

Incoming invoices

Import received e-invoices and PDFs, read their content, validate, review and pay them.

Source: <https://alyabooks.ai/docs/guides/incoming-invoices>

Incoming invoices are documents your organization receives from suppliers. Alya Books records them from three channels: the app (upload), the organization's inbox address and the ASP's Peppol inbox, and this API.

Import

POST /import/incoming accepts:

- multipart/form-data with a file field (plus optional extra files, sender, reference),
- raw application/xml or application/pdf bodies (?filename= optional),
- JSON { "xml": "<Invoice ...>" } or { "base64": "...", "filename": "invoice.pdf", "mime": "application/pdf" }.

```
curl -X POST "$BASE/import/incoming" -H "X-API-KEY: $KEY" -F file=@supplier-invoice.xml -F reference=P0-4711
```

What happens next depends on the file:

Input	Processing
UBL XML (Invoice or CreditNote)	Parsed, validated against PINT AE and against your organization's TRN as buyer, stored with lines, VAT breakdown and embedded attachments; a visualisation is rendered.
PDF or image	Stored; structured data is extracted by the AI extraction service when it is configured on the environment, otherwise the document waits in draft for manual completion.

The response is 202 when the document was recorded (even with warnings), or 422 with code: "E_VALIDATION" when it failed (duplicate supplier TRN and number, wrong buyer TRN, unparsable file). In both cases the import is logged and extra /the body carry importId, incomingInvoiceId, status and validation[].

Reading an import

All sub-resources take the import id; the incoming invoice id is accepted too.

Route	Returns
GET /import/incoming/{importId}	Import record and invoice summary

Route	Returns
.../status	status (pending, accepted, failed), validation and visualisation status, review and payment status
.../validation	validity, errors[], warnings[] (PINT AE rules such as BT-31, BT-131, BR-C0-15)
.../content	Full content: seller, buyer, lineItems[], taxes[], paymentTerms, paymentMeans, delivery, metadata
.../content/summary, /lineitems, /seller, /buyer, /taxes, /paymentterms, /paymentmeans, /delivery	One part of the content
.../content/pdf	The original PDF, or the visualisation rendered to PDF (needs an HTML-to-PDF service; without one it answers 503, so use ?format=html)
.../source	The original file as received
.../attachments, .../attachment/{id}, .../attachment/{id}/file	Files that came with the document

GET /import/incoming lists imports (status, channel, keyword, startDate, endDate); GET /incoming lists the incoming invoices themselves (status, paymentStatus, supplierId, keyword, issue date range).

Other currencies

For documents in another currency than AED the incoming invoice carries exchangeRate (AED per 1 unit) and exchangeRateSource: document when the supplier's XML stated it (BTAE-04, cac:TaxExchangeRate; rates stated from AED or with MathematicOperatorCode Divide are turned around), cbuae when Alya Books took the UAE Central Bank rate for the delivery date or else the issue date (exchangeRateDate is the publication date), or manual when a reviewer set it. aedTotals is VAT and the total in AED at that rate; statedAed holds the AED amounts from the supplier's XML (IBT-111, BTAE-20). Validation warns when the supplier stated no rate (IBR-159-AE), no AED amounts, or AED VAT that does not match VAT × rate (IBR-175-AE). PDFs and images read by AI get the Central Bank rate for the issue date. Send exchangeRate in PATCH to change it while the invoice is in review (null removes it); changing currency without a rate looks up the Central Bank rate again.

Review

PATCH /import/incoming/{importId} corrects the content while the invoice is a draft or in review, and changes the review state and bookkeeping fields at any time. Content edits return 409 once the invoice is accepted or rejected; send the decision in a separate request.

```
{
  "invoiceNumber": "G05-2026-0006",
  "dueDate": "2026-10-26",
  "seller": { "name": "Gulf Office Supplies LLC", "street": "Al Quoz Industrial Area 3",
  "city": "Dubai", "country": "AE", "trn": "100555666777003", "email":
```

```

"billing@gulfoffice.ae" },
  "buyer": { "name": "SAB Engineering LLC", "city": "Dubai", "trn": "100123456789003" },
  "delivery": { "name": "Warehouse 4", "street": "Plot 598-1024, Dubai Investments Park",
"city": "Dubai", "deliveryNote": "Gate 2" },
  "lineItems": [
    { "description": "A4 copier paper, 80gsm", "itemDescription": "Box of 5 reams",
"quantity": 3, "unitCode": "BX", "unitPrice": 95, "taxCategory": "S", "taxRate": 5 },
    { "description": "Toner cartridge HP 26A", "quantity": 4, "unitPrice": 285, "taxRate":
5 }
  ],
  "paymentTerms": "Net 30",
  "earlyPaymentDiscounts": [{ "days": 10, "percent": 2 }],
  "paymentMeans": [{ "code": "30", "accountId": "AE560330000001234567890", "accountName":
"Gulf Office Supplies LLC", "bic": "EBILAEAD" }]
}

```

- Parties (seller, buyer, delivery) take name, title, firstName, lastName, legalForm, street, additional, city, emirate, postalCode, country, email, phone, contactName, trn, tin, peppolId, licenseNumber, gln, mandateReference and, for delivery, deliveryNote. A party you send replaces the stored one.
- lineItems replaces all lines and recalculates the VAT breakdown and the totals. vatBreakdown (category, rate, taxable, tax, exemptionReason) replaces the breakdown and recalculates the totals; subtotal, taxTotal and total override them.
- paymentMeans replaces all payment means (code from UNTDID 4461, accountId or iban, accountName, bic, bankName, reference, information, mandateReference). earlyPaymentDiscounts holds up to three tiers.
- GET .../content returns the same shapes, so you can read, change and send back.

Review decisions and bookkeeping:

```

{ "status": "accepted" }
{ "status": "rejected", "rejectionReason": "Duplicate of INV-2201" }
{ "status": "draft" }
{ "status": "archived", "archiveNote": "Paid outside Alya Books" }
{ "categoryId": "...", "costCentreId": "...", "projectId": "...", "notes": "...", "metadata": {
"cost_code": "OPS-3" } }

```

Accepted invoices are locked in the app (members need the *edit accepted incoming invoices* permission to change them). Payments to the supplier are recorded with `POST /import/incoming/{importId}/payments` once the invoice is accepted; `paymentStatus` and `paidAmount` update automatically. Imports that were not accepted can be removed with `DELETE`.

Events

`incoming.created`, `incoming.validated`, `incoming.accepted`, `incoming.rejected` and `incoming.paid` are emitted as [webhooks](#). The payload carries the incoming invoice summary, including `importId`.

Receiving without the API

- **Inbox address:** each organization can create `<local-part>@in.alyabooks.ai` under Settings → Sending. Attachments of emails to that address run through the same pipeline with channel `email`.
- **Peppol:** the app's *Fetch from Peppol* action, and automatically every 15 minutes for UAT and live connections, asks the ASP for documents addressed to the organization's participant id (channel `peppol`).

Both channels produce the same imports and the same webhooks, so an integration built on `GET /import/incoming` sees every document regardless of how it arrived.

Simulate deliveries in Test mode

While the organization's e-invoicing connection is in Test (simulated) mode, you can make Alya Books deliver supplier invoices to you, to exercise your integration end to end (webhooks, polling, accept/reject, payments) without waiting for real suppliers:

```
curl -X POST "$BASE/incoming/simulate" -H "X-API-KEY: $KEY" -H "Content-Type: application/json" \
  -d '{ "scenario": "mixed_vat", "supplier": "it", "count": 2 }'
```

Scenario	Result
<code>standard</code> , <code>with_pdf</code> , <code>mixed_vat</code> , <code>reverse_charge</code> , <code>credit_note</code> , <code>foreign_currency</code>	Valid PINT AE documents; an incoming invoice in status <code>draft</code> (to review). <code>with_pdf</code> embeds the supplier's PDF in the XML, which the app shows as the supplier copy
<code>wrong_buyer</code> , <code>totals_mismatch</code>	Import recorded with <code>validationStatus: failed</code> and the issues
<code>duplicate</code>	Repeats the supplier's previous number; the import fails as a duplicate

Use `contactId` instead of `supplier` to have one of your contacts (with a TRN) act as the supplier. The response lists the import and incoming invoice ids; `incoming.created`, `incoming.validated` and failure webhooks fire as for real deliveries. In UAT and Live mode the endpoint returns `409`.