

JG2 - API Reference

4. API Reference

“ **Base URL:** `https://{host}/api` **Auth:** Bearer JWT (Keycloak OIDC) — header `Authorization: Bearer <token>` **Tenant:** estratto automaticamente dal claim `groups` del JWT

4.1 Autenticazione

Flusso OIDC (produzione)

```
sequenceDiagram
    participant U as Browser
    participant KC as Keycloak gsauth
    participant BE as GPE_NEXT Backend

    U->>KC: Redirect login
    KC-->>U: Access Token (JWT)
    U->>BE: Request + Authorization: Bearer JWT
    BE->>KC: Valida token (JWKS)
    BE-->>U: Response (tenant isolato)
```

Realm: `gsauth` | **Client:** `gpenext`

Token claims usati:

- `sub` → user ID
- `groups` → `/client/{keycloak_group_path}` → tenant
- `realm_access.roles` → `SUPER_ADMIN`, `OPERATOR`, `CLIENT`

4.2 Endpoint — Schemas (Tipi Documento)

GET /schemas

Lista tutti gli schemi del tenant corrente. **Ruoli:** OPERATOR, SUPER_ADMIN

Response 200:

```
[
  {
    "id": "uuid",
    "name": "EFATT121_1P",
    "documentType": "EFATT121_1P",
    "documentLabel": "Fattura",
    "erpMinVersion": "6.9.0.0",
    "forceMigration": false,
    "hasDittaXpath": true,
    "xmlFilesCount": 3
  }
]
```

POST /schemas

Crea schema vuoto. **Ruoli:** OPERATOR

POST /schemas/{id}/xsd

Upload file XSD. `multipart/form-data`: `xsdFile`, `displayName`

POST /schemas/{id}/xml-files

Upload file XML di test. `multipart/form-data`: `xmlFile`, `displayName`

GET /schemas/{id}/fields

Albero dei campi XSD per il data tree nel designer.

Response 200:

```
{
  "root": "RFAMEZ2D",
  "fields": [
    {
      "name": "TESTA",
      "type": "complex",
      "xpath": "/RFAMEZ2D/TESTA",
      "children": [
        { "name": "CODE_DITT", "type": "string", "xpath": "/RFAMEZ2D/TESTA/CODE_DITT" }
      ]
    }
  ]
}
```

PUT /schemas/{id}/ditta-xpath

Aggiorna XPath per il codice ditta. **Ruoli:** OPERATOR

```
{ "dittaCodeXPath": "./TESTA/CODE_DITT" }
```

4.3 Endpoint — Templates (Modelli)

GET /templates

Lista modelli del tenant. Params: `schemaId`, `isBase`

POST /templates

Crea nuovo modello. **Ruoli:** OPERATOR

```
{
  "name": "Fattura Personalizzata",
  "profileName": "custom",
  "schemaId": "uuid",
  "xmlFileId": "uuid",
}
```

```
"orientation": "portrait",
"pageFormat": "A4"
}
```

GET /templates/{id}

Dettaglio completo template (include contenuto JSON JSONB).

PUT /templates/{id}

Salva il modello (bozza). **Ruoli:** OPERATOR

DELETE /templates/{id}

Elimina modello. **Ruoli:** OPERATOR

POST /templates/{id}/clone

Clona modello (crea profilo cliente).

```
{
  "profileName": "profilo-cliente-xyz",
  "targetTenantId": "uuid"
}
```

PUT /templates/{id}/activate

Attiva un profilo come default per il tipo documento del tenant. **Ruoli:** OPERATOR, SUPER_ADMIN

POST /templates/{id}/publish

Pubblica il template (crea versione pubblicata). **Ruoli:** OPERATOR

Response 200:

```
{
  "versionId": "uuid",
  "publishedAt": "2026-06-19T10:00:00Z",
}
```

```
"erpMinVersion": "6.9.0.0"
}
```

GET /templates/{id}/versions

Storico versioni pubblicate del template.

PATCH /templates/{id}/restore/{versionId}

Ripristina una versione storica come contenuto corrente.

POST /templates/{id}/publish-xslfo

Genera e scarica il file `.xslt-xfo` (funzionalità secondaria).

Query params: `writeToDisk` (boolean, default: false)

Response 200:

```
Content-Type: application/xml; charset=ISO-8859-1
Content-Disposition: attachment; filename="EFATT121_1P_xfo.xml"
X-Published-Path: /gsnr/gpe-next/data/xslfo/tenant-uuid/EFATT121_1P_xfo.xml
```

GET /templates/{id}/xslfo-source

Visualizza il codice XSLT-FO generato senza scaricare. **Ruoli:** OPERATOR

POST /templates/{id}/preview

Genera anteprima PDF del template.

```
{ "xmlFileId": "uuid", "overrides": {} }
```

POST /templates/bulk-import

Import massivo file `.mdl` legacy. `multipart/form-data`: `files[]`

POST /templates/import-mdl

Import singolo file `.mdl`. `multipart/form-data`: `mdlFile`, `schemaId`, `mode` (auto|manual)

4.4 Endpoint — Render (Integrazione ERP)

POST /render

Endpoint principale per ERP. Riceve XML dati ERP, restituisce PDF. Usato sia dal nuovo ERP Angular che dal bridge ERP COBOL eSIGEA.

Ruoli: OPERATOR, CLIENT, SUPER_ADMIN (tipicamente chiamato da ERP con service account)

Request:

```
Content-Type: application/xml
Authorization: Bearer <JWT>

<?xml version="1.0" encoding="UTF-8"?>
<EFATT121>
  <estr_cont>...dati fattura...</estr_cont>
</EFATT121>
```

Query params:

- `documentType` (string) — tipo documento (es. `EFATT121_1P`)
- `erpVersion` (string, opzionale) — versione ERP per selezione template
- `profileName` (string, opzionale) — profilo specifico

Response 200: PDF binario **Response 404:** nessun template attivo per il tipo documento

Response 422: XML malformato o non valido contro XSD

GET /render/version-info

Info versione template che verrebbe usato per un dato documento.

4.5 Endpoint — Client Dashboard

GET /client/dashboard

Dashboard con tutti i documenti disponibili per il tenant. **Ruoli:** CLIENT, OPERATOR

GET

/client/documents/{documentType}/profiles

Lista profili disponibili per un tipo documento. **Ruoli:** CLIENT

4.6 Endpoint — Me / Tenant Info

GET /me/tenant

Info sul tenant e utente corrente.

Response 200:

```
{
  "tenantId": "uuid",
  "tenantSlug": "azienda-xyz",
  "companyName": "Azienda XYZ S.r.l.",
  "role": "OPERATOR",
  "erpVersion": "6.9.0.0"
}
```

4.7 Endpoint — Immagini

POST /images

Upload immagine/logo. `multipart/form-data`: `image`, `displayName`

GET /images/{id}/content

Scarica il contenuto binario dell'immagine.

4.8 Endpoint — Admin (SUPER_ADMIN only)

`GET /admin/tenants` — Lista tutti i tenant

`PUT /admin/tenants/{id}/erp-version` — Imposta versione ERP per un tenant

`GET /admin/templates` — Lista tutti i template cross-tenant

4.9 Codici di errore

Codice	Situazione
400	Richiesta malformata
401	Token JWT mancante o scaduto
403	Ruolo insufficiente o tentativo cross-tenant
404	Risorsa non trovata (o non appartiene al tenant)
409	Conflitto (es. nome duplicato)
422	XML non valido contro XSD
500	Errore interno (log su Sentry)

Formato errore:

```
{
  "status": 404,
  "error": "Not Found",
  "message": "Template not found",
  "timestamp": "2026-06-19T10:00:00Z"
}
```

4.10 Esempi curl

```
# Render PDF da XML ERP (nuovo ERP o bridge COBOL)
curl -X POST https://gpenext.host/api/render \
  -H "Authorization: Bearer $TOKEN" \
  -H "Content-Type: application/xml" \
  -d @fattura.xml \
  --query "documentType=EFATT121_1P" \
  -o output.pdf

# Scaricare XSLT-F0 (funzionalità secondaria)
curl -X POST https://gpenext.host/api/templates/{id}/publish-xslfo \
  -H "Authorization: Bearer $TOKEN" \
  -o EFATT121_1P_xfo.xsl
```

Revisione #1

Creato 22 giugno 2026 10:16:00 da Giuseppe Anelli

Aggiornato 22 giugno 2026 10:16:19 da Giuseppe Anelli