

Documentação de Endpoints da API - Pegasus NFSe

Localização dos Arquivos

Arquivo Principal: `server/nfse-api.ts`

Contém todas as funções de integração com a API Nacional e endpoints de download.

Caminho completo: `/home/ubuntu/portal_xml/server/nfse-api.ts`

Endpoints Configuráveis

1. API Nacional NFSe (Download de XMLs)

Localização no código: Linhas iniciais de `nfse-api.ts`

```
const NFSE_API_BASE = "https://adn.nfse.gov.br";
```

Funções que usam:

- `fetchNfseDocuments()` - Consulta documentos por NSU
- `downloadAllDocuments()` - Download com paginação e filtros
- `fetchNfseDocumentsSingle()` - Requisição única

URL Completa: `https://adn.nfse.gov.br/contribuintes/DFe/{NSU}?cnpjConsulta={CNPJ}&lote=true`

2. API de DANFSe (Download de PDFs)

Localização no código: Linhas iniciais de `nfse-api.ts`

```
const DANFSE_API_BASE = "https://adn.nfse.gov.br/danfse";
```

Funções que usam:

- `fetchDanfsePdf()` - Download do PDF da DANFSe
- `getDanfseUrl()` - Gera URL do PDF

URL Completa: `https://adn.nfse.gov.br/danfse/{chaveAcesso}`

3. API de Consulta CNPJ (BrasilAPI)

Localização no código: `server/routers.ts` (função `consultarReceitaEmLote`)

```
const BRASIL_API_URL = "https://brasilapi.com.br/api/cnpj/v1/{cnpj}";
```

Alternativa (ReceitaWS):

```
const RECEITA_WS_URL = "https://receitaws.com.br/v1/cnpj/{cnpj}";
```



Como Trocar um Endpoint no Futuro

Passo 1: Abrir o arquivo

```
nano /home/ubuntu/portal_xml/server/nfse-api.ts
```

Passo 2: Localizar a constante

Procure por:

- NFSE_API_BASE (para API de XMLs)
- DANFSE_API_BASE (para API de PDFs)

Passo 3: Alterar o valor

Exemplo - se a API mudar para novo domínio:

```
// ANTES
const NFSE_API_BASE = "https://adn.nfse.gov.br";

// DEPOIS
const NFSE_API_BASE = "https://novo-dominio.gov.br";
```

Passo 4: Salvar e reiniciar

```
cd /home/ubuntu/portal_xml
pnpm dev
```

Autenticação

mTLS (Mutual TLS)

Todos os endpoints usam certificado digital do cliente (mTLS):

Função: `createMtlsAgent(certPem, keyPem)`

```
function createMtlsAgent(certPem: string, keyPem: string): https.Agent {  
    return new https.Agent({  
        cert: certPem,  
        key: keyPem,  
        rejectUnauthorized: true,  
        minVersion: "TLSv1.2",  
    });  
}
```

O certificado é extraído do arquivo `.pfx` do cliente e usado em todas as requisições.



Retry Strategy

Para XMLs (fetchNfseDocuments)

- **Tentativas:** 3 retries
- **Backoff:** Exponencial (3s, 6s, 12s)
- **Erros que trigam retry:** HTTP 429, 5xx, erros de rede

Para PDFs (fetchDanfsePdf)

- **Tentativas:** 5 retries
- **Backoff:** Exponencial (2s, 4s, 8s, 16s, 32s)
- **Erros que trigam retry:** HTTP 429, 503, erros de rede, timeout
- **Erros que NÃO fazem retry:** HTTP 404 (PDF não existe)

Validação Final

- Se houver divergência XML vs PDF após download
- Sistema faz **3 tentativas finais adicionais** com delays de 3s, 6s, 9s

Funções Principais

`fetchNfseDocuments(certPem, keyPem, cnpj, startNsu)`

Busca documentos da API Nacional com retry automático.

Parâmetros:

- `certPem` : Certificado em formato PEM
- `keyPem` : Chave privada em formato PEM
- `cnpj` : CNPJ do cliente
- `startNsu` : NSU inicial (padrão: 1)

Retorno: `NfseApiResponse` com lista de documentos

`fetchDanfsePdf(certPem, keyPem, chaveAcesso, retryCount, maxRetries)`

Baixa o PDF da DANFSe com retry agressivo.

Parâmetros:

- `certPem` : Certificado em formato PEM
- `keyPem` : Chave privada em formato PEM
- `chaveAcesso` : Chave de acesso da NFSe
- `retryCount` : Tentativa atual (interno, começa em 0)
- `maxRetries` : Máximo de tentativas (padrão: 5)

Retorno: `Buffer | null` (PDF ou null se falhar)

`downloadAllDocuments(certPem, keyPem, cnpj, startNsu, onProgress, periodFilter)`

Orquestração completa de download com paginação e filtros.

Parâmetros:

- `certPem` : Certificado em formato PEM
- `keyPem` : Chave privada em formato PEM
- `cnpj` : CNPJ do cliente
- `startNsU` : NSU inicial
- `onProgress` : Callback para atualizar progresso
- `periodFilter` : Filtro de período (competência ou data)

Retorno: Array de `ParsedNfse` (documentos parseados)



Logging

Todos os eventos são logados com prefixos:

- `[API NFSe]` - Eventos da API de XMLs
- `[PDF]` - Eventos de download de PDFs
- `[Validação]` - Eventos de validação XML vs PDF

Exemplo de log:

```
[PDF] 3534.01.02.000000000000650110001234567890123: HTTP 503 - aguardando
4000ms antes de retry 2/5
[Validação] Empresa XYZ: DIVERGÊNCIA DETECTADA - 50 XMLs vs 45 PDFs
(diferença: 5)
[Validação] Empresa XYZ: PDF recuperado na tentativa 1 -
3534.01.02.000000000000650110001234567890123
```

Tratamento de Erros

Erros Comuns e Soluções

Erro	Causa	Solução
HTTP 401	Certificado não autorizado	Verificar validade do certificado
HTTP 403	Acesso negado	Verificar permissões do CNPJ
HTTP 404	Documento não encontrado	Documento não existe na API
HTTP 429	Rate limiting	Sistema faz retry automaticamente
HTTP 500/502/503	API indisponível	Sistema faz retry automaticamente
ECONNRESET	Conexão perdida	Sistema faz retry automaticamente
ETIMEDOUT	Timeout	Sistema faz retry automaticamente

Contato

Para dúvidas sobre endpoints ou mudanças futuras, consulte:

- **Arquivo:** `/home/ubuntu/portal_xml/server/nfse-api.ts`
- **Documentação:** Este arquivo (`API_ENDPOINTS.md`)
- **Logs:** `.manus-logs/devserver.log`

Última atualização: Fevereiro 2026 **Versão:** 1.0