



Catálogo de API

Contracts

Sumário

1. Histórico de revisão	3
2. Introdução	4
3. Verbos http.....	7
4. Endpoint	7
5. Endereços API	7
6. Swagger.....	7
7. Métodos / Taxonomia.....	8
7.1 Reports - Consulta files disponíveis	8
8. Response Code.....	12
9. Autenticação	13
9.1 Pinagem de Certificado.....	13
9.2 Token JWS (JWT assinado digitalmente com algoritmo RSA 256).....	14

1. HISTÓRICO DE REVISÃO

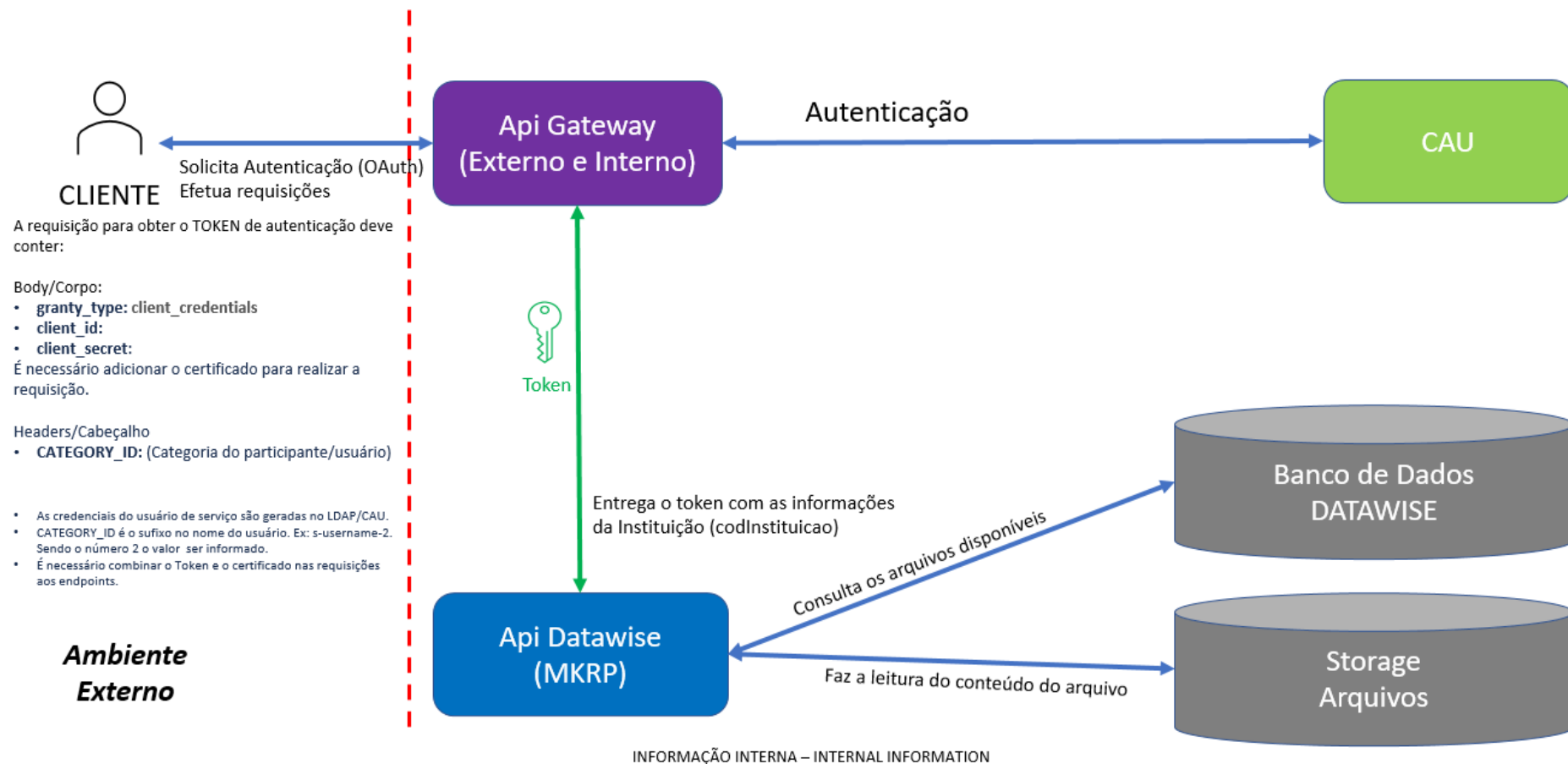
Data	Versão	Descrição
05/08/2022	1.0	1. Criação da API

2. INTRODUÇÃO

Este documento descreve o conjunto de definições da API desenvolvida pela B3 que serão utilizadas no processo de consulta das informações dos contratos da instituição junto ao sistema DATAWISE MarketReport.

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 – CONTRACTS

As APIs (Application Programming Interface) WEB possibilitam aos participantes do produto consultar a base da B3 para obter as informações dos contratos efetivados de sua instituição junto à base de dados do Sistema Datawise Market Report.

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 às informações dos contratos da instituição.

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 – CONTRACTS

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/contracts/v1”](#)

Exemplo de uma URL completa, para o ambiente de produção, referente ao recurso “/lists”

<https://datawise.b3.com.br/api/contracts/v1/lists>

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/contracts/v1>
- Produção:

Acesso por meio da Extranet (RCB):

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

6. SWAGGER

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

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

Documentos: “Swagger da API Contracts”

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 Contracts.

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	/api/contracts/v1/lists	Lista com informações dos contratos da Instituição

7.1 Contracts – Lists

GET - /api/contracts/v1/lists

Esse método é responsável por realizar consulta de informações dos contratos da Instituição.

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.
- O campo typeDate deve ser combinado com o campo startDate, caso esse esteja ausente, o campo typeDate será ignorado pela API.
- O campo typeDate reconhece os valores: CONTRACT (Busca por data de contrato), START_OF_AVAILABILITY (Busca por início de disponibilização), END_OF_AVAILABILITY (Busca por data final de disponibilização), quando estiver ausente será considerado o valor START_OF_AVAILABILITY.
- Caso o campo typeDate receba um valor diferente dos aceitáveis, será apresentado uma mensagem de erro com o código 422.
- Os parâmetros de entrada desta api, para consulta de arquivos disponíveis, entram via query.

Exemplo

```
curl --location --request GET 'https://<server>/api/reports/v1/files-contents?fileDate=2022-01-24&fileCode=184' \  
--header 'Authorization: Bearer conteudo_do_token' \  

```

Json de saída

```
{  
  "data": {  
    "contracts": [  
      {  
        "contractCode": null,  
        "productCode": null,  
        "productName": null,  
        "frequencyTypeName": null,  
        "availableDay": null,  
        "availableHour": null,  
        "contractDay": null,  
        "availabilityStartDate": null,  
        "availabilityEndDate": 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		Start Date.
2.0	endDate	0..1	Date		End of date.
3.0	typeDate	0..1	String		Specific which type of date should be considered in the search (contract date, availability start date and availability end date).
4.0	Page	1..1	Int		Number of the page being requested (the value of the first page is 1).
5.0	page-size	1..1	Int		Total number of records per page.
6.0	productCode	0..1	Int		Number of code of product.
7.0	contractCode	0..1	Int		Number od code of contract.

API – CONTRACTS

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	contracts	1..1			
2.1.1	contractCode	1..1	string		Código do Contrato.
2.1.2	productCode	1..1	int		Código do Produto.
2.1.3	productName	1..1	string	MaxLength=175	Guaranty product name
2.1.4	frequencyTypeName	1..1	string	MaxLength=50	Frequency at which the product will be made available.
2.1.5	availableDay	1..1	string	MaxLength=50	Product availability day.
2.1.6	availableHour	1..1	string	MaxLength=50	Product availability time.
2.1.8	contractDay	0..1	date		Date of signing contract
2.1.9	availabilityStartDate	0..1	date		Availability start date.
2.1.10	availabilityEndDate	0..1	date		Availability end date.
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
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

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.
403	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).
404	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.
422	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.
500	500	Internal Server Error.	Indica que a operação falhou. Ocorreu um erro no gateway da API ou no microserviço.

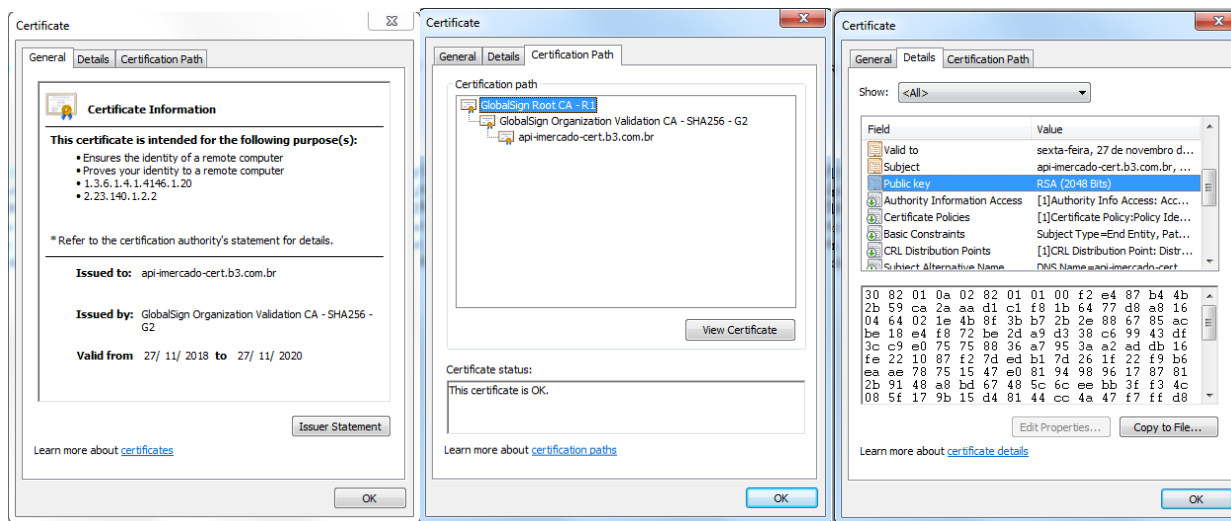
9. AUTENTICAÇÃO

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 – CONTRACTS

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.eyJzdWIiOiI5OTkiLCAiZmxhZ3oiOiJMTU2MDAwNzgyOSIsICJpc3MiOiJDTj1JbnRlcm1lZCBDZXJ0aWZpY2F0ZSA0IEpXVCxPVT1HQ01DLE89QjMgU0EsTD1TYW8gUGF1bG8sUz1TUCxDPUJSIn0.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>