

e-MCF API

Technical Specifications & Analysis

Democratic Republic of Congo
Standardized Invoice System

Document Date: November 17, 2025

Table of Contents

1. Overview & Introduction	3
2. Technical Specifications	4
3. Invoice Registration Workflow	5
4. Data Objects & Structures	7
5. Invoice Types	10
6. Information API Endpoints	11
7. Response Objects	12
8. Error Codes	13
9. Business Rules	14
10. Implementation Guide	15

1. Overview & Introduction

What is e-MCF?

The **e-MCF (electronic Module de Comptabilisation Fiscale)** is a cloud-based fiscal electronic device managed by the DGI (Direction Générale des Impôts) in the Democratic Republic of Congo. It is part of the country's standardized invoice system reform, designed to ensure compliance with tax regulations.

Purpose

Companies with their own **SFE (Système de Facturation d'Entreprise - Enterprise Invoicing System)** can connect to the e-MCF API to:

- 1 Submit invoice data for validation
- 2 Receive calculated tax amounts and totals
- 3 Obtain security codes (DEF/DGI codes) required for legal invoices
- 4 Generate QR codes for invoice verification

Key Concept

In this process, the SFE sends invoice data to the e-MCF, which validates the data, calculates taxes according to DRC regulations, and returns security elements necessary to produce a normalized (compliant) invoice. Without these security elements, the invoice is not legally valid for tax purposes.

2. Technical Specifications

API Base URLs

Environment	API Type	URL
Production	Invoicing	https://edef.dgirdc.cd/api/invoice
Production	Information	https://edef.dgirdc.cd/api/info
Test	Invoicing	https://developper.dgirdc.cd/edef/api/invoice
Test	Information	https://developper.dgirdc.cd/edef/api/info

Authentication

Method: JWT Bearer Token

Header: Authorization: Bearer <token>

Format: JSON for both requests and responses

Required Headers:

Header	Value	Description
Content-Type	application/json	Request contains JSON data
Accept	application/json	Response must be in JSON format
Authorization	Bearer <token>	JWT token from e-MCF management system

API Version

Current Version: 1.0

Target Users: Taxpayers with certified SFE systems approved to connect to e-MCF

3. Invoice Registration Workflow

The e-MCF API follows a three-step process for invoice registration. Each invoice must go through all steps to receive the security codes required for compliance.

Step 1: Check API Status (Optional)

Endpoint: GET /

Purpose: Verify API availability after periods of inactivity

When to use: Recommended after system idle time or before critical operations

Returns:

- API operational status (true/false)
- API version
- Taxpayer NIF (tax ID)
- e-MCF NIM (device number)
- Token validity timestamp
- Server date and time
- Number of pending invoice requests
- List of pending invoices with UIDs

Step 2: Submit Invoice Request

Endpoint: POST /

Purpose: Send invoice data and receive calculated totals

Important: This step does NOT finalize the invoice - it only validates and calculates

Key Data to Send:

Field	Description	Required
nif	Seller's tax identification number (13 digits)	Yes
type	Invoice type (FV, FA, EV, etc.)	Yes
items	Array of invoice line items with prices and tax groups	Yes
client	Client/buyer information	No
operator	POS operator details	Yes
payment	Payment type and amount details	No*
reference	Original invoice reference (for credit notes only)	No**

currency	Currency code and exchange rate (if not CDF)	No
----------	--	----

**Defaults to ESPECES (cash) if not provided*

***Required for credit note types (FA, EA)*

API Response Includes:

- **UID:** Unique transaction identifier (required for Step 3)
- **Tax calculations:** Amounts by tax group (A-I)
- **Totals:** Total amount including and excluding VAT
- **VAT totals:** Total VAT amount on the invoice

Step 3: Finalize Invoice

Endpoint: POST /{UID}/{ACTION}

Purpose: Confirm or cancel the invoice normalization

Critical: Must be done within 2 minutes of Step 2

Two Actions Available:

Action	Description	Result
CONFIRM	Validate and normalize the invoice	Receives DEF/DGI code and QR code. Invoice becomes legally valid.
CANCEL	Abort the normalization process	Invoice is marked as cancelled. No security codes issued.

Verification Required:

Before confirming (CONFIRM action), the SFE **MUST** verify that the totals calculated by e-MCF match its own calculations. This prevents errors and ensures data integrity. If totals don't match, use CANCEL and review the invoice data.

Success Response (CONFIRM):

- **dateTime:** Official invoice date and time
- **qrCode:** QR code content (66 characters) for customer verification
- **codeDEFDGI:** 29-character security code (legally required on invoice)
- **counters:** Invoice sequence counters
- **nim:** e-MCF device identification number

4. Data Objects & Structures

InvoiceRequestDataDto

Main invoice object sent in Step 2:

Field	Type	Req	Description
nif	string(13)	Yes	Seller's tax ID
rn	string	No	Internal invoice reference number
mode	string	Yes	Unit price mode: TTC (incl. tax) or HT (excl. tax)
isf	string(10)	Yes	SFE identifier code
type	enum	Yes	Invoice type (see section 5)
items	array	Yes	Array of invoice line items
client	object	No	Client/buyer details
operator	object	Yes	POS operator information
payment	array	No	Payment details (default: ESPECES)
reference	string(24)	No*	Original invoice ref (for credit notes)
curCode	string(8)	No	Currency code (blank = CDF)
curDate	datetime	No	Exchange rate date
curRate	decimal	No	Exchange rate value

*Required for FA and EA invoice types

ItemDto (Invoice Line Item)

Field	Type	Req	Description
code	string	No	Item/product code
type	string	Yes	Item type from /info/itemTypes
name	string	Yes	Item description/name
price	decimal	Yes	Unit price
quantity	decimal	Yes	Quantity sold
taxGroup	enum(A-I)	Yes	Tax classification group
taxSpecificValue	string	No	Specific tax per unit or % (e.g., '230' or '10%')
taxSpecificAmount	decimal	No	Total specific tax for quantity
originalPrice	decimal	No	Price before discount
priceModification	string	No	Discount/modification description

ClientDto (Client Information)

Field	Type	Required	Description
nif	string	No	Client's tax identification number

name	string	No	Client name or company name
contact	string	No	Contact phone number or email
address	string	No	Physical address
type	string	No	Client type code (from /info/clientTypes)
typeDesc	string	No	Client type description

PaymentDto (Payment Information)

Field	Type	Required	Description
name	enum	Yes	ESPECES, VIREMENT, CARTEBANCAIRE, MOBILEMONEY, CHEQUES, CREDIT, AUTRE
amount	decimal	Yes	Payment amount
currencyCode	string	No	Currency code (e.g., USD, EUR)
currencyRate	decimal	No	Exchange rate for this payment

5. Invoice Types

The e-MCF API supports six different invoice types, each with specific use cases:

Code	Full Name	Description	Use Case
FV	Facture de vente	Sales Invoice	Standard domestic sales
EV	Facture de vente à l'exportation	Export Sales Invoice	Sales to foreign buyers
FT	Facture d'acompte	Advance Payment Invoice	Deposits/partial payments
FA	Facture d'avoir	Credit Note	Returns, corrections, refunds
EA	Facture d'avoir à l'export	Export Credit Note	Export returns/corrections
ET	Facture d'acompte à l'export	Export Advance Payment	Export deposits

Important Note on Credit Notes (FA and EA):

When creating a credit note, you **must** include the **reference** field with the 24-character DEF/DGI code from the original invoice. This links the credit note to the original sale and ensures proper tax accounting. The API validates that the credit note amount does not exceed the original invoice amount.

6. Information API Endpoints

The Information API provides reference data and configuration information. These endpoints do not require invoice data and can be called anytime to retrieve current settings and valid codes.

Endpoint	Method	Description
/status	GET	API status and list of all e-MCF devices
/taxGroups	GET	Current tax rates for groups A through P
/invoiceTypes	GET	Available invoice type codes and descriptions
/paymentTypes	GET	Valid payment method codes
/clientTypes	GET	Client classification codes
/referenceTypes	GET	Reference types for credit notes
/itemTypes	GET	Item/product classification codes
/currencyRates	GET	Current exchange rates (primarily USD)

Usage Recommendation:

Call these endpoints during system initialization or periodically (e.g., daily) to cache reference data. This reduces API calls during invoice processing and ensures your system always has current tax rates and valid codes.

7. Response Objects

InvoiceResponseDataDto

Returned by POST / (Step 2 - Submit Invoice). Contains calculated tax amounts:

Field	Type	Description
uid	string(36)	Unique transaction ID - save this for Step 3
ta - ti	decimal	Tax rate percentages for groups A through I
taa - tai	decimal	Total amounts for each tax group A through I
haa - hai	decimal	Amounts excluding VAT for groups A through I
vaa - vai	decimal	VAT amounts for each group A through I
ts	decimal	Total specific tax amount
total	decimal	Grand total including all taxes
vtotal	decimal	Total VAT on entire invoice
errorCode	string	Error code if validation failed
errorDesc	string	Error description if validation failed

FinalizeInvoiceResponseDataDto

Returned by POST /{UID}/CONFIRM (Step 3 - Finalize). Contains security elements:

Field	Length	Description
dateTime	19	Official invoice timestamp (format: DD/MM/YYYY HH:MM:SS)
qrCode	66	QR code content for customer verification
codeDEFDGI	29	Security code - MUST be printed on invoice
counters	variable	Invoice sequence counters (e.g., '23/56 FV')
nim	10	e-MCF device identification number
errorCode	variable	Error code if finalization failed
errorDesc	variable	Error description if finalization failed

Critical: The codeDEFDGI and qrCode values are legally required on the printed invoice. Without these elements, the invoice is not compliant with DRC tax regulations.

8. Error Codes

The API returns specific error codes when validation fails. Handle these errors appropriately in your application:

Code	Description
1	Maximum number of pending invoices exceeded (limit: 10)
3	Invoice type is not valid
4	Original invoice reference is missing (required for FA/EA)
5	Original invoice reference must be exactly 24 characters
7	Payment type is not valid
8	Invoice must contain at least one item
9	Tax group at item level is not valid
10	Original invoice reference cannot be validated - retry later
11	Original invoice reference not found in system
12	Credit note amount exceeds original invoice amount
20	Invoice does not exist or is already finalized/cancelled
99	Error processing request - contact support

Error Handling Best Practices:

- **Error 1:** Implement queue management to stay under 10 pending invoices
- **Error 10:** Implement automatic retry logic with exponential backoff
- **Error 20:** Check for race conditions in your finalization logic
- **Error 99:** Log full request/response for debugging and contact DGI support

9. Key Business Rules

Understanding these rules is critical for successful integration:

Rule	Description	Impact
2-Minute Timeout	Pending invoices expire 2 minutes after submission	Must finalize quickly after validation
10 Invoice Limit	Maximum 10 pending invoices at once	Implement proper queue management
Total Verification	SFE must verify calculated totals before confirmation	Prevents tax calculation errors
Credit Note Reference	FA/EA types require 24-char reference to original	Must store DEF/DGI codes from sales
Tax Group Assignment	Each item must have valid tax group (A-I)	Get current groups from /taxGroups
Security Code Display	codeDEFDGI must appear on printed invoice	Legal requirement for compliance
JWT Token Validity	Tokens expire - check tokenValid field	Refresh token before expiration
Sequential Processing	Must complete Steps 1→2→3 in order	Cannot skip validation step

Currency Handling:

- Default currency is CDF (Congolese Franc)
- For foreign currency invoices, include curCode, curDate, and curRate
- Exchange rates can be retrieved from /currencyRates endpoint
- Multiple currencies can be used in payment array

10. Implementation Guide

Integration Workflow

Follow this sequence for each invoice:

Step	Action	API Call	Notes
1	Check API Status	GET /	Optional but recommended
2	Build Invoice Data	N/A	Prepare InvoiceRequestDataDto
3	Submit Invoice	POST /	Receive UID and totals
4	Verify Totals	N/A	Compare with your calculations
5a	Confirm (if OK)	POST /{UID}/CONFIRM	Get security codes
5b	Cancel (if error)	POST /{UID}/CANCEL	Abort normalization
6	Print Invoice	N/A	Include DEF/DGI code and QR
7	Store Record	N/A	Save UID and codes for audit

Code Example (Pseudo-code):

```
// Step 1: Check API Status (optional) response =
httpGet("https://edef.dgirdc.cd/api/invoice/", headers) if (!response.status) {
log("API unavailable") return } // Step 2: Submit Invoice invoice = buildInvoiceData()
// Create InvoiceRequestDataDto response =
httpPost("https://edef.dgirdc.cd/api/invoice/", invoice, headers) uid = response.uid
calculatedTotal = response.total calculatedVAT = response.vtotal // Step 3: Verify
totals if (calculatedTotal != myTotal || calculatedVAT != myVAT) { // Cancel if mismatch
httpPost("https://edef.dgirdc.cd/api/invoice/" + uid + "/CANCEL", null, headers)
log("Total mismatch - cancelled") return } // Step 4: Confirm invoice finalizeData = {
total: calculatedTotal, vtotal: calculatedVAT } response =
httpPost("https://edef.dgirdc.cd/api/invoice/" + uid + "/CONFIRM", finalizeData,
headers) // Step 5: Print invoice with security codes printInvoice(invoice,
response.codeDEFDGI, response.qrCode, response.dateTime) // Step 6: Store for audit
saveToDatabase(uid, response.codeDEFDGI, response.nim)
```

Testing Recommendations:

- Use the test environment URL for all initial development
- Test all invoice types (FV, FA, EV, EA, FT, ET)
- Test error scenarios (invalid tax groups, missing references, etc.)
- Verify timeout handling (delay between Step 2 and 3)
- Test with multiple payment types and currencies
- Validate QR code generation and scanning
- Perform load testing with concurrent requests
- Test token expiration and refresh logic

Production Deployment Checklist:

- ✓ Switch to production URLs
- ✓ Obtain production JWT token from DGI
- ✓ Implement proper error logging and monitoring
- ✓ Set up alerts for API failures
- ✓ Implement retry logic with exponential backoff
- ✓ Configure timeout handling (2-minute limit)
- ✓ Test invoice printing with real DEF/DGI codes
- ✓ Verify QR codes are scannable
- ✓ Train staff on error handling procedures
- ✓ Establish support channel with DGI

Additional Resources

Support Information:

Ministry of Finance

Direction Générale des Impôts (DGI)
Democratic Republic of Congo

Technical Support:

For integration issues, API questions, or technical support, contact the DGI technical team through official channels.

Developer Portal:

<https://developper.dgirdc.cd/edef>

Production Portal:

<https://edef.dgirdc.cd>

Document Information:

Document Generated: November 17, 2025 at 18:24

API Version: 1.0

Source Document: e-MCF_API_v1_0.pdf

Analysis Type: Technical Specification & Implementation Guide

This document is an English analysis and interpretation of the official e-MCF API technical specifications. For legal compliance, always refer to the official French documentation provided by the DGI.