API Endpoints Documentation

Instanvi Payment Platform - Backend API Specification

This document outlines all API endpoints required for the Instanvi payment platform frontend application. This is a comprehensive guide for backend developers to implement a fully functional payment platform specifically designed for the Cameroon market.

Version: 1.0
Last Updated: 2025
Market Focus: Cameroon (CM)
Primary Currencies: XAF (Central African CFA Franc), USD
Database: PostgreSQL (Required)

Base URL

- **Development**: `http://localhost:YOUR_BACKEND_PORT`
- **Production**: `https://api.instanvi.com` or `https://api.yourdomain.com`

All endpoints should be prefixed with `/api` unless otherwise specified.

Important Notes for Cameroon Market

- **Payment Methods Supported:**
    - MTN Mobile Money (Primary)
    - Orange Money (Primary)
    - No credit cards, bank transfers, or other methods currently supported
- **Currency Support:**
    - XAF (Central African CFA Franc) - Primary currency
    - USD (United States Dollar) - Secondary currency
- **Regional Requirements:**
    - Uses **Region/Province** instead of postal codes (Cameroon doesn't use postal codes)
    - Registration number format: `RC/DLA/2024/A/12345`
    - VAT number format: `M123456789` (10 digits starting with M)
    - Phone number format: `+237 6 12 34 56 78`

Database

PostgreSQL Database

The backend **MUST** use **PostgreSQL** as the primary database system. This is a required specification for the Instanvi platform.

Database Requirements:

- **Database System:** PostgreSQL (version 12.0 or higher recommended)
- **Connection Pooling:** Required for production environments
- **Transactions:** Use database transactions for critical operations (payments, refunds, balance updates)

- **Migrations:** Use a migration system (e.g., Knex.js, TypeORM, Sequelize) for schema management
- **Backups:** Implement automated daily backups with point-in-time recovery capability

Key Database Considerations:

1. **ACID Compliance:** PostgreSQL provides full ACID compliance, essential for financial transactions
2. **JSON Support:** Use JSON/JSONB columns for flexible metadata storage
3. **Full-Text Search:** Leverage PostgreSQL's full-text search for search functionality
4. **Indexing:** Implement proper indexes for frequently queried fields (user_id, payment_id, email, etc.)
5. **Foreign Keys:** Use foreign key constraints to maintain referential integrity
6. **Unique Constraints:** Enforce uniqueness on emails, usernames, API keys, etc.

Recommended Database Schema Patterns:

- Use UUIDs (`uuid` type) for primary keys instead of auto-incrementing integers for better scalability
- Implement soft deletes using a `deleted_at` timestamp column
- Use timestamps (`created_at` , `updated_at`) on all tables
- Store monetary values as `NUMERIC` or `DECIMAL` with appropriate precision (e.g., `NUMERIC(19, 4)`)
- Use ENUM types for status fields (payment status, refund status, etc.)

Database Connection:

- Use connection strings with SSL enabled in production
- Implement connection retry logic with exponential backoff
- Monitor connection pool usage and adjust pool size accordingly
- Use read replicas for scaling read operations if needed

Authentication

All protected endpoints require authentication via Bearer token in the Authorization header:

```
Authorization: Bearer <access_token>
```

Table of Contents

Cameroon-Specific Data Formats

Phone Number Format

- Format: `+237 6 12 34 56 78` or `237612345678`
- Country code: `+237`
- Length: 9 digits after country code
- Example: `+237 6 12 34 56 78`

Business Registration Number

- Format: `RC/DLA/2024/A/12345`
- Pattern: `RC/[CITY_CODE]/[YEAR]/[LETTER]/[NUMBER]`
- Example: `RC/DLA/2024/A/12345` (Douala, 2024)

VAT Number

- Format: `M123456789`
- Pattern: `M` followed by 9 digits
- Example: `M123456789`

Regions (Provinces)

Valid Cameroon regions:

- Adamaoua
- Centre
- Est
- Extrême-Nord
- Littoral
- Nord
- Nord-Ouest
- Ouest
- Sud
- Sud-Ouest

Supported Countries & Nationalities

- CM - Cameroon
- NG - Nigeria
- TD - Chad
- CF - Central African Republic
- CG - Congo
- GA - Gabon
- GQ - Equatorial Guinea
- FR - France
- GB - United Kingdom
- US - United States

- DE - Germany
- OTHER - Other

---

1. Authentication Endpoints

1.1 User Registration

POST `/api/auth/signup`

Request Body:

```json
{
  "username": "string",
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "password": "string"
}
```

Response (201):

```json
{
  "success": true,
  "message": "Account created successfully",
  "data": {
    "userId": "string",
    "email": "string",
    "isVerified": false
  }
}
```

1.2 User Login

POST `/api/auth/login`

Request Body:

```json
{
  "email": "string",
  "password": "string"
}
```

Response (200):

```json
{
  "success": true,
  "data": {
    "accessToken": "string",
    "refreshToken": "string",
    "user": {
      "id": "string",
      "email": "string",
      "username": "string",
      "firstName": "string",
      "lastName": "string",
      "isVerified": boolean,
      "role": "customer" | "admin"
    }
  }
}
```

1.3 Send Email Verification OTP

POST `/api/auth/send-verification-otp`

Request Body:

```
{
  "email": "string"
}
```

1.4 Verify Email OTP

POST `/api/auth/verify-otp`

Request Body:

```
{
  "email": "string",
  "otp": "string"
}
```

1.5 Logout

POST `/api/auth/logout`

Headers: `Authorization: Bearer <token>`

1.6 Refresh Token

POST `/api/auth/refresh`

Request Body:

```
{
  "refreshToken": "string"
}
```

1.7 Forgot Password

POST `/api/auth/forgot-password`

Request Body:

```json
{
  "email": "string"
}
```

1.8 Reset Password

POST `/api/auth/reset-password`

Request Body:

```json
{
  "token": "string",
  "newPassword": "string",
  "confirmPassword": "string"
}
```

---

2. User Profile Endpoints

2.1 Get Current User Profile

GET `/api/users/me`

Headers: `Authorization: Bearer <token>`

Response (200):

```json
{
  "success": true,
  "data": {
    "id": "string",
    "email": "string",
    "username": "string",
    "firstName": "string",
    "lastName": "string",
    "phone": "string | null",
    "isVerified": boolean,
    "role": "string"
  }
}
```

2.2 Update User Profile

PUT `/api/users/me`

Headers: `Authorization: Bearer <token>`

Request Body:

```json
{
  "firstName": "string",
  "lastName": "string",
  "phone": "string"
}
```

2.3 Change Password

PUT `/api/users/me/password`

Headers: `Authorization: Bearer <token>`

Request Body:

```json
{
  "currentPassword": "string",
  "newPassword": "string",
  "confirmPassword": "string"
}
```

2.4 Enable/Disable Two-Factor Authentication

PUT `/api/users/me/2fa`

Headers: `Authorization: Bearer <token>`

Request Body:

```json
{
  "enabled": boolean
}
```

---

3. Account Settings Endpoints

3.1 Get Account Settings

GET `/api/users/me/settings`

Headers: `Authorization: Bearer <token>`

Response (200):

```json
{
  "success": true,
  "data": {
    "language": "en" | "fr" | "de" | "nl" | "es",
    "timezone": "string",
    "testMode": boolean,
    "emailNotifications": boolean,
    "pushNotifications": boolean,
    "marketingEmails": boolean
  }
}
```

3.2 Update Account Settings

PUT `/api/users/me/settings`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "emailNotifications": boolean,
  "pushNotifications": boolean,
  "marketingEmails": boolean
}
```

Notes:

- Language, timezone, and test mode settings have been removed from the frontend
- Only notification preferences are configurable

---

4. Onboarding Endpoints

4.1 Submit Stakeholder Information

POST `/api/onboarding/stakeholder`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "phone": "string",
  "dateOfBirth": "YYYY-MM-DD",
  "nationality": "CM" | "NG" | "TD" | "CF" | "CG" | "GA" | "GQ" | "FR" | "GB" | "US" | "DE" | "OTHER",
  "address": "string",
  "city": "string",
  "region": "string | null",
  "country": "CM" | "NG" | "TD" | "CF" | "CG" | "GA" | "GQ" | "FR" | "GB" | "US" | "DE" | "OTHER"
}
```

Notes:

- `region` replaces `postalCode` (Cameroon doesn't use postal codes)
- Valid regions for Cameroon: "Adamaoua", "Centre", "Est", "Extrême-Nord", "Littoral", "Nord", "Nord-Ouest", "Ouest", "Sud", "Sud-Ouest"

- `nationality` and `country` support Cameroon and neighboring countries

4.2 Submit Business Activity

POST `/api/onboarding/business-activity`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "businessName": "string",
  "businessType": "entreprise-individuelle" | "sarl" | "sa" | "snc" | "association" |
"autre",
  "industry": "agriculture" | "commerce" | "ecommerce" | "services" | "technologie" |
"telecom" | "transport" | "sante" | "education" | "tourisme" | "bancaire" | "extraction" |
"autre",
  "businessRegistrationNumber": "string",
  "vatNumber": "string | null",
  "website": "string | null",
  "description": "string"
}
```

Notes:

- `businessType` options:
  - `entreprise-individuelle` - Sole Proprietorship
  - `sarl` - SARL (Limited Liability Company)
  - `sa` - SA (Public Limited Company)
  - `snc` - SNC (General Partnership)
  - `association` - Association
  - `autre` - Other
- `businessRegistrationNumber` format: `RC/DLA/2024/A/12345` (Cameroon registration format)
- `vatNumber` format: `M123456789` (10 digits starting with M)
- Website format: `https://www.exemple.cm`

4.3 Submit Payment Methods

POST `/api/onboarding/payment-methods`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "paymentMethods": ["mtn-momo", "orange-money"]
}
```

Notes:

- **Cameroon Market:** Only MTN Mobile Money and Orange Money are supported
- `mtn-momo` - MTN Mobile Money
- `orange-money` - Orange Money

- Other payment methods are not available for this market

4.4 Upload ID Document

POST `/api/onboarding/id-document`

Headers: `Authorization: Bearer <token>` , `Content-Type: multipart/form-data`

Request Body (Form Data):

```
documentType: "Passport" | "National ID" | "Driver's License"
frontImage: File (required)
backImage: File (required for "National ID", optional for others)
```

Notes:

- For `documentType: "National ID"` (CNI), **both** `frontImage` and `backImage` are **required**
- For `Passport` and `Driver's License` , only `frontImage` is required
- Accepted file types: PNG, JPG, PDF
- Maximum file size: 10MB per file

4.5 Get Onboarding Progress

GET `/api/onboarding/progress`

Headers: `Authorization: Bearer <token>`

Response (200):

```
{
  "success": true,
  "data": {
    "completedSteps": ["stakeholder", "business", "payment", "document"],
    "currentStep": "stakeholder" | "business" | "payment" | "document",
    "isComplete": boolean,
    "progressPercentage": 0-100
  }
}
```

Notes:

- Onboarding consists of **4 steps** (bank account linking step removed):
  1. `stakeholder` - Stakeholder Information
  2. `business` - Business Activity
  3. `payment` - Payment Methods
  4. `document` - ID Document

---

5. Payment Endpoints

5.1 Create Payment Link

POST `/api/payments/links`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "type": "Fixed" | "Subscription" | "Donation",
  "amount": number,
  "currency": "XAF" | "USD",
  "description": "string",
  "expiryDate": "YYYY-MM-DD" | null,
  "redirectUrl": "string | null",
  "reusable": boolean,
  "paymentMethods": ["MTN Mobile Money", "Orange Money"],
  "saveUrl": boolean
}
```

Notes:

- **Cameroon Market:** Only `XAF` and `USD` currencies supported
- **Payment Methods:** Must be one or both of: `"MTN Mobile Money"`, `"Orange Money"`
- `type` options: `Fixed`, `Subscription`, `Donation`

Response (201):

```
{
  "success": true,
  "data": {
    "paymentLinkId": "string",
    "url": "string",
    "amount": number,
    "currency": "string",
    "status": "active",
    "createdAt": "ISO8601"
  }
}
```

5.2 List Payment Links

GET `/api/payments/links`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `status` (optional): "active" | "expired" | "cancelled"
- `search` (optional): string

5.3 Get Payment Link Details

GET `/api/payments/links/:linkId`

Headers: `Authorization: Bearer <token>`

5.4 List Payments

GET `/api/payments`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `status` (optional): "pending" | "paid" | "failed" | "expired" | "cancelled"
- `startDate` (optional): YYYY-MM-DD
- `endDate` (optional): YYYY-MM-DD
- `search` (optional): string
- `amountMin` (optional): number
- `amountMax` (optional): number

Response (200):

```
{
  "success": true,
  "data": {
    "payments": [
      {
        "id": "string",
        "paymentLinkId": "string",
        "amount": number,
        "currency": "string",
        "description": "string",
        "status": "string",
        "paymentMethod": "string",
        "customer": {
          "name": "string",
          "email": "string"
        },
        "createdAt": "ISO8601",
        "paidAt": "ISO8601 | null"
      }
    ],
    "pagination": {
      "page": number,
      "limit": number,
      "total": number,
      "totalPages": number
    }
  }
}
```

5.5 Get Payment Details

GET `/api/payments/:paymentId`

Headers: `Authorization: Bearer <token>`

5.6 Cancel Payment

POST `/api/payments/:paymentId/cancel`

Headers: `Authorization: Bearer <token>`

---

6. Refunds Endpoints

6.1 Create Refund

POST `/api/payments/:paymentId/refunds`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "amount": number,
  "description": "string",
  "reason": "string | null"
}
```

Response (201):

```
{
  "success": true,
  "data": {
    "refundId": "string",
    "paymentId": "string",
    "amount": number,
    "currency": "string",
    "status": "pending" | "processing" | "completed" | "failed",
    "createdAt": "ISO8601"
  }
}
```

6.2 List Refunds

GET `/api/refunds`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `status` (optional): "pending" | "processing" | "completed" | "failed"
- `paymentId` (optional): string

6.3 Get Refund Details

GET `/api/refunds/:refundId`

Headers: `Authorization: Bearer <token>`

---

7. Chargebacks Endpoints

7.1 List Chargebacks

GET `/api/chargebacks`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `status` (optional): "open" | "won" | "lost"

Response (200):

```
{
  "success": true,
  "data": {
    "chargebacks": [
      {
        "id": "string",
        "paymentId": "string",
        "amount": number,
        "currency": "string",
        "status": "string",
        "reason": "string",
        "createdAt": "ISO8601"
      }
    ],
    "pagination": {
      "page": number,
      "limit": number,
      "total": number,
      "totalPages": number
    }
  }
}
```

7.2 Get Chargeback Details

GET `/api/chargebacks/:chargebackId`

Headers: `Authorization: Bearer <token>`

7.3 Submit Chargeback Evidence

POST `/api/chargebacks/:chargebackId/evidence`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "documents": ["string"],
  "description": "string",
  "additionalInfo": "string | null"
}
```

8. Orders Endpoints

8.1 List Orders

GET `/api/orders`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `status` (optional): "pending" | "paid" | "shipped" | "cancelled"
- `customerId` (optional): string

Response (200):

```
{
  "success": true,
  "data": {
    "orders": [
      {
        "id": "string",
        "customer": {
          "id": "string",
          "name": "string",
          "email": "string"
        },
        "amount": number,
        "currency": "string",
        "items": [
          {
            "name": "string",
            "quantity": number,
            "price": number
          }
        ],
        "status": "string",
        "createdAt": "ISO8601"
      }
    ],
    "pagination": {
      "page": number,
      "limit": number,
      "total": number,
      "totalPages": number
    },
    "summary": {
      "totalOrders": number,
      "paid": number,
      "pending": number,
      "totalRevenue": number
    }
  }
}
```

8.2 Get Order Details

GET `/api/orders/:orderId`

Headers: `Authorization: Bearer <token>`

---

9. Balance & Transactions Endpoints

9.1 Get Balance

GET `/api/balance`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `currency` (optional): string (default: "XAF")

Response (200):

```
{
  "success": true,
  "data": {
    "currency": "string",
    "available": number,
    "pending": number,
    "reserved": number,
    "lastUpdated": "ISO8601"
  }
}
```

9.2 List Transactions

GET `/api/transactions`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `type` (optional): "payment" | "refund" | "payout" | "fee" | "chargeback"

- `startDate` (optional): YYYY-MM-DD
- `endDate` (optional): YYYY-MM-DD

9.3 Request Payout

POST `/api/payouts`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "amount": number,
  "currency": "string",
  "description": "string | null"
}
```

Response (201):

```
{
  "success": true,
  "data": {
    "payoutId": "string",
    "amount": number,
    "currency": "string",
    "status": "pending",
    "estimatedArrival": "ISO8601",
    "createdAt": "ISO8601"
  }
}
```

9.4 List Payouts

GET `/api/payouts`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `status` (optional): "pending" | "processing" | "completed" | "failed"

9.5 Get Payout Details

GET `/api/payouts/:payoutId`

Headers: `Authorization: Bearer <token>`

---

10. Statistics & Analytics Endpoints

10.1 Get Statistics

GET `/api/statistics`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `period` (required): "days" | "weeks" | "months" | "quarters" | "years"
- `value` (required): string (e.g., "November 2025", "Q1 2025")
- `comparePrevious` (optional): boolean (default: false)

Response (200):

```
{
  "success": true,
  "data": {
    "period": "string",
    "value": "string",
    "revenue": number,
    "transactions": number,
    "refunds": number,
    "chargebacks": number,
    "dataPoints": [
      {
        "label": "string",
        "revenue": number,
        "date": "ISO8601",
        "transactions": number
      }
    ],
    "previousPeriod": {
      "revenue": number,
      "transactions": number
    } | null,
    "totals": {
      "revenue": number,
      "transactions": number,
      "refunds": number,
      "chargebacks": number
    }
  }
}
```

10.2 Get Daily Statistics

GET `/api/statistics/daily`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `startDate` (required): YYYY-MM-DD
- `endDate` (required): YYYY-MM-DD

11. Reports Endpoints

11.1 Get Settlements Report

GET `/api/reports/settlements`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `startDate` (optional): YYYY-MM-DD
- `endDate` (optional): YYYY-MM-DD
- `status` (optional): "pending" | "paid"

Response (200):

```
{
  "success": true,
  "data": {
    "settlements": [
      {
        "id": "string",
        "date": "YYYY-MM-DD",
        "amount": number,
        "currency": "string",
        "status": "string"
      }
    ]
  }
}
```

11.2 Get Balance Report

GET `/api/reports/balance`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `startDate` (required): YYYY-MM-DD
- `endDate` (required): YYYY-MM-DD
- `currency` (optional): string

Response (200):

```
{
  "success": true,
  "data": {
    "startDate": "YYYY-MM-DD",
    "endDate": "YYYY-MM-DD",
    "openingBalance": number,
    "closingBalance": number,
    "categories": [
      {
        "category": "string",
        "pending": number,
        "available": number
      }
    ]
  }
}
```

11.3 Export Report

POST `/api/reports/export`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "reportType": "settlements" | "balance" | "invoices" | "payments",
  "format": "pdf" | "csv" | "xlsx",
  "startDate": "YYYY-MM-DD",
  "endDate": "YYYY-MM-DD",
  "filters": {}
}
```

---

12. Invoicing Endpoints

12.1 Create Invoice

POST `/api/invoices`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "type": "one-off" | "recurring",
  "profileId": "string",
  "customerId": "string",
  "paymentTerm": "string",
  "memo": "string | null",
  "items": [
    {
      "productId": "string | null",
      "name": "string",
      "quantity": number,
      "price": number,
      "vatRate": number
    }
  ],
  "discount": {
    "type": "percentage" | "fixed",
    "value": number
  } | null,
  "vatDisplay": "including" | "excluding"
}
```

Response (201):

```
{
  "success": true,
  "data": {
    "invoiceId": "string",
    "invoiceNumber": "string",
    "amount": number,
    "currency": "string",
    "status": "draft" | "sent" | "paid" | "overdue",
    "dueDate": "ISO8601",
    "createdAt": "ISO8601"
  }
}
```

12.2 List Invoices

GET `/api/invoices`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `status` (optional): "draft" | "sent" | "paid" | "overdue"
- `customerId` (optional): string
- `startDate` (optional): YYYY-MM-DD
- `endDate` (optional): YYYY-MM-DD
- `search` (optional): string

Response (200):

```
{
  "success": true,
  "data": {
    "invoices": [
      {
        "id": "string",
        "invoiceNumber": "string",
        "customer": {
          "id": "string",
          "name": "string",
          "email": "string"
        },
        "amount": number,
        "currency": "string",
        "status": "string",
        "date": "YYYY-MM-DD",
        "dueDate": "ISO8601"
      }
    ],
    "pagination": {
      "page": number,
      "limit": number,
      "total": number,
      "totalPages": number
    }
  }
}
```

12.3 Get Invoice Details

GET `/api/invoices/:invoiceId`

Headers: `Authorization: Bearer <token>`

12.4 Send Invoice

POST `/api/invoices/:invoiceId/send`

Headers: `Authorization: Bearer <token>`

12.5 Download Invoice

GET `/api/invoices/:invoiceId/download`

Headers: `Authorization: Bearer <token>`

Response: PDF file download

12.6 Create Recurring Invoice

POST `/api/invoices/recurring`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "customerId": "string",
  "frequency": "weekly" | "monthly" | "quarterly" | "yearly",
  "startDate": "YYYY-MM-DD",
  "endDate": "YYYY-MM-DD | null",
  "items": [],
  "memo": "string | null"
}
```

12.7 List Recurring Invoices

GET `/api/invoices/recurring`

Headers: `Authorization: Bearer <token>`

12.8 Create Credit Note

POST `/api/invoices/:invoiceId/credit-notes`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "amount": number,
  "reason": "string",
  "items": []
}
```

12.9 List Credit Notes

GET `/api/invoices/credit-notes`

Headers: `Authorization: Bearer <token>`

---

13. Customers Endpoints (Invoicing)

13.1 List Customers

GET `/api/customers`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `search` (optional): string

Response (200):

```
{
  "success": true,
  "data": {
    "customers": [
      {
        "id": "string",
        "name": "string",
        "email": "string",
        "phone": "string | null",
        "totalInvoices": number,
        "totalAmount": number,
        "currency": "string"
      }
    ],
    "pagination": {
      "page": number,
      "limit": number,
      "total": number,
      "totalPages": number
    }
  }
}
```

13.2 Create Customer

POST `/api/customers`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "name": "string",
  "email": "string",
  "phone": "string | null",
  "address": {},
  "taxId": "string | null"
}
```

13.3 Get Customer Details

GET `/api/customers/:customerId`

Headers: `Authorization: Bearer <token>`

---

14. Products Endpoints (Invoicing)

14.1 List Products

GET `/api/products`

Headers: `Authorization: Bearer <token>`

Response (200):

```
{
  "success": true,
  "data": {
    "products": [
      {
        "id": "string",
        "name": "string",
        "price": number,
        "currency": "string",
        "usedIn": number,
        "createdAt": "ISO8601"
      }
    ]
  }
}
```

14.2 Create Product

POST `/api/products`

Headers: `Authorization: Bearer <token>`

Request Body:

```json
{
  "name": "string",
  "description": "string | null",
  "price": number,
  "currency": "string",
  "vatRate": number
}
```

---

15. Organization Endpoints

15.1 Get Organization Details

GET `/api/organization`

Headers: `Authorization: Bearer <token>`

Response (200):

```json
{
  "success": true,
  "data": {
    "name": "string",
    "legalForm": "string",
    "registrationNumber": "string",
    "address": "string",
    "region": "string",
    "city": "string",
    "country": "string",
    "countryName": "string"
  }
}
```

Notes:

- Organization structure matches Cameroon business registration format
- `region` is used instead of postal code
- `legalForm` : Business legal structure (e.g., "Sole Proprietorship", "SARL", etc.)
- `registrationNumber` : Format `RC/DLA/2024/A/12345`

15.2 Update Organization

PUT `/api/organization`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "name": "string",
  "address": "string",
  "city": "string",
  "region": "string"
}
```

Notes:

- Only name, address, city, and region are editable from the frontend
- Other organization fields (legalForm, registrationNumber) are set during onboarding

---

16. Bank Account Endpoints

Note: Bank account linking has been removed from the onboarding flow. These endpoints may still be available for future use but are not currently required by the frontend.

16.1 List Bank Accounts

GET `/api/bank-accounts`

Headers: `Authorization: Bearer <token>`

Response (200):

```
{
  "success": true,
  "data": {
    "bankAccounts": [
      {
        "id": "string",
        "accountHolderName": "string",
        "bankName": "string",
        "accountNumber": "string (masked)",
        "accountType": "string",
        "currency": "string",
        "verificationStatus": "pending" | "verified" | "failed",
        "isDefault": boolean,
        "createdAt": "ISO8601"
      }
    ]
  }
}
```

16.2 Add Bank Account

POST `/api/bank-accounts`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "accountHolderName": "string",
  "accountNumber": "string",
  "routingNumber": "string",
  "bankName": "string",
  "accountType": "checking" | "savings",
  "country": "string",
  "currency": "string"
}
```

16.3 Delete Bank Account

DELETE `/api/bank-accounts/:bankAccountId`

Headers: `Authorization: Bearer <token>`

---

17. API Keys Endpoints

17.1 List API Keys

GET `/api/api-keys`

Headers: `Authorization: Bearer <token>`

Response (200):

```
{
  "success": true,
  "data": {
    "liveApiKey": "string (masked)",
    "testApiKey": "string (masked)",
    "createdAt": "ISO8601",
    "lastUsed": "ISO8601 | null"
  }
}
```

17.2 Generate API Key

POST `/api/api-keys/generate`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "type": "live" | "test"
}
```

17.3 Revoke API Key

DELETE `/api/api-keys/:keyId`

Headers: `Authorization: Bearer <token>`

17.4 Get API Logs

GET `/api/api-keys/logs`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `startDate` (optional): YYYY-MM-DD
- `endDate` (optional): YYYY-MM-DD

---

18. Notifications Endpoints

18.1 List Notifications

GET `/api/notifications`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `type` (optional): "primary" | "notification"
- `filter` (optional): "all" | "unread"
- `page` (optional): number (default: 1)
- `limit` (optional): number (default: 20)
- `search` (optional): string

Response (200):

```
{
  "success": true,
  "data": {
    "notifications": [
      {
        "id": "string",
        "type": "string",
        "category": "success" | "info" | "warning" | "error",
        "title": "string",
        "message": "string",
        "read": boolean,
        "actionUrl": "string | null",
        "createdAt": "ISO8601"
      }
    ],
    "unreadCount": {
      "primary": number,
      "notification": number
    },
    "pagination": {
      "page": number,
      "limit": number,
      "total": number,
      "totalPages": number
    }
  }
}
```

18.2 Mark Notification as Read

PUT `/api/notifications/:notificationId/read`

Headers: `Authorization: Bearer <token>`

18.3 Mark All Notifications as Read

PUT `/api/notifications/read-all`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "type": "primary" | "notification" | null
}
```

---

19. Webhooks Endpoints

19.1 List Webhooks

GET `/api/webhooks`

Headers: `Authorization: Bearer <token>`

Response (200):

```
{
  "success": true,
  "data": {
    "webhooks": [
      {
        "id": "string",
        "url": "string",
        "events": ["string"],
        "status": "active" | "inactive",
        "createdAt": "ISO8601"
      }
    ]
  }
}
```

19.2 Create Webhook

POST `/api/webhooks`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "url": "string",
  "events": ["payment.*", "refund.*", "chargeback.*"]
}
```

19.3 Update Webhook

PUT `/api/webhooks/:webhookId`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "url": "string",
  "events": ["string"],
  "status": "active" | "inactive"
}
```

19.4 Delete Webhook

DELETE `/api/webhooks/:webhookId`

Headers: `Authorization: Bearer <token>`

---

20. Payment Provider Endpoints

20.1 Make Payment

POST `/api/payments/make-payment`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "amount": number,
  "currency": "XAF" | "USD",
  "provider": "mtn-momo" | "orange-money",
  "phoneNumber": "string",
  "description": "string",
  "metadata": {}
}
```

Notes:

- **Cameroon Market:** Only MTN Mobile Money (`mtn-momo`) and Orange Money (`orange-money`) providers supported
- Phone number format: `+237 6 12 34 56 78` or `237612345678`
- Currency must be `XAF` or `USD`

Response (201):

```
{
  "success": true,
  "data": {
    "transactionId": "string",
    "status": "pending",
    "providerReference": "string"
  }
}
```

20.2 Get Payment Status

GET `/api/payments/status/:transactionId`

Headers: `Authorization: Bearer <token>`

20.3 Verify Account Holder

GET `/api/payments/verify-account-holder`

Headers: `Authorization: Bearer <token>`

Query Parameters:

- `phoneNumber` (required): string

Response (200):

```
{
  "success": true,
  "data": {
    "isActive": boolean,
    "name": "string",
    "phoneNumber": "string"
  }
}
```

---

21. Webhook Callback Endpoints (Public)

21.1 MTN Callback

PUT `/api/webhooks/mtn-callback`

Note: Public endpoint - verify callback signature

Request Body:

```
{
  "transactionId": "string",
  "status": "string",
  "amount": number,
  "reference": "string"
}
```

21.2 Mobile Money IPN

ALL `/api/payments/ipn/momo`

Note: Public endpoint - accepts all HTTP methods, verify IPN signature

---

22. Support & Feedback Endpoints

22.1 Submit Support Request

POST `/api/support`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "subject": "string",
  "category": "string",
  "message": "string",
  "priority": "low" | "medium" | "high"
}
```

22.2 Submit Feedback

POST `/api/feedback`

Headers: `Authorization: Bearer <token>`

Request Body:

```
{
  "type": "feature" | "bug" | "improvement" | "other",
  "rating": number,
  "title": "string",
  "message": "string",
  "contact": boolean,
  "email": "string | null"
}
```

Error Responses

All endpoints should return consistent error responses:

Error Response (4xx/5xx):

```
{
  "success": false,
  "error": {
    "code": "ERROR_CODE",
    "message": "Human-readable error message",
    "details": {} | null
  }
}
```

Common Error Codes

- `AUTH_REQUIRED` - Authentication required
- `AUTH_INVALID` - Invalid authentication token
- `AUTH_EXPIRED` - Authentication token expired
- `FORBIDDEN` - Insufficient permissions
- `NOT_FOUND` - Resource not found
- `VALIDATION_ERROR` - Request validation failed
- `EMAIL_EXISTS` - Email already registered
- `USERNAME_EXISTS` - Username already taken
- `INVALID_CREDENTIALS` - Invalid email or password
- `ACCOUNT_NOT_VERIFIED` - Email not verified
- `INSUFFICIENT_BALANCE` - Insufficient balance for operation
- `PAYMENT_FAILED` - Payment processing failed
- `RATE_LIMIT_EXCEEDED` - Too many requests
- `SERVER_ERROR` - Internal server error

HTTP Status Codes

- `200` - Success
- `201` - Created
- `400` - Bad Request
- `401` - Unauthorized
- `403` - Forbidden
- `404` - Not Found
- `422` - Validation Error
- `429` - Too Many Requests
- `500` - Internal Server Error

Pagination

All list endpoints support pagination via query parameters:

- `page` - Page number (default: 1)
- `limit` - Items per page (default: 20, max: 100)

Pagination Response:

```json
{
  "pagination": {
    "page": number,
    "limit": number,
    "total": number,
    "totalPages": number,
    "hasNext": boolean,
    "hasPrev": boolean
  }
}
```

Rate Limiting

API endpoints are rate-limited:

- **Authenticated requests**: 1000 requests per hour per user
- **Public endpoints**: 100 requests per hour per IP
- Rate limit headers:
    - `X-RateLimit-Limit` : Maximum requests allowed
    - `X-RateLimit-Remaining` : Remaining requests
    - `X-RateLimit-Reset` : Unix timestamp when limit resets

Webhooks

Webhooks are sent as HTTP POST requests to configured URLs. Webhook payloads include:

```json
{
  "event": "payment.completed",
  "timestamp": "ISO8601",
  "data": {},
  "signature": "string"
}
```

Always verify webhook signatures for security.

Notes for Backend Developers

General Implementation Guidelines

1. **Database**: **REQUIRED** - Use PostgreSQL as the primary database system

    - Version 12.0 or higher recommended
    - Use UUIDs for primary keys where possible
    - Implement proper indexing and foreign key constraints
    - Use database transactions for financial operations
    - Store monetary values as NUMERIC/DECIMAL types

2. **Authentication**: Implement JWT-based authentication with refresh tokens

3. **CORS**: Enable CORS for frontend origin

4. **Validation**: Validate all input data, especially Cameroon-specific formats

5. **Security**: Encrypt sensitive data, use HTTPS in production

6. **Logging**: Log all API requests and errors

7. **Testing**: Implement comprehensive test coverage including database tests

8. **Documentation**: Use OpenAPI/Swagger for interactive documentation

9. **Versioning**: Consider API versioning (e.g., `/api/v1/...`)

10. **Idempotency**: Make payment operations idempotent where possible

11. **Webhooks**: Implement retry logic for failed webhook deliveries

Cameroon Market-Specific Requirements

1. **Payment Providers**: Integrate with MTN Mobile Money and Orange Money APIs

    - MTN Mobile Money API integration required
    - Orange Money API integration required
    - Handle payment callbacks and webhooks from both providers

2. **Currency Handling**: Support XAF (Central African CFA Franc) as primary currency

    - Primary currency: XAF
    - Secondary currency: USD
    - Currency conversion rates if needed

3. **Region Validation**: Validate against the 10 Cameroon regions list

    - See "Cameroon-Specific Data Formats" section for valid regions
    - No postal code validation needed

4. **Phone Number Format**: Support both formats

    - `+237 6 12 34 56 78` (with spaces)
    - `237612345678` (without spaces)
    - Normalize to consistent format for storage

5. **Business Registration**: Validate registration number format

    - Pattern: `RC/[CITY]/[YEAR]/[LETTER]/[NUMBER]`
    - Example: `RC/DLA/2024/A/12345`

6. **VAT Number**: Validate format

    - Pattern: `M` followed by 9 digits

- Example: `M123456789`

7. **ID Document Upload**: Special handling for National ID (CNI)
   - Require both front and back images for CNI
   - Single image sufficient for Passport and Driver's License
8. **Address Structure**: Use region-based addressing
   - No postal code field
   - Use region/province selection
   - Store city and region separately
9. **Database Requirements**: PostgreSQL is mandatory
   - Use PostgreSQL 12.0 or higher
   - Implement proper schema with UUIDs, indexes, and foreign keys
   - Use transactions for all financial operations
   - Store monetary values as NUMERIC/DECIMAL types with proper precision

---

Summary of Changes for Cameroon Market

Key Market-Specific Updates:

1. **Payment Methods:**
   - Only MTN Mobile Money and Orange Money supported
   - No credit cards, bank transfers, or other methods
2. **Currency:**
   - Primary: XAF (Central African CFA Franc)
   - Secondary: USD (United States Dollar)
   - EUR and other currencies removed
3. **Onboarding Flow:**
   - Removed bank account linking step (now 4 steps instead of 5)
   - Steps: Stakeholder Info → Business Activity → Payment Methods → ID Document
4. **ID Document Upload:**
   - National ID (CNI) requires both front and back images
   - Passport and Driver's License require only front image
5. **Address Fields:**
   - Replaced `postalCode` with `region` (Cameroon doesn't use postal codes)
   - Valid regions: 10 Cameroon regions
6. **Business Registration:**
   - Registration number format: `RC/DLA/2024/A/12345`
   - VAT number format: `M123456789`
   - Cameroon-specific business types and industry sectors
7. **Account Settings:**

   - Removed Language, Timezone, and Test Mode settings
   - Only notification preferences available

8. **Organization Structure:**

   - Uses Region instead of postal code
   - Cameroon-specific legal forms and registration formats

9. **Database:**

   - **PostgreSQL is REQUIRED** for the backend implementation
   - Version 12.0 or higher recommended
   - Use proper indexing, transactions, and ACID compliance for financial operations

---

Summary

This API documentation covers **100+ endpoints** across 22 categories:

1. Authentication (8 endpoints)
2. User Profile (4 endpoints)
3. Account Settings (2 endpoints)
4. Onboarding (5 endpoints - Bank account linking removed)
5. Payments (6 endpoints)
6. Refunds (3 endpoints)
7. Chargebacks (3 endpoints)
8. Orders (2 endpoints)
9. Balance & Transactions (5 endpoints)
10. Statistics & Analytics (2 endpoints)
11. Reports (3 endpoints)
12. Invoicing (9 endpoints)
13. Customers (3 endpoints)
14. Products (2 endpoints)
15. Organization (2 endpoints)
16. Bank Accounts (3 endpoints)
17. API Keys (4 endpoints)
18. Notifications (3 endpoints)
19. Webhooks (4 endpoints)
20. Payment Providers (3 endpoints)
21. Webhook Callbacks (2 endpoints)
22. Support & Feedback (2 endpoints)

Total: 99+ API endpoints required for full functionality (Bank account linking endpoint removed from onboarding).

Changelog - Cameroon Market Updates

Removed Features:

- ❌ Bank account linking from onboarding flow (Step 5 removed)
- ❌ Language settings from account settings
- ❌ Timezone settings from account settings
- ❌ Test mode toggle from account settings
- ❌ EUR currency support
- ❌ Credit card payment methods

- ❌ Bank transfer payment methods
- ❌ Postal code field (replaced with Region/Province)

Added/Modified Features:

- ✅ Region/Province field instead of postal code
- ✅ Cameroon-specific business registration formats
- ✅ National ID (CNI) requires both front and back uploads
- ✅ MTN Mobile Money and Orange Money as primary payment methods
- ✅ XAF currency as primary currency
- ✅ Cameroon regions support (10 regions)
- ✅ Cameroon-specific legal forms and industry sectors

Updated Endpoint Counts:

- **Onboarding**: 5 endpoints (was 6, removed bank account)
- **Account Settings**: Notification preferences only (3 settings removed)
- **Payments**: Limited to MTN MoMo and Orange Money
- **Organization**: Uses region-based address structure