

Disbursement Channel API

Author: Josh — Lead Backend Team **Last Updated:** 2026-03-06 **Version:** v1.0

Base URL: `https://api.nextgate.co.tz/api/v1`

Description: Manages withdrawal destinations for NextGate users. A channel is a verified mobile money or bank account to which a user can withdraw their wallet balance. Channels require OTP verification to add and have a cooling period before first use.

Key notes:

- Adding a channel is a **three-step flow**: `/lookup` → `/add` → `/add/confirm`
- The `/lookup` step returns a signed `confirmationToken` required by `/add` — expires in 10 minutes
- Selcom name lookup is performed at both `/lookup` and `/add` — verified twice server-side
- The **first channel** on an account activates immediately — 24-hour cooling applies to all subsequent channels
- Deleting a channel requires OTP confirmation — two-step: `DELETE /{channelId}` → `DELETE /{channelId}/confirm`
- Deleted channels cannot be used for new withdrawals — in-flight disbursements snapshot channel data at initiation time and are unaffected

Security & Client Guidelines

Authentication

All endpoints require a valid JWT Bearer token:

```
Authorization: Bearer <your_token>
```

On `401 UNAUTHORIZED`, silently refresh the token and retry once. If refresh fails, redirect to login.

Never store the JWT in localStorage — use secure device storage on mobile, or HttpOnly cookies on web.

OTP Token Handling

Both the add and delete flows are two-step. The first step returns an `otpToken`. Store this in memory and pass it to the confirm step along with the SMS OTP code. Do not log it or persist it beyond the confirmation flow.

The OTP expires after ~5 minutes. If the user is too slow, the confirm step returns an error — prompt them to **restart the entire flow from the beginning**. The server must clean up the previous pending channel record before a new one can be created.

If the wrong OTP is entered too many times, the OTP is locked and the flow is terminated. The user must start over.

Name Lookup and Confirmation Token

The `/lookup` step returns a signed `confirmationToken` that encodes the destination, channel type, and account ID. It is valid for **10 minutes**.

The client **must** pass this token to `/add`. The server rejects the request if the token is missing, expired, or does not match the destination and channel type in the request body.

Always display `accountHolderName` prominently to the user and require explicit confirmation before calling `/add`. This protects users from accidentally adding a wrong destination.

Cooling Period UX

After OTP confirmation, `isUsable` is `false` until `activatesAt`. Show a message like:

““This channel will be ready for withdrawals on {activatesAt date}””

Visually distinguish usable channels from those still in the cooling period when listing.

Exception: The first channel added to an account activates immediately (`isUsable: true`, `activatesAt` set to now).

Deleting a Channel

Deletion is irreversible. Show a clear confirmation dialog before initiating the delete flow.

If the deleted channel was the primary channel, the server automatically promotes another active channel as primary.

Pending disbursements that reference a deleted channel are not affected — the disbursement holds a snapshot of the channel data from initiation time.

Network Error Handling

HTTP Status	Meaning	Client action
400	Business rule violation	Show <code>message</code> to user, do not retry
401	Token expired or invalid	Refresh token silently, retry once
403	Forbidden	Show error, do not retry
422	Validation error	Show field errors, fix and resubmit
500	Server error	Show generic error, allow manual retry
Network timeout	No response	Lookup and list: safe to retry. Initiate/confirm: check status first

Standard Response Format

Success

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Operation completed successfully",
  "action_time": "2026-03-06T10:30:45",
  "data": {}
}
```

Error

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Error description",
  "action_time": "2026-03-06T10:30:45",
  "data": "Error description"
}
```

```
}
```

Flow Diagram

ADD CHANNEL FLOW

User

```
|
|--- POST /disbursement/channels/lookup -----> Selcom Name Lookup
|       { channelType, destination, bankCode? }       |
|<-- { accountHolderName, destinationDisplay, Returns verified name
|       channelType, confirmationToken }
|
|                               <-- store confirmationToken (expires 10 min)
| [User reviews name, confirms]
|
|--- POST /disbursement/channels/add -----> Validate confirmationToken
|       { channelType, destination, Re-verify name with Selcom
|       bankCode?, confirmationToken }       Check for duplicate
|
|                               Save channel record
|
|                               Generate + send OTP
|<-- { otpToken }
|
|--- POST /disbursement/channels/add/confirm --> Verify OTP
|       ?otpToken=...&otpCode=...           Activate channel
|<-- { channelId, isUsable, activatesAt, ... } (cooling 24h, except first channel)
|
| [Channel becomes usable after activatesAt]
|
|--- GET /disbursement/channels -----> List active channels
|<-- [ { channelId, isUsable, activatesAt, ... } ]
```

DELETE CHANNEL FLOW

User

```
|
|--- DELETE /disbursement/channels/{channelId} -----> Generate + send OTP
|<-- { otpToken }
```

```
|  
|--- DELETE /disbursement/channels/{channelId}/confirm --> Verify OTP  
|           ?otpToken=...&otpCode=...                               Soft delete channel  
|<-- 200 OK                                                         Reassign primary if needed
```

Channel Status Reference

Status	Description
PENDING_ACTIVATION	Channel OTP confirmed, within 24-hour cooling period — not yet usable
ACTIVE	Channel verified and activated
DELETED	Channel removed by user

“ A cleanup job runs nightly and soft-deletes any channel records whose OTP was never verified (i.e. the add flow was abandoned before confirm).

Endpoints

1. Lookup Account Name

Purpose: Verifies a destination account and returns the registered account holder name from Selcom, along with a signed `confirmationToken` required for the next step. Always call this before `/add`.

Endpoint: `POST /disbursement/channels/lookup`

Access: `⏏` Protected — Bearer Token required

Request Body:

```
{  
  "channelType": "MPESA",  
  "destination": "255712345678",
```

```
"bankCode": null
}
```

Parameter	Type	Required	Description
<code>channelType</code>	string	Yes	MPESA, AIRTEL, TIGOPESA, HALOPESA, SELCOM_PESA, BANK
<code>destination</code>	string	Yes	Phone number (255XXXXXXXX) or bank account number
<code>bankCode</code>	string	Conditional	Required when <code>channelType</code> is BANK

Success Response:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Account verified successfully",
  "action_time": "2026-03-06T10:30:45",
  "data": {
    "accountHolderName": "JOHN DOE",
    "destinationDisplay": "2557****678",
    "channelType": "MPESA",
    "confirmationToken": "<signed-token>"
  }
}
```

Field	Description
<code>accountHolderName</code>	Verified name from Selcom — display to user for confirmation
<code>destinationDisplay</code>	Masked destination
<code>channelType</code>	The channel type
<code>confirmationToken</code>	Signed token required by <code>/add</code> . Valid for 10 minutes. Store in memory

Errors:

Scenario	Message
Account not found	"Account not found. Please check the number and try again."

Scenario	Message
Phone not verified	"Your phone number must be verified before adding a withdrawal channel."
Destination already added	"This destination is already added as a withdrawal channel."
Bank code missing	"Bank code is required for bank channels."
Selcom lookup failed	"Could not verify account. Please check the details and try again."

2. Initiate Add Channel

Purpose: Initiates adding a new channel. Validates the `confirmationToken` from `/lookup`, re-verifies the name with Selcom server-side, saves the channel record, and sends an OTP to the user's verified phone.

Endpoint: `POST /disbursement/channels/add`

Access: Protected — Bearer Token required

Request Body (Mobile Money):

```
{
  "channelType": "MPESA",
  "destination": "255712345678",
  "bankCode": null,
  "confirmationToken": "<token from /lookup>"
}
```

Request Body (Bank):

```
{
  "channelType": "BANK",
  "destination": "0012345678901",
  "bankCode": "CRDB",
  "confirmationToken": "<token from /lookup>"
}
```

Parameter	Type	Required	Description
<code>channelType</code>	string	Yes	MPESA, AIRTEL, TIGOPESA, HALOPESA, SELCOM_PESA, BANK

Parameter	Type	Required	Description
<code>destination</code>	string	Yes	Phone or account number
<code>bankCode</code>	string	Conditional	Required when <code>channelType</code> is <code>BANK</code>
<code>confirmationToken</code>	string	Yes	Token from <code>/lookup</code> . Must match destination, channel type, and account

Success Response:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "OTP sent to your verified phone number",
  "action_time": "2026-03-06T10:30:45",
  "data": {
    "otpToken": "otp-token-uuid-here"
  }
}
```

Field	Description
<code>otpToken</code>	Pass to <code>/add/confirm</code> with the SMS OTP code

Errors:

Scenario	Message
Token expired	"Confirmation token expired. Please look up the account again."
Token invalid/mismatch	"Invalid confirmation token."
Channel already active	"This destination is already an active withdrawal channel."
Max channels reached	"Maximum of {n} withdrawal channels allowed."
Add window closed	"Withdrawal channel add window has closed."

3. Confirm Add Channel

Purpose: Confirms adding a new channel by verifying the OTP. On success the channel is activated with a cooling period before first use (first channel activates immediately).

Endpoint: `POST /disbursement/channels/add/confirm`

Access: `Protected` — Bearer Token required

Query Parameters:

Parameter	Type	Required	Description
<code>otpToken</code>	string	Yes	Token returned from <code>/add</code>
<code>otpCode</code>	string	Yes	6-digit OTP received via SMS

Success Response:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Channel added successfully",
  "action_time": "2026-03-06T10:30:45",
  "data": {
    "channelId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
    "channelType": "MPESA",
    "destinationDisplay": "2557****678",
    "accountHolderName": "JOHN DOE",
    "bankName": null,
    "isPrimary": false,
    "status": "ACTIVE",
    "isUsable": false,
    "activatesAt": "2026-03-07T10:31:00"
  }
}
```

Field	Description
<code>channelId</code>	UUID — use this when initiating withdrawals
<code>channelType</code>	Channel type
<code>destinationDisplay</code>	Masked destination
<code>accountHolderName</code>	Verified name from Selcom
<code>bankName</code>	Bank name — only for <code>BANK</code> channels, null otherwise
<code>isPrimary</code>	True if this is now the user's primary channel
<code>status</code>	<code>ACTIVE</code> or <code>PENDING_ACTIVATION</code>

Field	Description
<code>isUsable</code>	<code>false</code> if still in cooling period
<code>activatesAt</code>	When the channel becomes usable for withdrawals

Errors:

Scenario	Message
Invalid OTP	"Invalid OTP code."
OTP locked	"OTP locked – max attempts exceeded."
Channel no longer valid	"This channel is no longer valid. Please add it again."

4. List My Channels

Purpose: Returns all active withdrawal channels for the authenticated user.

Endpoint: `GET /disbursement/channels`

Access: `🔒` Protected — Bearer Token required

“ Only `ACTIVE` channels are returned. Deleted and pending-activation channels that have not yet been OTP-confirmed are excluded.

Success Response:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Channels retrieved successfully",
  "action_time": "2026-03-06T10:30:45",
  "data": [
    {
      "channelId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
      "channelType": "MPESA",
      "destinationDisplay": "2557***678",
      "accountHolderName": "JOHN DOE",
      "bankName": null,
      "isPrimary": true,
    }
  ]
}
```

```

    "status": "ACTIVE",
    "isUsable": true,
    "activatesAt": "2026-03-05T10:31:00"
  },
  {
    "channelId": "b2c3d4e5-f6a7-8901-bcde-f12345678901",
    "channelType": "BANK",
    "destinationDisplay": "0012****789",
    "accountHolderName": "JOHN DOE",
    "bankName": "CRDB Bank",
    "isPrimary": false,
    "status": "ACTIVE",
    "isUsable": false,
    "activatesAt": "2026-03-07T08:00:00"
  }
]
}

```

Field	Description
<code>channelId</code>	UUID of the channel
<code>channelType</code>	Channel type
<code>destinationDisplay</code>	Masked destination
<code>accountHolderName</code>	Verified name from Selcom
<code>bankName</code>	Bank name — only for <code>BANK</code> channels, null otherwise
<code>isPrimary</code>	Whether this is the user's primary channel
<code>status</code>	<code>ACTIVE</code>
<code>isUsable</code>	<code>true</code> if cooling period has passed and channel can be used now
<code>activatesAt</code>	When the cooling period ends

5. Initiate Delete Channel

Purpose: Initiates deletion of a withdrawal channel. Sends OTP to user's verified phone for confirmation.

Endpoint: `DELETE /disbursement/channels/{channelId}`

Access: Protected — Bearer Token required

Path Parameters:

Parameter	Type	Required	Description
<code>channelId</code>	UUID	Yes	ID of the channel to delete

Success Response:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "OTP sent to your verified phone number",
  "action_time": "2026-03-06T10:30:45",
  "data": {
    "otpToken": "otp-token-uuid-here"
  }
}
```

Errors:

Scenario	Message
Channel not found / not owned	<code>"Channel not found."</code>
Channel not active	<code>"Only active channels can be deleted."</code>
Phone not verified	<code>"Your phone number must be verified to delete a withdrawal channel."</code>

6. Confirm Delete Channel

Purpose: Confirms channel deletion by verifying the OTP. Channel is permanently removed. If it was the primary channel, another active channel is automatically promoted.

Endpoint: `DELETE /disbursement/channels/{channelId}/confirm`

Access: Protected — Bearer Token required

Path Parameters:

Parameter	Type	Required	Description
<code>channelId</code>	UUID	Yes	ID of the channel to delete

Query Parameters:

Parameter	Type	Required	Description
otpToken	string	Yes	Token returned from initiate delete
otpCode	string	Yes	6-digit OTP received via SMS

Success Response:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Channel deleted successfully",
  "action_time": "2026-03-06T10:30:45",
  "data": null
}
```

Errors:

Scenario	Message
Invalid OTP	"Invalid OTP code."
OTP does not match channel	"OTP does not match this channel."
Channel not active	"Only active channels can be deleted."

Quick Reference

Method	Endpoint	Description
POST	/disbursement/channels/lookup	Verify account name, get confirmationToken
POST	/disbursement/channels/add	Start add channel flow (requires confirmationToken)
POST	/disbursement/channels/add/confirm	Complete add channel with OTP
GET	/disbursement/channels	List all active user channels
DELETE	/disbursement/channels/{channelId}	Start delete channel flow
DELETE	/disbursement/channels/{channelId}/confirm	Complete delete with OTP

Updated 11 March 2026 10:38:48 by Admin Qbit