

# Transaction API Documentation

## Overview

This document outlines the API endpoints for managing transactions, including the new features for listing (with filters) and cancelling transactions.

## 1. Get Transactions (List)

Retrieve a list of transactions with support for pagination, status filtering, and searching.

**Endpoint:** GET /transaction

**Query Parameters:**

| Parameter | Type    | Default | Description  |
|-----------|---------|---------|--|
| page      | Integer | 1       | Page number for pagination.                            |
| limit     | Integer | 10      | Number of items per page.                              |
| status    | String  | -       | Filter by status: Success, Pending, Cancelled, Failed. |
| search    | String  | -       | Search term for invoiceNumber or receiptNumber.        |

**Example Request:**

```
GET /transaction?page=1&limit=10&status=Cancelled&search=INV-001
Authorization: Bearer <token>
```

**Success Response (200 OK):**

```
{
  "data": [
    {
      "_id": "653b6...",
      "customerId": {
        "_id": "...",
        "fullName": "John Doe",
        "coreCustId": "1000659"
      },
      "invoiceNumber": "INV-001",
      "status": "Cancelled",
      "totalAmount": 1000,
      "createdAt": "2023-10-27T..."
    }
  ],
  "pagination": {
    "total": 50,
    "page": 1,
    "limit": 10,
  }
}
```

```
    "pages": 5
  }
}
```

---

## 2. Cancel Transaction

Cancel an existing transaction by its unique ID. This updates the transaction status to 'Cancelled'.

**Endpoint:** PATCH /transaction/:id/cancel

**Path Parameters:**

| Parameter | Type   | Description                                   |
|-----------|--------|---|
| id        | String | The MongoDB _id of the transaction to cancel. |

**Example Request:**

```
PATCH /transaction/653b6f9a.../cancel
Authorization: Bearer <token>
```

**Success Response (200 OK):**

```
{
  "message": "Transaction cancelled successfully",
  "transaction": {
    "_id": "653b6f9a...",
    "status": "Cancelled",
    "invoiceNumber": "INV-001",
    ...
  }
}
```

**Error Responses:**

- 400 Bad Request : "Transaction is already cancelled"
- 404 Not Found : "Transaction not found"
- 500 Internal Server Error

---

## 3. Create Transaction

Create a new transaction record.

**Endpoint:** POST /transaction

**Body:**

```
{
  "customerId": "1000659",
  "customerType": "IndividualProfile",
```

```
"transactionDate": "2023-10-27",
"invoiceNumber": "INV-001",
"totalAmount": 1000,
...
}
```

**Note:** Ensure you include the `Authorization: Bearer <token>` header in all requests.

## 4. List Customers (for Transaction Entry)

Retrieve a list of customers (Individual/Corporate) with support for specific search fields used in the Transaction Entry UI.

**Endpoint:** `GET /transaction/customers`

**Query Parameters:**

| Parameter    | Type    | Default | Description  |
|--------------|---------|---------|--|
| page         | Integer | 1       | Page number.                                       |
| limit        | Integer | 10      | Items per page.                                    |
| customerType | String  | -       | Individual or Corporate. If omitted, returns both. |
| searchType   | String  | -       | Field to search: Name, Customer ID, Mobile.        |
| searchValue  | String  | -       | Value to search for.                               |

**Example Request:**

```
GET /transaction/customers?
customerType=Individual&searchType=Mobile&searchValue=9715000000
Authorization: Bearer <token>
```