

API Documentation

Jabuntu e-Invoicing
NRS MBS System Integrator

Laflor Logic Limited — Version 1.0, 2026-10-12

Overview

The Jabuntu Public API lets a business read its invoices, drafts, customers, products, NRS submissions and financial reports, and create drafts and customers. Invoices are prepared, validated and signed through the NRS MBS inside the application; the API gives integrators read access to the results and the ability to prepare drafts. Nothing in the API submits directly to NRS — drafts are reviewed and submitted in the app.

Base URL

`https://erp.staging.jabuntu.com/api/public/v1` (staging)
<production base>/api/public/v1 (production)

Authentication

Every request carries a Bearer credential: `Authorization: Bearer <credential>`

- API key — created by the business owner under Developer; prefix `jbt_pk_...`
- OAuth access token — issued via the OAuth flow; prefix `jbt_at_...`

Each credential carries scopes, a per-minute rate limit, and optionally an IP allowlist.

Scopes

- `invoices:read`
- `drafts:read` / `drafts:write`
- `customers:read` / `customers:write`
- `products:read`
- `submissions:read`
- `reports:read`

Rate limiting & idempotency

Exceeding the per-minute limit returns HTTP 429 with `Retry-After`. POST requests require an `Idempotency-Key` header (8–80 chars: letters, digits, - or `_`); repeating it returns the first result, reusing it with a different body returns 422.

Endpoints

Endpoint	Purpose
<code>GET /me</code>	The business, the credential's scopes and whether the business
<code>GET /invoices</code>	NRS-cleared and other invoices (scope <code>invoices:read</code>)
<code>GET /invoices/{number}</code>	One invoice (scope <code>invoices:read</code>)
<code>GET /drafts</code>	Invoice drafts (scope <code>drafts:read</code>)
<code>POST /drafts</code>	Create an invoice draft (scope <code>drafts:write</code>); it is not submitted to
<code>GET /drafts/{id}</code>	One draft (scope <code>drafts:read</code>)
<code>GET /customers</code>	Customers (scope <code>customers:read</code>)
<code>POST /customers</code>	Create a customer (scope <code>customers:write</code>)
<code>GET /products</code>	Products with stock on hand (scope <code>products:read</code>)
<code>GET /submissions</code>	Where each invoice is on its way to NRS (scope
<code>GET /reports/trial-balance</code>	Trial balance (scope <code>reports:read</code>)
<code>GET /reports/profit-and-loss</code>	Profit and loss (scope <code>reports:read</code>)

Errors

- 401 — missing/invalid credential
- 403 — missing scope or IP not allowlisted
- 404 — not found for this business
- 422 — validation failed or idempotency key reused with a different body
- 429 — rate limit exceeded (honour Retry-After)

Example

```
curl -H "Authorization: Bearer jbt_pk_..." \
https://erp.staging.jabuntu.com/api/public/v1/invoices
```

E-invoicing (NRS MBS) flow

Invoices are prepared, validated and signed against the NRS MBS inside the application; signing (IRN and QR) is performed by NRS.

Step	Endpoint
Create / edit a draft	POST /api/erp/drafts ; PUT /api/erp/drafts/{id}
Local validation	POST /api/erp/drafts/{id}/validate-local
Prepare the MBS payload	POST /api/erp/drafts/{id}/prepare-mbs-local
Submit for validation & signing	POST /api/erp/drafts/{id}/submit
Check status (NRS confirmation, QR)	GET /api/erp/mbs/journey ; POST .../journey/{requestId}/check

Example invoice payload (MBS)

Mandatory fields: irn, issue_date/issue_time, invoice_type_code, invoice_kind (B2B/B2G/B2C), document_currency_code, accounting_supplier_party (tin), accounting_customer_party for B2B/B2G, invoice_line (hsn_code for goods, isic_code for services), tax_total, legal_monetary_total. Values are synthetic test data.

```
{
  "business_id": "11111111-1111-4111-8111-111111111111",
  "irn": "LOCALTEST001-TEST0001-20260907",
  "issue_date": "2026-09-07",
  "issue_time": "10:00:00",
  "due_date": "2026-09-21",
  "invoice_type_code": "380",
  "invoice_kind": "B2B",
  "document_currency_code": "NGN",
  "tax_currency_code": "NGN",
  "accounting_supplier_party": {
    "party_name": "Synthetic Supplier - NOT A REAL TAXPAYER",
    "tin": "00000000-0001",
    "email": "supplier@example.invalid",
    "postal_address": {
      "street_name": "1 Synthetic Test Street",
      "city_name": "Synthetic Test City",
      "postal_zone": "000000",
      "lga": "NG-LA-IKE",
      "state": "NG-LA",
      "country": "NG"
    }
  },
  "accounting_customer_party": {
    "party_name": "Synthetic Buyer - NOT A REAL TAXPAYER",
    "tin": "00000000-0002",
    "email": "buyer@example.invalid",
    "postal_address": {
      "street_name": "1 Synthetic Test Street",
      "city_name": "Synthetic Test City",
      "postal_zone": "000000",
      "lga": "NG-LA-IKE",
      "state": "NG-LA",
      "country": "NG"
    }
  }
}
```

```

},
"invoice_line": [
  {
    "hsn_code": "8471.30",
    "product_category": "Portable computers",
    "invoiced_quantity": 2.0,
    "line_extension_amount": 200000.0,
    "item": {
      "name": "Synthetic portable computer",
      "description": "Synthetic portable computer"
    },
    "price": {
      "price_amount": 100000.0,
      "base_quantity": 1,
      "price_unit": "EA"
    }
  },
  {
    "isic_code": "6201",
    "service_category": "Computer programming",
    "invoiced_quantity": 1.0,
    "line_extension_amount": 50000.0,
    "item": {
      "name": "Synthetic software service",
      "description": "Synthetic software service"
    },
    "price": {
      "price_amount": 50000.0,
      "base_quantity": 1,
      "price_unit": "EA"
    }
  }
],
"tax_total": [
  {
    "tax_amount": 18750.0,
    "tax_subtotal": [
      {
        "taxable_amount": 250000.0,
        "tax_amount": 18750.0,
        "tax_category": {
          "id": "STANDARD_VAT",
          "percent": 7.5
        }
      }
    ]
  }
],
"legal_monetary_total": {
  "line_extension_amount": 250000.0,
  "tax_exclusive_amount": 250000.0,
  "tax_inclusive_amount": 268750.0,
  "payable_amount": 268750.0
}
}

```

The irn shown is a placeholder: the authoritative IRN, cryptographic stamp (CSID) and QR code are issued by NRS on validation and signing.

A full OpenAPI 3 specification is published with the service and is the authoritative reference for request/response schemas.