oec.cat · NTSP

Manual tècnic d’integració

APIs **id.ntsp.dev**, signatura digital PDF, autenticació JWT i arquitectura de microserveis per a programadors i enginyers.

Versió 1.0 · Juliol 2026

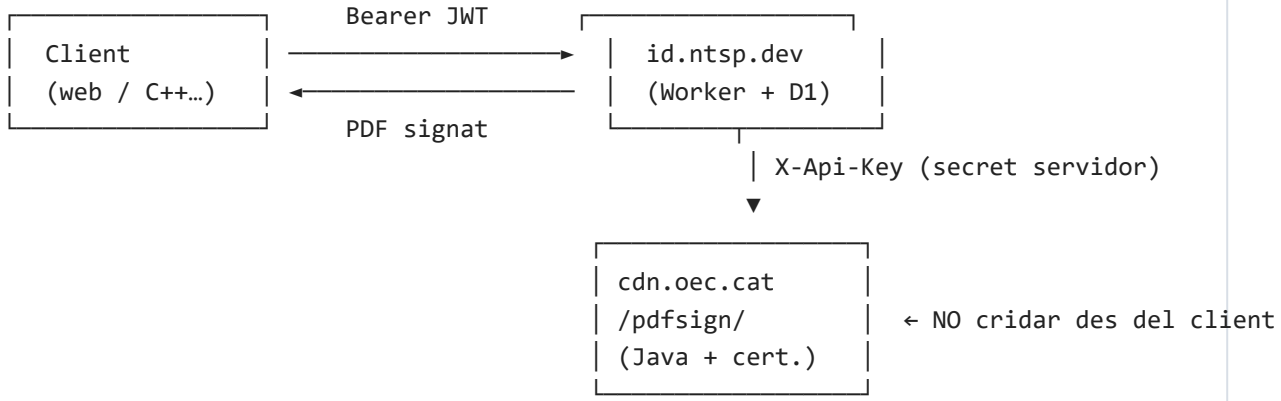
Tipus de lletra: Helvetica LT Std · Domini API: `https://id.ntsp.dev`

Públic destinatari: desenvolupadors d'aplicacions web, escriptori (C/C++, .NET) o scripts que integrin login OEC, signatura de documents o conversió ofimàtica.
No inclou: configuració d'usuaris al panell d'administració (vegeu `control.html` intern).

1. Índex

2. Arquitectura general

Tots els clients (navegador, app nativa, script) parlen amb el **Worker Cloudflare** a `https://id.ntsp.dev`. El Worker valida l'usuari (JWT), comprova permisos i, per signatura, fa de proxy cap al VPS OEC.



Component	URL / rol
API pública auth + signatura	`https://id.ntsp.dev`
Signatura criptogràfica (intern)	`https://cdn.oec.cat/pdfsign/` — només Worker
Certificat confiança	`oec-pdfsign.crt` (manual certificat-pdf)
Portal aplicacions	`https://aplicatius.oec.cat`
Login web OEC	`https://e-identificacio.oec.cat`

Regla crítica: La clau `PDFSIGN_SECRET` / `X-Api-Key` del microservei Java *mai* ha d'estar en una app client. Si algú la obté, podria signar PDFs sense passar per l'autenticació NTSP.

3. Autenticació

3.1 Login

POST `/login`

Content-Type	`application/json`
Body	`{ "codi": "usuari_o_uid", "pass": "PIN", "xarxa_required": "oec.cat" }` (xarxa opcional)
Resposta 200	`{ "access_token": "…", "refresh_token": "…" }`
Errors	401 credencials; 429 massa intents; 400 validació

```
curl -sS -X POST "https://id.ntsp.dev/login" \
  -H "Content-Type: application/json" \
  -d '{"codi":"joan@oec.cat","pass":"1234","xarxa_required":"oec.cat"}'
```

El camp `codi` accepta **username** o **UID** (alias configurat a la base de dades). El PIN té entre 4 i 128 caràcters.

3.2 Refresh token

POST `/refresh`

```
{
  "refresh_token": "eyJ…"
}
```

Retorna nous `access_token` i `refresh_token` (rotació). El token antic queda invalidat al KV.

3.3 Logout

POST `/logout` — invalida refresh token (cookie o body).

3.4 Validar sessió

GET `/me` — Header: `Authorization: Bearer <access_token>`

```
{
  "sub": "joan@oec.cat",
  "uid": "…",
  "nom_complert": "Joan Garcia",
  "role": "user",
```

```
  "pwd_change_required": false
}
```

4. JWT i sessió

Access token	Durada ~15 minuts. S'envia a cada API protegida.
Refresh token	~30 dies. Guardar de forma segura (Keychain, Credential Manager, no plaintext).
Header	`Authorization: Bearer <access_token>`
CORS	Només orígens `*.ntsp.dev` i `*.oec.cat` (no afecta apps natives).

5. Permisos per aplicació

Cada usuari pot tenir apps habilitades o deshabilitades (`user_app_prefs`). Exemples d' `app_id` :

- `oec:app1` — e-Certificat document PDF
- `oec:merge` — signatura i conversió merge
- `oec:convert` — conversor ofimàtic
- `oec:manuals` — manuals oec.cat

GET `/me/apps` — llista apps disponibles per l'usuari.

Si l'app està deshabilitada, l'API retorna **403** amb missatge `Accés denegat a oec:…` .

6. e-Certificat PDF (App1)

Aplica segell visual al marge + metadades + signatura PAdES. Requereix `nom_complert` a la fitxa d'usuari.

POST `/api/app1/e-cert-document-pdf`

Auth	Bearer JWT + permís `oec:app1`
Body	`multipart/form-data`, camp `file` = PDF
Límits	Màx. 25 MB; màx. 150 pàgines; ha de ser `%PDF`
Resposta	Cos = PDF signat; capçalera `X-E-Cert-Doc-Id: 00001234`

Flux intern

1. Increment comptador document (`e_cert_doc_counter`)
2. Segell visual Helvetica al marge esquerre (text vertical)
3. Títol signatura: `SIGNATURA DIGITAL oec.cat - NOM COMPLERT - CORREU`
4. Envia PDF al microservei Java PDFSIGN
5. Retorna PDF signat criptogràficament

```
curl -sS -X POST "https://id.ntsp.dev/api/app1/e-cert-document-pdf" \
  -H "Authorization: Bearer ACCESS_TOKEN" \
  -F "file=@document.pdf" \
  -o document-signat.pdf -D headers.txt
```

7. Signatura PDF (Merge)

Signatura PAdES sense segell e-Cert ni comptador. Només afegeix metadades de signatura si el client no les envia.

POST `/api/merge/sign-pdf`

Auth	Bearer JWT + `oec:merge`
Body	`multipart/form-data`, camp `file`
Camps opcionals	`title`, `signatureTitle`, `reason`, `location`, `contact`

Si no s'envia títol, el Worker omple automàticament el format OEC estàndard amb nom complet i correu.

8. Conversió ofimàtica

Tots requereixen JWT i permisos adequats. Proxy cap a Gotenberg o microserveis al CDN.

Endpoint	Funció	App
`POST /api/merge/docx-to-pdf`	DOCX → PDF	oec:merge
`POST /api/convert/office-to-pdf`	Office → PDF	oec:convert
`POST /api/convert/pdf-to-docx`	PDF → DOCX	oec:convert
`POST /api/merge/pdf-to-docx`	PDF → DOCX (merge)	oec:merge

Body: multipart, camp `file`. Límits típics: 20–25 MB per fitxer.

9. Microservei PDFSIGN (ús intern)

URL configurada al Worker: `PDFSIGN_URL` (p.ex. `https://cdn.oec.cat/pdfsign`).

Mètode	POST `/sign-pdf` (path configurable amb `PDFSIGN_SIGN_PATH`)
Auth	Capçalera `X-Api-Key: PDFSIGN_SECRET`
Body	multipart `file` + camps metadades (`title`, `reason`, …)

> **Comprovació de seguretat (juliol 2026):** Peticions amb PDF però sense clau o amb clau incorrecta retornen **401 unauthorized**. Executar `.\scripts\verify-pdfsign-security.ps1` després de canvis al VPS.

Nginx al VPS ha d'incloure `scripts/snippets/oec-pdfsign.conf.example` (substituir `SIGN_API_KEY`) per defensa en profunditat abans del Java.

10. Integració C / C++ / Python

10.1 Flux recomanat (apps natives)

1. POST `/login` → obtenir tokens
2. Desar `refresh_token` de forma segura
3. Abans de cada operació (o quan 401): POST `/refresh`
4. POST multipart a l'endpoint de signatura amb `Authorization: Bearer …`
5. Escriure bytes de resposta a fitxer PDF

10.2 Exemple Python (requests)

```
import requests

API = "https://id.ntsp.dev"
r = requests.post(f"{API}/login", json={"codi": "user", "pass": "1234"})
tokens = r.json()
access = tokens["access_token"]

with open("doc.pdf", "rb") as f:
    resp = requests.post(
        f"{API}/api/app1/e-cert-document-pdf",
        headers={"Authorization": f"Bearer {access}"},
        files={"file": ("doc.pdf", f, "application/pdf")},
    )
resp.raise_for_status()
open("signat.pdf", "wb").write(resp.content)
print(resp.headers.get("X-E-Cert-Doc-Id"))
```

10.3 Exemple C++ (libcurl)

```
// 1) Login JSON → parsejar access_token
// 2) curl_mime: camp "file" = ruta PDF
// 3) Header: Authorization: Bearer <token>
// 4) URL: https://id.ntsp.dev/api/app1/e-cert-document-pdf
// 5) Callback WRITE per guardar el PDF de resposta
```

10.4 Windows (.NET)

Utilitzeu `HttpClient` amb `MultipartFormDataContent` i `AuthenticationHeaderValue("Bearer", token)`. Per tokens, `ProtectedData` o Windows Credential Locker.

11. Certificat i verificació PDF

La signatura és del certificat corporatiu **oec-pdfsign**, no d'un certificat personal de l'usuari. La identitat del signant es reflecteix al segell visual i al títol de signatura.

- Descarregar `oec-pdfsign.crt` des del portal certificat-pdf
- Adobe Acrobat: Trusted Certificates → Import
- Windows: instal·lar a Trusted Root (Windows no mostra signatures PDF a Propietats; useu Acrobat)

12. Seguretat i auditoria

Control	Detall
Rate limiting login	KV Cloudflare; bloqueig temporal per IP/usuari
Hash contrasenya	PBKDF2-SHA256, 100k iteracions + salt
Refresh rotation	Token únic actiu per usuari al KV
Logs	`E_CERT_DOC_STAMPED`, `PDF_SIGNED`, `PDFSIGN_FAIL`
Informe complet	`docs/SECURITY-AUDIT-OEC-APIS.md` al repo

Acció pendent (ops): Si `GOTENBERG_SECRET` va estar al git en clar, rotar la clau al VPS i a Wrangler. El fitxer `scripts/cdn-oec-nginx.conf` ara usa placeholder.

13. Errors comuns

Missatge	Causa / solució
Token no proporcionat / invàlid	Falta Bearer o token caducat → refresh
Accés denegat a oec:app1	App deshabilitada per l'administrador
Falta el nom complet	Admin ha d'emplenar `nom_complert` a la fitxa
PDF massa gran / massa pàgines	Límits 25 MB / 150 pàgines
PDF protegit o corrupte	Exportar sense contrasenya; comprovar integritat
Signatura VPS ha fallat	Revisar PDFSIGN_URL/SECRET; logs `[E_CERT_PDFSIGN]`

14. Referència ràpida d’endpoints

Mètode	Path	Auth	Descripció
POST	`/login`	No	Obtenir JWT
POST	`/refresh`	Refresh token	Renovar sessió
POST	`/logout`	Opcional	Tancar sessió
GET	`/me`	Bearer	Perfil
GET	`/me/apps`	Bearer	Apps permeses
POST	`/api/app1/e-cert-document-pdf`	Bearer + app1	e-Cert + signatura
POST	`/api/merge/sign-pdf`	Bearer + merge	Signatura PDF
POST	`/api/merge/docx-to-pdf`	Bearer + merge	Conversió DOCX
POST	`/change-password`	Bearer	Canvi PIN

Manual API OEC · oec.cat / NTSP · Generat per a integradors · Helvetica LT Std

Còpia PDF: `manual-api-oec-developers.pdf` · Informe seguretat: `docs/SECURITY-AUDIT-OEC-APIS.md`