**API Pública**

A API Pública do Datajud é uma ferramenta que permite o acesso público aos metadados de processos judiciais de todo o Brasil. Os dados disponibilizados pela API são originários da Base Nacional de Dados do Poder Judiciário (Datajud) e seguem os critérios estabelecidos pela [Portaria Nº 160 de 09/09/2020](https://atos.cnj.jus.br/atos/detalhar/3453) bem como o resguardo de processos sigilosos e dados de partes.

Através dessa API, desenvolvedores e pesquisadores têm acesso a informações de capas processuais e movimentações relacionadas aos processos judiciais de todas as instâncias do Judiciário brasileiro. Esses dados podem ser usados para diversos fins, como pesquisas acadêmicas, desenvolvimento de aplicativos que facilitem o acesso à informação jurídica ou até mesmo análise de tendências e padrões do sistema de Justiça; observando os critérios estabelecidos no [Termo de Uso](https://datajud-wiki.cnj.jus.br/api-publica/termo-uso).

**Termo de Uso**

Atenção!

API Pública do Datajud está estritamente sujeito às regras e condições estabelecidas em nosso [Termo de Uso](https://formularios.cnj.jus.br/wp-content/uploads/2023/11/Termos-de-uso-api-publica-V1.2.pdf). Além disso, reforçamos aqui algumas das responsabilidades quanto ao cosumo dos dados da API Pública:

* **1.2 O usuário manifestará tacitamente sua aceitação às condições deste termo ao utilizar a interface;**
* **3.2. O usuário da API pública se responsabiliza pelo uso da interface e das informações, pesquisas, documentos ou qualquer espécie de informação derivada de seu uso;**
* **3.3. A API é fornecida exclusivamente para fins legais, não comerciais e autorizados, sendo seu uso indevido, abusivo, ilegal, malicioso ou imoral estritamente proibido**;
* **3.6. O CNJ não garante a precisão, integridade ou atualidade dos dados fornecidos pela API;**
* **3.8. O usuário concorda em não modificar, distribuir, vender ou explorar comercialmente a API ou qualquer informação derivada dela;**
* **3.9. O usuário concorda em dar ciência ao CNJ de qualquer informação, notícia, estudo, relatório ou documento de qualquer natureza que seja disponibilizado ao público em geral.**

É importante destacar que o simples ato de consumir a API implica automaticamente na aceitação integral das cláusulas presentes no referido Termo de Uso. Recomendamos enfaticamente que todos os usuários revisem cuidadosamente o documento antes de utilizar a API para garantir a compreensão das responsabilidades, diretrizes e restrições associadas ao seu uso.

**Acesso**

A API Pública do Datajud é uma ferramenta que disponibiliza ao público o acesso aos metadados dos processos públicos dos tribunais do judiciário brasileiro. Os dados disponibilizados pela API são de origem da Base Nacional de Dados do Poder Judiciário – Datajud e atendem aos critérios estabelecidos [Portaria Nº 160 de 09/09/2020](https://atos.cnj.jus.br/atos/detalhar/3453).

**API Key**[**​**](https://datajud-wiki.cnj.jus.br/api-publica/acesso#api-key)

A autenticação da API Pública do Datajud é realizada através de uma **Chave Pública**, gerada e disponibilizada pelo DPJ/CNJ. A chave vigente estará sempre acessível nesta seção da Wiki, garantindo transparência e facilitando seu acesso. Importante ressaltar que, por razões de segurança e gestão do sistema, a chave poderá ser alterada pelo CNJ a qualquer momento.

Para incorporar a API Key em suas requisições, utilize o formato "Authorization: APIKey [Chave Pública]" no cabeçalho da requisição.

* **APIKey atual**:
  + Authorization: APIKey **cDZHYzlZa0JadVREZDJCendQbXY6SkJlTzNjLV9TRENyQk1RdnFKZGRQdw==**

***Endpoints***

A API Pública do Datajud oferece várias rotas para pesquisa de informações processuais devido à natureza e organização do Judiciário brasileiro.

A url principal de acesso é a url <https://api-publica.datajud.cnj.jus.br/> e deverá ser seguida de um aliase correspondente ao Tribunal que deseja obter os dados processuais.

|  |  |
| --- | --- |
| Tribunal de Justiça do Ceará | <https://api-publica.datajud.cnj.jus.br/api_publica_tjce/_search> |

**Exemplos**

As possibilidades de pesquisas na API Pública não estão restritas apenas aos exemplos aba seguir. Para a obtenção de resultados mais satisfatórios em suas pesquisas, recomenda-se a utilização dos diversos atributos dos metadados processuais listados no [Glossário de Dados](https://datajud-wiki.cnj.jus.br/api-publica/glossario) desta wiki.

**Ex. 1 - Pesquisar pelo número de processo**

No exemplo abaixo é realizada a consulta de um processo judicial utilizando a **numeração única do processo** como parâmetro de pesquisa no tribunal do TRF1.

**POST /api\_publica\_tribunal/\_search[​](https://datajud-wiki.cnj.jus.br/api-publica/exemplos/exemplo1" \l "post-api_publica_tribunal_search" \o "Link direto para POST /api_publica_tribunal/_search)**

* Postman
* Python
* R

1. Abra o **Postman** e clique em *"New Request"*.
2. Defina o método HTTP como POST.
3. Digite a URL: <https://api-publica.datajud.cnj.jus.br/api_publica_trf1/_search>
4. Selecione a aba *"Headers"* e inclua a chave "Authorization" com o valor "APIKey [Chave Pública]";
5. O valor [Chave Pública] corresponde a chave pública disponível em [Chave Pública](https://datajud-wiki.cnj.jus.br/api-publica/exemplos/api-publica/acesso);
6. Ainda em *"Headers"* inclua a chave "Content-Type" com o valor "application/json";
7. Selecione a aba *"Body"* e escolha a opção *"raw"*. Insira o corpo da solicitação JSON conforme o exemplo abaixo:

Query DSL

{  
 "query": {  
 "match": {  
 "numeroProcesso": "00008323520184013202"  
 }  
 }  
}

1. Clique em *Send* para enviar e aguarde a resposta da Api.

**Resposta**[**​**](https://datajud-wiki.cnj.jus.br/api-publica/exemplos/exemplo1#resposta)

A resposta esperado é um JSON com os metadados de 1 ou mais processos conforme o critério da busca:

JSON com os metadados processuais

{  
 "took": 6679,  
 "timed\_out": false,  
 "\_shards": {  
 "total": 7,  
 "successful": 7,  
 "skipped": 0,  
 "failed": 0  
 },  
 "hits": {  
 "total": {  
 "value": 1,  
 "relation": "eq"  
 },  
 "max\_score": 13.917725,  
 "hits": [  
 {  
 "\_index": "api\_publica\_trf1",  
 "\_type": "\_doc",  
 "\_id": "TRF1\_436\_JE\_16403\_00008323520184013202",  
 "\_score": 13.917725,  
 "\_source": {  
 "numeroProcesso": "00008323520184013202",  
 "classe": {  
 "codigo": 436,  
 "nome": "Procedimento do Juizado Especial Cível"  
 },  
 "sistema": {  
 "codigo": 1,  
 "nome": "Pje"  
 },  
 "formato": {  
 "codigo": 1,  
 "nome": "Eletrônico"  
 },  
 "tribunal": "TRF1",  
 "dataHoraUltimaAtualizacao": "2023-07-21T19:10:08.483Z",  
 "grau": "JE",  
 "@timestamp": "2023-08-14T11:50:51.994Z",  
 "dataAjuizamento": "2018-10-29T00:00:00.000Z",  
 "movimentos": [  
 {  
 "complementosTabelados": [  
 {  
 "codigo": 2,  
 "valor": 1,  
 "nome": "competência exclusiva",  
 "descricao": "tipo\_de\_distribuicao\_redistribuicao"  
 }  
 ],  
 "codigo": 26,  
 "nome": "Distribuição",  
 "dataHora": "2018-10-30T14:06:24.000Z"  
 },  
 ...  
 {  
 "codigo": 14732,  
 "nome": "Conversão de Autos Físicos em Eletrônicos",  
 "dataHora": "2020-08-05T01:15:18.000Z"  
 }  
 ],  
 "id": "TRF1\_436\_JE\_16403\_00008323520184013202",  
 "nivelSigilo": 0,  
 "orgaoJulgador": {  
 "codigoMunicipioIBGE": 5128,  
 "codigo": 16403,  
 "nome": "JEF Adj - Tefé"  
 },  
 "assuntos": [  
 {  
 "codigo": 6177,  
 "nome": "Concessão"  
 }  
 ]  
 }  
 }  
 ]  
 }  
}

**Ex. 3 - Exemplo 3: Pesquisa com paginação (search\_after):**

Por padrão, as pesquisas na API do Elasticsearch retornam até 10 registros por solicitação. No entanto, é possível aumentar o número de registros retornados utilizando o parâmetro "size" de paginação dos registros. Esse parâmetro permite especificar quantos resultados devem ser retornados por página, variando de 10 até 10.000 registros por página.

Quando se tem uma necessidade de percorrer uma maior quantidade de resultados, é possível fazer uso do recurso "search\_after". Esse recurso é prioritariamente recomendado para paginação de dados, pois permite que a API do Datajud continue a partir do ponto onde a última página parou, sem a necessidade de recarregar todos os resultados a cada nova página. O "search\_after" é um ponteiro que aponta para o último registro retornado na página anterior e pode ser informado como parâmetro para a próxima solicitação, permitindo que a API retorne os resultados seguintes.

É importante ressaltar que a utilização do "search\_after" não prejudica a performance da API na busca de grandes volumes de dados, pois permite que a API do Datajud execute consultas de forma mais eficiente, sem a necessidade de recarregar todos os resultados em cada página. Combinando o uso do parâmetro "size" com o "search\_after", é possível percorrer grandes volumes de dados de forma eficiente e com baixo impacto no desempenho da API.

Para paginar os resultados utilizando o search\_after, é necessário a utilização da ordenação (sort) dos dados utilizando o atributo **“@timestamp”** conforme exemplo abaixo:

Query DSL

{  
 "size": 100,  
 "query": {  
 "bool": {  
 "must": [  
 {"match": {"classe.codigo": 1116}},  
 {"match": {"orgaoJulgador.codigo": 13597}}  
 ]  
 }  
},  
"sort": [  
{  
 "@timestamp": {  
 "order": "asc"  
 }  
 }  
 ]  
}

Após a primeira consulta, a resposta da API incluirá um array chamado "sort" que contém os valores do campo de ordenação para cada documento retornado. Esse array pode ser utilizado como o valor do parâmetro "search\_after" na próxima consulta, juntamente com o parâmetro "size" que define a quantidade de documentos a serem retornados na próxima página

Query DSL

{  
 "\_index" : "api\_publica\_tjdft",  
 "\_type" : "\_doc",  
 "\_id" : "TJDFT\_1116\_G1\_13597\_00356079220168070018",  
 "\_score" : null,  
 "\_source" : {...},  
 "sort" : [  
 1681366085550  
 ]  
}

Para buscar os próximos 100 processos, basta adicionar o parâmetro "search\_after" na próxima consulta, utilizando o valor do campo “sort” do último documento retornado na página anterior conforme exemplo abaixo:

Query DSL

{  
 "size": 100,  
 "query": {  
 "bool": {  
 "must": [  
 {"match": {"classe.codigo": 1116}},  
 {"match": {"orgaoJulgador.codigo": 13597}}  
 ]  
}  
},  
 "sort": [  
 {  
 "@timestamp": {  
 "order": "asc"  
 }  
 }  
],  
 "search\_after": [ 1681366085550 ]  
}

Observe que o valor do campo "search\_after" é um array com os valores do campo de ordenação para o último documento retornado na página anterior. É importante lembrar que o "search\_after" deve ser utilizado em conjunto com o "sort" e o "size" para garantir uma paginação eficiente dos resultados.

**Ex. 2 - Pesquisar por Classe Processual e Órgão Julgador**

No exemplo abaixo é realizada a consulta de processos que possuam a Classe Processual **1116 – "Execução Fiscal"** do Órgão Julgador **13597 - VARA DE EXECUÇÃO FISCAL DO DF** no tribunal **TJDFT**.

**POST /api\_publica\_tribunal/\_search[​](https://datajud-wiki.cnj.jus.br/api-publica/exemplos/exemplo2" \l "post-api_publica_tribunal_search" \o "Link direto para POST /api_publica_tribunal/_search)**

* Postman
* Python
* R

1. Abra o **Postman** e clique em *"New Request"*.
2. Defina o método HTTP como POST.
3. Digite a URL: <https://api-publica.datajud.cnj.jus.br/api_publica_tjdft/_search>
4. Selecione a aba *"Headers"* e inclua a chave "Authorization" com o valor "APIKey [Chave Pública]";
5. O valor [Chave Pública] corresponde a chave pública disponível em [Chave Pública](https://datajud-wiki.cnj.jus.br/api-publica/exemplos/api-publica/acesso);
6. Ainda em *"Headers"* inclua a chave "Content-Type" com o valor "application/json";
7. Selecione a aba *"Body"* e escolha a opção *"raw"*. Insira o corpo da solicitação JSON conforme o exemplo abaixo:

Query DSL

{  
 "query": {  
 "bool": {  
 "must": [  
 {"match": {"classe.codigo": 1116}},  
 {"match": {"orgaoJulgador.codigo": 13597}}  
 ]  
 }  
 }  
}

1. Clique em *Send* para enviar e aguarde a resposta da Api.

**Resposta**[**​**](https://datajud-wiki.cnj.jus.br/api-publica/exemplos/exemplo2#resposta)

A resposta esperado é um JSON com os metadados de 1 ou mais processos conforme o critério da busca:

JSON com os metadados processuais

{  
 "took": 213,  
 "timed\_out": false,  
 "\_shards": {  
 "total": 3,  
 "successful": 3,  
 "skipped": 0,  
 "failed": 0  
 },  
 "hits": {  
 "total": {  
 "value": 10000,  
 "relation": "gte"  
 },  
 "max\_score": 2.0,  
 "hits": [  
 {  
 "\_index": "api\_publica\_tjdft",  
 "\_type": "\_doc",  
 "\_id": "TJDFT\_1116\_G1\_13597\_07223914020178070001",  
 "\_score": 2.0,  
 "\_source": {  
 "classe": {  
 "codigo": 1116,  
 "nome": "Execução Fiscal"  
 },  
 "numeroProcesso": "07223914020178070001",  
 "sistema": {  
 "codigo": 1,  
 "nome": "Pje"  
 },  
 "formato": {  
 "codigo": 1,  
 "nome": "Eletrônico"  
 },  
 "tribunal": "TJDFT",  
 "dataHoraUltimaAtualizacao": "2022-09-06T12:03:20.257Z",  
 "grau": "G1",  
 "@timestamp": "2023-04-13T17:59:46.214Z",  
 "dataAjuizamento": "2017-08-21T10:05:32.000Z",  
 "movimentos": [  
 {  
 "complementosTabelados": [  
 {  
 "codigo": 2,  
 "valor": 2,  
 "nome": "sorteio",  
 "descricao": "tipo\_de\_distribuicao\_redistribuicao"  
 }  
 ],  
 "codigo": 26,  
 "nome": "Distribuição",  
 "dataHora": "2017-08-21T10:05:32.000Z"  
 },  
 ...  
 {  
 "codigo": 11382,  
 "nome": "Bloqueio/penhora on line",  
 "dataHora": "2022-07-13T07:25:59.000Z"  
 },  
 {  
 "codigo": 132,  
 "nome": "Recebimento",  
 "dataHora": "2022-07-13T07:26:00.000Z"  
 }  
 ],  
 "id": "TJDFT\_1116\_G1\_13597\_07223914020178070001",  
 "nivelSigilo": 0,  
 "orgaoJulgador": {  
 "codigoMunicipioIBGE": 5300108,  
 "codigo": 13597,  
 "nome": "VARA DE EXECU??O FISCAL DO DF"  
 },  
 "assuntos": [  
 [  
 {  
 "codigo": 6017,  
 "nome": "Dívida Ativa (Execução Fiscal)"  
 }  
 ]  
 ]  
 }  
 },  
 {  
 "\_index": "api\_publica\_tjdft",  
 "\_type": "\_doc",  
 "\_id": "TJDFT\_1116\_G1\_13597\_00073039720138070015",  
 "\_score": 2.0,  
 "\_source": {  
 "classe": {  
 "codigo": 1116,  
 "nome": "Execução Fiscal"  
 },  
 "numeroProcesso": "00073039720138070015",  
 "sistema": {  
 "codigo": 1,  
 "nome": "Pje"  
 },  
 "formato": {  
 "codigo": 1,  
 "nome": "Eletrônico"  
 },  
 "tribunal": "TJDFT",  
 "dataHoraUltimaAtualizacao": "2022-09-06T17:26:23.938Z",  
 "grau": "G1",  
 "@timestamp": "2023-04-13T18:02:23.754Z",  
 "dataAjuizamento": "2019-05-30T03:17:56.000Z",  
 "movimentos": [  
 {  
 "complementosTabelados": [  
 {  
 "codigo": 2,  
 "valor": 1,  
 "nome": "competência exclusiva",  
 "descricao": "tipo\_de\_distribuicao\_redistribuicao"  
 }  
 ],  
 "codigo": 26,  
 "nome": "Distribuição",  
 "dataHora": "2013-02-18T13:17:23.000Z"  
 },  
 ...  
 {  
 "codigo": 245,  
 "nome": "Provisório",  
 "dataHora": "2019-05-30T11:10:02.000Z"  
 }  
 ],  
 "id": "TJDFT\_1116\_G1\_13597\_00073039720138070015",  
 "nivelSigilo": 0,  
 "orgaoJulgador": {  
 "codigoMunicipioIBGE": 5300108,  
 "codigo": 13597,  
 "nome": "VARA DE EXECU??O FISCAL DO DF"  
 },  
 "assuntos": [  
 [  
 {  
 "codigo": 6017,  
 "nome": "Dívida Ativa (Execução Fiscal)"  
 }  
 ],  
 [  
 {  
 "codigo": 10394,  
 "nome": "Dívida Ativa não-tributária"  
 }  
 ]  
 ]  
 }  
 }  
 ...  
 ]  
 }  
}

**Glossário de Dados**

O glossário de dados da API Pública do Datajud oferece de maneira detalhada os termos, conceitos e estruturas de dados específicos dessa API. A compreensão desse glossário é essencial para otimizar suas consultas, pois ele oferece uma referência sobre como os dados estão organizados e como podem ser acessados. Isso não só agiliza suas pesquisas, mas também garante a precisão e relevância dos resultados obtidos.

Portanto, recomendamos enfaticamente a exploração e compreensão do glossário como um passo essencial para aproveitar plenamente os recursos da API.

| **Atributos** | **Tipo** | **Descrição** |
| --- | --- | --- |
| **id** | text/keyword | Identificador da origem do processo no Datajud - Chave **Tribunal\_Classe\_Grau\_OrgaoJulgador\_NumeroProcesso** |
| **tribunal** | text/keyword | Identificação do Tribunal pela sigla |
| **numeroProcesso** | text/keyword | Numeração Única (CNJ) do processo sem formatação |
| **dataAjuizamento** | datetime | Data de ajuizamento da capa do processo |
| **grau** | text/keyword | Identificação do instância/grau (G1, G2, JE, etc...) |
| **nivelSigilo** | long | Nível de sigilo |
| **formato** | **object{}** | **Identificação de processo Físico ou Eletrônico** |
| **formato.codigo** | long | Código de identificação do formato do processo |
| **formato.nome** | text/keyword | Identificação se é Físico ou Eletrônico |
| **sistema** | **object{}** | **Sistema processual de origem do processo no Tribunal** |
| **sistema.codigo** | long | Código do sistema processual |
| **sistema.nome** | text/keyword | Descrição do sistema processual |
| **classe** | **object{}** | **Classe Processual conforme TPU** |
| **classe.codigo** | long | Código da classe processual |
| **classe.nome** | text/keyword | Descrição da classe processual |
| **assuntos** | **array []** | **Assuntos do Processo conforme TPU** |
| **assuntos.codigo** | **long** | Código do assunto |
| **assuntos.nome** | text/keyword | Descrição do assunto |
| **orgaoJulgador** | **object {}** | **Órgão Julgador** |
| **orgaoJulgador.codigo** | long | Código da serventia/vara atual do processo |
| **orgaoJulgador.nome** | text/keyword | Nome da serventia/vara |
| **orgaoJulgador.codigoMunicipioIBGE** | long | Identificação do município pelo código do IBGE |
| **movimentos** | **array []** | **Movimentos Processuais** |
| **movimentos.codigo** | long | Código da movimentação processual conforme TPU |
| **movimentos.nome** | text/keyword | Descrição da movimentação |
| **movimentos.dataHora** | datetime | Data e hora da ocorrência de movimentação |
| **movimentos.complementosTabelados** | **array []** | **Lista de complementos tabelados daquela movimentação** |
| **movimentos.complementosTabelados.codigo** | long | Código da variável de movimento tabelado |
| **movimentos.complementosTabelados.descricao** | text/keyword | Descrição da variável do movimento tabelado |
| **movimentos.complementosTabelados.valor** | long | Código do complemento tabelado |
| **movimentos.complementosTabelados.nome** | text/keyword | Descrição do complemento tabelado |
| **movimentos.orgaoJulgador** | object {}\*\* | **Órgão julgador do movimento** |
| **movimentos.orgaoJulgador.codigoOrgao** | long | Código da serventia/vara do movimento |
| **movimentos.orgaoJulgador.nomeOrgao** | text/keyword | Nome da serventia/vara |

Os atributos abaixo são utilizados para controle interno da aplicação:

| **Atributos** | **Tipo** | **Descrição** |
| --- | --- | --- |
| **dataHoraUltimaAtualizacao** | datetime | Milisegundos do atributo *millisInsercao* da origem do dado |
| **@timestamp** | datetime | *Timestamp* da atualização do documento no índice |

**Considerações Finais**

A API Pública do Datajud suporta parâmetros de buscas e agregações no padrão do Elasticsearch, tornando a busca de dados mais eficiente e precisa. Para mais detalhes sobre como utilizar esses recursos, é possível consultar o guia [Search your data | Elasticsearch Guide](https://www.elastic.co/guide/en/elasticsearch/reference/current/search-your-data.html).

A versão atual da API está em fase Beta de testes, e permite o acesso público aos metadados de processos judiciais em todo o Brasil. Os dados fornecidos pela API são provenientes da **Base Nacional de Dados do Poder Judiciário (Datajud)**, mas com a devida proteção aos processos sigilosos e dados de partes.

Desenvolvedores e pesquisadores podem utilizar a API para acessar informações de capas processuais e movimentações relacionadas a processos judiciais em todas as instâncias do Judiciário brasileiro. Os dados disponibilizados podem ser utilizados para diversas finalidades, tais como pesquisas acadêmicas, desenvolvimento de aplicativos que facilitem o acesso à informação jurídica, observando os critérios estabelecidos no [Termo de Uso](https://datajud-wiki.cnj.jus.br/api-publica/termo-uso).