

Unblock Transfers

Documentation

Api Verestro | Transfers flow documentation

Scope

This document describes the currently implemented partner-facing flow for retrieving incoming transfers through the `GET /transfers` endpoint.

Implemented endpoint

- `GET /transfers`

Authentication

Requests to this endpoint require:

- `api-key` header
- `signature` header

Query parameters

Name	Description	Type	Required
accountNumber	Account number used to retrieve incoming transfers	String	Yes

Name	Description	Type	Required
limit	Maximum number of transfers returned per request. Default: 20. Min: 1. Max: 100	Number	No
lastId	Pagination cursor. Used to retrieve the next page of results	String	No

Example request

```
GET /transfers?accountNumber=MT84MALT011000012345MTLCAST001S&limit=20&lastId=12345
api-key: <your-api-key>
signature: <request-signature>
```

Implemented transfer retrieval flow

1. The partner calls GET /transfers with the destination accountNumber.
2. The API returns incoming transfers associated with the authenticated merchant and the provided account number.
3. Each transfer item includes the title field.
4. The title field is currently populated from the transfer reference.
5. If no reference is available, title is returned as an empty string ("").
6. Results are returned in pages using pagination.lastId and pagination.hasNext.

Response format

The endpoint returns:

- data: list of transfer items
- pagination: pagination metadata

Transfer item fields returned by the presenter

Field	Type	Description
id	String	Transfer identifier
title	String	Transfer title. Currently mapped from reference. If no reference is available, this field is returned as ""

Field	Type	Description
createdAt	String	Transfer creation timestamp in ISO 8601 format
settledAt	String null	Settlement timestamp in ISO 8601 format, if available
status	String	Transfer processing status. COMPLETED is exposed as SETTLED
direction	String	Always INCOMING
amount.value	Number	Transfer amount
amount.currency	String null	Transfer currency
sender.name	String null	Currently returned as null
sender.address	String null	Currently returned as null
sender.accountNumber	String null	Sender account number
sender.bic	String null	Currently returned as null
receiver.name	String null	Currently returned as null
receiver.address	String null	Currently returned as null
receiver.accountNumber	String null	Receiver account number
receiver.bic	String null	Currently returned as null

Pagination fields

Field	Type	Description
pagination.lastId	String null	Identifier of the last item returned in the current page
pagination.hasNext	Boolean	Indicates whether more transfers are available

Example success response

```
{
  "data": [
    {
      "id": "12346",
      "title": "USR-ABC123",
      "createdAt": "2026-06-09T14:23:11.000Z",
```

```
"settledAt": "2026-06-09T14:25:40.000Z",
"status": "SETTLED",
"direction": "INCOMING",
"amount": {
  "value": 150.75,
  "currency": "EUR"
},
"sender": {
  "name": null,
  "address": null,
  "accountNumber": "ES9121000418450200051332",
  "bic": null
},
"receiver": {
  "name": null,
  "address": null,
  "accountNumber": "MT84MALT011000012345MTLCAST001S",
  "bic": null
}
},
{
  "id": "12345",
  "title": "",
  "createdAt": "2026-06-08T10:11:12.000Z",
  "settledAt": null,
  "status": "RECEIVED",
  "direction": "INCOMING",
  "amount": {
    "value": 80,
    "currency": "EUR"
  },
  "sender": {
    "name": null,
    "address": null,
    "accountNumber": "DE89370400440532013000",
    "bic": null
  },
  "receiver": {
    "name": null,
```

```
    "address": null,
    "accountNumber": "MT84MALT0110000012345MTLCAST001S",
    "bic": null
  }
},
"pagination": {
  "lastId": "12345",
  "hasNext": true
}
```

Error responses

The endpoint may return the following HTTP status codes:

- 400 Bad Request
- 401 Unauthorized
- 500 Internal Server Error

Example 400 Bad Request

```
{
  "status": "INVALID_INPUT",
  "message": "field required accountNumber"
}
```

Example 401 Unauthorized

```
{
  "status": "API_KEY_IS_MISSING"
}
```

Example 500 Internal Server Error

```
{
  "status": "DATABASE_ERROR",
```

```
"message": "Error trying to retrieve transfers."
}
```

Flow diagram

flowchart TD

```
A[Bank deposit arrives] --> B[Store transfer in database]
B --> C[Increase merchant balance]
C --> D{Does the transfer title contain a micro-transaction user code?}

D -- No --> E[Notify Verestro through financial-institution/notifications/receivings]
E --> F{Routing decision}
F -- Credit user balance --> G[Funds assigned to user balance]
F -- Keep as merchant funds --> H[Funds remain on merchant balance]

D -- Yes --> I[Validate micro-transaction code in title]
I --> J[Example code: VU-14-VERIF]
J --> K[Notify Verestro through financial-institution/notifications/receivings]
K --> L{Routing decision}
L -- Credit user balance --> M[Funds assigned to user balance]
L -- Keep as merchant funds --> N[Funds remain on merchant balance]
```

Notes:

- Example micro-transaction user code: `VU-14-VERIF`.
- In both flows, the merchant balance is increased before notifying Verestro.

Revision #1

Created 19 June 2026 11:52:46 by Krzysztof Wereski

Updated 19 June 2026 11:53:50 by Krzysztof Wereski