



# API Documentation

Oman e-invoicing integration guide for ERP, billing and POS systems

Sample API documentation as per PINT-OM version v1.0.1

| Document       | Details                                                                      |
|----------------|------------------------------------------------------------------------------|
| Title          | Faturathi API Documentation                                                  |
| Version        | V1.0.1                                                                       |
| Date           | 24 September 2026                                                            |
| Status         | Sample / draft for UAT — subject to OTA confirmation                         |
| Specification  | PINT-OM v1.0.1 (Peppol International Invoice, Oman) · UBL 2.1 · Peppol AS4   |
| Audience       | Customers, ERP integrators and implementation partners                       |
| Classification | Confidential — Faturathi and its partners                                    |
| Prepared by    | NETBUE — International Intelligence Solutions LLC, Muscat, Sultanate of Oman |
| Contact        | faturathi@netbue.com · www.faturathi.com                                     |

Powered by NETBUE — Beyond Technology

## Confidentiality and intellectual property

This document and the information it contains are the intellectual property of International Intelligence Solutions LLC, trading as NETBUE (“NETBUE”), and are provided in confidence to Faturathi customers, ERP integrators and partners for the sole purpose of integrating with the Faturathi e-invoicing platform.

By receiving this document you agree that you will:

- Use it only to build, test and operate an integration with Faturathi.
- Not copy, publish, distribute or disclose it, in whole or in part, to any third party without the prior written consent of NETBUE.
- Keep API keys, tokens, endpoints and test data confidential and store them securely.
- Return or destroy the document on request, or when the integration work ends.

Faturathi, the Faturathi logo and NETBUE are trademarks of International Intelligence Solutions LLC. Peppol is a registered trademark of OpenPeppol AISBL. All other names are the property of their respective owners.

**Sample documentation.** This is sample API documentation prepared in line with PINT-OM version v1.0.1. Field rules, code lists and validation behaviour are governed by the Oman Tax Authority (OTA) and OpenPeppol and may change. Where this document and the official PINT-OM specification differ, the specification applies. Items marked “to be confirmed” will be finalised before production release.

### Document control

| Version | Date        | Author                  | Change                                                                                                           |
|---------|-------------|-------------------------|------------------------------------------------------------------------------------------------------------------|
| V1.0.1  | 24 Sep 2026 | NETBUE Integration Team | First release aligned to PINT-OM v1.0.1. Six document profiles, UAT endpoint, API-key and bearer-token security. |

## Contents

1. Introduction
2. How Faturathi works
3. Environments and endpoint
4. Security: API key and bearer token
5. Request structure and naming
6. Document types
7. API examples (JSON and cURL)
8. Responses and invoice status
9. Errors and troubleshooting
10. Field reference
11. Validation rules to watch
12. Duplicates, retries and limits
13. Peppol technical reference
14. Testing checklist and go-live
15. Support and glossary
- Appendix A. Internal review notes (remove before distribution)

# 1. Introduction

Faturathi is the Oman e-invoicing platform of NETBUE. It connects your ERP, billing or POS system to the Oman Tax Authority's e-invoicing framework through the Peppol network. You send invoice data as JSON to one endpoint. Faturathi turns it into a PINT-OM electronic invoice (UBL 2.1 XML), validates it, delivers it to your buyer over Peppol and reports the tax data to OTA.

## 1.1 Purpose of this document

This guide gives customers, ERP integrators and partners everything needed to build the integration: the endpoint, the security headers, the request structure for all six document types, complete JSON and cURL examples, the field reference and the validation rules that most often cause rejections.

## 1.2 Who should read it

- ERP and integration developers (SAP ABAP, Oracle, Dynamics, Odoo, ERPNext, Tally TDL, custom systems).
- Finance and tax teams who own invoice data quality.
- Implementation partners who onboard customers to Faturathi.

## 1.3 What you build and what Faturathi handles

| You build                                         | Faturathi handles                                                 |
|---------------------------------------------------|-------------------------------------------------------------------|
| Map your ERP invoice fields to the Faturathi JSON | PINT-OM UBL 2.1 XML generation                                    |
| Send the JSON and store the returned invoice ID   | Schema and Schematron validation (shared Peppol and Oman rules)   |
| Fix data when you receive a validation error      | Buyer lookup in the Oman SMP and Peppol AS4 delivery              |
| Update your ERP when the status changes           | Message Level Status (MLS) handling and tax data reporting to OTA |
| Receive webhooks or poll the status endpoint      | Retries, archiving of XML and receipts, audit log                 |

**A PDF is not an e-invoice.** The legal invoice is the structured PINT-OM XML that Faturathi generates from your JSON. The PDF is a readable copy for people.

## 2. How Faturathi works

### 2.1 The Oman five-corner model

Oman e-invoicing follows the Peppol five-corner model. The seller (C1) sends through its access point (C2) to the buyer's access point (C3) and the buyer (C4). Tax data goes to the Oman Tax Authority (C5). Faturathi is your access point: it acts as C2 when you send and as C3 when you receive.

| Corner | Who                               | Role                                                                     |
|--------|-----------------------------------|--------------------------------------------------------------------------|
| C1     | Seller (you)                      | Creates the invoice in the ERP and sends it to Faturathi                 |
| C2     | Faturathi (seller's access point) | Validates, converts to PINT-OM XML, delivers over Peppol, reports to OTA |
| C3     | Buyer's access point              | Receives, validates, returns a Message Level Status, reports to OTA      |
| C4     | Buyer                             | Receives the invoice in its own system                                   |
| C5     | Oman Tax Authority                | Receives the tax data document from both access points                   |

### 2.2 Two jobs for every invoice

Every invoice has two jobs that finish separately: delivery to the buyer and reporting to OTA. Both happen after your API call returns. Your integration should therefore submit the invoice, store the returned ID, and then track the status through webhooks or the status endpoint.

### 2.3 End-to-end flow

1. Your ERP creates the invoice and calls POST invoices/ with the JSON body.
2. Faturathi checks the structure and the PINT-OM rules. A clean document is accepted and queued; a faulty one is returned with a list of problems and nothing is sent.
3. Faturathi builds the UBL 2.1 XML, looks up the buyer in the Oman SMP and delivers over Peppol AS4.
4. The buyer's access point returns a Message Level Status (accepted or rejected).
5. Tax data is reported to OTA. When both jobs are done the invoice is completed.
6. Your ERP receives the final status and stores the invoice ID, UUID and XML for its records.

### 3. Environments and endpoint

Use UAT for development and user acceptance testing. UAT documents are never reported to OTA production. API keys and tokens for one environment do not work in the other.

| Environment | Endpoint                                                                                                    | Use for                                                                            |
|-------------|-------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| UAT         | <a href="https://uat-api.faturathi.com/api/v1/invoices/">https://uat-api.faturathi.com/api/v1/invoices/</a> | Development, UAT, Peppol test network                                              |
| Production  | <a href="https://api.faturathi.com/api/v1/invoices/">https://api.faturathi.com/api/v1/invoices/</a>         | Live documents reported to OTA (available at go-live, subject to OTA confirmation) |

| Method | Path                               | Purpose                                                            | Status          |
|--------|------------------------------------|--------------------------------------------------------------------|-----------------|
| POST   | <a href="#">invoices/</a>          | Submit an invoice, credit note, debit note or self-billed document | Available       |
| POST   | <a href="#">invoices/validate/</a> | Validate only — nothing is sent to Peppol or OTA                   | Available       |
| GET    | <a href="#">invoices/{id}/</a>     | Current status of one document                                     | To be confirmed |
| GET    | <a href="#">invoices/</a>          | List and filter documents                                          | To be confirmed |
| GET    | <a href="#">invoices/{id}/xml/</a> | Download the PINT-OM XML                                           | To be confirmed |
| GET    | <a href="#">invoices/{id}/pdf/</a> | Download the readable PDF                                          | To be confirmed |

All paths are relative to the base URL and end with a slash. Example: <https://uat-api.faturathi.com/api/v1/invoices/validate/>

## 4. Security: API key and bearer token

Credentials are created in the Faturathi portal on the REST API keys page. Each key belongs to one company (one VATIN). A group with several companies needs one key per company. Give each ERP, POS or script its own key so that one can be switched off without stopping the others.

Every key has two parts, and both are sent on every request:

- API key, sent in the X-API-KEY header. It identifies your key and company.
- Bearer token, sent in the Authorization header. The portal shows the token only once, when you generate it. Store it in your server's secret store immediately. If you lose it, generate a new one.

### 4.1 Request headers

| Header        | Value                                                            | Need     |
|---------------|------------------------------------------------------------------|----------|
| X-API-KEY     | API key from the portal                                          | Required |
| Authorization | Bearer followed by your token, for example Bearer <ACCESS_TOKEN> | Required |
| Content-Type  | application/json                                                 | Required |
| Accept        | application/json                                                 | Optional |

#### Headers on every request

```
Accept: application/json
Content-Type: application/json
X-API-KEY: <API_KEY>
Authorization: Bearer <ACCESS_TOKEN>
```

### 4.2 Transport and data protection

- All traffic uses HTTPS with TLS 1.2 or later. Plain HTTP is refused.
- Invoice data is processed and stored on infrastructure hosted in Oman.
- Every API call, status change and user action is written to an audit log.
- Rotate keys regularly and immediately if one is exposed. Delete unused keys in the portal.

**Keep credentials on the server.** Never place the API key or token in browser code, POS terminals, spreadsheets, emails or support tickets. Call Faturathi from a server you control.

## 5. Request structure and naming

The request body is a JSON object with seven groups. Field names are lower snake case and keep the PINT-OM business term identifier (IBT) or Oman term identifier (BTOM) as a prefix, followed by a readable name. This makes every field traceable to the specification and to the UBL 2.1 element it maps to.

| Group           | Contains                                                                               |
|-----------------|----------------------------------------------------------------------------------------|
| header          | Document identity: number, UUID, dates, type code, specification, currency, references |
| seller_details  | Seller legal name, identifiers, VATIN, Peppol address, postal address, contact         |
| buyer_details   | Buyer legal name, identifiers, VATIN, Peppol address, postal address                   |
| payment_details | Payment means, account (IBAN) and terms                                                |
| totals          | Document totals: net, VAT, gross, paid and amount due                                  |
| vat_breakdown   | One row per VAT category and rate                                                      |
| lines           | One row per invoice line: quantity, unit, price, VAT, classification                   |

### 5.1 Naming convention

Example: `ibt_001_invoice_number` is PINT-OM business term IBT-001 (Invoice number). `btom_001_invoice_transaction_type` is the Oman-specific term BTOM-001 (Transaction type).

### 5.2 Amounts and quantities are strings

Send every monetary value and quantity as a string with the decimals from your ERP, for example "525.000". Omani rial uses three decimals (baisa). Strings keep this precision exactly; JSON numbers can be rounded by some libraries. Faturathi checks that line amounts, the VAT breakdown and the totals add up.

### 5.3 Dates and times

Dates use YYYY-MM-DD. Times use HH:MM:SS in Oman local time.

## 6. Document types

Choose the document type that matches the business event. Corrections are made with a credit note or debit note that refers to the original invoice. An invoice is never edited after it has been sent.

| Type code | Document                   | UBL document | Specification identifier           | When to use                                                              |
|-----------|----------------------------|--------------|------------------------------------|--------------------------------------------------------------------------|
| 380       | B2B standard tax invoice   | Invoice      | urn:peppol:pint:billing-1@om-1     | Sale to a VAT-registered business or government buyer                    |
| 380       | B2C simplified tax invoice | Invoice      | urn:peppol:pint:billing-1@om-1     | Sale to a consumer; marked through the transaction type                  |
| 381       | Credit note                | CreditNote   | urn:peppol:pint:billing-1@om-1     | Reduce or cancel an invoice: returns, discounts, errors                  |
| 383       | Debit note                 | Invoice      | urn:peppol:pint:billing-1@om-1     | Increase the amount of an invoice already sent                           |
| 389       | Self-billed invoice        | Invoice      | urn:peppol:pint:selfbilling-1@om-1 | Buyer issues the invoice for the supplier under a self-billing agreement |
| 261       | Self-billed credit note    | CreditNote   | urn:peppol:pint:selfbilling-1@om-1 | Correct a self-billed invoice                                            |

The business process is `urn:peppol:bis:billing` for the billing documents and `urn:peppol:bis:selfbilling` for the self-billed documents.

**Self-billed documents.** You are the buyer and you issue the document, but the roles do not swap in the JSON: `seller_details` still holds your supplier and `buyer_details` holds your company.

## 7. API examples (JSON and cURL)

Each example is a complete request body. Copy one, change the invoice number and UUID, and send it. The same body is used for both endpoints: send it to `invoices/validate/` to check it, and to `invoices/` to submit it.

### 7.1 Sending with cURL

The example below sends the B2B standard tax invoice to the validate endpoint. Replace `<API_KEY>` and `<ACCESS_TOKEN>` with your credentials.

```
cURL · POST invoices/validate/
curl --request POST "https://uat-api.faturathi.com/api/v1/invoices/validate/" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "X-API-KEY: <API_KEY>" \
  --header "Authorization: Bearer <ACCESS_TOKEN>" \
  --data @380-b2b-standard-tax-invoice.json
```

To submit for real, change the path to `invoices/`. To send another document type, point `--data` at that sample file.

### 7.2 Using the Postman collection

1. Import `Faturathi-API-v1.0.1.postman_collection.json` and `Faturathi-UAT.postman_environment.json`.
2. Open the environment and paste your `api_key` and `bearer_token`. Never share the environment file once it contains real values.
3. Run folder 1 (Validate) first, then folder 2 (Submit), then folder 3 (Track). Folder 4 holds negative tests.

The collection generates a new invoice number and UUID on every run, so you can repeat tests without hitting the duplicate check. Credit and debit notes automatically refer to the last invoice you submitted.

### 7.3 Standard tax invoice — type code 380

A normal B2B sale. `buyer_details.electronic_address` tells Faturathi where to deliver on Peppol.

## Request body · 380-b2b-standard-tax-invoice.json

```
{
  "header": {
    "btom_001_invoice_transaction_type": "10000000000000000000",
    "btom_002_invoice_uuid": "b41c72aa-9912-5f2b-8d34-77c1e09a55d0",
    "ibt_001_invoice_number": "INV-2026-0001",
    "ibt_002_invoice_issue_date": "2026-10-01",
    "ibt_168_invoice_issue_time": "10:30:00",
    "ibt_003_invoice_type_code": "380",
    "ibt_024_specification_identifier": "urn:peppol:pint:billing-1@om-1",
    "ibt_023_business_process_type": "urn:peppol:bis:billing",
    "ibt_005_invoice_currency_code": "OMR",
    "ibt_007_vat_point_date": "2026-10-01",
    "ibt_009_payment_due_date": "2026-10-31"
  },
  "seller_details": {
    "ibt_027_seller_name": "Al Noor Trading LLC",
    "ibt_029_seller_identifier": "112345678900003",
    "ibt_029_1_seller_identifier_scheme": "OTHID",
    "ibt_031_seller_vatin": "OM1100123456",
    "ibt_031_1_tax_scheme_code": "VAT",
    "ibt_034_seller_electronic_address": "OM1100123456",
    "ibt_034_1_scheme_identifier": "0248",
    "ibt_035_seller_address_line_1": "Way 2317, Building 192",
    "ibt_036_seller_address_line_2": "Ruwi",
    "ibt_162_seller_address_line_3": "Muscat Governorate",
    "ibt_037_seller_city": "Muscat",
    "ibt_038_seller_post_code": "112",
    "ibt_040_seller_country_code": "OM",
    "ibt_041_seller_contact_name": "Ahmed Al-Harthy",
    "ibt_042_seller_contact_telephone": "+96824000000",
    "ibt_043_seller_contact_email": "billing@example.om"
  },
  "buyer_details": {
    "ibt_044_buyer_name": "Muscat Retail SAOC",
    "ibt_046_buyer_identifier": "112345679000001",
    "ibt_046_1_buyer_identifier_scheme": "OTHID",
    "ibt_048_buyer_vatin": "OM1100654321",
    "ibt_048_1_tax_scheme_code": "VAT",
    "ibt_049_buyer_electronic_address": "OM1100654321",
    "ibt_049_1_scheme_identifier": "0248",
    "ibt_050_buyer_address_line_1": "Al Khuwair Commercial Area",
    "ibt_051_buyer_address_line_2": "Building 44",
    "ibt_163_buyer_address_line_3": "Bawshar",
    "ibt_052_buyer_city": "Muscat",
    "ibt_053_buyer_post_code": "133",
    "ibt_055_buyer_country_code": "OM"
  },
  "payment_details": {
    "ibt_081_payment_means_code": "30",
    "ibt_084_payment_account_identifier": "OM9300001234567890123456",
    "ibt_020_payment_terms_note": "Net 30 days"
  },
  "totals": {
    "ibt_106_sum_of_line_net_amount": "500.000",
    "ibt_107_sum_of_allowances": "0.000",
    "ibt_108_sum_of_charges": "0.000",
    "ibt_109_invoice_total_without_vat": "500.000",
  }
}
```

```
"ibt_110_invoice_total_vat_amount": "25.000",
"ibt_111_vat_amount_accounting_currency": "25.000",
"ibt_112_invoice_total_with_vat": "525.000",
"ibt_113_paid_amount": "0.000",
"ibt_114_rounding_amount": "0.000",
"ibt_115_amount_due_for_payment": "525.000"
},
"vat_breakdown": [
  {
    "ibt_116_vat_category_taxable_amount": "500.000",
    "ibt_117_vat_category_tax_amount": "25.000",
    "ibt_118_vat_category_code": "S",
    "ibt_118_1_tax_scheme_code": "VAT",
    "ibt_119_vat_category_rate": "5.00"
  }
],
"lines": [
  {
    "ibt_126_line_identifier": "1",
    "ibt_127_line_note": "Enterprise integration service",
    "ibt_129_invoiced_quantity": "1.000",
    "ibt_130_quantity_unit_code": "EA",
    "ibt_131_line_net_amount": "500.000",
    "ibt_146_item_net_price": "500.000",
    "ibt_151_item_vat_category_code": "S",
    "ibt_152_item_vat_rate": "5.00",
    "ibt_153_item_name": "PINT-OM Gateway Integration",
    "btom_016_line_vat_amount": "25.000",
    "btom_017_line_total_including_vat": "525.000",
    "btom_019_goods_or_services_identification": "S",
    "btom_033_industrial_classification_code": "620101",
    "btom_033_1_scheme_name": "CC"
  }
]
}
```

## Complete authenticated cURL · POST invoices/

```
curl --request POST "https://uat-api.faturathi.com/api/v1/invoices/" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "X-API-KEY: <API_KEY>" \
  --header "Authorization: Bearer <ACCESS_TOKEN>" \
  --data '{
"header": {
  "btom_001_invoice_transaction_type": "10000000000000000000",
  "btom_002_invoice_uuid": "b41c72aa-9912-5f2b-8d34-77c1e09a55d0",
  "ibt_001_invoice_number": "INV-2026-0001",
  "ibt_002_invoice_issue_date": "2026-10-01",
  "ibt_168_invoice_issue_time": "10:30:00",
  "ibt_003_invoice_type_code": "380",
  "ibt_024_specification_identfier": "urn:peppol:pint:billing-1@om-1",
  "ibt_023_business_process_type": "urn:peppol:bis:billing",
  "ibt_005_invoice_currency_code": "OMR",
  "ibt_007_vat_point_date": "2026-10-01",
  "ibt_009_payment_due_date": "2026-10-31"
},
"seller_details": {
  "ibt_027_seller_name": "Al Noor Trading LLC",
  "ibt_029_seller_identfier": "112345678900003",
  "ibt_029_1_seller_identfier_scheme": "OTHID",
  "ibt_031_seller_vatin": "OM1100123456",
  "ibt_031_1_tax_scheme_code": "VAT",
  "ibt_034_seller_electronic_address": "OM1100123456",
  "ibt_034_1_scheme_identfier": "0248",
  "ibt_035_seller_address_line_1": "Way 2317, Building 192",
  "ibt_036_seller_address_line_2": "Ruwi",
  "ibt_162_seller_address_line_3": "Muscat Governorate",
  "ibt_037_seller_city": "Muscat",
  "ibt_038_seller_post_code": "112",
  "ibt_040_seller_country_code": "OM",
  "ibt_041_seller_contact_name": "Ahmed Al-Harthy",
  "ibt_042_seller_contact_telephone": "+96824000000",
  "ibt_043_seller_contact_email": "billing@example.om"
},
"buyer_details": {
  "ibt_044_buyer_name": "Muscat Retail SAOC",
  "ibt_046_buyer_identfier": "112345679000001",
  "ibt_046_1_buyer_identfier_scheme": "OTHID",
  "ibt_048_buyer_vatin": "OM1100654321",
  "ibt_048_1_tax_scheme_code": "VAT",
  "ibt_049_buyer_electronic_address": "OM1100654321",
  "ibt_049_1_scheme_identfier": "0248",
  "ibt_050_buyer_address_line_1": "Al Khuwair Commercial Area",
  "ibt_051_buyer_address_line_2": "Building 44",
  "ibt_163_buyer_address_line_3": "Bawshar",
  "ibt_052_buyer_city": "Muscat",
  "ibt_053_buyer_post_code": "133",
  "ibt_055_buyer_country_code": "OM"
},
"payment_details": {
  "ibt_081_payment_means_code": "30",
  "ibt_084_payment_account_identfier": "OM9300001234567890123456",
  "ibt_020_payment_terms_note": "Net 30 days"
},
}
```

```
"totals": {
  "ibt_106_sum_of_line_net_amount": "500.000",
  "ibt_107_sum_of_allowances": "0.000",
  "ibt_108_sum_of_charges": "0.000",
  "ibt_109_invoice_total_without_vat": "500.000",
  "ibt_110_invoice_total_vat_amount": "25.000",
  "ibt_111_vat_amount_accounting_currency": "25.000",
  "ibt_112_invoice_total_with_vat": "525.000",
  "ibt_113_paid_amount": "0.000",
  "ibt_114_rounding_amount": "0.000",
  "ibt_115_amount_due_for_payment": "525.000"
},
"vat_breakdown": [
  {
    "ibt_116_vat_category_taxable_amount": "500.000",
    "ibt_117_vat_category_tax_amount": "25.000",
    "ibt_118_vat_category_code": "S",
    "ibt_118_1_tax_scheme_code": "VAT",
    "ibt_119_vat_category_rate": "5.00"
  }
],
"lines": [
  {
    "ibt_126_line_identifier": "1",
    "ibt_127_line_note": "Enterprise integration service",
    "ibt_129_invoiced_quantity": "1.000",
    "ibt_130_quantity_unit_code": "EA",
    "ibt_131_line_net_amount": "500.000",
    "ibt_146_item_net_price": "500.000",
    "ibt_151_item_vat_category_code": "S",
    "ibt_152_item_vat_rate": "5.00",
    "ibt_153_item_name": "PINT-OM Gateway Integration",
    "btom_016_line_vat_amount": "25.000",
    "btom_017_line_total_including_vat": "525.000",
    "btom_019_goods_or_services_identification": "S",
    "btom_033_industrial_classification_code": "620101",
    "btom_033_1_scheme_name": "CC"
  }
]
}'
```

## 7.4 Simplified tax invoice (B2C) — type code 380

A sale to a consumer. No buyer VATIN or Peppol address. The transaction type code for B2C must be confirmed before use.

## Request body · 380-b2c-simplified-tax-invoice.json

```
{
  "header": {
    "btom_001_invoice_transaction_type": "[B2C transaction type code]",
    "btom_002_invoice_uuid": "d2f8alc3-4b7e-5a90-9c11-3e5f7a8b9c01",
    "ibt_001_invoice_number": "POS-2026-000481",
    "ibt_002_invoice_issue_date": "2026-10-01",
    "ibt_168_invoice_issue_time": "18:42:10",
    "ibt_003_invoice_type_code": "380",
    "ibt_024_specification_identifier": "urn:peppol:pint:billing-1@om-1",
    "ibt_023_business_process_type": "urn:peppol:bis:billing",
    "ibt_005_invoice_currency_code": "OMR",
    "ibt_007_vat_point_date": "2026-10-01"
  },
  "seller_details": {
    "ibt_027_seller_name": "Al Noor Trading LLC",
    "ibt_029_seller_identifier": "112345678900003",
    "ibt_029_1_seller_identifier_scheme": "OTHID",
    "ibt_031_seller_vatin": "OM1100123456",
    "ibt_031_1_tax_scheme_code": "VAT",
    "ibt_034_seller_electronic_address": "OM1100123456",
    "ibt_034_1_scheme_identifier": "0248",
    "ibt_035_seller_address_line_1": "Way 2317, Building 192",
    "ibt_036_seller_address_line_2": "Ruwi",
    "ibt_162_seller_address_line_3": "Muscat Governorate",
    "ibt_037_seller_city": "Muscat",
    "ibt_038_seller_post_code": "112",
    "ibt_040_seller_country_code": "OM",
    "ibt_041_seller_contact_name": "Ahmed Al-Harthy",
    "ibt_042_seller_contact_telephone": "+96824000000",
    "ibt_043_seller_contact_email": "billing@example.om"
  },
  "buyer_details": {
    "ibt_044_buyer_name": "Walk-in customer",
    "ibt_055_buyer_country_code": "OM"
  },
  "payment_details": {
    "ibt_081_payment_means_code": "10"
  },
  "totals": {
    "ibt_106_sum_of_line_net_amount": "30.000",
    "ibt_107_sum_of_allowances": "0.000",
    "ibt_108_sum_of_charges": "0.000",
    "ibt_109_invoice_total_without_vat": "30.000",
    "ibt_110_invoice_total_vat_amount": "1.500",
    "ibt_111_vat_amount_accounting_currency": "1.500",
    "ibt_112_invoice_total_with_vat": "31.500",
    "ibt_113_paid_amount": "0.000",
    "ibt_114_rounding_amount": "0.000",
    "ibt_115_amount_due_for_payment": "31.500"
  },
  "vat_breakdown": [
    {
      "ibt_116_vat_category_taxable_amount": "30.000",
      "ibt_117_vat_category_tax_amount": "1.500",
      "ibt_118_vat_category_code": "S",
      "ibt_118_1_tax_scheme_code": "VAT",
      "ibt_119_vat_category_rate": "5.00"
    }
  ]
}
```

```
    }
  ],
  "lines": [
    {
      "ibt_126_line_identifier": "1",
      "ibt_129_invoiced_quantity": "2.000",
      "ibt_130_quantity_unit_code": "EA",
      "ibt_131_line_net_amount": "30.000",
      "ibt_146_item_net_price": "15.000",
      "ibt_151_item_vat_category_code": "S",
      "ibt_152_item_vat_rate": "5.00",
      "ibt_153_item_name": "Laptop setup service",
      "btom_016_line_vat_amount": "1.500",
      "btom_017_line_total_including_vat": "31.500",
      "btom_019_goods_or_services_identification": "S",
      "btom_033_industrial_classification_code": "620101",
      "btom_033_1_scheme_name": "CC"
    }
  ]
}
```

#### cURL · POST invoices/

```
curl --request POST "https://uat-api.faturathi.com/api/v1/invoices/" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "X-API-KEY: <API_KEY>" \
  --header "Authorization: Bearer <ACCESS_TOKEN>" \
  --data @380-b2c-simplified-tax-invoice.json
```

## 7.5 Credit note — type code 381

Reduces an invoice already sent. preceding\_invoice\_number points to the original invoice.

## Request body · 381-credit-note.json

```
{
  "header": {
    "btom_001_invoice_transaction_type": "10000000000000000000",
    "btom_002_invoice_uuid": "7c3e9b10-2a4d-5f6e-8b1a-9d0c2e4f6a82",
    "ibt_001_invoice_number": "CN-2026-0001",
    "ibt_002_invoice_issue_date": "2026-10-05",
    "ibt_168_invoice_issue_time": "10:30:00",
    "ibt_003_invoice_type_code": "381",
    "ibt_024_specification_identifier": "urn:peppol:pint:billing-1@om-1",
    "ibt_023_business_process_type": "urn:peppol:bis:billing",
    "ibt_005_invoice_currency_code": "OMR",
    "ibt_007_vat_point_date": "2026-10-05",
    "ibt_025_preceding_invoice_reference": "INV-2026-0001",
    "ibt_026_preceding_invoice_issue_date": "2026-10-01"
  },
  "seller_details": {
    "ibt_027_seller_name": "Al Noor Trading LLC",
    "ibt_029_seller_identifier": "112345678900003",
    "ibt_029_1_seller_identifier_scheme": "OTHID",
    "ibt_031_seller_vatin": "OM1100123456",
    "ibt_031_1_tax_scheme_code": "VAT",
    "ibt_034_seller_electronic_address": "OM1100123456",
    "ibt_034_1_scheme_identifier": "0248",
    "ibt_035_seller_address_line_1": "Way 2317, Building 192",
    "ibt_036_seller_address_line_2": "Ruwi",
    "ibt_162_seller_address_line_3": "Muscat Governorate",
    "ibt_037_seller_city": "Muscat",
    "ibt_038_seller_post_code": "112",
    "ibt_040_seller_country_code": "OM",
    "ibt_041_seller_contact_name": "Ahmed Al-Harthy",
    "ibt_042_seller_contact_telephone": "+96824000000",
    "ibt_043_seller_contact_email": "billing@example.om"
  },
  "buyer_details": {
    "ibt_044_buyer_name": "Muscat Retail SAOC",
    "ibt_046_buyer_identifier": "112345679000001",
    "ibt_046_1_buyer_identifier_scheme": "OTHID",
    "ibt_048_buyer_vatin": "OM1100654321",
    "ibt_048_1_tax_scheme_code": "VAT",
    "ibt_049_buyer_electronic_address": "OM1100654321",
    "ibt_049_1_scheme_identifier": "0248",
    "ibt_050_buyer_address_line_1": "Al Khuwair Commercial Area",
    "ibt_051_buyer_address_line_2": "Building 44",
    "ibt_163_buyer_address_line_3": "Bawshar",
    "ibt_052_buyer_city": "Muscat",
    "ibt_053_buyer_post_code": "133",
    "ibt_055_buyer_country_code": "OM"
  },
  "totals": {
    "ibt_106_sum_of_line_net_amount": "100.000",
    "ibt_107_sum_of_allowances": "0.000",
    "ibt_108_sum_of_charges": "0.000",
    "ibt_109_invoice_total_without_vat": "100.000",
    "ibt_110_invoice_total_vat_amount": "5.000",
    "ibt_111_vat_amount_accounting_currency": "5.000",
    "ibt_112_invoice_total_with_vat": "105.000",
    "ibt_113_paid_amount": "0.000",
  }
}
```

```

    "ibt_114_rounding_amount": "0.000",
    "ibt_115_amount_due_for_payment": "105.000"
  },
  "vat_breakdown": [
    {
      "ibt_116_vat_category_taxable_amount": "100.000",
      "ibt_117_vat_category_tax_amount": "5.000",
      "ibt_118_vat_category_code": "S",
      "ibt_118_1_tax_scheme_code": "VAT",
      "ibt_119_vat_category_rate": "5.00"
    }
  ],
  "lines": [
    {
      "ibt_126_line_identifier": "1",
      "ibt_127_line_note": "Partial credit: scope reduced",
      "ibt_129_invoiced_quantity": "1.000",
      "ibt_130_quantity_unit_code": "EA",
      "ibt_131_line_net_amount": "100.000",
      "ibt_146_item_net_price": "100.000",
      "ibt_151_item_vat_category_code": "S",
      "ibt_152_item_vat_rate": "5.00",
      "ibt_153_item_name": "PINT-OM Gateway Integration",
      "btom_016_line_vat_amount": "5.000",
      "btom_017_line_total_including_vat": "105.000",
      "btom_019_goods_or_services_identification": "S",
      "btom_033_industrial_classification_code": "620101",
      "btom_033_1_scheme_name": "CC"
    }
  ]
}

```

#### cURL · POST invoices/

```

curl --request POST "https://uat-api.faturathi.com/api/v1/invoices/" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "X-API-KEY: <API_KEY>" \
  --header "Authorization: Bearer <ACCESS_TOKEN>" \
  --data @381-credit-note.json

```

## 7.6 Debit note — type code 383

Adds to an invoice already sent, for example extra hours. `preceding_invoice_number` points to the original invoice.

## Request body · 383-debit-note.json

```
{
  "header": {
    "btom_001_invoice_transaction_type": "10000000000000000000",
    "btom_002_invoice_uuid": "5e1a7c92-8b3f-5d04-a6e2-1f9b3c5d7e04",
    "ibt_001_invoice_number": "DN-2026-0001",
    "ibt_002_invoice_issue_date": "2026-10-06",
    "ibt_168_invoice_issue_time": "10:30:00",
    "ibt_003_invoice_type_code": "383",
    "ibt_024_specification_identifier": "urn:peppol:pint:billing-1@om-1",
    "ibt_023_business_process_type": "urn:peppol:bis:billing",
    "ibt_005_invoice_currency_code": "OMR",
    "ibt_007_vat_point_date": "2026-10-06",
    "ibt_009_payment_due_date": "2026-11-05",
    "ibt_025_preceding_invoice_reference": "INV-2026-0001",
    "ibt_026_preceding_invoice_issue_date": "2026-10-01"
  },
  "seller_details": {
    "ibt_027_seller_name": "Al Noor Trading LLC",
    "ibt_029_seller_identifier": "112345678900003",
    "ibt_029_1_seller_identifier_scheme": "OTHID",
    "ibt_031_seller_vatin": "OM1100123456",
    "ibt_031_1_tax_scheme_code": "VAT",
    "ibt_034_seller_electronic_address": "OM1100123456",
    "ibt_034_1_scheme_identifier": "0248",
    "ibt_035_seller_address_line_1": "Way 2317, Building 192",
    "ibt_036_seller_address_line_2": "Ruwi",
    "ibt_162_seller_address_line_3": "Muscat Governorate",
    "ibt_037_seller_city": "Muscat",
    "ibt_038_seller_post_code": "112",
    "ibt_040_seller_country_code": "OM",
    "ibt_041_seller_contact_name": "Ahmed Al-Harthy",
    "ibt_042_seller_contact_telephone": "+96824000000",
    "ibt_043_seller_contact_email": "billing@example.om"
  },
  "buyer_details": {
    "ibt_044_buyer_name": "Muscat Retail SAOC",
    "ibt_046_buyer_identifier": "112345679000001",
    "ibt_046_1_buyer_identifier_scheme": "OTHID",
    "ibt_048_buyer_vatin": "OM1100654321",
    "ibt_048_1_tax_scheme_code": "VAT",
    "ibt_049_buyer_electronic_address": "OM1100654321",
    "ibt_049_1_scheme_identifier": "0248",
    "ibt_050_buyer_address_line_1": "Al Khuwair Commercial Area",
    "ibt_051_buyer_address_line_2": "Building 44",
    "ibt_163_buyer_address_line_3": "Bawshar",
    "ibt_052_buyer_city": "Muscat",
    "ibt_053_buyer_post_code": "133",
    "ibt_055_buyer_country_code": "OM"
  },
  "payment_details": {
    "ibt_081_payment_means_code": "30",
    "ibt_084_payment_account_identifier": "OM9300001234567890123456",
    "ibt_020_payment_terms_note": "Net 30 days"
  },
  "totals": {
    "ibt_106_sum_of_line_net_amount": "30.000",
    "ibt_107_sum_of_allowances": "0.000",
```

```

    "ibt_108_sum_of_charges": "0.000",
    "ibt_109_invoice_total_without_vat": "30.000",
    "ibt_110_invoice_total_vat_amount": "1.500",
    "ibt_111_vat_amount_accounting_currency": "1.500",
    "ibt_112_invoice_total_with_vat": "31.500",
    "ibt_113_paid_amount": "0.000",
    "ibt_114_rounding_amount": "0.000",
    "ibt_115_amount_due_for_payment": "31.500"
  },
  "vat_breakdown": [
    {
      "ibt_116_vat_category_taxable_amount": "30.000",
      "ibt_117_vat_category_tax_amount": "1.500",
      "ibt_118_vat_category_code": "S",
      "ibt_118_1_tax_scheme_code": "VAT",
      "ibt_119_vat_category_rate": "5.00"
    }
  ],
  "lines": [
    {
      "ibt_126_line_identifier": "1",
      "ibt_129_invoiced_quantity": "2.000",
      "ibt_130_quantity_unit_code": "HUR",
      "ibt_131_line_net_amount": "30.000",
      "ibt_146_item_net_price": "15.000",
      "ibt_151_item_vat_category_code": "S",
      "ibt_152_item_vat_rate": "5.00",
      "ibt_153_item_name": "Additional integration hours",
      "btom_016_line_vat_amount": "1.500",
      "btom_017_line_total_including_vat": "31.500",
      "btom_019_goods_or_services_identification": "S",
      "btom_033_industrial_classification_code": "620101",
      "btom_033_1_scheme_name": "CC"
    }
  ]
}

```

#### cURL · POST invoices/

```

curl --request POST "https://uat-api.faturathi.com/api/v1/invoices/" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "X-API-KEY: <API_KEY>" \
  --header "Authorization: Bearer <ACCESS_TOKEN>" \
  --data @383-debit-note.json

```

## 7.7 Self-billed invoice — type code 389

You are the buyer and issue the invoice for your supplier. seller\_details still holds the supplier; buyer\_details is you.

## Request body · 389-self-billed-invoice.json

```
{
  "header": {
    "btom_001_invoice_transaction_type": "10000000000000000000",
    "btom_002_invoice_uuid": "9a4b6c8d-0e1f-5a2b-9c3d-4e5f6a7b8c93",
    "ibt_001_invoice_number": "SB-2026-0001",
    "ibt_002_invoice_issue_date": "2026-10-01",
    "ibt_168_invoice_issue_time": "10:30:00",
    "ibt_003_invoice_type_code": "389",
    "ibt_024_specification_identifier": "urn:peppol:pint:selfbilling-1@om-1",
    "ibt_023_business_process_type": "urn:peppol:bis:selfbilling",
    "ibt_005_invoice_currency_code": "OMR",
    "ibt_007_vat_point_date": "2026-10-01",
    "ibt_009_payment_due_date": "2026-10-31"
  },
  "seller_details": {
    "ibt_027_seller_name": "Gulf IT Support LLC",
    "ibt_031_seller_vatin": "OM1100222333",
    "ibt_031_1_tax_scheme_code": "VAT",
    "ibt_034_seller_electronic_address": "OM1100222333",
    "ibt_034_1_scheme_identifier": "0248",
    "ibt_035_seller_address_line_1": "Way 4410, Building 18",
    "ibt_036_seller_address_line_2": "Falaj Al Qabail",
    "ibt_037_seller_city": "Sohar",
    "ibt_038_seller_post_code": "311",
    "ibt_040_seller_country_code": "OM"
  },
  "buyer_details": {
    "ibt_044_buyer_name": "Al Noor Trading LLC",
    "ibt_048_buyer_vatin": "OM1100123456",
    "ibt_048_1_tax_scheme_code": "VAT",
    "ibt_049_buyer_electronic_address": "OM1100123456",
    "ibt_049_1_scheme_identifier": "0248",
    "ibt_050_buyer_address_line_1": "Way 2317, Building 192",
    "ibt_051_buyer_address_line_2": "Ruwi",
    "ibt_052_buyer_city": "Muscat",
    "ibt_053_buyer_post_code": "112",
    "ibt_055_buyer_country_code": "OM"
  },
  "payment_details": {
    "ibt_081_payment_means_code": "30",
    "ibt_084_payment_account_identifier": "OM9300001234567890123456",
    "ibt_020_payment_terms_note": "Net 30 days"
  },
  "totals": {
    "ibt_106_sum_of_line_net_amount": "480.000",
    "ibt_107_sum_of_allowances": "0.000",
    "ibt_108_sum_of_charges": "0.000",
    "ibt_109_invoice_total_without_vat": "480.000",
    "ibt_110_invoice_total_vat_amount": "24.000",
    "ibt_111_vat_amount_accounting_currency": "24.000",
    "ibt_112_invoice_total_with_vat": "504.000",
    "ibt_113_paid_amount": "0.000",
    "ibt_114_rounding_amount": "0.000",
    "ibt_115_amount_due_for_payment": "504.000"
  },
  "vat_breakdown": [
    {
```

```
    "ibt_116_vat_category_taxable_amount": "480.000",
    "ibt_117_vat_category_tax_amount": "24.000",
    "ibt_118_vat_category_code": "S",
    "ibt_118_1_tax_scheme_code": "VAT",
    "ibt_119_vat_category_rate": "5.00"
  }
],
"lines": [
  {
    "ibt_126_line_identifier": "1",
    "ibt_129_invoiced_quantity": "12.000",
    "ibt_130_quantity_unit_code": "EA",
    "ibt_131_line_net_amount": "480.000",
    "ibt_146_item_net_price": "40.000",
    "ibt_151_item_vat_category_code": "S",
    "ibt_152_item_vat_rate": "5.00",
    "ibt_153_item_name": "IT support services, September",
    "btom_016_line_vat_amount": "24.000",
    "btom_017_line_total_including_vat": "504.000",
    "btom_019_goods_or_services_identification": "S",
    "btom_033_industrial_classification_code": "620101",
    "btom_033_1_scheme_name": "CC"
  }
]
}
```

**cURL · POST invoices/**

```
curl --request POST "https://uat-api.faturathi.com/api/v1/invoices/" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "X-API-KEY: <API_KEY>" \
  --header "Authorization: Bearer <ACCESS_TOKEN>" \
  --data @389-self-billed-invoice.json
```

## 7.8 Self-billed credit note — type code 261

Corrects a self-billed invoice. `preceding_invoice_number` points to the self-billed invoice.

## Request body · 261-self-billed-credit-note.json

```
{
  "header": {
    "btom_001_invoice_transaction_type": "10000000000000000000",
    "btom_002_invoice_uuid": "3f5a7b9c-1d2e-5f40-8a6b-7c8d9e0f1a25",
    "ibt_001_invoice_number": "SBCN-2026-0001",
    "ibt_002_invoice_issue_date": "2026-10-08",
    "ibt_168_invoice_issue_time": "10:30:00",
    "ibt_003_invoice_type_code": "261",
    "ibt_024_specification_identifier": "urn:peppol:pint:selfbilling-1@om-1",
    "ibt_023_business_process_type": "urn:peppol:bis:selfbilling",
    "ibt_005_invoice_currency_code": "OMR",
    "ibt_007_vat_point_date": "2026-10-08",
    "ibt_025_preceding_invoice_reference": "SB-2026-0001",
    "ibt_026_preceding_invoice_issue_date": "2026-10-01"
  },
  "seller_details": {
    "ibt_027_seller_name": "Gulf IT Support LLC",
    "ibt_031_seller_vatin": "OM1100222333",
    "ibt_031_1_tax_scheme_code": "VAT",
    "ibt_034_seller_electronic_address": "OM1100222333",
    "ibt_034_1_scheme_identifier": "0248",
    "ibt_035_seller_address_line_1": "Way 4410, Building 18",
    "ibt_036_seller_address_line_2": "Falaj Al Qabail",
    "ibt_037_seller_city": "Sohar",
    "ibt_038_seller_post_code": "311",
    "ibt_040_seller_country_code": "OM"
  },
  "buyer_details": {
    "ibt_044_buyer_name": "Al Noor Trading LLC",
    "ibt_048_buyer_vatin": "OM1100123456",
    "ibt_048_1_tax_scheme_code": "VAT",
    "ibt_049_buyer_electronic_address": "OM1100123456",
    "ibt_049_1_scheme_identifier": "0248",
    "ibt_050_buyer_address_line_1": "Way 2317, Building 192",
    "ibt_051_buyer_address_line_2": "Ruwi",
    "ibt_052_buyer_city": "Muscat",
    "ibt_053_buyer_post_code": "112",
    "ibt_055_buyer_country_code": "OM"
  },
  "totals": {
    "ibt_106_sum_of_line_net_amount": "40.000",
    "ibt_107_sum_of_allowances": "0.000",
    "ibt_108_sum_of_charges": "0.000",
    "ibt_109_invoice_total_without_vat": "40.000",
    "ibt_110_invoice_total_vat_amount": "2.000",
    "ibt_111_vat_amount_accounting_currency": "2.000",
    "ibt_112_invoice_total_with_vat": "42.000",
    "ibt_113_paid_amount": "0.000",
    "ibt_114_rounding_amount": "0.000",
    "ibt_115_amount_due_for_payment": "42.000"
  },
  "vat_breakdown": [
    {
      "ibt_116_vat_category_taxable_amount": "40.000",
      "ibt_117_vat_category_tax_amount": "2.000",
      "ibt_118_vat_category_code": "S",
      "ibt_118_1_tax_scheme_code": "VAT",
    }
  ]
}
```

```
    "ibt_119_vat_category_rate": "5.00"
  }
],
"lines": [
  {
    "ibt_126_line_identifier": "1",
    "ibt_129_invoiced_quantity": "1.000",
    "ibt_130_quantity_unit_code": "EA",
    "ibt_131_line_net_amount": "40.000",
    "ibt_146_item_net_price": "40.000",
    "ibt_151_item_vat_category_code": "S",
    "ibt_152_item_vat_rate": "5.00",
    "ibt_153_item_name": "IT support services, credit for 1 visit",
    "btom_016_line_vat_amount": "2.000",
    "btom_017_line_total_including_vat": "42.000",
    "btom_019_goods_or_services_identification": "S",
    "btom_033_industrial_classification_code": "620101",
    "btom_033_1_scheme_name": "CC"
  }
]
}
```

**cURL · POST invoices/**

```
curl --request POST "https://uat-api.faturathi.com/api/v1/invoices/" \
  --header "Accept: application/json" \
  --header "Content-Type: application/json" \
  --header "X-API-KEY: <API_KEY>" \
  --header "Authorization: Bearer <ACCESS_TOKEN>" \
  --data @261-self-billed-credit-note.json
```

## 8. Responses and invoice status

A document that passes validation is accepted and queued. Acceptance means Faturathi has the document; it does not mean the buyer or OTA has it yet. Delivery and reporting finish later on their own schedule.

202 Accepted · response body (format to be confirmed)

```
{
  "id": "inv_01J9Z6T4Q8M2",
  "invoice_number": "INV-2026-0001",
  "uuid": "b41c72aa-9912-5f2b-8d34-77c1e09a55d0",
  "status": "processing",
  "validation": { "result": "passed", "warnings": [] },
  "delivery": { "status": "pending" },
  "reporting": { "status": "pending" },
  "created_at": "2026-10-01T09:30:12+04:00"
}
```

Store the returned id against the invoice in your ERP. You need it to track the document, download files and match webhook events.

### 8.1 Status values

| Field     | Meaning                                          | Values                                                 |
|-----------|--------------------------------------------------|--------------------------------------------------------|
| status    | Overall summary for display                      | processing · completed · rejected · failed             |
| delivery  | Peppol delivery to the buyer's access point (C3) | pending · delivered · rejected · failed · not_required |
| reporting | Tax data reporting to OTA (C5)                   | pending · reported · rejected · failed                 |

**Rejected and failed are different.** rejected means the buyer's access point or OTA looked at the document and refused it: fix the data and issue a new document. failed with retrying = true means Faturathi could not reach them yet and is still trying: do not resubmit, or the same sale may be reported twice.

### 8.2 Message Level Status (MLS)

The buyer's access point answers every delivery with a Peppol Message Level Status. AP means accepted, AB means acknowledged, RE means rejected. Faturathi maps these to the delivery status and keeps the original MLS for audit.

### 8.3 Webhooks (to be confirmed)

Faturathi can call a URL on your server when a document changes: invoice.delivered, invoice.reported, invoice.rejected, invoice.failed and invoice.received (an invoice a supplier sent to you). Your endpoint must use HTTPS, reply within 10 seconds and verify the signature header. Events may be sent more than once, so store each event ID and skip duplicates.

## 9. Errors and troubleshooting

| HTTP | Meaning                                            | What to do                                                      |
|------|----------------------------------------------------|-----------------------------------------------------------------|
| 400  | Malformed JSON or wrong field type                 | Fix the request format; check quotes, commas and string amounts |
| 401  | API key or token missing, wrong or expired         | Check both headers; generate a new token in the portal          |
| 403  | Credentials valid but not allowed for this company | Check that the key belongs to the seller's company              |
| 404  | Document ID not found                              | Check the ID and the environment (UAT vs production)            |
| 409  | Invoice number already used                        | Do not resend; look up the existing document                    |
| 422  | Failed PINT-OM validation                          | Fix the fields listed in the error details and resend           |
| 429  | Too many requests                                  | Wait for the time in the Retry-After header                     |
| 5xx  | Faturathi or network problem                       | Retry with growing waits (2, 4, 8, 16 seconds)                  |

### 422 Unprocessable Entity · error body (format to be confirmed)

```
{
  "error": {
    "code": "VALIDATION_FAILED",
    "message": "The document failed PINT-OM validation.",
    "details": [
      {
        "field": "lines[0].ibt_152_item_vat_rate",
        "rule": "<PINT-OM rule id>",
        "message": "Standard rated lines must use a 5.00 VAT rate."
      }
    ]
  },
  "request_id": "req_7Hk2P9"
}
```

Always log the request\_id. Faturathi support uses it to find your call. Build your logic on the error code and rule, not on the message text, which may change.

### 9.1 Where the problem usually is

| Symptom              | Layer                | First thing to check                                                        |
|----------------------|----------------------|-----------------------------------------------------------------------------|
| 401 / 403            | Security             | Both headers present; key belongs to the right company; UAT key used on UAT |
| 400                  | Request format       | Amounts sent as strings; valid JSON; correct group names                    |
| 422 with field names | Data / PINT-OM rules | The listed fields: VAT rate, totals arithmetic, buyer address, code lists   |
| delivery = rejected  | Buyer's access point | Buyer Peppol address; document type the buyer                               |

| Symptom                        | Layer          | First thing to check                                                                  |
|--------------------------------|----------------|---------------------------------------------------------------------------------------|
|                                |                | can receive                                                                           |
| delivery = failed,<br>retrying | Peppol network | Nothing to do; Faturathi retries. Contact support if it lasts more than a working day |
| reporting = rejected           | OTA            | Read the reason in the status response; correct and issue a new document              |

## 10. Field reference

Required fields must always be sent. Conditional fields depend on the document type, buyer or VAT category. Optional fields can be left out. Field names are shown exactly as the API expects them.

### Document header · header

| Field                                | Type          | Need               | Description                                                                                                       |
|--------------------------------------|---------------|--------------------|-------------------------------------------------------------------------------------------------------------------|
| btom_001_invoice_transaction_type    | string (20)   | <b>Required</b>    | Oman transaction type code: 20 characters, each 0 or 1. Each position switches one transaction feature on or off. |
| btom_002_invoice_uuid                | string (UUID) | <b>Required</b>    | Unique ID for this document. Never reuse it, even after a rejection.                                              |
| ibt_001_invoice_number               | string        | <b>Required</b>    | Your own document number. Must be unique for your company.                                                        |
| ibt_002_invoice_issue_date           | date          | <b>Required</b>    | Date the document was issued, YYYY-MM-DD.                                                                         |
| ibt_168_invoice_issue_time           | time          | <b>Required</b>    | Time of issue, HH:MM:SS, Oman time.                                                                               |
| ibt_003_invoice_type_code            | string        | <b>Required</b>    | 380, 381, 383, 389 or 261. See Document types.                                                                    |
| ibt_024_specification_identifier     | string        | <b>Required</b>    | urn:peppol:pint:billing-1@om-1, or urn:peppol:pint:selfbilling-1@om-1 for self-billed documents.                  |
| ibt_023_business_process_type        | string        | <b>Required</b>    | urn:peppol:bis:billing, or urn:peppol:bis:selfbilling for self-billed documents.                                  |
| ibt_005_invoice_currency_code        | string (3)    | <b>Required</b>    | ISO 4217 currency, for example OMR.                                                                               |
| ibt_007_vat_point_date               | date          | <b>Optional</b>    | Date VAT becomes due, if different from the issue date.                                                           |
| ibt_009_payment_due_date             | date          | <b>Conditional</b> | Date payment is due. Needed when an amount is due for payment.                                                    |
| ibt_025_preceding_invoice_reference  | string        | <b>Conditional</b> | Number of the original invoice. Required on credit notes and debit notes. (Field name to be confirmed.)           |
| ibt_026_preceding_invoice_issue_date | date          | <b>Conditional</b> | Issue date of the original invoice. (Field name to be confirmed.)                                                 |

## Seller · seller\_details

| Field                               | Type       | Need               | Description                                                          |
|-------------------------------------|------------|--------------------|----------------------------------------------------------------------|
| ibt_027_seller_name                 | string     | <b>Required</b>    | Seller's registered legal name.                                      |
| ibt_029_seller_identifier           | string     | <b>Optional</b>    | Another ID for the seller, such as a commercial registration number. |
| ibt_029_1_seller_identifier_scheme  | string     | <b>Conditional</b> | Type of identifier. Required when identifier is sent.                |
| ibt_031_seller_vatin                | string     | <b>Required</b>    | Seller's Oman VAT identification number, for example OM1100123456.   |
| ibt_031_1_tax_scheme_code           | string     | <b>Required</b>    | Always VAT.                                                          |
| ibt_034_seller_electronic_addresses | string     | <b>Required</b>    | Seller's Peppol address. In Oman this is the VATIN.                  |
| ibt_034_1_scheme_identifier         | string     | <b>Required</b>    | Always 0248 for Oman.                                                |
| ibt_035_seller_address_line_1       | string     | <b>Required</b>    | Street, way and building.                                            |
| ibt_036_seller_address_line_2       | string     | <b>Optional</b>    | Area or district.                                                    |
| ibt_162_seller_address_line_3       | string     | <b>Optional</b>    | Governorate or other detail.                                         |
| ibt_037_seller_city                 | string     | <b>Required</b>    | City.                                                                |
| ibt_038_seller_post_code            | string     | <b>Optional</b>    | Postal code.                                                         |
| ibt_040_seller_country_code         | string (2) | <b>Required</b>    | ISO 3166 country, OM.                                                |
| ibt_041_seller_contact_name         | string     | <b>Optional</b>    | Contact person for this invoice.                                     |
| ibt_042_seller_contact_telephone    | string     | <b>Optional</b>    | Contact phone, international format.                                 |
| ibt_043_seller_contact_email        | string     | <b>Optional</b>    | Contact email.                                                       |

## Buyer · buyer\_details

| Field                             | Type   | Need               | Description                                                         |
|-----------------------------------|--------|--------------------|---------------------------------------------------------------------|
| ibt_044_buyer_name                | string | <b>Required</b>    | Buyer's registered legal name.                                      |
| ibt_046_buyer_identifier          | string | <b>Optional</b>    | Another ID for the buyer, such as a commercial registration number. |
| ibt_046_1_buyer_identifier_scheme | string | <b>Conditional</b> | Type of identifier. Required when identifier is sent.               |
| ibt_048_buyer_vatin               | string | <b>Conditional</b> | Buyer's Oman VATIN. Required for B2B.                               |
| ibt_048_1_tax_scheme_code         | string | <b>Conditional</b> | VAT when vatin is sent.                                             |
| ibt_049_buyer_electronic_address  | string | <b>Conditional</b> | Buyer's Peppol address (VATIN). Required for delivery over Peppol.  |
| ibt_049_1_scheme_identifier       | string | <b>Conditional</b> | Always 0248 for Oman.                                               |

| Field                        | Type       | Need        | Description                       |
|------------------------------|------------|-------------|-----------------------------------|
| ibt_050_buyer_address_line_1 | string     | Conditional | Street, way and building.         |
| ibt_051_buyer_address_line_2 | string     | Optional    | Building or area.                 |
| ibt_163_buyer_address_line_3 | string     | Optional    | Wilayat or other detail.          |
| ibt_052_buyer_city           | string     | Conditional | City.                             |
| ibt_053_buyer_post_code      | string     | Optional    | Postal code.                      |
| ibt_055_buyer_country_code   | string (2) | Required    | ISO 3166 country, for example OM. |

**B2C documents.** For a sale to a consumer, send only the buyer details you have. A name and country code are enough; leave out the VATIN and the electronic address. (To be confirmed against PINT-OM v1.0.1.)

## Payment · payment\_details

| Field                              | Type   | Need        | Description                                                     |
|------------------------------------|--------|-------------|-----------------------------------------------------------------|
| ibt_081_payment_means_code         | string | Conditional | How the buyer pays, for example 30 credit transfer or 10 cash.  |
| ibt_084_payment_account_identifier | string | Conditional | IBAN or account number to pay into. Needed for credit transfer. |
| ibt_020_payment_terms_note         | string | Optional    | Payment terms in words, for example Net 30 days.                |

## Totals · totals

| Field                                  | Type           | Need        | Description                                                    |
|----------------------------------------|----------------|-------------|----------------------------------------------------------------|
| ibt_106_sum_of_line_net_amount         | decimal string | Required    | Sum of all lines[].net_amount.                                 |
| ibt_107_sum_of_allowances              | decimal string | Optional    | Sum of document-level discounts.                               |
| ibt_108_sum_of_charges                 | decimal string | Optional    | Sum of document-level charges.                                 |
| ibt_109_invoice_total_without_vat      | decimal string | Required    | line_net_total – allowance_total + charge_total.               |
| ibt_110_invoice_total_vat_amount       | decimal string | Required    | Total VAT. Must equal the sum of vat_breakdown[].tax_amount.   |
| ibt_111_vat_amount_accounting_currency | decimal string | Conditional | Total VAT in OMR. Needed when the invoice currency is not OMR. |
| ibt_112_invoice_total_with_vat         | decimal string | Required    | total_without_vat + total_vat.                                 |

| Field                          | Type           | Need     | Description                                     |
|--------------------------------|----------------|----------|-------------------------------------------------|
| ibt_113_paid_amount            | decimal string | Optional | Amount already paid, such as an advance.        |
| ibt_114_rounding_amount        | decimal string | Optional | Rounding applied to the payable amount.         |
| ibt_115_amount_due_for_payment | decimal string | Required | total_with_vat – paid_amount + rounding_amount. |

## VAT breakdown (one row per VAT category and rate) · vat\_breakdown

| Field                               | Type           | Need        | Description                                    |
|-------------------------------------|----------------|-------------|------------------------------------------------|
| ibt_116_vat_category_taxable_amount | decimal string | Required    | Net amount taxed in this category.             |
| ibt_117_vat_category_tax_amount     | decimal string | Required    | VAT for this category.                         |
| ibt_118_vat_category_code           | string         | Required    | VAT category, for example S for standard rate. |
| ibt_118_1_tax_scheme_code           | string         | Required    | Always VAT.                                    |
| ibt_119_vat_category_rate           | decimal string | Conditional | VAT rate, for example 5.00.                    |

## Lines · lines

| Field                          | Type           | Need        | Description                                              |
|--------------------------------|----------------|-------------|----------------------------------------------------------|
| ibt_126_line_identifier        | string         | Required    | Line number, unique within the document.                 |
| ibt_127_line_note              | string         | Optional    | Free-text note for the line.                             |
| ibt_129_invoiced_quantity      | decimal string | Required    | Quantity invoiced.                                       |
| ibt_130_quantity_unit_code     | string         | Required    | UN/ECE Rec 20 unit, for example EA (each) or HUR (hour). |
| ibt_131_line_net_amount        | decimal string | Required    | quantity × net_price, before VAT.                        |
| ibt_146_item_net_price         | decimal string | Required    | Price of one unit, before VAT.                           |
| ibt_151_item_vat_category_code | string         | Required    | VAT category for the line, for example S.                |
| ibt_152_item_vat_rate          | decimal string | Conditional | VAT rate, 5.00 for category S.                           |
| ibt_153_item_name              | string         | Required    | Name of the item or service.                             |

| Field                                     | Type           | Need               | Description                                                   |
|-------------------------------------------|----------------|--------------------|---------------------------------------------------------------|
| btom_016_line_vat_amount                  | decimal string | <b>Required</b>    | VAT for this line.                                            |
| btom_017_line_total_including_vat         | decimal string | <b>Required</b>    | net_amount + vat_amount.                                      |
| btom_019_goods_or_services_identification | string         | <b>Required</b>    | G for goods or S for services.                                |
| btom_033_industrial_classification_code   | string         | <b>Conditional</b> | Classification code for the item or service.                  |
| btom_033_1_scheme_name                    | string         | <b>Conditional</b> | Code list the classification_code comes from, for example CC. |

## 11. Validation rules to watch

Faturathi validates essential fields such as the invoice number, invoice date and customer name, and checks that the calculation totals add up, before the document goes through the shared Peppol and Oman Schematron rules. The following rules cause most rejections during onboarding.

| Area             | Rule                                                                                                                                                |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Invoice type     | Use 380 for a tax invoice, 381 for a credit note, 389 for a self-billed invoice, 261 for a self-billed credit note.                                 |
| Transaction type | A 20-character string of 0 and 1. Use the code that matches the transaction (B2B, B2C, self-billing and other cases).                               |
| VAT              | Category S is the standard rate and carries a rate of exactly 5.00. Exempt and zero-rated lines need the correct category and reason.               |
| Arithmetic       | Line net = quantity × net price. Sum of lines = line net total. Total with VAT = total without VAT + total VAT. VAT breakdown must match the lines. |
| Currency         | For OMR documents do not send an exchange rate. Foreign-currency documents need the rate and the VAT amount in OMR.                                 |
| Peppol addresses | Seller and buyer electronic address and scheme 0248 must both be present on B2B documents.                                                          |
| Buyer address    | Street, city and country are required on B2B documents.                                                                                             |
| Identifiers      | The invoice number is unique per seller. The UUID is unique per document and is never reused, even after a rejection.                               |
| References       | Credit and debit notes must refer to the original invoice number and date.                                                                          |

## 12. Duplicates, retries and limits

- **Invoice numbers are unique.** Faturathi refuses a second document with the same invoice number for your company. This protects you from reporting one sale twice to OTA.
- **Retry only when it helps.** Retry 429 and 5xx responses with growing waits. Do not retry 400, 401, 403 or 422 without fixing something first.
- **Network timeouts.** If you never received a response, look the invoice up by number before sending it again.
- **Peppol and OTA retries are automatic.** Once a document is accepted, Faturathi retries delivery and reporting itself. You do not resubmit.
- **Rate limits.** Limits per key are set at onboarding. For month-end volumes, talk to us about bulk submission.

## 13. Peppol technical reference

You do not need these values to use the API. They are provided for IT teams and partners who want to understand what Faturathi sends on the network. Values follow the PINT-OM specifications published for Oman.

| Item                    | Value                                                                                                                                   |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Participant ID scheme   | iso6523-actorid-upis::0248:{VATIN}                                                                                                      |
| Syntax                  | UBL 2.1 XML, UTF-8                                                                                                                      |
| Transport               | Peppol AS4 (AS4 v2 transport profile)                                                                                                   |
| PINT-OM invoice         | peppol-doctype-wildcard::urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:peppol:pint:billing-1@om-1::2.1           |
| PINT-OM credit note     | peppol-doctype-wildcard::urn:oasis:names:specification:ubl:schema:xsd:CreditNote-2::CreditNote##urn:peppol:pint:billing-1@om-1::2.1     |
| Billing process         | cenbii-procid-ubl::urn:peppol:bis:billing                                                                                               |
| Self-billed invoice     | peppol-doctype-wildcard::urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##urn:peppol:pint:selfbilling-1@om-1::2.1       |
| Self-billed credit note | peppol-doctype-wildcard::urn:oasis:names:specification:ubl:schema:xsd:CreditNote-2::CreditNote##urn:peppol:pint:selfbilling-1@om-1::2.1 |
| Self-billing process    | cenbii-procid-ubl::urn:peppol:bis:selfbilling                                                                                           |
| Message Level Status    | busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2::ApplicationResponse##urn:peppol:edec:mls:1.0::2.1 |
| MLS process             | cenbii-procid-ubl::urn:peppol:edec:mls                                                                                                  |
| Tax reporting to OTA    | OM Tax Data Document, generated and sent by Faturathi                                                                                   |

## 14. Testing checklist and go-live

Run these scenarios in UAT and keep the results for your UAT sign-off. Go-live follows discovery, field mapping, UAT, a pilot with a small group of buyers, and then a phased roll-out.

| #  | Scenario                                        | Expected result                                          |
|----|-------------------------------------------------|----------------------------------------------------------|
| 1  | B2B standard tax invoice (380)                  | Accepted, delivered to the buyer, reported to OTA        |
| 2  | B2C simplified tax invoice (380)                | Accepted and reported; QR data available for the receipt |
| 3  | Credit note (381) referring to an invoice       | Accepted; reference validated                            |
| 4  | Debit note (383) referring to an invoice        | Accepted; reference validated                            |
| 5  | Self-billed invoice (389) and credit note (261) | Accepted with the self-billing specification             |
| 6  | Invoice with a wrong VAT rate                   | 422 with the field and rule named                        |
| 7  | Same invoice number sent twice                  | Second attempt refused                                   |
| 8  | Invoice to a buyer not on Peppol                | Delivery not possible; status explains why               |
| 9  | Foreign-currency invoice with exchange rate     | Accepted with OMR VAT amount                             |
| 10 | Request without credentials                     | 401 refused                                              |

### 14.1 Go-live checklist

- Production API key created for each company; secrets stored in the server's secret store.
- ERP updated with invoice ID, UUID and status fields; XML archived with the accounting record.
- Webhook or polling in place; exception report for rejected and failed documents.
- Reconciliation between ERP, Faturathi and OTA agreed with finance.
- Support contacts and escalation path shared with the operations team.

## 15. Support and glossary

Technical support is available 24/7 from Muscat. Email [faturathi@netbue.com](mailto:faturathi@netbue.com) and include the request\_id, invoice number and environment. Never send passwords, keys or tokens by email.

### 15.1 Glossary

| Term              | Meaning                                                                                  |
|-------------------|------------------------------------------------------------------------------------------|
| OTA               | Oman Tax Authority                                                                       |
| Peppol            | International network for exchanging electronic business documents between access points |
| Access point (AP) | A certified gateway to the Peppol network; Faturathi is your access point                |
| SMP / SML         | Directories that tell an access point where and how a participant can receive documents  |
| AS4               | The secure messaging protocol used between Peppol access points                          |
| PINT-OM           | Peppol International Invoice specification for Oman, based on UBL 2.1                    |
| IBT / BTOM        | Business term identifiers from the Peppol (IBT) and Oman (BTOM) specifications           |
| MLS               | Message Level Status: the receiving access point's technical response to a document      |
| TDD               | Tax Data Document: the tax data reported to OTA by the access points                     |
| VATIN             | Oman VAT identification number, for example OM1100123456                                 |
| UUID              | Universally unique identifier assigned to each document                                  |

## Appendix A. Internal review notes (remove before distribution)

Items to confirm with the Faturathi development team before this document is shared externally:

| #  | Item                                                                                                                                                      | Where          |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| 1  | Logo: replace the placeholder wordmark with the official Faturathi and NETBUE logo files (cover, header).                                                 | Cover, header  |
| 2  | Confirm that both X-API-KEY and the bearer token are required on every request, and whether generating a new token cancels the old one.                   | Section 4      |
| 3  | Confirm the response body for 202 Accepted and the error body for 422, including the rule identifier.                                                     | Sections 8, 9  |
| 4  | Confirm the paths for get, list, XML and PDF endpoints, and the webhook events and signature.                                                             | Sections 3, 8  |
| 5  | Confirm the transaction type code for B2C and self-billed documents. The B2C sample carries a placeholder.                                                | Sections 7, 11 |
| 6  | Confirm the field names for the preceding invoice reference (ibt_025 / ibt_026) used on credit and debit notes.                                           | Sections 7, 10 |
| 7  | Seller/buyer identifier 112345678900003 with scheme OTHID in the samples resembles a UAE tax number. Replace with the correct Oman identifier and scheme. | Section 7      |
| 8  | Confirm whether ibt_111 (VAT amount in accounting currency) is allowed when the document currency is already OMR.                                         | Sections 7, 10 |
| 9  | Confirm that debit note 383 is in the PINT-OM v1.0.1 invoice type code list.                                                                              | Section 6      |
| 10 | Confirm the production base URL and the go-live date; mark accreditation status accurately (subject to OTA confirmation).                                 | Section 3      |