GEEPAY

Effortless Payments

GEEPAY API DOCUMENTATION

Introduction

Our comprehensive API suite enables you to process payments, manage checkouts, and verify account information seamlessly. All APIs use REST architecture, return JSON responses, and require authentication using OAuth 2.0 client credentials.

API Versioning

All endpoints except "https://api.example.com/oauth/token" are prefixed with `/api/v1` to support future versioning.

Authentication

All API requests must be authenticated using OAuth 2.0 client credentials flow:

Step 1: Obtain an access token

Endpoint: `POST /oauth/token`

Headers:

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

Accept: application/json
```

Request Parameters:

```
grant_type=client_credentials

 client_id=YOUR_CLIENT_ID

client_secret=YOUR_CLIENT_SECRET
```

Example Request:

```
curl -X POST https://api.example.com/oauth/token \
 -H "Content-Type: application/x-www-form-urlencoded" \
  -H "Accept: application/json" \
  -d
'grant_type=client_credentials&client_id=YOUR_CLIENT_ID&client_secret=YO
UR_CLIENT_SECRET'
```

Example Response:

```
{
  "token_type": "Bearer",
  "expires_in": 31536000,
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9..."
}
```

Step 2: Include the access token in all API requests

```
Authorization: Bearer YOUR_ACCESS_TOKEN
X-Client-ID: YOUR_CLIENT_ID
Content-Type: application/json
Accept: application/json
```

Collection API

The Collection API enables you to accept payments through mobile money. It handles the entire payment lifecycle from creation to confirmation, with support for status checking and name lookups.

Note: Collection transactions are processed **asynchronously**. When you initiate a collection, the API returns immediately with a pending status, and you'll receive the final result via your callback URL or by checking the transaction status.

Request to Pay

Initiate a payment collection request.

Endpoint: `POST /api/v1/mobile-money/collect`

Headers:

```
Authorization: Bearer YOUR_ACCESS_TOKEN
X-Client-ID: YOUR_CLIENT_ID
X-CALLBACK-URL: CALLBACK_URL
X-Transaction-Ref:UNIQUE_TRANSACTION_ID
```

```
Content-Type: application/json

Accept:  application/json
```

Request Parameters:

```
{
  "phone_number": "260972439891", // Required: The mobile number to collect
payment from. The mobile number must be 12 digits long.
  "amount": 10                     // Required: The amount to collect
}
```

Pending Response (202):

```
{

  "code": 200,
  "status": "pending",
  "message": "Payment request sent. Awaiting customer action.",
  "data":
   {
    "transaction_reference": " eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIVRExgyr…",
      }

}
```

Success Response (200):

```
{
  "code": 200,
  "status": "success",
  "message": "Payment was successful.",
  "data":
 {
    "transaction_reference": "278bedc0-d292-42e1-8dd3-ca91d63fa302",
    "external_reference": "000799895807"
  }
}
```

Failed Response (402):

```
{
  "code": 402,
  "status": "failed",
  "message": "Payment failed: low balance or payee limit reached or not allowed",
"data":
{
    "transaction_reference": "001943a1-a594-4003-941d-27abfb60a2f4"
  }
}
```

Check Transaction Status

Verify the status of a payment collection transaction.

Endpoint: `GET /api/v1/mobile-money/check-status/{transactionRef}`

Headers:

```
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json Accept: application/json
```

Success Response (200):

```
{
  "code": 200,
  "status": "success",
  "message": "Transaction fetched successfully.",
  "data": {
    "status": "successful",
    "message": "Transaction was processed successfully.",
    "transaction_reference": "278bedc0-d292-42e1-8dd3-ca91d63fa302",
    "external_reference": "000799895807"
  }
}
```

Not Found Response (404):

```
{
  "code": 404,
  "status": "error",
  "message": "Transaction not found."
}
```

Name Lookup

Verify a mobile number's account holder name before processing a transaction.

Endpoint: `GET /api/v1/mobile-money/name-lookup/{phone}`

Headers:

```
Authorization: Bearer YOUR_ACCESS_TOKEN

Content-Type: application/json

Accept: application/json
```

Response:

```
{
  "code": 200,
  "status": "success",
  "message": "Name lookup completed successfully.",
"data":
{
    "status": "success",
    "provider": "MTN",
    "phone_number": "260765631424",
    "names": "Chisha Chipili"
  }
}
```

Disbursement API

The Disbursement API facilitates sending money to mobile wallets. Perfect for marketplace payouts, affiliate commissions, it offers both individual payout options with comprehensive status tracking.

Note: Disbursement transactions are processed **synchronously**. When you initiate a disbursement, the API processes the transaction immediately and returns the final result in the response.

Disburse Transaction

Send funds to a mobile money account.

Endpoint: `POST /api/v1/mobile-money/disburse`

Headers:

```
Authorization: Bearer YOUR_ACCESS_TOKEN
```

```
Content-Type: application/json
X-Client-ID: YOUR_CLIENT_ID
X-Auth-Signature: YOUR_AUTH_SIGNATURE
X-Transaction-Ref: UNIQUE_TRANSACTION_ID
X-Callback-URL: YOUR_CALLBACK_URL
Accept: application/json
```

Request Parameters:

```
{
  "phone_number": 260765631424,   // Required: The mobile number to send funds to.
Mobile number must be 12 digits long.
  "amount": 0.5,                  // Required: The amount to disburse
"narration": "test transaction"  // Optional: Description of the transaction
}
```

Success Response (200):

```
{
  "code": 200,
  "status": "successful",
  "message": "Disbursement was processed successfully",
  "data": {
    "transaction_id": "8c404fd1-6c2e-46b4-a6ed-8210e78fc9cc",
    "external_reference": "000799902816"
  }
}
```

Insufficient Balance Response (400):

```
{
  "code": 400,
  "status": "error",

 "message": "Insufficient disbursement balance, please top up"

}
```

Check Disbursement Status

Check the status of a disbursement transaction.

Endpoint: `GET /api/v1/mobile-money/disburse/status/{transactionRef}`

Headers:

```
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json
X-Client-ID: YOUR_CLIENT_ID
Accept: application/json
```

Response:

```
{
  "code": 200,
  "status": "success",
  "message": "Transaction fetched successfully.",
 "data":
   {
"status":
"successful
",
    "message": "Funds disbursed successfully.",
    "transaction_reference":"f8775c84-9d03-413d-apdb-68624ce84abf",
      "external_reference":"disburs-0CDYKRN7PB-f8775c84-9d03-413d-a94b-
      68624ce84abf "
   }
}
```

Check Disbursement Balance

Check your disbursement wallet balance.

Endpoint: `GET /api/v1/mobile-money/disburse/balance`

Headers:

```
Authorization: Bearer YOUR_ACCESS_TOKEN
X-Client-ID: YOUR_CLIENT_ID
X-Auth-Signature: YOUR_AUTH_SIGNATURE

Accept: application/json Content-Type:

application/json
```

Response:

```
{
  "status": "success",
  "message": " Funds disbursed successfully.",
  "data": {
    "merchant": "chin",
    "balance": "996.92",
    "currency": "ZMW",
    "last_updated": "2025-04-09 09:48:29"
  }
}
```

Hosted Checkout API (HPP)

The Hosted Checkout API provides a secure, PCI-compliant payment experience without handling sensitive payment data on your servers. It creates customizable checkout sessions with support for success, failure, and cancel redirects.

Create Checkout Session

Create a new checkout session for your customer. If optional fields are not included as request parameters, they will be editable during checkout. This allows users to modify values like the amount if it's not fixed.

Endpoint: `POST /api/v1/checkout/session`

Headers:

```
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json
X-Client-ID: YOUR_CLIENT_ID
X-Transaction-Ref: UNIQUE_TRANSACTION_ID
X-Callback-URL: YOUR_CALLBACK_URL
Accept: application/json
```

Request Parameters:

```
{
  "amount": 150,// Optional
  "order_id": "ORD-789",// Optional
  "customer":
      {
      // Optional: Customer information
  "name": "Jane Doe",
  "email": "jane@example.com"
    },
    "return_url": "{{callback_url}}", // Required: URLs to redirect after payment
```

```
     "receipt_redirect": true // Required.
  }
}
```

Response:

```
{
  "status": "success",
  "message": "Checkout session created.",
  "checkout_url": "http://localhost:8000/checkout/3eb251c1-e36b-43a3-
8fbf-547ca95a9900"
}
```

Webhooks & Callbacks

When you integrate with our API, you can provide a callback URL to receive automatic notifications about the status of your transactions. This allows your system to update in real-time without needing to poll our API for updates.

Setting Up Callbacks

For endpoints that support callbacks, include the `X-Callback-URL` header in your request:

```
X-Callback-URL: https://yourserver.com/webhook/payment-callback
```

Callback Data Format

Your webhook endpoint will receive POST requests with JSON bodies containing information about the transaction.

Collection API Callbacks

Successful Collection

```
{
  "code": 200,
  "status": "successful",
  "message": "Transaction has been successfully processed and settled.",
  "data":
     {
    "transaction_reference": "278bedc0-d292-42e1-8dd3-ca91d63fa302",
    "external_reference": "000799895807",
    "customer": "260952867018",
    "amount": "1.00"
  }
}
```

Failed Collection

```
{
  "code": 402,
  "status": "failed",
  "message": "Transaction failed. Please try again.",
  "data":
  {
    "transaction_reference": "001943a1-a594-4003-941d-27abfb60a2f4",
    "external_reference": null,
    "customer": "260765631424",
    "amount": "10000.00"
  }
}
```

Disbursement API Callbacks

Successful Disbursement

```
{
  "code": 200,
  "status": "successful",
  "message": "Disbursement has been successfully processed and settled.",
"data":
     {
    "transaction_reference": "b075f74c-dce3-4e5e-9e16-4ba29e558627",
    "external_reference": "000798893390",
    "customer": "260952867018",
    "amount": "0.50"
  }
}
```

Failed Disbursement

```
{
  "code": 402,
  "status": "failed",
  "message": "Transaction failed. Please try again.",
  "data":
  {
    "transaction_reference": "001943a1-a594-4003-941d-27abfb60a2f4",
    "external_reference": null,
    "customer": "260765631424",
    "amount": "10000.00"
  }

}
```

1. **Store Credentials Securely**: Never hardcode your client ID and secret in your application code.
2. **Use HTTPS**: All API requests should be made over HTTPS to ensure data encryption.

Error Codes

Code	**Description**
200	Success - The request was processed successfully
202	Accepted - The request was accepted and is being processed
400	Bad Request - The request could not be understood or was missing required parameters
401	Unauthorized - Authentication failed or user does not have permissions
402	Payment Failed - The payment could not be processed due to payment issues
404	Not Found - The requested resource could not be found
409	Conflict - The request could not be completed due to a conflict with the current state of the resource
429	Too Many Requests - Rate limit has been exceeded
500	Server Error - An error occurred on the server

Support

For technical support or any questions regarding our API, please contact our support team at:

- Email: chisha@geepay.co.com
- Phone: +260972439891