API REFERENCE · DEMO TEMPLATE

AePS Cash Deposit API Documentation

Integrate Aadhaar-enabled cash deposit workflows with agent onboarding, KYC, biometric authentication, transaction status, reversal, webhooks and audit-ready responses.

Specification notice. This is a complete documentation portal template. Replace all values marked `[VERIFY]` with your approved live API contract before publishing or using in production.

Quickstart

Use this sequence for a typical AePS cash deposit: register or identify the agent, complete the required daily authentication, capture the customer and transaction details, submit the cash deposit request, and track the final status. Keep every client reference and provider reference in your ledger.

```
curl --request POST \\
  --url https://[SANDBOX_BASE_URL]/v1/aeps/cash-deposit \\
  --header 'Authorization: Bearer [ACCESS_TOKEN]' \\
  --header 'Content-Type: application/json' \\
  --data '{
    "agent_id": "[AGENT_ID]",
    "client_ref": "[UNIQUE_REFERENCE]",
    "customer_id": "[CUSTOMER_ID]",
    "amount": 1000,
    "bank_code": "[BANK_CODE]",
    "biometric_data": "[PID_DATA]"
  }'
```

ON THIS PAGE

EXAMPLE REQUEST

Cash Deposit · cURL

```
curl --request POST \\
  --url https://
[SANDBOX_BASE_URL]/v1/
aeps/cash-deposit \\
  --header
'Authorization: Bearer
[TOKEN]'
```

EXAMPLE RESPONSE

200 Response

Authentication

Send credentials only from a secure server. This template uses bearer authentication; confirm the real token endpoint, header names, signing method, expiry and scope with your API provider.

POST `/v1/auth/token`

Field	Type	Required	Description
`client_id`	`string`	**required**	API client identifier. **[VERIFY]**
`client_secret`	`string`	**required**	Secret kept on your server. **[VERIFY]**

Environments and versioning

Environment	Base URL	Purpose
Sandbox	`https://[SANDBOX_BASE_URL]/v1`	Non-production testing.
Production	`https://[PRODUCTION_BASE_URL]/v1`	Live transaction processing.

Agent onboarding

Register an agent before enabling AePS services. The exact KYC fields depend on the provider and applicable program rules. Store approval status, user code and provider reference.


```
{
  "status":"success",
  "data":{

"transaction_status":"SUC
  }
}
```

POST /v1/agents

```
{
  "initiator_id": "[REGISTERED_MOBILE]",
  "user_code": "[UNIQUE_AGENT_CODE]",
  "customer_id": "[AGENT_MOBILE]",
  "name": "[AGENT_NAME]",
  "kyc_reference": "[KYC_REFERENCE]"
}
```

Daily KYC

Daily authentication may be required before the agent's first AePS transaction of the day. Follow the provider's exact sequence for first-time KYC, OTP, biometric verification and daily re-authentication.

POST `/v1/agents/{agent_id}/kyc/daily`

Field	Type	Required	Description
`initiator_id`	`string`	**required**	Registered initiator mobile number. **[VERIFY]**
`client_ref`	`string`	optional	Unique request reference.
`aadhaar`	`string`	**required**	Use only when allowed; never log raw sensitive values.
`piddata`	`string`	**required**	Encrypted biometric PID XML from an approved device.
`latlong`	`string`	**required**	Agent location in the accepted format.
`bank_code`	`string`	**required**	Bank code from the approved bank list.

```
{
  "status": "success",
  "response_code": "[DAILY_KYC_SUCCESS_CODE]",
  "message": "[VERIFY_LIVE_MESSAGE]",
  "data": {"agent_id":"[AGENT_ID]","authenticated_until":"[TIMESTAMP]"}
}
```

Handle invalid biometric data, expired daily authentication, unsupported device, invalid bank code and duplicate daily KYC according to the live error contract.

Biometric authentication

Capture biometric data only through an approved device and SDK flow. Do not alter, decode, store or expose raw biometric payloads outside the approved secure process.

Cash Deposit

Submit a cash deposit only after validating the agent, customer and bank details. Use an idempotent client reference to prevent duplicate requests.

POST `/v1/aeps/cash-deposit`

Field	Type	Required	Description
`initiator_id`	`string`	**required**	Registered initiator ID or mobile. **[VERIFY]**
`agent_id`	`string`	**required**	Approved AePS agent identifier.
`client_ref`	`string`	**required**	Unique reference generated by your system.
`customer_id`	`string`	**required**	Registered customer or agent mobile as specified by provider.
`aadhaar`	`string`	**required**	Use only as permitted by the live contract and applicable controls.
`amount`	`number`	**required**	Deposit amount within the account's configured limits.
`bank_code`	`string`	**required**	Approved bank code.
`piddata`	`string`	**required**	Encrypted PID data from approved biometric SDK.

```
{
  "initiator_id": "[INITIATOR_ID]",
  "agent_id": "[AGENT_ID]",
  "client_ref": "[UNIQUE_REFERENCE]",
  "customer_id": "[CUSTOMER_ID]",
  "aadhaar": "[PROTECTED_VALUE]",
  "amount": 1000,
  "bank_code": "[BANK_CODE]",
  "piddata": "[ENCRYPTED_PID_DATA]"
}
```

Balance Inquiry

POST `/v1/aeps/balance`

Returns the bank balance response for an authenticated customer or agent. Confirm the identifier and biometric requirements.

```
{
  "client_ref": "[UNIQUE_REFERENCE]",
  "agent_id": "[AGENT_ID]",
  "customer_id": "[CUSTOMER_ID]",
  "bank_code": "[BANK_CODE]",
  "piddata": "[ENCRYPTED_PID_DATA]"
}
```

Mini Statement

POST `/v1/aeps/mini-statement`

Returns recent account entries where supported. Apply masking and access controls to all statement data.

Transaction status

Initial acceptance does not always equal final settlement. Save the transaction ID and client reference, then query status or process a verified webhook.

`GET` `/v1/transactions/{transaction_id}`

```
{
  "status": "success",
  "data": {
    "transaction_id": "[TRANSACTION_ID]",
    "client_ref": "[CLIENT_REF]",
    "transaction_status": "PENDING",
    "provider_reference": "[PROVIDER_REFERENCE]",
    "message": "[VERIFY_LIVE_MESSAGE]"
  }
}
```

Status	Meaning	Recommended action
`SUCCESS`	Provider confirmed the deposit.	Issue receipt and update ledger.
`PENDING`	Final status is not available.	Poll or wait for webhook; do not duplicate.
`FAILED`	Provider rejected or could not complete.	Follow reversal and dispute rules.

Reversal and refund

Keep reversal records separate from the original deposit. Store original transaction ID, reversal reference, amount, reason, status and timestamps. The exact eligibility window and endpoint are provider-specific.

POST /v1/transactions/{transaction_id}/reversal

```
{
  "client_ref": "[REVERSAL_REFERENCE]",
  "reason": "[REVERSAL_REASON]",
  "amount": 1000
}
```

Webhooks and callbacks

Verify webhook signatures, prevent replayed events, acknowledge quickly and process status updates asynchronously. Confirm the signing algorithm and header name before implementation.

EVENT aeps.transaction.updated

```
{
  "event": "aeps.transaction.updated",
  "event_id": "[EVENT_ID]",
  "created_at": "[ISO_8601_TIMESTAMP]",
  "data": {
    "transaction_id": "[TRANSACTION_ID]",
    "transaction_status": "SUCCESS"
  }
}
```

Errors and response codes

Code	Meaning	Client action
400	Invalid request or missing field.	Fix validation and resubmit safely.
401	Invalid authentication.	Refresh credentials and check environment.
403	Service or agent not permitted.	Check onboarding and account scope.
409	Duplicate client reference.	Fetch original transaction result.
429	Rate limit exceeded.	Back off and retry safely.
500	Unexpected provider error.	Use status check and support escalation.

```
{
  "status": "error",
  "error": {
    "code": "[ERROR_CODE]",
    "message": "[HUMAN_READABLE_MESSAGE]",
    "request_id": "[REQUEST_ID]"
  }
}
```

Security and privacy

Use HTTPS, keep credentials server-side, protect biometric payloads, mask sensitive identifiers in logs, restrict access by role, verify callbacks and maintain an audit trail. Do not publish production secrets or raw customer/biometric data in examples.

Sandbox checklist

1. Confirm sandbox base URL and credentials.
2. Test agent onboarding, first-time KYC and daily KYC.
3. Test valid, invalid, duplicate, pending, failed and reversal cases.
4. Verify status polling and webhook signature validation.
5. Check ledger, receipt, settlement and audit entries.
6. Replace every `[VERIFY]` field with approved live values.
7. Use separate production credentials and request go-live approval.

FAQ

Is this documentation ready for production?

The portal structure is ready, but the marked placeholders must be replaced with your real API contract before production use.

Should biometric data be stored?

Follow the approved provider SDK and applicable security and privacy requirements. Do not store or log raw biometric payloads unless your approved design explicitly requires it.

What should happen when a cash deposit is pending?

Keep it pending, query the status endpoint or wait for a verified callback, and prevent duplicate requests using a unique client reference.