

IMT Remittance API — Integration Guide

Moko Afrika · Aggregator-mode integration for MTOs, banks and fintechs

Moko Afrika · Integration Team

2026-08-14

Introduction

- Aggregator model

- What is new in this guide (v2.2)

- Previously new in v2.1

- What was new in v2.0 (2026-06-25)

Base URLs and environments

Authentication

- Basic auth (API endpoints)

- Bearer auth (dashboard / self-service)

Getting started

- Step 1 — Register

- Step 2 — Login (dashboard JWT)

- Step 3 — Top up your wallet

Endpoint reference

- Health check

- Account lookup (mandatory pre-flight)

- Inbound remittance

 - Required fields

 - Optional fields

 - Request example — full-compliance payload

 - Error responses

- Transaction status

- Transaction list

- Transaction timeline

- Balance

- FX rates

 - Current rate card

 - Live rate

- Supported currencies

- KYC verification (phone)

- Outbound (currently disabled)

Webhooks (callback)

- Headers

- Payload

- Signature verification (Python)

- Signature verification (Node.js)

Error codes

Recommended integration flow

- Step-by-step

Operational checklist

Support Changelog

v2.2 — 2026-08-14
v2.1 — 2026-08-13
v2.0 — 2026-06-25
v1.0 — 2026-03-26

Introduction

The IMT Remittance API is the aggregator-mode integration point used by partner MTOs (Money Transfer Operators), banks and fintechs to send international remittances to mobile money wallets in the Democratic Republic of Congo (DRC).

Moko Afrika operates on the ground in DRC and provides the only direct interface to the local mobile money operators (Vodacom M-Pesa, Airtel DRC) for cross-border remittance. Partners outside DRC use this API to disburse funds to DRC beneficiaries without managing the complexity of telco integrations, FX, sanctions screening or compliance reporting.

Aggregator model

This API is **not** a B2C product. The integrating party is always a regulated financial institution that has its own end-user (sender) base. The aggregator model has three actors:

Actor	Role
Sender	End-user in the originating country who initiates the transfer. Outside the scope of this API.
Partner (you)	MTO, bank or fintech. Holds the sender relationship, performs originator KYC, calls this API.
Moko Afrika	Provides the DRC corridor: telco connectivity, FX, sanctions screening, settlement.

Because Moko is the only party with direct access to the DRC telco subscriber registries, certain compliance checks (live beneficiary name lookup, sanctions screening on the beneficiary leg) are performed by Moko and surfaced back to the partner.

What is new in this guide (v2.2)

- **reason_of_transaction enum extended** with 12 new values (INVESTMENT, CONSTRUCTION, RENT, GENERAL, FUEL, REPAIRS, PERSONAL_CARE, FOOD_AND_GROCERIES, TRANSPORT, SHOPPING, ENTERTAINMENT, DONATIONS) to align with Wakapay-style finer-grained purpose categorization. Full enum now 23 values (was 11).
- No other API contract change (auth, endpoint URLs, callback contract, required fields unchanged).

Previously new in v2.1

- **Eight KYC fields promoted from soft-required to hard-required** on /inbound, per BCC RTS Art.66 Instr.15 + FATF R.16:
 - sender_birthplace, sender_document_type (enum), reason_of_transaction (enum), source_of_funds (enum), receiver_birthdate, receiver_document_type, receiver_document_number, and at least one of receiver_current_address / receiver_permanent_address
- sender_birthdate corrected from *optional* to *required* (was already enforced since 2026-07-23, doc was out of sync)
- Two Vodacom Reporting fields documented as *optional* (design intent: cash-counter and web-only channels have no MSISDN): sender_msisdn, reason_details
- Two culture-dependent name fields documented as *optional*: sender_middle_name, receiver_middle_name
- New error codes: RMT-1015 (invalid sender_document_type enum), RMT-1016 (invalid reason_of_transaction enum), RMT-1017 (invalid source_of_funds enum), RMT-1018 (both receiver_*_address missing)

What was new in v2.0 (2026-06-25)

- Mandatory pre-flight /account-lookup step before /inbound
- New required field on /inbound: expected_receiver_name (FATF R.16 compliance)
- New endpoint /transactions/<reference>/timeline for full per-transaction event audit
- Live FX support for sender currencies beyond USD (EUR, NGN, GHS, UGX, CAD)
- Receiver currency on DRC corridor is always USD (Vodacom and Airtel IMT only accept USD)
- New error codes RMT-1005 (missing field) and RMT-4006 (name mismatch)
- Outbound endpoint is explicitly disabled (RMT-9001) until further notice

Base URLs and environments

The API is provided in two environments:

Environment	Base URL	Notes
Sandbox	https://staging.remittance.gofreshpay.com	Default after self-registration. Transaction amount capped at 1 USD.
Production	https://remittance.gofreshpay.com	Activated after a successful Go Live request.

All routes share the prefix `/api/v1/remittance/...` or `/api/v1/merchant/...`.

Authentication

Two authentication mechanisms are used. Pick the right one based on which endpoint you are calling.

Basic auth (API endpoints)

Used for all transaction-flow endpoints: `/inbound`, `/outbound`, `/account-lookup`, `/balance`, `/transactions`, `/transaction/status`, `/transactions/<ref>/timeline`, `/settlement/list`, `/wallet/history`.

Authorization: Basic `base64(merchant_code:merchant_secret)`

The `merchant_code` and `merchant_secret` are returned in the response of `/merchant/register`. The secret is shown only once at registration — store it in your secrets vault immediately. It can be regenerated later via `/merchant/secret` (this invalidates the previous one).

Bearer auth (dashboard / self-service)

Used for dashboard endpoints: `/merchant/me`, `/merchant/profile`, `/merchant/change-password`, `/merchant/go-live`, `/merchant/whitelist`, `/merchant/callback`.

Authorization: Bearer `<access_token>`

Obtain the JWT `access_token` from `POST /merchant/login`. Re-login when expired.

Getting started

Step 1 — Register

Self-service merchant registration. Returns your API credentials (`merchant_code`, `merchant_secret`) and provisions a default M-Pesa USD inbound channel + wallet + dashboard user.

```
curl -X POST https://staging.remittance.gofreshpay.com/api/v1/merchant/register \
-H "Content-Type: application/json" \
-d '{
  "company_name": "Acme Money Transfer Ltd",
  "email":       "integration@acme-mt.example",
  "password":    "a-strong-password-min-8-chars",
```

```
"phone":      "+33123456789",
"country":    "France",
"city":       "Paris"
}'
```

Response (201 Created):

```
{
  "status": "success",
  "message": "Merchant registered successfully",
  "data": {
    "merchant_code": "MKA-XXXX-XXXX",
    "merchant_secret": "sk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    "email": "integration@acme-mt.example",
    "company_name": "Acme Money Transfer Ltd",
    "environment": "sandbox",
    "wallet_code": "wc_mpesa_usd_..."
  }
}
```

Step 2 — Login (dashboard JWT)

Exchange email + password for a JWT. Use this only for dashboard endpoints.

```
curl -X POST https://staging.remittance.gofreshpay.com/api/v1/
      merchant/login \
      -H "Content-Type: application/json" \
      -d '{
        "email": "integration@acme-mt.example",
        "password": "a-strong-password-min-8-chars"
      }'
```

Response (200 OK):

```
{
  "status": "success",
  "message": "Login successful",
  "data": {
    "merchant_code": "MKA-XXXX-XXXX",
    "company_name": "Acme Money Transfer Ltd",
    "email": "integration@acme-mt.example",
    "role": "merchant_admin",
    "environment": "sandbox",
    "country": "France",
    "wallets": [
      {
        "wallet_code": "wc_mpesa_usd_...",
        "currency": "USD",
        "direction": "inbound",
        "operator": "vodacom",
        "balance": 0.0,
        "reserved": 0.0,
      }
    ]
  }
}
```

```
    "available": 0.0
  }
],
  "access_token": "eyJhbGciOiJIUzI1NiIs..."
}
```

Step 3 — Top up your wallet

Before you can send transactions, your inbound wallet must be funded. In sandbox, contact the integration team for a top-up. In production, fund via the agreed nostro/bank wire workflow.

Endpoint reference

Health check

GET /api/v1/remittance/health

No authentication. Use from your monitoring stack.

Response (200 OK):

```
{
  "status": "healthy",
  "service": "remittance-api",
  "checks": { "database": "ok" },
  "timestamp": "2026-06-25T18:14:48.979634"
}
```

Returns HTTP 503 when a dependency is degraded.

Account lookup (mandatory pre-flight)

POST /api/v1/remittance/account-lookup

Basic auth required. Performs a live lookup at the DRC telco for the beneficiary MSISDN, returns the canonical name on file, and compares it against the receiver name your end-user declared.

This call is **read-only**: no debit, no transaction created. It is the recommended (and effectively mandatory) first step before every /inbound call.

Body parameters:

Field	Type	Required	Description
customer_number	string	Yes	Beneficiary MSISDN (243...)

Field	Type	Required	Description
amount	number	Yes	Transaction amount
currency	string	Yes	Source currency mpesa or airtel.
operator	string	No	Auto-detected from MSISDN prefix if omitted.
receiver_first_name	string	No	Used for name_match.
receiver_last_name	string	No	Used for name_match.
sender_first_name	string	No	Required for Vodacom (M-Pesa) lookup.
sender_last_name	string	No	Required for Vodacom (M-Pesa) lookup.
sender_msisdn	string	No	Defaults to customer_number.
sender_country_code	string	No	Sender ISO country code.
sender_birthdate	string	No	YYYY-MM-DD
reference	string	No	Your partner-side lookup id, echoed back.

Request example:

```
curl -X POST https://staging.remittance.gofreshpay.com/api/v1/remittance/account-lookup \
-u 'MKA-XXXX-XXXX:sk_live_XXXXX' \
-H 'Content-Type: application/json' \
-d '{
  "customer_number": "243812345678",
  "amount": 10,
  "currency": "USD",
  "operator": "mpesa",
  "receiver_first_name": "Marie",
  "receiver_last_name": "Kabila",
  "sender_first_name": "John",
  "sender_last_name": "Doe"
}'
```

Response — name matches:

```
{
  "status": "success",
  "data": {
    "valid": true,
    "subscriber_name": "MARIE KABILA",
    "operator": "vodacom",
    "response_code": "INS-GAR-0",
  }
}
```

```

    "response_desc": "Request processed successfully",
    "telco_reference": "DFP163Y...",
    "lookup_reference": "LOOKUP-20260625-A1B2C3D4",
    "reference": null,
    "name_match": {
      "matches": true,
      "score": 1.0,
      "expected": "Marie Kabila",
      "telco": "MARIE KABILA",
      "reason": "matched 2/2 tokens"
    }
  },
  "timestamp": "2026-06-25T14:00:00.000000"
}

```

Response — name mismatch (still 200, MSISDN valid):

```

{
  "status": "success",
  "data": {
    "valid": true,
    "subscriber_name": "MARIE KABILA",
    "name_match": {
      "matches": false,
      "score": 0.0,
      "expected": "John Smith",
      "telco": "MARIE KABILA",
      "reason": "insufficient match: 0/2 matched,
        unmatched=['john', 'smith']"
    }
  }
}

```

Name match — fuzzy rules:

Rule	Behaviour
Case	Case-insensitive (Henock = HENOCK).
Accents	Accent-folded (h��lo��se = HEL��ISE).
Punctuation	Stripped.
Order	Order-independent (Henock Barakael = BARAKAEL HENOCK).
Middle names	Up to 1 missing token tolerated.
Typos	Levenshtein distance ≤ 2 per token (Henoq $\approx$ HENOCK).

Decision matrix:

valid	matches	Recommended action
true	true	Proceed to /inbound. Reuse subscriber_name as expected_receiver_name.

valid	matches	Recommended action
true	false	Show the canonical telco name to the user, obtain re-confirmation, then send the confirmed name on /inbound.
true	null	You did not send receiver_first_name + receiver_last_name. Comparison skipped.
false	—	Do not attempt /inbound. The MSISDN is invalid or the account was rejected.

Inbound remittance

POST /api/v1/remittance/inbound

Basic auth required. Submits a money transfer to a beneficiary's DRC mobile money wallet. Returns HTTP 202 with an acknowledgement. The final status (Successful / Failed) is delivered later via webhook.

Required fields

All fields below are hard-required. A missing value returns RMT-1005 MISSING_FIELD (or RMT-1018 for the address either-or).

Transaction identity and routing:

Field	Type	Description
amount	number	Transaction amount in source currency.
currency	string	Source currency. See "Supported currencies". Receiver always gets USD on DRC corridor.
customer_number	string	Beneficiary MSISDN (243...).
reference	string	Your unique transfer reference (max 80 chars).
operator	string	mpesa, airtel, orange, afrimoney, or fbnbank (cash pickup).
expected_receiver_name	string	Receiver name confirmed with your end-user. Fuzzy-matched against the telco record (v2.0).

Sender KYC — FATF R.16 + BCC RTS Art.66:

Field	Type	Description
sender_first_name	string	Legal first name as on the sender's identity document.
sender_last_name	string	Legal last name as on the sender's identity document.
sender_nationality	string	Sender nationality.
sender_birthdate	string (ISO date)	Date of birth, YYYY-MM-DD.
sender_birthplace	string	City of birth.
sender_document_type	string (enum)	One of: PASSPORT, NATIONAL_ID, DRIVING_LICENCE, VOTER_CARD, OTHER. Invalid values return RMT-1015.
identity_number	string	Sender ID / passport number matching sender_document_type.
sending_country	string	ISO-3166-1 alpha-2 country code of the sending country.

Receiver KYC — FATF R.16 + BCC RTS Art.66:

Field	Type	Description
receiver_first_name	string	Legal first name.
receiver_last_name	string	Legal last name.
receiver_nationality	string	Receiver nationality.
receiver_birthdate	string (ISO date)	YYYY-MM-DD.
receiver_document_type	string (enum)	One of: PASSPORT, NATIONAL_ID, DRIVING_LICENCE, VOTER_CARD. Required for all operators, not just cash pickup.
receiver_document_number	string	Receiver ID / passport number.
receiving_country	string	ISO country code (typically CD for DRC).
receiver_current_address or receiver_permanent_address	string	At least one must be provided. Missing both returns RMT-1018.

Transaction context — FATF R.16:

Field	Type	Description
reason_of_transaction	string (enum)	One of: FAMILY_SUPPORT, EDUCATION, MEDICAL, BUSINESS_PAYMENT, SALARY, GIFT, SAVINGS, PROPERTY_PURCHASE, TRAVEL, LOAN_REPAYMENT, INVESTMENT, CONSTRUCTION, RENT, GENERAL, FUEL, REPAIRS, PERSONAL_CARE, FOOD_AND_GROCERIES, TRANSPORT, SHOPPING, ENTERTAINMENT, DONATIONS, OTHER (23 values). Invalid values return RMT-1016.
source_of_funds	string (enum)	One of: SALARY, BUSINESS_INCOME, SAVINGS, INVESTMENT, PENSION, GIFT, LOAN, SALE_OF_ASSET, INHERITANCE, OTHER. Invalid values return RMT-1017.

Optional fields

Field	Type	Description
sender_middle_name	string	Culture-dependent; send when the sender has a middle name on their ID.
receiver_middle_name	string	Culture-dependent; send when the receiver has a middle name on their ID.
sender_msisdn	string	The originating sender's own MSISDN (not the beneficiary's). Send when the sender initiated the transfer from a mobile wallet (Wave, Orange Money app, etc.). Cash-counter (Western Union guichet) and web-only channels have no MSISDN — leave empty. Used by Vodacom Reporting.
reason_details	string	Free-text qualifier of reason_of_transaction. Example: reason_of_transaction=EDUCATION, reason_details="tuition for Nairobi university".
sender_city	string	Sender city of residence (in addition to birthplace).

Field	Type	Description
current_address, permanent_address	string	Sender addresses (distinct from the <i>receiver</i> addresses above).
profession	string	Sender profession.
receiver_birthplace	string	Receiver city of birth.
document_issue_date, document_expiration_date	string (ISO date)	Receiver document dates.
provider_name	string	Sender-side provider label if useful for your reconciliation.
callback_url	string	Per-request webhook URL override. If unset, uses the merchant-channel default.

Request example — full-compliance payload

```
curl -X POST https://staging.remittance.gofreshpay.com/api/v1/
remittance/inbound \
-u 'MKA-XXXX-XXXX:sk_live_XXXXX' \
-H 'Content-Type: application/json' \
-d '{
  "amount": 10,
  "currency": "USD",
  "customer_number": "243812345678",
  "reference": "TXN_001",
  "operator": "mpesa",
  "expected_receiver_name": "Marie Kabila",

  "sender_first_name": "John",
  "sender_last_name": "Doe",
  "sender_nationality": "Ugandan",
  "sender_birthdate": "1985-04-12",
  "sender_birthplace": "Kampala",
  "sender_document_type": "PASSPORT",
  "identity_number": "UG123456789",
  "sending_country": "UG",

  "receiver_first_name": "Marie",
  "receiver_last_name": "Kabila",
  "receiver_nationality": "Congolaise",
  "receiver_birthdate": "1990-08-22",
  "receiver_document_type": "NATIONAL_ID",
  "receiver_document_number": "CD-NAT-990822-4567",
  "receiver_current_address": "Avenue Kabambare 12, Gombe,
    Kinshasa",
  "receiving_country": "CD",

  "reason_of_transaction": "FAMILY_SUPPORT",
  "source_of_funds": "SALARY",

  "sender_msisdn": "256770000001",
```

```
    "reason_details": "monthly family support"
  }
}
```

Response (202 Accepted):

```
{
  "status": "accepted",
  "message": "Transaction received and is being processed. Final
             status will be sent via callback.",
  "data": {
    "transaction_id": "IMT-IN-20260813-A1B2C3-D4E5",
    "reference": "TXN_001",
    "amount": 10,
    "currency": "USD",
    "operator": "mpesa",
    "customer_number": "243812345678",
    "subscriber_name": "MARIE KABILA",
    "created_at": "2026-08-13T14:00:00.000000"
  },
  "timestamp": "2026-08-13T14:00:00.000000"
}
```

Error responses

400 — RMT-1005 missing required field (any of the required fields above):

```
{
  "status": "error",
  "error": {
    "code": "RMT-1005",
    "message": "Required field missing: sender_birthplace",
    "field": "sender_birthplace"
  },
  "timestamp": "2026-08-13T14:00:00.000000"
}
```

For expected_receiver_name specifically, the response also includes a telco_name_hint so you can retry with the right name without re-calling / account-lookup:

```
{
  "status": "error",
  "error": {
    "code": "RMT-1005",
    "field": "expected_receiver_name",
    "message": "Required field missing: expected_receiver_name.
               Aggregator KYC (FATF R.16) requires the merchant to
               declare the receiver name confirmed with the end-user.",
    "telco_name_hint": "MARIE KABILA"
  },
  "timestamp": "2026-08-13T14:00:00.000000"
}
```

400 — RMT-1015 / RMT-1016 / RMT-1017 invalid enum value:

```
{
  "status": "error",
  "error": {
    "code": "RMT-1016",
    "field": "reason_of_transaction",
    "message": "reason_of_transaction must be one of
      ['BUSINESS_PAYMENT', 'CONSTRUCTION', 'DONATIONS',
      'EDUCATION', 'ENTERTAINMENT', 'FAMILY_SUPPORT',
      'FOOD_AND_GROCERIES', 'FUEL', 'GENERAL', 'GIFT',
      'INVESTMENT', 'LOAN_REPAYMENT', 'MEDICAL', 'OTHER',
      'PERSONAL_CARE', 'PROPERTY_PURCHASE', 'RENT', 'REPAIRS',
      'SALARY', 'SAVINGS', 'SHOPPING', 'TRANSPORT', 'TRAVEL']"
  },
  "timestamp": "2026-08-14T14:00:00.000000"
}
```

Same shape for RMT-1015 (sender_document_type) and RMT-1017 (source_of_funds).

400 — RMT-1018 both receiver addresses missing:

```
{
  "status": "error",
  "error": {
    "code": "RMT-1018",
    "field": "receiver_current_address|
      receiver_permanent_address",
    "message": "At least one of receiver_current_address or
      receiver_permanent_address is required"
  },
  "timestamp": "2026-08-13T14:00:00.000000"
}
```

422 — RMT-4006 receiver name mismatch:

```
{
  "status": "error",
  "error": {
    "code": "RMT-4006",
    "message": "Receiver name mismatch with telco record",
    "expected": "John Smith",
    "telco": "MARIE KABILA",
    "score": 0.0,
    "reason": "insufficient match: 0/2 matched,
      unmatched=['john', 'smith']"
  },
  "timestamp": "2026-08-13T14:00:00.000000"
}
```

Transaction status

GET /api/v1/remittance/transaction/status?reference=TXN_001

Basic auth required. Returns the current state of a transaction by reference (your merchant ref) or transaction_id (Moko internal ref).

Response:

```
{
  "status": "success",
  "data": {
    "transaction_id": "IMT-IN-20260625-A1B2C3-D4E5",
    "reference": "TXN_001",
    "amount": 10.00,
    "currency": "USD",
    "operator": "vodacom",
    "customer_number": "243812345678",
    "transaction_status": "Successful",
    "status_description": "Request processed successfully",
    "switch_reference": "MOL26062500FF12AB34CD",
    "telco_reference": "DFP163Y...",
    "created_at": "2026-06-25T14:00:00",
    "updated_at": "2026-06-25T14:00:07"
  }
}
```

Transaction list

GET /api/v1/remittance/transactions?
page=1&limit=50&status=Successful&action=inbound

Basic auth required. Paginated list.

Query parameters: page (default 1), limit (default 50, max 100), status (optional filter: Successful, Failed, Pending, Submitted, Rejected), action (optional filter: inbound, outbound).

Response:

```
{
  "status": "success",
  "data": {
    "transactions": [ /* ... */ ],
    "total": 1234,
    "page": 1,
    "limit": 50,
    "pages": 25
  }
}
```

Transaction timeline

GET /api/v1/remittance/transactions/<reference>/timeline

Basic auth required. Returns the full per-transaction event chronology: received, sanctions screening, FX, telco subscriber validate, switch dispatch, telco callback. Same data as the dashboard drawer “Timeline” tab.

Response:

```
{
  "status": "success",
  "data": {
    "summary": {
      "transaction_reference": "IMT-IN-20260625-A1B2C3-D4E5",
      "thirdparty_reference": "TXN_001",
      "operator": "vodacom",
      "amount": 10.00,
      "currency": "USD",
      "status": "Successful",
      "duration_seconds": 7.0
    },
    "events": [
      { "kind": "received", "title": "Inbound received",
        "status": "ok", "ts": "2026-06-25 14:00:00", "detail": { /
        * ... */ } } },
      { "kind": "screening", "title": "Sanctions screening",
        "status": "ok", "ts": "2026-06-25 14:00:00", "detail": { /
        * ... */ } } },
      { "kind": "telco_validate", "title": "Vodacom subscriber
        validate",
        "status": "ok", "ts": "2026-06-25 14:00:01",
        "detail": { "subscriber_name": "MARIE KABILA", "outcome":
        "PASSED" } } },
      { "kind": "switch_send", "title": "Submitted to Vodacom
        switch",
        "status": "ok", "ts": "2026-06-25 14:00:03",
        "detail": { "switch_reference": "MOL2606..." } } },
      { "kind": "telco_callback", "title": "Telco callback –
        Successful",
        "status": "ok", "ts": "2026-06-25 14:00:07",
        "detail": { "telco_reference": "DFP163Y..." } }
    ]
  }
}
```

Event kinds: received, screening, compliance_hold, fx, pricing, wallet_reserve, wallet_release, wallet_debit, telco_validate, switch_send, telco_callback.

Event status: ok, warn, fail, info.

Balance

GET /api/v1/remittance/balance

Basic auth required.

Response:

```
{
  "status": "success",
  "data": {
    "merchant_code": "MKA-XXXX-XXXX",
    "wallets": [
      {
        "wallet_code": "wc_mpesa_usd_...",
        "currency": "USD",
        "operator": "vodacom",
        "direction": "inbound",
        "balance": 5000.00,
        "reserved": 0.00,
        "available": 5000.00,
        "last_updated": "2026-06-25T14:00:00"
      }
    ]
  }
}
```

FX rates

Current rate card

GET /api/v1/remittance/rate?currency=CDF

No auth required. Returns the published rate card (cached, 6h validity).

```
{
  "status": "success",
  "data": {
    "currency": "CDF",
    "currency_pair": "USD/CDF",
    "buy_rate": 2261.51,
    "sell_rate": 2342.07,
    "spread": 80.56,
    "rate_card_id": "RC-20260625-1701",
    "confidence": "high",
    "expired": false,
    "fetched_at": "2026-06-25T17:01:00",
    "valid_from": "2026-06-25T17:01:00",
    "valid_until": "2026-06-25T23:01:00",
    "validity_hours": 6
  }
}
```

```
}  
}
```

Live rate

GET /api/v1/remittance/rate/live?currency=NGN

No auth required. Bypasses cache, hits the upstream FX provider. Returns rate metadata, market data and the margin applied.

Supported currencies

The receiver on the DRC IMT corridor (Vodacom M-Pesa or Airtel DRC) **always receives USD**. CDF is not a valid receiver currency on the IMT rail.

Supported sender-side currencies (the currency field on /inbound):

Code	Currency	FX behaviour
USD	US Dollar	Pass-through (1:1, no conversion)
EUR	Euro	Live FX vs USD
NGN	Nigerian Naira	Live FX vs USD
GHS	Ghanaian Cedi	Live FX vs USD
UGX	Ugandan Shilling	Live FX vs USD
CAD	Canadian Dollar	Live FX vs USD

KYC verification (phone)

GET /api/v1/remittance/kyc?phone=243812345678&operator=mpesa

No auth required. Verifies a beneficiary phone number — format, operator resolution, blacklist check.

```
{  
  "status": "success",  
  "data": {  
    "phone_normalized": "243812345678",  
    "operator": "mpesa",  
    "resolved_operator": "vodacom",  
    "blacklisted": false  
  }  
}
```

Outbound (currently disabled)

POST /api/v1/remittance/outbound

This endpoint exists but is not yet open to merchants. Calls return HTTP 403 with code RMT-9001:

```
{
  "status": "error",
  "error": {
    "code": "RMT-9001",
    "message": "Outbound remittance is not yet available. Only
               inbound transfers are currently supported."
  }
}
```

When enabled, the request and response shapes will mirror `/inbound`, with the same KYC and name-check requirements.

Webhooks (callback)

When a transaction reaches a final state (Successful, Failed, Rejected), Moko POSTs a callback to the URL you registered on your channel (via the dashboard or via `PUT /merchant/callback`).

Headers

Header	Description
X-Signature	HMAC-SHA256 signature: <code>sha256=HMAC(timestamp.body, webhook_secret)</code>
X-Timestamp	Unix timestamp when the callback was sent
Content-Type	<code>application/json</code>

Payload

```
{
  "event": "transaction.completed",
  "data": {
    "transaction_id": "IMT-IN-20260625-A1B2C3-D4E5",
    "reference": "TXN_001",
    "amount": 10.00,
    "currency": "USD",
    "operator": "mpesa",
    "status": "Successful",
    "customer_number": "243812345678",
    "action": "inbound",
    "completed_at": "2026-06-25T14:00:07Z"
  },
  "timestamp": "2026-06-25T14:00:07Z"
}
```

Your server must respond HTTP 200.

Signature verification (Python)

```
import hmac, hashlib

def verify_webhook(request):
    signature = request.headers.get("X-Signature", "")
    timestamp = request.headers.get("X-Timestamp", "")
    body      = request.body.decode("utf-8")
    message   = f"{timestamp}.{body}"
    expected  = "sha256=" + hmac.new(
        WEBHOOK_SECRET.encode("utf-8"),
        message.encode("utf-8"),
        hashlib.sha256,
    ).hexdigest()
    return hmac.compare_digest(signature, expected)
```

Signature verification (Node.js)

```
const crypto = require('crypto');

function verifyWebhook(req) {
    const signature = req.headers['x-signature'] || '';
    const timestamp = req.headers['x-timestamp'] || '';
    const body      = JSON.stringify(req.body);
    const message   = timestamp + '.' + body;
    const expected  = 'sha256=' + crypto
        .createHmac('sha256', WEBHOOK_SECRET)
        .update(message)
        .digest('hex');
    return crypto.timingSafeEqual(
        Buffer.from(signature),
        Buffer.from(expected),
    );
}
```

Error codes

All errors share a consistent envelope:

```
{
  "status": "error",
  "error": { "code": "RMT-XXXX", "message": "..." },
  "timestamp": "2026-06-25T14:00:00"
}
```

Code	HTTP	Description
RMT-1001	400	Validation error: invalid amount / currency / reference / operator

Code	HTTP	Description
RMT-1005	400	Required field missing — response includes the field name and a telco_name_hint when available
RMT-1006	400	Invalid phone number format for the operator
RMT-2001	404	Merchant identifier not recognized
RMT-2003	403	Merchant not authorized for this operation
RMT-2004	403	Request origin IP not whitelisted
RMT-3001	402	Insufficient wallet balance
RMT-3003	409	Duplicate reference
RMT-4001	503	Service temporarily unavailable (database)
RMT-4002	502	Switch / payment processor communication error
RMT-4003	500	Internal error
RMT-4005	422	Telco pre-validation failed (invalid MSISDN, closed account, AML, etc.)
RMT-4006	422	Receiver name mismatch with telco record. Response includes expected, telco, score, reason for re-prompt
RMT-5001	403	Phone number blacklisted
RMT-5002	403	Sender matched on sanctions list
RMT-9001	403	Outbound endpoint disabled — only inbound transfers are currently supported

Recommended integration flow

```

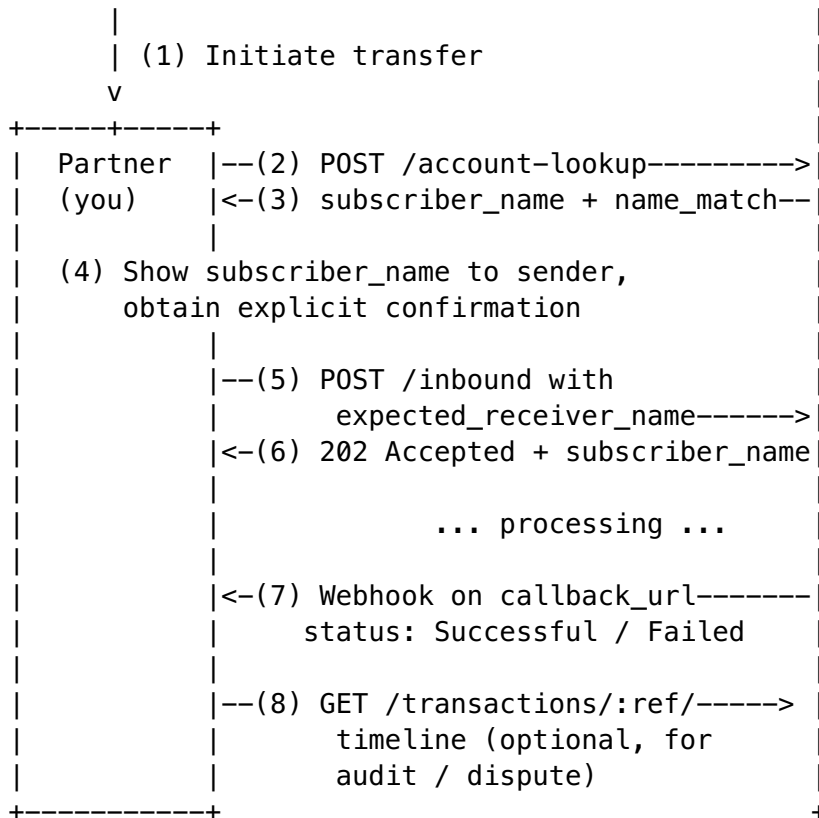
+-----+
| Sender |
| (end-  |
| user)  |
+-----+

```

```

+-----+
| Moko Afrika |
| remittance- |
| api         |
+-----+

```



Step-by-step

1. **Sender initiates** a transfer in your interface.
2. You call POST /account-lookup with the receiver MSISDN and the names your end-user declared.
3. Moko returns the canonical subscriber_name from the DRC telco + a fuzzy name_match block.
4. You display the canonical name to your end-user and obtain explicit confirmation that this is the intended beneficiary.
5. You call POST /inbound with expected_receiver_name set to the confirmed name.
6. Moko returns 202 with a transaction_id and the canonical subscriber_name for your logs.
7. Moko POSTs the final webhook to your registered callback_url once the telco confirms or rejects.
8. (Optional) You can fetch the full event timeline with GET /transactions/<reference>/timeline for audit and dispute resolution.

Operational checklist

Before going live:

☐

API credentials (merchant_code + merchant_secret) stored in a secret manager, not in code or env files

☐

Webhook URL configured per channel (M-Pesa, Airtel)

- ☐ Webhook signature verification implemented and tested
- ☐ HTTPS endpoint with valid TLS for receiving callbacks
- ☐ IP whitelist configured for outbound calls (see `POST /merchant/whitelist`)
- ☐ `/health` endpoint integrated in your monitoring (probe every 30–60 s)
- ☐ `/account-lookup` step integrated **before every** `/inbound`
- ☐ `expected_receiver_name` propagated from the sender confirmation UI
- ☐ Mismatch flow: when `RMT-4006` is returned, your UI re-prompts with the canonical telco name
- ☐ Wallet funded (test in sandbox with small amounts, then top up in production)
- ☐ Go Live request submitted via `POST /merchant/go-live` once integration testing is complete

Support

Contact	Use case
WhatsApp integration thread	Day-to-day integration questions, sandbox top-ups
<code>integration@gofreshpay.com</code>	Onboarding, contractual, Go Live, KYC compliance
Sandbox docs portal	<code>https://sandbox.gofreshpay.com/remittance-api</code>
Merchant dashboard docs	<code>https://dashboard.remittance.gofreshpay.com/docs</code>
Postman collection	<code>IMT_Remittance_Merchant.postman_collection.json</code> (distributed separately)

Changelog

v2.2 — 2026-08-14

- Extended `reason_of_transaction` enum with 12 new values (INVESTMENT, CONSTRUCTION, RENT, GENERAL, FUEL, REPAIRS, PERSONAL_CARE, FOOD_AND_GROCERIES, TRANSPORT, SHOPPING, ENTERTAINMENT, DONATIONS) to align with the finer-grained Wakapay `purposeOfTransfer` set. Full enum now 23 values (was 11).
- No other API contract change.

v2.1 — 2026-08-13

- **Hard-required KYC fields on /inbound** (BCC RTS Art.66 Instr.15 + FATF R.16): sender_birthplace, sender_document_type (enum), reason_of_transaction (enum), source_of_funds (enum), receiver_birthdate, receiver_document_type, receiver_document_number, and either-or receiver_current_address / receiver_permanent_address
- Corrected: sender_birthdate moved from *optional* to *required* (was actually enforced since 2026-07-23; doc was out of sync)
- Documented as *optional*: sender_msisdn, reason_details, sender_middle_name, receiver_middle_name
- New error codes: RMT-1015 (invalid sender_document_type enum), RMT-1016 (invalid reason_of_transaction enum), RMT-1017 (invalid source_of_funds enum), RMT-1018 (both receiver_*_address missing)
- No change to authentication, endpoint URLs, or the callback contract.

v2.0 — 2026-06-25

- Added: mandatory expected_receiver_name on /inbound (FATF R.16)
- Added: live name match in /account-lookup response (name_match block)
- Added: GET /transactions/<ref>/timeline endpoint
- Added: multi-currency sender support (EUR, NGN, GHS, UGX, CAD in addition to USD)
- Added: GET /remittance/rate/live endpoint
- Added: error codes RMT-1005, RMT-4006, RMT-9001
- Changed: /outbound explicitly disabled with RMT-9001
- Changed: receiver currency on DRC corridor confirmed as USD-only

v1.0 — 2026-03-26

- Initial release: /inbound, /balance, /transactions, /transaction/status, /rate, webhooks