

Guia de Integração com a API de Emissão de NFSe

Visão Geral

Este guia descreve o fluxo completo para consumir a API de emissão de NFSe, desde a autenticação do operador até o processamento em massa de DPS. Todos os exemplos de requisição utilizam JSON e assumem codificação UTF-8.

1. Autorização da API Pública (Processos em Massa)

Endpoints sob `/api/public/**` exigem **Basic Auth**.

Envie o cabeçalho:

Authorization: Basic base64(usuario:senha)

Sem credenciais válidas a API retorna 401 com cabeçalho `WWW-Authenticate`.

2. Cadastro de Tomador (opcional)

- `POST /api/public/tomadores`
- Payload segue o `publicTomadorCreateSchema`, que estende o cadastro de tomador exigindo `prestadorId`.

Exemplo (tomador nacional):

```
``json
{
  "prestadorId": "2f1b3d8c-4d5e-4c2f-8a11-123456789abc",
  "tipoTomador": "NACIONAL",
  "tipoDocumento": "CNPJ",
  "documento": "12.345.678/0001-99",
  "nomeRazaoSocial": "Cliente Exemplo LTDA",
  "email": "financeiro@cliente.com",
  "telefone": "(11) 99999-0000",
  "codigoMunicipio": "3550308",
  "cidade": "São Paulo",
  "estado": "SP",
  "cep": "01001000",
  "logradouro": "Praça da Sé",
  "numero": "100",
  "bairro": "Sé"
}
```

Validações garantem consistência (CPF/CNPJ, CEP, UF, etc.). Sucesso retorna `{ "id": "<UUID>" }`.

3. Criação de DPS (Declaração para Prestação de Serviços)

- `POST /api/public/dps`
- Payload base (`dpsCreateSchema`):
- `json`
- `{`
- `"prestadorId": "2f1b3d8c-4d5e-4c2f-8a11-123456789abc",`
- `"tomadorId": "d6a8c877-9f10-4b19-9edc-abcdef123456",`
- `"servicoId": "f3bcac71-2b9c-4c01-8b93-abcdef987654",`
- `"competencia": "2026-02-01T00:00:00.000Z",`
- `"dataEmissao": "2026-02-10T12:15:00.000Z",`
- `"tipoEmissao": 1,`
- `"observacoes": "Serviço recorrente"`
- `}`
- `...`
- O backend gera identificador, número, XML e atualiza o contador da configuração do prestador.
- Quando não houver tomador identificado, envie `tomadorNaoIdentificado: true` (o XML sai sem a tag `ctoma` e fluxos posteriores usam o email do prestador para notificações).
- Resposta (201) contém informações completas da DPS criada.

4. Tomador e DPS em uma única chamada (opcional)

```
- POST /api/public/tomadores-com-dps
- Payload combina objetos `tomador` e `dps` mínimos. Exemplo:
``json
{
  "tomador": {
    "prestadorId": "2f1b3d8c-4d5e-4c2f-8a11-123456789abc",
    "tipoTomador": "NACIONAL",
    "tipoDocumento": "CPF",
    "documento": "123.456.789-09",
    "nomeRazaoSocial": "Maria da Silva",
    "email": "maria@cliente.com",
    "codigoMunicipio": "3550308",
    "cidade": "São Paulo",
    "estado": "SP",
    "cep": "01311000",
    "logradouro": "Av. Paulista",
    "numero": "1000",
    "bairro": "Bela Vista"
  },
  "dps": {
    "prestadorId": "2f1b3d8c-4d5e-4c2f-8a11-123456789abc",
    "servicoId": "f3bcac71-2b9c-4c01-8b93-abcdef987654",
    "competencia": "2026-02-01T00:00:00.000Z",
    "dataEmissao": "2026-02-10T12:15:00.000Z",
    "tipoEmissao": 1
  }
}
```

Sucesso retorna `{ "tomadorId": "...", "dpsId": "..." }`.

5. Tratamento de Erros Comuns

- **400 - Validações Zod**: payload inválido retorna detalhes em `issues`.
- **401 / 403 - Autenticação**: credenciais Basic Auth ausentes ou divergentes (usuário/senha diferentes de `PUBLIC_API_USER`/`PUBLIC_API_PASSWORD`).
- **404 - Certificados/DPS**: certificado não disponível para o prestador ou DPS inexistente/inativa.
- **Erros SEFIN**: respostas com `statusCode >= 400` incluem o XML bruto em `rawResponseContent` para análise.

6. Boas Práticas

1. Realize a criação ou atualização de tomadores e DPS antes do lote, reduzindo validações em massa.
2. Ao usar `/api/public/processar-dps`, registre o resultado `steps` para reprocessar apenas o necessário.
3. Monitore e registre o retorno de `/api/public/processar-dps` para diagnosticar rapidamente falhas de assinatura ou emissão.
4. Armazene as respostas e logs relevantes (por exemplo, `steps`, mensagens de erro e XMLs retornados) para auditorias e suporte.
