

Getting started

Create an API key, send an invoice, and follow it to the FTA in ten minutes.

Source: <https://alyabooks.ai/docs/getting-started>

1. Create an API key

Sign in to Alya Books, open **API access** in the sidebar and click **New key**. Choose a name, the scopes it needs, and whether it is a **test key**. Copy the key when it is shown: it is stored hashed and cannot be displayed again.

Start with a test key. Test documents behave exactly like real ones (validation, numbering, PDF, XML, webhooks) but are marked `isTest`, are excluded from reports and can only be submitted while the organization's e-invoicing connection is in **Test (simulated)** mode.

2. Call the API

Every request carries the key in the `X-API-KEY` header.

```
export BASE=https://api.alyabooks.ai/api/v1
export KEY=sk_test_...
```

```
curl "$BASE/me" -H "X-API-KEY: $KEY"
```

```
{
  "organization": { "id": "...", "name": "Seller LLC", "trn": "100000000000003", "peppolId": "0235:10000000000" },
  "key": { "name": "Integration", "scopes": ["invoices:read", "invoices:write", "..."],
  "isTest": true },
  "environment": "test",
  "e-invoicing": { "provider": "taxilla", "mode": "mock" }
}
```

3. Create an invoice

```
curl -X POST "$BASE/invoices" -H "X-API-KEY: $KEY" -H "Content-Type: application/json" -d '{
  "invoices": [{
    "buyer": { "name": "Buyer LLC", "trn": "1000000000000011", "email": "ap@buyer.ae",
      "address": { "street": "Al Maryah Island", "city": "Abu Dhabi", "country":
```

```
"AE" } },
  "issueDate": "2026-09-23",
  "payment": { "termsDays": 30, "accountId": "AE070331234567890123456" },
  "items": [
    { "description": "Consulting, September", "quantity": 10, "unitPrice": 500 },
    { "description": "Travel (zero rated)", "quantity": 1, "unitPrice": 850,
    "taxCategory": "zero_rated" }
  ],
  "submit": true,
  "send": { "to": ["ap@buyer.ae"] }
}]
}'
```

The response is `202 Accepted` with one result per invoice:

```
[{
  "index": 0, "success": true, "invoiceId": "3f2e...", "invoiceNumber": "INV-0042",
  "status": "submitted", "total": 6100,
  "validation": { "validity": true, "errors": [], "warnings": [] },
  "submission": { "status": "processing", "requestId": "...", "mode": "mock" },
  "delivery": { "state": "skipped", "notes": ["No email provider configured..."] },
  "detailsPage": "https://app.alyabooks.ai/invoices/3f2e.../edit"
}]
```

Omit `invoiceNumber` and Alya Books allocates the next number of the series. The [sample documents](#) show a request and the resulting XML for six document kinds, and the [integration checklist](#) lists the steps from here to the first invoice reported through Taxilla. Omit `submit` to keep the invoice as **validated** and submit later with `POST /invoices/{id}/submit`. Add `draftOnly: true` to store an invoice even when it fails validation.

4. Follow the status

```
curl "$BASE/invoices/3f2e.../status" -H "X-API-KEY: $KEY"
```

`invoiceStatus` moves through `draft` → `validated` → `submitted` → `delivered` → `accepted` (or `rejected`). `authority[]` lists the ASP submissions with their stage, `delivery[]` the emails, `validation` the last PINT AE result. See [ASP and FTA lifecycle](#).

Prefer push over polling: create a [webhook](#) for `invoice.reported`, `invoice.rejected` and `invoice.paid`.

5. Download the documents

```
curl "$BASE/invoices/3f2e...?format=xml" -H "X-API-KEY: $KEY" -o INV-0042.xml
curl "$BASE/invoices/3f2e...?format=pdf" -H "X-API-KEY: $KEY" -o INV-0042.pdf
```

The XML is the PINT AE UBL document (`urn:peppol:pint:billing-1@ae-1`). For issued invoices it is byte-identical to what was sent to the ASP. To hand a file to a browser or another system, `GET`

`/invoices/{id}/download-link?format=pdf` returns a signed URL that works for 15 minutes without a key.

Conventions

- JSON in and out, camelCase keys, dates as `YYYY-MM-DD` (`YYYYMMDD` is accepted on input), amounts as numbers in the invoice currency (AED by default, two decimals).
- Lists: `?page=1&perPage=25&sort=issueDate&order=desc&keyword=...` and a `{ data, meta }` envelope.
- Errors: `{ code, statusCode, message, extra?, field? }`; see [Errors](#).
- Async operations return `202`; poll `/status` or use webhooks.
- Rate limit: 600 requests per minute per key, `429` with `Retry-After` beyond that.
- CA firms: send `X-Organization-Id: <client id>` to act for a managed client; `GET /organizations/managed` lists them.

Prefer clicking

Import the Postman collection at `/docs/api.postman_collection.json` on the Alya Books site, set its `baseUrl` and `apiKey` variables, and start with the folder "Start here: ERP integration". The site's test console (`/console`) runs freight samples and a dummy data feed without any tool at all.

Next steps

- [Outgoing invoices](#)
- [Webhooks](#)
- [API reference](#)