

Transaction History

Author: Josh S. Sakweli, Backend Lead Team

Last Updated: 2025-10-02

Version: v1.0

Base URL: `https://apinexgate.globeauth.com/api/v1/transaction-history`

Short Description: The Transaction History API provides user-facing records of all financial activities within the platform. It tracks purchases, refunds, wallet operations, sales earnings, and platform fees with complete details for each transaction. All transactions are linked to underlying ledger entries for full auditability.

Hints:

- All monetary values are in TZS (Tanzanian Shillings)
- Each transaction has a unique reference number (format: #YYYYTNNNNNN)
- Transactions are read-only - they cannot be modified or deleted
- DEBIT transactions represent money leaving the user's wallet (negative)
- CREDIT transactions represent money entering the user's wallet (positive)
- All transactions are linked to ledger entries for audit purposes
- Transactions are automatically created by the system during financial operations
- Pagination is supported with default 20 items per page
- Transactions can be filtered by type, direction, and date range

Standard Response Format

All API responses follow a consistent structure using our Globe Response Builder pattern:

Success Response Structure

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2025-10-02T10:30:45",
  "data": {
    // Actual response data goes here
  }
}
```

```
}  
}
```

Error Response Structure

```
{  
  "success": false,  
  "httpStatus": "BAD_REQUEST",  
  "message": "Error description",  
  "action_time": "2025-10-02T10:30:45",  
  "data": "Error description"  
}
```

Standard Response Fields

Field	Type	Description
success	boolean	Always <code>true</code> for successful operations, <code>false</code> for errors
httpStatus	string	HTTP status name (OK, CREATED, BAD_REQUEST, NOT_FOUND, etc.)
message	string	Human-readable message describing the operation result
action_time	string	ISO 8601 timestamp of when the response was generated
data	object/string	Response payload for success, error details for failures

HTTP Method Badge Standards

For better visual clarity, all endpoints use colored badges for HTTP methods:

- **GET** - GET - Green (Safe, read-only operations)

Transaction Types

Type	Description	Direction
------	-------------	-----------

WALLET_TOPUP	Money added to wallet from external source	CREDIT
WALLET_WITHDRAWAL	Money withdrawn from wallet to external account	DEBIT
PURCHASE	Payment made for product purchase	DEBIT
PURCHASE_REFUND	Refund received for cancelled purchase	CREDIT
SALE	Earnings received from selling products	CREDIT
SALE_REFUND	Refund issued to buyer for sale	DEBIT
PLATFORM_FEE_COLLECTED	Platform fee collected (admin tracking)	CREDIT
GROUP_PURCHASE	Payment for group buying deal	DEBIT
GROUP_REFUND	Refund for failed group deal	CREDIT
INSTALLMENT_PAYMENT	One installment payment (future)	DEBIT
INSTALLMENT_REFUND	Refund for cancelled installment (future)	CREDIT
ESCROW_HOLD	Money held in escrow (admin tracking)	DEBIT
ESCROW_RELEASE	Money released from escrow (admin tracking)	CREDIT
ESCROW_REFUND	Money refunded from escrow	CREDIT

Transaction Directions

Direction	Description	Display
DEBIT	Money leaving wallet	Negative amount (-)
CREDIT	Money entering wallet	Positive amount (+)

Endpoints

1. Get My Transactions

Purpose: Retrieves all transactions for the authenticated user with pagination support. Returns transactions ordered by creation date (newest first).

Endpoint: GET `{base_url}`

Access Level: 🔒 Protected (Requires Authentication)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Query Parameters:

Parameter	Type	Required	Description	Default
page	integer	No	Page number (0-indexed)	0
size	integer	No	Number of items per page	20

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Transactions retrieved successfully",
  "action_time": "2025-10-02T10:30:45",
  "data": {
    "content": [
      {
        "id": "t1r2a3n4-s5a6-7890-abcd-ef1234567890",
        "transactionRef": "#2025T000123",
        "type": "PURCHASE",
        "direction": "DEBIT",
        "amount": 150000.00,
        "displayAmount": -150000.00,
        "currency": "TZS",
        "title": "Purchase Payment",
        "description": "Payment for order (Escrow: ESC-2025-000045)",
        "status": "COMPLETED",
```

```
    "createdAt": "2025-10-02T10:15:30",
    "referenceType": "ESCROW",
    "referenceId": "e1s2c3r4-o5w6-7890-abcd-ef1234567890"
  },
  {
    "id": "t2r3a4n5-s6a7-8901-bcde-f12345678901",
    "transactionRef": "#2025T000122",
    "type": "WALLET_TOPUP",
    "direction": "CREDIT",
    "amount": 50000.00,
    "displayAmount": 50000.00,
    "currency": "TZS",
    "title": "Wallet Topup",
    "description": "M-Pesa top-up from +255712345678",
    "status": "COMPLETED",
    "createdAt": "2025-10-02T09:30:15",
    "referenceType": "WALLET",
    "referenceId": "w1a2l3l4-e5t6-7890-abcd-ef1234567890"
  }
],
"pageable": {
  "pageNumber": 0,
  "pageSize": 20,
  "sort": {
    "sorted": true,
    "unsorted": false,
    "empty": false
  },
  "offset": 0,
  "paged": true,
  "unpaged": false
},
"totalElements": 47,
"totalPages": 3,
"last": false,
"size": 20,
"number": 0,
"sort": {
  "sorted": true,
  "unsorted": false,
```

```
        "empty": false
    },
    "numberOfElements": 20,
    "first": true,
    "empty": false
}
}
```

Success Response Fields:

Field	Description
id	Unique identifier for the transaction
transactionRef	Unique transaction reference number
type	Transaction type (see Transaction Types table)
direction	DEBIT or CREDIT
amount	Absolute transaction amount
displayAmount	Amount with sign (negative for DEBIT, positive for CREDIT)
currency	Currency code (TZS)
title	Short transaction title
description	Detailed transaction description
status	Transaction status (COMPLETED, PENDING, FAILED)
createdAt	ISO 8601 timestamp when transaction was created
referenceType	Type of related entity (WALLET, ESCROW, ORDER, etc.)
referenceId	UUID of related entity

Error Response Examples:

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Authentication token is required",
  "action_time": "2025-10-02T10:30:45",
  "data": "Authentication token is required"
}
```

2. Get Transaction by ID

Purpose: Retrieves detailed information about a specific transaction by its ID. Only the transaction owner can access their transactions.

Endpoint: GET `{base_url}/{id}`

Access Level: 🔒 Protected (Requires Authentication and Ownership)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Path Parameters:

Parameter	Type	Required	Description	Validation
id	string (UUID)	Yes	Unique identifier of the transaction	Valid UUID format

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Transaction retrieved successfully",
  "action_time": "2025-10-02T10:35:45",
  "data": {
    "id": "t1r2a3n4-s5a6-7890-abcd-ef1234567890",
    "transactionRef": "#2025T000123",
    "type": "PURCHASE",
    "direction": "DEBIT",
    "amount": 150000.00,
    "displayAmount": -150000.00,
    "currency": "TZS",
    "title": "Purchase Payment",
    "description": "Payment for order (Escrow: ESC-2025-000045)",
    "status": "COMPLETED",
    "createdAt": "2025-10-02T10:15:30",
```

```
"referenceType": "ESCROW",
"referenceId": "e1s2c3r4-o5w6-7890-abcd-ef1234567890"
}
}
```

Error Response Examples:

Not Found (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Transaction not found",
  "action_time": "2025-10-02T10:35:45",
  "data": "Transaction not found"
}
```

3. Get Transaction by Reference

Purpose: Retrieves a transaction by its unique reference number (e.g., #2025T000123).

Endpoint: **GET** {base_url}/ref/{transactionRef}

Access Level: ☐ Protected (Requires Authentication and Ownership)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Path Parameters:

Parameter	Type	Required	Description	Validation
transactionRef	string	Yes	Unique transaction reference number	Format: #YYYYTNNNNNN

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Transaction retrieved successfully",
  "action_time": "2025-10-02T10:40:45",
  "data": {
    "id": "t1r2a3n4-s5a6-7890-abcd-ef1234567890",
    "transactionRef": "#2025T000123",
    "type": "PURCHASE",
    "direction": "DEBIT",
    "amount": 150000.00,
    "displayAmount": -150000.00,
    "currency": "TZS",
    "title": "Purchase Payment",
    "description": "Payment for order (Escrow: ESC-2025-000045)",
    "status": "COMPLETED",
    "createdAt": "2025-10-02T10:15:30",
    "referenceType": "ESCROW",
    "referenceId": "e1s2c3r4-o5w6-7890-abcd-ef1234567890"
  }
}
```

Error Response Examples:

Not Found (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Transaction not found: #2025T000123",
  "action_time": "2025-10-02T10:40:45",
  "data": "Transaction not found: #2025T000123"
}
```

4. Get Transactions by Type

Purpose: Retrieves all transactions of a specific type for the authenticated user with pagination support.

Endpoint: GET {base_url}/filter/type

Access Level: 🔒 Protected (Requires Authentication)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Query Parameters:

Parameter	Type	Required	Description	Default
type	string (enum)	Yes	Transaction type to filter by	-
page	integer	No	Page number (0-indexed)	0
size	integer	No	Number of items per page	20

Valid Type Values:

- WALLET_TOPUP
- WALLET_WITHDRAWAL
- PURCHASE
- PURCHASE_REFUND
- SALE
- SALE_REFUND
- PLATFORM_FEE_COLLECTED
- GROUP_PURCHASE
- GROUP_REFUND
- INSTALLMENT_PAYMENT
- INSTALLMENT_REFUND
- ESCROW_HOLD
- ESCROW_RELEASE
- ESCROW_REFUND

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Transactions retrieved successfully",
}
```

```
"action_time": "2025-10-02T10:45:45",
"data": {
  "content": [
    {
      "id": "t1r2a3n4-s5a6-7890-abcd-ef1234567890",
      "transactionRef": "#2025T000123",
      "type": "PURCHASE",
      "direction": "DEBIT",
      "amount": 150000.00,
      "displayAmount": -150000.00,
      "currency": "TZS",
      "title": "Purchase Payment",
      "description": "Payment for order (Escrow: ESC-2025-000045)",
      "status": "COMPLETED",
      "createdAt": "2025-10-02T10:15:30",
      "referenceType": "ESCROW",
      "referenceId": "e1s2c3r4-o5w6-7890-abcd-ef1234567890"
    }
  ],
  "totalElements": 15,
  "totalPages": 1,
  "size": 20,
  "number": 0
}
}
```

Error Response Examples:

Bad Request - Invalid Type (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Invalid transaction type",
  "action_time": "2025-10-02T10:45:45",
  "data": "Invalid transaction type"
}
```

5. Get Transactions by Direction

Purpose: Retrieves all transactions of a specific direction (DEBIT or CREDIT) for the authenticated user with pagination support.

Endpoint: GET `{base_url}/filter/direction`

Access Level: 🔒 Protected (Requires Authentication)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Query Parameters:

Parameter	Type	Required	Description	Default
direction	string (enum)	Yes	Transaction direction (DEBIT or CREDIT)	-
page	integer	No	Page number (0-indexed)	0
size	integer	No	Number of items per page	20

Valid Direction Values:

- DEBIT (money out)
- CREDIT (money in)

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Transactions retrieved successfully",
  "action_time": "2025-10-02T10:50:45",
  "data": {
    "content": [
      {
        "id": "t2r3a4n5-s6a7-8901-bcde-f12345678901",
```

```
    "transactionRef": "#2025T000122",
    "type": "WALLET_TOPUP",
    "direction": "CREDIT",
    "amount": 50000.00,
    "displayAmount": 50000.00,
    "currency": "TZS",
    "title": "Wallet Topup",
    "description": "M-Pesa top-up from +255712345678",
    "status": "COMPLETED",
    "createdAt": "2025-10-02T09:30:15",
    "referenceType": "WALLET",
    "referenceId": "w1a2l3l4-e5t6-7890-abcd-ef1234567890"
  }
],
"totalElements": 23,
"totalPages": 2,
"size": 20,
"number": 0
}
}
```

Error Response Examples:

Bad Request - Invalid Direction (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Invalid transaction direction",
  "action_time": "2025-10-02T10:50:45",
  "data": "Invalid transaction direction"
}
```

6. Get Transactions by Date Range

Purpose: Retrieves all transactions within a specified date range for the authenticated user with pagination support.

Endpoint: **GET** `{base_url}/filter/date-range`

Access Level: ☐ Protected (Requires Authentication)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Query Parameters:

Parameter	Type	Required	Description	Default
startDate	string (ISO 8601)	Yes	Start date and time (inclusive)	-
endDate	string (ISO 8601)	Yes	End date and time (inclusive)	-
page	integer	No	Page number (0-indexed)	0
size	integer	No	Number of items per page	20

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Transactions retrieved successfully",
  "action_time": "2025-10-02T10:55:45",
  "data": {
    "content": [
      {
        "id": "t1r2a3n4-s5a6-7890-abcd-ef1234567890",
        "transactionRef": "#2025T000123",
        "type": "PURCHASE",
        "direction": "DEBIT",
        "amount": 150000.00,
        "displayAmount": -150000.00,
        "currency": "TZS",
        "title": "Purchase Payment",
        "description": "Payment for order (Escrow: ESC-2025-000045)",
        "status": "COMPLETED",
```

```
      "createdAt": "2025-10-02T10:15:30",
      "referenceType": "ESCROW",
      "referenceId": "e1s2c3r4-o5w6-7890-abcd-ef1234567890"
    }
  ],
  "totalElements": 8,
  "totalPages": 1,
  "size": 20,
  "number": 0
}
```

Error Response Examples:

Bad Request - Invalid Date Format (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Invalid date format. Use ISO 8601 format",
  "action_time": "2025-10-02T10:55:45",
  "data": "Invalid date format. Use ISO 8601 format"
}
```

7. Get Transaction Count

Purpose: Retrieves the total count of all transactions for the authenticated user.

Endpoint: **GET** `{base_url}/count`

Access Level:  Protected (Requires Authentication)

Authentication: Bearer Token required in Authorization header

Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authenticated user

Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Transaction count retrieved successfully",
  "action_time": "2025-10-02T11:00:45",
  "data": 47
}
```

Error Response Examples:

Unauthorized (401):

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Authentication token is required",
  "action_time": "2025-10-02T11:00:45",
  "data": "Authentication token is required"
}
```

Integration Examples

Example 1: Get Recent Transactions

```
GET /api/v1/transaction-history?page=0&size=10
Authorization: Bearer {token}
```

Response: Returns 10 most recent transactions

Example 2: View All Purchases

```
GET /api/v1/transaction-history/filter/type?type=PURCHASE&page=0&size=20
Authorization: Bearer {token}
```

Response: Returns all purchase transactions

Example 3: Check Money Received (Credits)

```
GET /api/v1/transaction-history/filter/direction?direction=CREDIT&page=0&size=20
Authorization: Bearer {token}
```

Response: Returns all credit (incoming) transactions

Example 4: Get Monthly Statement

```
GET /api/v1/transaction-history/filter/date-range?startDate=2025-09-01T00:00:00Z&endDate=2025-09-30T23:59:59Z&page=0&size=100
Authorization: Bearer {token}
```

Response: Returns all transactions for September 2025

Example 5: Track Specific Transaction

Step 1: Get Transaction by Reference

```
GET /api/v1/transaction-history/ref/#2025T000123
Authorization: Bearer {token}
```

Step 2: Get Full Details by ID

```
GET /api/v1/transaction-history/t1r2a3n4-s5a6-7890-abcd-ef1234567890
Authorization: Bearer {token}
```

Rate Limiting

Rate Limits:

- All Endpoints: 60 requests per minute per user

Rate Limit Headers:

```
X-RateLimit-Limit: 60
X-RateLimit-Remaining: 45
```

Rate Limit Exceeded:

```
{
  "success": false,
  "httpStatus": "TOO_MANY_REQUESTS",
  "message": "Rate limit exceeded. Please try again later.",
  "action_time": "2025-10-02T11:05:45",
  "data": "Rate limit exceeded. Please try again later."
}
```

Best Practices

For Developers

1. **Use pagination** for large result sets
2. **Cache transaction lists** with short TTL (1-2 minutes)
3. **Filter by type or direction** for specific views
4. **Use date ranges** for generating statements
5. **Display displayAmount** field for user-friendly amounts with signs
6. **Show transaction reference** for user tracking and support
7. **Link to related entities** using referenceType and referenceId

For UI/UX

1. **Color-code transactions:** Green for CREDIT, Red for DEBIT
2. **Group by date:** Show transactions by day/month
3. **Provide filters:** Type, direction, date range
4. **Show running balance:** Calculate balance after each transaction
5. **Enable search:** By reference number or description
6. **Export capability:** Allow CSV/PDF export of statements
7. **Infinite scroll or pagination:** For better mobile experience

Revision #1

Created 2 October 2025 06:49:23 by Admin Qbit

Updated 11 December 2025 08:29:03 by Admin Qbit