

AEPS (Aadhaar Enabled Payment System) API Documentation

Author: Your Company Name

Version: 1.0

Date: July 2026

Overview

The Aadhaar Enabled Payment System (AePS) is a bank-led model that allows interoperable financial transactions at Point of Sale (PoS) devices (Micro ATMs) using the customer's Aadhaar number as a unique identity. This API enables agents or merchants to perform financial transactions securely using biometric authentication without requiring a debit card or PIN.

This documentation covers the complete AePS Fingpay integration including agent onboarding, eKYC (one-time and daily), and core transaction APIs.

Authentication & Headers

All API requests require the following headers for authentication:

Header	Type	Description
developer_key	string	Static API key issued to your account after KYC
secret-key	string	Dynamic per-request signature: base64(HMAC-SHA256(timestamp, base64(access_key)))
secret-key-timestamp	string	Current time in milliseconds since UNIX epoch
content-type	string	application/json

Requests accept both application/json and application/x-www-form-urlencoded bodies. Use multipart/form-data for file uploads (e.g., Activate Agent).

Environments & Base URL

Environment	Base URL	Description
UAT / Sandbox	<code>https://staging.eko.in/ekoapi/v3</code>	Self-serve credentials available immediately on signup
Production	<code>https://api.eko.in/ekoicici/v3</code>	Credentials issued after organizational KYC

The full endpoint URL is always `baseUrl + path` .

Common Response Structure

All AEPS APIs share a common response envelope:

Field	Type	Description
<code>status</code>	number	Primary success indicator (0 = success)
<code>message</code>	string	Human-readable response / error message
<code>response_status_id</code>	number	Granular status id; see the shared error-codes table
<code>response_type_id</code>	number	A unique id for every possible response shape (success or error) — useful for client logic branching and analytics
<code>data</code>	object	API-specific response payload

Activate for Agent

1. Activate AePS Fingpay for Agent

Enable AePS Fingpay service for your agent by submitting their biometric device details and KYC documents.

PUT `/admin/network/agent/{user_code}/aeps-fingpay/activate`

This API enables an agent (identified by their `user_code`) to use the AePS Fingpay service. It accepts the agent's biometric device model, serial number, address proofs, and KYC documents (PAN card, Aadhaar front and back) as a multipart form submission. After submission, the activation enters a 'pending' state and is approved within 2–3 business days. Only activated agents can perform AePS transactions. File uploads must be JPEG/JPG/PDF format, each under 1 MB; PNG is not accepted.

Important: This API must be called after onboarding your agent using the **Onboard User API**. **Warning:** The approval may take 1-2 business days.

Request

Path parameters

Field	Type	Required	Description
<code>user_code</code>	string	required	Unique code of the agent for whom AePS Fingpay service is being activated. e.g. 20810200

Body parameters

Field	Type	Required	Description
<code>initiator_id</code>	string	required	Registered mobile number of the API user (see Platform Credentials). e.g. 9962981729
<code>client_ref_id</code>	string	optional	Unique reference ID per API call, generated by your system (max 20 characters). e.g. 2026010100123456789
<code>modelname</code>	string	required	Model name/designation of the UIDAI-certified biometric device (e.g., Morpho 1300E3,

			Mantra MFS100). e.g. Morpho 1300E3
devicenumber	string	required	Serial number of the biometric device as printed on the device or its packaging. e.g. SN1234567890
shop_type	number	required	The shop-type ID of the Agent. Use Get Shop Types API for a list of shop-types and corresponding IDs. e.g. 4215
office_address	object	required	Agent's current office/operating address as a JSON object with keys: line , city , state , state_id , pincode . e.g. { "line": "Shop No. 5", "city": "Gurgaon", "state": "Haryana", "state_id": 23, "pincode": "122003" }
address_as_per_proof	object	required	Agent's address exactly as it appears on the submitted address proof document. e.g. { "line": "Shop No. 5", "city": "Gurgaon", "state": "Haryana", "state_id": 23, "pincode": "122003" }
pan_card	file	required	PAN card document upload (multipart/form-data). Accepted formats: JPEG, JPG, PDF. Max size: 1 MB.
aadhar	string	required	12-digit Aadhaar number of the agent. e.g. 123456789012

aadhar_front	file	required	Front side of the Aadhaar card (multipart/form-data). Accepted formats: JPEG, JPG, PDF. Max size: 1 MB.
aadhar_back	file	required	Back side of the Aadhaar card (multipart/form-data). Accepted formats: JPEG, JPG, PDF. Max size: 1 MB.
latlong	string	required	GPS coordinates of the agent. Format: latitude,longitude . e.g. 28.6139,77.2090

Response types

response_type_id	Meaning	Next step
1259	AePS Registration Successful	Onboard Sender
1297	User does not exist	Onboard User

Responses

200 Successful response

1259 — AePS Registration Successful

JSON

```
{
  "status": 0,
  "message": "AePS Registration Successful",
  "response_status_id": 0,
  "response_type_id": 1259,
  "data": {
    "service_status": "pending",
    "service_status_desc": "Documents submitted and under review"
  }
}
```

```
}  
}
```

200 Agent not onboarded

JSON

```
{  
  "status": 1,  
  "message": "User does not exist. Please onboard the agent first.",  
  "response_status_id": 1297,  
  "response_type_id": 1297,  
  "data": {}  
}
```

2. Get Shop Types

List the Merchant Category Codes (MCC) available for AePS Fingpay agent onboarding.

GET /user/collection/aeps-fingpay/get-Mcc-Category

This endpoint returns a list of all valid shop types (MCC codes) that can be assigned to an agent during the activation process. Use the returned IDs as the `shop_type` parameter in the Activate Agent API.

Request

No parameters required.

Response types

response_type_id	Meaning	Next step
2110	Shop-type list returned successfully	Activate AePS Fingpay for Agent

Responses

200 Successful response

JSON

```
{  
  "response_status_id": 0,
```

```

    "param_attributes": {
      "list_elements": [
        {
          "label": "Courier services – air and ground and freight forwarders",
          "value": 4215
        },
        {
          "label": "Groceries and supermarkets",
          "value": 5411
        }
      ]
    },
    "response_type_id": 2110,
    "message": "Success",
    "status": 0
  }
}

```

3. Get States

List the states (with their state_id) available for AePS Fingpay agent onboarding.

GET /user/collection/aeps-fingpay/get-states

This endpoint returns a list of all Indian states. Use the returned `state_id` values in the `office_address` and `address_as_per_proof` objects during agent activation.

Request

No parameters required.

Response types

response_type_id	Meaning	Next step
2127	State list returned successfully	Activate AePS Fingpay for Agent

Responses

200 Successful response

JSON

```

{
  "response_status_id": 0,
  "param_attributes": {

```

```

    "list_elements": [
      {
        "stateCode": "HR",
        "label": "Haryana",
        "value": 8
      },
      {
        "stateCode": "MH",
        "label": "Maharashtra",
        "value": 15
      }
    ]
  },
  "response_type_id": 2127,
  "message": "Success",
  "status": 0
}

```

Agent eKYC (1-Time)

The one-time eKYC flow consists of three mandatory steps that must be completed **in order**: Send OTP → Verify OTP → Biometric eKYC. This is distinct from the daily authentication (2FA) required once per calendar day before transacting.

4. Send OTP (eKYC)

Initiate AePS Fingpay eKYC by sending an OTP to the agent's registered Aadhaar-linked mobile number.

POST `/user/collection/aeaps-fingpay/kyc/otp`

First-time KYC follows three steps in order: `Send OTP` → `Verify OTP` → `Biometric`. On subsequent days, use the Daily KYC (biometric-only) endpoint.

This is the first step in the one-time AePS Fingpay eKYC flow. The OTP is delivered to the number passed in `customer_id`. The eKYC flow must be completed once per agent before they can perform any AePS transactions.

Request

Body parameters

Field	Type	Required	Description
-------	------	----------	-------------

initiator_id	string	required	Registered mobile number of the API user (see Platform Credentials). e.g. 9962981729
client_ref_id	string	optional	Unique reference ID per API call, generated by your system (max 20 characters). e.g. 2026010100123456789
user_code	string	required	Unique code of your user/agent/retailer the service is run for. e.g. 20810200
aadhar	string	required	RSA-encrypted, Base64-encoded Aadhaar number of the agent undergoing eKYC. e.g. BASE64_ENCRYPTED_AADHAAR
customer_id	string	required	Registered mobile number of the agent/merchant undergoing eKYC. e.g. 9123456789
latlong	string	required	Agent's GPS coordinates as latitude,longitude . e.g. 28.6139,77.2090

Response types

response_type_id	Meaning	Next step
1600	OTP request has been sent	Verify OTP (eKYC)

Responses

200 Successful response

JSON

```
{
  "status": 0,
  "message": "OTP request has been sent",
  "response_status_id": 0,
  "response_type_id": 1600,
  "data": {
    "reference_tid": "EKYKF4719702240123152147525I",
    "otp_ref_id": "2465238"
  }
}
```

5. Verify OTP (eKYC)

Verify the eKYC OTP sent to the agent's Aadhaar-linked mobile number to advance the one-time AePS Fingpay eKYC.

PUT /user/collection/aeaps-fingpay/kyc/otp/verify

The second step of the one-time AePS Fingpay eKYC flow (Send OTP → Verify OTP → Biometric). Submits the OTP the agent received, together with the `otp_ref_id` and `reference_tid` returned by the Send OTP API, to validate the agent's identity before biometric capture.

Request

Body parameters

Field	Type	Required	Description
<code>initiator_id</code>	string	required	Registered mobile number of the API user (see Platform Credentials). e.g. 9962981729

client_ref_id	string	optional	Unique reference ID per API call, generated by your system (max 20 characters). e.g. 2026010100123456789
user_code	string	required	Unique code of your user/agent/retailer the service is run for. e.g. 20810200
customer_id	string	required	Registered mobile number of the agent/merchant undergoing eKYC. e.g. 9123456789
aadhar	string	required	RSA-encrypted, Base64-encoded Aadhaar number of the agent undergoing eKYC. e.g. BASE64_ENCRYPTED_AADHAAR
otp	string	required	OTP received on the agent's Aadhaar-registered mobile number. e.g. 123456
otp_ref_id	string	required	Reference ID returned by the Send OTP (eKYC) API. e.g. 2465238
reference_tid	string	required	Transaction reference ID returned by the Send OTP (eKYC) API. e.g. EKYKF4719702240123152147525I
latlong	string	required	Agent's GPS coordinates as latitude,longitude . e.g. 28.6139,77.2090

Response types

response_type_id	Meaning	Next step
1604	Validation successful	Biometric eKYC

Responses

200 Successful response

JSON

```
{
  "status": 0,
  "message": "Validation successful",
  "response_status_id": 0,
  "response_type_id": 1604,
  "data": {
    "reference_tid": "BEKYF4719702240123152147525I",
    "otp_ref_id": "815031"
  }
}
```

6. Biometric eKYC

Complete one-time AePS Fingpay eKYC by submitting the agent's Aadhaar and live biometric fingerprint capture.

PUT /user/collection/aeps-fingpay/kyc/biometric

The final step in the one-time AePS Fingpay eKYC flow, called after OTP verification. Submits the agent's Aadhaar and live biometric PID data to UIDAI for identity verification. On success the agent's eKYC is marked complete and they can start performing AePS transactions (subject to completing daily 2FA each day).

Important: You need to encrypt the Aadhaar number before passing it as a parameter. The same RSA-encrypted Aadhaar + PID XML format is used by all AePS transaction APIs.

Aadhaar Encryption

1. Decode the public key using Base64.

2. Compute the RSA-encrypted signature using the decoded key and the Aadhaar message.
3. Base64-encode the encrypted signature before sending it on the API.

E-KYC steps

For Fingpay AePS, it is mandatory to use e-KYC OTP request, e-KYC OTP verification and biometric before any AePS transaction. Make sure you complete these APIs in order:

Step	API	Purpose
1	Send OTP (eKYC)	OTP to the agent's Aadhaar-linked mobile
2	Verify OTP (eKYC)	Validate the OTP and start the eKYC session
3	Biometric eKYC	Submit Aadhaar + live fingerprint PID (this API)

Request

Body parameters

Field	Type	Required	Description
<code>initiator_id</code>	string	required	Registered mobile number of the API user (see Platform Credentials). e.g. 9962981729
<code>client_ref_id</code>	string	optional	Unique reference ID per API call, generated by your system (max 20 characters). e.g. 2026010100123456789
<code>user_code</code>	string	required	Unique code of your user/agent/retailer the service is run for. e.g. 20810200
<code>aadhar</code>	string	required	RSA-encrypted, Base64-encoded

			Aadhaar number of the agent. e.g. <code>BASE64_ENCRYPTED_AADHAAR</code>
<code>customer_id</code>	string	required	Registered mobile number of the agent/merchant undergoing eKYC. e.g. <code>9123456789</code>
<code>latlong</code>	string	required	Agent's GPS coordinates as <code>latitude,longitude</code> . e.g. <code>28.6139,77.2090</code>
<code>piddata</code>	string	required	PID XML string from the UIDAI-certified biometric device (fType=2, DataType='X', mc in DeviceInfo). e.g. <code><?xml version='1.0'?><PidData><DataType='X'>...</DataType><DeviceInfo mc='...' /></PidData></code>
<code>bank_code</code>	string	required	Eko bank code of the agent's bank. Resolve it using the Get List of Banks API. e.g. <code>HDFC</code>
<code>otp_ref_id</code>	string	required	Reference ID returned by the preceding OTP step, linking this biometric capture to the verified OTP session. e.g. <code>2465238</code>
<code>reference_tid</code>	string	required	Transaction reference ID returned by the preceding OTP step. e.g. <code>EKYKF4719702240123152147525I</code>

Response types

response_type_id	Meaning	Next step
1605	eKYC successful — agent may now complete Daily KYC	Daily KYC

Responses

200 Successful response

JSON

```
{
  "status": 0,
  "message": "eKYC successful",
  "response_status_id": 0,
  "response_type_id": 1605,
  "data": {
    "user_code": "20810200"
  }
}
```

Agent eKYC (Daily)

7. Daily KYC

Perform the mandatory daily biometric re-verification that authorises an agent to carry out AePS transactions for the current calendar day.

PUT /user/collection/aeaps-fingpay/kyc/biometric/daily

Biometric-only re-verification for the days after the one-time eKYC — no OTP step is required. AePS Fingpay requires every agent to re-authenticate themselves biometrically at the start of each working day, before their first transaction of the day.

If this fails with reason `Please complete bank eKYC to process the transaction.`, re-run the full first-time eKYC sequence — Send OTP → Verify OTP → Biometric — before retrying.

To capture the `piddata` PID block with an RDSservice-compliant fingerprint scanner, see the

Aadhaar Biometric Authentication guide.

Tip: This endpoint is one step in a complete workflow:

AePS (Fingpay) — Cash Withdrawal — Aadhaar-enabled cash withdrawal: one-time agent activation and eKYC, daily KYC, then the biometric withdrawal.

Request

Body parameters

Field	Type	Required	Description
<code>initiator_id</code>	string	required	Registered mobile number of the API user (see Platform Credentials). e.g. <code>9962981729</code>
<code>client_ref_id</code>	string	optional	Unique reference ID per API call, generated by your system (max 20 characters). e.g. <code>2026010100123456789</code>
<code>user_code</code>	string	required	Unique code of your user/agent/retailer the service is run for. e.g. <code>20810200</code>
<code>aadhar</code>	string	required	RSA-encrypted, Base64-encoded Aadhaar number of the agent performing daily KYC. e.g. <code>BASE64_ENCRYPTED_AADHAAR</code>
<code>customer_id</code>	string	required	Registered mobile number of the agent/merchant. e.g. <code>9123456789</code>
<code>latlong</code>	string	required	GPS coordinates of the agent's location at the time of daily KYC. e.g. <code>28.6139,77.2090</code>

<code>piddata</code>	string	required	PID XML string from the UIDAI-certified biometric device (fType=2, Data type='X', mc in DeviceInfo). This represents the agent's own fingerprint, not the customer's. If you generate the PID block yourself, the value of <code>wadh</code> must be blank/empty for Daily KYC. e.g. <code><?xml version='1.0'?> <PidData><Data type='X'>...</Data> <DeviceInfo mc='...' /> </PidData></code>
<code>bank_code</code>	string	required	Eko bank code of the agent's bank. Resolve it using the Get List of Banks API. e.g. <code>HDFC</code>

Response types

response_type_id	Meaning	Next step
<code>1713</code>	Daily KYC successful — the agent may transact today	—
<code>1714</code>	Daily KYC failed — if <code>data.reason</code> is 'Please complete bank eKYC to process the transaction.', re-run the full eKYC from Send OTP	Send OTP (eKYC)

Responses

200 Successful response

JSON

```
{
  "response_status_id": 0,
  "data": {
    "user_code": "20810200"
  },
  "response_type_id": 1713,
  "message": "KYC success",
  "status": 0
}
```

200 KYC failed — no reason returned

JSON

```
{
  "status": 1,
  "message": "KYC failed",
  "response_status_id": 1,
  "response_type_id": 1714,
  "data": {}
}
```

200 Invalid biometric data — check the wadh value in the PID block

JSON

```
{
  "status": 1,
  "message": "Authentication Failed. Invalid Biometric data.",
  "response_status_id": 1,
  "response_type_id": 1714,
  "data": {
    "reason": "Invalid biometric data"
  }
}
```

200 Bank eKYC pending — re-run Send OTP → Verify OTP → Biometric first

JSON

```
{
  "status": 1,
  "message": "Please complete bank eKYC to process the transaction.",
  "response_status_id": 1,
  "response_type_id": 1714,
  "data": {
```

```
  "reason": "Please complete bank eKYC to process the transaction."
}
```

AePS Transactions

8. AePS Cash Withdrawal

Withdraw cash from any Aadhaar-linked bank account using biometric fingerprint authentication — no card or PIN required.

POST /customer/collection/aeps-fingpay/cash-withdrawal/{customer_id}

Lets a customer withdraw cash from any Aadhaar-linked bank account using a live fingerprint scan — no card or PIN. The agent's biometric device produces a PID XML blob that is forwarded verbatim, and the customer's Aadhaar number is RSA-encrypted before transmission.

Prerequisites

The agent must have completed, in order:

1. AePS Fingpay activation
2. One-time eKYC (Send OTP → Verify OTP → Biometric eKYC)
3. **Daily KYC** for the current day

Important: Due to NPCI compliance, the agent's Daily KYC must succeed for the current calendar day before any cash-withdrawal is attempted.

Request

Path parameters

Field	Type	Required	Description
customer_id	string	required	Customer's registered mobile number. e.g. 9000000000

Body parameters

Field	Type	Required	Description
<code>initiator_id</code>	string	required	Registered mobile number of the API user. e.g. 9962981729
<code>client_ref_id</code>	string	optional	Unique reference ID per API call (max 20 characters). e.g. 2026010100123456789
<code>user_code</code>	string	required	Unique code of your user/agent/retailer. e.g. 10000001
<code>bank_code</code>	string	required	Short bank code identifying the customer's Aadhaar-linked bank (e.g. HDFC , SBIN). e.g. HDFC
<code>aadhar</code>	string	required	RSA-encrypted, Base64-encoded Aadhaar number. e.g. BASE64_ENCRYPTED_AADHAAR
<code>latlong</code>	string	required	GPS coordinates of the transaction origin in 'latitude,longitude' format. e.g. 28.6139,77.2090
<code>piddata</code>	string	required	PID data captured from the UIDAI-certified biometric device, as a raw XML string. Must use Data type='X' (XML, not Protobuf). e.g. <?xml version='1.0'?> <PidData><Data type='X'>...</Data> <DeviceInfo mc='...' /> </PidData>

amount	number	required	Withdrawal amount in Indian Rupees (integer). Must be greater than 0. e.g. 1000
--------	--------	----------	--

Response types

response_type_id	Meaning	Next step
1463	Transaction Successful	—
1464	Transaction Fail	—
1465	Transaction Pending — check final status later	Transaction Inquiry

Responses

200 Successful response

1463 — Transaction Successful

JSON

```
{
  "response_status_id": 0,
  "data": {
    "tx_status": "0",
    "tds": "0.00",
    "shop": "Test User",
    "sender_name": "Test User",
    "tid": "0000000000",
    "bank": "Example Bank",
    "balance": "0.00",
    "user_code": "10000001",
    "merchantname": "Test User",
    "aadhar": "XXXX XXXX 0000",
    "customer_balance": "0.00",
    "transaction_time": "01-01-24 00:00:00",
    "commission": "0.00",
    "bank_ref_num": "0000000000",
    "transaction_date": "01-01-24 00:00:00",
    "amount": "1000.00",
```

```
    "comment": "Request Completed"
  },
  "response_type_id": 1463,
  "message": "Transaction Successful",
  "status": 0
}
```

200 Transaction Fail — incorrect merchant credentials

JSON

```
{
  "status": 1,
  "message": "Transaction failed",
  "response_status_id": 1,
  "response_type_id": 1464,
  "data": {
    "tx_status": "1",
    "reason": "Incorrect merchant credentials"
  }
}
```

9. AePS Balance Enquiry

Check a customer's bank account balance using Aadhaar number and biometric fingerprint — no card or PIN required.

POST /customer/collection/aeps-fingpay/balance-enquiry/{customer_id}

Retrieves the real-time account balance from any Aadhaar-linked bank. Uses the dedicated `balance-enquiry` endpoint — the request shape matches Cash Withdrawal without the `amount` field. No money movement occurs and no debit takes place.

The agent must have completed AePS Fingpay activation and the current-day daily authentication before calling this API.

Request

Path parameters

Field	Type	Required	Description
-------	------	----------	-------------

customer_id	string	required	Customer's registered mobile number. e.g. 9000000000
-------------	--------	----------	---

Body parameters

Field	Type	Required	Description
initiator_id	string	required	Registered mobile number of the API user. e.g. 9962981729
client_ref_id	string	optional	Unique reference ID per API call (max 20 characters). e.g. 2026010100123456789
user_code	string	required	Unique code of your user/agent/retailer. e.g. 10000001
bank_code	string	required	Short bank code identifying the customer's Aadhaar-linked bank (e.g. HDFC , SBIN). e.g. HDFC
aadhar	string	required	RSA-encrypted, Base64-encoded Aadhaar number. e.g. BASE64_ENCRYPTED_AADHAAR
latlong	string	required	GPS coordinates of the transaction origin in 'latitude,longitude' format. e.g. 28.6139,77.2090
piddata	string	required	PID XML string from the UIDAI-certified biometric device (fType=2, Data type='X', mc present in

DeviceInfo).

e.g. `<?xml version='1.0'?>`
`<PidData><Data`
`type='X'>...</Data>`
`<DeviceInfo mc='...' />`
`</PidData>`

Response types

response_type_id	Meaning	Next step
1466	Transaction Successful	—
1464	Transaction Fail	—

Responses

200 Successful response

1466 — Transaction Successful

JSON

```
{
  "response_status_id": 0,
  "data": {
    "tx_status": "0",
    "tid": "0000000000",
    "bank": "Example Bank",
    "customer_balance": "5000.00",
    "user_code": "10000001",
    "amount": "0.00",
    "shop": "Test User",
    "sender_name": "Test User",
    "merchantname": "Test User",
    "transaction_date": "01-01-24 00:00:00",
    "transaction_time": "01-01-24 00:00:00",
    "comment": "Request Completed"
  },
  "response_type_id": 1466,
  "message": "Transaction Successful",
  "status": 0
}
```

200 Transaction Fail — incorrect merchant credentials

JSON

```
{
  "status": 1,
  "message": "Transaction failed",
  "response_status_id": 1,
  "response_type_id": 1464,
  "data": {
    "tx_status": "1",
    "reason": "Incorrect merchant credentials"
  }
}
```

10. AePS Mini Statement

Retrieve the last few transactions from an Aadhaar-linked bank account via biometric authentication.

POST /customer/collection/aeps-fingpay/mini-statement/{customer_id}

Fetches a mini statement (typically the last 5–10 transactions) from a customer's bank account by authenticating through Aadhaar biometrics. Uses the dedicated mini-statement endpoint — the request shape matches Balance Enquiry (no amount). No money movement occurs. The response includes a list of recent debit/credit transactions with amounts and dates.

Request

Path parameters

Field	Type	Required	Description
customer_id	string	required	Customer's registered mobile number. e.g. 9000000000

Body parameters

Field	Type	Required	Description
initiator_id	string	required	Registered mobile number of the API user.

			e.g. 9962981729
client_ref_id	string	optional	Unique reference ID per API call (max 20 characters). e.g. 2026010100123456789
user_code	string	required	Unique code of your user/agent/retailer. e.g. 10000001
bank_code	string	required	Short bank code identifying the customer's Aadhaar-linked bank. e.g. HDFC
aadhar	string	required	RSA-encrypted, Base64-encoded Aadhaar number. e.g. BASE64_ENCRYPTED_AADHAAR
latlong	string	required	GPS coordinates of the transaction origin in 'latitude,longitude' format. e.g. 28.6139,77.2090
piddata	string	required	PID XML string from the UIDAI-certified biometric device (fType=2, Data type='X', mc present in DeviceInfo). e.g. <?xml version='1.0'?><PidData><Data type='X'>...</Data><DeviceInfo mc='...' /></PidData>

Response types

response_type_id	Meaning	Next step
------------------	---------	-----------

1527	Transaction Successful	—
1528	Transaction Fail	—

Responses

200 Successful response

1527 — Transaction Successful

JSON

```
{
  "response_status_id": 0,
  "data": {
    "tx_status": "0",
    "tid": "0000000000",
    "mini_statement_list": [
      { "date": "01/01", "amount": "500.00", "narration": "NEFT Credit", "txnTy": "C", "tx": "0000000000" },
      { "date": "02/01", "amount": "200.00", "narration": "ATM Withdrawal", "tx": "0000000000", "txnTy": "W" },
      { "date": "03/01", "amount": "1000.00", "narration": "UPI Transfer", "txnTy": "T", "tx": "0000000000" }
    ],
    "customer_balance": "5000.00",
    "user_code": "10000001",
    "sender_name": "Test User",
    "merchantname": "Test User",
    "commission": "5.00",
    "transaction_date": "01-01-24 00:00:00",
    "transaction_time": "01-01-24 00:00:00",
    "comment": "Request Completed"
  },
  "response_type_id": 1527,
  "message": "Transaction Successful",
  "status": 0
}
```

200 Transaction Fail

JSON

```
{
  "status": 1,
  "message": "Transaction failed",
  "response_status_id": 1,
  "response_type_id": 1528,
  "data": {
```

```

    "tx_status": "1",
    "reason": "Incorrect merchant credentials"
  }
}

```

Error Codes & Response Types

Complete Response Type Reference

response_type_id	API	Meaning
1259	Activate Agent	AePS Registration Successful
1297	Activate Agent	User does not exist
1600	Send OTP (eKYC)	OTP request has been sent
1601	Send OTP (eKYC)	AePS Fingpay service not activated
1602	Send OTP (eKYC)	Incorrectly encrypted Aadhaar
1604	Verify OTP (eKYC)	Validation successful
1605	Biometric eKYC	eKYC successful — agent may now complete Daily KYC
1606	Biometric eKYC	OTP steps skipped
1713	Daily KYC	Daily KYC successful — the agent may transact today
1714	Daily KYC	Daily KYC failed
1463	Cash Withdrawal	Transaction Successful
1464	Cash Withdrawal / Balance / Mini Statement	Transaction Fail
1465	Cash Withdrawal	Transaction Pending
1466	Balance Enquiry	Transaction Successful
1527	Mini Statement	Transaction Successful
1528	Mini Statement	Transaction Fail

Transaction Status Codes (tx_status)

tx_status	Meaning
0	Success
1	Fail
2	Awaited / Pending
3	Refund Pending
4	Refunded
5	On Hold

SDK Integration

Node.js

Bash

```
npm i @yourcompany/aeps-sdk
```

JavaScript

```
import { AepsClient } from "@yourcompany/aeps-sdk";

const client = new AepsClient({
  developerKey: process.env.AEPS_DEVELOPER_KEY,
  accessKey: process.env.AEPS_ACCESS_KEY,
  initiatorId: "9962981729",
  userCode: "20810200",
  environment: "sandbox",
});

// Cash Withdrawal
const result = await client.call("aeps-cash-withdrawal", {
  "customer_id": "90000000000",
  "bank_code": "HDFC",
  "aadhar": "BASE64_ENCRYPTED_AADHAAR",
  "latlong": "28.6139,77.2090",
  "piddata": "<?xml version='1.0'?><PidData><Data type='X'>...</Data><DeviceInf>
```

```

        "amount": 1000
    });
    console.log(result);

    // Balance Enquiry
    const balance = await client.call("aeps-balance-enquiry", {
        "customer_id": "90000000000",
        "bank_code": "HDFC",
        "aadhar": "BASE64_ENCRYPTED_AADHAAR",
        "latlong": "28.6139,77.2090",
        "piddata": "<?xml version='1.0'?><PidData><Data type='X'>...</Data><DeviceInf
    });
    console.log(balance);

```

PHP

PHP

```

$client = new \YourCompany\Aeps\AepsClient([
    'developerKey' => $_ENV['AEPS_DEVELOPER_KEY'],
    'accessKey' => $_ENV['AEPS_ACCESS_KEY'],
    'initiatorId' => '9962981729',
    'userCode' => '20810200',
    'environment' => 'sandbox',
]);

// Cash Withdrawal
$result = $client->call('aeps-cash-withdrawal', [
    'customer_id' => '90000000000',
    'bank_code' => 'HDFC',
    'aadhar' => 'BASE64_ENCRYPTED_AADHAAR',
    'latlong' => '28.6139,77.2090',
    'piddata' => '<?xml version=\'1.0\'?><PidData><Data type=\'X\'>...</Data><D
    'amount' => 1000,
]);
echo json_encode($result);

```

cURL Examples

Cash Withdrawal

Bash

```
curl -X POST "https://staging.eko.in/ekoapi/v3/customer/collection/aeps-fingpay/
-H "developer_key: YOUR_DEVELOPER_KEY" \
-H "secret-key: YOUR_HMAC_SIGNATURE" \
-H "secret-key-timestamp: 1700000000000" \
-H "content-type: application/json" \
-d '{
  "initiator_id": "9962981729",
  "user_code": "10000001",
  "bank_code": "HDFC",
  "aadhar": "BASE64_ENCRYPTED_AADHAAR",
  "latlong": "28.6139,77.2090",
  "piddata": "<?xml version='\"'1.0'\"'?><PidData><Data type='\"'X'\"'>...</D
  "amount": 1000
}'
```

Daily KYC

Bash

```
curl -X PUT "https://staging.eko.in/ekoapi/v3/user/collection/aeps-fingpay/kyc/
-H "developer_key: YOUR_DEVELOPER_KEY" \
-H "secret-key: YOUR_HMAC_SIGNATURE" \
-H "secret-key-timestamp: 1700000000000" \
-H "content-type: application/json" \
-d '{
  "initiator_id": "9962981729",
  "user_code": "20810200",
  "aadhar": "BASE64_ENCRYPTED_AADHAAR",
  "customer_id": "9123456789",
  "latlong": "28.6139,77.2090",
  "piddata": "<?xml version='\"'1.0'\"'?><PidData><Data type='\"'X'\"'>...</D
  "bank_code": "HDFC"
}'
```

Complete Workflow Summary

Step	API	Method	Endpoint	Purpose
1	Onboard User	POST	/admin/network/agent/onboard	Register a new agent
2	Activate Agent	PUT	/admin/network/agent/{user_code}/aep	Enable AePS service for agent

			s-fingpay/activate	
3	Send OTP	POST	/user/collection/aeps-fingpay/kyc/otp	Initiate eKYC
4	Verify OTP	PUT	/user/collection/aeps-fingpay/kyc/otp/verify	Validate OTP
5	Biometric eKYC	PUT	/user/collection/aeps-fingpay/kyc/biometric	Complete one-time eKYC
6	Daily KYC	PUT	/user/collection/aeps-fingpay/kyc/biometric/daily	Daily biometric auth (each day)
7	Cash Withdrawal	POST	/customer/collection/aeps-fingpay/cash-withdrawal/{id}	Withdraw cash
8	Balance Enquiry	POST	/customer/collection/aeps-fingpay/balance-enquiry/{id}	Check balance
9	Mini Statement	POST	/customer/collection/aeps-fingpay/mini-statement/{id}	Get recent transactions

For any queries or support, contact: support@yourcompany.com