Fatoora Organisation API — Integration Guide

Audience: organizations subscribed to a **DigiGO** (`digigo-starter`, `digigo-business`, `digigo-max`), **SEAL Max** (`seal-max`), or **Fatoora Trust** (`trust-starter`, `trust-business`, `trust-max`) plan who want to create, sign and track their own TEIF electronic invoices programmatically, using an API key they generate themselves from `/app/api-keys`.

Not covered here: the **Fatoora Partner API**, a different product line (`partner-starter`/`partner-business`/`partner-max`) for third-party integrators submitting invoices *on behalf of* other client organizations, which requires an extra org-discovery/authorization step this guide's audience doesn't need — see `PARTNER_API_GUIDE.md` instead if that's you. The single biggest structural difference: your API key is **already permanently bound to your own organization** — there is no `org_id` to resolve or pass on any request.

Production base URL: `https://business.fatoora.tn` — every endpoint in this guide (`/oauth/token`, `/api/v2/*`) is served from this host. Code samples below use `<base_url>` as a stand-in for this value.

Table of contents

1. How it works

Step	What happens	Where
1 — Create a key	From your Fatoora dashboard, generate an API key (`client_id` + `client_secret`) scoped to your own organization.	`/app/api-keys`
2 — Authenticate	Exchange your `client_id` / `client_secret` for a short-lived (1h) Bearer token via OAuth2 `client_credentials`. No `org_id` parameter — the key already identifies your organization.	§3
3 — Submit	Submit a TEIF invoice (JSON or XML) — validated synchronously against the TEIF 1.8.8 schema on submission.	§5
4 — Sign & file with TTN	Trigger sign-and-send (SEAL or DigiGO, depending on your plan) — Fatoora signs it and submits it to TTN (Tunisia Trade Net) asynchronously.	§7
5 — Track	Poll the job, list/read invoices, or register a webhook to be notified in real time as status changes.	§8, §9

Unlike the Partner API, there is no "acting for another organization" concept anywhere in this guide — every endpoint operates on your own organization's data, resolved automatically from your API key.

---

2. Getting your API key

1. Log into Fatoora with an account on a **DigiGO**, **SEAL Max**, or **Fatoora Trust** plan (`owner`/`admin` role required to manage keys).
2. Go to `/app/api-keys` → **New API key**.
3. Name the key, pick the scopes you need (§4), and create it. The `client_secret` is shown **once** at creation (and again via **Reveal secret**, owner role only) — store it securely, e.g. in your server's secret manager, never in client-side code.
4. The same page's **"Key Usage"** tab has an interactive, always-up-to-date version of this guide, with your real `client_id` pre-filled into every example and a **Download Postman collection** button that pre-fills your key.

Each plan caps how many keys you can create at once (`maxServiceApps` — 0 on starter tiers, higher on business/max tiers; unlimited on some). Deleting a key immediately invalidates any integration using it.

---

3. Authentication (OAuth2 client credentials)

```
curl -X POST "https://<base_url>/oauth/token" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "grant_type=client_credentials" \
  -d "client_id=YOUR_CLIENT_ID" \
  -d "client_secret=YOUR_CLIENT_SECRET"
```

Response:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiJ9....",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "invoice:write invoice:read ttn:submit"
}
```

Use `Authorization: Bearer <access_token>` on every subsequent call. Tokens expire after 1 hour (`expires_in`) — request a new one rather than trying to refresh; there is no refresh-token grant.

---

4. Scopes

Scope	Grants
`invoice:write`	Create, submit, sign-and-send, and delete invoices
`invoice:read`	List and read invoices, jobs, stats, webhooks
`ttn:submit`	Manually (re)submit an already-signed invoice to TTN, or sync its TTN status
`seal:sign`	Trigger SEAL signing — **only available on the `seal-max` plan**
`digigo:sign`	Trigger DigiGO signing — **only available on `digigo-*` plans**

Scopes are chosen per key at creation time (capped by what your plan allows) and are baked into every access token you obtain with that key — request a new key, or use a different one, if you need a scope combination you don't currently have.

---

5. Submitting invoices

TEIF (Tunisian Electronic Invoicing Format) is an XML-based standard, currently at **version 1.8.8**. Submit either structured JSON (Fatoora converts it to TEIF XML) or a pre-built TEIF XML document directly.

`POST /api/v2/invoices/submit` — JSON TEIF

Requires `invoice:write`.

```
{
  "version": "1.8.8",
  "controlingAgency": "TTN",
  "header": {
    "MessageSenderIdentifier":  { "@type": "I-01", "#text": "YOUR_TAX_ID" },
    "MessageRecieverIdentifier": { "@type": "I-01", "#text": "BUYER_TAX_ID" }
  },
  "body": {
    "bgm": { "documentIdentifier": "FACT-2026-000001", "documentTypeCode": "I-11", "documentType":
        "Facture" },
    "dtm": { "dateText": [{ "functionCode": "I-31", "format": "ddMMyy", "value": "150726" }] },
    "partnerSection": {
      "partnerDetails": [
        { "functionCode": "I-62", "nad": { "partnerIdentifier": { "type": "I-01", "value":
        "YOUR_TAX_ID" }, "partnerName": { "nameType": "Qualification", "value": "Ma Société SARL"
        },
          "partnerAdresses": [{ "adressDescription": "Siège social", "street": "Avenue Habib
        Bourguiba", "cityName": "Tunis", "postalCode": "1000", "country": { "codeList": "ISO_3166-
        1", "value": "TN" } }] },
          "ctaSection": [], "rffSection": [], "loc": [] },
        { "functionCode": "I-61", "nad": { "partnerIdentifier": { "type": "I-01", "value":
        "BUYER_TAX_ID" }, "partnerName": { "nameType": "Qualification", "value": "Mon Client SARL"
        },
          "partnerAdresses": [{ "adressDescription": "Siège social", "street": "Rue de Marseille",
        "cityName": "Sfax", "postalCode": "3000", "country": { "codeList": "ISO_3166-1", "value":
        "TN" } }] },
          "ctaSection": [], "rffSection": [], "loc": [] }
      ]
    },
    "linSection": { "lin": [{ "itemIdentifier": "1", "linImd": { "lang": "fr", "itemCode": "ART-
        001", "itemDescription": "Mon article" },
      "linQty": { "quantity": { "measurementUnit": "PCE", "value": "2" } },
      "linTax": { "taxTypeName": { "code": "I-1602", "value": "TVA" }, "taxDetails": { "taxRate":
        "19", "taxRateBasis": "Percentage" } },
      "linMoa": { "moaDetails": [{ "moa": { "amountTypeCode": "I-188", "currencyCodeList":
        "ISO_4217", "amount": { "currencyIdentifier": "TND", "value": "1000.000" } } }] },
      "subLin": [] }] },
    "invoiceTax": { "invoiceTaxDetails": [{ "tax": { "taxTypeName": { "code": "I-1602", "value":
        "TVA" }, "taxCategory": "I-1602", "taxDetails": { "taxRate": "19.000", "taxRateBasis":
        "Percentage" } },
      "amountDetails": [{ "moa": { "amountTypeCode": "I-177", "currencyCodeList": "ISO_4217",
        "amount": { "currencyIdentifier": "TND", "value": "1000.000" } } },
                        { "moa": { "amountTypeCode": "I-178", "currencyCodeList": "ISO_4217",
        "amount": { "currencyIdentifier": "TND", "value": "190.000" } } }] }] },
    "invoiceMoa": { "amountDetails": [
      { "moa": { "amountTypeCode": "I-172", "currencyCodeList": "ISO_4217", "amount": {
        "currencyIdentifier": "TND", "value": "1000.000" } } },
      { "moa": { "amountTypeCode": "I-181", "currencyCodeList": "ISO_4217", "amount": {
        "currencyIdentifier": "TND", "value": "195.000" } } },
      { "moa": { "amountTypeCode": "I-180", "currencyCodeList": "ISO_4217", "amount": {
        "currencyIdentifier": "TND", "value": "1195.000" } } },
      { "moa": { "amountTypeCode": "I-177", "currencyCodeList": "ISO_4217", "amount": {
        "currencyIdentifier": "TND", "value": "1000.000" } } },
```

```
      { "moa": { "amountTypeCode": "I-176", "currencyCodeList": "ISO_4217", "amount": {
        "currencyIdentifier": "TND", "value": "1000.000" } } }

    ] },
    "pyt": { "pytSectionDetails": [{ "pytPai": { "paiConditionCode": "I-122", "paiMeansCode": "42"
        } }] }

  }

}
```

Response — `201 Created` — the full invoice object:


```
{
  "id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "orgId": "your-org-uuid",
  "invoiceNumber": "FACT-2026-000001",
  "invoiceTypeCode": "I-11",
  "invoiceDate": "2026-07-15",
  "status": "VALIDATED",
  "senderName": "Ma Société SARL",
  "receiverName": "Mon Client SARL",
  "totalAmountExclTax": 1000,
  "totalTaxAmount": null,
  "totalAmountInclTax": null,
  "currencyCode": "TND",
  "ttnGeneratedRef": null,
  "createdAt": "2026-07-15T10:00:00Z",
  "submittedAt": "2026-07-15T10:00:00Z",
  "signedAt": null,
  "createdBy": "YOUR_CLIENT_ID",
  "direction": "SENT"
}
```


> **Status starts at `VALIDATED`, not `DRAFT`.** Submission runs synchronous TEIF validation — a successful `201` already reflects a validated invoice, ready for sign-and-send.

`POST /api/v2/invoices/submit-xml` — raw TEIF XML

Same auth/scope, same response shape. `Content-Type: application/xml`, body is the raw TEIF document — see §13 for a full minimal example.

Document type codes (`documentTypeCode` in `bgm`)

Code	Meaning
`I-11`	Invoice (Facture)
`I-12`	Credit note (Avoir) — requires a `BillingReference` linking the original invoice
`I-13`	Proforma invoice
`I-14`	Advance invoice
`I-15`	Balance invoice
`I-16`	Other

---

6. Validating a TEIF document

`POST /api/v2/teif/validate`

Runs XSD 1.8.8 schema validation against a TEIF XML document **without creating an invoice** — useful as a pre-flight check while building your integration. Requires `invoice:read` or `invoice:write`. Body: `{ "xml": "<TEIF>...</TEIF>" }` (field is `xml`).

```
{ "valid": true, "issues": [] }
```

or, on failure:

```
{
  "valid": false,
  "issues": [
    { "type": "XML_SCHEMA_VIOLATION", "message": "...", "severity": "ERROR", "xpath": "line:1" }
  ]
}
```

---

7. Signing and TTN submission

Requires `invoice:write`, plus `seal:sign` or `digigo:sign` depending on your plan and the provider you pick.

`POST /api/v2/invoices/:id/sign-and-send` — recommended, one call

```
{ "signatureType": "SEAL" }
```

(or `"DIGIGO"`, matching your plan). Response — `200 OK` (job created, not the final result):

```
{
  "jobId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "invoiceId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "jobType": "SIGN_AND_SUBMIT_SEAL",
  "status": "PENDING",
  "success": true,
  "message": "Invoice queued for signature and TTN submission"
}
```

Poll until it completes:

```
GET /api/v2/jobs/:jobId
GET /api/v2/jobs/:jobId/logs?limit=50
```

`status` transitions `PENDING` → `COMPLETED` or `FAILED`. On failure, `outputData` names the failing step (e.g. a missing signing certificate configuration — a common real-world issue, not a bug). Once complete, re-fetch the invoice — `status` will be `ACCEPTED_TTN` on success.

Allowed source statuses: `DRAFT`, `VALIDATED`, `VALIDATION_FAILED`, `SIGNATURE_FAILED`, `REJECTED_TTN`, `ERROR_TTN`, `TIMEOUT_TTN` — otherwise `409 Conflict`.

Manual two-step alternative

If you need to inspect the signed document before sending it to TTN:

```
POST /api/v2/invoices/:id/sign          { "provider": "seal" }   # or "digigo"
POST /api/v2/invoices/:id/submit-ttn
```

Unlike the Partner API, **your own organization's key is allowed to call `submit-ttn` directly** — that restriction only applies to partner (M2M-on-behalf-of) tokens.

Batch

```
POST /api/v2/invoices/batch/sign-and-send   { "invoiceIds": [...], "signatureType": "SEAL" }
POST /api/v2/invoices/batch/delete          [ "inv_1", "inv_2" ]
```

Signing quota

Each sign-and-send consumes one unit of your monthly signing quota (or, once exhausted, one signature-pack credit):

```
{ "error": "NO_SIGNATURE_CREDITS", "message": "No signing quota or signature-pack credits
        available. Buy a pack or subscribe/upgrade to sign and submit to TTN.", "packsRemaining":
        0 }
```

with HTTP `402`.

---

8. Listing, retrieving and the invoice artifacts (XML / PDF)

All require `invoice:read`.

Method	Path	Notes
GET	`/api/v2/invoices`	Paginated list, filterable (`direction`, `status`, dates)
GET	`/api/v2/invoices/:id`	Full invoice detail — **wrapped response**, see below
GET	`/api/v2/invoices/:id/pdf`	Binary PDF, any status (drafts included). Optional query: `pdfTemplate`, `lang`
GET	`/api/v2/invoices/:id/generated-xml`	Generated TEIF XML — **see warning below**
GET	`/api/v2/invoices/stats`	Dashboard-style aggregate stats
GET	`/api/v2/invoices/api-key-stats`	Usage stats broken down per API key (`invoicesCreated` / `invoicesSigned` / `invoicesSubmittedToTtn`)

`GET /api/v2/invoices/:id` — full detail (wrapped)

```
{
  "invoice": { "id": "...", "status": "ACCEPTED_TTN", "...": "..." },
  "teifJson": "{\"body\":{...},\"header\":{...},\"version\":\"1.8.8\"}",
  "teifXml": "<TEIF controlingAgency=\"TTN\" version=\"1.8.8\">...</TEIF>",
  "ttnQrCodeBase64": null,
  "localSignedXml": null,
  "validationErrors": null,
  "ttnAcknowledgments": null
}
```

`teifJson` / `teifXml` are already available here — no separate call to `generated-xml` needed for the common case.

Downloading the PDF

```
curl -X GET "https://<base_url>/api/v2/invoices/INVOICE_ID/pdf" \
  -H "Authorization: Bearer ACCESS_TOKEN" \
  -o facture.pdf
```

⚠️ `generated-xml` is not always the signed/final document

Internally, XML generation only returns the *stored* `teifXml` verbatim while it is still unsigned. The moment an invoice has been signed (and especially once TTN has accepted it and stamped a QR code), `GET /invoices/:id/generated-xml` instead **regenerates a fresh, unsigned XML on the fly from `teifJson`** — silently dropping the signature and the TTN QR code. **For the actual final artifact, read the `teifXml` field from `GET /invoices/:id` instead** — that field is the real persisted snapshot, not a live regeneration.

```
curl -X GET "https://<base_url>/api/v2/invoices/INVOICE_ID/generated-xml" \
  -H "Authorization: Bearer ACCESS_TOKEN"
```

Per-API-key usage stats

```
{
  "organizationId": "your-org-uuid",
  "totalInvoices": 42,
  "stats": [
    { "appId": "YOUR_CLIENT_ID", "invoicesCreated": 42, "invoicesSigned": 40,
        "invoicesSubmittedToTtn": 38 }
  ]
}
```

The same data is also visible in the **Audit / stats** card on `/app/api-keys`.

---

9. Webhooks

Instead of polling, register a webhook to receive a push notification whenever one of your invoices changes status. Uses the same `access_token` as every other endpoint — no extra setup beyond having `invoice:write` / `invoice:read`.

Method	Path	Scope
POST	`/api/v2/webhooks`	`invoice:write`
GET	`/api/v2/webhooks`	`invoice:read`
POST	`/api/v2/webhooks/:id/test`	`invoice:write`
DELETE	`/api/v2/webhooks/:id`	`invoice:write`

Create a subscription

```
curl -X POST "https://<base_url>/api/v2/webhooks" \
  -H "Authorization: Bearer ACCESS_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "url": "https://your-erp.example.com/webhooks/fatoora",
    "events": ["invoice.validated", "invoice.signed", "invoice.accepted_ttn",
        "invoice.rejected_ttn"]
  }'
```

```
{
  "id": "wh_xxxxxxxxxxxxxxx",
  "appId": "YOUR_CLIENT_ID",
  "url": "https://your-erp.example.com/webhooks/fatoora",
  "events": ["invoice.validated", "invoice.signed", "invoice.accepted_ttn",
        "invoice.rejected_ttn"],
  "status": "ACTIVE",
  "secret": "whsec_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "expiresAt": "2026-08-14T00:00:00",
  "createdAt": "2026-07-15T00:00:00"
}
```

`secret` is returned **once**, at creation — save it to verify delivered signatures. Default events when `events` is omitted: `invoice.created`, `invoice.validated`, `invoice.signed`, `invoice.submitted_ttn`, `invoice.accepted_ttn`, `invoice.rejected_ttn`, `webhook.test`. Add `invoice.validation_failed` explicitly if you also need it.

Verifying the delivered payload

Every delivery includes `X-Webhook-Id`, `X-Event-Id`, `X-Timestamp` (epoch ms) and `X-HMAC-Signature` (Base64, `HmacSHA256`) headers:

```
canonical = "POST:" + path + ":" + sha256Hex(body) + ":" + timestamp
X-HMAC-Signature = base64(HMAC-SHA256(canonical, webhook_secret))
```

Example delivered payload:

```
{
  "eventId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "eventType": "invoice.accepted_ttn",
  "occurredAt": "2026-07-15T10:00:00Z",
  "appId": "YOUR_CLIENT_ID",
  "resource": { "type": "invoice", "id": "inv_xxxxxxxxxxxxxxx" },
  "data": { "invoiceId": "inv_xxxxxxxxxxxxxxx", "invoiceStatus": "ACCEPTED_TTN", "invoiceNumber":
        "FACT-2026-000001" },
  "customData": {}
}
```

Delivery is single-attempt, fire-and-forget — no automatic retry. Use `POST /webhooks/:id/test` (fires `webhook.test`) to validate your receiver, signature verification included, before relying on real events.

Alternative: polling / pull notifications

If you'd rather not run a public HTTP endpoint, switch to `POLLING` mode (`PUT /api/v2/event-delivery/mode`) and consume `GET /api/v2/notifications` + `POST /api/v2/notifications/ack` instead. The two modes are mutually exclusive — switching requires no active webhooks (or no unacknowledged pending notifications, in the other direction).

---

10. Quotations

The same surface supports draft quotations (devis), convertible into invoices:

Method	Path	Scope
POST	`/api/v2/quotations/draft`	`invoice:write`
GET	`/api/v2/quotations`	`invoice:read`
GET	`/api/v2/quotations/:id`	`invoice:read`
PUT	`/api/v2/quotations/:id`	`invoice:write` (draft only)
PATCH	`/api/v2/quotations/:id/status`	`invoice:write` — body `{ "status": "ACCEPTED" | "REFUSED" | "CANCELLED" }`
POST	`/api/v2/quotations/:id/convert-to-invoice`	`invoice:write`

All return the same full invoice object described in §5, with `isQuotation` / `quotationStatus` set accordingly. `POST /draft` starts at `status: "DRAFT"`, `quotationStatus: "CREATED"` — quotations skip validation until conversion.

---

11. Error handling

OAuth errors (`/oauth/token`)

```
{ "error": "invalid_client", "error_description": "Invalid client credentials." }
```

Core-proxy / BFF errors

```
{ "error": "INSUFFICIENT_SCOPES", "message": "Required scopes: invoice:write", "requiredScopes":
        ["invoice:write"] }
```

Common error codes

Code	HTTP	Meaning / fix
`invalid_client`	401	Wrong `client_id` / `client_secret`
No token provided	401	Missing `Authorization` header
Invalid or expired token	401	Request a new token via `POST /oauth/token`
`INSUFFICIENT_SCOPES`	401/403	Your key lacks the required scope — create a new key with the right scopes, or edit an existing one's scopes if your plan allows it
`NO_ORGANIZATION`	400	Malformed/corrupted token — regenerate your `client_secret`
`NO_SERVICE_APP`	404	No active API access for your organization — check your DigiGO/SEAL/Trust subscription is active
`VALIDATION_ERROR` / `INVOICE_VALIDATION_ERROR`	400/422	TEIF payload fails schema/business validation — check `details[]`
`NO_SIGNATURE_CREDITS`	402	Your signing quota and signature packs are exhausted
`RATE_LIMIT_EXCEEDED` / `TOO_MANY_REQUESTS`	429	Back off and retry (see §12)

12. Rate limits and quotas

Limiter	Scope	Limit
`POST /oauth/token`	per IP	30 requests / 15 min
All other `/api/*` calls	per IP	300 requests / 15 min (configurable server-side)

Rate-limit responses include a `Retry-After` header and:

```
{ "error": "RATE_LIMIT_EXCEEDED", "message": "Too many requests, please try again later.",
        "retryAfter": 900 }
```

Plan-level quotas (independent of rate limits): your subscription's `monthlyDocLimit` caps total invoices per calendar month (higher on business/max tiers, unlimited on some); signing specifically also draws down your **signing quota** (subscription-based, or signature-pack credits as overage) — see §7. During a trial period, every plan gets a fixed cap regardless of its normal monthly limit.

---

13. Code examples

Get a token, then submit and sign an invoice (JavaScript)

```javascript
// 1. Get the token — no org_id, your key is already org-bound
const tokenRes = await fetch('https://<base_url>/oauth/token', {
  method: 'POST',
  headers: { 'Content-Type': 'application/x-www-form-urlencoded' },
  body: new URLSearchParams({
    grant_type: 'client_credentials',
    client_id: 'YOUR_CLIENT_ID',
    client_secret: 'YOUR_CLIENT_SECRET',
  }),
});
const { access_token } = await tokenRes.json();

// 2. Submit the invoice
const submitRes = await fetch('https://<base_url>/api/v2/invoices/submit', {
  method: 'POST',
  headers: { Authorization: `Bearer ${access_token}`, 'Content-Type': 'application/json' },
  body: JSON.stringify(invoice), // your TEIF object, see §5
});
const { id: invoiceId } = await submitRes.json();

// 3. Sign and send to TTN (async job)
const signRes = await fetch(`https://<base_url>/api/v2/invoices/${invoiceId}/sign-and-send`, {
  method: 'POST',
  headers: { Authorization: `Bearer ${access_token}`, 'Content-Type': 'application/json' },
  body: JSON.stringify({ signatureType: 'SEAL' }),
});
const { jobId } = await signRes.json();

// 4. Poll until COMPLETED or FAILED
const jobRes = await fetch(`https://<base_url>/api/v2/jobs/${jobId}`, {
  headers: { Authorization: `Bearer ${access_token}` },
});
console.log((await jobRes.json()).status);
```

Python

```python
import requests

BASE = "https://<base_url>"

token = requests.post(f"{BASE}/oauth/token", data={
    "grant_type": "client_credentials",
    "client_id": "YOUR_CLIENT_ID",
    "client_secret": "YOUR_CLIENT_SECRET",
}).json()["access_token"]

headers = {"Authorization": f"Bearer {token}"}
res = requests.post(f"{BASE}/api/v2/invoices/submit", headers=headers, json=invoice)
print(res.json()["id"], res.json()["status"])
```

Minimal TEIF XML submission

```bash
curl -X POST "https://<base_url>/api/v2/invoices/submit-xml" \
  -H "Authorization: Bearer ACCESS_TOKEN" \
  -H "Content-Type: application/xml" \
  --data-binary @invoice.xml
```

```xml
<TEIF controlingAgency="TTN" version="1.8.8">
  <InvoiceHeader>
    <MessageSenderIdentifier type="I-01">YOUR_TAX_ID</MessageSenderIdentifier>
    <MessageRecieverIdentifier type="I-01">BUYER_TAX_ID</MessageRecieverIdentifier>
  </InvoiceHeader>
  <InvoiceBody>
    <Bgm>
      <DocumentIdentifier>FACT-2026-000001</DocumentIdentifier>
      <DocumentType code="I-11">Facture</DocumentType>
    </Bgm>
    <Dtm>
      <DateText format="ddMMyy" functionCode="I-31">150726</DateText>
    </Dtm>
    <PartnerSection>
      <PartnerDetails functionCode="I-62">
        <Nad>
          <PartnerIdentifier type="I-01">YOUR_TAX_ID</PartnerIdentifier>
          <PartnerName nameType="Qualification">Ma Société SARL</PartnerName>
          <PartnerAdresses lang="fr">
            <Street>Avenue Habib Bourguiba</Street>
            <CityName>Tunis</CityName>
            <PostalCode>1000</PostalCode>
            <Country codeList="ISO_3166-1">TN</Country>
          </PartnerAdresses>
        </Nad>
      </PartnerDetails>
      <PartnerDetails functionCode="I-61">
        <Nad>
          <PartnerIdentifier type="I-01">BUYER_TAX_ID</PartnerIdentifier>
          <PartnerName nameType="Qualification">Mon Client SARL</PartnerName>
          <PartnerAdresses lang="fr">
            <Street>Rue de Marseille</Street>
            <CityName>Sfax</CityName>
            <PostalCode>3000</PostalCode>
            <Country codeList="ISO_3166-1">TN</Country>
          </PartnerAdresses>
        </Nad>
      </PartnerDetails>
    </PartnerSection>
```

```
    <LinSection>
      <Lin>
        <ItemIdentifier>1</ItemIdentifier>
        <LinImd lang="fr">
          <ItemCode>ART-001</ItemCode>
          <ItemDescription>Mon article</ItemDescription>
        </LinImd>
        <LinQty><Quantity measurementUnit="PCE">2</Quantity></LinQty>
        <LinTax>
          <TaxTypeName code="I-1602">TVA</TaxTypeName>
          <TaxDetails><TaxRate>19</TaxRate></TaxDetails>
        </LinTax>
        <LinMoa>
          <MoaDetails>
            <Moa amountTypeCode="I-188" currencyCodeList="ISO_4217">
              <Amount currencyIdentifier="TND">1000.000</Amount>
            </Moa>
          </MoaDetails>
        </LinMoa>
      </Lin>
    </LinSection>
    <InvoiceMoa>
      <AmountDetails><Moa amountTypeCode="I-172" currencyCodeList="ISO_4217"><Amount
        currencyIdentifier="TND">1000.000</Amount></Moa></AmountDetails>
      <AmountDetails><Moa amountTypeCode="I-181" currencyCodeList="ISO_4217"><Amount
        currencyIdentifier="TND">195.000</Amount></Moa></AmountDetails>
      <AmountDetails><Moa amountTypeCode="I-180" currencyCodeList="ISO_4217"><Amount
        currencyIdentifier="TND">1195.000</Amount></Moa></AmountDetails>
    </InvoiceMoa>
    <InvoiceTax>
      <InvoiceTaxDetails>
        <Tax>
          <TaxTypeName code="I-1602">TVA</TaxTypeName>
          <TaxDetails><TaxRate>19</TaxRate></TaxDetails>
        </Tax>
        <AmountDetails><Moa amountTypeCode="I-178" currencyCodeList="ISO_4217"><Amount
        currencyIdentifier="TND">195.000</Amount></Moa></AmountDetails>
      </InvoiceTaxDetails>
    </InvoiceTax>
  </InvoiceBody>
</TEIF>
```

14. Appendix — reference tables

Invoice status values

`DRAFT`, `SUBMITTED`, `VALIDATED`, `VALIDATION_FAILED`, `SIGNED`, `SIGNATURE_FAILED`, `SENT_TTN`, `ACCEPTED_TTN`, `REJECTED_TTN`, `ERROR_TTN`, `TIMEOUT_TTN`, `CANCELLED`, `ARCHIVED`

MOA amount type codes used in the examples above

Code	Meaning
`I-172`	Total excl. tax
`I-181`	Total taxes
`I-180`	Total incl. tax (amount due)
`I-177`	Taxable base
`I-176`	Invoice total excl. tax
`I-188`	Line net amount

Default webhook event types

`invoice.created`, `invoice.validated`, `invoice.signed`, `invoice.submitted_ttn`, `invoice.accepted_ttn`, `invoice.rejected_ttn`, `webhook.test` (plus opt-in `invoice.validation_failed`).

---

15. OpenAPI spec and Postman collection

Also saved as standalone files: `organisation-api-openapi.yaml` 📎 `organisation-api-postman-collection.json`.

The Postman collection JSON below is the literal output of `buildApiKeysPostmanCollection()` in `fatooraLive/src/components/api-keys/ApiKeysUsageGuide.tsx` (extracted and executed with Node, same process used for the Partner API guide), generated with placeholder credentials and the production `base_url` — replace `client_id` / `client_secret` with your own, or download your pre-filled copy from the **Key Usage** tab of `/app/api-keys`.

```
{
  "info": {
    "name": "Fatoora API Keys — DigiGO / SEAL / Trust",
    "description": "Collection pour les clients disposant d'une clé API (DigiGO, SEAL Max, Fatoora
        Trust) : créer, valider, signer et envoyer des factures TEIF à TTN.",
    "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json"
  },
  "variable": [
    {
      "key": "base_url",
      "value": "https://business.fatoora.tn",
      "type": "string"
    },
    {
      "key": "client_id",
      "value": "svc_org_sample00000000",
      "type": "string"
    },
    {
      "key": "client_secret",
      "value": "YOUR_CLIENT_SECRET_HERE",
      "type": "string"
    },
    {
      "key": "access_token",
      "value": "",
      "type": "string"
    },
    {
      "key": "invoice_id",
      "value": "",
      "type": "string"
    },
    {
      "key": "job_id",
      "value": "",
      "type": "string"
    },
    {
      "key": "seller_tax_id",
```

```
      "value": "1234567ABC000",
      "type": "string"
    },
    {
      "key": "buyer_tax_id",
      "value": "1234567RAM000",
      "type": "string"
    },
    {
      "key": "seller_name",
      "value": "Ma Société SARL",
      "type": "string"
    },
    {
      "key": "buyer_name",
      "value": "Mon Client SARL",
      "type": "string"
    },
    {
      "key": "seller_street",
      "value": "Avenue Habib Bourguiba",
      "type": "string"
    },
    {
      "key": "seller_city",
      "value": "Tunis",
      "type": "string"
    },
    {
      "key": "seller_postal_code",
      "value": "1000",
      "type": "string"
    },
    {
      "key": "buyer_street",
      "value": "Rue de Marseille",
      "type": "string"
    },
    {
      "key": "buyer_city",
```

```
      "value": "Sfax",
      "type": "string"
    },
    {
      "key": "buyer_postal_code",
      "value": "3000",
      "type": "string"
    },
    {
      "key": "invoice_number",
      "value": "FACT-2026-000001",
      "type": "string"
    },
    {
      "key": "invoice_date_ddMMyy",
      "value": "150726",
      "type": "string"
    },
    {
      "key": "line1_item_code",
      "value": "ART-001",
      "type": "string"
    },
    {
      "key": "line1_description",
      "value": "Mon article",
      "type": "string"
    },
    {
      "key": "line1_qty",
      "value": "2",
      "type": "string"
    },
    {
      "key": "line1_ht",
      "value": "1000.000",
      "type": "string"
    },
    {
      "key": "total_ht",
```

```
      "value": "1000.000",
      "type": "string"
    },
    {
      "key": "total_tva",
      "value": "190.000",
      "type": "string"
    },
    {
      "key": "total_taxes",
      "value": "195.000",
      "type": "string"
    },
    {
      "key": "total_ttc",
      "value": "1195.000",
      "type": "string"
    },
    {
      "key": "payment_condition_code",
      "value": "I-122",
      "type": "string"
    }
  ],
  "item": [
    {
      "name": "🔐 Authentification",
      "item": [
        {
          "name": "Obtenir un token d'accès",
          "event": [
            {
              "listen": "test",
              "script": {
                "type": "text/javascript",
                "exec": [
                  "const json = pm.response.json();",
                  "if (json.access_token) {",
                  "  pm.collectionVariables.set('access_token', json.access_token);",
                  "  console.log(\"✅ access_token sauvegardé, expire dans\" + json.expires_in +
      's');",
```

```
            "} else {",
            "  console.error(\"❌ Pas de access_token :\", JSON.stringify(json));",
            "}"
          ]
        }
      }
    ],
    "request": {
      "method": "POST",
      "header": [
        {
          "key": "Content-Type",
          "value": "application/x-www-form-urlencoded"
        }
      ],
      "body": {
        "mode": "urlencoded",
        "urlencoded": [
          {
            "key": "grant_type",
            "value": "client_credentials"
          },
          {
            "key": "client_id",
            "value": "{{client_id}}"
          },
          {
            "key": "client_secret",
            "value": "{{client_secret}}"
          }
        ]
      },
      "url": {
        "raw": "{{base_url}}/oauth/token",
        "host": [
          "{{base_url}}"
        ],
        "path": [
          "oauth",
          "token"
```

```
            ]
          },
          "description": "Votre clé API est déjà liée à votre organisation — aucun `org_id` n'est requis. Le script de test sauvegarde automatiquement le token dans `{{access_token}}`."
        },
        "response": [
          {
            "name": "200 OK",
            "originalRequest": {
              "method": "POST",
              "header": [],
              "url": {
                "raw": "{{base_url}}/oauth/token",
                "host": [
                  "{{base_url}}/oauth/token"
                ]
              }
            },
            "status": "OK",
            "code": 200,
            "_postman_previewlanguage": "json",
            "header": [
              {
                "key": "Content-Type",
                "value": "application/json"
              }
            ],
            "cookie": [],
            "body": "{\n  \"access_token\": \"eyJhbGciOiJIUzI1NiJ9....\",\n  \"token_type\": \"Bearer\",\n  \"expires_in\": 3600,\n  \"scope\": \"invoice:write invoice:read ttn:submit\"\n}"
          },
          {
            "name": "401 Unauthorized",
            "originalRequest": {
              "method": "POST",
              "header": [],
              "url": {
                "raw": "{{base_url}}/oauth/token",
                "host": [
                  "{{base_url}}/oauth/token"
```

```
            ]
          }
        },
        "status": "Unauthorized",
        "code": 401,
        "_postman_previewlanguage": "json",
        "header": [
          {
            "key": "Content-Type",
            "value": "application/json"
          }
        ],
        "cookie": [],
        "body": "{\n  \"error\": \"invalid_client\",\n  \"error_description\": \"Invalid
    client credentials.\"\n}"
      }
    ]
  }
  ]
},
{
  "name": "📄 Factures — JSON TEIF",
  "item": [
    {
      "name": "Soumettre une facture (JSON)",
      "event": [
        {
          "listen": "test",
          "script": {
            "type": "text/javascript",
            "exec": [
              "if (pm.response.code === 201 || pm.response.code === 200) {",
              "  const json = pm.response.json();",
              "  if (json.id) {",
              "    pm.collectionVariables.set('invoice_id', json.id);",
              "    console.log(\"✅ invoice_id sauvegardé :\", json.id);",
              "  }",
              "} else {",
              "  console.error(\"❌ Facture non créée :\", pm.response.text());",
              "}"
            ]
```

```
        }
      }
    ],
    "request": {
      "method": "POST",
      "header": [
        {
          "key": "Content-Type",
          "value": "application/json"
        },
        {
          "key": "Authorization",
          "value": "Bearer {{access_token}}"
        }
      ],
      "body": {
        "mode": "raw",
```

```
        "raw": "{\n  \"version\": \"1.8.8\",\n  \"controlingAgency\": \"TTN\",\n
\"header\": {\n    \"MessageSenderIdentifier\": {\n      \"@type\": \"I-01\",\n
\"#text\": \"{{seller_tax_id}}\"\n    },\n    \"MessageRecieverIdentifier\": {\n
\"@type\": \"I-01\",\n      \"#text\": \"{{buyer_tax_id}}\"\n    }\n  },\n  \"body\": {\n
\"bgm\": {\n      \"documentIdentifier\": \"{{invoice_number}}\",\n
\"documentTypeCode\": \"I-11\",\n      \"documentType\": \"Facture\"\n    },\n    \"dtm\":
{\n      \"dateText\": [\n        {\n          \"functionCode\": \"I-31\",\n
\"format\": \"ddMMyy\",\n          \"value\": \"{{invoice_date_ddMMyy}}\"\n        }\n
]\n    },\n    \"partnerSection\": {\n      \"partnerDetails\": [\n        {\n
\"functionCode\": \"I-62\",\n          \"nad\": {\n            \"partnerIdentifier\": {\n
\"type\": \"I-01\",\n              \"value\": \"{{seller_tax_id}}\"\n            },\n
\"partnerName\": {\n              \"nameType\": \"Qualification\",\n
\"value\": \"{{seller_name}}\"\n            },\n            \"partnerAdresses\": [\n
{\n                \"adressDescription\": \"Siège social\",\n                \"street\":
\"{{seller_street}}\",\n                \"cityName\": \"{{seller_city}}\",\n
\"postalCode\": \"{{seller_postal_code}}\",\n                \"country\": {\n
\"codeList\": \"ISO_3166-1\",\n                \"value\": \"TN\"\n              }\n
}\n            ]\n          },\n          \"ctaSection\": [],\n          \"rffSection\":
[],\n          \"loc\": []\n        },\n        {\n          \"functionCode\": \"I-61\",\n
\"nad\": {\n            \"partnerIdentifier\": {\n              \"type\": \"I-01\",\n
\"value\": \"{{buyer_tax_id}}\"\n            },\n            \"partnerName\": {\n
\"nameType\": \"Qualification\",\n              \"value\": \"{{buyer_name}}\"\n
},\n            \"partnerAdresses\": [\n              {\n
\"adressDescription\": \"Siège social\",\n                \"street\": \"
{{buyer_street}}\",\n                \"cityName\": \"{{buyer_city}}\",\n
\"postalCode\": \"{{buyer_postal_code}}\",\n                \"country\": {\n
\"codeList\": \"ISO_3166-1\",\n                \"value\": \"TN\"\n              }\n
}\n            ]\n          },\n          \"ctaSection\": [],\n          \"rffSection\":
[],\n          \"loc\": []\n        }\n      ]\n    },\n    \"linSection\": {\n
\"lin\": [\n        {\n          \"itemIdentifier\": \"1\",\n          \"linImd\": {\n
\"lang\": \"fr\",\n            \"itemCode\": \"{{line1_item_code}}\",\n
\"itemDescription\": \"{{line1_description}}\"\n          },\n          \"linQty\": {\n
\"quantity\": {\n              \"measurementUnit\": \"PCE\",\n              \"value\": \"
{{line1_qty}}\"\n            }\n          },\n          \"linTax\": {\n
\"taxTypeName\": {\n              \"code\": \"I-1602\",\n              \"value\":
\"TVA\"\n            },\n            \"taxDetails\": {\n              \"taxRate\":
\"19\",\n              \"taxRateBasis\": \"Percentage\"\n            }\n          },\n
\"linMoa\": {\n            \"moaDetails\": [\n              {\n                \"moa\":
{\n                  \"amountTypeCode\": \"I-188\",\n
\"currencyCodeList\": \"ISO_4217\",\n                  \"amount\": {\n
\"currencyIdentifier\": \"TND\",\n                  \"value\": \"{{line1_ht}}\"\n
}\n                }\n              }\n            ]\n          },\n          \"subLin\":
[]\n        }\n      ]\n    },\n    \"invoiceTax\": {\n      \"invoiceTaxDetails\": [\n
{\n          \"tax\": {\n            \"taxTypeName\": {\n              \"code\": \"I-
1602\",\n              \"value\": \"TVA\"\n            },\n            \"taxCategory\":
\"I-1602\",\n            \"taxDetails\": {\n              \"taxRate\": \"19.000\",\n
\"taxRateBasis\": \"Percentage\"\n            }\n          },\n
\"amountDetails\": [\n            {\n              \"moa\": {\n
\"amountTypeCode\": \"I-177\",\n                \"currencyCodeList\": \"ISO_4217\",\n
\"amount\": {\n                  \"currencyIdentifier\": \"TND\",\n
\"value\": \"{{total_ht}}\"\n                }\n              }\n            },\n
{\n              \"moa\": {\n                \"amountTypeCode\": \"I-178\",\n
\"currencyCodeList\": \"ISO_4217\",\n                \"amount\": {\n
\"currencyIdentifier\": \"TND\",\n                \"value\": \"{{total_tva}}\"\n
}\n              }\n            }\n          ]\n        }\n      ]\n    },\n
\"invoiceMoa\": {\n      \"amountDetails\": [\n        {\n          \"moa\": {\n
\"amountTypeCode\": \"I-172\",\n          \"currencyCodeList\": \"ISO_4217\",\n
\"amount\": {\n            \"currencyIdentifier\": \"TND\",\n            \"value\": \"
{{total_ht}}\"\n          }\n        },\n        {\n          \"moa\": {\n
\"amountTypeCode\": \"I-181\",\n          \"currencyCodeList\": \"ISO_4217\",\n
\"amount\": {\n            \"currencyIdentifier\": \"TND\",\n            \"value\": \"
{{total_taxes}}\"\n          }\n        },\n        {\n          \"moa\":
{\n            \"amountTypeCode\": \"I-180\",\n            \"currencyCodeList\":
\"ISO_4217\",\n            \"amount\": {\n              \"currencyIdentifier\": \"TND\",\n
\"value\": \"{{total_ttc}}\"\n            }\n          }\n        },\n        {\n
\"moa\": {\n            \"amountTypeCode\": \"I-177\",\n            \"currencyCodeList\":
\"ISO_4217\",\n            \"amount\": {\n              \"currencyIdentifier\": \"TND\",\n
\"value\": \"{{total_ht}}\"\n            }\n          }\n        },\n        {\n
\"moa\": {\n            \"amountTypeCode\": \"I-176\",\n            \"currencyCodeList\":
\"ISO_4217\",\n            \"amount\": {\n              \"currencyIdentifier\": \"TND\",\n
\"value\": \"{{total_ht}}\"\n            }\n          }\n        }\n      ]\n    },\n
\"pyt\": {\n      \"pytSectionDetails\": [\n        {\n          \"pytPai\": {\n
\"paiConditionCode\": \"{{payment_condition_code}}\",\n            \"paiMeansCode\":
\"42\"\n          }\n        }\n      ]\n    }\n  },\n  \"additionnalDocuments\": null,\n
\"refTtnVal\": null,\n  \"signatures\": []\n}",
```

```
      "options": {
        "raw": {
          "language": "json"
        }
      }
    },
    "url": {
      "raw": "{{base_url}}/api/v2/invoices/submit",
      "host": [
        "{{base_url}}"
      ],
      "path": [
        "api",
        "v2",
        "invoices",
        "submit"
      ]
    },
    "description": "Scope requis : `invoice:write`."
  },
  "response": [
    {
      "name": "201 Created",
      "originalRequest": {
        "method": "POST",
        "header": [],
        "url": {
          "raw": "{{base_url}}/api/v2/invoices/submit",
          "host": [
            "{{base_url}}/api/v2/invoices/submit"
          ]
        }
      },
      "status": "Created",
      "code": 201,
      "_postman_previewlanguage": "json",
      "header": [
        {
          "key": "Content-Type",
          "value": "application/json"
```

```
            }
          ],
          "cookie": [],
          "body": "{\n  \"id\": \"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\",\n
        \"invoiceNumber\": \"{{invoice_number}}\",\n  \"status\": \"VALIDATED\",\n  \"direction\":
        \"SENT\"\n}"
        }
      ]
    },
    {
      "name": "Lister les factures",
      "request": {
        "method": "GET",
        "header": [
          {
            "key": "Authorization",
            "value": "Bearer {{access_token}}"
          }
        ],
        "url": {
          "raw": "{{base_url}}/api/v2/invoices?page=0&pageSize=20&direction=SENT",
          "host": [
            "{{base_url}}"
          ],
          "path": [
            "api",
            "v2",
            "invoices"
          ],
          "query": [
            {
              "key": "page",
              "value": "0"
            },
            {
              "key": "pageSize",
              "value": "20"
            },
            {
              "key": "direction",
              "value": "SENT"
```

```
            }
          ]
        },
        "description": "Scope requis : `invoice:read`."
      },
      "response": [
        {
          "name": "200 OK",
          "originalRequest": {
            "method": "GET",
            "header": [],
            "url": {
              "raw": "{{base_url}}/api/v2/invoices",
              "host": [
                "{{base_url}}/api/v2/invoices"
              ]
            }
          },
          "status": "OK",
          "code": 200,
          "_postman_previewlanguage": "json",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            }
          ],
          "cookie": [],
          "body": "{\n  \"content\": [\n    {\n      \"id\": \"{{invoice_id}}\",\n      \"invoiceNumber\": \"{{invoice_number}}\",\n      \"status\": \"ACCEPTED_TTN\"\n    }\n  ],\n  \"totalElements\": 1\n}"
        }
      ]
    },
    {
      "name": "Détail d'une facture",
      "request": {
        "method": "GET",
        "header": [
          {
            "key": "Authorization",
```

```
            "value": "Bearer {{access_token}}"
          }
        ],
        "url": {
          "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}",
          "host": [
            "{{base_url}}"
          ],
          "path": [
            "api",
            "v2",
            "invoices",
            "{{invoice_id}}"
          ]
        },
        "description": "Renvoie `invoice`, `teifJson`, `teifXml` et `ttnAcknowledgments`.
Scope requis : `invoice:read`."
      },
      "response": [
        {
          "name": "200 OK",
          "originalRequest": {
            "method": "GET",
            "header": [],
            "url": {
              "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}",
              "host": [
                "{{base_url}}/api/v2/invoices/{{invoice_id}}"
              ]
            }
          },
          "status": "OK",
          "code": 200,
          "_postman_previewlanguage": "json",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            }
          ],
          "cookie": [],
```

```
            "body": "{\n  \"invoice\": {\n    \"id\": \"{{invoice_id}}\",\n    \"status\":
        \"ACCEPTED_TTN\"\n  },\n  \"teifJson\": \"{...}\",\n  \"teifXml\": \"<TEIF>...
        </TEIF>\"\n}"
          }
        ]
      }
    ]
  },
  {
    "name": "📄 Factures — XML TEIF",
    "item": [
      {
        "name": "Soumettre une facture (XML)",
        "event": [
          {
            "listen": "test",
            "script": {
              "type": "text/javascript",
              "exec": [
                "if (pm.response.code === 201 || pm.response.code === 200) {",
                "  const json = pm.response.json();",
                "  if (json.id) {",
                "    pm.collectionVariables.set('invoice_id', json.id);",
                "    console.log(\"✅ invoice_id sauvegardé :\", json.id);",
                "  }",
                "} else {",
                "  console.error(\"❌ Facture non créée :\", pm.response.text());",
                "}"
              ]
            }
          }
        ],
        "request": {
          "method": "POST",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/xml"
            },
            {
              "key": "Authorization",
```

```
        "value": "Bearer {{access_token}}"
      }
    ],
    "body": {
      "mode": "raw",
      "raw": "<TEIF controlingAgency=\"TTN\" version=\"1.8.8\">\n  <InvoiceHeader>\n
<MessageSenderIdentifier type=\"I-01\">{{seller_tax_id}}</MessageSenderIdentifier>\n
<MessageRecieverIdentifier type=\"I-01\">{{buyer_tax_id}}</MessageRecieverIdentifier>\n
</InvoiceHeader>\n  <InvoiceBody>\n    <Bgm>\n      <DocumentIdentifier>{{invoice_number}}
</DocumentIdentifier>\n      <DocumentType code=\"I-11\">Facture</DocumentType>\n
</Bgm>\n    <Dtm>\n      <DateText format=\"ddMMyy\" functionCode=\"I-31\">
{{invoice_date_ddMMyy}}</DateText>\n    </Dtm>\n    <PartnerSection>\n
<PartnerDetails functionCode=\"I-62\">\n        <Nad>\n          <PartnerIdentifier
type=\"I-01\">{{seller_tax_id}}</PartnerIdentifier>\n          <PartnerName
nameType=\"Qualification\">{{seller_name}}</PartnerName>\n          <PartnerAdresses
lang=\"fr\">\n            <Street>{{seller_street}}</Street>\n            <CityName>
{{seller_city}}</CityName>\n            <PostalCode>{{seller_postal_code}}</PostalCode>\n
<Country codeList=\"ISO_3166-1\">TN</Country>\n          </PartnerAdresses>\n
</Nad>\n      </PartnerDetails>\n      <PartnerDetails functionCode=\"I-61\">\n
<Nad>\n          <PartnerIdentifier type=\"I-01\">{{buyer_tax_id}}</PartnerIdentifier>\n
<PartnerName nameType=\"Qualification\">{{buyer_name}}</PartnerName>\n
<PartnerAdresses lang=\"fr\">\n            <Street>{{buyer_street}}</Street>\n
<CityName>{{buyer_city}}</CityName>\n            <PostalCode>{{buyer_postal_code}}
</PostalCode>\n            <Country codeList=\"ISO_3166-1\">TN</Country>\n
</PartnerAdresses>\n        </Nad>\n      </PartnerDetails>\n    </PartnerSection>\n
<LinSection>\n      <Lin>\n        <ItemIdentifier>1</ItemIdentifier>\n        <LinImd
lang=\"fr\">\n          <ItemCode>{{line1_item_code}}</ItemCode>\n
<ItemDescription>{{line1_description}}</ItemDescription>\n        </LinImd>\n
<LinQty><Quantity measurementUnit=\"PCE\">{{line1_qty}}</Quantity></LinQty>\n
<LinTax>\n          <TaxTypeName code=\"I-1602\">TVA</TaxTypeName>\n          <TaxDetails>
<TaxRate>19</TaxRate></TaxDetails>\n        </LinTax>\n        <LinMoa>\n
<MoaDetails>\n            <Moa amountTypeCode=\"I-188\" currencyCodeList=\"ISO_4217\">\n
<Amount currencyIdentifier=\"TND\">{{line1_ht}}</Amount>\n            </Moa>\n
</MoaDetails>\n        </LinMoa>\n      </Lin>\n    </LinSection>\n    <InvoiceMoa>\n
<AmountDetails>\n        <Moa amountTypeCode=\"I-172\" currencyCodeList=\"ISO_4217\">
<Amount currencyIdentifier=\"TND\">{{total_ht}}</Amount></Moa>\n      </AmountDetails>\n
<AmountDetails>\n        <Moa amountTypeCode=\"I-181\" currencyCodeList=\"ISO_4217\">
<Amount currencyIdentifier=\"TND\">{{total_taxes}}</Amount></Moa>\n
</AmountDetails>\n      <AmountDetails>\n        <Moa amountTypeCode=\"I-180\"
currencyCodeList=\"ISO_4217\"><Amount currencyIdentifier=\"TND\">{{total_ttc}}</Amount>
</Moa>\n      </AmountDetails>\n    </InvoiceMoa>\n    <InvoiceTax>\n
<InvoiceTaxDetails>\n        <Tax>\n          <TaxTypeName code=\"I-
1602\">TVA</TaxTypeName>\n          <TaxDetails><TaxRate>19</TaxRate></TaxDetails>\n
</Tax>\n        <AmountDetails>\n          <Moa amountTypeCode=\"I-178\"
currencyCodeList=\"ISO_4217\"><Amount currencyIdentifier=\"TND\">{{total_tva}}</Amount>
</Moa>\n        </AmountDetails>\n      </InvoiceTaxDetails>\n    </InvoiceTax>\n
</InvoiceBody>\n</TEIF>",
      "options": {
        "raw": {
          "language": "xml"
        }
      }
    },
    "url": {
      "raw": "{{base_url}}/api/v2/invoices/submit-xml",
      "host": [
        "{{base_url}}"
      ],
      "path": [
```

```
              "api",
              "v2",
              "invoices",
              "submit-xml"
            ]
          },
          "description": "Alternative au format JSON si votre système génère déjà du XML TEIF.
      Scope requis : `invoice:write`."
        },
        "response": [
          {
            "name": "201 Created",
            "originalRequest": {
              "method": "POST",
              "header": [],
              "url": {
                "raw": "{{base_url}}/api/v2/invoices/submit-xml",
                "host": [
                  "{{base_url}}/api/v2/invoices/submit-xml"
                ]
              }
            },
            "status": "Created",
            "code": 201,
            "_postman_previewlanguage": "json",
            "header": [
              {
                "key": "Content-Type",
                "value": "application/json"
              }
            ],
            "cookie": [],
            "body": "{\n  \"id\": \"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\",\n
      \"invoiceNumber\": \"{{invoice_number}}\",\n  \"status\": \"VALIDATED\"\n}"
          }
        ]
      }
    ]
  },
  {
    "name": "✅ Validation TEIF",
```

```
    "item": [
      {
        "name": "Valider un XML (XSD uniquement)",
        "request": {
          "method": "POST",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            },
            {
              "key": "Authorization",
              "value": "Bearer {{access_token}}"
            }
          ],
          "body": {
            "mode": "raw",
            "raw": "{\n  \"xml\": \"<TEIF controlingAgency=\\\"TTN\\\" version=\\\"1.8.8\\\">...
</TEIF>\"\n}",
            "options": {
              "raw": {
                "language": "json"
              }
            }
          },
          "url": {
            "raw": "{{base_url}}/api/v2/teif/validate",
            "host": [
              "{{base_url}}"
            ],
            "path": [
              "api",
              "v2",
              "teif",
              "validate"
            ]
          },
          "description": "Valide un XML TEIF contre le schéma XSD sans créer de facture. Champ
du corps : `xml`. Scope requis : `invoice:read` ou `invoice:write`."
        },
        "response": [
```

```
    {
      "name": "200 OK – valid",
      "originalRequest": {
        "method": "POST",
        "header": [],
        "url": {
          "raw": "{{base_url}}/api/v2/teif/validate",
          "host": [
            "{{base_url}}/api/v2/teif/validate"
          ]
        }
      },
      "status": "OK",
      "code": 200,
      "_postman_previewlanguage": "json",
      "header": [
        {
          "key": "Content-Type",
          "value": "application/json"
        }
      ],
      "cookie": [],
      "body": "{\n  \"valid\": true,\n  \"issues\": []\n}"
    },
    {
      "name": "200 OK – invalid",
      "originalRequest": {
        "method": "POST",
        "header": [],
        "url": {
          "raw": "{{base_url}}/api/v2/teif/validate",
          "host": [
            "{{base_url}}/api/v2/teif/validate"
          ]
        }
      },
      "status": "OK",
      "code": 200,
      "_postman_previewlanguage": "json",
      "header": [
```

```
              {
                "key": "Content-Type",
                "value": "application/json"
              }
            ],
            "cookie": [],
            "body": "{\n  \"valid\": false,\n  \"issues\": [\n    {\n      \"type\":
      \"XML_SCHEMA_VIOLATION\",\n      \"severity\": \"ERROR\",\n      \"message\": \"...\"\n
      }\n  ]\n}"
          }
        ]
      }
    ]
  },
  {
    "name": "✍️ Signature et envoi TTN",
    "item": [
      {
        "name": "Signer une facture (SEAL)",
        "request": {
          "method": "POST",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            },
            {
              "key": "Authorization",
              "value": "Bearer {{access_token}}"
            }
          ],
          "body": {
            "mode": "raw",
            "raw": "{\n  \"provider\": \"seal\"\n}",
            "options": {
              "raw": {
                "language": "json"
              }
            }
          },
          "url": {
```

```
        "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/sign",
        "host": [
          "{{base_url}}"
        ],
        "path": [
          "api",
          "v2",
          "invoices",
          "{{invoice_id}}",
          "sign"
        ]
      },
      "description": "Signer une facture au statut VALIDATED. Scope requis : `seal:sign` (SEAL Max) ou `digigo:sign` (DigiGO, avec `{\"provider\":\"digigo\"}`). Ne soumet pas à TTN — utilisez ensuite `submit-ttn`, ou préférez `sign-and-send` ci-dessous pour tout faire en un appel."
    },
    "response": [
      {
        "name": "200 OK",
        "originalRequest": {
          "method": "POST",
          "header": [],
          "url": {
            "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/sign",
            "host": [
              "{{base_url}}/api/v2/invoices/{{invoice_id}}/sign"
            ]
          }
        },
        "status": "OK",
        "code": 200,
        "_postman_previewlanguage": "json",
        "header": [
          {
            "key": "Content-Type",
            "value": "application/json"
          }
        ],
        "cookie": [],
        "body": "{\n  \"id\": \"{{invoice_id}}\",\n  \"status\": \"SIGNED\",\n  \"signedAt\": \"2026-07-12T10:00:00Z\"\n}"
```

```
        }
      ]
    },
    {
      "name": "Envoyer une facture signée à TTN",
      "request": {
        "method": "POST",
        "header": [
          {
            "key": "Authorization",
            "value": "Bearer {{access_token}}"
          }
        ],
        "url": {
          "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/submit-ttn",
          "host": [
            "{{base_url}}"
          ],
          "path": [
            "api",
            "v2",
            "invoices",
            "{{invoice_id}}",
            "submit-ttn"
          ]
        },
        "description": "Soumet une facture déjà signée à TTN (Tunisia Trade Net). Scope requis : `ttn:submit`. Consomme le quota de signature mensuel (sauf si déjà comptabilisé par un job sign-and-send)."
      },
      "response": [
        {
          "name": "200 OK",
          "originalRequest": {
            "method": "POST",
            "header": [],
            "url": {
              "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/submit-ttn",
              "host": [
                "{{base_url}}/api/v2/invoices/{{invoice_id}}/submit-ttn"
              ]
```

```
            }
          },
          "status": "OK",
          "code": 200,
          "_postman_previewlanguage": "json",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            }
          ],
          "cookie": [],
          "body": "{\n  \"id\": \"{{invoice_id}}\",\n  \"status\": \"ACCEPTED_TTN\",\n
\"ttnGeneratedRef\": \"TTN-REF-000123\"\n}"
        }
      ]
    },
    {
      "name": "Signer + envoyer à TTN (recommandé, async)",
      "event": [
        {
          "listen": "test",
          "script": {
            "type": "text/javascript",
            "exec": [
              "const json = pm.response.json();",
              "if (json.jobId) {",
              "  pm.collectionVariables.set('job_id', json.jobId);",
              "  console.log(\"✅ job_id sauvegardé :\", json.jobId);",
              "} else {",
              "  console.error(\"❌ Pas de jobId :\", JSON.stringify(json));",
              "}"
            ]
          }
        }
      ],
      "request": {
        "method": "POST",
        "header": [
          {
            "key": "Content-Type",
```

```
            "value": "application/json"
          },
          {
            "key": "Authorization",
            "value": "Bearer {{access_token}}"
          }
        ],
        "body": {
          "mode": "raw",
          "raw": "{\n  \"signatureType\": \"SEAL\"\n}",
          "options": {
            "raw": {
              "language": "json"
            }
          }
        },
        "url": {
          "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/sign-and-send",
          "host": [
            "{{base_url}}"
          ],
          "path": [
            "api",
            "v2",
            "invoices",
            "{{invoice_id}}",
            "sign-and-send"
          ]
        },
        "description": "Met en file d'attente la signature et l'envoi à TTN en une seule opération asynchrone. `signatureType` : `SEAL` ou `DIGIGO` selon votre plan. Le script de test sauvegarde le `jobId` retourné dans `{{job_id}}`."
      },
      "response": [
        {
          "name": "200 OK — job créé",
          "originalRequest": {
            "method": "POST",
            "header": [],
            "url": {
              "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/sign-and-send",
```

```
            "host": [
              "{{base_url}}/api/v2/invoices/{{invoice_id}}/sign-and-send"
            ]
          }
        },
        "status": "OK",
        "code": 200,
        "_postman_previewlanguage": "json",
        "header": [
          {
            "key": "Content-Type",
            "value": "application/json"
          }
        ],
        "cookie": [],
        "body": "{\n  \"jobId\": \"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\",\n  \"invoiceId\": \"{{invoice_id}}\",\n  \"jobType\": \"SIGN_AND_SUBMIT_SEAL\",\n  \"status\": \"PENDING\",\n  \"success\": true\n}"
      },
      {
        "name": "402 Payment Required — quota épuisé",
        "originalRequest": {
          "method": "POST",
          "header": [],
          "url": {
            "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/sign-and-send",
            "host": [
              "{{base_url}}/api/v2/invoices/{{invoice_id}}/sign-and-send"
            ]
          }
        },
        "status": "Payment Required",
        "code": 402,
        "_postman_previewlanguage": "json",
        "header": [
          {
            "key": "Content-Type",
            "value": "application/json"
          }
        ],
        "cookie": [],
```

```
        "body": "{\n  \"error\": \"NO_SIGNATURE_CREDITS\",\n  \"message\": \"No signing
quota or signature-pack credits available.\",\n  \"packsRemaining\": 0\n}"
      }
    ]
  },
  {
    "name": "Suivre le job (poll)",
    "request": {
      "method": "GET",
      "header": [
        {
          "key": "Authorization",
          "value": "Bearer {{access_token}}"
        }
      ],
      "url": {
        "raw": "{{base_url}}/api/v2/jobs/{{job_id}}",
        "host": [
          "{{base_url}}"
        ],
        "path": [
          "api",
          "v2",
          "jobs",
          "{{job_id}}"
        ]
      },
      "description": "`status` passe de `PENDING` à `COMPLETED` ou `FAILED`. Scope requis :
`invoice:read`."
    },
    "response": [
      {
        "name": "200 OK — completed",
        "originalRequest": {
          "method": "GET",
          "header": [],
          "url": {
            "raw": "{{base_url}}/api/v2/jobs/{{job_id}}",
            "host": [
              "{{base_url}}/api/v2/jobs/{{job_id}}"
            ]
```

```
            }
          },
          "status": "OK",
          "code": 200,
          "_postman_previewlanguage": "json",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            }
          ],
          "cookie": [],
          "body": "{\n  \"jobId\": \"{{job_id}}\",\n  \"invoiceId\": \"{{invoice_id}}\",\n  \"status\": \"COMPLETED\",\n  \"progress\": 100,\n  \"ttnFinalResult\": {\n    \"status\": \"ACCEPTED_TTN\"\n  }\n}"
        }
      ]
    }
  ]
},
{
  "name": "📎 Artefacts (XML / PDF)",
  "item": [
    {
      "name": "Télécharger le PDF",
      "request": {
        "method": "GET",
        "header": [
          {
            "key": "Authorization",
            "value": "Bearer {{access_token}}"
          }
        ],
        "url": {
          "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/pdf",
          "host": [
            "{{base_url}}"
          ],
          "path": [
            "api",
            "v2",
```

```
            "invoices",
            "{{invoice_id}}",
            "pdf"
          ],
          "query": [
            {
              "key": "pdfTemplate",
              "value": "",
              "description": "Optionnel — sinon le modèle par défaut de l'organisation est utilisé",
              "disabled": true
            },
            {
              "key": "lang",
              "value": "",
              "description": "Optionnel — langue du document (fr/en/ar)",
              "disabled": true
            }
          ]
        },
        "description": "Réponse binaire (`application/pdf`), quel que soit le statut de la facture (brouillon inclus). Scope requis : `invoice:read`."
      },
      "response": []
    },
    {
      "name": "Récupérer le XML généré",
      "request": {
        "method": "GET",
        "header": [
          {
            "key": "Authorization",
            "value": "Bearer {{access_token}}"
          }
        ],
        "url": {
          "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/generated-xml",
          "host": [
            "{{base_url}}"
          ],
          "path": [
```

```
              "api",
              "v2",
              "invoices",
              "{{invoice_id}}",
              "generated-xml"
            ]
          },
          "description": "⚠️ Au-delà du statut SIGNED, cet endpoint régénère un XML NON SIGNÉ à la volée (perd la signature et le QR code TTN). Pour l'artefact final réellement signé/horodaté TTN, lisez plutôt le champ `teifXml` de `GET /invoices/:id`. Scope requis : `invoice:read`."
        },
        "response": [
          {
            "name": "200 OK",
            "originalRequest": {
              "method": "GET",
              "header": [],
              "url": {
                "raw": "{{base_url}}/api/v2/invoices/{{invoice_id}}/generated-xml",
                "host": [
                  "{{base_url}}/api/v2/invoices/{{invoice_id}}/generated-xml"
                ]
              }
            },
            "status": "OK",
            "code": 200,
            "_postman_previewlanguage": "json",
            "header": [
              {
                "key": "Content-Type",
                "value": "application/json"
              }
            ],
            "cookie": [],
            "body": "{\n  \"note\": \"raw application/xml body, not JSON\"\n}"
          }
        ]
      }
    ]
  },
  {
```

```
    "name": "📊 Consultation et statistiques",
    "item": [
      {
        "name": "Statistiques du tableau de bord",
        "request": {
          "method": "GET",
          "header": [
            {
              "key": "Authorization",
              "value": "Bearer {{access_token}}"
            }
          ],
          "url": {
            "raw": "{{base_url}}/api/v2/invoices/stats",
            "host": [
              "{{base_url}}"
            ],
            "path": [
              "api",
              "v2",
              "invoices",
              "stats"
            ]
          }
        },
        "response": [
          {
            "name": "200 OK",
            "originalRequest": {
              "method": "GET",
              "header": [],
              "url": {
                "raw": "{{base_url}}/api/v2/invoices/stats",
                "host": [
                  "{{base_url}}/api/v2/invoices/stats"
                ]
              }
            },
            "status": "OK",
            "code": 200,
```

```
        "_postman_previewlanguage": "json",
        "header": [
          {
            "key": "Content-Type",
            "value": "application/json"
          }
        ],
        "cookie": [],
        "body": "{\n  \"total\": 42,\n  \"sentThisMonth\": 5\n}"
      }
    ]
  },
  {
    "name": "Statistiques par clé API",
    "request": {
      "method": "GET",
      "header": [
        {
          "key": "Authorization",
          "value": "Bearer {{access_token}}"
        }
      ],
      "url": {
        "raw": "{{base_url}}/api/v2/invoices/api-key-stats",
        "host": [
          "{{base_url}}"
        ],
        "path": [
          "api",
          "v2",
          "invoices",
          "api-key-stats"
        ]
      }
    },
    "response": [
      {
        "name": "200 OK",
        "originalRequest": {
          "method": "GET",
```

```
            "header": [],
            "url": {
              "raw": "{{base_url}}/api/v2/invoices/api-key-stats",
              "host": [
                "{{base_url}}/api/v2/invoices/api-key-stats"
              ]
            }
          },
          "status": "OK",
          "code": 200,
          "_postman_previewlanguage": "json",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            }
          ],
          "cookie": [],
          "body": "{\n  \"organizationId\": \"org_xxxx\",\n  \"totalInvoices\": 42,\n  \"stats\": [\n    {\n      \"appId\": \"{{client_id}}\",\n      \"invoicesCreated\": 42,\n  \"invoicesSigned\": 40,\n      \"invoicesSubmittedToTtn\": 38\n    }\n  ]\n}"
        }
      ]
    }
  ]
},
{
  "name": "🔔 Webhooks",
  "item": [
    {
      "name": "Créer un webhook",
      "request": {
        "method": "POST",
        "header": [
          {
            "key": "Content-Type",
            "value": "application/json"
          },
          {
            "key": "Authorization",
            "value": "Bearer {{access_token}}"
```

```
        }
      ],
      "body": {
        "mode": "raw",
        "raw": "{\n  \"url\": \"https://votre-erp.example.com/webhooks/fatoora\",\n
\"events\": [\n    \"invoice.validated\",\n    \"invoice.signed\",\n
\"invoice.accepted_ttn\",\n    \"invoice.rejected_ttn\"\n  ]\n}",
        "options": {
          "raw": {
            "language": "json"
          }
        }
      },
      "url": {
        "raw": "{{base_url}}/api/v2/webhooks",
        "host": [
          "{{base_url}}"
        ],
        "path": [
          "api",
          "v2",
          "webhooks"
        ]
      },
      "description": "Le secret du webhook est retourné une seule fois — sauvegardez-le pour
vérifier la signature des événements reçus."
    },
    "response": [
      {
        "name": "201 Created",
        "originalRequest": {
          "method": "POST",
          "header": [],
          "url": {
            "raw": "{{base_url}}/api/v2/webhooks",
            "host": [
              "{{base_url}}/api/v2/webhooks"
            ]
          }
        },
        "status": "Created",
        "code": 201,
```

```
      "_postman_previewlanguage": "json",
      "header": [
        {
          "key": "Content-Type",
          "value": "application/json"
        }
      ],
      "cookie": [],
      "body": "{\n  \"id\": \"wh_xxxx\",\n  \"url\": \"https://votre-erp.example.com/webhooks/fatoora\",\n  \"secret\": \"whsec_xxxxxxxxxxxx\"\n}"
    }
  ]
},
{
  "name": "Lister les webhooks",
  "request": {
    "method": "GET",
    "header": [
      {
        "key": "Authorization",
        "value": "Bearer {{access_token}}"
      }
    ],
    "url": {
      "raw": "{{base_url}}/api/v2/webhooks",
      "host": [
        "{{base_url}}"
      ],
      "path": [
        "api",
        "v2",
        "webhooks"
      ]
    }
  },
  "response": [
    {
      "name": "200 OK",
      "originalRequest": {
        "method": "GET",
        "header": [],
```

```
            "url": {
              "raw": "{{base_url}}/api/v2/webhooks",
              "host": [
                "{{base_url}}/api/v2/webhooks"
              ]
            }
          },
          "status": "OK",
          "code": 200,
          "_postman_previewlanguage": "json",
          "header": [
            {
              "key": "Content-Type",
              "value": "application/json"
            }
          ],
          "cookie": [],
          "body": "{\n  \"webhooks\": [\n    {\n      \"id\": \"wh_xxxx\",\n      \"url\":
\"https://votre-erp.example.com/webhooks/fatoora\",\n      \"isActive\": true\n    }\n
]\n}"
        }
      ]
    },
    {
      "name": "Tester un webhook",
      "request": {
        "method": "POST",
        "header": [
          {
            "key": "Authorization",
            "value": "Bearer {{access_token}}"
          }
        ],
        "url": {
          "raw": "{{base_url}}/api/v2/webhooks/wh_xxxx/test",
          "host": [
            "{{base_url}}"
          ],
          "path": [
            "api",
            "v2",
```

```
                "webhooks",
                "wh_xxxx",
                "test"
              ]
            }
          },
          "response": [
            {
              "name": "200 OK",
              "originalRequest": {
                "method": "POST",
                "header": [],
                "url": {
                  "raw": "{{base_url}}/api/v2/webhooks/wh_xxxx/test",
                  "host": [
                    "{{base_url}}/api/v2/webhooks/wh_xxxx/test"
                  ]
                }
              },
              "status": "OK",
              "code": 200,
              "_postman_previewlanguage": "json",
              "header": [
                {
                  "key": "Content-Type",
                  "value": "application/json"
                }
              ],
              "cookie": [],
              "body": "{\n  \"delivered\": true\n}"
            }
          ]
        }
      ]
    }
  ]
}
```

This guide covers the `/api/v2/*` surface documented in `fatooraBusiness/src/routes/core-proxy.routes.ts` (mounted twice, identically, at `/api/core-proxy/*` and `/api/v2/*`) and `fatooraCore` (`InvoiceController`, `QuotationController`, `WebhookController`). Written 2026-07-15 alongside the Partner API guide's webhooks/artifacts update — response shapes and error codes are shared with that guide since both surfaces are backed by the same fatooraCore controllers, verified against the same source at the same time.

The Postman collection JSON in §15 is not hand-written — it's the literal output of `buildApiKeysPostmanCollection()` in `fatooraLive/src/components/api-keys/ApiKeysUsageGuide.tsx`, extracted and executed with Node to guarantee fidelity with the code, following the same convention as the Partner API guide's §17. The OpenAPI YAML in §15 is adapted from `fatooraLive/public/openapi/api-oauth.yaml` (the existing org/API-key-scoped spec already linked from `/app/api-keys`), extended with the webhook and artifact-retrieval endpoints.