

Technical Documentation – Neero Gateway API

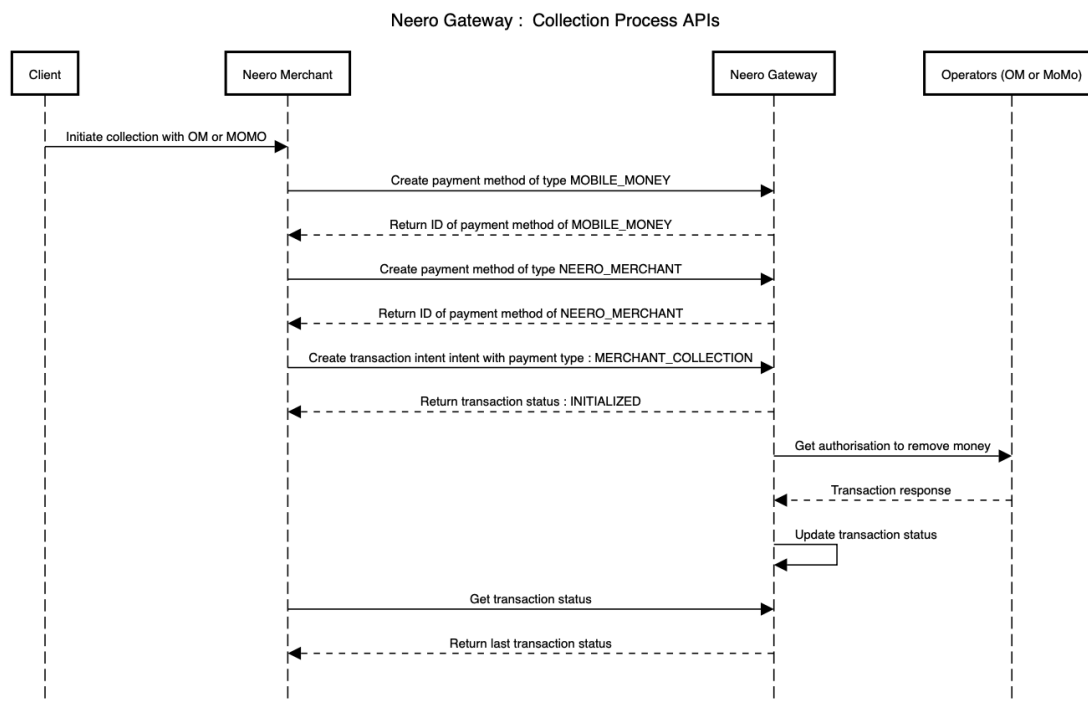
The Neero Gateway payment system allows merchants to collect and transfer money through Mobile Money (Orange Money, MTN Money) and their Neero Business accounts.

This document explains how each API works and how developers can integrate them into their systems.

Integration steps summary:

1. Authenticate using your secret key (Basic Auth).
2. Create both Payment Methods (Mobile Money and Merchant).
3. Save their IDs.
4. Initiate a collection(payment) or transfer using transaction intent Cash-In / Cash-Out endpoints.
5. Retrieve transaction status using GET /transaction-intents/:id.

Neero collection sequence diagram

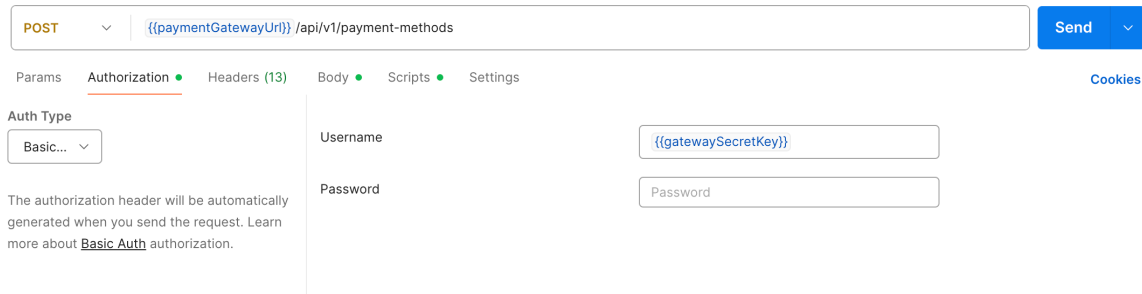


2. Authentication

All API requests use Basic Authentication.

- **Username**: the secret key generated in your Neero Dashboard.
- **Password**: leave empty.

This key is unique to your merchant account and must be kept confidential.



POST Send

Params Authorization Headers (13) Body Scripts Settings Cookies

Auth Type: Basic...

The authorization header will be automatically generated when you send the request. Learn more about [Basic Auth](#) authorization.

Username:

Password:

2. Payment Method Creation

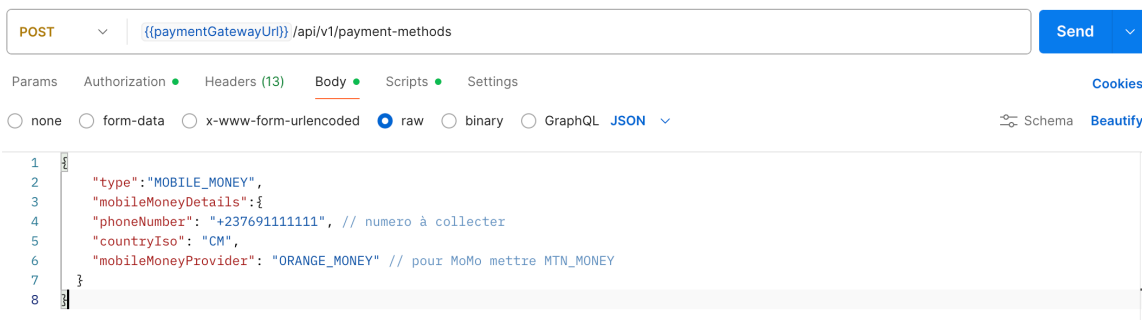
There are two types of Payment Methods:

1. **MOBILE_MONEY** – identifies a customer's Orange Money or MTN Money number.
2. **NEERO_MERCHANT** – identifies the merchant's Neero account (fixed and unique).

2.1 Create a Mobile Money Payment Method

Endpoint: POST /api/v1/payment-methods

Request body:



POST Send

Params Authorization Headers (13) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Schema Beautify

```
1 {
2   "type": "MOBILE_MONEY",
3   "mobileMoneyDetails": {
4     "phoneNumber": "+237691111111", // numero à collecter
5     "countryIso": "CM",
6     "mobileMoneyProvider": "ORANGE_MONEY" // pour MoMo mettre MTN_MONEY
7   }
8 }
```

```
{
  "type": "MOBILE_MONEY",
  "mobileMoneyDetails": {
    "phoneNumber": "+237691111111",
    "countryIso": "CM",
    "mobileMoneyProvider": "ORANGE_MONEY"
  }
}
```

```
}
}
```

The response includes a unique Payment Method ID that must be saved. For the same phone number, this ID will always remain the same.



```

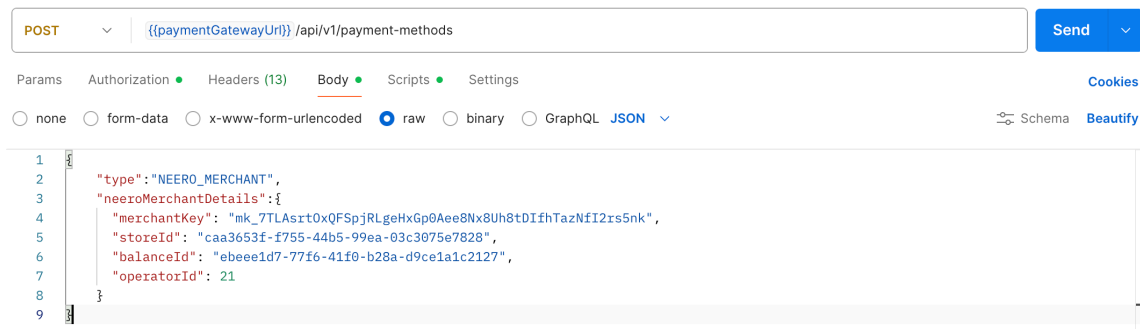
1 {
2   "createdAt": "2025-10-19T08:47:19.032",
3   "updatedAt": "2025-10-19T09:27:24.161875693",
4   "id": "68f4a597b0fb816249d75dd0",
5   "metadata": {},
6   "operatorDetails": {
7     "operatorId": 200,
8     "merchantKey": "mk_7TLAsrt0xQFSpjRLgeHxGp0Aee8Nx8Uh8tDIfhTazNfi2rs5nk"
9   },
10  "active": true,
11  "type": "NEERO_MERCHANT",
12  "walletTypeProductName": null,
13  "mobileMoneyDetails": null,
14  "neeroPersonDetails": null,
15  "neeroMerchantDetails": {
16    "merchantKey": "mk_7TLAsrt0xQFSpjRLgeHxGp0Aee8Nx8Uh8tDIfhTazNfi2rs5nk",
17    "storeId": "caa3653f-f755-44b5-99ea-03c3075e7828",
18    "balanceId": "ebeeee1d7-77f6-41f0-b28a-d9ce1alc2127",
19    "operatorId": 21,

```

2.2 Create a Merchant Payment Method

Endpoint: POST /api/v1/payment-methods

Request body:



```

1 {
2   "type": "NEERO_MERCHANT",
3   "neeroMerchantDetails": {
4     "merchantKey": "mk_7TLAsrt0xQFSpjRLgeHxGp0Aee8Nx8Uh8tDIfhTazNfi2rs5nk",
5     "storeId": "caa3653f-f755-44b5-99ea-03c3075e7828",
6     "balanceId": "ebeeee1d7-77f6-41f0-b28a-d9ce1alc2127",
7     "operatorId": 21
8   }
9 }

```

```

{
  "type": "NEERO_MERCHANT",
  "neeroMerchantDetails": {
    "merchantKey": "mk_XXXXXX", // information available on neero dashboard
    "storeId": "store_uuid", // information available on neero dashboard
    "balanceId": "balance_uuid", // information available on neero dashboard
    "operatorId": 21 // information available on neero dashboard
  }
}

```

```
}

```

The response will content an ID. This ID is permanent and must be stored securely by the merchant.



```

1  {
2    "createdAt": "2025-10-19T08:47:19.032",
3    "updatedAt": "2025-10-19T09:27:24.161875693",
4    "id": "68f4a597b0fb816249d75dd0",
5    "metadata": {},
6    "operatorDetails": {
7      "operatorId": 200,
8      "merchantKey": "mk_7TLAsrt0xQFSpjRLgeHxGp0Aee8Nx8Uh8tDifhTazNfi2rs5nk"
9    },
10   "active": true,
11   "type": "NEERO_MERCHANT",
12   "walletTypeProductName": null,
13   "mobileMoneyDetails": null,
14   "neeroPersonDetails": null,
15   "neeroMerchantDetails": {
16     "merchantKey": "mk_7TLAsrt0xQFSpjRLgeHxGp0Aee8Nx8Uh8tDifhTazNfi2rs5nk",
17     "storeId": "caa3653f-f755-44b5-99ea-03c3075e7828",
18     "balanceId": "ebee1d7-77f6-41f0-b28a-d9ce1alc2127",
19     "operatorId": 21,

```

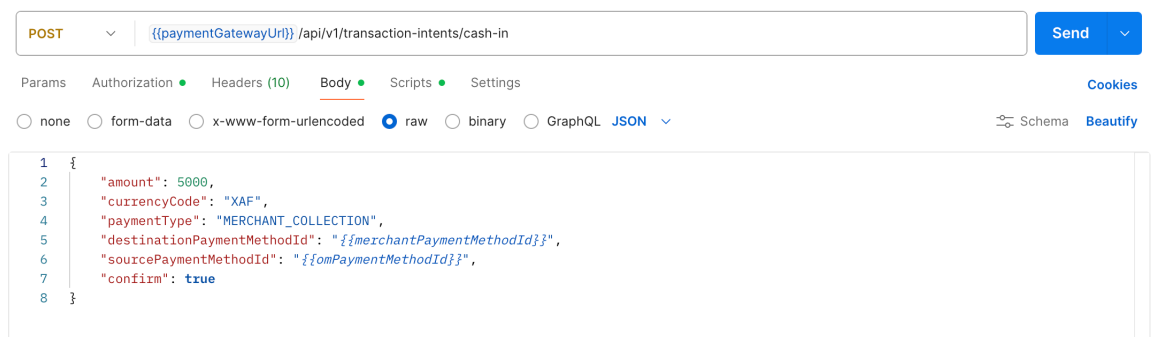
3. Collect Funds (Cash-In)

Endpoint: POST /api/v1/transaction-intents/cash-in

This operation allows a collection from a customer's Mobile Money wallet (Orange or MTN) to the merchant's Neero account.

Note : For collect on sourcePaymentMethodId you add mobileMoney payment method Id and on destination destinationPaymentMethodId you add merchantPaymentMethodId

Request body:



```

1  {
2    "amount": 5000,
3    "currencyCode": "XAF",
4    "paymentType": "MERCHANT_COLLECTION",
5    "destinationPaymentMethodId": "{{merchantPaymentMethodId}}",
6    "sourcePaymentMethodId": "{{omPaymentMethodId}}",
7    "confirm": true
8  }

```

```

{
  "amount": 5000,
  "currencyCode": "XAF",

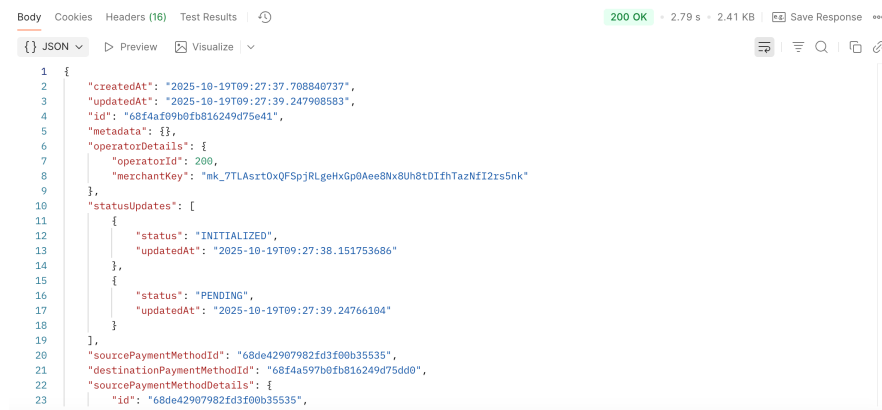
```

```

"paymentType": "MERCHANT_COLLECTION",
"destinationPaymentMethodId": "{{merchantPaymentMethodId}}",
"sourcePaymentMethodId": "{{omPaymentMethodId}}",
"confirm": true
}

```

Simplified response example:



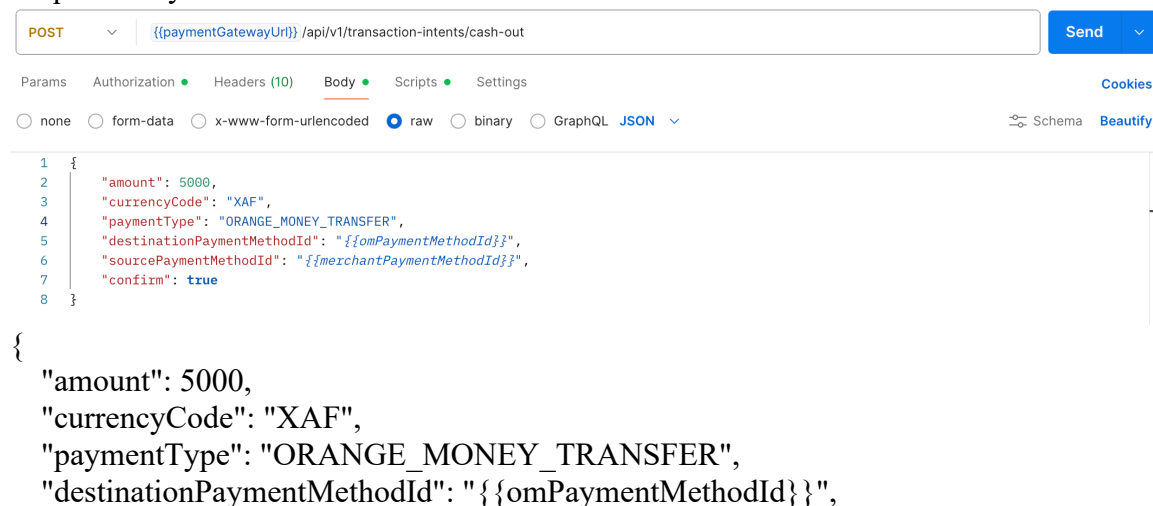
4. Transfer Funds (Cash-Out)

Endpoint: POST /api/v1/transaction-intents/cash-out

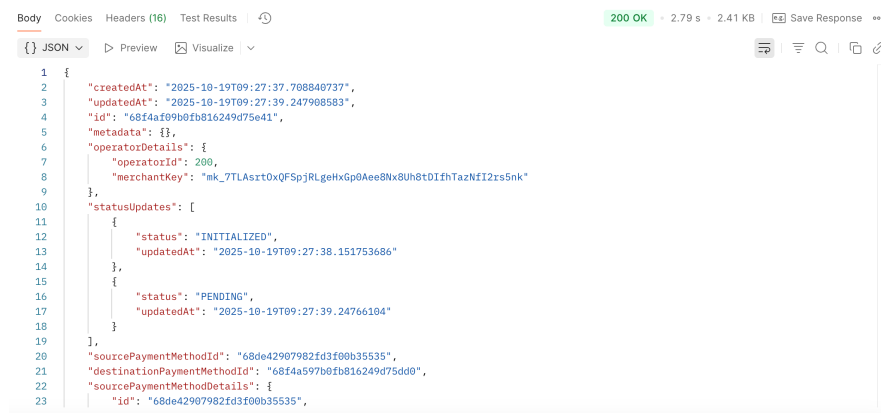
Allows the merchant to send money from their Neero account to a customer's Mobile Money wallet.

Note : For transfer on sourcePaymentMethodId you add merchantPaymentMethodId and on destinationPaymentMethodId you add mobileMoneyPaymentMethodId.

Request body:



```
"sourcePaymentMethodId": "{{merchantPaymentMethodId}}",
"confirm": true
}
```



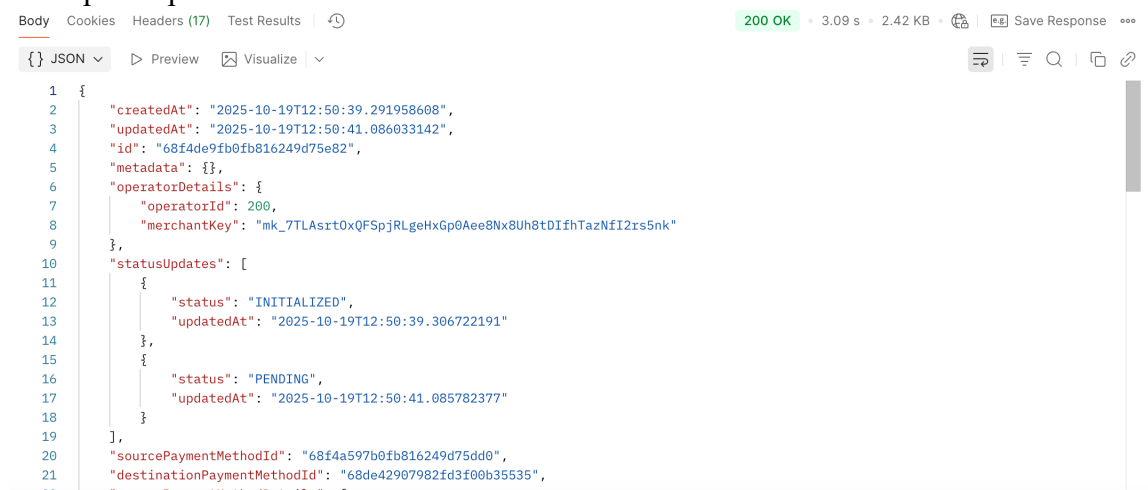
```
1 {
2   "createdAt": "2025-10-19T09:27:37.768840737",
3   "updatedAt": "2025-10-19T09:27:39.247908583",
4   "id": "68f4af09b0fb816249d75e41",
5   "metadata": {},
6   "operatorDetails": {
7     "operatorId": 200,
8     "merchantKey": "mk_7TLAszt0xQFSpjRLgeHxGp9Aee8Nx8Uh8tDIfhTazNfI2rs5nk"
9   },
10  "statusUpdates": [
11    {
12      "status": "INITIALIZED",
13      "updatedAt": "2025-10-19T09:27:38.151753686"
14    },
15    {
16      "status": "PENDING",
17      "updatedAt": "2025-10-19T09:27:39.24766104"
18    }
19  ],
20  "sourcePaymentMethodId": "68de42907982fd3f00b35535",
21  "destinationPaymentMethodId": "68f4a597b0fb816249d75dd0",
22  "sourcePaymentMethodDetails": {
23    "id": "68de42907982fd3f00b35535",
```

5. Check Transaction Status

Endpoint: GET /api/v1/transaction-intents/:id

This endpoint retrieves the current status of a transaction (either Cash-In or Cash-Out).

Example response:



```
1 {
2   "createdAt": "2025-10-19T12:50:39.291958608",
3   "updatedAt": "2025-10-19T12:50:41.086033142",
4   "id": "68f4de9fb0fb816249d75e82",
5   "metadata": {},
6   "operatorDetails": {
7     "operatorId": 200,
8     "merchantKey": "mk_7TLAszt0xQFSpjRLgeHxGp9Aee8Nx8Uh8tDIfhTazNfI2rs5nk"
9   },
10  "statusUpdates": [
11    {
12      "status": "INITIALIZED",
13      "updatedAt": "2025-10-19T12:50:39.306722191"
14    },
15    {
16      "status": "PENDING",
17      "updatedAt": "2025-10-19T12:50:41.085782377"
18    }
19  ],
20  "sourcePaymentMethodId": "68f4a597b0fb816249d75dd0",
21  "destinationPaymentMethodId": "68de42907982fd3f00b35535",
```

6. Possible Statuses

- ****INITIALIZED****: Transaction created but not yet processed.
- ****PENDING****: Awaiting confirmation from the mobile wallet.
- ****SUCCESSFUL****: Transaction completed successfully.
- ****FAILED****: Transaction failed.
- ****CANCELLED****: Transaction canceled by the user or system.

Conclusion

Integration steps summary:

1. Authenticate using your secret key (Basic Auth).
2. Create both Payment Methods (Mobile Money and Merchant).
3. Save their IDs.
4. Initiate a collection or transfer using Cash-In / Cash-Out endpoints.
5. Retrieve transaction status using GET /transaction-intents/:id.

Payment Method IDs are permanent and must be securely stored for future transactions.

Complete documentation : <https://app.theneo.io/neero/neero-payment-gateway/transaction-intents>