



# Catálogo de API Reports

**Sumário**

|  |           |
|--|-----------|
| <b>1. Histórico de revisão .....</b>                                 | <b>3</b>  |
| <b>2. Introdução .....</b>   | <b>4</b>  |
| <b>3. Verbos http.....</b>   | <b>7</b>  |
| <b>4. Endpoint.....</b>  | <b>7</b>  |
| <b>5. Endereços API.....</b>   | <b>7</b>  |
| <b>6. Swagger.....</b>   | <b>8</b>  |
| <b>7. Métodos / Taxonomia.....</b>                                   | <b>8</b>  |
| 7.1 Reports - Consulta files disponíveis .....                       | 9         |
| 7.2 Reports - Consulta Conteúdo de arquivos disponíveis .....        | 12        |
| <b>8. Response Code.....</b>   | <b>14</b> |
| <b>9. Autenticação .....</b>   | <b>15</b> |
| 9.1 Pinagem de Certificado.....                                      | 15        |
| 9.2 Token JWS (JWT assinado digitalmente com algoritmo RSA 256)..... | 17        |

**1. HISTÓRICO DE REVISÃO**

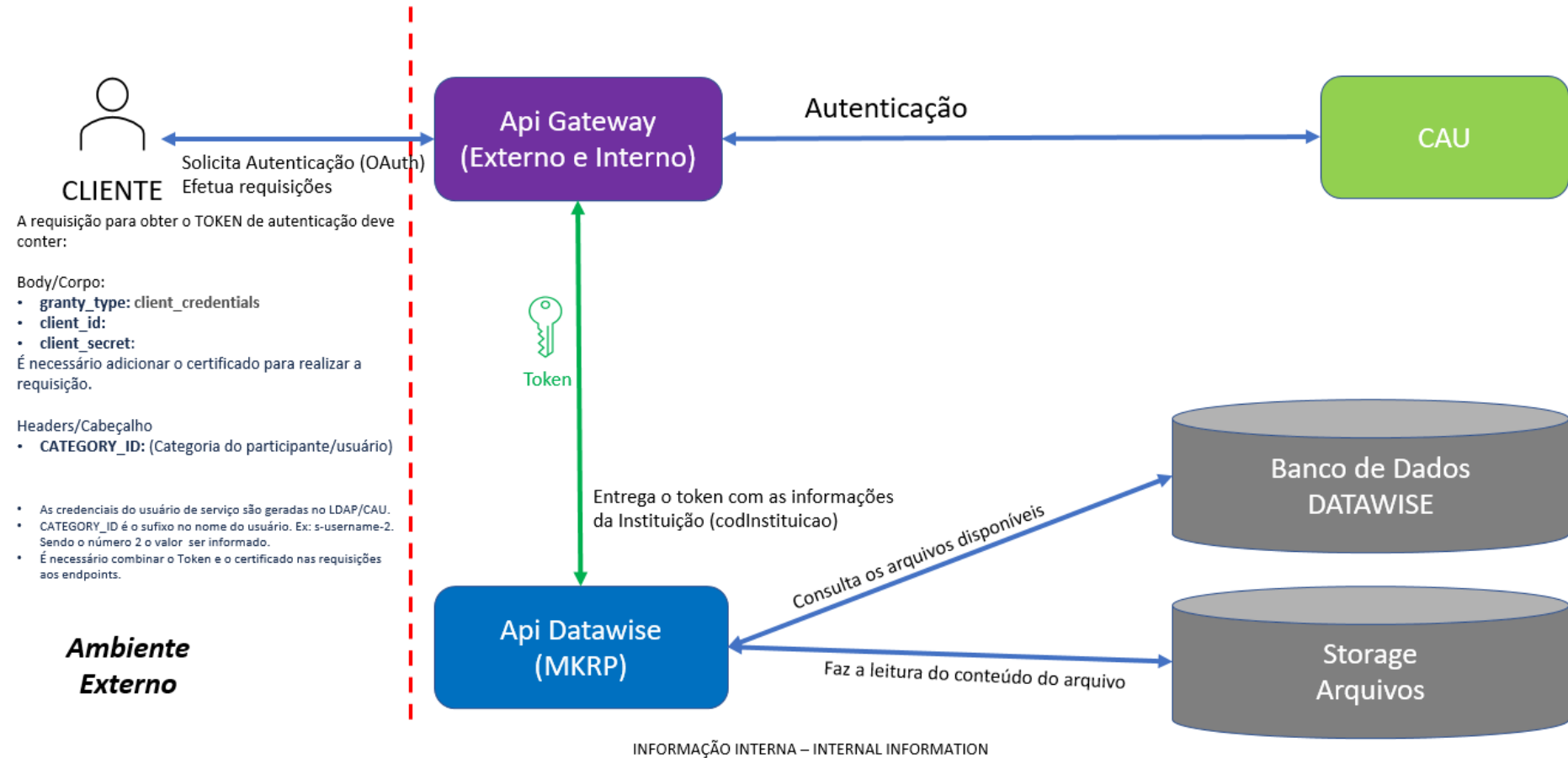
| Data       | Versão | Descrição         |
|------------|--------|-------------------|
| 05/08/2022 | 1.0    | 1. Criação da API |

## 2. INTRODUÇÃO

Este documento descreve o conjunto de definições da API desenvolvida pela B3 que serão utilizadas no processo de consulta de arquivos disponibilizados aos clientes contratantes dos serviços do Sistema DATAWISE Market Report.

A seguir é apresentado fluxograma do processo.

# Fluxo Básico da API



Esse documento trata especificamente da interface de comunicação API Web (REST) via internet.

## API – REPORTS

As APIs (Application Programming Interface) WEB possibilitam aos participantes do produto consultar a base da B3 para obter os dados dos arquivos contratados disponíveis.

Construídas no padrão REST (Representational State Transfer), consistem em um conjunto de URL's (Uniform Resource Locator) que viabilizam a realização de coleta e envio de mensagens executadas por sistema. As informações aqui presentes são específicas do processo de consulta dos arquivos contratados disponíveis e seu conteúdo.

Nas páginas a seguir será informada toda a documentação necessária para as APIs (Application Programming Interfaces), que possibilitam conexões diretas entre os sistemas dos participantes.

A API do Reports usa verbos HTTPS (Hyper Text Transfer Protocol Secure) em uma arquitetura REST (Representational State Transfer), o que facilita a solicitação e o recebimento dos mesmos. A autenticação básica é administrada por HTTPS e as respostas são entregues no formato JSON (Java Script Object Notation).

## API – REPORTS

### 3. VERBOS HTTP

Para a API Reports será utilizado somente o verbo GET. Os demais verbos HTTP não são suportados.

### 4. ENDPOINT

URI base (todas as URIs iniciam com): “[/api/reports/v1](#)”

Exemplo de uma URL completa, para o ambiente de produção, referente aos recursos “[/availables-files](#)” e “[/files-contents](#)”

<https://datawise.b3.com.br/api/reports/v1/availables-files>

<https://datawise.b3.com.br/api/reports/v1/files-contents>

### 5. ENDEREÇOS API

A B3 possui dois ambientes, um dedicado a certificação dos participantes e outro é o ambiente produtivo.

Seguem os endereços para conexão referente cada ambiente:

#### **Acesso por meio da Internet:**

- Certificação: <https://api-listados-cert.b3.com.br/api/reports/v1>
- Produção:

#### **Acesso por meio da Extranet (RCB):**

- Certificação: <https://api-listados.cert.netb3.com.br>
- Produção:

## API – REPORTS

**6. SWAGGER**

A documentação técnica detalhada da **API Reports** está disponível no Swagger em:

<https://bitbucket.intraservice.corp/projects/APIS/repos/apis/browse/Externa/Dados/Datawise/Reports/v1/Swagger>

**Documentos:** “Swagger da API Reports”

**7. MÉTODOS / TAXONOMIA**

Os itens abaixo têm como objetivo apresentar a taxonomia (estruturas, nomes, tipos, descrição, valores disponíveis etc.) dos dados que compõem os recursos da API Reports.

Essas informações poderão ser atualizadas conforme a necessidade de ajustes ou manutenções/melhorias que possam surgir e sempre que ocorrer versionamento ou inclusão de novas funcionalidades na API.

| Método | Endpoint                                      | Descrição  |
|--------|---|--|
| GET    | <code>/api/reports/v1/availables-files</code> | Consulta Arquivos disponíveis para download                              |
| GET    | <code>/api/reports/v1/files-contents</code>   | Consulta o conteúdo do arquivo solicitado, formatando e entregando JSON. |

## 7.1 Reports - Consulta de arquivos disponíveis

### GET - /api/reports/v1/availables-files

Esse método é responsável por realizar consulta de arquivos disponíveis.

Este recurso tem como característica os seguintes itens:

- Os campos de request são informados via query.
- Campos de paginação (page e page-size) são obrigatórios.
- Caso haja alguma falha no recebimento ou na conversão dos valores referente à page-size, o valor 20 será considerado pela API.
- Valor máximo para page-size é 50, caso seja atribuído um valor maior, o mesmo será convertido para 50.
- Com exceção dos campos de paginação, nenhum outro campo é obrigatório.
- Caso não haja nenhum campo (além dos de paginação) na requisição, a busca será efetuada com base na data atual, considerando somente os contratos vigentes.
- Quando o campo startDate estiver ausente, a data atual será considerada nas buscas e o campo endDate será ignorado.
- Para buscas entre períodos, ou seja, com os campos startDate e endDate presentes na requisição, o campo contractCode deve ser informado, caso contrário o campo endDate será ignorado.
- Os parâmetros de entrada desta api, para consulta de arquivos disponíveis, entram via query.

### Exemplo

```
curl --location --request GET 'https://<server>/api/reports/v1/availables-files?startDate=2022-01-01&endDate=2022-07-27&contractCode=224&page=1&page-size=10' \  
--header 'Authorization: Bearer conteudo_do_token' \  

```

## Json de saída

```
{
  "data": {
    "availablefiles": [
      {
        "fileName": null,
        "fileDate": null,
        "fileCode": null,
      }
    ]
  },
  "links": {
    "self": null,
    "first": null,
    "prev": null,
    "next": null,
    "last": null
  }
}
```

## Parâmetros de Entrada

| Índice | Campo     | Card. | Tipo de Dado | Detalhe do Tipo de Dado | Descrição    |
|--------|-----------|-------|--------------|-------------------------|--------------|
| 1.0    | startDate | 0..1  | date         | String, date format     | Start Date.  |
| 2.0    | endDate   | 0..1  | date         | String, date format     | End of date. |

## API – REPORTS

|     |              |      |        |                         |   |
|-----|--------------|------|--------|-------------------------|---|
| 3.0 | fileDate     | 0..1 | date   | String, date format     | Specifies the Date the file is available<br>Specifies the Date the file is available. |
| 4.0 | contractCode | 0..1 | int    | Integer number positive | Contract Code.  |
| 5.0 | Page         | 1..1 | int    | Integer number positive | Number of the page being requested (the value of the first page is 1).                |
| 6.0 | page-size    | 1..1 | int    | Integer number positive | Total number of records per page.   |
| 7.0 | Errors       | 0..* |        |                         |   |
| 7.1 | Code         | 1..1 | string | MaxLength=10            | Código do erro.   |
| 7.2 | Title        | 1..1 | string | MaxLength=100           | Mensagem do erro.   |
| 7.3 | Detail       | 1..1 | string | MaxLength=2000          | Informações complementares sobre o erro.  |

## Parâmetros de Saída

| Índice | Campo          | Card. | Tipo de Dado | Detalhe do Tipo de Dado | Descrição   |
|--------|----------------|-------|--------------|-------------------------|---|
| 1.0    | x-v            | 1..1  | string       |                         | Versão da api. Ex: 1.0.0  |
| 2.0    | data           | 1..1  |              |                         |   |
| 2.1.0  | Availablefiles | 1..1  |              |                         |   |
| 2.1.1  | fileName       | 1..1  | string       | MaxLength=50            |   |
| 2.1.2  | fileDate       | 1..1  | date         |                         | Specifies the Date the file is available<br>Specifies the Date the file is available. |
| 2.1.3  | fileCode       | 1..1  | integer      |                         |   |
| 3.0    | errors         | 0..*  |              |                         |   |
| 3.1    | code           | 1..1  | string       | MaxLength=10            | Código do erro.   |
| 3.2    | title          | 1..1  | string       | MaxLength=100           | Mensagem do erro.   |
| 3.3    | detail         | 1..1  | string       | MaxLength=2000          | Informações complementares sobre o erro.  |
| 4.0    | links          | 1..1  |              |                         |   |
| 4.1    | self           | 1..1  | string       | MaxLength=2000          | Current page link   |

## API – REPORTS

|     |       |      |        |                |                 |
|-----|-------|------|--------|----------------|-----------------|
| 4.2 | first | 0..1 | string | MaxLength=2000 | First page link |
| 4.3 | prev  | 0..1 | string | MaxLength=2000 | Next page link  |
| 4.4 | next  | 0..1 | string | MaxLength=2000 | Next page link  |
| 4.5 | last  | 0..1 | string | MaxLength=2000 | Last page link  |

## 7.2 Reports - Consulta Conteúdo de arquivos disponíveis

### GET - /api/reports/v1/files-contents

Esse método é responsável por realizar consulta de conteúdo dos arquivos disponíveis.

Este recurso tem como característica os seguintes itens:

- Este recurso não possui campos de paginação.
- O campo **fileDate** é obrigatório, devendo ser informado em todas as requisições.
- Os campos **fileName** e **fileCode** não são obrigatórios, porém é necessário que ao menos um deles esteja presente na requisição.
- Caso os campos **fileName** e **fileCode** estejam presentes em uma mesma requisição, o campo **fileName** será ignorado.
- Os parâmetros de entrada desta api, para consulta dos conteúdos de arquivos disponíveis, são informados via query.

#### Exemplo

```
curl --location --request GET 'https://<server>/api/reports/v1/available-files?startDate=2022-01-01&endDate=2022-07-27&contractCode=224&page=1&page-size=10' \
--header 'Authorization: Bearer conteudo_do_token' \
```

#### Json de saída

```
{
  "data": {
    "filesContents": [
```

## API – REPORTS

```

    {
      "fileName": null,
      "fileDate": null,
      "fileCode": null,
      "listHeader": null,
      "listContent": null
    }
  ],
  "links": {
    "self": null,
    "first": null,
    "prev": null,
    "next": null,
    "last": null
  }
}

```

## Parâmetros de Entrada

| Índice | Campo    | Card. | Tipo de Dado | Detalhe do Tipo de Dado | Descrição   |
|--------|----------|-------|--------------|-------------------------|---|
| 1.0    | fileDate | 1..1  | Date         |                         | Specifies the Date the file is available<br>Specifies the Date the file is available. |
| 2.0    | fileName | 1..1  | String       | MaxLength=50            | Nome do arquivo.  |
| 3.0    | fileCode | 1..1  | Integer      |                         | Código do arquivo   |
| 4.0    | Errors   | 0..*  |              |                         |   |
| 4.1    | Code     | 1..1  | String       | MaxLength=10            | Código do erro.   |
| 4.2    | Title    | 1..1  | String       | MaxLength=100           | Mensagem do erro.   |
| 4.3    | Detail   | 1..1  | String       | MaxLength=2000          | Informações complementares sobre o erro.  |

## Parâmetros de Saída

## API – REPORTS

| Índice | Campo         | Card. | Tipo de Dado | Detalhe do Tipo de Dado | Descrição  |
|--------|---------------|-------|--------------|-------------------------|--|
| 1.0    | x-v           | 1..1  | string       |                         | Versão da api. Ex: 1.0.0   |
| 2.0    | data          | 1..1  |              |                         |  |
| 2.1.0  | filesContents | 1..1  |              |                         |  |
| 2.1.1  | fileName      | 1..1  | string       | MaxLength=50            |  |
| 2.1.2  | fileDate      | 1..1  | date         |                         | Specifies the Date the file is available                                 |
| 2.1.3  | fileCode      | 1..1  | Integer      |                         |  |
| 2.1.3  | listHeader    | 1..1  | string       |                         | List with headers, where each column/item is equivalent to an attribute. |
| 2.1.4  | listContent   | 1..1  | string       |                         | List with content, where each line is equivalent to a record.            |
| 3.0    | errors        | 0..*  |              |                         |  |
| 3.1    | code          | 1..1  | string       | MaxLength=10            | Código do erro.  |
| 3.2    | title         | 1..1  | string       | MaxLength=100           | Mensagem do erro.  |
| 3.3    | detail        | 1..1  | string       | MaxLength=2000          | Informações complementares sobre o erro.                                 |

## 8. RESPONSE CODE

A B3 usa códigos de resposta HTTP normais para indicar o sucesso ou falha de uma solicitação da API. Um código de resposta de 200, por exemplo, significa sucesso, enquanto os códigos no intervalo 4xx indicam um erro em relação às informações fornecidas. Já os códigos no intervalo 5xx apontam um erro de comunicação com os nossos servidores. Confira quais são os códigos de erro utilizados pelos recursos da API Data-CorporateActions.

### Http codes retornados

| Código | Código Retorno | Mensagem     | Descrição  |
|--------|----------------|--------------|--|
| 200    | 200            | Ok           | Indica que a requisição foi recebida e executada com sucesso.  |
| 201    | 201            | Created.     | Indica que a requisição foi bem sucedida e que um novo recurso foi criado.   |
| 204    | 204            | No Content.  | Operação de exclusão ou alteração concluída com sucesso.   |
| 400    | 400            | Bad Request  | Indica que a requisição não foi executada porque está mal formada.   |
| 401    | 401            | Unauthorized | O servidor não pôde verificar se você está autorizado a acessar o documento solicitado. As credenciais fornecidas estão erradas (por exemplo, senha incorreta) ou seu navegador não sabe como fornecer as credenciais necessárias. |

## API – REPORTS

|            |     |                        |   |
|------------|-----|------------------------|---|
|            |     |                        |   |
| <b>403</b> | 403 | Forbidden              | Indica que o servidor entendeu o pedido, mas se recusa a autorizá-lo. Esse status é semelhante ao 401, mas neste caso a re-autenticação não fará diferença. O acesso é permanentemente proibido e vinculado à lógica da aplicação (como uma senha incorreta). |
| <b>404</b> | 404 | Not Found              | O erro 404 é um código de resposta HTTP que indica que o cliente não pôde comunicar com o servidor, ou o servidor não pôde encontrar o que foi pedido, ou foi configurado para não cumprir o pedido e não revelar a razão ou a página não existe mais.        |
| <b>422</b> | 422 | Unprocessable Entity.  | Indica que solicitação foi bem formada, mas não pôde ser processada devido à lógica de negócios específica da solicitação. Podem retornar objeto errors ou erros.   |
| <b>500</b> | 500 | Internal Server Error. | Indica que a operação falhou. Ocorreu um erro no gateway da API ou no microsserviço.  |

## 9. AUTENTICAÇÃO

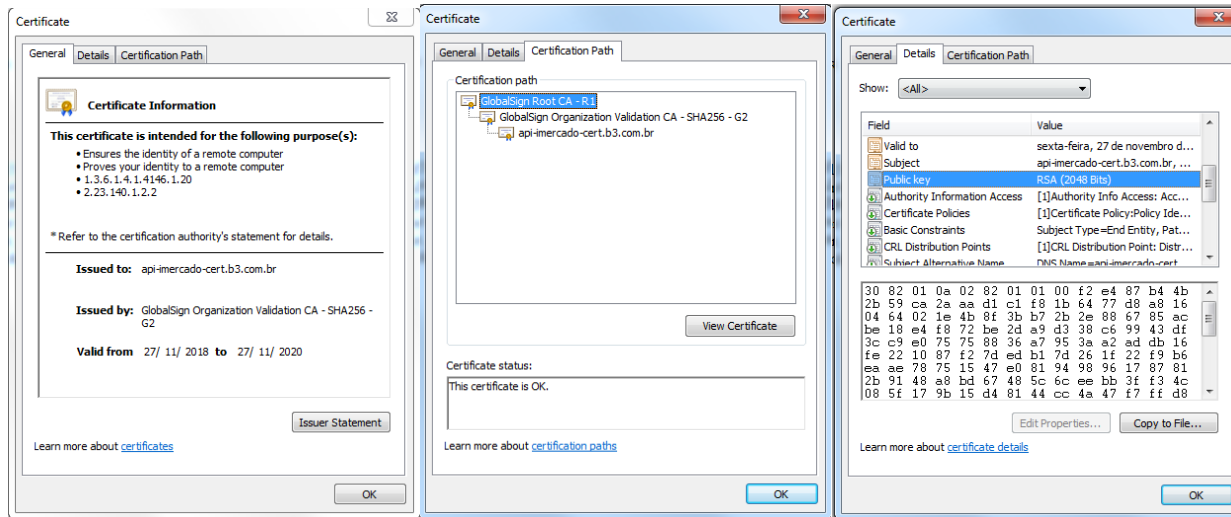
Neste capítulo temos as informações relacionadas à Segurança da Informação.

A B3 definiu como modelo de segurança para as APIs expostas o uso de pinagem de certificado e tokens JWS (JWT assinado digitalmente com algoritmo RSA 256) para as requisições enviadas.

### 9.1 Pinagem de Certificado

Os clientes que irão consumir as APIs deverão implementar a pinagem do certificado raiz do endereço de conexão. As imagens abaixo ilustram como obter as informações para implementação da pinagem de certificado.

# API – REPORTS



## 9.2 Token JWS (JWT assinado digitalmente com algoritmo RSA 256)

Para consumir a API, o usuário deverá realizar a requisição de um token JWS utilizando o grant ROPC – Resource Owner Password Credentials, que garante autorização para seguir. Ele precisará informar os dados abaixo na requisição do token JWT:

| Método | Endpoint         | Descrição  |
|--------|------------------|--|
| POST   | /api/oauth/token | Lista com informações dos contratos da Instituição |

### CABEÇALHOS/HEADERS:

**Content-Type:** application/x-www-form-urlencoded

**CATEGORY\_ID:** <código/categoria fantasia sincad do participante> (Sufixo do nome do usuário. Exemplo: “s-999-1”, onde CATEGORY\_ID é 1).

### CORPO/BODY:

**grant\_type** = client\_credentials (Este valor é fixo, ou seja, é igual para todas as requisições.)

**client\_id** = <cliente id informado pela b3> (Trata-se do Client ID que o usuário recebeu no pacote de acesso.)

**client\_secret** = <cliente secret informado pela b3> (Refere-se ao Client Secret que o usuário recebeu no pacote de acesso.)

### Exemplo de requisição

```
curl --location --request POST 'https://<api-server>/api/oauth/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'CATEGORY_ID: 4' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=XXXXXXXXXX-XXXXXXXXXX' \
--data-urlencode 'client_secret=XXXXXXXXXX-XXXXXXXXXX'
```

### A resposta da requisição será semelhante a abaixo

```
{
  "access_token": "eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiI5OTkiLCAiZmxhZ3oiOiMTU2MDAwNzgyOSIsICJpc3MiOiJDTj1JbnRlcm1lZCBDZXJ0aWZpY2F0ZSA0IEpXVCxPVT1HQ01DLE89QjMgU0EsTD1TYW8gUGF1bG8sUz1TUCxDPUJSIn0.JSWxLOVX6xWmeLAhwdTP2xKt2eK3JAJ2oB61Yn06PROQdpaCk8E_CaS-xGc2xz9iBEnLTxZTrfdhyYstkBv90fLXVQnhEVFhfXLq2Ov-xWAO_DFPeGLXzy5_7WOpbZ3oKbjJ1XVxbCZnoDIt3VRZGNHAYis8dZJzxV0n9D8qa_HhtZhOJNbH0ynhPyoE8qMULvgJQ5DzjXIvk2mP-",
  "token_type": "Bearer",
  "expires_in": "7200",
  "scope": "resource.WRITE resource.READ"
}
```

### CONTEÚDO DA RESPOSTA:

- **access\_token**: Conteúdo do token JWS. Ele deverá ser informado no cabeçalho das requisições às APIs.
- **expires\_in**: A validade do token é de 7200 segundos (2 horas). Após esse período, o token JWS não será mais válido, e o usuário deverá requisitar um novo token para consumir as APIs.
- **token\_type**: Especifica o authentication schema utilizado pela API. Nesse caso é do tipo Bearer.

**REFERÊNCIAS:**

OAuth 2.0: <https://tools.ietf.org/html/rfc6749>

JWT: <https://tools.ietf.org/html/rfc7519>