

# Unblock Integration

## Use Cases

### KYC Verification

```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
autonumber
actor "Partner-User" as user
participant Verestro as ver
participant Unblock as unblock
user -> ver: perform KYC
note left: Process not fully described\nshows only where unique code comes from
ver -> unblock: POST /user/create
unblock --> ver: response(uniqueCode)
ver -> ver: storeUniqueCode
@enduml
```

#### **Key information:**

- Unblock creates uniqueCode for user
- Verestro will store this uniqueCode to provide later for partner's users
- At this stage there is only one uniqueCode created on Unblock side and stored on Verestro
- To distinguish verification and normal transfers Verestro and Unblock agreed to following:
  - Verestro will add **-VERIF** postfix to uniqueCode and provide it to partner's users
- **Format of uniqueCode is V-UB-U-000001 which means following regexp `^V-UB-U-\d{6}$`**

## Transfer

@startuml

skinparam ParticipantPadding 30

skinparam BoxPadding 30

skinparam noteFontColor #FFFFFF

skinparam noteBackgroundColor #1C1E3F

skinparam noteBorderColor #1C1E3F

skinparam noteBorderThickness 1

skinparam sequence {

ArrowColor #1C1E3F

ArrowFontColor #1C1E3F

ActorBorderColor #1C1E3F

ActorBackgroundColor #FFFFFF

ActorFontStyle bold

ParticipantBorderColor #1C1E3F

ParticipantBackgroundColor #1C1E3F

ParticipantFontColor #FFFFFF

ParticipantFontStyle bold

LifeLineBackgroundColor #1C1E3F

LifeLineBorderColor #1C1E3F

}

autonumber

actor "Partner-User" as user

participant Verestro as ver

participant Unblock as unblock

participant "Unblock-Bank" as unbank

...Make transfer...

user -> ver: getTransferInfo

ver --> user: response:\n(accountNumber\ntransferCode=V-UB-U-000001\nverificationCode=V-UB-U-000001-verif)

user -> unbank: transfer

unblock->unbank: getTransfers

unblock -> unblock: creditMB

unblock -> unblock: recognizeDestination\nVERIFICATION: recognized by postfix -

```

verif\nINDIVIDUAL: has unique code\nMASTER_BALANCE: other
unblock -> ver: notify:\n/secure/financial-institution/notifications/receivings
ver -> ver: queueGetTransfers
ver --> unblock: response
...Get Transfers...
ver -> unblock: getTransfers(accountNumber, lastId?)
unblock -->ver: response(transfers)
...Process transfer...
ver -> ver: recognizeTransfer
alt hasUniqueCode
ver -> ver:checkAml
alt accepted
ver -> ver: creditMBAndUserBalance
note right: Verestro will take first balance in given currency for credit user balance
ver -> unblock: notify(type=masterBalanceCredit)
unblock -> unblock: ignoreMBCredit
else rejected
ver -> unblock: refund
unblock -> unblock: debitMB
note right: Someone from unblock manually refund transfer to user
end
else
alt isForMasterBalance
ver -> ver: creditMB
ver -> unblock: notify(type=masterBalanceCredit)
unblock -> unblock: ignoreMBCredit
else unrecognized destination
ver -> unblock: refund
unblock -> unblock: debitMB
note right: Someone from unblock manually refund transfer to user
end
end
@enduml

```

## Key Information

- Verestro exposes transferCode=uniqueCode and verificationCode=uniqueCode-verif with accountNumber to partner's users
- Unblock gets transfer from account number dedicated for partner
- Every deposit transfer will be credited on MB on Unblock side
  - In scenario when Verestro will be processing this transfer there will be difference between MB on Verestro side and Unblock side(Unblock will have grater MB). This is acceptable because Verestro check MB on own side before sending request to Unblock so there is no risk that someone will withdraw money that can be refunded

in some time

- Unblock will send notification to Verestro after every transfer. Will call api described in [Transfer Notification](#)
- Verestro will call [Proposed Get Transfers](#) and retrieve transfers
  - **IMPORTANT: Transfers from Unblock are returned in order from oldest to newest. This will allow to incrementally download next transfers which are not stored on Verestro side E.g:**

- **GET /transfers - first call. Sample response:**

| id - transfer id | timestamp - transfer timestamp |
|------------------|--------------------------------|
| 1                | 14.06.2026                     |

- **Get /transfers?lastId=1 - next call will return only newer transfers than lastId=1. Sample response:**

| id - transfer id | timestamp - transfer timestamp |
|------------------|--------------------------------|
| 2                | 15.06.2026                     |

- In case where transfer has to returned to sender. Verestro will call proper api which will be prepared by Unblock
  - The refund will debit MB on Unblock side since previously it was credited
  - Some employee from Unblock will make proper refund using bank
- Unblock will ignore transactionType from notificai

# Documentations

## Unblock

### Received

Documentation received from Unblock is:

- [Users](#)
- [Transfers](#)
- [Signature](#)

## Proposed Get Transfers

@swagger="https://s3.verestro.com/products-documentation-public/unblock/unblock\_get\_transfers.yaml"

---

Revision #46

Created 14 June 2026 05:42:16 by Krzysztof Wereski

Updated 29 June 2026 15:22:10 by Krzysztof Wereski