

# Installment Purchase - Customer Endpoints

**Author:** Josh S. Sakweli, Backend Lead Team

**Last Updated:** 2025-10-18

**Version:** v1.0

**Base URL:** `https://api.nextgate.com/api/v1/installments`

**Short Description:** This API provides endpoints for customers to manage their installment purchase agreements. Customers can view their agreements, track payment schedules, make manual payments, retry failed payments, calculate and process early payoffs, and cancel agreements. All endpoints require authentication and operate on the principle that customers can only access their own agreement data.

## Hints:

- All customer endpoints require authentication via Bearer token
- Customers can only access their own agreements (ownership validation applied)
- Payment processing uses wallet-based transactions
- Early payoff provides 75% discount on remaining interest
- Agreements can only be cancelled before the first payment is completed
- Failed payments can be retried up to 5 times
- Grace periods apply before first payment due date
- All amounts are in TZS (Tanzanian Shillings)

---

## 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": "2025-10-18T10:30:45",
```

```
"data": {  
  // Actual response data goes here  
}  
}
```

## Error Response Structure

```
{  
  "success": false,  
  "httpStatus": "BAD_REQUEST",  
  "message": "Error description",  
  "action_time": "2025-10-18T10:30:45",  
  "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, 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 with the following standard colors:

- **GET** - **GET** - Green (Safe, read-only operations)
- **POST** - **POST** - Blue (Create new resources)
- **DELETE** - **DELETE** - Red (Remove resources)

# Endpoints

## 1. Get My Agreements

**Purpose:** Retrieve all installment agreements for the authenticated customer, optionally filtered by status

**Endpoint:** GET `{base_url}/my-agreements`

**Access Level:**  Protected (Requires Authentication)

**Authentication:** Bearer Token

### Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication

### Query Parameters:

Parameter	Type	Required	Description	Validation	Default
status	string	No	Filter by agreement status	enum: PENDING_FIRST_PAYMENT, ACTIVE, COMPLETED, DEFAULTED, CANCELLED	null (all statuses)

### Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Agreements retrieved successfully",
  "action_time": "2025-10-18T10:30:45",
  "data": [
    {
      "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "agreementNumber": "INST-2025-12345",
      "productId": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
```

```

    "productName": "Samsung Galaxy S24 Ultra",
    "productImage": "https://cdn.example.com/products/samsung-s24.jpg",
    "shopId": "8d3a7b12-9c4e-4f8a-b5d2-3e6f7a8b9c0d",
    "shopName": "Tech World Store",
    "totalAmount": 2500000.00,
    "amountPaid": 500000.00,
    "amountRemaining": 2000000.00,
    "currency": "TZS",
    "paymentsCompleted": 1,
    "paymentsRemaining": 11,
    "totalPayments": 12,
    "progressPercentage": 20.0,
    "nextPaymentDate": "2025-11-18T00:00:00",
    "nextPaymentAmount": 166666.67,
    "agreementStatus": "ACTIVE",
    "agreementStatusDisplay": "ACTIVE",
    "createdAt": "2025-10-18T09:00:00",
    "completedAt": null,
    "canMakeEarlyPayment": true,
    "canCancel": false
  }
]
}

```

### Success Response Fields:

Field	Description
agreementId	Unique identifier for the agreement
agreementNumber	Human-readable agreement number (format: INST-YYYY-NNNNN)
productId	ID of the product being purchased
productName	Name of the product
productImage	URL to product image
shopId	ID of the shop selling the product
shopName	Name of the shop
totalAmount	Total amount to be paid including interest
amountPaid	Amount paid so far
amountRemaining	Amount still owed

Field	Description
currency	Currency code (TZS)
paymentsCompleted	Number of payments made
paymentsRemaining	Number of payments left
totalPayments	Total number of scheduled payments
progressPercentage	Percentage of payments completed
nextPaymentDate	Date when next payment is due
nextPaymentAmount	Amount of next payment
agreementStatus	Current status of the agreement
agreementStatusDisplay	Human-readable status
createdAt	When agreement was created
completedAt	When agreement was completed (null if not completed)
canMakeEarlyPayment	Whether early payoff is allowed
canCancel	Whether agreement can be cancelled

### Error Response JSON Sample:

```
{
  "success": false,
  "httpStatus": "UNAUTHORIZED",
  "message": "Token has expired",
  "action_time": "2025-10-18T10:30:45",
  "data": "Token has expired"
}
```

### Standard Error Types:

## Application-Level Exceptions (400-499)

- 401 UNAUTHORIZED: Authentication issues (empty, invalid, expired, or malformed tokens)
- 404 NOT\_FOUND: User not found or not authenticated

### Error Response Examples:

#### Unauthorized - Token Issues (401):

```
{
  "success": false,
```

```
"httpStatus": "UNAUTHORIZED",
"message": "Token has expired",
"action_time": "2025-10-18T10:30:45",
"data": "Token has expired"
}
```

*Not Found (404):*

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "User not found",
  "action_time": "2025-10-18T10:30:45",
  "data": "User not found"
}
```

## 2. Get My Active Agreements

**Purpose:** Retrieve only active installment agreements for the authenticated customer

**Endpoint:** **GET** `{base_url}/my-agreements/active`

**Access Level:**  Protected (Requires Authentication)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication

**Success Response JSON Sample:**

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Active agreements retrieved successfully",
  "action_time": "2025-10-18T10:30:45",
  "data": [
```

```
{
  "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "agreementNumber": "INST-2025-12345",
  "productId": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
  "productName": "Samsung Galaxy S24 Ultra",
  "productImage": "https://cdn.example.com/products/samsung-s24.jpg",
  "shopId": "8d3a7b12-9c4e-4f8a-b5d2-3e6f7a8b9c0d",
  "shopName": "Tech World Store",
  "totalAmount": 2500000.00,
  "amountPaid": 500000.00,
  "amountRemaining": 2000000.00,
  "currency": "TZS",
  "paymentsCompleted": 1,
  "paymentsRemaining": 11,
  "totalPayments": 12,
  "progressPercentage": 20.0,
  "nextPaymentDate": "2025-11-18T00:00:00",
  "nextPaymentAmount": 166666.67,
  "agreementStatus": "ACTIVE",
  "agreementStatusDisplay": "ACTIVE",
  "createdAt": "2025-10-18T09:00:00",
  "completedAt": null,
  "canMakeEarlyPayment": true,
  "canCancel": false
}
]
}
```

**Success Response Fields:** Same as Get My Agreements endpoint

**Error Response Examples:** Same error types as Get My Agreements endpoint

---

## 3. Get Agreement By ID

**Purpose:** Retrieve detailed information about a specific installment agreement by its ID

**Endpoint:** **GET** `{base_url}/agreements/{agreementId}`

**Access Level:**  Protected (Requires Authentication, Owner Only)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication

**Path Parameters:**

Parameter	Type	Required	Description	Validation
agreementId	UUID	Yes	Unique identifier of the agreement	Valid UUID format

**Success Response JSON Sample:**

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Agreement details retrieved successfully",
  "action_time": "2025-10-18T10:30:45",
  "data": {
    "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "agreementNumber": "INST-2025-12345",
    "customerId": "9b2e4d56-7c8a-4f9b-a3d1-5e6f7a8b9c0d",
    "customerName": "John Doe",
    "customerEmail": "john.doe@example.com",
    "productId": "7c9e6679-7425-40de-944b-e07fc1f90ae7",
    "productName": "Samsung Galaxy S24 Ultra",
    "productImage": "https://cdn.example.com/products/samsung-s24.jpg",
    "productPrice": 2000000.00,
    "quantity": 1,
    "shopId": "8d3a7b12-9c4e-4f8a-b5d2-3e6f7a8b9c0d",
    "shopName": "Tech World Store",
    "selectedPlanId": "4b5c6d7e-8f9a-4b1c-9d2e-3f4a5b6c7d8e",
    "planName": "12 Month Standard Plan",
    "paymentFrequency": "MONTHLY",
    "paymentFrequencyDisplay": "Monthly",
    "customFrequencyDays": null,
    "numberOfPayments": 12,
    "duration": "12 months",
```

```
"apr": 15.00,
"gracePeriodDays": 30,
"downPaymentAmount": 400000.00,
"financedAmount": 1600000.00,
"monthlyPaymentAmount": 166666.67,
"totalInterestAmount": 400000.00,
"totalAmount": 2400000.00,
"currency": "TZS",
"paymentsCompleted": 1,
"paymentsRemaining": 11,
"amountPaid": 566666.67,
"amountRemaining": 1833333.33,
"progressPercentage": 8.33,
"nextPaymentDate": "2025-11-18T00:00:00",
"nextPaymentAmount": 166666.67,
"agreementStatus": "ACTIVE",
"defaultCount": 0,
"createdAt": "2025-10-18T09:00:00",
"firstPaymentDate": "2025-11-18T00:00:00",
"lastPaymentDate": "2026-10-18T00:00:00",
"completedAt": null,
"fulfillmentTiming": "IMMEDIATE",
"shippedAt": "2025-10-18T14:00:00",
"deliveredAt": null,
"orderId": "1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d",
"shippingAddress": {
  "fullName": "John Doe",
  "phoneNumber": "+255712345678",
  "street": "123 Main Street",
  "city": "Dar es Salaam",
  "state": "Dar es Salaam",
  "postalCode": "12345",
  "country": "Tanzania"
},
"billingAddress": {
  "fullName": "John Doe",
  "phoneNumber": "+255712345678",
  "street": "123 Main Street",
  "city": "Dar es Salaam",
  "state": "Dar es Salaam",
```

```

    "postalCode": "12345",
    "country": "Tanzania"
  },
  "payments": [
    {
      "paymentId": "5c6d7e8f-9a0b-4c1d-8e2f-3a4b5c6d7e8f",
      "paymentNumber": 1,
      "scheduledAmount": 166666.67,
      "paidAmount": 166666.67,
      "principalPortion": 146666.67,
      "interestPortion": 20000.00,
      "remainingBalance": 1453333.33,
      "lateFee": null,
      "currency": "TZS",
      "paymentStatus": "COMPLETED",
      "paymentStatusDisplay": "COMPLETED",
      "dueDate": "2025-11-18T00:00:00",
      "paidAt": "2025-11-18T10:30:00",
      "attemptedAt": "2025-11-18T10:29:45",
      "paymentMethod": "WALLET",
      "transactionId": "TXN-2025-67890",
      "failureReason": null,
      "retryCount": 0,
      "daysUntilDue": null,
      "daysOverdue": null,
      "canPay": false,
      "canRetry": false
    }
  ],
  "canMakeEarlyPayment": true,
  "canCancel": false,
  "canUpdatePaymentMethod": false
}

```

### Success Response Fields:

Field	Description
agreementId	Unique identifier for the agreement
agreementNumber	Human-readable agreement number

Field	Description
customerId	ID of the customer
customerName	Customer's full name
customerEmail	Customer's email address
productId	ID of the product
productName	Name of the product
productImage	URL to product image
productPrice	Original product price
quantity	Quantity purchased
shopId	ID of the shop
shopName	Name of the shop
selectedPlanId	ID of the selected installment plan
planName	Name of the installment plan
paymentFrequency	Payment frequency enum value
paymentFrequencyDisplay	Human-readable payment frequency
customFrequencyDays	Custom frequency in days (if applicable)
numberOfPayments	Total number of payments
duration	Human-readable duration
apr	Annual Percentage Rate
gracePeriodDays	Grace period before first payment
downPaymentAmount	Down payment amount
financedAmount	Amount being financed
monthlyPaymentAmount	Amount per payment
totalInterestAmount	Total interest to be paid
totalAmount	Grand total (product + interest)
currency	Currency code
paymentsCompleted	Number of completed payments
paymentsRemaining	Number of remaining payments
amountPaid	Total amount paid so far
amountRemaining	Total amount remaining
progressPercentage	Percentage of completion
nextPaymentDate	Next payment due date
nextPaymentAmount	Next payment amount

Field	Description
agreementStatus	Current agreement status
defaultCount	Number of missed payments
createdAt	Agreement creation timestamp
firstPaymentDate	First payment due date
lastPaymentDate	Last payment due date
completedAt	Completion timestamp (null if not completed)
fulfillmentTiming	When product is shipped (IMMEDIATE or AFTER_PAYMENT)
shippedAt	Shipment timestamp
deliveredAt	Delivery timestamp
orderId	Associated order ID
shippingAddress	Shipping address object
billingAddress	Billing address object
payments	Array of payment objects with full details
canMakeEarlyPayment	Whether early payoff is allowed
canCancel	Whether agreement can be cancelled
canUpdatePaymentMethod	Whether payment method can be updated

### Error Response Examples:

#### Bad Request (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "You do not have access to this agreement",
  "action_time": "2025-10-18T10:30:45",
  "data": "You do not have access to this agreement"
}
```

#### Not Found (404):

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Agreement not found with ID: 3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "action_time": "2025-10-18T10:30:45",
}
```

```
"data": "Agreement not found with ID: 3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

## 4. Get Agreement By Number

**Purpose:** Retrieve detailed information about a specific installment agreement by its agreement number

**Endpoint:** **GET** `{base_url}/agreements/number/{agreementNumber}`

**Access Level:**  Protected (Requires Authentication, Owner Only)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication

**Path Parameters:**

Parameter	Type	Required	Description	Validation
agreementNumber	string	Yes	Agreement number (format: INST-YYYY-NNNNN)	Pattern: INST-\d{4}-\d{5}

**Success Response JSON Sample:** Same as Get Agreement By ID endpoint

**Success Response Fields:** Same as Get Agreement By ID endpoint

**Error Response Examples:** Same as Get Agreement By ID endpoint, with agreement number in error messages instead of ID

## 5. Get Agreement Payments

**Purpose:** Retrieve all payment records for a specific installment agreement

**Endpoint:** **GET** `{base_url}/agreements/{agreementId}/payments`

**Access Level:**  Protected (Requires Authentication, Owner Only)

## Authentication: Bearer Token

### Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication

### Path Parameters:

Parameter	Type	Required	Description	Validation
agreementId	UUID	Yes	Unique identifier of the agreement	Valid UUID format

### Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Payment history retrieved successfully",
  "action_time": "2025-10-18T10:30:45",
  "data": [
    {
      "paymentId": "5c6d7e8f-9a0b-4c1d-8e2f-3a4b5c6d7e8f",
      "paymentNumber": 1,
      "scheduledAmount": 166666.67,
      "paidAmount": 166666.67,
      "principalPortion": 146666.67,
      "interestPortion": 20000.00,
      "remainingBalance": 1453333.33,
      "lateFee": null,
      "currency": "TZS",
      "paymentStatus": "COMPLETED",
      "paymentStatusDisplay": "COMPLETED",
      "dueDate": "2025-11-18T00:00:00",
      "paidAt": "2025-11-18T10:30:00",
      "attemptedAt": "2025-11-18T10:29:45",
      "paymentMethod": "WALLET",
      "transactionId": "TXN-2025-67890",
      "failureReason": null,
      "retryCount": 0,
    }
  ]
}
```

```

    "daysUntilDue": null,
    "daysOverdue": null,
    "canPay": false,
    "canRetry": false
  },
  {
    "paymentId": "6d7e8f9a-0b1c-4d2e-9f3a-4b5c6d7e8f9a",
    "paymentNumber": 2,
    "scheduledAmount": 166666.67,
    "paidAmount": null,
    "principalPortion": 148533.33,
    "interestPortion": 18133.34,
    "remainingBalance": 1304800.00,
    "lateFee": null,
    "currency": "TZS",
    "paymentStatus": "SCHEDULED",
    "paymentStatusDisplay": "SCHEDULED",
    "dueDate": "2025-12-18T00:00:00",
    "paidAt": null,
    "attemptedAt": null,
    "paymentMethod": null,
    "transactionId": null,
    "failureReason": null,
    "retryCount": 0,
    "daysUntilDue": 61,
    "daysOverdue": null,
    "canPay": false,
    "canRetry": false
  }
]
}

```

### Success Response Fields:

Field	Description
paymentId	Unique identifier for the payment
paymentNumber	Sequential payment number (1, 2, 3...)
scheduledAmount	Amount scheduled to be paid
paidAmount	Amount actually paid (null if not paid)

Field	Description
principalPortion	Amount going toward principal
interestPortion	Amount going toward interest
remainingBalance	Balance after this payment
lateFee	Late fee if applicable
currency	Currency code
paymentStatus	Current payment status
paymentStatusDisplay	Human-readable status
dueDate	When payment is due
paidAt	When payment was made (null if not paid)
attemptedAt	Last payment attempt timestamp
paymentMethod	Payment method used
transactionId	Transaction reference ID
status	Payment status after processing
processedAt	When payment was processed
message	Success message
agreementUpdate	Updated agreement information
agreementUpdate.paymentsCompleted	Updated number of completed payments
agreementUpdate.paymentsRemaining	Updated number of remaining payments
agreementUpdate.amountPaid	Updated total amount paid
agreementUpdate.amountRemaining	Updated remaining amount
agreementUpdate.nextPaymentDate	Next payment due date
agreementUpdate.nextPaymentAmount	Next payment amount
agreementUpdate.agreementStatus	Updated agreement status
agreementUpdate.isCompleted	Whether agreement is now completed

### Error Response Examples:

#### *Bad Request - Insufficient Balance (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient wallet balance. Required: 166666.67 TZS, Available: 50000.00 TZS.
Please top up your wallet before the next payment attempt.",
```

```
"action_time": "2025-10-18T10:30:45",
"data": "Insufficient wallet balance. Required: 166666.67 TZS, Available: 50000.00 TZS.
Please top up your wallet before the next payment attempt."
}
```

#### *Bad Request - Payment Already Completed (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Payment is already completed",
  "action_time": "2025-10-18T10:30:45",
  "data": "Payment is already completed"
}
```

#### *Bad Request - Payment Not Due (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Payment is not due yet. Due date: 2025-12-18T00:00:00",
  "action_time": "2025-10-18T10:30:45",
  "data": "Payment is not due yet. Due date: 2025-12-18T00:00:00"
}
```

## 6. Get Upcoming Payments

**Purpose:** Retrieve all upcoming payments across all active agreements for the authenticated customer

**Endpoint:** GET `{base_url}/upcoming-payments`

**Access Level:**  Protected (Requires Authentication)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
--------	------	----------	-------------

Authorization	string	Yes	Bearer token for authentication
---------------	--------	-----	---------------------------------

### Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Upcoming payments retrieved successfully",
  "action_time": "2025-10-18T10:30:45",
  "data": [
    {
      "paymentId": "6d7e8f9a-0b1c-4d2e-9f3a-4b5c6d7e8f9a",
      "paymentNumber": 2,
      "scheduledAmount": 166666.67,
      "paidAmount": null,
      "principalPortion": 148533.33,
      "interestPortion": 18133.34,
      "remainingBalance": 1304800.00,
      "lateFee": null,
      "currency": "TZS",
      "paymentStatus": "SCHEDULED",
      "paymentStatusDisplay": "SCHEDULED",
      "dueDate": "2025-12-18T00:00:00",
      "paidAt": null,
      "attemptedAt": null,
      "paymentMethod": null,
      "transactionId": null,
      "failureReason": null,
      "retryCount": 0,
      "daysUntilDue": 61,
      "daysOverdue": null,
      "canPay": false,
      "canRetry": false
    }
  ]
}
```

**Success Response Fields:** Same as Get Agreement Payments endpoint

**Error Response Examples:** Standard authentication errors only

# 7. Make Manual Payment

**Purpose:** Process a manual payment for a specific scheduled installment

**Endpoint:** POST `{base_url}/agreements/{agreementId}/payments/{paymentId}/pay`

**Access Level:**  Protected (Requires Authentication, Owner Only)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication

**Path Parameters:**

Parameter	Type	Required	Description	Validation
agreementId	UUID	Yes	Unique identifier of the agreement	Valid UUID format
paymentId	UUID	Yes	Unique identifier of the payment	Valid UUID format

**Success Response JSON Sample:**

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Payment processed successfully",
  "action_time": "2025-10-18T10:30:45",
  "data": {
    "paymentId": "6d7e8f9a-0b1c-4d2e-9f3a-4b5c6d7e8f9a",
    "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "agreementNumber": "INST-2025-12345",
    "amount": 166666.67,
    "currency": "TZS",
    "paymentMethod": "WALLET",
    "transactionId": "TXN-2025-67891",
    "status": "COMPLETED",
  }
}
```

```

"processedAt": "2025-10-18T10:30:45",
"message": "Payment processed successfully",
"agreementUpdate": {
  "paymentsCompleted": 2,
  "paymentsRemaining": 10,
  "amountPaid": 733333.34,
  "amountRemaining": 1666666.66,
  "nextPaymentDate": "2026-01-18T00:00:00",
  "nextPaymentAmount": 166666.67,
  "agreementStatus": "ACTIVE",
  "isCompleted": false
}
}
}

```

### Success Response Fields:

Field	Description
paymentId	ID of the processed payment
agreementId	ID of the agreement
agreementNumber	Agreement number
amount	Amount paid
currency	Currency code
paymentMethod	Payment method used
transaction	

## 8. Retry Failed Payment

**Purpose:** Retry a failed payment that has not exceeded maximum retry attempts

**Endpoint:** POST `{base_url}/payments/{paymentId}/retry`

**Access Level:**  Protected (Requires Authentication, Owner Only)

**Authentication:** Bearer Token

### Request Headers:

Header	Type	Required	Description
--------	------	----------	-------------

Authorization	string	Yes	Bearer token for authentication
---------------	--------	-----	---------------------------------

### Path Parameters:

Parameter	Type	Required	Description	Validation
paymentId	UUID	Yes	Unique identifier of the payment to retry	Valid UUID format

### Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Payment retry processed successfully",
  "action_time": "2025-10-18T10:30:45",
  "data": {
    "paymentId": "6d7e8f9a-0b1c-4d2e-9f3a-4b5c6d7e8f9a",
    "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "agreementNumber": "INST-2025-12345",
    "amount": 166666.67,
    "currency": "TZS",
    "paymentMethod": "WALLET",
    "transactionId": "TXN-2025-67892",
    "status": "COMPLETED",
    "processedAt": "2025-10-18T10:30:45",
    "message": "Payment retry successful",
    "agreementUpdate": {
      "paymentsCompleted": 2,
      "paymentsRemaining": 10,
      "amountPaid": 733333.34,
      "amountRemaining": 1666666.66,
      "nextPaymentDate": "2026-01-18T00:00:00",
      "nextPaymentAmount": 166666.67,
      "agreementStatus": "ACTIVE",
      "isCompleted": false
    }
  }
}
```

**Success Response Fields:** Same as Make Manual Payment endpoint

## Error Response Examples:

### Bad Request - Cannot Retry (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Payment cannot be retried",
  "action_time": "2025-10-18T10:30:45",
  "data": "Payment cannot be retried"
}
```

### Bad Request - Max Retries Exceeded (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Maximum retry attempts (5) exceeded",
  "action_time": "2025-10-18T10:30:45",
  "data": "Maximum retry attempts (5) exceeded"
}
```

# 9. Preview Flexible Payment

**Purpose:** Calculate and preview how a flexible payment amount will be distributed across installment payments before processing

**Endpoint:** **POST** `{base_url}/installments/agreements/{agreementId}/early-flexible-payment/preview`

**Access Level:**  Protected (Requires Authentication)

**Authentication:** Bearer Token (JWT)

### Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication
Content-Type	string	Yes	Must be <code>application/json</code>

### Path Parameters:

Parameter	Type	Required	Description	Validation
agreementId	UUID	Yes	The ID of the installment agreement	Must be a valid UUID, agreement must belong to authenticated user

### Request JSON Sample:

```
{
  "amount": 500000.00
}
```

### Request Body Parameters:

Parameter	Type	Required	Description	Validation
amount	decimal	Yes	Amount the customer wants to pay	Must be > 0, cannot exceed remaining balance

### Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Flexible payment preview calculated successfully",
  "action_time": "2025-11-06T10:30:45",
  "data": {
    "requestedAmount": 500000.00,
    "minimumRequired": 150000.00,
    "maximumAllowed": 1200000.00,
    "isValid": true,
    "validationMessage": null,
    "impactedPayments": [
      {
        "paymentNumber": 3,
        "dueDate": "2025-12-01T00:00:00",
        "scheduledAmount": 200000.00,
        "currentPaid": 50000.00,
        "willApply": 150000.00,
        "willRemain": 0.00,
        "resultStatus": "Will be COMPLETED"
      }
    ],
  },
}
```

```

{
  "paymentNumber": 4,
  "dueDate": "2026-01-01T00:00:00",
  "scheduledAmount": 200000.00,
  "currentPaid": 0.00,
  "willApply": 200000.00,
  "willRemain": 0.00,
  "resultStatus": "Will be COMPLETED"
},
{
  "paymentNumber": 5,
  "dueDate": "2026-02-01T00:00:00",
  "scheduledAmount": 200000.00,
  "currentPaid": 0.00,
  "willApply": 150000.00,
  "willRemain": 50000.00,
  "resultStatus": "Will be PARTIALLY_PAID"
}
],
"paymentsWillComplete": 2,
"paymentsWillBePartial": 1,
"remainingAfter": 700000.00
}
}

```

### Success Response Fields:

Field	Description
requestedAmount	The amount customer wants to pay
minimumRequired	Minimum payment required (next incomplete payment amount)
maximumAllowed	Maximum payment allowed (total remaining balance)
isValid	Whether the requested amount is valid
validationMessage	Error message if amount is invalid, null if valid
impactedPayments	Array of payments that will be affected by this payment
impactedPayments[].paymentNumber	Sequential payment number
impactedPayments[].dueDate	When this payment is due
impactedPayments[].scheduledAmount	Original scheduled amount for this payment

Field	Description
impactedPayments[].currentPaid	Amount already paid towards this payment
impactedPayments[].willApply	How much of the flexible payment will apply here
impactedPayments[].willRemain	How much will remain unpaid after applying
impactedPayments[].resultStatus	Final status: "Will be COMPLETED" or "Will be PARTIALLY_PAID"
paymentsWillComplete	Number of payments that will be fully completed
paymentsWillBePartial	Number of payments that will be partially paid
remainingAfter	Total remaining balance after this payment

### Error Response Examples:

#### *Bad Request - Amount Too Low (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Minimum payment required: 150000.00 TZS",
  "action_time": "2025-11-06T10:30:45",
  "data": "Minimum payment required: 150000.00 TZS"
}
```

#### *Bad Request - Amount Too High (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Payment amount exceeds remaining balance. Use early payoff endpoint if paying off completely.",
  "action_time": "2025-11-06T10:30:45",
  "data": "Payment amount exceeds remaining balance. Use early payoff endpoint if paying off completely."
}
```

#### *Forbidden - Not Agreement Owner (403):*

```
{
  "success": false,
  "httpStatus": "FORBIDDEN",
  "message": "You do not have access to this agreement",
}
```

```
"action_time": "2025-11-06T10:30:45",
"data": "You do not have access to this agreement"
}
```

*Not Found - Agreement Not Found (404):*

```
{
  "success": false,
  "httpStatus": "NOT_FOUND",
  "message": "Agreement not found",
  "action_time": "2025-11-06T10:30:45",
  "data": "Agreement not found"
}
```

*Validation Error - Invalid Amount (422):*

```
{
  "success": false,
  "httpStatus": "UNPROCESSABLE_ENTITY",
  "message": "Validation failed",
  "action_time": "2025-11-06T10:30:45",
  "data": {
    "amount": "must be greater than 0"
  }
}
```

---

## 10. Process Flexible Payment

**Purpose:** Process a flexible payment that can pay multiple installments or partially pay upcoming installments

**Endpoint:** POST `{base_url}/installments/agreements/{agreementId}/early-flexible-payment`

**Access Level:**  Protected (Requires Authentication, Sufficient Wallet Balance)

**Authentication:** Bearer Token (JWT)

**Request Headers:**

Header	Type	Required	Description
--------	------	----------	-------------

Authorization	string	Yes	Bearer token for authentication
Content-Type	string	Yes	Must be <code>application/json</code>

### Path Parameters:

Parameter	Type	Required	Description	Validation
agreementId	UUID	Yes	The ID of the installment agreement	Must be a valid UUID, agreement must belong to authenticated user

### Request JSON Sample:

```
{
  "amount": 500000.00,
  "note": "Paying 3 months ahead"
}
```

### Request Body Parameters:

Parameter	Type	Required	Description	Validation
amount	decimal	Yes	Amount to pay	Must be > 0, between minimumRequired and maximumAllowed
note	string	No	Optional note about the payment	Max 500 characters

### Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Successfully paid 2 installments and partially paid 1 more",
  "action_time": "2025-11-06T10:30:45",
  "data": {
    "agreementId": "a8b3c4d5-e6f7-8901-2345-6789abcdef01",
    "agreementNumber": "INST-2025-12345",
    "totalAmountPaid": 500000.00,
    "currency": "TZS",
    "transactionId": "TXN-2025-67890",
  }
}
```

```
"processedAt": "2025-11-06T10:30:45",
"paymentsAffected": [
  {
    "paymentId": "p1a2b3c4-d5e6-f789-0123-456789abcdef",
    "paymentNumber": 3,
    "dueDate": "2025-12-01T00:00:00",
    "scheduledAmount": 200000.00,
    "amountApplied": 150000.00,
    "previouslyPaid": 50000.00,
    "newPaidAmount": 200000.00,
    "remaining": 0.00,
    "status": "COMPLETED",
    "wasCompleted": true
  },
  {
    "paymentId": "p2a3b4c5-d6e7-f890-1234-56789abcdef0",
    "paymentNumber": 4,
    "dueDate": "2026-01-01T00:00:00",
    "scheduledAmount": 200000.00,
    "amountApplied": 200000.00,
    "previouslyPaid": 0.00,
    "newPaidAmount": 200000.00,
    "remaining": 0.00,
    "status": "COMPLETED",
    "wasCompleted": true
  },
  {
    "paymentId": "p3a4b5c6-d7e8-f901-2345-6789abcdef01",
    "paymentNumber": 5,
    "dueDate": "2026-02-01T00:00:00",
    "scheduledAmount": 200000.00,
    "amountApplied": 150000.00,
    "previouslyPaid": 0.00,
    "newPaidAmount": 150000.00,
    "remaining": 50000.00,
    "status": "PARTIALLY_PAID",
    "wasCompleted": false
  }
],
"agreementUpdate": {
```

```

    "paymentsCompleted": 4,
    "paymentsPartial": 1,
    "paymentsRemaining": 8,
    "amountPaid": 1300000.00,
    "amountRemaining": 700000.00,
    "nextPaymentDate": "2026-02-01T00:00:00",
    "nextPaymentAmount": 50000.00,
    "agreementStatus": "ACTIVE",
    "isCompleted": false
  },
  "message": "Successfully paid 2 installments and partially paid 1 more"
}
}

```

### Success Response Fields:

Field	Description
agreementId	ID of the installment agreement
agreementNumber	Human-readable agreement number
totalAmountPaid	Total amount paid in this transaction
currency	Currency code (TZS)
transactionId	Unique transaction identifier from ledger system
processedAt	Timestamp when payment was processed
paymentsAffected	Array of payments that were affected
paymentsAffected[].paymentId	ID of the affected payment
paymentsAffected[].paymentNumber	Sequential payment number
paymentsAffected[].dueDate	When this payment is due
paymentsAffected[].scheduledAmount	Original scheduled amount
paymentsAffected[].amountApplied	How much of flexible payment was applied here
paymentsAffected[].previouslyPaid	Amount that was previously paid
paymentsAffected[].newPaidAmount	Total amount now paid for this payment
paymentsAffected[].remaining	Amount still remaining for this payment
paymentsAffected[].status	New payment status (COMPLETED or PARTIALLY_PAID)
paymentsAffected[].wasCompleted	Whether this payment became fully completed
agreementUpdate	Updated agreement summary
agreementUpdate.paymentsCompleted	Total number of completed payments

Field	Description
agreementUpdate.paymentsPartial	Number of partially paid payments
agreementUpdate.paymentsRemaining	Number of payments still remaining
agreementUpdate.amountPaid	Total amount paid on agreement
agreementUpdate.amountRemaining	Total amount still remaining
agreementUpdate.nextPaymentDate	When next payment is due
agreementUpdate.nextPaymentAmount	Amount of next payment (remaining amount)
agreementUpdate.agreementStatus	Current agreement status
agreementUpdate.isCompleted	Whether agreement is fully completed
message	Human-readable success message

### Error Response Examples:

#### *Bad Request - Insufficient Balance (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient wallet balance. Required: 500000.00 TZS, Available: 300000.00 TZS",
  "action_time": "2025-11-06T10:30:45",
  "data": "Insufficient wallet balance. Required: 500000.00 TZS, Available: 300000.00 TZS"
}
```

#### *Bad Request - Agreement Not Active (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Cannot make payment on inactive agreement. Status: COMPLETED",
  "action_time": "2025-11-06T10:30:45",
  "data": "Cannot make payment on inactive agreement. Status: COMPLETED"
}
```

#### *Bad Request - Wallet Not Active (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Wallet is not active",
}
```

```
"action_time": "2025-11-06T10:30:45",
"data": "Wallet is not active"
}
```

#### Validation Error (422):

```
{
  "success": false,
  "httpStatus": "UNPROCESSABLE_ENTITY",
  "message": "Validation failed",
  "action_time": "2025-11-06T10:30:45",
  "data": {
    "amount": "must be greater than 0",
    "note": "size must be between 0 and 500"
  }
}
```

## 11. Calculate Early Payoff

**Purpose:** Calculate the amount required to pay off the entire agreement early with interest discount

**Endpoint:** **GET** `{base_url}/agreements/{agreementId}/early-payoff`

**Access Level:**  Protected (Requires Authentication, Owner Only)

**Authentication:** Bearer Token

#### Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication

#### Path Parameters:

Parameter	Type	Required	Description	Validation
agreementId	UUID	Yes	Unique identifier of the agreement	Valid UUID format

#### Success Response JSON Sample:

```

{
  "success": true,
  "httpStatus": "OK",
  "message": "Early payoff calculation completed",
  "action_time": "2025-10-18T10:30:45",
  "data": {
    "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "paymentsCompleted": 2,
    "paymentsRemaining": 10,
    "amountPaid": 733333.34,
    "remainingPrincipal": 1320000.00,
    "unaccruedInterest": 346666.66,
    "interestRebate": 260000.00,
    "payoffWithRebate": 1406666.66,
    "payoffWithoutRebate": 1666666.66,
    "savingsVsScheduled": 260000.00,
    "rebatePolicy": "75% discount on remaining interest for early payoff",
    "calculatedAt": "2025-10-18T10:30:45"
  }
}

```

### Success Response Fields:

Field	Description
agreementId	ID of the agreement
paymentsCompleted	Number of payments already made
paymentsRemaining	Number of payments left
amountPaid	Total amount paid so far
remainingPrincipal	Principal amount remaining
unaccruedInterest	Interest not yet charged
interestRebate	Interest discount amount (75% of remaining interest)
payoffWithRebate	Early payoff amount with discount applied
payoffWithoutRebate	Full remaining amount if paid on schedule
savingsVsScheduled	Amount saved by paying off early
rebatePolicy	Description of rebate policy
calculatedAt	When calculation was performed

### Error Response Examples:

### Bad Request - Not Eligible (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Early payoff not available for this agreement",
  "action_time": "2025-10-18T10:30:45",
  "data": "Early payoff not available for this agreement"
}
```

### Bad Request - Agreement Not Active (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Agreement is not active. Status: COMPLETED",
  "action_time": "2025-10-18T10:30:45",
  "data": "Agreement is not active. Status: COMPLETED"
}
```

## 12. Process Early Payoff

**Purpose:** Process full early payment of the agreement with interest discount applied

**Endpoint:** POST `{base_url}/agreements/{agreementId}/early-payoff`

**Access Level:**  Protected (Requires Authentication, Owner Only)

**Authentication:** Bearer Token

#### Request Headers:

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication

#### Path Parameters:

Parameter	Type	Required	Description	Validation
agreementId	UUID	Yes	Unique identifier of the agreement	Valid UUID format

## Success Response JSON Sample:

```
{
  "success": true,
  "httpStatus": "OK",
  "message": "Early payoff processed successfully",
  "action_time": "2025-10-18T10:30:45",
  "data": {
    "paymentId": null,
    "agreementId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "agreementNumber": "INST-2025-12345",
    "amount": 1406666.66,
    "currency": "TZS",
    "paymentMethod": "WALLET",
    "transactionId": null,
    "status": "COMPLETED",
    "processedAt": "2025-10-18T10:30:45",
    "message": "Early payoff processed successfully",
    "agreementUpdate": {
      "paymentsCompleted": 12,
      "paymentsRemaining": 0,
      "amountPaid": 2400000.00,
      "amountRemaining": 0.00,
      "nextPaymentDate": null,
      "nextPaymentAmount": null,
      "agreementStatus": "COMPLETED",
      "isCompleted": true
    }
  }
}
```

## Success Response Fields:

Field	Description
paymentId	Payment ID (null for early payoff)
agreementId	ID of the agreement
agreementNumber	Agreement number
amount	Total early payoff amount paid
currency	Currency code

Field	Description
paymentMethod	Payment method used
transactionId	Transaction reference (null for early payoff)
status	Payment status (COMPLETED)
processedAt	When payoff was processed
message	Success message
agreementUpdate	Updated agreement information
agreementUpdate.paymentsCompleted	Updated to total number of payments
agreementUpdate.paymentsRemaining	Set to 0
agreementUpdate.amountPaid	Updated to total amount
agreementUpdate.amountRemaining	Set to 0
agreementUpdate.nextPaymentDate	Set to null
agreementUpdate.nextPaymentAmount	Set to null
agreementUpdate.agreementStatus	Set to COMPLETED
agreementUpdate.isCompleted	Set to true

### Error Response Examples:

#### Bad Request - Insufficient Balance (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Insufficient wallet balance for early payoff. Required: 1406666.66 TZS,
Available: 500000.00 TZS",
  "action_time": "2025-10-18T10:30:45",
  "data": "Insufficient wallet balance for early payoff. Required: 1406666.66 TZS, Available:
500000.00 TZS"
}
```

#### Bad Request - Not Eligible (400):

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Agreement is not active. Status: DEFAULTED",
  "action_time": "2025-10-18T10:30:45",
  "data": "Agreement is not active. Status: DEFAULTED"
}
```

```
}
```

# 13. Cancel Agreement

**Purpose:** Cancel an installment agreement before any payments have been completed

**Endpoint:** **POST** `{base_url}/agreements/{agreementId}/cancel`

**Access Level:**  Protected (Requires Authentication, Owner Only)

**Authentication:** Bearer Token

**Request Headers:**

Header	Type	Required	Description
Authorization	string	Yes	Bearer token for authentication
Content-Type	string	Yes	application/json

**Path Parameters:**

Parameter	Type	Required	Description	Validation
agreementId	UUID	Yes	Unique identifier of the agreement	Valid UUID format

**Request JSON Sample:**

```
{
  "reason": "Changed my mind about the purchase. Found a better deal elsewhere."
}
```

**Request Body Parameters:**

Parameter	Type	Required	Description	Validation
reason	string	Yes	Reason for cancellation	Min: 1, Max: 500 characters

**Success Response JSON Sample:**

```
{
  "success": true,
}
```

```
"httpStatus": "OK",
"message": "Agreement cancelled successfully",
"action_time": "2025-10-18T10:30:45",
"data": null
}
```

**Success Response Fields:** No data returned on successful cancellation

### **Error Response Examples:**

*Bad Request - Cannot Cancel (400):*

```
{
  "success": false,
  "httpStatus": "BAD_REQUEST",
  "message": "Agreement cannot be cancelled. Only agreements with no completed payments can be cancelled.",
  "action_time": "2025-10-18T10:30:45",
  "data": "Agreement cannot be cancelled. Only agreements with no completed payments can be cancelled."
}
```

*Validation Error (422):*

```
{
  "success": false,
  "httpStatus": "UNPROCESSABLE_ENTITY",
  "message": "Validation failed",
  "action_time": "2025-10-18T10:30:45",
  "data": {
    "reason": "Cancellation reason is required"
  }
}
```

---

# Quick Reference Guide

## HTTP Method Badge Code Templates

**GET Badge:**

```
<span style="background-color: #28a745; color: white; padding: 4px 8px; border-radius: 4px; font-family: monospace; font-size: 12px; font-weight: bold;">GET</span>
```

### POST Badge:

```
<span style="background-color: #007bff; color: white; padding: 4px 8px; border-radius: 4px; font-family: monospace; font-size: 12px; font-weight: bold;">POST</span>
```

## Common HTTP Status Codes

- **200 OK**: Successful GET/POST request
- **400 Bad Request**: Invalid request data or business logic violation
- **401 Unauthorized**: Authentication required/failed
- **403 Forbidden**: Insufficient permissions
- **404 Not Found**: Resource not found
- **422 Unprocessable Entity**: Validation errors
- **500 Internal Server Error**: Server error

## Authentication

- **Bearer Token**: Include `Authorization: Bearer your_token` in headers
- All customer endpoints require valid authentication
- Token must belong to the customer who owns the agreement

## Agreement Status Flow

1. **PENDING\_FIRST\_PAYMENT**: Down payment made, waiting for grace period
2. **ACTIVE**: Currently paying installments
3. **COMPLETED**: Fully paid off
4. **DEFAULTED**: Missed 2+ payments, in collections
5. **CANCELLED**: User cancelled before first payment

## Payment Status Types

- **SCHEDULED**: Not due yet
- **PENDING**: Due today, awaiting payment
- **PROCESSING**: Payment in progress
- **COMPLETED**: Successfully paid
- **FAILED**: Payment attempt failed
- **LATE**: Past due date
- **SKIPPED**: Missed completely
- **WAIVED**: Forgiven (special cases)

# Business Rules

- **Early Payoff Discount:** 75% off remaining interest
- **Maximum Retries:** 5 attempts per payment
- **Default Threshold:** 2 missed payments triggers DEFAULTED status
- **Cancellation Window:** Only before first payment completed
- **Payment Method:** Wallet payments only
- **Currency:** All amounts in TZS (Tanzanian Shillings)

# Common Error Scenarios

1. **Insufficient Wallet Balance:** Top up wallet before payment
  2. **Payment Not Due:** Wait until due date or make early payoff
  3. **Agreement Defaulted:** Contact support for resolution
  4. **Max Retries Exceeded:** Contact support
  5. **Ownership Validation Failed:** Can only access own agreements
- 

# Notes for Developers

## Idempotency

- Payment processing endpoints are idempotent
- Duplicate payment requests will return existing payment status
- Use transaction IDs to track payment processing

## Rate Limiting

- Standard rate limits apply: 100 requests per hour per user
- Payment processing has additional throttling for security

## Webhooks

- Payment success/failure events trigger notifications
- Agreement completion triggers order creation (for AFTER\_PAYMENT fulfillment)
- Integration with notification service for email/SMS alerts

## Testing

- Use sandbox environment for testing: `https://sandbox-api.nextgate.com`

- Test user accounts available with pre-loaded wallets
- Mock payment processing for integration testing

# Support

- For API issues: [api-support@nextgate.com](mailto:api-support@nextgate.com)
  - For business logic questions: [product@nextgate.com](mailto:product@nextgate.com)
  - Emergency contact: +255-XXX-XXX-XXXId | Transaction reference ID | | failureReason | Reason if payment failed | | retryCount | Number of retry attempts | | daysUntilDue | Days until payment is due (negative if overdue) | | daysOverdue | Days payment is overdue (null if not overdue) | | canPay | Whether payment can be made now | | canRetry | Whether payment can be retried |
- 
- 

Revision #6

Created 18 October 2025 19:40:46 by Admin Qbit

Updated 11 December 2025 08:27:31 by Admin Qbit