

Guia Completo e Amigável: API REST para Acesso Simplificado ao MNI PJe

Documentação Fornecida

10 de maio de 2025

Bem-vindo! Este guia foi criado para ajudar você a entender e utilizar nossa API REST, que simplifica a interação com o sistema de Processo Judicial Eletrônico (PJe) do Brasil através do Modelo Nacional de Interoperabilidade (MNI). Mesmo que termos como "API", "JSON" ou "Chave de API" pareçam complicados, vamos explicá-los de forma fácil!

Conteúdo

1 Contexto: O MNI e o PJe	3
2 Entendendo o Mundo das APIs: Uma Explicação para Todos	3
2.1 O que é uma API REST? (Seu Assistente Pessoal Digital!)	3
2.2 O que é JSON? (A Forma Como o Assistente Anota e Entrega as Informações)	4
2.3 O que são Chaves de API (API Keys)? (Seu Crachá de Acesso Exclusivo!)	4
3 Nossa API REST: Simplificando seu Acesso ao MNI PJe	5
3.1 Como Nossa API Facilita Sua Vida	5
3.2 Autenticação: Seu "Crachá" para Nossa API	6
3.3 Endpoints da Nossa API REST	6
3.4 Exemplos Práticos de Uso	7
3.5 Estrutura de Resposta Detalhada da API (JSON com Explicações)	9

3.6	Tratamento de Erros	11
3.7	Considerações Importantes ao Usar Nossa API	12
4	Nos Bastidores: O Sistema MNI/PJe (O "Motor"da Nossa API)	12
4.1	O Que é o MNI e o PJe? (Relembrando)	12
4.2	Por que o Acesso Direto ao MNI/SOAP é Complicado?	12
4.3	Nossa API REST vs. Acesso Direto MNI/SOAP	13
4.4	Tribunais e Como Nossa API Lida Com Eles	13
5	Conclusão	14
6	Suporte e Contato	15

1 Contexto: O MNI e o PJe

(Esta seção é baseada no seu texto original para fornecer o contexto fundamental.)

O **Modelo Nacional de Interoperabilidade (MNI)** representa o padrão estabelecido para troca de informações processuais no âmbito do Poder Judiciário brasileiro. Este modelo permite a integração entre sistemas por meio da tecnologia WebService (originalmente SOAP), garantindo a integridade, inviolabilidade e segurança dos processos judiciais, inclusive quanto ao sigilo processual quando necessário.

O **Processo Judicial Eletrônico (PJe)** é o sistema de tramitação de processos judiciais desenvolvido pelo Conselho Nacional de Justiça (CNJ) para atender às necessidades dos diversos segmentos do Poder Judiciário brasileiro.

A integração original MNI-PJe é feita via WebServices SOAP/WSDL. Embora poderosa, essa abordagem pode ser tecnicamente complexa. **Nossa API REST foi desenvolvida para ser uma ponte, tornando esse acesso muito mais simples e moderno.**

2 Entendendo o Mundo das APIs: Uma Explicação para Todos

Antes de usarmos nossa API, vamos entender alguns conceitos básicos de forma bem simples.

2.1 O que é uma API REST? (Seu Assistente Pessoal Digital!)

Imagine que a internet é uma cidade gigante, cheia de lojas, bibliotecas e escritórios (que são os diferentes sistemas e sites, como o PJe).

- **Você (Seu Aplicativo/Software):** Você quer uma informação específica ou realizar uma tarefa (ex: "Preciso dos dados do processo número 12345").
- **A API (Interface de Programação de Aplicativos):** Pense nela como um **assistente pessoal muito eficiente e bem treinado** para um serviço específico. Você não vai direto ao "depósito" do PJe; você chama esse assistente.
- **O Pedido (Requisição):** É a instrução clara que você dá ao assistente (ex: "Assistente, por favor, busque para mim o processo número 12345").
- **O Sistema de Origem (Servidor PJe):** É onde a informação está guardada (como uma grande biblioteca ou arquivo).
- **A Entrega (Resposta):** O assistente volta com a informação que você pediu, de forma organizada (ex: "Aqui estão os dados do processo 12345").

REST (Representational State Transfer) é um conjunto de regras que faz esses "assistentes"(APIs) serem padronizados e fáceis de trabalhar. Eles usam:

- **Endereços Web (URLs):** Para saber onde encontrar o serviço (como o endereço da "biblioteca").
- **"Verbos"HTTP comuns:** Para dizer o que querem fazer. O mais comum para nós será:
 - **GET:** Para buscar/ler informações.

Nossa API REST é, portanto, um "assistente digital"que você pode usar para "pedir"informações sobre processos judiciais do PJe de uma maneira muito mais fácil do que falar diretamente com o sistema complexo do MNI.

2.2 O que é JSON? (A Forma Como o Assistente Anota e Entrega as Informações)

Quando seu "assistente"(a API) traz a informação para você, ela precisa estar num formato que você entenda facilmente. **JSON (JavaScript Object Notation)** é uma maneira super popular de organizar essas informações, como uma ficha de cadastro ou uma lista de compras bem detalhada.

É um formato de texto, leve e fácil para humanos lerem e para computadores processarem. Ele usa "rótulos"(chaves) e "informações"(valores).

Exemplo simples de JSON (informações de um contato):

```
1 {  
2   "nome": "Maria Silva",  
3   "telefone": "(11) 98765-4321",  
4   "email": "maria.silva@email.com",  
5   "idade": 30, // Valores podem ser n meros  
6   "clienteAtivo": true // Valores podem ser verdadeiro/falso  
7 }
```

Listing 1: Exemplo JSON: Informações de Contato

No exemplo acima, **"nome"** é uma chave (o rótulo), e **"Maria Silva"** é o valor (a informação). Nossa API REST vai te entregar os dados dos processos nesse formato JSON amigável! Veremos exemplos mais detalhados adiante.

2.3 O que são Chaves de API (API Keys)? (Seu Crachá de Acesso Exclusivo!)

Continuando com a analogia da cidade e da biblioteca (PJe): Algumas áreas da biblioteca ou alguns livros raros só podem ser acessados por pessoas autorizadas. Uma **Chave de API (API Key)** é como um **crachá de acesso especial e exclusivo** ou a **chave do seu**

quarto de hotel.

- **Identificação (Quem é você?):** O crachá diz à biblioteca quem está pedindo o livro (ex: "Eu sou o usuário com CPF X e Senha Y").
- **Autorização (O que você pode fazer/ver?):** A biblioteca verifica se esse "crachá"(suas credenciais) dá permissão para acessar aquele livro específico.
- **Segurança:** Garante que só pessoas/aplicativos autorizados acessem as informações, protegendo dados sensíveis.
- **Controle:** Permite ao sistema registrar quem está acessando e o quê.

Como você usa essa "chave" com nossa API? No nosso caso, suas **credenciais de acesso ao PJe (CPF/CNPJ e Senha)** funcionam como sua "Chave de API". Você as envia de forma segura junto com cada pedido (requisição) que faz à nossa API. A API então usa essas credenciais para acessar o PJe em seu nome.

MUITO IMPORTANTE: Suas credenciais do PJe são extremamente sensíveis.

- **NUNCA** as compartilhe desnecessariamente.
- **NUNCA** as coloque diretamente em códigos que rodam no navegador do usuário (frontend).
- Utilize-as apenas em ambientes seguros (backend/servidor) ao chamar nossa API.

3 Nossa API REST: Simplificando seu Acesso ao MNI PJe

Agora que entendemos os conceitos, vamos ver como nossa API específica funciona.

3.1 Como Nossa API Facilita Sua Vida

Nossa API REST atua como uma camada tradutora e simplificadora:

1. Você faz um pedido simples para nossa API usando um endereço web (URL) e os métodos REST (como **GET**).
2. Você envia suas credenciais do PJe (seu "crachá") de forma segura nos cabeçalhos do pedido.
3. Nossa API "conversa" com o sistema MNI/PJe nos bastidores, lidando com toda a complexidade do SOAP/XML.
4. Ela então te devolve a resposta de forma organizada e fácil de usar, em formato JSON.

Isso significa menos código complicado para você escrever e uma integração mais rápida!

3.2 Autenticação: Seu "Crachá" para Nossa API

Para usar nossa API, você precisa se autenticar enviando suas credenciais do PJe em cada requisição, através de **cabeçalhos HTTP (HTTP Headers)**. Pense nos cabeçalhos como um envelope onde você coloca seu crachá junto com seu pedido.

Os cabeçalhos necessários são:

- **X-MNI-CPF:** Seu CPF ou CNPJ (apenas números) cadastrado e autorizado no PJe.
- **X-MNI-SENHA:** Sua senha de acesso ao PJe.

Exemplo de como seriam esses cabeçalhos em uma ferramenta de teste ou no seu código:

X-MNI-CPF: 12345678900

X-MNI-SENHA: SuaSenhaSuperSegura123

3.3 Endpoints da Nossa API REST

"Endpoints" são os endereços web (URLs) específicos que você usará para diferentes ações ou para buscar diferentes tipos de informação.

URL Base da API: <https://machine-learning-toolkit-marcosmarf27.replit.app> (Todos os endpoints abaixo são caminhos que você adiciona após esta URL base)

Tabela 1: Endpoints da API REST

Endpoint (Caminho após a URL Base)	Método	Descrição
/api/v1/processo/{num_processo}	GET	Retorna dados completos do processo, incluindo metadados e documentos.
/api/v1/processo/{num_processo}/documento/{id_doc}	GET	Realiza o download de um documento específico do processo.
/api/v1/processo/{num_processo}/peticao-inicial	GET	Retorna a petição inicial e seus documentos anexos.
/api/v1/processo/{num_processo}/documentos/ids	GET	Lista os IDs e descrições de todos os documentos do processo.
/api/v1/processo/{num_processo}/capa	GET	Retorna apenas os metadados (informações da capa) do processo.

Substituições:

- `{num_processo}`: Substitua pelo número do processo no formato CNJ (ex: `01234567820238060001` ou `0123456-78.2023.8.06.0001` - a API deve ser flexível a traços e pontos, mas prefira apenas números se possível).
- `{id_doc}`: Substitua pelo ID do documento que você deseja baixar (obtido, por exemplo, na consulta completa do processo).

3.4 Exemplos Práticos de Uso

Exemplo 1: Consultando um Processo Completo (usando Python)

```

1 import requests
2 import json
3
4 # --- Configura es Essenciais ---
5 API_URL_BASE = "https://machine-learning-toolkit-marcosmarf27.replit.app/api/v1"
6 SEU_CPF = "SEU_CPF_AQUI_SEM_PONTOS_OU_TRACOS" # << SUBSTITUA AQUI
7 SUA_SENHA_PJE = "SUA_SENHA_PJE_AQUI" # << SUBSTITUA AQUI
8 NUMERO_PROCESSO_CNJ = "07108025520188020001" # << EXEMPLO, SUBSTITUA PELO PROCESSO DESEJADO
9
10 # --- Montando os Cabe alhos para Autentica o ---
11 headers = {
12     'X-MNI-CPF': SEU_CPF,
13     'X-MNI-SENHA': SUA_SENHA_PJE,
14     'Accept': 'application/json' # Informa que esperamos uma resposta JSON
15 }
16
17 # --- Montando a URL completa do Endpoint ---
18 url_consulta = f"{API_URL_BASE}/processo/{NUMERO_PROCESSO_CNJ}"
19
20 print(f"Fazendo requisit o GET para: {url_consulta}")
21
22 try:
23     # --- Fazendo a Requisit o ---
24     response = requests.get(url_consulta, headers=headers, timeout=30) # timeout de 30 segundos
25
26     # --- Verificando se a requisit o foi bem-sucedida (c digo HTTP 2xx) ---
27     response.raise_for_status() # Se houver erro HTTP (4xx ou 5xx), isso lan ar uma exce o
28
29     # --- Convertendo a Resposta JSON em um objeto Python ---
30     dados_do_processo = response.json()

```

```

31
32 print("\n--- Resposta da API (JSON) ---")
33 print(json.dumps(dados_do_processo, indent=2, ensure_ascii=False)) # Imprime o JSON formatado
34
35 # --- Exemplo de como acessar informa es especificas ---
36 if dados_do_processo.get("sucesso"):
37     print(f"\nConsulta bem-sucedida para o processo: {dados_do_processo.get('processo', {}).get('dadosBasicos',
38     {}).get('numero')}")
39     if 'documentos' in dados_do_processo.get('processo', {}):
40         print(f"N mero de documentos encontrados: {len(dados_do_processo['processo']['documentos'])}")
41     else:
42         print("N o foram encontrados documentos ou a chave 'documentos' est ausente.")
43 else:
44     print(f"\nA API retornou sucesso=false: {dados_do_processo.get('mensagem')}")
45
46 except requests.exceptions.HTTPError as http_err:
47     print(f"\nErro HTTP: {http_err}")
48     print(f"C digo de Status: {response.status_code}")
49     try:
50         # Tenta imprimir o corpo da resposta do erro, que pode ser JSON
51         print(f"Corpo da Resposta (Erro): {json.dumps(response.json(), indent=2, ensure_ascii=False)}")
52     except json.JSONDecodeError:
53         print(f"Corpo da Resposta (Erro n o JSON): {response.text}")
54 except requests.exceptions.Timeout:
55     print("\nA requisiti o demorou demais e excedeu o tempo limite (Timeout).")
56 except requests.exceptions.RequestException as req_err:
57     print(f"\nErro na Requisiti o: {req_err}")
58 except json.JSONDecodeError:
59     print("\nErro ao decodificar a resposta JSON. A resposta n o um JSON v lido.")
60     print(f"Conte do da resposta recebida: {response.text}")

```

Listing 2: Python: Consultando Processo Completo

Exemplo 2: Baixando um Documento Específico (usando cURL na linha de comando) Suponha que, após consultar o processo, você identificou que o documento com `id_doc = "12345678"` é o que você quer.

```

1 curl -X GET "https://machine-learning-toolkit-marcosmarf27.replit.app/api/v1/processo/07108025520188020001/documento
2 /12345678" \
3     -H "X-MNI-CPF: SEU_CPF_AQUI" \
4     -H "X-MNI-SENHA: SUA_SENHA_PJE_AQUI" \
5     -o documento_baixado.pdf # Salva o arquivo como documento_baixado.pdf

```

```
5
6 echo "Download concluído. Arquivo salvo como documento_baixado.pdf (se o documento for PDF)."
```

Listing 3: cURL: Baixando Documento

Nota: O nome do arquivo (`documento_baixado.pdf`) é um exemplo. O tipo de arquivo real dependerá do documento no processo. A API deve retornar o tipo de conteúdo correto (ex: `application/pdf`, `image/jpeg`) no cabeçalho `Content-Type` da resposta.

3.5 Estrutura de Resposta Detalhada da API (JSON com Explicações)

Nossa API responde em formato JSON. Veja como são as respostas:

Exemplo de Resposta de SUCESSO ao Consultar um Processo (`/api/v1/processo/{num_processo}`):

```
1 {
2   "sucesso": true, // true significa: "Conseguimos o que voc pediu!"
3   "mensagem": "Processo consultado com sucesso.", // Uma mensagem amig vel.
4   "processo": { // O "pacote" principal com todos os dados do processo.
5     "dadosBasicos": { // Informa es gerais, como a "capa" do processo.
6       "numero": "07108025520188020001", // O n mero nico do processo (formato CNJ).
7       "classeProcessualNome": "PROCEDIMENTO COMUM C VEL", // Tipo da a o (ex: Div rcio, Cobran a).
8       "codigoClasseProcessual": "7", // C digo num rico para esse tipo de a o.
9       "dataAjuizamento": "2018-03-15T14:30:25Z", // Quando o processo come ou (data e hora UTC).
10      "nivelSigilo": 0, // N vel de segredo (0 = P blico, 1 = Segredo de Justi a, etc.).
11      "orgaoJulgador": { // Quem est julgando o caso.
12        "codigoOrgao": "8020001", // C digo da Vara/Comarca.
13        "nomeOrgao": "1 Vara C vel da Capital", // Nome da Vara/Comarca.
14        "instancia": "ORIG" // Se 1 instancia ("ORIG"), 2 instancia ("RECURSAL"), etc.
15      },
16      "assuntos": [ // Lista dos temas principais tratados no processo.
17        {
18          "codigoNacional": "10944", // C digo CNJ do assunto.
19          "descricao": "Indeniza o por Dano Moral" // Descri o do assunto.
20        }
21        // Pode haver outros assuntos aqui.
22      ]
23      // ... Outros campos importantes como 'valorCausa', 'tribunal', etc., podem estar aqui.
24    },
25    "documentos": [ // Uma lista de todos os documentos dentro do processo.
26      {
27        "idDocumento": "23011510000012300000123456789", // ID nico deste documento.
```

```

28 "tipoDocumentoNome": "PETIÇÃO INICIAL", // O que este documento (ex: Petição, Sentença).
29 "tipoDocumentoCodigo": "1", // Código do tipo de documento.
30 "descricao": "Petição inicial referente ao caso de reparação de danos.", // Descrição opcional.
31 "dataHoraInclusao": "2018-03-15T14:35:10Z", // Quando o documento foi adicionado.
32 "mimetype": "application/pdf", // Tipo de arquivo (PDF, JPG, HTML, etc.).
33 "nivelSigilo": 0, // Nível de sigilo deste documento específico.
34 "documentosVinculados": [ // Documentos que são "anexos" deste.
35   {
36     "idDocumento": "23011510000012400000123456790",
37     "tipoDocumentoNome": "PROCURAÇÃO",
38     "descricao": "Procuração do autor para o advogado."
39     // ... mais detalhes do documento vinculado
40   }
41 ]
42 },
43 {
44   "idDocumento": "23022011000023400000123456001",
45   "tipoDocumentoNome": "CONTESTAÇÃO",
46   "dataHoraInclusao": "2018-04-10T11:20:00Z",
47   "mimetype": "application/pdf",
48   "nivelSigilo": 0
49 }
50 // ... Pode haver muitos outros documentos.
51 ],
52 "polos": [ // Quem são as partes envolvidas (Autores, Réus).
53   {
54     "polo": "poloAtivo", // "Lado" da parte (Ativo = quem entra com a ação).
55     "parte": [ // Lista de pessoas/empresas neste polo.
56       {
57         "nome": "Fulano de Tal",
58         "tipoPessoa": "FISICA", // Se pessoa física ou jurídica.
59         "numeroDocumentoPrincipal": "11122233344" // CPF ou CNPJ.
60       }
61     ]
62   },
63   {
64     "polo": "poloPassivo", // "Lado" da parte (Passivo = contra quem a ação foi movida).
65     "parte": [
66       {
67         "nome": "Empresa Exemplo Ltda",

```

```

68         "tipoPessoa": "JURIDICA",
69         "numeroDocumentoPrincipal": "00111222000199"
70     }
71 ]
72 }
73 ]
74 // ... Outras se es como 'movimentosProcessuais' podem estar presentes.
75 }
76 }

```

Listing 4: JSON: Resposta de Sucesso - Consulta de Processo

Exemplo de Resposta de ERRO:

```

1 {
2 "sucesso": false, // false significa: "Algo deu errado."
3 "erro": "AUTENTICACAO_INVALIDA", // Um c digo curto para o tipo de erro.
4 "mensagem": "As credenciais (CPF/Senha) fornecidas s o invlidas ou o usu rio n o tem permiss o para acessar o
   processo solicitado no PJe.", // Explica o do que aconteceu.
5 "detalhes": { // Informa es extras que podem ajudar a depurar.
6     "codigo_http_retornado": 401,
7     "sugestao": "Verifique se seu CPF e Senha (X-MNI-CPF, X-MNI-SENHA) est o corretos e se foram enviados nos
   cabe alhos da requis i o. Confirme suas permiss es de acesso no sistema PJe."
8 }
9 }

```

Listing 5: JSON: Resposta de Erro

3.6 Tratamento de Erros

Nossa API usa códigos de status HTTP padrão para indicar o resultado da requisição:

- **200 OK:** Sucesso! A requisição foi bem-sucedida.
- **400 Bad Request:** Requisição inválida. Verifique os parâmetros enviados (ex: número do processo malformatado). O JSON da resposta geralmente tem mais detalhes.
- **401 Unauthorized:** Autenticação falhou. Suas credenciais (**X-MNI-CPF**, **X-MNI-SENHA**) estão erradas, ausentes ou seu usuário não tem permissão no PJe para o que foi solicitado.
- **403 Forbidden:** Proibido. Mesmo autenticado, você não tem permissão para acessar *este* recurso específico.
- **404 Not Found:** Não encontrado. O processo ou documento solicitado não existe.

- **429 Too Many Requests:** Você fez requisições demais em um curto período. Aguarde um pouco antes de tentar novamente.
- **500 Internal Server Error:** Erro no nosso servidor. Algo inesperado aconteceu do nosso lado ou na comunicação com o PJe. Tente novamente mais tarde. Se persistir, entre em contato com o suporte.
- **502 Bad Gateway / 503 Service Unavailable / 504 Gateway Timeout:** Problemas temporários de comunicação entre nossa API e o sistema MNI do tribunal. Tente novamente mais tarde.

Sempre verifique o código de status E o corpo da resposta JSON para entender completamente o resultado.

3.7 Considerações Importantes ao Usar Nossa API

- **Segurança das Credenciais:** REFORÇANDO: Suas credenciais **X-MNI-CPF** e **X-MNI-SENHA** são secretas. Proteja-as!
- **Limites de Uso (Rate Limiting):** Para proteger os sistemas, pode haver limites na frequência de chamadas à API. Faça chamadas conscientes.
- **Timeouts:** Consultas complexas ou downloads grandes podem demorar. Configure timeouts adequados no seu cliente HTTP (ex: 30-60 segundos).
- **Tamanho das Respostas:** Processos podem ter muitos dados. Prepare sua aplicação para lidar com JSONs grandes ou downloads de arquivos volumosos.
- **Permissões do PJe:** A API respeita as permissões do seu usuário no PJe. Você só verá o que seu usuário PJe pode ver.
- **Disponibilidade dos Tribunais:** O acesso depende da disponibilidade do serviço MNI do tribunal específico.

4 Nos Bastidores: O Sistema MNI/PJe (O "Motor" da Nossa API)

Entender um pouco do sistema "por trás" da nossa API ajuda a valorizar a simplificação que ela oferece.

4.1 O Que é o MNI e o PJe? (Relembrando)

- **PJe:** O sistema eletrônico dos processos.
- **MNI:** O padrão técnico (originalmente SOAP/WebServices) para sistemas externos conversarem com o PJe.

4.2 Por que o Acesso Direto ao MNI/SOAP é Complicado?

Interagir diretamente com o MNI/SOAP é como tentar montar um motor complexo sem manual:

- **Protocolo SOAP:** Usa XML, um formato de dados muito detalhado e "pesado".

```

1 <!-- Exemplo MUITO simplificado de um pedaço de XML SOAP -->
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
3   <soap:Header>...</soap:Header>
4   <soap:Body>
5     <ns1:consultarProcesso xmlns:ns1="http://www.cnj.jus.br/mni">
6       <idConsultante>SEU_CPF</idConsultante>
7       <senhaConsultante>SUA_SENHA</senhaConsultante>
8       <numeroProcesso>NUM_PROCESSO</numeroProcesso>
9     </ns1:consultarProcesso>
10  </soap:Body>
11 </soap:Envelope>
12

```

Listing 6: Exemplo Simplificado XML SOAP

Compare isso com a simplicidade de uma URL REST e cabeçalhos!

- **WSDL:** É um arquivo descritivo (o "manual do motor") que define como o serviço SOAP funciona. É técnico e difícil de ler.
- **Ferramentas Específicas:** Requer bibliotecas de programação especializadas em SOAP.

4.3 Nossa API REST vs. Acesso Direto MNI/SOAP

4.4 Tribunais e Como Nossa API Lida Com Eles

O Brasil tem muitos tribunais (Estaduais, Federais, etc.), e cada um deles (ou até cada instância/comarca) pode ter seu próprio "endereço técnico"(endpoint WSDL do MNI/SOAP) para acesso direto.

- **O Desafio:** Saber qual o WSDL correto para cada processo seria um grande trabalho manual. A lista de WSDLs é extensa e pode mudar (exemplos como <https://pje.tjce.jus.br/pje1grau/intercomunicacao?wsdl>, <https://pje.trf1.jus.br/pje2g/intercomunicacao?wsdl>, etc.).
- **A Simplificação da Nossa API REST:**
 - **Você informa apenas o número do processo.**
 - **Nossa API faz o trabalho pesado:** Idealmente, nossa API contém lógica para, a partir do número do processo (que segue um padrão nacional do CNJ), identificar o tribunal de origem e se conectar ao sistema MNI/SOAP correto daquele tribunal específico nos bastidores.

Tabela 2: Comparativo: API REST vs. Acesso Direto MNI/SOAP

Característica	Acesso Direto MNI/SOAP	Nossa API REST	Vantagem para Você
Formato de Dados	XML (complexo, verboso)	JSON (simples, leve, fácil de ler)	Mais fácil de usar e entender.
Protocolo	SOAP (regras rígidas, pesado)	HTTP/REST (padrões web, mais leve)	Integração mais simples.
Descoberta Serviço	WSDL (manual técnico)	URLs intuitivas (ex: <code>/processo/{num}</code>)	Fácil de usar, autoexplicativo.
Download Arquivos	Codificações complexas (MTOM/XOP)	Download HTTP direto	Simples como baixar um arquivo web.
Curva Aprendizado	Alta	Baixa a Moderada	Desenvolve mais rápido.

– **Você não precisa se preocupar com os diferentes endereços WSDL dos tribunais.** A URL da nossa API REST é única para você, e nós cuidamos da "mágica" de encontrar o caminho certo para o PJe daquele tribunal.

- **Quais Tribunais são Suportados?** Nossa API se esforça para cobrir os tribunais que utilizam o PJe e disponibilizam o MNI. A capacidade de consulta depende da disponibilidade e conformidade do serviço MNI de cada tribunal. Se você tiver problemas com um processo de um tribunal específico, entre em contato com o suporte da API.
- **Onde Encontrar Listas Oficiais de Endpoints MNI/SOAP (para curiosidade)?** O Conselho Nacional de Justiça (CNJ) é a fonte primária. Informações sobre os WebServices MNI podem ser encontradas no site do CNJ ou em portais para desenvolvedores do PJe.

Ambiente de Testes MNI (Para quem quer se aventurar no SOAP direto) O CNJ geralmente oferece um ambiente para testes diretos com MNI/SOAP (ex: <https://wwwh.cnj.jus.br/pjemni-2x/intercomunicacao?wsdl> – verifique a URL atual no site do CNJ). Nossa API REST já interage com os ambientes de produção dos tribunais.

5 Conclusão

O sistema MNI/PJe é essencial para a justiça digital no Brasil. Nossa API REST foi criada para ser sua aliada, tornando o acesso a esses dados processuais muito mais simples, rápido e moderno. Ao lidar com a complexidade do SOAP e oferecer uma interface amigável com

JSON, queremos que você possa construir aplicações incríveis e eficientes.

Lembre-se sempre da importância da segurança das suas credenciais e das boas práticas ao usar qualquer API. _____

6 Suporte e Contato

Se você tiver dúvidas, encontrar problemas ou tiver sugestões para melhorar nossa API REST:

- **Contato por whatsapp: (88) 993650420**

Esta documentação será atualizada conforme a API evolui. _____