InvoiceBridge E-Invoice API — Quick start

Read, generate, convert and validate EU e-invoices: XRechnung, ZUGFeRD / Factur-X, Peppol BIS. Three calls, five minutes.

Formats

Format id (`to=`)	What it is	Read	Write
`xrechnung-ubl`, `xrechnung-cii`	XRechnung 3.0 — German public sector; B2B e-invoicing in Germany from 2025–2028	✓	✓
`peppol-bis`	Peppol BIS Billing 3.0 (UBL)	✓	✓
`facturx-cii`	Factur-X / ZUGFeRD 2.x, profile EN 16931 (XML; reading also from the hybrid PDF)	✓ XML + PDF	✓ XML
`en16931-ubl`	Plain EN 16931 UBL	✓	✓

Authentication

Every request carries your RapidAPI key and the API host. Send XML as `application/xml`, a PDF as `application/pdf`, JSON as `application/json`.

```
-H "X-RapidAPI-Key: YOUR_RAPIDAPI_KEY" \
-H "X-RapidAPI-Host: invoicebridge-e-invoice-api.p.rapidapi.com"
```

1 · Read an e-invoice into JSON

`POST` `/v1/parse` — XRechnung, Peppol or ZUGFeRD XML, or a Factur-X / ZUGFeRD PDF with the XML inside.

```
curl -X POST "https://invoicebridge-e-invoice-api.p.rapidapi.com/v1/parse" \
  -H "X-RapidAPI-Key: YOUR_RAPIDAPI_KEY" -H "X-RapidAPI-Host: invoicebridge-e-invoice-
api.p.rapidapi.com" \
  -H "Content-Type: application/xml" --data-binary @invoice.xml
```

```
{
  "format": "XRechnung (UBL)",
  "syntax": "ubl",
  "fromPdf": false,
  "invoice": {
    "number": "RE-2026-0042",
    "issueDate": "2026-11-02",
    "currency": "EUR",
    "seller": {
      "name": "Muster IT GmbH",
      ...
    },
    "buyer": {
      "name": "Stadtverwaltung Musterstadt",
      ...
    },
    "lines": [
      /* 2 lines */
    ],
    "totals": {
      "taxExclusiveAmount": 1362.86,
      "taxAmount": 239.4,
      "payableAmount": 1602.26
    }
  },
  "validation": {
    "isValid": true,
```

```
    "errorCount": 0,
    "warningCount": 0,
    "issues": []
  }
}
```

2 · Generate an XRechnung from JSON

`POST` `/v1/convert?to=xrechnung-ubl` — line amounts, VAT breakdown and totals are calculated when you leave them out. The response is the XML; add `&report=true` to get JSON with the XML, the validation report and the official KoSIT result.

```
curl -X POST "https://invoicebridge-e-invoice-api.p.rapidapi.com/v1/convert?to=xrechnung-ubl" \
  -H "X-RapidAPI-Key: YOUR_RAPIDAPI_KEY" -H "X-RapidAPI-Host: invoicebridge-e-invoice-
api.p.rapidapi.com" \
  -H "Content-Type: application/json" --data-binary @invoice.json
```

invoice.json

```
{
  "number": "INV-1001",
  "issueDate": "2026-11-04",
  "dueDate": "2026-12-04",
  "currency": "EUR",
  "buyerReference": "991-12345-67",
  "seller": {
    "name": "Example Software GmbH",
    "vatId": "DE123456789",
    "electronicAddress": { "value": "billing@example.com", "scheme": "EM" },
    "address": { "line1": "Hauptstr. 1", "city": "Berlin", "postalCode": "10115", "countryCode":
"DE" },
    "contact": { "name": "Billing", "phone": "+49 30 000000", "email": "billing@example.com" }
  },
  "buyer": {
    "name": "Kunde AG",
    "electronicAddress": { "value": "ap@kunde.example", "scheme": "EM" },
    "address": { "line1": "Marktplatz 3", "city": "München", "postalCode": "80331",
"countryCode": "DE" }
  },
  "payment": { "meansCode": "58", "accounts": [{ "id": "DE02120300000000202051" }] },
  "lines": [
    { "name": "Consulting", "quantity": 8, "unitCode": "HUR", "netPrice": 120.00, "vatCategory":
"S", "vatRate": 19 }
  ]
}
```

Response (application/xml, headers `X-Validation-Errors: 0`, `X-Dropped-Data: 0`)

```
<?xml version="1.0" encoding="utf-8"?>
<Invoice xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComponents-2"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponents-2"
xmlns="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2">

<cbc:CustomizationID>urn:cen.eu:en16931:2017#compliant#urn:xeinkauf.de:kosit:xrechnung_3.0</cbc:
CustomizationID>
  <cbc:ProfileID>urn:fdc:peppol.eu:2017:poacc:billing:01:1.0</cbc:ProfileID>
  <cbc:ID>INV-1001</cbc:ID>
  <cbc:IssueDate>2026-11-04</cbc:IssueDate>
  ...
```

Totals calculated: 8 h × 120.00 = 960.00, VAT 19 % = 182.40, amount due 1,142.40 EUR. The official KoSIT validator accepts this invoice as XRechnung 3.0.

3 · Validate with the official rules

`POST` `/v1/validate` — the fast built-in check (`issues`) plus the official KoSIT validator with the XRechnung 3.0, EN 16931 and Peppol BIS Billing 3.0 rules (`official`). For JSON input add `?profile=xrechnung-ubl` (or another format id).

```
curl -X POST "https://invoicebridge-e-invoice-api.p.rapidapi.com/v1/validate" \
  -H "X-RapidAPI-Key: YOUR_RAPIDAPI_KEY" -H "X-RapidAPI-Host: invoicebridge-e-invoice-
api.p.rapidapi.com" \
  -H "Content-Type: application/xml" --data-binary @invoice.xml
```


```
{
  "isValid": true,
  "errorCount": 0,
  "warningCount": 0,
  "issues": [],
  "official": {
    "status": "accepted",
    "scenario": "EN16931 XRechnung (UBL Invoice)",
    "issues": [
      {
        "ruleId": "BR-DE-TMP-32",
        "severity": "information",
        "message": "An invoice should state the delivery date, an invoicing period or line
periods. (abridged)"
      }
    ]
  }
}
```


More endpoints

Endpoint	Does
`POST` `/v1/transform?to=peppol-bis`	XML or PDF in, another format out (e.g. ZUGFeRD → Peppol). `&report=true` adds the report.
`POST` `/v1/render`	Any supported e-invoice as a readable, printable HTML page.
`GET` `/v1/formats`	The format ids above.

Reading the report

- `issues[].ruleId` — the official rule id (`BR-…`, `BR-DE-…`, `PEPPOL-…`) when our check matches the official rule; `IB-…` for our own checks. `severity`: error, warning, information. `isValid` is false only for errors.
- `official.status` — `accepted`; `rejected` (official messages in `official.issues`); `noScenario` (not an XRechnung / EN 16931 / Peppol document); `unavailable` (only the fast check ran).
- `IB-DROPPED` — data in the source document the target cannot carry (e.g. a delivery period). Nothing is lost silently: each element is listed with its EN 16931 term and source path.
- Plain XML responses carry the counts as headers: `X-Validation-Errors`, `X-Dropped-Data`.

Good to know

- Up to 20 MB per document. Hybrid PDFs: Factur-X / ZUGFeRD 2.x; ZUGFeRD 1.0 is reported as not supported.
- Tested on the official KoSIT XRechnung test suite and the official FeRD ZUGFeRD samples.
- Documents are processed in memory and not stored.
- Prefer no code? The same engine runs as an Apify Actor for batch jobs: apify.com/bahs/e-invoice-reader-converter