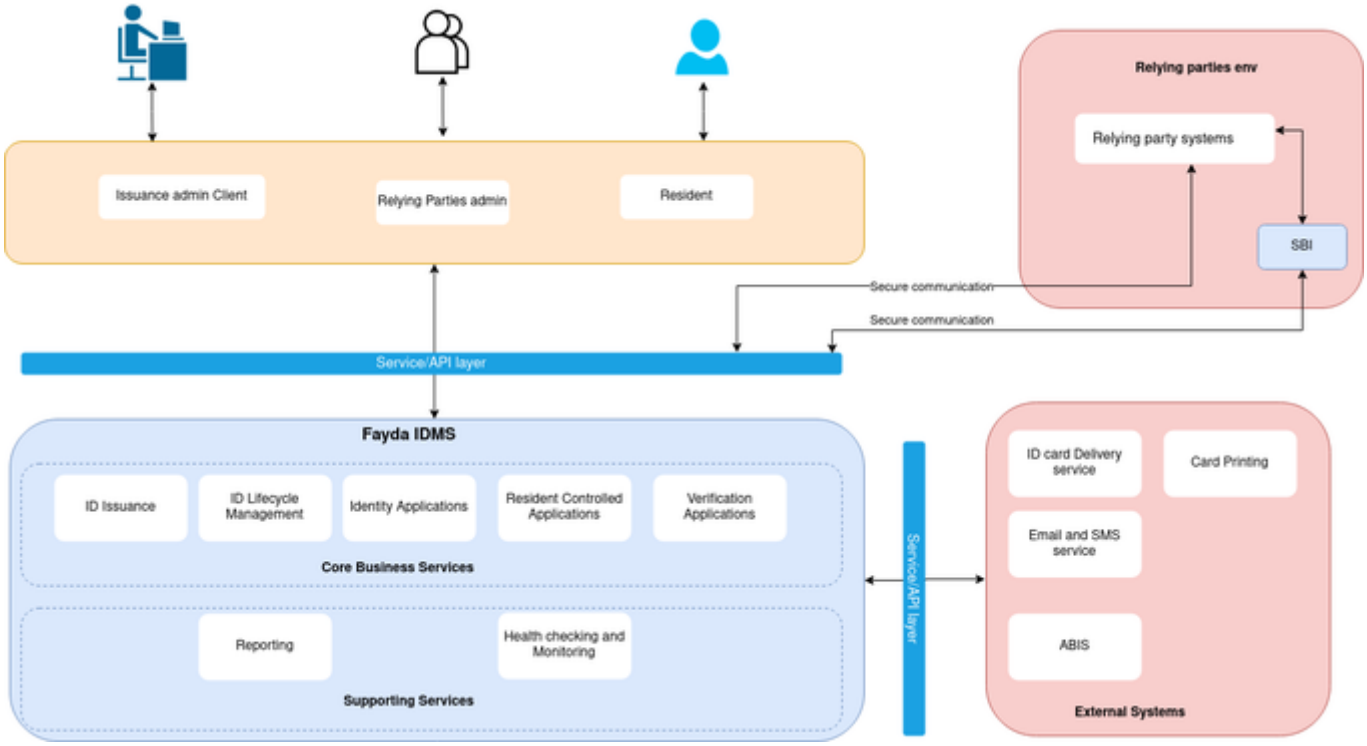
1. **Fayda IDMS as the Infrastructure Service Provider** – under current context, Fayda's role will include providing authentication as a service as well as the required infrastructure and acting as a service source and a gatekeeper for all authentication requests sent to this service. The Fayda IDMS is also responsible for the policy creation so that authentication partners will follow the set policy.
2. **Authentication Partners** - Authentication Partners are relying parties that register themselves with Fayda, under a defined policy. Authentication requests are sent by relying parties directly or through designated authentication partners.

3. **Partner-API-Key** - For a partner to opt for an authentication policy, they have to generate a PartnerAPIKey request with following sample parameters - PartnerCode, UseCaseDescription, SupportingInfo, Status etc. Once the PartnerAPIKey request is approved by Partner Manager, Partner is provided PartnerAPIKey that contains details like - PartnerAPIKey (combination of PartnerCode, policy group and policy), issuedOn, validTill, isActive etc)

Below are various authentication types currently supported by Fayda IDMS service:

- **OTP based** - TOTP (Time based OTP)
  - Using the FAYDA's authentication service a registered authentication partner can request for OTP authentication. Before performing OTP based authentication the Partner needs to request for an OTP using the individual's ID and use it for OTP based authentication.
- **Demographic based**
  - Using the FAYDA's authentication service a registered authentication partner can request for demographic authentication. Currently, we support demographic authentication for the following id attributes - Name, DOB, Age, Gender, and FullAddress
- **Biometric based** - Fingerprint, Iris and Face
  - Using the FAYDA's authentication service a registered authentication partner can request for biometric authentication. Currently, we support biometrics authentication using face, finger and iris.
- **Multi-factor Authentication**
  - Using various combination of above authentication modalities (fingerprint, face, iris, demographics or OTP based authentication) we can also perform authentication using the same authentication service.

Proposed integration model



API Technical Specification

> **Note:**
>
> - For API integration and testing base_url is dev.fayda.et
> - Credential Details will be provided to Integrator upon request.

1. Client Authentication Service

API to authenticate client application using clientId and secretKey

Method: POST

https://{base_url}/v1/authmanager/authenticate/clientidsecretkey

Request Body Parameters

Name	required	Data type
id	Y	string
version	Y	string
requesttime	Y	string <date-time>
metadata	N	object
request	Y	object (ClientSecret) • clientId: string • secretKey: string • appId: string

Request

```
{
  "id": "string",
  "version": "string",
  "requesttime": "2022-01-24T14:15:22Z",
  "metadata": {},
  "request": {
      "clientId": "string",
      "secretKey": "string",
      "appId": "string"
  }
}
```

Responses

Success Response

```
Response Cookie:

Set-Cookie
authorization: xxxxxxxxxxx...xxx
```

Invalid credentials: If the passed credentials is not correct.

```
{
  "id": "string",
  "version": "string",
  "responsetime": "2021-01-06T06:00:17.962Z",
  "metadata": null,
  "response": null,
  "errors": [
    {
      "errorCode": "500",
      "message": "401 Unauthorized"
    }
  ]
}
```


Incorrect Application ID: If wrong application ID is passed


```
{
  "id": "string",
  "version": "string",
  "responsetime": "2021-01-06T06:00:45.374Z",
  "metadata": null,
  "response": null,
  "errors": [
    {
      "errorCode": "KER-ATH-026",
      "message": "Realm not found:: adminXX"
    }
  ]
}
```


2. OTP Request Service

This service enables authentication partners to request for an OTP for an individual. The OTP will be send via message or email as requested to the individual. This OTP can then be used to authenticate the individual using authentication or eKYC service.

Users of OTP Request service

1. **FISP (FAYDA Infrastructure Service Provider)** - FISP acts as a gate keeper for any OTP requests sent to this service. FISP is also responsible for the policy creation on the Fayda servers so their partners will follow the set policy.
2. **Partners** - *Auth-Partners* and *eKYC-Partners* can send OTP Request to Fayda on behalf of the individual for Authentication and eKYC requests respectively, via FISP.
3. Partner-Api-Key - Associated against a policy.

Method: POST

This request will send an OTP to the individual whose UIN/VID is entered.

Resource URL

https://{base_url}/idauthentication/v1/otp/:FISP-LicenseKey/:Partner-ID/:Partner-Api-Key

Resource Details	Description
Response format	JSON
Requires Authentication	Yes

Request Header Parameters

Name	Required	description
Authorization	Y	response from client authentication service
Signature	Y	signature of the authentication request the whole body in JWT format.

Request Body Parameters

Name	Required	description
id	Y	API ID- “fayda.identity.otp”
Version	Y	The API version to be used
transactionID	Y	Transaction ID of the request. Eg: “1234567890”
requestTime	Y	Request capture time. Eg: “2021-09/17T11: 07:48.086+03:00
env	Y	Target Environment. “Staging”, “Developer”, “Pree-Production” “Production”
domainUri	Y	Unique URI per auth providers (if any). For now it is the Fayda Platform itself.
idType	Y	ID type of the individual. Values are - VID, UIN. Default is VID.
otpChannel	Y	Channel to send the OTP. Values are - EMAIL, PHONE

Request Body

```
{
  "id": "fayda.identity.otp",
  "version": "v1",
  "requestTime": "2019-02-15T07:22:57.086+05:30",
  "env": "<Target environment>",
  "domainUri": "<URI of the authentication server>",
  "transactionID": "<Transaction ID of the authentication request>",
  "individualId": "9830872690593682",
  "individualIdType": "VID",
  "otpChannel": [
    "EMAIL",
    "PHONE"
  ]
}
```


Responses

Success Response

Response Code : 200 (OK)


```
{
  "id": "fayda.identity.otp",
  "version": "v1",
  "responseTime": "2019-02-15T07:23:19.590+05:30",
  "transactionID": "<Transaction ID of the authentication request>",
  "response": {
    "maskedMobile": "XXXXXXX123",
    "maskedEmail": "abXXXXXXXXXcd@xyz.com"
  },
  "errors": null
}
```


Failed Response

Response Code : 200 (OK)

```
{
  "id": "fayda.identity.otp",
  "version": "v1",
  "responseTime": "2019-02-15T07:23:19.590+05:30",
  "transactionID": "<Transaction ID of the authentication request>",
  "response": null,
  "errors": [
    {
      "errorCode": "IDA-MLC-003",
      "errorMessage": "Invalid VID",
      "actionMessage": "Please retry with correct VID"
    }
  ]
}
```

3. Resident Authentication Service

Method: POST

This request will authenticate an individual, based on provided authentication type(s).

Resource URL

https://{base_url}/idauthentication/v1/auth/{:FISP-LicenseKey}/{:Auth-Partner-ID}/{:Partner-Api-Key}

Resource Details	Description
Response format	JSON
Requires Authentication	Yes

Request Header Parameters

Name	Required	description
Authorization	Y	For consent token
Signature	Y	For signature of the authentication request

Request Path Parameters

Name	Required	Description
FISP-LicenseKey	Y	License key provided to the FISP
eKYC-Partner-ID	Y	Partner ID of the authentication partner sending the request
Partner-API-Key	Y	API Key associated to the partner and the policy

Request Body Parameters

Name	Required	Description

id	Y	This represents the API ID. The value here should be "fayda.identity.auth".
version	Y	This represents the version of the API.
transactionID	Y	Transaction ID of the request.
requestTime	Y	The time when the request was created.
env	Y	This represents the environment. Allowed values are "Staging",”Developer”,“Production”
domainUri	Y	This represents the Unique URI per auth providers. This can be used to federate across multiple providers or countries or unions.
requestedAuth	Y	This represents the authentication types requested.
requestedAuth.otp	Y	This is used to inform that OTP authentication was performed as part of this request. Default Value here is false. Allowed values are true or false.
requestedAuth.demo	Y	This is used to inform that demographic authentication was performed as part of this request. Default value here is false. Allowed values are true or false.
requestedAuth.bio	Y	This is used to inform that biometric authentication was performed as part of this request. Default Value here is false. Allowed values are true or false.
individualId	Y	This represents the ID of resident (VID or UIN). Ex: "9830872690593682".
individualIdType	Y	ID Type used for authentication. Allowed Types of ID - VID, UIN. Default value here is VID.
consentObtained	Y	If consent of residnet is obtained? Default value here is true.
thumbprint	Y	Thumbprint of public key certificate used for encryption of sessionKey. This will be used during key rotation
requestSessionKey	Y	Symmetric Key to be created, and then encrypt the generated Symmetric Key using 'FAYDA Public Key' shared to Partner, and then Base-64-URL encoded. Algorithm used for encryption can be RSA/ECB /OAEPWITHSHA-256ANDMGF1PADDING.
requestHMAC	Y	SHA-256 hash of request block before encryption. Encryption is done using 'requestSessionKey' and then base64URL encoded. Algorithm used for encryption can be AES/GCM/PKCS5Padding.
request	Y	Request block to be used for authenticating the resident. This is encrypted using 'requestSessionKey' and then base64URL encoded. Algorithm used for encryption can be AES/GCM/PKCS5Padding.
request.otp	N	OTP used for authentication. This is mandatory when requestedAuth.otp is true.
request.timestamp	N	Timestamp when request block was captured.
request.demographics	N	Demographic data of the resident. This is mandatory when requestedAuth.demo is true.
request.biometrics	N	Biometric data of an Individual which is sent in the response from the Capture API of SBI spec v1.0. Refer to the SBI spec v1.0 specification provided below for complete information. This is mandatory when requestedAuth.bio is true.

Request Body

```
{
  "id": "fayda.identity.auth",
  "version": "v1",
  "requestTime": "2019-02-15T10:01:57.086+05:30",
  "env": "<Target environment>",
  "domainUri": "<URI of the authentication server>",
  "transactionID": "<Transaction ID of the authentication request>",
  "requestedAuth": {
    "otp": true,
    "demo": false,
    "bio": false
  },
  "consentObtained": true,
  "individualId": "9830872690593682",
  "individualIdType": "VID",
  "thumbprint": "<Thumbprint of the public key certficate used for
enryption of sessionKey. This is necessary for key rotaion>",
  "requestSessionKey": "<Encrypted and Base64-URL-encoded session
key>",
  "requestHMAC": "<SHA-256 of request block before encryption and
then hash is encrypted using the requestSessionKey>",
  //Encrypted with session key and base-64-URL encoded
  "request": {
    "timestamp": "2019-02-15T10:01:56.086+05:30 - ISO format
timestamp",
    "otp": "123456",
    "demographics": {
      "name": [
        {
          "language": "eng",
          "value": "Milkon Bulcha"
        },
        {
          "language": "amh",
          "value": " "
        }
      ],
      "gender": [
        {
          "language": "eng",
          "value": "male"
        },
        {
          "language": "amh",
          "value": ""
        }
      ],
      "age": "25",
```

```
      "dob": "25/11/1990",
      "fullAddress": [
        {
          "language": "eng",
          "value": "Woreda01, Yeka, Addis Ababa, Ethiopia "
        },
        {
          "language": "amh",
          "value": "01, ,  , "
        }
      ]
    },
    //Same as the response from the Capture API of SBI. Refer to the
[SBI specification]() for complete information.
    "biometrics": [
      {
        "specVersion" : "<SBI specification version>",
        "data": "<JWS signature format of data containing encrypted
biometrics and device details>",
        "hash": "<SHA-256 hash of (SHA-256 hash of previous data
block in hex format + SHA-256 of current data block before encrypting
in hex format) in hex format>", // For the first entry assume empty
string as previous data block
        "sessionKey": "<Encrypted and base64-URL-encoded session
key>",
        "thumbprint": "<SHA256 representation of thumbprint of the
certificate that was used for encryption of session key>"
      },
      {
        "specVersion" : "<SBI specification version>",
        "data": "<JWS signature format of data containing encrypted
biometrics and device details>",
        "hash": "<SHA-256 hash of (SHA-256 hash of previous data
block in hex format + SHA-256 of current data block before encrypting
in hex format) in hex format>",
        "sessionKey": "<Encrypted and base64-URL-encoded session
key>",
        "thumbprint": "<SHA256 representation of thumbprint of the
certificate that was used for encryption of session key>"
      }
    ]
  }
}
```

Responses

Success Response

Response Code : 200 (OK)

```
{
  "id": "fayda.identity.auth",
  "version": "v1",
  "responseTime": "2019-02-15T07:23:19.590+05:30",
  "transactionID": "<transaction_id used in request>",
  "response": {
    "authStatus": true,
    "authToken": "<authentication_token>"
  },
  "errors": null
}
```


Failed Response

Response Code : 200 (OK)


```
{
  "id": "fayda.identity.auth",
  "version": "v1",
  "responseTime": "2019-02-15T07:23:19.590+05:30",
  "transactionID": "<transaction_id used in request>",
  "response": {
    "authStatus": false,
    "authToken": null
  },
  "errors": [
    {
      "errorCode": "IDA-MLC-002",
      "errorMessage": "Invalid UIN",
      "actionMessage": "Please retry with the correct UIN"
    }
  ]
}
```


4. Resident e-KYC Service

Based on the policy linked to a FAYDA authentication partner, a partner can be eligible to perform e-KYC. In an e-KYC request, the FAYDA authentication partner can request to fetch the KYC details of the individual based on a pre-defined policy. KYC details in FAYDA is only provided to the partners after the individual's consent using OTP or biometric authentication.

This service details authentication (eKYC auth) that can be used by authentication partners to authenticate an individual and send individual's KYC details as response. Below are various authentication types supported by e-KYC authentication:

- OTP Authentication - OTP
- Biometric Authentication - Fingerprint, IRIS and Face

Users of KYC service

1. **FISP (Fayda Infrastructure Service Provider)** - FISP's role is limited to infrastructure provisioning and acting as a gate keeper for all KYC requests sent to this service. The FISP is also responsible for policy creation on the Fayda servers so their partners will follow the set policy.
2. **Partners** - *eKYC-Partners* register themselves with Fayda, under a FISP. KYC requests are captured by eKYC-Partners and sent to Fayda, via FISP.
3. Partner-Api-Key - Associated against a policy.

This request will provide KYC details of an individual, once the individual is successfully authenticated.

Method: POST

Resource URL

https://{base_url}/idauthentication/v1/kyc/:FISP-LicenseKey/:eKYC-Partner-ID/:Partner-Api-Key

Resource Details	Description
Response format	JSON
Requires Authentication	Yes

Request Header Parameters

Name	Required	description
Authorization	Y	For consent token
Signature	Y	signature of the authentication request in JWT format

Request Path Parameters

Name	Required	Description
FISP-LicenseKey	Y	License key provided to the FISP
eKYC-Partner-ID	Y	Partner ID of the authentication partner sending the request
Partner-API-Key	Y	API Key associated to the partner and the policy

Request Body Parameters

Name	Required	Description
id	Y	This represents the API ID. The value here should be "fayda. identity.kyc".
version	Y	This represents the version of the API.
transactionID	Y	Transaction ID of the request.
requestTime	Y	The time when the request was created.
env	Y	This represents the environment. Allowed values are "Staging","Developer","Production"
domainUri	Y	This represents the Unique URI per authentication providers. This can be used to federate across multiple providers or countries or unions.
requestedAuth	Y	This represents the authentication types requested.
requestedAuth.otp	Y	This is used to inform that OTP authentication was performed as part of this request. Default Value here is false. Allowed values are true or false.
requestedAuth.demo	Y	

		This is used to inform that demographic authentication was performed as part of this request. Default value here is false. Allowed values are true or false.
requestedAuth.bio	Y	This is used to inform that biometric authentication was performed as part of this request. Default Value here is false. Allowed values are true or false.
individualId	Y	This represents the ID of resident (VID or UIN). Ex: "9830872690593682".
individualIdType	Y	ID Type used for authentication. Allowed Types of ID - VID, UIN. Default value here is VID.
consentObtained	Y	If consent of residnet is obtained? Default value here is true.
thumbprint	Y	Thumbprint of public key certificate used for encryption of sessionKey. This will be used during key rotation
requestSessionKey	Y	Symmetric Key to be created, and then encrypt the generated Symmetric Key using 'FAYDA Public Key' shared to Partner, and then Base-64-URL encoded. Algorithm used for encryption can be RSA/ECB/OAEPWITHSHA-256ANDMGF1PADDING.
requestHMAC	Y	SHA-256 hash of request block before encryption. Encryption is done using 'requestSessionKey' and then base64URL encoded. Algorithm used for encryption can be AES/GCM /PKCS5Padding.
request	Y	Request block to be used for authenticating the resident. This is encrypted using 'requestSessionKey' and then base64URL encoded. Algorithm used for encryption can be AES/GCM /PKCS5Padding.
request.otp	N	OTP used for authentication. This is mandatory when requestedAuth.otp is true.
request.timestamp	N	Timestamp when request block was captured.
request.demographics	N	Demographic data of the resident. This is mandatory when requestedAuth.demo is true.
request.biometrics	N	Biometric data of an Individual which is sent in the response from the Capture API of SBI spec v1.0. Refer to the SBI spec v1.0 specification provided below for complete information. This is mandatory when requestedAuth.bio is true.
secondaryLangCode	N	Secondary language code. If specifed, the KYC response will contain KYC data for the give secondary language code also along with primary language data. Otherwise, the response will contain only primary language data.

Request Body

```
{
  "id": "fayda.identity.kyc",
  "version": "v1",
  "requestTime": "2019-02-15T10:01:57.086+05:30",
  "env": "<Target environment>",
  "domainUri": "<URI of the authentication server>",
  "transactionID": "<Transaction ID of the authentication request>",
  "requestedAuth": {
    "otp": true,
    "demo": false,
    "bio": true
```

```
  },
  "consentObtained": true,
  "individualId": "9830872690593682",
  "individualIdType": "VID",
  "thumbprint": "<SHA256 representation of thumb-print of the FAYDA
public key certificate used for encryption of sessionKey>",
  "requestSessionKey": "<Encrypted using FAYDA public key and base64-
URL-encoded session key>",
  "requestHMAC": "<SHA-256 of request block before encryption and
then hash is encrypted using the requestSessionKey>",
  //request section is first encrypted with the session key and then
base64-URL-encoded
  "request": {
    "timestamp": "2019-02-15T10:01:56.086+05:30 - ISO format time-
stamp",
    "otp": "123456",
    //biometric section is same as the response from Capture API
mentioned in [SBIv1.0 specification]()
    "biometrics": [
      {
        "specVersion" : "<SBI specification version>",
        "data": "<JWS signature format of data containing encrypted
biometrics and device details>",
        "hash": "<SHA-256 hash of (SHA-256 hash of previous data
block in hex format + SHA-256 of current data block before encrypting
in hex format) in hex format>", // For the first entry assume empty
string as previous data block
        "sessionKey": "<Encrypted with FAYDA public key and base64-
URL-encoded session key>",
        "thumbprint": "<SHA256 representation of thumb-print of the
FAYDA public key that was used for encryption of session key>"
      },
      {
        "specVersion" : "<SBI specification version>",
        "data": "<JWS signature format of data containing encrypted
biometrics and device details>",
        "hash": "<SHA-256 hash of (SHA-256 hash of previous data
block in hex format + SHA-256 of current data block before encrypting
in hex format) in hex format>",
        "sessionKey": "<Encrypted and base64-URL-encoded session
key>",
        "thumbprint": "<SHA256 representation of thumb-print of the
FAYDA public key that was used for encryption of session key>"
      }
    ]
  },
  "secondaryLangCode": "eng"
}
```

Responses

Success Response

Response Code : 200 (OK)

```
{
  "id": "fayda.identity.kyc",
  "version": "v1",
  "responseTime": "2019-02-15T07:23:19.590+05:30",
  "transactionID": "<Transaction ID received in request>",
  "response": {
    "kycStatus": true,
    "authResponseToken": "<Authentication response token>",
    //Encrypted KYC info using session key which intern is encrypted
using Partner's public key and base64-URL-encoded
    //Session key and data content are splited using #KEY_SPLITTER#
text
    "identity": {
      "name": [
        {
          "language": "eng",
          "value": "Milkon Bulcha"
        },
        {
          "language": "amh",
          "value": " "
        }
      ],
      "dob": "25/11/1990",
      "gender": [
        {
          "language": "eng",
          "value": "male"
        }
      ],
      "phoneNumber": "+212-5398-12345",
      "emailId": "sample@samplamail.com",
      "fullAddress": [
        {
          "language": "eng",
          "value": "Woreda01, Yeka, Addis Ababa, Ethiopia "
        },
        {
          "language": "amh",
          "value": "01, ,  , "
        }
      ]

    },
    "thumbnail": "<SHA256 representation of thumb-print of the
Partner's public key used for encryption of identity block>"
```

```
  },
  "errors": null
}
```

Failed Response

Response Code : 200 (OK)

```
{
  "id": "fayda.identity.kyc",
  "version": "v1",
  "responseTime": "2019-02-15T07:23:19.590+05:30",
  "transactionID": "<Transaction ID received in request>",
  "response": {
    "kycStatus": false,
    "authResponseToken": null,
    "identity": null,
    "thumbnail": null
  },
  "errors": [
    {
      "errorCode": "IDA-MLC-002",
      "errorMessage": "Invalid UIN",
      "actionMessage": "Please retry with the correct UIN"
    }
  ]
}
```

Failure Details:

Error Code	Error Message	Description	Action Message
IDA-MLC-001	Time request to be received at Fayda	Invalid Time stamp	Please send the request witn in x hrs /mins
IDA-MLC-002	Invalid UIN	Invalid UIN	Please Retry with a correct UIN
IDA-MLC-003	UIN has been deactivated	UIN Deactivated	UIN status is not active
IDA-MLC-004	Invalid VID	Invalid VID	Please Retry with a valid VID
IDA-MLC-005	Wrong VID	Expired, used, Revoked VID	Please Regenerate VID and Retry
IDA-MLC-006	Missing input parameter …	Missing Input Parameter -attribute- list of mandatory missing inputs	
IDA-MLC-007	Request could not be processed.	Could not process request/Unknown error; Invalid Auth Request; Unable to encrypt eKYC response	Please try again
IDA-MLC-009	Invalid Input parameter- attribute	Invalid Input parameter- attribute	
IDA-MLC-010	VID has been deactivated	VID corresponding to a deactivated UIN	
IDA-MLC-014	<Notification Channel> not registered. Individual has to register and try again	<Notification Channel> not Registered (Phone/e-mail/both)	Please register your <Notification Chann and try again
IDA-MLC-015	Identity Type - <Identity Type> not configured for the country	ID Type (UIN/VID) not supported for a country	
IDA-MLC-017	Invalid UserID	Invalid UserID	
IDA-MLC-018	%s not available in database	UIN,VID, User ID not available in database	

IDA-MPA-004	Fayda Public key expired.	Fayda Public key expired	Please reinitiate the request with updated public key
IDA-MPA-005	OTP Request Usage not allowed as per policy	OTP Trigger Usage not allowed as per policy	
IDA-MPA-007	License key does not belong to a registered	License key does not belong to a registered	
IDA-MPA-008	License key expired	License key expired	
IDA-MPA-009	Partner not registered	PartnerID invalid	
IDA-MPA-010	FISP and partner not mapped	FISP and partner not mapped	
IDA-MPA-011	License key of FISP is suspended	License key of FISP is suspended	
IDA-MPA-012	Partner is deactivated	PartnerID not Active	
IDA-MPA-014	Partner not assigned any policy	PartnerID not mapped to a policy	
IDA-MPA-017	License key of FISP is blocked	License key status is blocked	
IDA-OTA-001	Numerous OPT requests received	OTP Flooding error	
IDA-OTA-002	Could not generate / send OTP	Could not generate / send OTP	
IDA-OTA-006	UIN is locked for OTP	Frozen Account	Please try again later
IDA-OTA-008	OTP Notification channel not provided	No OTP channel in input	
IDA-OTA-009	<Channel> not configured for Ethiopia	channel not configured (Phone / Email / both)	