

Integração e Funcionalidade WebFeeder

15 de outubro

Responsável: Rafael Gomes Vieira

Índice

Conteúdo

Índice	2
Integração WebFeeder	3
Introdução	3
Integração WebFeeder	4
I- Considerações Iniciais	4
II- Objetivo	4
III- Modelo Operacional.....	4
Arquitetura de Comunicação do Consumidor	4
I- Modelo Conceitual	4
II- Padrões Técnicos	5
III- Autenticação.....	6
IV- Consumo.....	8
V- Erros.....	8
Funcionalidade WebFeeder.....	8
Módulos	9
I- Módulo Quotes.....	9
II- Módulo News	28
II- Módulo Negotiation	42

Integração WebFeeder

Preparado pela equipe: WEB

Introdução

Este manual descreve os passos necessários para efetuar a integração do WebFeeder com a aplicação cliente que irá consumir os dados oriundos do mesmo.

Integração WebFeeder

I-Considerações Iniciais

O WebFeeder Cedro foi desenvolvido usando a arquitetura REST (Representational State Transfer), uma técnica de engenharia de software para sistemas hipermídia distribuídos como a World Wide Web.

Está arquitetura é usada para descrever qualquer interface web simples que utiliza XML e HTTP (ou mesmo YAML, JSON e texto puro), sem as abstrações adicionais dos protocolos baseados em padrões de trocas de mensagem como o protocolo de serviços web SOAP.

O REST se utiliza do protocolo HTTP diferente de SOAP que cria um novo protocolo sobre o HTTP e trabalha com classes de retorno que são 5, o número inicial do código de retorno indica a classe de retorno, por exemplo o retorno 200 é da classe 2 que é sucesso.

II- Objetivo

O WebFeeder tem como objetivo a disponibilidade de informações dinâmicas, onde o mesmo provê informações de acordo com a requisição do cliente, sendo ele o ator principal na utilização do serviço fornecido.

III- Modelo Operacional

O WebFeeder exige que o consumo de dados seja efetuado após autenticação do consumidor destas informações, onde o retorno destas, podem ser no formato JSON e XML, dependendo da forma que a requisição do consumidor for executada.

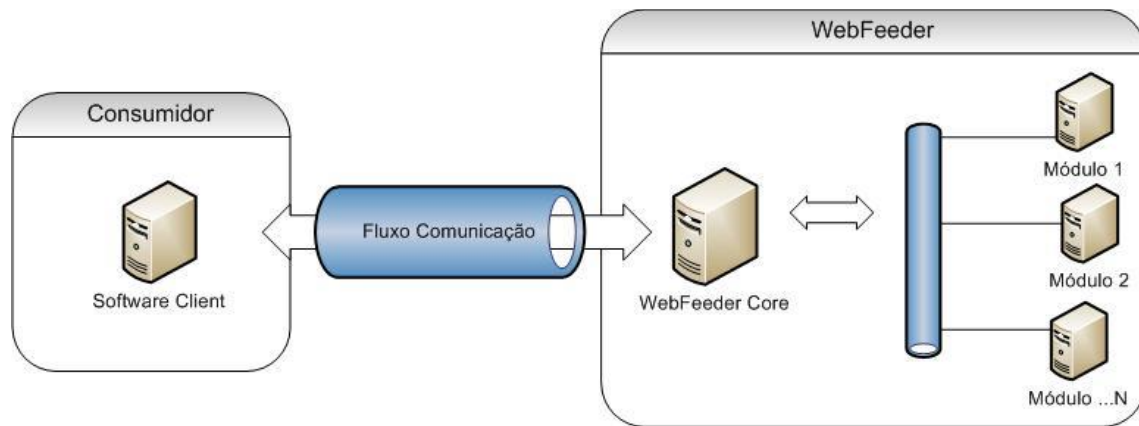
Arquitetura de Comunicação do Consumidor

I- Modelo Conceitual

O WebFeeder fornece diversas informações, onde o fluxo da informação é iniciado através da autenticação do consumidor, o mesmo efetua a autenticação e através do "HTTPSession" gerado pode-se iniciar o consumo, a cada requisição o tempo de expiração deste "HTTPSession" é reciclado, sendo de responsabilidade do consumidor efetuar nova autenticação quando o mesmo identificar que a sessão não é mais válida.

A arquitetura de comunicação do WebFeeder permite que o cliente faça uma requisição, e o mesmo executa a busca desta solicitação de acordo com o requisitado, processa esta requisição e a devolve no formato desejado.

Arquitetura de Comunicação - Visual



II- Padrões Técnicos

O serviço pode ser consumido e ter duas formas de retorno JSON e XML, sendo eles:

1. JSON (JavaScript Object Notation) - é uma das principais notações para intercâmbio de dados entre aplicativos, que é possível utilizar com mais de 20 linguagens de programação diferentes. Para maiores informações esta disponível a documentação completa e exemplos de estruturas no site oficial do Json: <http://www.json.org/>, segue abaixo um exemplo do retorno JSON:

```
{
  "Aluno": [
    {
      "nome": "João",
      "notas": [8, 9, 7]
    },
    {
      "nome": "Maria",
      "notas": [8, 10, 7]
    },
    {
      "nome": "Pedro",
      "notas": [10, 10, 9]
    }
  ]
}
```

2. XML (Extensible Markup Language) - é uma recomendação W3C para gerar linguagens de marcação para uma necessidade especial, seu propósito principal é a facilidade de compartilhamento de informações na internet, e a principal característica é de criar uma infraestrutura única par diversas linguagens. Para maiores informações esta disponível a documentação completa e exemplos de estruturas no site oficial do XML: <http://www.w3.org/XML/>, segue abaixo um exemplo de retorno XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<receita nome="pão" tempo_de_preparo="5 minutos" tempo_de_cozimento="1 hora">
  <titulo>Pão simples</titulo>
  <ingredientes>
    <ingrediente quantidade="3" unidade="xícaras">Farinha</ingrediente>
    <ingrediente quantidade="7" unidade="gramas">Fermento</ingrediente>
  </ingredientes>
</receita>
```

```
<ingredientequantidade="1.5"unidade="xícaras"estado="morna">Água</ingrediente>
<ingredientequantidade="1"unidade="colheres de chá">Sal</ingrediente>
</ingredientes>
<instrucoes>
<passo>Misture todos os ingredientes, e dissolva bem.</passo>
<passo>Cubra com um pano e deixe por uma hora em um local morno.</passo>
<passo>Misture novamente, coloque numa bandeja e asse num forno.</passo>
</instrucoes>
</receita>
```

III- Autenticação

O WebFeeder exige que o consumidor de dados possua um sessão válida para efetuar o consumo de dados do mesmo, segue abaixo um exemplo na linguagem JAVA de um evento de *login* no WebFeeder:

Imports

```
import java.io.IOException;
import java.io.InputStream;
import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.HttpException;
import org.apache.commons.httpclient.HttpMethodBase;
import org.apache.commons.httpclient.methods.PostMethod;
import org.springframework.http.HttpStatus;
import br.com.cedro.cm.logs.WSLog;
import br.com.cedro.cm.utils.JsonUtil;
```

Autenticação

```
private static HttpClient httpClient = new HttpClient();

public String loginWS() {
    String result = null;
    PostMethod method = new PostMethod("http://webfeeder.cedrofinances.com.br/SignIn");
    method.addParameter("login", "teste");
    method.addParameter("password", "123456");
    method.setDoAuthentication(true);

    try {
        httpClient.getHttpClientConnectionManager().getParams().setSoTimeout(7000);
        httpClient.executeMethod(method);
        result = (method.getStatusCode() == HttpStatus.OK.value()) ? getResonseBody(method) : null;
    } catch (HttpException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        method.releaseConnection();
    }
    return result;
}

private static String getResonseBody(HttpMethodBase method) throws IOException {
    StringBuilder output = new StringBuilder();
    int r;
    InputStream reader = method.getResponseBodyAsStream();

    while ((r = reader.read()) != -1) {
        output.append((char) r);
    }

    return output.toString();
}
```

Podemos notar que o "httpClient" que iremos utilizar é uma única instância para a aplicação, pois a sessão que iremos utilizar para as requisições de consumo será sempre a mesma, quando esta sessão não conseguir se autenticar, deve executar o mesmo método de *login* para renovar a sessão.

A requisição do evento *login* tem um retorno de uma String contendo "true" ou "false".

URL de Autenticação: <http://webfeeder.cedrofinances.com.br/SignIn>

O token é utilizado como um dispositivo de segurança adicional.

URL de Autenticação:

http://webfeeder.cedrofinances.com.br/token/authentication?login=<login>&password=<password>

IV- Consumo

Para efetuar o consumo de um serviço disponibilizado pelo WebFeeder é necessário enviar uma requisição utilizando o mesmo "httpClient" que recebeu o *session* do evento *login*, segue abaixo um exemplo de consumo de dados:

```
public void getQuote(){
    GetMethod getMethod = new GetMethod("http://webfeeder.cedrofinances.com.br/services/quotes/quote/petr4");
    String result = HttpUtils.get(getMethod);

    System.out.println(result); //Resposta Json do Serviço solicitado
}
```

V- Erros

Possíveis erros que a aplicação pode retornar:

ERRO	DESCRIÇÃO
401	UNAUTHORIZED
404	NOT_FOUND
405	METHOD_NOT_ALLOWED
408	REQUEST_TIMEOUT

Funcionalidade WebFeeder

Preparado pela equipe: WEB

Módulos

Para que a requisição venha no formato JSON, não é necessário adicionar nenhum parâmetro, porém para que a requisição tenha o retorno XML é necessário ao fim da URL adicionar `"/xml"`.

Exemplo Requisição XML:

`http://webfeeder.cedrofinances.com.br/services/quotes/quote/petr4/xml`

I- Módulo Quotes

1. Métodos disponíveis:

- **quote**
- **highList**
- **fallList**
- **fallVolumeList**
- **highVolumeList**
- **quotesIndex**
- **listMarket**
- **listTypeQuotes**
- **listQuote**
- **listQuoteMarket**
- **listQuoteMarketType**
- **listCompanyMarket**
- **candleLast**
- **candleDate**
- **crossRanking**
- **playerRanking**
- **optionsQuote**
- **companyQuotes**
- **auctionList**
- **quotesQuery**
- **proventos**
- **marketSectors**
- **marketSubSectorQuotes**
- **listContractBMF**
- **quoteInformation**
- **quoteTimesTrade**
- **highListOptions**
- **fallListOptions**
- **highVolumeListOptions**
- **fallVolumeListOptions**

- highListNegotiation
- highListFuturo
- fallListFuturo
- highVolumeListFuturo
- fallVolumeListFuturo
- playerNames
- quoteTrade
- indexList
- marketSectorQuotes
- quoteTimesTradeDate
- selicRate
- events
- eventsCedro
- eventsDatagro
- datesContractExpiration
- holidays
- volumeAtPrice
- quotesQueryMarket
- permissionsMarketUser

2. Requisição e Resposta:

quote: o serviço retorna os dados referentes ao ativo ou ativos solicitados, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quote/<ativo>`

`http://webfeeder.cedrofinances.com.br/services/quotes/quote/<ativo>/<ativo>/<ativo>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quote/petr4`

`http://webfeeder.cedrofinances.com.br/services/quotes/quote/petr4/vale5/itub4`

Resposta:

[Objeto Quote]

- symbol
- timeUpdate
- dateTrade
- lastTrade

- previous
- change
- changeMonth
- bid
- ask
- timeLastTrade
- quantity
- quantityLast
- quantityTrades
- volumeAmount
- volumeFinancier
- high
- low
- open
- timeBid
- timeAsk
- volumeBid
- volumeAsk
- volumeBetterBid
- volumeBetterAsk
- lastTradeLastWeek
- lastTradeLastMonth
- lastTradeLastYear
- playerBid
- interest
- situation

highList: o serviço retorna os dez ativos com maiores altas de um determinado índice, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/highList/<índice>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/highList/ibov`

Resposta:

[Objeto Resume]

- quote
- price
- value

- date
- volfin
- change
- qtdeNeg

fallList: o serviço retorna os dez ativos com maiores baixas de um determinado índice, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/fallList/<índice>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/fallList/ibov>

Resposta:

[Objeto Resume]

- quote
- price
- value
- date
- volfin
- change
- qtdeNeg

fallVolumeList: o serviço retorna os dez ativos com volumes mais baixos de um determinado índice, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/fallVolumeList/<índice>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/fallVolumeList/ibov>

Resposta:

[Objeto Resume]

- quote
- price
- value
- date

- volfin
- change
- qtdeNeg

highVolumeList: o serviço retorna os dez ativos com volumes mais altos de um determinado índice, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/highVolumeList/<índice>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/highVolumeList/ibov`

Resposta:

[Objeto Resume]

- quote
- price
- value
- date
- volfin
- change
- qtdeNeg

quotesIndex: o serviço retorna todos os valores dos ativos de um determinado índice de um determinado mercado, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quotesIndex/<mercado>/<índice>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quotesIndex/bovespa/ibov`

Resposta:

[Objeto Quote]

- symbol
- timeUpdate
- dateTrade
- lastTrade

- previous
- change
- changeMonth
- bid
- ask
- timeLastTrade
- quantity
- quantityLast
- quantityTrades
- volumeAmount
- volumeFinancier
- high
- low
- open
- timeBid
- timeAsk
- volumeBid
- volumeAsk
- volumeBetterBid
- volumeBetterAsk
- lastTradeLastWeek
- lastTradeLastMonth
- lastTradeLastYear
- playerBid
- interest
- situation

listMarket: o serviço retorna uma lista de todos os mercados, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/listMarket>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/listMarket>

Resposta:

[Objeto Market]

- code
- name

listTypeQuotes: o serviço retorna uma lista de todos tipos de Quotes possíveis, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/listTypeQuotes`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/listTypeQuotes`

Resposta:

[Objeto TypeQuotes]

- code
- description

listQuote: o serviço retorna uma lista de Quotes de acordo com os filtros passados, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/listQuote/<código do mercado>/<código do tipo de quote>/<speak>/<descrição>/<expired>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/listQuote/1/2/petrobras/pet/0`

Resposta:

[Objeto Quote]

- description
- speak
- expirationDate
- marketName
- typeQuotesDescription

OBS.: Caso não queira passar um filtro colocar os seguintes dados:

Para <código do mercado> passar "-1"

Para <código do tipo de quote> passar "-1"

Para <speak> passar "-"

Para <descrição> passar "-"

Para <expired> ao passar "0" os ativos expirados são retornados e para "1" é

retornado apenas ativos ainda validos.

Exemplo:

<http://webfeeder.cedrofinances.com.br/services/quotes/listQuote/-1/-1/-/-/0>

<http://webfeeder.cedrofinances.com.br/services/quotes/listQuote/1/2/-/-/1>

<http://webfeeder.cedrofinances.com.br/services/quotes/listQuote/1/2/-/-/pet/1>

Filtro Empresa: Filtro que é habilitado quando é selecionado o mercado Bovespa, traz as empresas para esse mercado.

Filtro Contrato: Filtro que é habilitado quando é selecionado o mercado BMF, traz os contratos para esse mercado.

listQuoteMarket: o serviço retorna uma lista de Quotes de um determinado mercado, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/listQuoteMarket/<código do mercado>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/listQuoteMarket/2>

Resposta:

[Objeto Quote]

- description
- speak
- expirationDate
- marketName
- typeQuotesDescription

listQuoteMarketType: o serviço retorna uma lista de Quotes de um determinado mercado e determinado tipo de Quote, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/listQuoteMarketType/<código do mercado>/<código do tipo de quote>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/listQuoteMarket/2/1>

Resposta:

[Objeto Quote]

- description
- speak
- expirationDate
- marketName
- typeQuotesDescription

listCompanyMarket: o serviço retorna uma lista de Quotes de um determinado mercado e determinado tipo de Quote, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/listCompanyMarket/<código do mercado>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/listCompanyMarket/2`

Resposta:**[Objeto Company]**

- codeLocal
- marketCode
- name
- marketName
- segmentCode

candleLast: o serviço retorna uma lista de Candles de um determinado Quote limitado pela quantidade passada, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/candleLast/<ativo>/<período>/<quantidade>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/candleLast/petr4/15/2`

Resposta:**[Objeto Candle]**

- symbol
- price

- open
- high
- low
- previous
- quantityTrades
- volumeAmount
- volumeFinancier
- timeTrade

candleDate: o serviço retorna uma lista de Candle de um determinado Quote entre a data passada, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/candleLast/<ativo>/<YYYYMMDDHHMM>/<YYYYMMDDHHMM>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/candleDate/petr4/15/201201010000/201212300000`

Resposta:

[Objeto Candle]

- symbol
- price
- open
- high
- low
- previous
- quantityTrades
- volumeAmount
- volumeFinancier
- timeTrade

crossRanking: o serviço retorna uma lista de CrossRanking de um determinada Corretora, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/crossRanking/<mercado>/<código da corretora>`

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/crossRanking/bovespa/1>

Resposta:

[Objeto CrossRanking]

- quote
- volumeFinancier
- volumeAmount
- quantityTrades
- formattedQuantityTrades
- formattedVolumeFinancier
- formattedVolumeAmount
- type

playerRanking: o serviço retorna uma lista de PlayerRanking de um determinada Corretora, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/playerRanking/<ativo>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/playerRanking/petr4>

Resposta:

[Objeto PlayerRanking]

- palyerCode
- volumeFinancier
- volumeAmount
- quantityTrades
- player
- formattedQuantityTrades
- formattedVolumeFinancier
- formattedVolumeAmount
- type

optionsQuote: O serviço retorna as opções de um determinado ativo, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/optionsQuote/<ativo>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/optionsQuote/petr4>

Resposta:

[String[] quotesDescription]

- ["petra12","petra13","petra15","petra16", ...]

companyQuotes: O serviço retorna uma lista dos papéis de uma determinada empresa, o método é do tipo GET.

Formato da Requisição:

http://webfeeder.cedrofinances.com.br/services/quotes/companyQuotes?company=<Empresa>&types=<Tipo_do_ativo>&markets=<Mercado>

Types:

- 1 - vista
- 2 - opções
- 3 - índice
- 7 - Forex
- 10 – NYSE

Markets:

- 1 - Bovespa
- 3 - BMF
- 6 - Soma

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/companyQuotes?company=PETROBRAS&types=1,2,3,4&markets=1,2,3,4>

Resposta:

[String[] quotesDescription]

- ["petr3","petr4",...]

quotesQuery: O serviço retorna uma lista dos papéis que iniciam com a informação passada, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quotesQuery?query=<String>&n=<qtd (int)>&f=<fracionários (boolean)>&o=<opções (boolean)>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quotesQuery?query=wi&n=10&f=true&o=true`

Resposta:

[String[] quotesDescription]

- ["WINQ17","WINV17","WINZ17","WING18",...]

auctionList: O serviço retorna uma lista dos papéis que estão em leilão, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/auctionList?market=<Mercado>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/auctionList?market=bmf`

Resposta:

[Objeto AuctionList]

- Values[].Quote
- Values[].Type
- Values[].ScheduledTime
- Values[].ReScheduledTime
- Values[].Value
- Values[].Quantity

proventos: O serviço retorna uma lista dos proventos registrados, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/proventos?n=<numero de registros>"e=<ativo>`

`http://webfeeder.cedrofinances.com.br/services/quotes/proventos?initDate=<date>&fimDate=<date>"e=<ativo>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/proventos?n=10"e=petr4`

`http://webfeeder.cedrofinances.com.br/services/quotes/proventos/n/10/petr4`

`http://webfeeder.cedrofinances.com.br/services/quotes/proventos?initDate=20170701&fimDate=20170801"e=petr4`

`http://webfeeder.cedrofinances.com.br/services/quotes/proventos/d/20170701/20170801/petr4`

Resposta:

[Objeto Earnings]

- Values[].Quote
- Values[].DateTime
- Values[].Proceed

marketSectors: O serviço retorna uma lista dos setores e subsetores registrados, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/marketSectors`

`http://webfeeder.cedrofinances.com.br/services/quotes/marketSectors?market=<Mercado>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/marketSectors`

`http://webfeeder.cedrofinances.com.br/services/quotes/marketSectors?market=bmf`

Resposta:

[Objeto MarketSectors]

- Bovespa | Bmf.Sec | Sub[].IdSec
- Bovespa | Bmf.Sec | Sub[].Desc
- Bovespa | Bmf.Sub[].IdSub

marketSubSectorQuotes: O serviço retorna uma lista dos ativos de um determinado sub setor, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/marketSubSectorQuotes?sub=IdSub>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/marketSubSectorQuotes?sub=3>

Resposta:

[String[] quotesDescription]

- ["APT13", "APT14", "CSNA3", "FBMC3", "FBMC4", ...]

listContractBMF: Serviço que retorna todos os contratos BMF. Método do tipo GET

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/listContractBMF>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/listContractBMF>

Resposta:

[Objeto BMFContracts]

- bmfContractsCode
- bmfContractsSymbol
- bmfContractsName
- quotesIsinPaper
- bmfContractsCode

quoteInformation: Serviço utilizado para solicitar informações de um ativo. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/quoteInformation?description=ATIV O>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/quoteInformation?description=petr4>

Resposta:

[Objeto QuoteInformation]

- marketCode
- quotesLotDefault
- quotesQuotationForm
- quotesIsinPaper
- companyName
- segmentName
- typeQuotesCode
- classificationName
- quotesCurrencyDescription
- quotesExpirationDate
- stockOptionsPe
- stockOptionsYear

quoteTimesTrade: Utilizado para requisitar dados sobre as cotações do ativo, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/quoteTimesTrade/<símbolo-do-ativo>/<indicador-de-negocio>/<limite>/<offset>/<dataInicio>/<dataFim>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/quoteTimesTrade/petr4/0/50/0/20170101/20171010>

Campos para o envio:

Símbolo do ativo* : Símbolo do ativo no mercado

indicador de negocio* : 1 = indica negocio / 0 = indica qualquer cotação

limite* : Número de linhas retornadas, se limite = -1 retorna um número constante de linhas (a média de cotações de um dia)

offset* : Quantidade de linhas que deverá ignorar do início do retorno da consulta

dataInicio * : Data inicial do histórico. yyyyymmdd ou yyyyymmddHHmm

dataFim * : Data final do histórico. `yyyymmdd` ou `yyyymmddHHmm`

*Campos obrigatórios

Resposta:

- Preço : Preço do negócio
- Quantidade : Quantidade do negócio
- Corretora comprando : Identificador da corretora que efetuou a compra
- Corretora vendendo : Identificador da corretora que efetuou a venda
- bid : Melhor oferta de compra
- ask : Melhor oferta de venda
- volume bid : Volume acumulado das melhores ofertas de compra
- volume ask : Volume acumulado das melhores ofertas de venda
- volume better bid : Volume da melhor oferta de compra
- volume better ask : Volume da melhor oferta de venda
- indicador de negócio : 1 = indica negocio / 0 = indica qualquer cotação
- Data/Hora : Formato: `yyyymmddHHMMSS`
- Identificador do negócio : Identificador único do negócio, se o indicador de negócio for igual a 0, esse campo será vazio.

Resumo de Mercado

Utilizado para solicitar o resumo de um mercado.

highListOptions: O serviço retorna uma lista das maiores altas de Opções, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/highListOptions/<mercado>`

Exemplo

`http:// webfeeder.cedrofinances.com.br /services/quotes/highListOptions/bovespa`

fallListOptions: O serviço retorna uma lista das maiores baixas de Opções, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/fallListOptions/<mercado>`

Exemplo

`http:// webfeeder.cedrofinances.com.br /services/quotes/fallListOptions/bovespa`

highVolumeListOptions: O serviço retorna uma lista das maiores volumes altos de Opções, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/highVolumeListOptions/<mercado>`

Exemplo

`http:// webfeeder.cedrofinances.com.br /services/quotes/ highVolumeListOptions /bovespa`

fallVolumeListOptions: O serviço retorna uma lista das maiores volumes baixos de Opções, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/fallVolumeListOptions/<mercado>`

Exemplo

`http:// webfeeder.cedrofinances.com.br /services/quotes/fallVolumeListOptions /bovespa`

highListNegotiation: O serviço retorna uma lista dos mais negociadas a vista, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/highListNegotiation/<mercado>`

Exemplo

`http:// webfeeder.cedrofinances.com.br/services/quotes/highListNegotiation/bovespa`

highListFuturo: O serviço retorna uma lista das maiores altas futuro, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/highListFuturo/<mercado>`

Exemplo

`http:// webfeeder.cedrofinances.com.br /services/quotes/highListFuturo /bmf`

fallListFuturo: O serviço retorna uma lista das maiores baixas futuro, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/fallListFuturo/<mercado>`

Exemplo

`http://webfeeder.cedrofinances.com.br/services/quotes/fallListFuturo/bmf`

highVolumeListFuturo: O serviço retorna uma lista das maiores volumes futuro, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/highVolumeListFuturo/<mercado>`

Exemplo

`http://webfeeder.cedrofinances.com.br/services/quotes/highVolumeListFuturo/bmf`

fallVolumeListFuturo: O serviço retorna uma lista dos menores volumes futuro, o método é do tipo GET.

Formato

`http://webfeeder.cedrofinances.com.br/services/quotes/fallVolumeListFuturo/<mercado>`

Exemplo

`http://webfeeder.cedrofinances.com.br/services/quotes/fallVolumeListFuturo/bmf`

quoteIndexesComposition: Serviço utilizado para solicitar os indexes e a proporção de um ativo. Método do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quoteIndexesComposition/ativo`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quoteIndexesComposition/petr4`

Resposta:

[Objeto QuoteIndexesComposition]

- description
- speak
- proportion

playerNames: Serviço utilizado para requisitar o nome e o código das corretoras em um determinado mercado. Método do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/playerNames?market=<mercado code>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/playerNames?market=1`

Resposta:

```
[{"playersCode":119,"playerMarketCode":30,"playerNameFull":"ABN AMRO REAL CCVM S.A.", "playerName":"ABN", "marketCode":1}, {"playersCode":190,"playerMarketCode":6, "playerNameFull":"ABN AMRO SEC (BRASIL) CVM S.A.", "playerName":"ABN AMRO SEC", "marketCode":1}, {"playersCode":120,"playerMarketCode":39, "playerNameFull":"AGORA CTVM S.A.", "playerName":"AGORA", "marketCode":1}, {"playersCode":121,"playerMarketCode":4, "playerNameFull":"ALFA CCVM S.A.", "playerName":"ALFA", "marketCode":1}, {"playersCode":191,"playerMarketCode":98, "playerNameFull":"ALPES CCTVM LTDA.", "playerName":"ALPES", "marketCode":1}, {"playersCode":122,"playerMarketCode":226, "playerNameFull":"AMARIL FRANKLIN CTV LTDA.", "playerName":"AMARIL FRANKLIN", "marketCode":1}, {"playersCode":124,"playerMarketCode":147, "playerNameFull":"ATIVA S.A. CTCV", "playerName":"ATIVA", "marketCode":1}, {"playersCode":192,"playerMarketCode":196, "playerNameFull":"AUREUM SCCVM LTDA.", "playerName":"AUREUM", "marketCode":1}]
```

indexList: Serviço utilizado para solicitar o recebimento da lista de índices de um determinado mercado. Método do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/indexList?marketcode=<mercado code>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/indexList?marketcode=1`

Resposta:

```
[{"code":747360,"description":"BDRX"}, {"code":12299,"description":"IBCO"}, {"code":2743,"description":"IBOV"}, {"code":733016,"description":"IBOVE"}, {"code":388537,"description":"IBOVF"}, {"code":388536,"description":"IBOVG"}, {"code":388539,"description":"IBOVO"}, {"code":388538,"description":"IBOVT"}, {"code":388540,"description":"IBOVV"}, {"code":734056,"description":"IBRA"}, {"code":2938,"description":"IBVM"}]
```

```
["IBXL"],{"code":2960,"description":"IBXX"},{"code":713916,"description":"ICO2"},{"code":128127,"description":"ICON"},{"code":732819,"description":"IDIV"},{"code":2967,"description":"IEEX"},{"code":822361,"description":"IFIX"},{"code":666356,"description":"IFNC"},{"code":715477,"description":"IGCT"},{"code":3004,"description":"IGCX"},{"code":822363,"description":"IGNM"},{"code":12300,"description":"ILID"},{"code":734058,"description":"IMAT"},{"code":128129,"description":"IMOB"},{"code":23969,"description":"INDX"},{"code":3056,"description":"ISEE"},{"code":3058,"description":"ITAG"},{"code":3160,"description":"ITEL"},{"code":3221,"description":"IVBX"},{"code":3514,"description":"MIBV"},{"code":116943,"description":"MLCX"},{"code":116957,"description":"SMLL"},{"code":1070672,"description":"TFIX"},{"code":732846,"description":"UTIL"]}]
```

marketSectorQuotes: Serviço utilizado para solicitar o recebimento de informações dos setores e sub-setores de um mercado. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/marketSectorQuotes?market=<mercado>&sec=<sector>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/marketSectorQuotes?market=1&sec=8>

Resposta:

```
[{"subsector":30,"quote":"EBTL9"}, {"subsector":30,"quote":"EBTP1"}, {"subsector":30,"quote":"EBTP2"}, {"subsector":30,"quote":"EBTP3"}, {"subsector":30,"quote":"EBTP4"}, {"subsector":30,"quote":"LFFE3"}, {"subsector":30,"quote":"LFFE3L"}, {"subsector":30,"quote":"LFFE4"}, {"subsector":30,"quote":"LFFE4L"}, {"subsector":30,"quote":"MLFT3"}, {"subsector":30,"quote":"MLFT4"}, {"subsector":30,"quote":"OIBR3"}, {"subsector":30,"quote":"OIBR4"}, {"subsector":30,"quote":"OIBR99"}, {"subsector":30,"quote":"OIBR-DEB91B0"}, {"subsector":30,"quote":"OIBR-DEB91L0"}, {"subsector":30,"quote":"OIBR-DEB91L1"}, {"subsector":30,"quote":"OIBR-DEB92B0"}, {"subsector":30,"quote":"OIBR-DEB92L0"}, {"subsector":30,"quote":"OIBR-DEB92L1"}, {"subsector":30,"quote":"PITI3"}, {"subsector":30,"quote":"VIVT3"}, {"subsector":30,"quote":"VIVT4"}, {"subsector":30,"quote":"VIVT-DEB41B0"}, {"subsector":30,"quote":"VIVT-DEB41L0"}, {"subsector":30,"quote":"VIVT-DEB41L1"}, {"subsector":30,"quote":"VIVT-DEB42B0"}, {"subsector":30,"quote":"VIVT-DEB42L0"}, {"subsector":30,"quote":"VIVT-DEB42L1"}, {"subsector":31,"quote":"ATOM3"}, {"subsector":31,"quote":"INET1"}, {"subsector":31,"quote":"INET3"}, {"subsector":31,"quote":"TIMP3"}]
```

quoteTimesTradeDate *: Serviço utilizado para requisitar dados sobre as cotações do ativo. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/quoteTimesTradeDate/<quote>/<data>>

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/quoteTimesTradeDate/petr4/20180218`

* o serviço `quoteTimesTradeDate` retorna os dados de maneira compressada.

selicRate: Serviço de retorno da taxa selic. Método do tipo GET.

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/selicRate`

Resposta

`{ String selic }`

events: Serviço de retorno de eventos. Método do tipo GET.

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/events`

Resposta

`{ String eventsDate , String eventsTitleBr , String eventsTitleUs , String eventsDescBr , String eventsDescUs , String eventsUrl , String eventsDate }`

eventsDatagro: Serviço de retorno de eventos da datagro. Método do tipo GET.

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/quotes/eventsDatagro`

Resposta

`{ String eventsDate , String eventsTitleBr , String eventsTitleUs , String eventsDescBr , String eventsDescUs , String eventsUrl , String eventsDate }`

eventsCedro: Serviço de retorno de eventos da Cedro. Método do tipo GET.

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/eventsCedro>

Resposta

```
{ String eventsDate , String eventsTitleBr , String eventsTitleUs , String eventsDescBr, String eventsDescUs, String eventsUrl, String eventsDate }
```

datesContractExpiration: Serviço de retorno das datas de expiração de contrato.
Método do tipo GET.

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/datesContractExpiration>

Resposta

```
{ String eventsFuturesExpirationDate, String marketDescription , String eventsFuturesExpirationSymbol }
```

holidays: Serviço de retorno de feriados. Método do tipo GET.

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/holidays?countryCode=<countryCode>>

Resposta

```
{ String country_name, String holiday_date, String holiday_description, String country_name }
```

volumeAtPrice: Utilizado para apontamento de volume negociado por um ativo em determinado valor. Método do tipo GET.

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/volumeAtPrice/petr4>

Resposta

```
{"values":[{"symbol":"PETR4","price":"21.1","businessQuantityBuy":"0","volumeBuy":"0","businessQuantitySell":"44","volumeSell":"58700","clean":false},{symbol":"PETR4","price":"21.11","businessQuantityBuy":"5","volumeBuy":"10900","businessQuantitySell":"53","volumeSell":"54800","clean":false}]}
```

permissionsMarketUser: Utilizado para consultar as permissões de mercado de um usuário. Método do tipo GET.

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/permissionsMarketUser?user=34565>

Resposta

```
[{"marketCode":7,"marketName":"Forex"}, {"marketCode":8,"marketName":"Indicators"}, {"marketCode":1,"marketName":"Bovespa"}, {"marketCode":3,"marketName":"BMF"}, {"marketCode":4,"marketName":"Indices"}, {"marketCode":5,"marketName":"Money"}, {"marketCode":6,"marketName":"Soma"}, {"marketCode":2,"marketName":"Dow Jones"}, {"marketCode":10,"marketName":"NYSE"}, {"marketCode":12,"marketName":"Nasdaq"}, {"marketCode":11,"marketName":"Bats"}, {"marketCode":13,"marketName":"CFD"}, {"marketCode":14,"marketName":"BVL"}, {"marketCode":15,"marketName":"SPIndexes"}, {"marketCode":18,"marketName":"CME"}, {"marketCode":19,"marketName":"CME MINI"}, {"marketCode":16,"marketName":"Liffe"}, {"marketCode":17,"marketName":"Euronext Indices"}, {"marketCode":20,"marketName":"Reuters"}, {"marketCode":21,"marketName":"CBOT"}, {"marketCode":23,"marketName":"NYMEX"}, {"marketCode":24,"marketName":"ICE"}, {"marketCode":22,"marketName":"COMEX"}, {"marketCode":25,"marketName":"KANSAS"}, {"marketCode":39,"marketName":"BTC"}, {"marketCode":30,"marketName":"Bitcoin"}, {"marketCode":44,"marketName":"DATAGRO"}, {"marketCode":45,"marketName":"INews"}]
```

quotesQueryMarket: Utilizado para consultar os ativos em uma lista de mercados. Método do tipo GET.

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/quotes/quotesQueryMarket?query=petr&marketList=1,3>

Resposta

```
[{"symbol":"PTPA2","speak":"PETROPAR","marketCode":"1"}, {"symbol":"PTPA10","speak":"PETROPAR","marketCode":"1"}, {"symbol":"PTPA9","speak":"PETROPAR","marketCode":"1"}, {"symbol":"PETR4","speak":"PETROBRAS","marketCode":"1"}, {"symbol":"PETR3","speak":"PETROBRAS","marketCode":"1"}, {"symbol":"OGXP3","speak":"OGX PETROLEO","marketCode":"1"}, {"symbol":"HRT3","speak":"HRT PETROLEO","marketCode":"1"}, {"symbol":"PRIO3","speak":"PETRORIO","marketCode":"1"}, {"symbol":"BRDT3","speak":"PETROBRAS BR","marketCode":"1"}]
```


II- Módulo News

1. Métodos disponíveis:

- **newsByDate**
- **newsByCode**
- **newsByAgency**
- **newsLast**
- **newsRelevantFacts**
- **newsPagination**
- **newsQuery**
- **newsQueryPagination**
- **newsByDatePagination**
- **newsByAgenciesPagination**
- **newsAgencyByDatePagination**
- **newsAgencyQueryPagination**
- **newsAgency**
- **newsRelevantFactsByQuote**
- **newsRelevantFactsByAgency**

2. Requisição e Resposta:

newsByDate: o serviço retorna as notícias contidas entre a data inicial informada e a data final, sendo ela limitada a cem notícias, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsByDate/<YYYY-MM-DD>/<YYYY-MM-DD>`

`http://webfeeder.cedrofinances.com.br/services/news/newsByDate/<YYYY-MM-DD HH-MM-SS>/<YYYY-MM-DD HH-MM-SS>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsByDate/2012-01-01/2012-12-30`

`http://webfeeder.cedrofinances.com.br/services/news/newsByDate/2012-01-01 10:00:00/2012-12-30 11:00:00`

Resposta:

[Objeto News]

- **newsCode**
- **agencyCode**

- agencyDescription
- title
- description
- categoryCode
- shadowCode
- newsDate

newsByCode: o serviço retorna as notícias que contenha o código solicitado, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsByCode/<código-notícia>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsByCode/1070192>

Resposta:

[Objeto News]

- newsCode
- agencyCode
- agencyDescription
- title
- description
- categoryCode
- shadowCode
- newsDate

newsByAgency: o serviço retorna todas as notícias que contenham o código da agência solicitada, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsByAgency/<código-agência>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsByAgency/11>

Resposta:

[Objeto News]

- newsCode
- agencyCode
- agencyDescription

- title
- description
- categoryCode
- shadowCode
- newsDate

newsLast: o serviço retorna as últimas notícias pela quantidade solicitada, sendo limitada a cem, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsLast/<quantidade-notícia>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsLast/2`

Resposta:

[Objeto News]

- newsCode
- agencyCode
- agencyDescription
- title
- description
- categoryCode
- shadowCode
- newsDate

newsRelevantFacts: o serviço retorna as notícias contidas entre a data inicial informada e a data final que contenham o texto "fato relevante" ou "fatos relevantes" na sua descrição ou título, sendo ela limitada a cem notícias, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsRelevantFacts/<YYYY-MM-DD>/<YYYY-MM-DD>`

`http://webfeeder.cedrofinances.com.br/services/news/newsRelevantFacts/<YYYY-MM-DD HH-MM-SS>/<YYYY-MM-DD HH-MM-SS>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsRelevantFacts/2012-01-01/2012-12-30`

`http://webfeeder.cedrofinances.com.br/services/news/newsRelevantFacts/2012-01-01 10:00:00/2012-12-30 11:00:00`

Resposta:

[Objeto News]

- newsCode
- agencyCode
- agencyDescription
- title
- description
- categoryCode
- shadowCode
- newsDate

newsPagination: o serviço retorna as notícias na página solicitado, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsPagination/<size>/<page>/<requestId>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsPagination/100/1`

`http://webfeeder.cedrofinances.com.br/services/news/newsPagination/100/5/14858`

Resposta:

[Objeto GenericPagination]

- Pagination.page
- Pagination.size
- Pagination.lastNewsCode
- Pagination.lastPage
- Pagination.requestId
- Pagination.sizeTotal
- List[].newsCode
- List[].agencyCode
- List[].agencyDescription
- List[].title
- List[].description
- List[].categoryCode
- List[].shadowCode
- List[].newsDate

newsQuery: o serviço retorna as notícias em um período entre datas que contenha o texto no título ou na descrição, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsQuery/<dtInicial>/<dtFinal>/<text
o>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsQuery/08122017000000/0912201
7235959/leilao`

`http://webfeeder.cedrofinances.com.br/services/news/newsQuery/-/-/leilao`

Resposta:

[Objeto News]

- newsCode
- agencyCode
- title
- description
- categoryCode
- shadowCode
- newsDate

newsQueryPagination: o serviço retorna as notícias em um período entre datas que contenha o texto no título ou na descrição de forma paginada, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsQueryPagination/<size>/<page>/
<requestId>/<query>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsQueryPagination/100/5/14858/le
ilao`

Resposta:

[Objeto News]

newsCode
agencyCode
title
description
categoryCode

shadowCode
newsDate

newsByDatePagination: o serviço retorna as notícias em um período entre datas de forma paginada, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsByDatePagination/<size>/<page>/<requestId>/<dateIni>/<dateFim>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsByDatePagination/100/5/14858/10102018/15102018`

Resposta:

[Objeto News]
newsCode
agencyCode
title
description
categoryCode
shadowCode
newsDate

newsByAgenciesPagination: Serviço retorna as notícias de várias agências em um período entre datas e que contenha o texto no título ou na descrição de forma paginada. O método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsByAgenciesPagination/<size>/<page>/<requestId>/<agenciesCode>/<dtInicial>*/<dtFinal>*/<query>**`

*Período entre datas opcionais

**Auto-complete opcional

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsByAgenciesPagination/100/5/14858/3-4-5-6/10102018/15102018/petr4`

Resposta:

[Objeto News]

newsCode
agencyCode
title
description
categoryCode
shadowCode
newsDate

newsAgencyByDatePagination: Serviço retorna as notícias de uma agência em um período entre datas, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsAgencyByDatePagination/<size>/<page>/<requestId>/<agencyCode>/<dtInicial>/<dtFinal>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsAgencyByDatePagination/100/5/14858/60/2017-10-10/2017-11-30`

Resposta:

[Objeto News]

newsCode
agencyCode
title
description
categoryCode
shadowCode
newsDate

newsAgencyQueryPagination: Serviço retorna as notícias de uma agência que contenha o texto no título ou na descrição de forma paginada, o método é do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/news/newsAgencyQueryPagination/<size>/<page>/<requestId>/<agencyCode>/<query>*`

* Auto-complete opcional (tornando-se noticias por agência paginada).

Exemplo da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsAgencyQueryPagination/100/5/14858/3/vale3>

Resposta:

[Objeto News]

newsCode
agencyCode
title
description
categoryCode
shadowCode
newsDate

newsAgency: Serviço retorna as agências de noticias. O método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsAgency>

Exemplo da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsAgency>

Resposta:

[Objeto News]

agency_code
agency_description
agency_symbol
agency_color

newsRelevantFactsByQuote: Serviço retorna as noticias mais relevantes por ativo. O método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsRelevantFactsByQuote/<dataInicio>/<dataFim>/<ativo>>

Exemplo da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsRelevantFactsByQuote/07062018000002/08062018235959/petr4>

Resposta:

[Objeto News]

news_code

agency_code

news_title

news_description

category_code

shadow_code

news_photo_description

news_date

newsRelevantFactsByAgency: Serviço retorna as notícias mais relevantes por agência. O método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsRelevantFactsByAgency/<dataInicio>/<dataFim>/<agencia1-agencia2- ...- agenciaN >>

Exemplo da Requisição:

<http://webfeeder.cedrofinances.com.br/services/news/newsRelevantFactsByAgency/07062018000002/08062018235959/1-3-4>

Resposta:

[Objeto News]

news_code

agency_code

news_title

news_description

category_code
shadow_code
news_photo_description
news_date

II- Módulo Negotiation

3. Métodos disponíveis:

- **sendNewOrderSingle**
- **editOrder**
- **cancelOrder**
- **financialAccountInformation**
- **financialAccountInformationCompl**
- **historyOrder**
- **positionBMF**
- **positionBovespa**
- **brokerServiceLogin**
- **brokerServiceLogout**
- **brokerServiceChangePassword**
- **brokerServiceForgotPassword**
- **brokerServiceUserInformation**
- **brokerServiceAccountInformation**
- **warrantyList**
- **warrantyOperationUpdate**
- **warrantyOperation**
- **blockUser**
- **brokeragePDF**
- **fixedIncomeList**
- **brokerageDatesRequestBovespa**
- **brokerageDatesRequestBMF**
- **limits**
- **createUpdateLimitRequest**
- **statementBmf**
- **brokerageBovespa**
- **brokerageBmf**
- **brokerServiceAccountList**
- **fixedIncomeAccountInformation**
- **fundsAccountInformation**
- **clubsAccountInformation**
- **treasuryAccountInformation**

- **rescueDepositInformation**
- **bankAccountInformation**
- **updateRescueDeposit**
- **fundsList**
- **quoteUpdate**

4. Requisição e Resposta:

sendNewOrderSingle: o serviço para enviar ordem para o OMS, o método é do tipo POST.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/sendNewOrderSingle?account=<conta>&market=<mercado>&price=<preço>"e=<ativo>&qtd=<quantidade>&validityorder=<data>&targettrigger=<preço de StopGain>&targetlimit=<preço limite StopGain>&stoptrigger=<preço de Stop>&stoplimit=<stopLimit>&type=<tipo>&side=<direção>&username=<usuário>&sourceaddress=<identificador do cliente>&orderstrategy=<Estratégia da ordem>&appname=<Nome da aplicação>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/sendNewOrderSingle?account=1000&market=XBSP&price=13.00"e=PETR4&qtd=100&type=Limited&side=BUY&username=10010&sourceaddress=10.1.11.12&orderstrategy=POSITION&appname=HB`

Campos para o envio da ordem:

side * - Direção (Venda 'SELL', Compra 'BUY')

qtd * - Quantidade

maxfloor - Número máximo de contratos da ordem a serem mostrados no livro de ofertas a cada momento

type * - Tipo da Ordem ("OnClose", "MarketProtected", "Limited", "Stop", "Market" e "Start")

stoptrigger - Preço de Stop para uma ordem Stop Limit. Obrigatório apenas quando OrdType = 4 ou S. (StopPx)

price - Ordem do tipo LIMITED deve ser informado o Price *.

stoplimit - Ordem do tipo STOP deve ser informado o StopLimit *. (Price)

targettrigger - Preço de StopGain para uma ordem Stop Limit. Possível apenas em OrdType = 4 ou S, mas não obrigatório. (StopGainPx)

targetlimit - Preço limite para StopGain. (Price2)

movingstart - Preço de início móvel, configurado para Ordens Start/Stop Móvel. Obrigatório quando InitialChange é informado

initialchangetype - Código que determina se valor em campo InitialChange será representação percentual ou absoluta. Valores possíveis: (1 = Percentual, 2 = Absoluto). Obrigatório quando InitialChange é informado.

initialchange - Preço de alteração inicial do Price, quando LastTrade ultrapassar MovingStart, configurado quando ordem Start/Stop Móvel. Obrigatório quando campos MovingStart e InitialChangeType forem informados. (Se percentual, 1.0 = 100%)

pricepercentage - Percentual usado para calcular o preço limite em uma ordem Start/Stop Móvel.

timeinforce - Especifica o tipo de duração da ordem. ("DAY", "GTC", "OPG", "IOC", "FOK", "GTD", "ATC" e "GFA")

validityorder - A data ser informada no formato ddmmyyyy

account * - Conta

username * - Usuário

bypasssuitability - Evitar a validação por suitability ("true", "false")

market * - Mercado

quote * - Ativo

sourceaddress* - Utilizado para armazenar a origem da ordem

orderstrategy - Estratégia da ordem ("POSITION" ou "DAYTRADE")

appname - Nome da aplicação

ordertag - Tag (observação) informada para a ordem

settltype - Tag do OMS SettlementType

fullfilled - Define se a estratégia OSO será disparada somente com a ordem totalmente executada

clordid - Identificador da ordem. Deve ser único por dia ou sessão, e definido pelo iniciador. Se não informado será gerado

contratrader - Tag do OMS ContraTrader

recopidfromapp - Tag do OMS ReceivingOperationIDFromApp

grossvalue - Tag do OMS GrossValue

settlaccountcc - Tag do OMS SettlementAccountCc

appdate - Tag do OMS ApplicationDate (ddmmyyyy)

minqty - Quantidade mínima da ordem a ser executada.

stopprocess - Tag do OMS StopProcess

Campos usados em estratégias One Send Other:

nlgstype - Tipo da ordem a ser disparada. Tipos possíveis: 2 = Limite, 4 = Stop Limite

nlgstimeinforce - Tipos de vencimentos a serem enviado

nlgsoffsettype - Tipo dos valores para a estratégia. Valores possíveis: 0 = Price, 2 = Ticks

nlgspercent - Percentual da perna irá consumir da ordem executada

nlgsprice - Preço somado ao valor da execução para o preço de perda enviado na estratégia

nlgsstoppx - Tick ou Preço somado ao valor da execução para o preço de disparo de perda enviado na estratégia

nlgsprice2 - Preço somado ao valor da execução para o preço de ganho enviado na estratégia

nlgsstopgainpx - Tick ou Preço somado ao valor da execução para o preço de disparo de ganho enviado na estratégia

nlgsoffsetvalue - Tick somado ao valor da execução para o preço de perda/ganho enviado na estratégia

Resposta:

[Objeto ResponseNegotiation]

- code
- message

editOrder: o serviço para editar ordem para o OMS, o método é do tipo POST.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/editOrder?account=<conta>&orderid=<orderid>"e=<ativo>&side=<side>&qtd=<quantidade>&validityorder=<data>&targettrigger=<preço de StopGain>&targetlimit=<preço limite StopGain>&stoptrigger=<preço de Stop>&stoplimit=<stopLimit>&username=<username>&market=<market>&price=<price>&type=<type>&sourceaddress=<identificador do cliente>&orderstrategy=<Estratégia da ordem>&appname=<Nome da aplicacao>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/editOrder?account=1000&orderid=20170126175003137"e=PETR4&side=BUY&qtd=100&validityorder=10222017&targettrigger=22.5&targetlimit=23.0&stoptrigger=22.0&stoplimit=22.0&username=100100&market=XBSP&price=13.00&type=Limited&sourceaddress=10.1.11.12&orderstrategy=POSITION &appname=HB`

Campos para o envio da ordem:

origclordid - ClOrdID da ordem que o cliente está tentando editar

orderid - Identificador da ordem como definido pelo Crystal Broker. Se esse campo estiver presente, o valor de OrigClOrdID é ignorado

side * - Direção (Venda 'SELL', Compra 'BUY')

qtd * - Quantidade

maxfloor - Número máximo de contratos da ordem a serem mostrados no livro de ofertas a cada momento

type * - Tipo da Ordem ("OnClose", "MarketProtected", "Limited", "Stop", "Market" e "Start")

stoptrigger - Preço de Stop para uma ordem Stop Limit. Obrigatório apenas quando OrdType = 4 ou S. (StopPx)

price - Ordem do tipo LIMITED deve ser informado o Price *.

stoplimit - Ordem do tipo STOP deve ser informado o StopLimit *. (Price)

targettrigger - Preço de StopGain para uma ordem Stop Limit. Possível apenas em OrdType = 4 ou S, mas não obrigatório. (StopGainPx)

targetlimit - Preço limite para StopGain. (Price2)

movingstart - Preço de início móvel, configurado para Ordens Start/Stop Móvel. Obrigatório quando InitialChange é informado

initialchangetype - Código que determina se valor em campo InitialChange será representação percentual ou absoluta. Valores possíveis: (1 = Percentual, 2 = Absoluto). Obrigatório quando InitialChange é informado.

initialchange - Preço de alteração inicial do Price, quando LastTrade ultrapassar MovingStart, configurado quando ordem Start/Stop Móvel. Obrigatório quando campos MovingStart e InitialChangeType forem informados. (Se percentual, 1.0 = 100%)

pricepercentage - Percentual usado para calcular o preço limite em uma ordem Start/Stop Móvel.

timeinforce - Especifica o tipo de duração da ordem. ("DAY", "GTC", "OPG", "IOC", "FOK", "GTD", "ATC" e "GFA")

validityorder - A data ser informada no formato ddmmyyyy

account * - Conta

username * - Usuário

market * - Mercado

quote * - Ativo

enteringtrader - Usuário que enviou a ordem originalmente.

sourceaddress* - Utilizado para armazenar a origem da ordem

orderstrategy - Estratégia da ordem ("POSITION" ou "DAYTRADE")

appname – Nome da aplicação

clordid – Identificador da ordem. Deve ser único por dia ou sessão, e definido pelo iniciador. Se não informado será gerado.

Resposta:

[Objeto ResponseNegotiation]

- code
- message

cancelOrder: o serviço para cancelar ordem para o OMS, o método é do tipo POST.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/cancelOrder?orderid=<orderid>&origclordid=<origclordid>"e=<ativo>&side=<side>&qtd=<quantidade>&username=<username>&market=<market>&sourceaddress=<identificador do cliente>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/cancelOrder?orderid=20170126175003137"e=PETR4&side=BUY&qtd=100&username=100100&market=XBSP&sourceaddress=10.1.11.12>

Campos para o envio da ordem:

origclordid* - ClOrdID da ordem que o cliente está tentando cancelar

orderid - Identificador da ordem como definido pelo Crystal Broker. Se esse campo estiver presente, o valor de OrigClOrdID é ignorado

quote * - Ativo

side * - Direção da ordem (Venda 'SELL', Compra 'BUY')

qtd * - Número de contratos a serem ordenados. Caso a ordem já esteja parcialmente executada enviar a quantidade de contratos não executados.

username * - Usuário

market * - Mercado

enteringtrader - Usuário diferente do dono da ordem, assessores ou administradores

sourceaddress* - Utilizado para armazenar a origem da ordem

appname – Nome da aplicação

clordid – Identificador da ordem. Deve ser único por dia ou sessão, e definido pelo iniciador. Se não informado será gerado.

contratrader – Tag do OMS ContraTrader

cancelreason – Tag do OMS CancellationReason

idcore – Tag do OMS IdCore

Resposta:

[Objeto ResponseNegotiation]

- code
- message

financialAccountInformation: Mensagem de resposta para a consulta financeira. Pode funcionar como um retrato da conta, ou mesmo ter os dados atualizados a toda mudança em alguma variável da conta. O método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/financialAccountInformation/account/market>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/financialAccountInformation/4101/XBSP>

Resposta:

[Objeto FinancialInformationBean]

marketID - (1301) Mercado;
account - (1) Conta;
financialAccountRejReason - (10319) Motivo da rejeição;
text - (58) Texto da rejeição;
saldoAbertura - INBA;
saldoEmConta - AVBA;
contaMargemTotal - ACMT;
contaMargemUtilizada - ACMU;
proventos - DVDS;
tesouroDireto - TRDI;
clubes - CLFU;
rendaFixa - INFU;
fundos - FNDS;
limiteOperacional - UNQL;
alavancagemVista - CHLL;
alavancagemOpcoes - OPLL;
alavancagemBMF - BMFLL;
lucroPrejuizoOpen - OPL;

lucroPrejuizoClose - CLPL;
lucroPrejuizoTotal - (OPL + CLPL);
patrimonioInicial - INEQ;
patrimonioOnline - TOEQ;
creditEntryDay0 - CED0;
creditEntryDay1 - CED1;
creditEntryDay2 - CED2;
creditEntryDay3 - CED3;
debitEntryDay0 - DED0;
debitEntryDay1 - DED1;
debitEntryDay2 - DED2;
debitEntryDay3 - DED3;
saldoProjOnline - (INBA + CED0 - DED0 + CED1 - DED1 + CED2 - DED2 + CED3 - DED3);
saldoD0 - (CED0 - DED0);
saldoD1 - (CED1 - DED1);
saldoD2 - (CED2 - DED2);
saldoD3 - (CED3 - DED3);
margemGarantias - BMFWT;
consideradoPI - BMCL;
margemBMFLivre - BMFMA;
ajustePregaoBMF - BTAD;
ajustePregaoBOV - BPAD;
saldoProjetadoInicial - INPRBA;
taxas - INPRBA;
custosCorretagemBov - BGBSP;
custosCorretagemBMF - BGBMF;
emolumentos - EMOL;
custodiaDesagiadaOpcoes - OPLVG;
custodiaDesagiadaVista - LEVE;
margemGarantiaBovTotal - BSPWT;
margemGarantiaBovConsideradoPL - GNMA;
openProfitLossDaytrade - OPLDT;
closeProfitLossDaytrade - CPLDT;
dayTradeAdjustment - DTAD;
blockedValueLimit - BLVL;
bmfNotExecutedPurchaseAmount - BNEP;
bmfNotExecutedSellAmount - BNES;
cashOpenPurchases - CHOP;
optionsNotExecutedPurchaseAmount - OPOP;
cashOpenSells - CHOS;
optionsNotExecutedSellAmount - OPOS;
warrantyRequired - WREQ;
warrantyDeposited - WDEP;

financialAccountInformationCompl: Mensagem de resposta para a consulta financeira completa. Pode funcionar como um retrato da conta, ou mesmo ter os dados atualizados a toda mudança em alguma variável da conta. O método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/financialAccountInformationCompl/account/market>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/financialAccountInformationCompl/4101/XBSP>

Resposta:

[Objeto FinancialInformationComplBean]

marketID - (1301) Mercado;
account - (1) Conta;
financialAccountRejReason - (10319) Motivo da rejeição;
text - (58) Texto da rejeição;
finalMarkToMarketAmount - FMTM;
incrementalMarkToMarketAmount - IMTM;
tradeVariationAmount - TVAR;
startOfDayMarkToMarketAmount - SMTM;
premiumAmount - PREM;
cashResidualAmount - CRES;
cashAmount - CASH;
valueAdjustedAmount - VADJ;
currentValueAmount - CVA;
marginAccountBalance - MACB;
totalBlockedBalance - TBOB;
blockedMargin - BLMA;
futureSettlementBalance - FUSB;
blockedFromOthers - BLFO;
availableWithdrawnBalance - AWTB;
adjustedBalance - ADBA;
initialMarginBalance - IMAB;
maintainanceMarginBalance - MMBA;
availableNegotiationBalance - ANBA;
recompositionValue - REVA;

availableBalance - AVBA;
currentDayBalance - CDBA;
entryDay0 - END0;
entryDay1 - END1;
entryDay2 - END2;
entryDay3 - END3;
financialVolume - FIVO;
projectedBalance - PRBA;
initialBalance - INBA;
creditEntryD0 - CED0;
creditEntryD1 - CED1;
creditEntryD2 - CED2;
creditEntryD3 - CED3;
debitEntryD0 - DED0;
debitEntryD1 - DED1;
debitEntryD2 - DED2;
debitEntryD3 - DED3;
startCreditEntryD1 - CESD1;
startCreditEntryD2 - CESD2;
startDebitEntryD1 - DESD1;
startDebitEntryD2 - DESD2;
financialLeverage - FNLE;
overdraftCredit - OVCR;
balanceCashMarketBovespa - BCMB;
balanceCashMarketBmf - BCMF;
balanceUncoveredOperation - BUOP;
balanceOptions - BOPT;
previousClosePrice - PCP;
currentInstrumentPrice - CIP;
openAvgValueToday - OAVT;
totalPurchaseAmt - TPV;
totalSaleAmt - TSV;
blocked - BCKA;
openedPurchaseAmount - OPAM;
openedSellAmount - OSAM;
cashLimit - CHLT;
optionLimit - OPLT;
dailyFinancialActivities - DFAT;
bspCashOperationCredit - BSCC;
bspOptionOperationCredit - BSOC;
bmfOperationCredit - BMFC;
operationEquityCredit - OEC;
financialAccountRisk - FARK;

overdraftLimit - OVLT;
overdraftAmmount - OVAM;
potencialOverdraftAmmount - POVA;
bmfBlockedMargin - BMBM;
adjustD0 - ADD0;
adjustD1 - ADD1;
bmfLimit - BMFL;
bmfExecutedBlockedMargin - BEBM;
bmfNotExecutedBlockedMargin - BNBM;
bmfPurchaseOpenMargin - BPOM;
bmfPurchaseExecutedMargin - BPEM;
bmfSellOpenMargin - BSOM;
bmfSellExecutedMargin - BSEM;
notExecutedPurchaseAmmount - NEPA;
notExecutedSellAmmount - NESA;
bmfNotExecutedPurchaseAmmount - BNEP;
bmfNotExecutedSellAmmount - BNES;
bmfUnblockedMargin - BUNM;
filledOverdraftOptionAmmount - FOOA;
notExecutedOverdraftOptionAmmount - NOOA;
xbmfMarginConsidered - BMCL;
cashOpenPurchases - CHOP;
optionsNotExecutedPurchaseAmmount - OPOP;
cashOpenSells - CHOS;
optionsNotExecutedSellAmmount - OPOS;
cashExecutedPurchases - CHEP;
optionsExecutedPurchases - OPEP;
cashExecutedSells - CHES;
optionsExecutedSells - OPES;
uniqueLimit - UNQL;
blockedValueLimit - BLVL;
cashLeveragedLimit - CHLL;
optionsLeveragedLimit - OPLL;
bmfLeveragedLimit - BMFLL;
cashPurchaseLimit - CHPL;
optionsPurchaseLimit - OPPL;
bmfPurchaseLimit - BMFPL;
cashSellLimit - CHSL;
optionsSellLimit - OPSL;
bmfSellLimit - BMFSL;
cashPurchaseAmmount - CHPA;
cashSellAmmount - CHSA;
optionsPurchaseAmmount - OPPA;

optionsSellAmmount - OPSA;
bmfPurchaseAmmount - BMFPA;
bmfSellAmmount - BMFSA;
initialProjectedBalance - INPRBA;
initialLimit - INLT;
initialLimitBmf - INLTB;
bslimitOperationCredit - BSOPCR;
initialStockPortfolioValue - INSPV;
initialOptionsPortfolioValue - INOPV;
initialBmfPortfolioValue - INBPV;
totalStockPortfolioValue - TSPV;
totalOptionsPortfolioValue - TOPV;
totalBmfPortfolioValue - TBPV;
totalOnlineStockPortfolioValue - TOSPV;
totalOnlineOptionsPortfolioValue - TOOPV;
totalOnlineBmfPortfolioValue - TOBPV;
openProfitLoss - OPL;
closedProfitLoss - CLPL;
openProfitLossAux - OPLX;
closedProfitLossAux - CLPLX;
openProfitLossDaytrade - OPLDT;
closedProfitLossDaytrade - CPLDT;
initialEquity - INEQ;
totalEquity - TOEQ;
xbspMarginConsidered - GNMA;
optionsMargin - OPMA;
bmfPositionAdjust - BPAD;
dayTradeAdjustment - DTAD;
bmfTradingAdjust - BTAD;
ipoBlocked - IPOB;
creditTed - CTED;
manualAdjust - MADJ;
balanceWithoutLimit - BWLT;
optionsLeverageLimit - LOLI;
treasuryDirect - TRDI;
fixedIncome - INFU;
clubs - CLFU;
funds - FNDS;
fixedIncomeConsidered - INFUC;
clubsConsidered - CLFUC;
fundsConsidered - FNDSC;
dividends - DVDS;
xbspWarrantyTotal - BSPWT;

xbmfWarrantyTotal - BMFWT;
xbmfMarginAvailable - BMFMA;
brokerageXbmf - BGBMF;
brokerageXbsp - BGBSP;
taxes - TAX;
emolument - EMOL;
accountMarginTotal - ACMT;
accountMarginUsed - ACMU;
leverage - LEVE;
quoteOptionLeverage - OPLVG;
equityLeveraged - EQLVG;
d1CreditDaytrade - D1CDT;
d1DebitDaytrade - D1DDT;
d3CreditDaytrade - D3CDT;
d3DebitDaytrade - D3DDT;
d1CreditPosition - D1CPO;
d1DebitPosition - D1DPO;
d3CreditPosition - D3CPO;
d3DebitPosition - D3DPO;
negativePositionAmount - NPA;
negativeOptPositionAmount - NOPA;
creditLimit - CRDL;
buyAveragePrice - BAVP;
sellAveragePrice - SAVP;
warrantyRequired - WREQ;
warrantyDeposited - WDEP;
profitTotalLoss - (OPL + CLPL);
balanceProjOnline - (INBA + CED0 - DED0 + CED1 - DED1 + CED2 - DED2 + CED3 - DED3);
balanceD0 - (CED0 - DED0);
balanceD1 - (CED1 - DED1);
balanceD2 - (CED2 - DED2);
balanceD3 - (CED3 - DED3);

historyOrder serviço para pedir histórico de ordens para o OMS, o método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/historyOrder/account/market/datalnicio/dataFim/symbol/status/direction/orderId/statusGeneric/queryType>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/historyOrder/4101/xbsp/25092017/26092017/petr4/new/buy>

<http://webfeeder.cedrofinances.com.br/services/negotiation/historyOrder/10000/XBS>
P/-/-/-/-/20171023110802230

<http://webfeeder.cedrofinances.com.br/services/negotiation/historyOrder/10000/XBS>
P/07112017/08112017/-/-/-/canceladas

<http://webfeeder.cedrofinances.com.br/services/negotiation/historyOrder/10000/XBS>
P/-/-/-/-/20171023110802230/0

Campos para o envio da ordem:

account * - Conta

market * - Mercado

dataInicio (Obrigatório se não informado o OrderId) - formato ddmmaaaa.

dataFim (Obrigatório se não informado o OrderId) - formato ddmmaaaa.

symbol – Ativo

status – Status da ordem

direction – Direção da ordem

orderId – Id da ordem

statusGeneric – Status genéricos (Abertas, Canceladas, Executadas, Vigentes)

queryType – Especificar se o OMS precisa ir no banco de dados ou apenas na memória.

Valores possíveis: 0 – Day (Apenas na memória), 1 – History (Ir no banco de dados)

execType – Descreve a ação que disparou a mensagem

startStopClOrdId – Identificador da ordem Start/Stop que originou esta ordem

applicationName – Tag que informa o identificador da aplicação utilizada pelo usuário

status:

-New

-PartiallyFilled

-Filled

-DoneForDay

-Canceled

-Replaced

-PendingCancel

-Rejected

-Suspend

-PendingNew

-Expired

-Received

PendingReplace

direction - BUY ou SELL

Resposta:

```
{
  "listBeans": [
    { "quote": { "value": "VALE3" }, "buyOrSell": { "value": "Venda" }, "account": { "value": "10000" },
    "state": { "value": "Completamente Executada", "code": "2" }, "quantitySupplied": { "value": "100" },
    "quantityExecuted": { "value": "100" }, "quantityRemaining": { "value": "0" }, "price": { "value": "R$ 2,00" },
    "priceStop": { "value": "" }, "priceStopGain": { "value": "" }, "price2": { "value": "" },
    "movingStart": { "value": "" }, "initialChangeType": { "value": "" }, "initialChange": { "value": "" },
    "pricePercentage": { "value": "" }, "maxFloor": { "value": "" }, "priceAverage": { "value": "R$ 42,44" },
    "type": { "value": "Limitada", "code": "2" }, "maturity": { "value": "" }, "mercado": { "value": "Bovespa", "code": "XBSP" },
    "transactTime": { "value": "21/03/2018 12:28:38.018" }, "side": { "value": "Venda", "code": "2" }, "currentCumQty": { "value": "" },
    "currentAvgPx": { "value": "" }, "orderID": { "value": "032115283812" }, "clOrdID": { "value": "" },
    "timeInForce": { "value": "Hoje", "code": "0" }, "username": { "value": "RODRIGO3" }, "text": { "value": "" },
    "sourceAddress": { "value": "" }, "orderStrategy": { "value": "DAYTRADE" }, "execType": { "value": "", "code": "F" },
    "startStopClOrdId": { "value": "" }, "applicationName": { "value": "FAST" }, "orderTag": { "value": "" },
    "settlType": { "value": "" }, "recOpIDFromApp": { "value": "" }, "grossValue": { "value": "" },
    "settlAccountCc": { "value": "" }, "applicationDate": { "value": "" }, "cancellationReason": { "value": "" },
    "idCore": { "value": "" }, "soldQty": { "value": "" }, "boughtQty": { "value": "" }, "buyAvgPx": { "value": "" },
    "sellAvgPx": { "value": "" }, "factor": { "value": "" }, "minQty": { "value": "" }, "securityType": { "value": "FUND" },
    "userId": { "value": "DMA" }, "securityDesc": { "value": "Fundo Multimercado Santander" } ] }
```

positionBMF: Mensagem de resposta à requisição de custódia da conta. O método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/positionBmf/account/ativo>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/positionBmf/4101>

<http://webfeeder.cedrofinances.com.br/services/negotiation/positionBmf/4101/petr3>

Resposta:

[Objeto CustodyBMFBean]

quote
position
initQuantity
profitLoss
executedBuy
executedSell
lastTrade
lastTradeDate
expired
valorOpcao
precoAjuste
qtdTomadorBTC
qtdDoadorBTC
precoMedioTomadorBTC
precoMedioDoadorBTC

positionBovespa: Mensagem de resposta à requisição de custódia da conta. O método é do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/positionBovespa/account/ativo>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/positionBovespa/4101/peetr3>

Resposta:

[Objeto CustodyBovBean]

quote
quoteType
currentQuantity
lastTrade
previusTrade
averagePrice
initQuantity
quantityD1

quantityD2
quantityD3
priceAverageInit
valueCurrent
valueInit
valueSpent
profitLoss
profitLossPercent
openBuy
openSell
executedBuy
executedSell
expired
quantityBlocked
quantityReleased
lastTradeDate
qtdTomadorBTC
qtdDoadorBTC
precoMedioTomadorBTC
precoMedioDoadorBTC
netDay
netPosition
netDaytrade
averagePricePos
openProfitLoss
closedProfitLoss

brokerServiceLogin: Autenticação secundária para possibilitar a utilização de mais de um usuário logado por sessão. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceLogin/username/password/ip/appName>

Exemplo Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceLogin/10034/123456*/127.0.0.1/hbGuide

Resposta:

[Objeto LoginOMSResponse]

authenticationReqId
isAuthenticated

username
daysUntilPasswordExpiration
text

brokerServiceLogout: Serviço de logout no OMS InternalLogout. Método do tipo GET.

Formato da Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceLogout/username/
ip

Exemplo Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceLogout/10034/127
.0.0.1

Resposta:

[Objeto LoginOMSResponse]

authenticationReqId
isAuthenticated
username
daysUntilPasswordExpiration
text

brokerServiceChangePassword : Serviço de alteração de senha no OMS. Método do tipo GET.

Formato da Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceChangePassword/
username/password/ip/newPassword

Exemplo Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceChangePassword/
10034/123456*/127.0.0.1/654321*

Resposta:

[Objeto UserRequestOMSResponse]

userRequestID
username
userStatus
userStatusText

brokerServiceForgotPassword: Serviço de esqueci a senha no OMS. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceForgotPassword/username/clientDocument>

*clientDocument = CPF

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceForgotPassword/10034/65223356897>

Resposta:

[Objeto PasswordResetOMSResponse]

requestId

text

username

clientDocument

passwordResetRejCode

brokerServiceUserInformation: Mensagem de requisição das informações de um usuário. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceUserInformation/username>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceUserInformation/10035>

Resposta:

[Objeto UserInformationOMSResponse]

userInformationRequestID

userInformationRejReason

text

username

clientID

clientName

email
clientDocument
assessor
accounts
BirthdayDate
accountsBean
address
city
stateOrProvinceOfIssue
blocked
omsMessage

brokerServiceAccountInformation: Mensagem de requisição das informações de uma conta usuário. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceUserInformation/username/account/market>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceAccountInformation/10035/10035/XBSP>

Resposta:

[Objeto AccountInformationResponseBean]

existeAccount
blocked
externalId
accountDetailList
 .brokerId
 .brokerName
 .targetAccountDesc
 .portId
 .portName
 .account

warrantyList: Mensagem de requisição para busca das garantias. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/warrantyList/account>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/warrantyList/10318>

Resposta:

[Objeto ResponseWarrantyList]

leverage - 10980

warrantyBeans

warrantyID - 10973;

typeWarranty - 10974;

descWarranty - 10979;

valueWarranty - 10975;

blockedValueWarranty - 10976;

quantity - 53;

isDepositabile - 10977;

isDeposited - 10978;

warrantyOperationUpdate: Mensagem para registrar a garantia a uma conta. Método do tipo POST.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/warrantyOperationUpdate?account=<conta>&warrantyID=<id>&descWarranty=<descricao>&quantity=<qtd>&valueWarranty=<valor>&operation=<operacao>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/warrantyOperationUpdate?account=10318&warrantyID=47833&descWarranty=garantia&quantity=10.5&valueWarranty=18.5&operation=l>

Resposta:

[Objeto ResponseWarranty]

code

message

warrantyOperation: Mensagem de requisição das informações de garantias de uma conta. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/warrantyOperation/account>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/warrantyOperation/10318>

Resposta:

[Objeto ResponseWarrantyOperation]

movementsBeans

 userListRequestID - 10072;
 status - 20003;
 text - 58;
 eventTime - 30030;

blockUser: Serviço de bloqueio/desbloqueio de usuário do OMS. Método do tipo POST.

Formato da Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/blockUser?username=<String>&blockUser*=<boolean>

Exemplo Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/blockUser?username=100100&blockUser=true

blockUser* : Bloquear = **true**.
Desbloquear = **false**.

Resposta:

[Objeto ClientResponse]

. code
. message

brokeragePDF: Serviço de nota de corretagem em arquivo de uma conta do OMS. Método do tipo GET.

Formato da Requisição:

HTTP://webfeeder.cedrofinances.com.br/services/negotiation/brokeragePDF/account/market/date(ddmmyyyy)

Exemplo Requisição:

HTTP://webfeeder.cedrofinances.com.br/services/negotiation/brokeragePDF/11785/XBSP/20052009

HTTP://webfeeder.cedrofinances.com.br/services/negotiation/brokeragePDF/4101/XBMF/11122014

Resposta:

Caso exista Nota de Corretagem para o dia irá retornar o arquivo **PDF da nota**. Caso não tenha, irá retornar:

[Objeto ResponseBrokeragePDF]

. code
. message

fixedIncomeList: Serviço para retornar os ativos de Renda Fixa. Método do tipo GET.

Formato da Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/fixedIncomeList

Exemplo Requisição:

http://webfeeder.cedrofinances.com.br/services/negotiation/fixedIncomeList

Resposta:

```
{
  "listBeans": [
    {
      "CODIGO_ATIVO": "20180716161949",
      "EMISSION": "7330523000312",
      "DESCRICAO": "teste",
      "VENCIMENTO": "2018-07-20",
      "PRAZO": "10",
      "TIPO": "", // Trata-se do subtipo do ativo
      "QUANTIDADE_DISP": "0.0",
      "PRECO_UNITARIO": "1000.24",
      "VALOR_DISP": "10.0",
      "VALOR_MIN_APLICACAO": "10.0",
      "TAXA_NEGOCIADA": "0.0",
      "TAXA_NEGOCIADA_360": "0.0",
      "INDEXADOR": "7",
      "PERC_INDEXADOR": "0.0",
      "TAXA_EMISSAO": "10.0",
      "DESCRICAO_EMISSION": "BANCO ORIGINAL S.A",
      "NOME_EMISSION": "BANCO ORIGINAL S.A",
      "CODIGO_PRODUTO": "3",
```

```
"ATIVO_DISPONIVEL": "Y",
"PRAZO_CARENCIA": "10",
"CARENCIA": "2018-07-20",
"TAXA_PRE_EMISSAO": "10.0",
"TAXA_CLIENTE": "10.0",
"TAXA_PRE_CLIENTE": "10.0",
"SE_IR_ATIVO": "N",
"SE_FGC_ATIVO": "N",
"LASTRO_ATIVO": "0",
"RISCO_ATIVO": "0",
"COMPLEXIDADE_ATIVO": "0",
"DESCRICAO_PRODUTO": "CDB PRE",
"NOME_PRODUTO": "CDB PRE",
"CODIGO_CLASSE": "2",
"DESCRICAO_CLASSE": "PRIVADO",
"DESCRICAO_INDEXADOR": "DI",
"DATA_LIMITE_RESERVA": "2018-07-20",
"SALDO_MIN_PERMANENCIA": "30",
"LIQUIDEZ_MERCADO": "Y",
"PERIODICIDADE_JUROS": "",
"PERIODICIDADE_AMORTIZACAO": "",
"PUBLICO_ALVO": "0",
"ISENTO_IOF": "N",
"ABERTO_RESGATE": "",
"URL_PDF": "",
"VERSAO_ATIVO": "",
"DATA_COTA": "",
"APLICACAO_MINIMA": "",
"ABERTO_RESGATE": "",
"VERSAO_ATIVO": ""},
"URL_PDF": ""},
"OBJ_EMISSOR": {
  "RATINGS": [
    "BBB Fitch"
  ]
}
}
```

fundsList: Serviço para retornar os ativos de Fundos. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/fundsList>

Exemplo Requisição: <http://webfeeder.cedrofinances.com.br/services/negotiation/fundsList>

Resposta:

```
{
  "listBeans":[
    {
      "CODIGO_ATIVO":"10",
      "VENCIMENTO":"2018-09-28",
      "PRAZO":"0",
      "TIPO": "", // Trata-se do subtipo do ativo
      "QUANTIDADE_DISP":"0.0",
      "PRECO_UNITARIO":"0.0",
      "VALOR_DISP":"0.0",
      "VALOR_MIN_APLICACAO":"0.0",
      "TAXA_NEGOCIADA":"0.0",
      "TAXA_NEGOCIADA_360":"0.0",
      "PERC_INDEXADOR":"0.0",
      "TAXA_EMISSAO":"0.0",
      "ATIVO_DISPONIVEL":"Y",
      "PRAZO_CARENCIA":"0",
      "CARENCIA":"2018-09-28",
      "TAXA_PRE_EMISSAO":"0.0",
      "TAXA_CLIENTE":"0.0",
      "TAXA_PRE_CLIENTE":"0.0",
      "SE_IR_ATIVO":"N",
      "SE_FGC_ATIVO":"N",
      "LASTRO_ATIVO":"0",
      "RISCO_ATIVO":"0",
      "COMPLEXIDADE_ATIVO":"0",
      "PUBLICO_ALVO":"0",
      "MOVIMENT_MINIMA":"0.0",
      "SALDO_MINIMO":"0.0",
      "TAXA_ADM":"0.0",
      "TAXA_PERFORMANCE":"0.0",
      "TAXA_SAIDA":"0.0"},
      "GESTOR ":"",
      "ADMINISTRADOR":"",
      "TAXA_RESGATE_ANTECIPADO":"",
      "TIPO_COTA":"",
      "COTA_APLICACAO":"",
      "COTA_RESGATE":"",
      "HORARIO_APLICACAO":"",
      "HORARIO_RESGATE":""}
```

```
"ABERTO_RESGATE":"","  
"URL_PDF":"","  
"VERSAO_ATIVO":"","  
"VALOR_INICIAL_APLICACAO":"","  
"BENCHMARK":"","  
"SALDO_MINIMO_PERMANENCIA":"","  
}}
```

brokerageDatesRequestBovespa: Serviço de requisição de datas de notas de corretagem. Utilizada para requisitar em quais datas houveram corretagem. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerageDatesRequestBovespa/account/dataIni/dataFim>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerageDatesRequestBovespa/10013/15022018/20022018>

Resposta:

brokerageDatesRequestBMF: Serviço de requisição de datas de notas de corretagem. Utilizada para requisitar em quais datas houveram corretagem. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerageDatesRequestBMF/account/dataIni/dataFim>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerageDatesRequestBovespa/10013/15022018/20022018>

Resposta:

limits: Serviço de requisição dos limites de um usuário ou conta. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/limits/username/account/marketId>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/limits/10013/10013/BOVESPA>

Resposta:

createUpdateLimitRequest: Serviço de atualização dos limites de um usuário ou conta. Método do tipo POST.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/createUpdateLimitRequest?account=<conta>&limitsupdateid=<idLimite>&market=<mercado>&securitytype=<tipo>&operation=<operação I-Inserção D-Deleção>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/createUpdateLimitRequest?account=10013&limitsupdateid=182930&market=bovespa&securitytype=OPT&operation=I>

Resposta:

statementBovespa: Serviço de requisição de extrato financeiro de uma conta. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/statementBovespa/account/statementType<FINAN,BOLSA>/dateStart/dateEnd>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/statementBovespa/10013/FINAN/18022018/25022018>

Resposta:

statementBmf: Serviço de requisição de extrato financeiro de uma conta. Método do tipo GET.

Formato da Requisição:

<http://localhost:8080/services/negotiation/statementBmf/account/statementType<FINAN,BO LSA>/dateStart/dateEnd>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/statementBmf/10013/FINAN/18022018/25022018>

Resposta:

brokerageBovespa: Serviço de requisição à nota de corretagem de uma conta. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerageBovespa/account/data>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerageBovespa/4101/30122014>

Resposta:

brokerageBmf: Serviço de requisição à nota de corretagem de uma conta. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerageBmf/account/data>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerageBmf/4101/30122014>

Resposta:

brokerServiceAccountList: Serviço de requisição das informações de um usuário. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceAccountList/account>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/brokerServiceAccountList/4101>

Resposta:

```
[
  {
    "account": "4101",
    "market": "XBSP"
  }
]
```

leverageQuote: Serviço de consulta de listas de alavancagem. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/leverageQuote/account/market/symbol>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/leverageQuote/10000/XBSP/PETR4>

fixedIncomeAccountInformation: Serviço de consulta de informações de custódia de renda fixa. Método do tipo GET.

Formato da Requisição:

[http://webfeeder.cedrofinances.com.br/services/negotiation/fixedIncomeAccountInformation/<usuario negociação>/<senha negociação md5>/<account>](http://webfeeder.cedrofinances.com.br/services/negotiation/fixedIncomeAccountInformation/<usuario%20negocia%C3%A7%C3%A3o>/<senha%20negocia%C3%A7%C3%A3o%20md5>/<account>)

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/fixedIncomeAccountInformation/MARCO/e10adc3949ba59abbe56e057f20f883e/10000>

Resposta:

```
{
  "listBeans": [
    {
      "cliente": "",
      "dtMovimento": "",
      "dtAplicacao": "",
      "dtVencimento": "",
      "nomeFundo": "",
      "rentabilidade": "",
      "qtdeTotal": "",
      "qtdeVinculado": "",
      "qtdeDisponivel": "",
      "vlAplicado": "",
      "vlBruto": "",
      "vlIR": "",
      "vlIOF": "",
      "vlLiquido": ""
    }
  ]
}
```

fundsAccountInformation: Serviço de consulta de informações de custódia de fundos.
Método do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/fundsAccountInformation/<usuario negociação>/<senha negociação md5>/<account>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/fixedIncomeAccountInformation/MARCO/e10adc3949ba59abbe56e057f20f883e/10000`

Resposta:

```
{ "listBeans": [ { "cliente": "", "dtMovimento": "", "dtAplicacao": "", "nomeFundo": "", "valorCota": "", "qtdeTotalCota": "", "valorAplic": "", "valorBruto": "", "ir": "", "iof": "", "valorLiq": "" } ] }
```

clubsAccountInformation: Serviço de consulta de informações de custódia de clubes.
Método do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/clubsAccountInformation/<usuario negociação>/<senha negociação md5>/<account>`

Exemplo Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/clubsAccountInformation/MARCO/e10adc3949ba59abbe56e057f20f883e/10000`

Resposta:

```
{ "listBeans": [ { "cliente": "", "dtMovimento": "", "dtAplicacao": "", "idClube": "", "nomeClube": "", "qtdeTotal": "", "valor": "" } ] }
```

treasuryAccountInformation: Serviço de consulta de informações de custódia de tesouro direto. Método do tipo GET.

Formato da Requisição:

`http://webfeeder.cedrofinances.com.br/services/negotiation/treasuryAccountInformation/<usuario negociação>/<senha negociação md5>/<account>`

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/treasuryAccountInformation/MARCO/e10adc3949ba59abbe56e057f20f883e/10000>

Resposta:

```
{"listBeans":[{"cliente":"","papel":"","dtVencimento":"","qtd":"","valor":""}]}
```

rescueDepositInformation: Serviço de informações de resgate depósito. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/rescueDepositInformation/<username>/<data inicial>/<data final>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/rescueDepositInformation/10013/25032018/30032018>

Resposta:

```
{"listBeans":[{"operationDate":"20180704","bank":"SANTANDER","bankAccount":"222222","bankAgency":"1111","value":"2000","valueDescription":"DOIS MIL REAIS","favored":"EU","rescueDepositType":"R","isVerified":"N","obs":"TESTE","financialAccount":"10006","advisorID":"10000","advisorName":"SIMULADOR","operationTypeRescue":"PARCIAL","status":"","userName":"","clientName":""}]}
```

bankAccountInformation: Serviço de informações bancárias da conta. Método do tipo GET.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/bankAccountInformation/<financialAccount>/<username>>

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/bankAccountInformation/10006/10006>

Resposta:

```
{"listBeans":[{"bank":"409","bankAccount":"820016-3 ","bankAgency":"0053"}]}
```

updateRescueDeposit: Serviço de atualização de resgate deposito. Método do tipo POST.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/updateRescueDeposit>

* Esse serviço é permitido passar um json no Body (JSON application/json)

```
{  
  "userName": "",  
  "operationDate": "",  
  "bank": "",  
  "operation": "",  
  "bankAccount": "",  
  "bankAgency": "",  
  "value": "",  
  "valueDescription": "",  
  "favored": "",  
  "rescueDepositType": "",  
  "isVerified": "",  
  "obs": "",  
  "financialAccount": "",  
  "advisorId": "",  
  "advisorName": "",  
  "operationTypeRescue": ""  
}
```

Exemplo Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/updateRescueDeposit>

```
{  
  "userName": "10006",  
  "operationDate": "20180705",  
  "bank": "SANTANDER",  
  "operation": "update",  
  "bankAccount": "222222",  
  "bankAgency": "1111",  
  "value": "2000",  
  "valueDescription": "DOIS MIL REAIS",  
}
```



```
"favored": "VOCE",
"rescueDepositType": "r",
"isVerified": "n",
"obs": "TESTE update",
"financialAccount": "10006",
"advisorId": "10",
"advisorName": "SIMULADOR",
"operationTypeRescue": "PARCIAL"
}
```

Resposta:

```
{" Update rescue enviada com sucesso. "}
```

quoteUpdate: Serviço de atualização/criação do ativo. Método do tipo POST.

Formato da Requisição:

<http://webfeeder.cedrofinances.com.br/services/negotiation/quoteUpdate>

* Esse serviço é permitido passar um json no Body (JSON application/json)

```
{
  "marketID": "",
  "securityType": "",
  "securitySubType": "",
  "maturityDate": "",
  "securityIDSource": "",
  "securityExchange": "",
  "isinPaper": "",
  "stockOptionsOwner": "",
  "eroperation": "",
  "lastUpdate": "",
  "quotationForm": "",
  "securityDesc": "",
  "financialAccount": "",
  "issuing": "",
  "applicationDays": "",
  "rewardDescription": "",
  "availableQuantity": "",
  "unitPrice": "",
  "availablePrice": "",
  "minimumApplicationPrice": "",
  "tradedRate": "",
  "tradedRateT": ""
}
```

```
"issuanceDate": "",
"liquidityDate": "",
"securityIndex": "",
"percentIndex": "",
"issuanceRate": "",
"productCode": "",
"securityAvailable": "",
"gracePeriod": "",
"issuancePreRate": "",
"clientRate": "",
"clientPreRate": "",
"securityIR": "",
"securityCoverage": "",
"securityRisk": "",
"securityComplexity": "",
"securityFGC": "",
"balanceMinPermanence": "",
"symbolLiquidity": "",
"marketSegmentation": "",
"interestPeriodicity": "",
"amortizationPeriodicity": "",
"exemptionIof": "",
"taxAdm": "",
"taxPerformance": "",
"taxExit": "",
"rescueLiquidity": "",
"applicationLiquidity": "",
"usefulDay": "",

"administrator": "",
"earlyRescueRate": "",
"quotaType": "",
"quotaApplication": "",
"quotaRescue": "",
"applicationHorary": "",
"rescueHorary": "",
"isOpenToRescue": "",
"urlLink": "",
"quoteVersion": "",
"quotaDate": "",
"quotaRescue": "",
```

```
}
```

Websocket

I-Autenticação

O WebFeeder exige que o consumidor de dados possua uma sessão válida para efetuar o consumo de dados do mesmo, segue abaixo um exemplo na linguagem JavaScript de um evento de login websocket no WebFeeder:

```
<script>

var wsUri = "ws://201.77.219.223/ws";

websocket = new WebSocket(wsUri);

websocket.onopen = function(evt) {

    console.log("Connected to Endpoint!");

};

websocket.onmessage = function(evt) {

    console.log("Message Received: " + evt.data);

};

websocket.onerror = function(evt) {

    console.log("ERROR: " + evt.data);

};

function doSend(message) {

    console.log("Message Sent: " + message);

    websocket.send(message);

}

doSend('{

    "module": "login",

    "service": "authentication",

    "parameters": {"login": "teste", "password": "123456"}

}');

</script>
```

Na chamada do método doSend irá passar um objeto **JSON** em que especifica qual é o modulo, o serviço e os parâmetros para login(login e password) como mostra o código acima. Caso seja autenticado irá retornar no método websocket.onmessage a mensagem como abaixo:

```
{"success":true,"token":"f0d66fcb-7055-4f86-b6a9-25ccae3f9a68"}
```

Caso deseje que seja registrado o **FEES** das requisições de quote, para relatório, é necessário que envie no login a propriedade **account**, assim quando for pedida a difusão de algum ativo para o serviço, será gravado no banco a conta e o mercado para qual este ativo pertence. Segue exemplo dos parameters:

```
"parameters": {"login": "teste", "password": "123456", "account": "10000"}
```

Para efetuar o consumo de um serviço disponibilizado pelo WebFeeder via websocket é necessário enviar uma requisição utilizando o valor retornado no atributo **token**.

II-Módulo Quotes

Quote: O serviço retorna os dados referentes ao ativo solicitado, para realizar a chamada, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso quote;

ParameterGet: Qual ativo está obtendo as informações;

Parameters.subsribetype: Qual tipo de ação (0-parar cotação, 1-iniciar cotação)

Parameters.filter: Os filtros da cotação, quais os índices que serão retornados, como na tabela de índices de resposta do difusor abaixo.

Parameters.delay: Informar de quanto e quanto tempo poderá receber os dados de cotação. Valor em milissegundos.

Segue o exemplo de envio:

```
doSend('{  
    "token": "f0d66fcb-7055-4f86-b6a9-25ccae3f9a68",  
    "module": "quotes",
```

```
"service": "quote",

"parameterGet": "petr4",

"parameters": {"subsribetype": "1", "filter": "2,3,4", "delay": "400"}

});
```

Ir  receber os dados do ativo informado:

Type: Indica o tipo da chamada;

Parameter: De qual ativo   a resposta;

Values: Os valores retornados pelo difusor.

Segue abaixo o exemplo da resposta:

```
{"values":{"2":"13,32","3":"13,31","4":"13,33"},"type":"QuoteType","parameter":"petr4"}
```

Corpo da mensagem do **values**: Este   composto de um ou mais pares de informa  o que   enviada da seguinte forma: “< ndice>:<valor>”

Tabela de  ndices de resposta do difusor:

�ndice	Significado	Tipo
0	Hor�rio da �ltima modifica��o	Hor�rio (HHMMSS)
1	Data da �ltima modifica��o	Data (YYYYMMDD)
2	Pre�o do �ltimo neg�cio	Float
3	Melhor oferta de compra	Float
4	Melhor oferta de venda	Float
5	Hor�rio do �ltimo neg�cio	Hor�rio (HHMMSS)
6	Quantidade do neg�cio atual	Inteiro (quantidade)
7	Quantidade do �ltimo neg�cio	Inteiro (quantidade)
8	Quantidade de neg�cios realizados	Inteiro
9	Volume acumulado dos neg�cios	Inteiro (quantidade)
10	Volume financeiro dos neg�cios	Float
11	Maior pre�o do dia	Float
12	Menor pre�o do dia	Float
13	Pre�o de fechamento do dia anterior	Float
14	Pre�o de abertura	Float
15	Hor�rio da melhor oferta de compra	Hor�rio (HHMMSS)
16	Hor�rio da melhor oferta de venda	Hor�rio (HHMMSS)
17	Volume acumulado das melhores ofertas de compra	Float
18	Volume acumulado das melhores ofertas de venda	Float
19	Volume da melhor oferta de compra	Inteiro (quantidade)
20	Volume da melhor oferta de venda	Inteiro (quantidade)
21	Varia��o	Float
36	Pre�o de fechamento da �ltima semana	Float
37	Pre�o de fechamento do �ltimo m�s	Float
38	Pre�o de fechamento do �ltimo ano	Float
39	Pre�o de abertura do dia anterior	Float

40	Maior preço do dia anterior	Float
41	Menor preço do dia anterior	Float
42	Média	Float
43	VHDaily	Float
44	Código do Mercado:	Inteiro
	1 - Bovespa	
	2 - Dow Jones	
	3 - BM&F	
	4 - Índices	
	5 - Money	
	6 - Soma	
	7 - Forex	
	8 - Indicators	
	9 - Others	
	10 - Nyse	
	11 - Bats	
	12 - Nasdaq	
	14 - BVL	
	15 - SPIndexes	
	16 - Liffe	
	17 - Euronext Indices	
	18 - CME	
	19 - CME Mini	
45	Código do tipo do ativo:	Inteiro
	1 - Ativo à vista	
	2 - Opção	
	3 - Índice	
	4 - Commodity	
	5 - Moeda	
	6 - Termo	
	7 - Futuro	
	8 - Leilão	
	9 - Bônus	
	10 - Facionário	
	11 - Exercício de opção	
	12 - Indicador	
	13 - ETF	
	15 - Volume	
	16 - Opção sobre à vista	
	17 - Opção sobre futuro	
	18 - Ativo de teste	
46	Lote padrão	Inteiro (quantidade)
47	Descrição do ativo	String(10) Exemplos: DOL, INDICE BOVESPA.
48	Nome de classificação	String
49	Forma de cotação	Inteiro
50	Intraday Date (FORCES)	Data e Horário (YYYYMMDDHHMMSS)
51	LastTrade Date (FORCES)	Data e Horário (YYYYMMDDHHMMSS)
52	Descrição abreviada do ativo	String
53	Identificador do negocio cancelado	String
54	Data do último negócio	Data (YYYYMMDD)
56	Sentido das ofertas não atendidas ao preço de abertura	Caracter: A – Compra

		V – Venda
		0 – Não informado
57	Quantidade não atendida ao preço de abertura	Inteiro
58	Horário programado para abertura do papel	Horário (HHMMSS)
59	Horário reprogramado para abertura do papel	Horário (HHMMSS)
60	Código da corretora que fez a melhor oferta de compra	Inteiro
61	Código da corretora que fez a melhor oferta de venda	Inteiro
62	Código da corretora que realizou a última compra	Inteiro
63	Código da corretora que realizou a última venda	Inteiro
64	Data do vencimento (Mercado de opções)	Data (YYYYMMDD)
65	Expirado	Inteiro
66	Número total de papéis	String
67	Status do instrumento	Inteiro
72	Tipo da opção (A = Americana, E = Européia, 0 = não existe)	char
82	Preço teórico de abertura	Float
83	Quantidade teórica	Inteiro
86	Diff (Preço Atual - Previous)	Float
87	Data do Previous	Data (YYYYMMDD)
90	Intervalo de Margem (mercado BTC)	Float
94	Volume médio nos últimos 20 dias	Float
10094	Campo calculado da variação de acordo com o volume médio dos ultimos 20 dias	Float
95	Market Capitalization	Float
96	Tipo de Mercado (RT = RealTime, D = Delay, EOD = End of Day)	String
97	Valor do fechamento em uma semana	Float
10097	Campo calculado da variação de acordo com o valor de fechamento de 7 dias	Float
98	Valor do fechamento em um mês	Float
10098	Campo calculado da variacao de acordo com o valor de fechamento de 1 mês	Float
99	Valor do fechamento em um ano	Float
10099	Campo calculado da variacao de acordo com o valor de fechamento de 1 ano	Float
110	TickSize	Inteiro
134	Variação do volume da hora com base na média do volume do horário nos últimos 20 dias	Float
135	Variação do volume até a hora com base na média do volume até o horário nos últimos 20 dias	Float

Valores referentes a ativos do mercado BOVESPA:

Índice	Significado	Tipo
84	Status do ativo (0 = Normal, 1 = Congelado, 2 = Suspenso, 3 = Leilão, 4 = Inibido)	Inteiro
85	Preço de Exercício	Float

88	Fase do grupo do ativo (P = Pré abertura, A = Abertura (sessão normal), PN = Pré fechamento, N = fechamento, E = Pré abertura do after, R = abertura After, NE = Fechamento do after, F = final)	String
89	Média do dia anterior	Float

Valores referentes a ativos do mercado BMF:

Índice	Significado	Tipo
100	Quantidade de contratos abertos	Inteiro
101	Número dias úteis até o vencimento	Inteiro
102	Número dias para o vencimento	Inteiro
103	Ajuste do dia	Float
104	Ajuste do dia anterior	Float
105	SecurityId (BMF FIX)	String
106	TickDirection(BMF FIX)	String(2) Valores possíveis: +, 0+, -, 0-.
107	TunnelUpperLimit	Float
108	TunnelLowerLimit	Float
109	TradingPhase(BMF FIX)	String(2)
111	Volume mínimo de negociação do instrumento	Inteiro (quantidade)
112	Intervalo mínimo para incrementos de preço	Float
113	Quantidade mínima para o instrumento em uma oferta	Inteiro (quantidade)
114	Quantidade máxima para o instrumento em uma oferta	Inteiro (quantidade)
115	Número único de identificação do instrumento	Inteiro
116	Moeda utilizada no preço	String Exemplos: BRL, EUR, USD.
117	SecurityType	String(32) Valores possíveis: FUT, OPT, SPOT, SOPT, FOPT, DTERM.
118	Código de negociação do instrumento	String(32) Exemplos: DOL, INDICE BOVESPA.
119	Produto associado ao instrumento	Inteiro
120	Mês e ano de vencimento	Data (YYYYMM)
121	Preço de exercício da opção	Float
122	Moeda do preço de exercício da opção	String Exemplos: BRL, EUR, USD.
123	Multiplicador do contrato	Float
124	Código que representa o tipo de preço do instrumento	Inteiro
125	Horário em que um instrumento não é mais passível de negociação	Data e Horário (YYYYMMDDHHMMSS)
126	Indica o grupo ao qual o ativo pertence	String(15)
127	Ajuste atual em taxa	Float
128	Ajuste anterior em taxa	Float
129	Data do Ajuste atual em taxa	Data(YYYYMMDD)
130	Número de saques até data de vencimento	Inteiro

Book: O serviço retorna os dados referentes ao book do ativo solicitado, para realizar a chamada, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso book;

ParameterGet: Qual ativo está obtendo as informações;

Parameters.subsbribetype: Qual tipo de ação (0-parar book, 1-iniciar book)

Parameters.filter: O filtro do book, qual é a quantidade de linhas do book deseja que seja retornados.

Parameters.delay: Informar de quanto e quanto tempo poderá receber os dados do book. Valor em milissegundos.

Segue o exemplo de envio:

```
doSend('{
    "token": "f0d66fcb-7055-4f86-b6a9-25ccae3f9a68",
    "module": "quotes",
    "service": "book",
    "parameterGet": "petr4",
    "parameters": {"subsbribetype": "1", "filter": "10", "delay": "5000"}
});
```

Irá receber os dados do ativo informado:

Type: Indica o tipo da chamada;

Parameter: De qual ativo é a resposta;

Book: O book retornado pelo difusor.

Segue abaixo o exemplo da resposta:

```
{
  "book": {
    "S": "petr4",
    "B": [
      {
        "Q": 2900, "P": 12.83, "C": 3, "FQ": "2.9K", "T": "L"
      },
      {
        "Q": 400, "P": 12.83, "C": 45, "FQ": "400", "T": "L"
      },
      {
        "Q": 800, "P": 12.83, "C": 40, "FQ": "800", "T": "L"
      },
      {
        "Q": 100, "P": 12.83, "C": 238, "FQ": "100", "T": "L"
      },
      {
        "Q": 200, "P": 12.83, "C": 40, "FQ": "200", "T": "L"
      },
      {
        "Q": 5000, "P": 12.83, "C": 8, "FQ": "5K", "T": "L"
      },
      {
        "Q": 50, "P": 12.83, "C": 8, "FQ": "500", "T": "L"
      },
      {
        "Q": 500, "P": 12.83, "C": 8, "FQ": "500", "T": "L"
      },
      {
        "Q": 400, "P": 12.83, "C": 8, "FQ": "400", "T": "L"
      },
      {
        "Q": 21600, "P": 12.84, "C": 3, "FQ": "21.6K", "T": "L"
      }
    ],
    "A": [
      {
        "Q": 100, "P": 12.82, "C": 16, "FQ": "100", "T": "L"
      },
      {
        "Q": 300, "P": 12.82, "C": 8, "FQ": "300", "T": "L"
      },
      {
        "Q": 5000, "P": 12.82, "C": 8, "FQ": "5K"
      }
    ]
  }
}
```

```
"T":"L"},{"Q":400,"P":12.82,"C":8,"FQ":400,"T":"L"},{"Q":1000,"P":12.82,"C":21,"FQ":1K,"T":"L"},{"Q":500,"P":12.82,"C":8,"FQ":500,"T":"L"},{"Q":500,"P":12.82,"C":8,"FQ":500,"T":"L"},{"Q":4800,"P":12.82,"C":386,"FQ":4.8K,"T":"L"},{"Q":500,"P":12.82,"C":120,"FQ":500,"T":"L"},{"Q":400,"P":12.81,"C":8,"FQ":400,"T":"L"}], "type": "BookSnapshotType", "parameter": "petr4"}
```

O **book** é um objeto composto por **"S" - Ativo, "A" - Compra e "B" - Venda**: Estessão compostos de um ou mais objetos composto por **"Q" - Quantidade, "P" - Preço, "C" - Corretora, "FQ" - Qtde formatada, "T" Tipo** (O tipo **L** indica oferta Limitada, **O** oferta ao preço de Abertura, **X** oferta ao melhor preço e **M** oferta ao preço de Mercado).

AggregatedBook: O serviço retorna os dados referentes ao book agregado do ativo solicitado, para realizar a chamada, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso book;

ParameterGet: Qual ativo está obtendo as informações;

Parameters.subsbribeType: Qual tipo de ação (0-parar book, 1-iniciar book)

Parameters.delay: Informar de quanto e quanto tempo poderá receber os dados do book. Valor em milissegundos.

Segue o exemplo de envio:

```
doSend('{
    "token": "f0d66fcb-7055-4f86-b6a9-25ccae3f9a68",
    "module": "quotes",
    "service": "aggregatedBook",
    "parameterGet": "petr4",
    "parameters": {"subsbribeType": "1", "delay": "5000"}
});
```

Irá receber os dados do ativo informado:

Type: Indica o tipo da chamada;

Parameter: De qual ativo é a resposta;

Book: O book agregado retornado pelo difusor, será retornado por default sempre 5 linhas de compra e venda.

Segue abaixo o exemplo da resposta:

```
{ "book": { "S": "petr4", "B": [ { "Q": 27700, "P": 12.83, "FQ": "27.7K" }, { "Q": 69000, "P": 12.84, "FQ": "69K" }, { "Q": 57200, "P": 12.85, "FQ": "57.2K" }, { "Q": 91500, "P": 12.86, "FQ": "91.5K" }, { "Q": 70300, "P": 12.87, "FQ": "70.3K" } ], "A": [ { "Q": 20200, "P": 12.82, "FQ": "20.2K" }, { "Q": 50700, "P": 12.81, "FQ": "50.7K" }, { "Q": 82400, "P": 12.8, "FQ": "82.4K" }, { "Q": 90500, "P": 12.79, "FQ": "90.5K" }, { "Q": 100600, "P": 12.78, "FQ": "100.6K" } ] }, "type": "Aggregated BookType", "parameter": "petr4" }
```

O **book** é um objeto composto por **"S" - Ativo**, **"A" - Compra** e **"B" - Venda**: Estes são compostos de um ou mais objetos composto por **"Q" - Quantidade**, **"P" - Preço**, **"FQ" - Qtde formatada**.

AuctionList: O serviço retorna os dados referentes aos ativos em leilão do mercado solicitado, para realizar a chamada, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso book;

ParameterGet: Qual mercado está obtendo as informações, Bovespa ou bmf;

Parameters.subsbribetype: Qual tipo de ação (0-parar, 1-iniciar)

Parameters.delay: Informar de quanto e quanto tempo poderá receber os ativos em leilão. Valor em milissegundos.

Segue o exemplo de envio:

```
doSend({
  "token": "f0d66fcb-7055-4f86-b6a9-25ccae3f9a68",
  "module": "quotes",
  "service": "auctionList",
  "parameterGet": "bovespa",
  "parameters": { "subsbribetype": "1", "delay": "5000" }
});
```

Irá receber os dados do ativo informado:

Type: Indica o tipo da chamada;

Parameter: De qual mercado é a resposta;

AuctionList: A lista de ativos em leilão retornado pelo difusor, será retornado sempre todos os ativos em leilão do mercado.

Segue abaixo o exemplo da resposta:

```
{ "auctionList": { "M": "bovespa", "L": [ { "Q": "ENMT3", "ST": "17:00:00", "RST": "00:00:00", "P": 15.8, "QT": 400 }, { "Q": "HAGA4", "ST": "17:00:00", "RST": "00:00:00", "P": 2.75, "QT": 1400 }, { "Q": "VULC3F", "ST": "17:00:00", "RST": "00:00:00", "P": 9.0, "QT": 84 }, { "Q": "VALET6", "ST": "16:55:19", "RST": "00:00:00", "P": 0.97, "QT": 26900 }, { "Q": "VALEH28", "ST": "16:55:30", "RST": "00:00:00", "P": 2.43, "QT": 200 } ], "type": "AuctionListType", "parameter": "bovespa" }
```

O **auctionList** é um objeto composto por **"M" - Mercado**, **"L" - Lista de ativos em leilão**: Esta é composta de um ou mais objetos compostos por **"Q" - Ativo**, **"ST" - Horário programado**, **"RST" - Horário reprogramado**.

Market Ranking: Utilizado para solicitar o recebimento das quantidades das negociações das corretoras em um determinado mercado. Para realizar a chamada, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso book;

ParameterGet: Qual mercado está obtendo as informações, Bovespa ou bmf;

Parameters.subsbribeType: Qual tipo de ação (0-parar, 1-iniciar)

Segue o exemplo de envio:

```
doSend({
  "token": "f0d66fcb-7055-4f86-b6a9-25ccae3f9a68",
  "module": "quotes",
  "service": "marketRanking",
  "parameterGet": "bovespa",
  "parameters": { "subsbribeType": "1" }
});
```

Irá receber os dados do ativo informado:

type: Indica o tipo da chamada;

playerCode: Código da corretora;

volumeFinancier: Volume financeiro.

volumeAmount: Volume de negociação.

quantityTrade: Quantidade de negociação.

Segue abaixo o exemplo da resposta:

```
{ "marketRanking": { "M": "bovespa", "L": [ { "PC": 0, "VF": 0.0, "VA": 0, "QT": 0, "T": "B" }, { "PC": 0, "VF": 0.0, "VA": 0, "QT": 0, "T": "S" }, { "PC": 1, "VF": 0.0, "VA": 0, "QT": 0, "T": "B" }, { "PC": 1, "VF": 0.0, "VA": 0, "QT": 0, "T": "S" }, { "PC": 10, "VF": 2.2927296E7, "VA": 3663560, "QT": 2639, "T": "B" }, { "PC": 10, "VF": 2.0814614E7, "VA": 3225061, "QT": 2326, "T": "S" }, { "PC": 100, "VF": 0.0, "VA": 0, "QT": 0, "T": "B" }, { "PC": 100, "VF": 0.0, "VA": 0, "QT": 0, "T": "S" }, { "PC": 102, "VF": 0.0, "VA": 0, "QT": 0, "T": "B" } ] }
```

statusDifusion: Mensagem de resposta à requisição de status do difusor. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso statusDifusion;

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (true, false);

Segue o exemplo de envio:

```
doSend({  
  "token": "bb70c590-c646-4b42-b705-814fe44db3ff",  
  "module": "quotes",  
  "service": "statusDifusion",  
  "parameters": { "dispatch": "true" }  
});
```

Resposta:

[Objeto StatusDifusionModel]

statusDifusion -> Valores possíveis (initialized/serving/stop/open/reload/close)

highList: Mensagem de resposta à requisição de maiores altas. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação;

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso highList;

Parameters. subsbribeType: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend('{  
  "token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",  
  "module": "quotes",  
  "service": "highList",  
  "parameterGet": "bovespa",  
  "parameters": {"subsbribeType": "1"}  
});
```

Irá receber os dados do ativo informado:

quote: Símbolo do ativo;

price: Preço do ativo;

value: Valor do item da lista, para as listas de altas e baixas é o valor da variação do preço e para as listas de volume, é o volume negociado no dia;

date: Horário em que a linha foi atualizada. Formato HHMMSS

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto StreamingResumeMarketList]

```
{"streamingResumeMarketList":{"M":"bovespa","L":[{"Q":"TRPN1","P":0.48,"V":128.57,"D":"08-06-2018 13:32:15"}, {"Q":"BAUH4","P":12.38,"V":12.55,"D":"08-06-2018 14:13:26"}, {"Q":"IGBR3","P":0.9,"V":11.11,"D":"08-06-2018 14:36:18"}, {"Q":"FNAM11","P":0.13,"V":8.33,"D":"08-06-2018 14:25:07"}, {"Q":"XPLG11","P":99.47,"V":7.73,"D":"08-06-2018 14:24:06"}, {"Q":"TEKA4","P":7.0,"V":7.69,"D":"08-06-2018 13:52:35"}, {"Q":"MNP3","P":2.65,"V":7.29,"D":"08-06-2018 11:19:30"}, {"Q":"ADHM3","P":1.7,"V":6.25,"D":"08-06-2018 11:19:12"}, {"Q":"IDVL4","P":0.52,"V":6.12,"D":"08-06-2018 14:38:21"}, {"Q":"BTOW3","P":24.39,"V":6.04,"D":"08-06-2018 14:47:06"}]}}
```

fallList: Mensagem de resposta à requisição de maiores baixas. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação;

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso fallList;

Parameters. subsbribe: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend({
  "token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",
  "module": "quotes",
  "service": "fallList",
  "parameterGet": "bovespa",
  "parameters": {"subbribe": "1"}
```

```
});
```

Irá receber os dados do ativo informado:

quote: Símbolo do ativo;

price: Preço do ativo;

value: Valor do item da lista, para as listas de altas e baixas é o valor da variação do preço e para as listas de volume, é o volume negociado no dia;

date: Horário em que a linha foi atualizada. Formato HHMMSS

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto StreamingResumeMarketList]

```
{"streamingResumeMarketList":{"M":"bovespa","L":[{"Q":"RBCB11","P":350.0,"V":-14.63,"D":"08-06-2018 14:54:08"},{"Q":"SUZB3","P":44.88,"V":-9.24,"D":"08-06-2018 15:01:57"},{"Q":"CTSA3","P":3.7,"V":-8.64,"D":"08-06-2018 14:58:44"},{"Q":"TCNO4","P":1.46,"V":-7.59,"D":"08-06-2018 13:59:42"},{"Q":"VIVR3","P":0.25,"V":-7.41,"D":"08-06-2018 14:59:45"},{"Q":"FIVN11","P":4.0,"V":-6.98,"D":"08-06-2018 12:46:59"},{"Q":"SLED4","P":3.33,"V":-6.46,"D":"08-06-2018 15:00:46"},{"Q":"ELET6","P":14.44,"V":-6.11,"D":"08-06-2018 15:01:56"},{"Q":"BRAP4","P":30.62,"V":-6.04,"D":"08-06-2018 15:01:50"},{"Q":"AAPL34","P":71.9,"V":-5.66,"D":"08-06-2018 14:57:43"}]},{"type":"ResumeMarketType","parameter":"bovespa"}
```

highVolumeList: Mensagem de resposta à requisição de maiores volumes altos. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação;

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso highVolumeList;

Parameters. subsbribeType: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend('{
```



```
"token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",  
"module": "quotes",  
"service": "highVolumeList",  
"parameterGet": "bovespa",  
"parameters": {"subsribetype": "1"}  
});
```

Irá receber os dados do ativo informado:

quote: Símbolo do ativo;

price: Preço do ativo;

value: Valor do item da lista, para as listas de altas e baixas é o valor da variação do preço e para as listas de volume, é o volume negociado no dia;

date: Horário em que a linha foi atualizada. Formato HHMMSS

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto StreamingResumeMarketList]

```
{"streamingResumeMarketList":{"M":"bovespa","L":[{"Q":"PETR4","P":15.56,"V":1.062644315E9,"D":"Fri Jun 08 15:26:01 BRT 2018"}, {"Q":"VALE3","P":51.46,"V":9.7471115E8,"D":"Fri Jun 08 15:26:09 BRT 2018"}, {"Q":"ITUB4","P":41.38,"V":5.68506243E8,"D":"Fri Jun 08 15:26:09 BRT 2018"}, {"Q":"BBAS3","P":27.3,"V":4.89043467E8,"D":"Fri Jun 08 15:26:00 BRT 2018"}, {"Q":"BBDC4","P":27.91,"V":4.25278358E8,"D":"Fri Jun 08 15:26:09 BRT 2018"}, {"Q":"SUZB3","P":45.0,"V":3.55615248E8,"D":"Fri Jun 08 15:26:09 BRT 2018"}, {"Q":"B3SA3","P":20.8,"V":3.53649745E8,"D":"Fri Jun 08 15:26:01 BRT 2018"}, {"Q":"JBSS3","P":8.85,"V":2.81930186E8,"D":"Fri Jun 08 15:26:09 BRT 2018"}, {"Q":"MGLU3","P":107.13,"V":2.82823325E8,"D":"Fri Jun 08 15:26:00 BRT 2018"}, {"Q":"ITSA4","P":9.53,"V":2.52135175E8,"D":"Fri Jun 08 15:14:13 BRT 2018"}, {"Q":"PETR3","P":18.36,"V":2.70939629E8,"D":"Fri Jun 08 15:26:09 BRT 2018"}]},{"type":"ResumeMarketType","parameter":"bovespa"}
```

fallVolumeList: Mensagem de resposta à requisição de maiores volumes baixas. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação;

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso fallVolumeList;

Parameters. subsbribeType: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend('{  
  "token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",  
  "module": "quotes",  
  "service": "fallVolumeList",  
  "parameterGet": "bovespa",  
  "parameters": {"subsbribeType": "1"}  
});
```

Irá receber os dados do ativo informado:

quote: Símbolo do ativo;

price: Preço do ativo;

value: Valor do item da lista, para as listas de altas e baixas é o valor da variação do preço e para as listas de volume, é o volume negociado no dia;

date: Horário em que a linha foi atualizada. Formato HHMMSS

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto StreamingResumeMarketList]

```
{"streamingResumeMarketList":{"M":"bovespa","L":[{"Q":"FIVN11","P":4.0,"V":48.0,"D":"08-06-2018  
12:46:59"}, {"Q":"CRDE3","P":1.0,"V":100.0,"D":"08-06-2018  
14:53:38"}, {"Q":"ADHM3","P":1.7,"V":170.0,"D":"08-06-2018  
11:19:12"}, {"Q":"FJTA3","P":1.9,"V":190.0,"D":"08-06-2018"}]}
```

```
11:54:01"},{"Q":"SCPF11","P":12.47,"V":212.25,"D":"08-06-2018
13:59:42"},{"Q":"IDVL3","P":1.2,"V":360.0,"D":"08-06-2018
13:18:10"},{"Q":"REDE3","P":5.32,"V":532.0,"D":"08-06-2018
14:00:44"},{"Q":"FNOR11","P":0.23,"V":717.37,"D":"08-06-2018
13:43:56"},{"Q":"NVHO11","P":8.8,"V":776.4,"D":"08-06-2018
13:35:18"},{"Q":"CTNM3","P":8.0,"V":800.0,"D":"08-06-2018
11:32:06"}]],{"type":"ResumeMarketType","parameter":"bovespa"}
```

highListOptions: Mensagem de resposta à requisição de maiores altas opções. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação;

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso highListOptions;

Parameters. subsbribetype: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend({
  "token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",
  "module": "quotes",
  "service": "highListOptions ",
  "parameterGet": "bovespa",
  "parameters": {"subsbribetype": "1"}
});
```

Irá receber os dados do ativo informado:

quote: Símbolo do ativo;

price: Preço do ativo;

value: Valor do item da lista, para as listas de altas e baixas é o valor da variação do preço e para as listas de volume, é o volume negociado no dia;

date: Horário em que a linha foi atualizada. Formato HHMMSS

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto StreamingResumeMarketList]

```
{"streamingResumeMarketList":{"M":"bovespa","L":[{"Q":"IBOVR68","P":213.0,"V":5225.0,"D":"08-06-2018 11:51:58"}, {"Q":"RAILF14","P":0.35,"V":1066.67,"D":"Fri Jun 08 15:12:49 BRT 2018"}, {"Q":"VALER435","P":0.06,"V":500.0,"D":"08-06-2018 14:43:27"}, {"Q":"RENTT227","P":1.38,"V":430.77,"D":"08-06-2018 11:19:22"}, {"Q":"ELETFF18","P":0.05,"V":400.0,"D":"08-06-2018 14:46:30"}, {"Q":"PETRS40","P":0.09,"V":350.0,"D":"08-06-2018 14:04:17"}, {"Q":"KROTR15","P":4.5,"V":336.89,"D":"08-06-2018 11:19:15"}, {"Q":"BRFSS21","P":1.0,"V":334.78,"D":"08-06-2018 12:36:17"}, {"Q":"RAILG14","P":0.6,"V":328.57,"D":"08-06-2018 14:50:03"}, {"Q":"JBSSF9","P":0.34,"V":325.0,"D":"08-06-2018 12:52:04"}, {"Q":"GOAUR678","P":0.42,"V":320.0,"D":"08-06-2018 11:19:19"}]},{"type":"ResumeMarketType","parameter":"bovespa"}
```

fallListOptions: Mensagem de resposta à requisição de maiores baixas opções. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação;

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso fallListOptions;

Parameters. subsbribeType: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend({  
  
  "token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",  
  
  "module": "quotes",  
  
  "service": "fallListOptions",  
})
```

```
"parameterGet": "bovespa",  
"parameters": {"subsribetype": "1"}  
});
```

Ir  receber os dados do ativo informado:

quote: S mbolo do ativo;

price: Pre o do ativo;

value: Valor do item da lista, para as listas de altas e baixas   o valor da varia  o do pre o e para as listas de volume,   o volume negociado no dia;

date: Hor rio em que a linha foi atualizada. Formato HHMMSS

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto StreamingResumeMarketList]

```
{"streamingResumeMarketList":{"M":"bovespa","L":[{"Q":"IBOVF74","P":490.0,"V":-95.71,"D":"08-06-2018 12:05:43"}, {"Q":"FIBRF79","P":0.01,"V":-95.24,"D":"08-06-2018 11:19:16"}, {"Q":"VALEU32","P":0.1,"V":-96.97,"D":"Fri Jun 08 15:10:15 BRT 2018"}, {"Q":"KROTG14","P":0.07,"V":-92.93,"D":"Fri Jun 08 15:02:35 BRT 2018"}, {"Q":"ABEVG192","P":0.45,"V":-90.43,"D":"08-06-2018 13:27:41"}, {"Q":"IBOVL105","P":213.0,"V":-89.76,"D":"08-06-2018 12:10:48"}, {"Q":"PETRH46","P":1.2,"V":-89.14,"D":"08-06-2018 14:44:28"}, {"Q":"IBOVF17","P":1358.0,"V":-88.49,"D":"08-06-2018 11:26:28"}, {"Q":"ITUBG511","P":0.05,"V":-87.5,"D":"08-06-2018 11:19:40"}, {"Q":"GGBRF19","P":0.01,"V":-87.5,"D":"08-06-2018 13:14:26"}, {"Q":"ABEVI83","P":0.08,"V":-93.89,"D":"08-06-2018 15:07:59"}, {"Q":"ABEVI224","P":0.13,"V":-92.86,"D":"08-06-2018 15:08:42"}, {"Q":"ABEVT77","P":0.04,"V":-94.29,"D":"08-06-2018 15:45:03"}]},{"type":"ResumeMarketType","parameter":"bovespa"}}
```

highListNegotiation: Mensagem de resposta   requisi  o de maiores altas negociados. Para o funcionamento, basta utilizar o mesmo websocket da autentica  o mudando a mensagem JSON de envio. O qual dever  ser passado:

Token: Valor retornado na autentica  o;

Module: O modulo em que o servi o se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso highListNegotiation;

Parameters. subsbrietype: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend('{  
  "token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",  
  "module": "quotes",  
  "service": "highListNegotiation",  
  "parameterGet": "bovespa",  
  "parameters": {"subsbrietype": "1"}  
});
```

Irá receber os dados do ativo informado:

quote: Símbolo do ativo;

quantity: Quantidade de negocios realizados para aquele ativo no dia;

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto streamingResumeMarketList]

```
{"streamingResumeMarketList":{"M":"bovespa","L":[{"Q":"PETR4","QT":78818,"AF":["PETR4","78.82K"]},  
{"Q":"VALE3","QT":50490,"AF":["VALE3","50.49K"]},{"Q":"BBAS3","QT":50349,"AF":["BBAS3","50.35K"]},  
{"Q":"RAIL3","QT":39555,"AF":["RAIL3","39.56K"]},{"Q":"B3SA3","QT":37906,"AF":["B3SA3","37.91K"]},  
{"Q":"JBSS3","QT":37539,"AF":["JBSS3","37.54K"]},{"Q":"ITUB4","QT":36743,"AF":["ITUB4","36.74K"]},  
{"Q":"BBDC4","QT":36344,"AF":["BBDC4","36.34K"]},{"Q":"KROT3","QT":31525,"AF":["KROT3","31.52K"]},  
{"Q":"ITSA4","QT":30367,"AF":["ITSA4","30.37K"]}],{"type":"MarketQuantityTradesType","parameter":"bovespa"}}
```

highListFuturo: Mensagem de resposta à requisição de maiores altas futuro. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação;

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso highListFuturo;

Parameters. subsbribeType: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend('{  
  "token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",  
  "module": "quotes",  
  "service": " highListFuturo",  
  "parameterGet": "bovespa",  
  "parameters": {"subsbribeType": "1"}  
});
```

Irá receber os dados do ativo informado:

quote: Símbolo do ativo;

price: Preço do ativo;

value: Valor do item da lista, para as listas de altas e baixas é o valor da variação do preço e para as listas de volume, é o volume negociado no dia;

date: Horário em que a linha foi atualizada. Formato HHMMSS

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto streamingResumeMarketList]

```
{ "streamingResumeMarketList": { "M": "bovespa", "L": { { "Q": "TRPN1", "P": 0.48, "V": 128.57, "D": "08-06-2018 13:32:15"}, { "Q": "BAUH4", "P": 12.35, "V": 12.27, "D": "Fri Jun 08 15:17:04 BRT 2018"}, { "Q": "IGBR3", "P": 0.88, "V": 8.64, "D": "Fri Jun 08 16:33:00 BRT 2018"}, { "Q": "UNIP3", "P": 39.3, "V": 8.2, "D": "Fri Jun 08 16:03:20 BRT 2018"}, { "Q": "BTOW3", "P": 24.68, "V": 7.3, "D": "Fri Jun 08 16:34:13 BRT 2018"}, { "Q": "FNAM11", "P": 0.13, "V": 8.33, "D": "08-06-2018 14:52:36"}, { "Q": "TEKA4", "P": 7.0, "V": 7.69, "D": "08-06-2018" } } }
```

```
13:52:35"},{"Q":"MNPR3","P":2.65,"V":7.29,"D":"08-06-2018
11:19:30"},{"Q":"ADHM3","P":1.7,"V":6.25,"D":"08-06-2018
11:19:12"},{"Q":"XPLG11","P":99.44,"V":7.7,"D":"Fri Jun 08 15:02:45 BRT
2018"},{"Q":"ALUP3","P":7.1,"V":7.58,"D":"Fri Jun 08 15:15:07 BRT
2018"},{"Q":"SHUL4","P":7.05,"V":7.31,"D":"Fri Jun 08 15:25:45 BRT
2018"},{"Q":"WIZS3","P":8.04,"V":7.92,"D":"Fri Jun 08 16:37:19 BRT
2018"},{"Q":"RAIL3","P":13.75,"V":7.42,"D":"Fri Jun 08 15:15:31 BRT
2018"},{"Q":"JHSF3","P":1.11,"V":7.77,"D":"Fri Jun 08 16:36:15 BRT
2018"},{"Q":"EVEN3","P":4.04,"V":6.32,"D":"Fri Jun 08 16:27:24 BRT
2018"},{"Q":"TGMA3","P":17.4,"V":6.49,"D":"Fri Jun 08 15:56:41 BRT
2018"},{"Q":"KROT3","P":10.82,"V":6.92,"D":"Fri Jun 08 16:37:54 BRT
2018"},{"Q":"VLID3","P":17.75,"V":6.29,"D":"Fri Jun 08 16:20:05 BRT
2018"},{"Q":"BEES4","P":3.84,"V":8.17,"D":"08-06-2018
16:27:38"},{"Q":"ETER3","P":0.4,"V":8.11,"D":"08-06-2018
16:28:32"}]], "type": "ResumeMarketType", "parameter": "bovespa"}
```

fallListFuturo: Mensagem de resposta à requisição de maiores baixas futuro. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação;

Module: O modulo em que o serviço se encontra, no caso quotes;

Service: O serviço o qual está requisitando, no caso fallListFuturo;

Parameters. subsbrietype: Indica se quer que o servidor envie os dados ou não, valores válidos (1, 0);

Segue o exemplo de envio:

```
doSend({
  "token": "32ec4ab4-905d-4724-82df-40c6c5fae16f",
  "module": "quotes",
  "service": " fallListFuturo",
  "parameterGet": "bovespa",
  "parameters": {"subsbrietype": "1"}
});
```

Irá receber os dados do ativo informado:

quote: Símbolo do ativo;

price: Preço do ativo;

value: Valor do item da lista, para as listas de altas e baixas é o valor da variação do preço e para as listas de volume, é o volume negociado no dia;

date: Horário em que a linha foi atualizada. Formato HHMMSS

Segue abaixo o exemplo da resposta:

Resposta:

[Objeto streamingResumeMarketList]

```
{"streamingResumeMarketList":{"M":"bovespa","L":[{"Q":"RBCB11","P":340.0,"V":-17.07,"D":"Fri Jun 08 15:47:17 BRT 2018"}, {"Q":"SUZB3","P":44.66,"V":-9.69,"D":"Fri Jun 08 16:42:33 BRT 2018"}, {"Q":"CTSA3","P":3.67,"V":-9.38,"D":"Fri Jun 08 16:07:48 BRT 2018"}, {"Q":"TCNO4","P":1.45,"V":-8.23,"D":"Fri Jun 08 16:38:48 BRT 2018"}, {"Q":"VIVR3","P":0.25,"V":-7.41,"D":"Fri Jun 08 15:47:09 BRT 2018"}, {"Q":"FIVN11","P":4.0,"V":-6.98,"D":"08-06-2018 12:46:59"}, {"Q":"SLED4","P":3.25,"V":-8.71,"D":"Fri Jun 08 16:40:40 BRT 2018"}, {"Q":"ELET6","P":14.5,"V":-5.72,"D":"Fri Jun 08 15:03:20 BRT 2018"}, {"Q":"BRAP4","P":30.3,"V":-7.03,"D":"Fri Jun 08 16:42:13 BRT 2018"}, {"Q":"AAPL34","P":70.72,"V":-7.2,"D":"Fri Jun 08 16:08:26 BRT 2018"}, {"Q":"CTKA4","P":3.2,"V":-5.6,"D":"08-06-2018 13:40:53"}, {"Q":"BRFS3","P":21.85,"V":-5.98,"D":"Fri Jun 08 15:53:35 BRT 2018"}, {"Q":"FSLR34","P":97.98,"V":-7.43,"D":"Fri Jun 08 16:08:26 BRT 2018"}, {"Q":"RNEW4","P":0.53,"V":-7.02,"D":"08-06-2018 15:05:30"}, {"Q":"ALUP4","P":4.03,"V":-6.28,"D":"08-06-2018 15:06:07"}, {"Q":"BPAC3","P":7.7,"V":-5.87,"D":"08-06-2018 15:26:46"}, {"Q":"HGBS11","P":206.0,"V":-7.62,"D":"Fri Jun 08 15:34:18 BRT 2018"}, {"Q":"CARE11","P":1.75,"V":-11.17,"D":"Fri Jun 08 15:52:28 BRT 2018"}, {"Q":"TRPN1","P":0.19,"V":-9.52,"D":"08-06-2018 15:56:02"}, {"Q":"COPH34","P":63.96,"V":-7.02,"D":"08-06-2018 16:09:02"}, {"Q":"EBAY34","P":74.32,"V":-7.65,"D":"08-06-2018 16:13:26"}]}}
```

III-Módulo Chart

CandleChart: O serviço retorna os dados referentes ao ativo solicitado, para realizar a construção do gráfico. Para o funcionamento, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso chart;

Service: O serviço o qual está requisitando, no caso candleChart;

Parameters.quote: Qual ativo deve ser retornado as informações;

Parameters.period: Qual período a ser buscado no serviço (S-semanal, D-diário, M-mensal, minutos passar valores (1, 5, 10, 15, 30, 60));

Parameters.quantity: Quantidade de linhas a serem retornadas pelo serviço na primeira resposta, as demais serão retornadas apenas para atualização;

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (**true**, **false**).

Parameters.initDate: Data inicial para que seja retornado os candles apenas a partir desta data e hora, formato YYYYMMDDHHMM.

Parameters.fimDate: Data final para que seja retornado os candles apenas até esta data e hora, formato YYYYMMDDHHMM.

Parameters.filter: Indica quais campos quer receber no processamento da mensagem, deve ser passado separados por vírgula os campos do **Candle** como por exemplo: **date,price**.

Segue o exemplo de envio:

```
doSend('{  
  "token": "f0d66fcb-7055-4f86-b6a9-25ccae3f9a68",  
  "module": "chart",  
  "service": "candleChart",  
  "parameters": {"quote": "petr4", "period": "S", "quantity": "2", "realTime": "true",  
  "dispatch": "true", "initDate": "201707271453", "fimDate": "201707271653", "filter": "date,price"}}');
```

Irá receber os dados de acordo com os parâmetros informado:

Quote: Indica o ativo o qual foi pedido;

Period: Indica o período o qual foi pedido;

Tick_size: Quantidade de casas decimais do ativo retornado pelo difusor;

Response_type: Tipo de comando no difusor, no caso **gch**;

Error: Texto de erro, caso haja;

Subscribe_id: Id de conexão com o difusor;

Candles[].date: Data da cotação;

Candles[].price: Valor do ativo na data;

Candles[].open: Valor do ativo de abertura do mercado;

Candles[].high: Maior valor do ativo na data;

Candles[].low: Menor valor do ativo na data;

Candles[].fVol: Volume financeiro;

Candles[].aVol: Volume quantitativo;

Candles[].isClose: Determina se o candle está fechado.

Segue abaixo o exemplo da resposta:

```
{"quote": "petr4", "period": "S", "tick_size": "2", "response_type": "gch", "error": "", "subscribe_id": "-30333404", "candles": [{"date": "201609050000", "price": "13.51", "open": "13.79", "high": "14.29", "low": "13.51", "fVol": "2114805595", "aVol": "152422000", "isClose": "true"}, {"date": "201609120000", "price": "13.26", "open": "13.32", "high": "13.95", "low": "12.94", "fVol": "3401912856", "aVol": "254291700", "isClose": "false"}]}
```

III-Módulo Negociação

dailyOrder: O serviço retorna os dados referentes as ordens da conta solicitada. Para o funcionamento, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso negotiation;

Service: O serviço o qual está requisitando, no caso dailyOrder;

Parameters.account: De qual conta deve ser retornado as informações;

Parameters.market: Qual mercado a conta pertence, valores válidos (**XBSP** – Bovespa , **XBMF** – BM&F);

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (**true, false**);

Parameters.history: Indica se quer que o servidor envie os dados de histórico das ordens ou não, valores válidos (**true, false**).

Parameters.statusGeneric: Indica se quer que o servidor as ordens de um determinado status genérico pré-definido, valores válidos (**Abertas, Canceladas, Executadas, Vigentes**).

Segue o exemplo de envio:

```
doSend('{  
  "token": "f0d66fcb-7055-4f86-b6a9-25ccae3f9a68",  
  "module": "negotiation",  
  "service": "dailyOrder",  
  "parameters": {"account": "123456", "market": "XBSP", "dispatch": "true", "history": "true",  
  "statusGeneric": "canceladas"}  
});
```

Irá receber os dados de acordo com os parâmetros informado:

listBeans[].quote: Indica o ativo da ordem;

listBeans[].buyOrSell: Direção da ordem;

listBeans[].account: Identificador da conta;

listBeans[].state: Status atual da ordem;

listBeans[].quantitySupplied: Número de contratos ordenados;

listBeans[].quantityExecuted: Quantidade de contratos já executados;

listBeans[].quantityRemaining: Quantidade de contratos ainda abertos para execução;

listBeans[].price: Preço por contrato;

listBeans[].priceStop: Preço de Stop em uma ordem do tipo Stop Limit;

listBeans[].priceStopGain: Preço de StopGain para uma ordem Stop Limit;

listBeans[].price2: Preço limite para StopGain;

listBeans[].movingStart: Preço de início móvel, configurado para Ordens Start/Stop Móvel;

listBeans[].initialChangeType: Código que determina se valor em campo InitialChange;

listBeans[].initialChange: Preço de alteração inicial do Price, quando LastTrade ultrapassar MovingStart, configurado quando ordem Start/Stop Móvel;

listBeans[].pricePercentage: Percentual usado para calcular o preço limite em uma ordem Start/Stop Móvel;

listBeans[].maxFloor: Quantidade aparente;

listBeans[].priceAverage: Preço médio dos contratos já negociados até agora na ordem;

listBeans[].type: Tipo da ordem;

listBeans[].maturity: Data e hora da expiração da ordem;

listBeans[].mercado: Mercado ao qual a ordem pertence;

listBeans[].transactTime: Hora da execução ou da criação da ordem;

listBeans[].side: Direção da ordem;

listBeans[].currentCumQty: Quantidade executada do dia;

listBeans[].currentAvgPx: Preço médio do dia;

listBeans[].orderId: Identificador da ordem;

listBeans[].clOrdID: Identificador da ordem como definido pelo iniciador;

listBeans[].timeInForce: Especifica o tipo de duração da ordem;

listBeans[].username: Identificador do Participante;

listBeans[].text: Texto de formato livre;

listBeans[].sourceAddress: Utilizado para armazenar a origem da Ordem;

listBeans[].orderStrategy: Utilizado para especificar a estratégia da Ordem;

listBeans[].execType – Descreve a ação que disparou a mensagem;

listBeans[].startStopClOrdId – Identificador da ordem Start/Stop que originou esta ordem;

listBeans[].applicationName – Tag que informa o identificador da aplicação utilizada pelo usuário;

OBS.: Campos retornados no oms da RICO, para fazer o calculo do lucro/prejuizo nos apps:

listBeans[].SoldQty: tag 10147;

listBeans[].BoughtQty: tag 10145;

listBeans[].BuyAvgPx: tag 35013;

listBeans[].SellAvgPx: tag 35014;

listBeans[].Factor: tag 228.

listBeans[].EnteringTrader: tag 448_A.

Segue abaixo o exemplo da resposta:

```
{
  "listBeans": [
    {
      "quote": {
        "name": "Ativo",
        "value": "VALE5"
      },
      "buyOrSell": {
        "name": "Compra/Venda",
        "value": "Venda"
      },
      "account": {
        "name": "Conta",
        "value": "123456"
      },
      "state": {
        "name": "Estado",
        "value": "Nova"
      },
      "quantitySupplied": {
        "name": "Quantidade Ofertada",
        "value": "100"
      },
      "quantityExecuted": {
        "name": "Quantidade Executada",
        "value": "0"
      },
      "quantityRemaining": {
        "name": "Quantidade Restante",
        "value": "100"
      },
      "price": {
        "name": "Preço",
        "value": "R$ 9,99"
      },
      "priceAverage": {
        "name": "Preço Médio",
        "value": "R$ 0,00"
      },
      "type": {
        "name": "Tipo",
        "value": "Limitada"
      },
      "maturity": {
        "name": "Vencimento",
        "value": "VAC"
      },
      "mercado": {
        "name": "Mercado",
        "value": "Bovespa"
      },
      "transactTime": {
        "name": "TransactTime",
        "value": "11/05/2016 15:14:55"
      },
      "side": {
        "name": "Side",
        "value": "Venda"
      },
      "currentCumQty": {
        "name": "CurrentCumQty",
        "value": ""
      },
      "currentAvgPx": {
        "name": "CurrentAvgPx",
        "value": ""
      },
      "orderID": {
        "name": "Identificador da Ordem",
        "value": "2016051118145547"
      },
      "timeInForce": {
        "name": "Tipo de duração da ordem",
        "value": "1"
      }
    }
  ]
}
```

position Mensagem de resposta à requisição de custódia da conta. Para o funcionamento, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso negotiation;

Service: O serviço o qual está requisitando, no caso position;

Parameters.account: De qual conta deve ser retornado as informações;

Parameters.market: Qual mercado a conta pertence, valores válidos (**XBSP** – Bovespa , **XBMF** – BM&F);

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (**true**, **false**);

Segue o exemplo de envio:

```
doSend({
  "token": "f0d66fcb-7055-4f86-b6a9-25ccae3f9a68",
  "module": "negotiation",
  "service": "position",
  "parameters": {
    "account": "10000",
    "market": "XBSP",
    "dispatch": "true"
  }
})
```

```
});
```

Resposta:

[Objeto PositionBean]

account
marketID
posMainRptID
quote
quoteType
currentQuantity
lastTrade
previusTrade
averagePrice
initQuantity
quantityD1
quantityD2
quantityD3
priceAveragelnit
valueCurrent
valueInit
valueSpent
profitLoss
profitLossPercent
openBuy
openSell
executedBuy
executedSell
expired
quantityBlocked
quantityReleased
lastTradeDate
qtdTomadorBTC
qtdDoadorBTC
precoMedioTomadorBTC
precoMedioDoadorBTC
netDay
netPosition
netDaytrade
averagePricePos
openProfitLoss
closedProfitLoss

Segue abaixo o exemplo da resposta:

```
{
  "listBeans": [
    {
      "account": { "value": "10000" },
      "marketID": { "value": "XBSP", "code": "XBSP" },
      "posMainRptID": { "value": "10000", "req7": "" },
      "quote": { "value": "ABEV3" },
      "quoteType": { "value": "" },
      "currentQuantity": { "value": "1.300" },
      "lastTrade": { "value": "R$ 20,76" },
      "previousTrade": { "value": "R$ 20,71" },
      "averagePrice": { "value": "R$ 30,43" },
      "initQuantity": { "value": "400" },
      "quantityD1": { "value": "0" },
      "quantityD2": { "value": "400" },
      "quantityD3": { "value": "900" },
      "priceAverageInit": { "value": "" },
      "valueCurrent": { "value": "" },
      "valueInit": { "value": "" },
      "valueSpent": { "value": "" },
      "profitLoss": { "value": "" },
      "profitLossPercent": { "value": "" },
      "openBuy": { "value": "0" },
      "openSell": { "value": "0" },
      "executedBuy": { "value": "900" },
      "executedSell": { "value": "0" },
      "expired": { "value": "" },
      "quantityBlocked": { "value": "0" },
      "quantityReleased": { "value": "" },
      "lastTradeDate": { "value": "07/12/2017 16:16:48.000" },
      "qtdTomadorBTC": { "value": "0" },
      "qtdDoadorBTC": { "value": "0" },
      "precoMedioTomadorBTC": { "value": "0" },
      "precoMedioDoadorBTC": { "value": "0" },
      "netDay": { "value": "900" },
      "netPosition": { "value": "0" },
      "netDaytrade": { "value": "900" },
      "averagePricePos": { "value": "R$ 20,69" },
      "openProfitLoss": { "value": "R$ 60,00" },
      "closedProfitLoss": { "value": "R$ 0,00" }
    }
  ]
}
```

cancelOrderReject: Notifica a rejeição de um Order Cancel Reject (Cancelamento de Ordem) ou de um Order Cancel.

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso negotiation;

Service: O serviço o qual está requisitando, no caso position;

Parameters.account: De qual conta deve ser retornado as informações;

Parameters.market: Qual mercado a conta pertence, valores válidos (**XBSP** – Bovespa, **XBMF** – BM&F);

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (**true**, **false**);

Segue o exemplo de envio:

```
doSend({
  "token": "bb70c590-c646-4b42-b705-814fe44db3ff",
  "module": "negotiation",
  "service": "cancelOrderReject",
  "parameters": { "account": "88564", "market": "XBSP", "dispatch": "true" }
});
```

Resposta:

[Objeto OrderCancelRejectBean]

clOrdID
orderID
volumeAmount
origClOrdID
ordStatus
side
cxlRejResponseTo
cxlRejReason
orderTag
cxlRejSource

statusOMS: Mensagem de resposta à requisição de status do OMS. Para o funcionamento, basta utilizar o mesmo websocket da autenticação mudando a mensagem JSON de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso negotiation;

Service: O serviço o qual está requisitando, no caso statusOMS;

Parameters.account: De qual conta deve ser retornado as informações;

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (true, false);

Segue o exemplo de envio:

```
doSend('{  
  "token": "96d9fae1-37e5-46ac-82bd-c3574a7ae5b6",  
  "module": "negotiation",  
  "service": "statusOMS",  
  "parameters": {"dispatch": "true"}  
});
```

Resposta:

[Objeto StatusOMSResponse]

statusOMS -> Valores possíveis (connected/disconnected)

financialAccountInformation Mensagem de resposta à requisição de financeiro da conta. Para o funcionamento, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso negotiation;

Service: O serviço o qual está requisitando, no caso financialAccountInformation;

Parameters.account: De qual conta deve ser retornado as informações;

Parameters.market: Qual mercado a conta pertence, valores válidos (**XBSP** – Bovespa , **XBMF** – BM&F);

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (**true**, **false**);

Segue o exemplo de envio:

```
doSend('{  
  "token": "ae37e8cd-1969-4e66-a09d-2e22b516b16a",  
  "module": "negotiation",  
  "service": "financialAccountInformation",  
  "parameters": {"account": "10000", "market": "XBSP", "dispatch": "true"}  
});
```

Resposta:

[Objeto FinancialInformationBean]

marketID (1301) - Mercado;

account (1) - Conta;

financialAccountRejReason (10319) - Motivo da rejeição;

text (58) - Texto da rejeição;

saldoAbertura - INBA;

saldoEmConta - AVBA;

contaMargemTotal - ACMT;

contaMargemUtilizada - ACMU;

proventos - DVDS;

tesouroDireto - TRDI;

clubes - CLFU;

rendaFixa - INFU;

fundos - FNDS;
limiteOperacional - UNQL;
alavancagemVista - CHLL;
alavancagemOpcoes - OPLL;
alavancagemBMF - BMFLL;
lucroPrejuizoOpen - OPL;
lucroPrejuizoClose - CLPL;
lucroPrejuizoTotal - (OPL + CLPL);
patrimonioInicial - INEQ;
patrimonioOnline - TOEQ;
creditEntryDay0 - CED0;
creditEntryDay1 - CED1;
creditEntryDay2 - CED2;
creditEntryDay3 - CED3;
debitEntryDay0 - DED0;
debitEntryDay1 - DED1;
debitEntryDay2 - DED2;
debitEntryDay3 - DED3;
saldoProjOnline - (INBA + CED0 - DED0 + CED1 - DED1 + CED2 - DED2 + CED3 - DED3);
saldoD0 - (CED0 - DED0);
saldoD1 - (CED1 - DED1);
saldoD2 - (CED2 - DED2);
saldoD3 - (CED3 - DED3);
margemGarantias - BMFWT;
consideradoPI - BMCL;
margemBMFLivre - BMFMA;
ajustePregaoBMF - BTAD;
ajustePregaoBOV - BPAD;
saldoProjetadoInicial - INPRBA;
taxas - INPRBA;
custosCorretagemBov - BGBSP;
custosCorretagemBMF - BGBMF;
emolumentos - EMOL;
custodiaDesagiadaOpcoes - OPLVG;
custodiaDesagiadaVista - LEVE;
margemGarantiaBovTotal - BSPWT;
margemGarantiaBovConsideradoPL - GNMA;
openProfitLossDaytrade - OPLDT;
closeProfitLossDaytrade - CPLDT;
dayTradeAdjustment - DTAD;
blockedValueLimit - BLVL;
bmfNotExecutedPurchaseAmount - BNEP;
bmfNotExecutedSellAmount - BNES;

cashOpenPurchases - CHOP;
optionsNotExecutedPurchaseAmount - OPOP;
cashOpenSells - CHOS;
optionsNotExecutedSellAmount - OPOS;
warrantyRequired - WREQ;
warrantyDeposited - WDEP;

Segue abaixo o exemplo da resposta:

```
{
  "code": 0,
  "financialInformationBean": {
    "marketID": {
      "value": "XBSP"
    },
    "account": {
      "value": "10000"
    },
    "financialAccountRejReason": {
      "value": ""
    },
    "text": {
      "value": ""
    },
    "saldoAbertura": {
      "value": "3926.03"
    },
    "saldoEmConta": {
      "value": "-592855.97"
    },
    "contaMargemTotal": {
      "value": ""
    },
    "contaMargemUtilizada": {
      "value": ""
    },
    "proventos": {
      "value": "76.08"
    },
    "tesouroDireto": {
      "value": "0"
    },
    "clubes": {
      "value": "0"
    },
    "rendaFixa": {
      "value": "0"
    },
    "fundos": {
      "value": "0"
    },
    "limiteOperacional": {
      "value": "-1188242.989"
    },
    "alavancagemVista": {
      "value": "0"
    },
    "alavancagemOpcoes": {
      "value": "0"
    },
    "alavancagemBMF": {
      "value": "0"
    },
    "lucroPrejuizoOpen": {
      "value": "3467"
    },
    "lucroPrejuizoClose": {
      "value": "10100.000000001"
    },
    "lucroPrejuizoTotal": {
      "value": "13567.000000"
    },
    "patrimonioInicial": {
      "value": "-946737.89"
    },
    "patrimonioOnline": {
      "value": "-933170.8899999999"
    },
    "creditEntryDay0": {
      "value": "12095170.3"
    },
    "creditEntryDay1": {
      "value": "22904"
    },
    "creditEntryDay2": {
      "value": "0"
    },
    "creditEntryDay3": {
      "value": "0"
    },
    "debitEntryDay0": {
      "value": "12691952.3"
    },
    "debitEntryDay1": {
      "value": "371042"
    },
    "debitEntryDay2": {
      "value": "5820"
    },
    "debitEntryDay3": {
      "value": "1997"
    },
    "saldoProjOnline": {
      "value": "-948810.970000"
    },
    "saldoD0": {
      "value": "-596782.000000"
    },
    "saldoD1": {
      "value": "-348138.000000"
    },
    "saldoD2": {
      "value": "-5820.000000"
    },
    "saldoD3": {
      "value": "-1997.000000"
    },
    "margemGarantias": {
      "value": "0"
    },
    "consideradoPL": {
      "value": "0"
    },
    "margemBMFLivre": {
      "value": "0"
    },
    "ajustePregaoBMF": {
      "value": "13575.000000001"
    },
    "ajustePregaoBOV": {
      "value": "0"
    },
    "saldoProjetadoInicial": {
      "value": "-946813.97"
    },
    "taxas": {
      "value": "-946813.97"
    },
    "custosCorretagemBov": {
      "value": ""
    },
    "custosCorretagemBMF": {
      "value": ""
    },
    "emolumentos": {
      "value": ""
    },
    "custodiaDesagiadaOpcoes": {
      "value": "0"
    },
    "custodiaDesagiadaVista": {
      "value": "1989"
    },
    "margemGarantiaBovTotal": {
      "value": "0"
    },
    "margemGarantiaBovConsideradoPL": {
      "value": "0"
    },
    "openProfitLossDaytrade": {
      "value": ""
    },
    "closeProfitLossDaytrade": {
      "value": ""
    },
    "dayTradeAdjustment": {
      "value": ""
    },
    "blockedValueLimit": {
      "value": ""
    },
    "bmfNotExecutedPurchaseAmount": {
      "value": ""
    },
    "bmfNotExecutedSellAmount": {
      "value": ""
    },
    "cashOpenPurchases": {
      "value": "1000"
    },
    "optionsNotExecutedPurchaseAmount": {
      "value": "0"
    },
    "cashOpenSells": {
      "value": "5000"
    },
    "optionsNotExecutedSellAmount": {
      "value": "0"
    },
    "warrantyRequired": {
      "value": ""
    },
    "warrantyDeposited": {
      "value": ""
    }
  }
}
```

financialAccountInformationCompl Mensagem de resposta à requisição de financeiro completo da conta. Para o funcionamento, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso negotiation;

Service: O serviço o qual está requisitando, no caso financialAccountInformationCompl;

Parameters.account: De qual conta deve ser retornado as informações;

Parameters.market: Qual mercado a conta pertence, valores válidos (**XBSP** – Bovespa , **XBMF** – BM&F);

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (**true, false**);

Segue o exemplo de envio:

```
doSend('{  
  
  "token": "ae37e8cd-1969-4e66-a09d-2e22b516b16a",  
  
  "module": "negotiation",  
  
  "service": "financialAccountInformationCompl",  
  
  "parameters": {"account": "10000", "market": "XBSP", "dispatch": "true"}  
  
});
```

Resposta:

[Objeto FinancialInformationComplBean]

marketID (1301) - Mercado;

account (1) - Conta;

financialAccountRejReason (10319) - Motivo da rejeição;

text (58) - Texto da rejeição;

finalMarkToMarketAmount - FMTM;

incrementalMarkToMarketAmount - IMTM;

tradeVariationAmount - TVAR;

startOfDayMarkToMarketAmount - SMTM;

premiumAmount - PREM;

cashResidualAmount - CRES;

cashAmount - CASH;

valueAdjustedAmount - VADJ;

currentValueAmount - CVA;

marginAccountBalance - MACB;

totalBlockedBalance - TBOB;

blockedMargin - BLMA;

futureSettlementBalance - FUSB;

blockedFromOthers - BLFO;

availableWithdrawnBalance - AWTB;
adjustedBalance - ADBA;
initialMarginBalance - IMAB;
maintainanceMarginBalance - MMBA;
availableNegotiationBalance - ANBA;
recompositionValue - REVA;
availableBalance - AVBA;
currentDayBalance - CDBA;
entryDay0 - END0;
entryDay1 - END1;
entryDay2 - END2;
entryDay3 - END3;
financialVolume - FIVO;
projectedBalance - PRBA;
initialBalance - INBA;
creditEntryD0 - CED0;
creditEntryD1 - CED1;
creditEntryD2 - CED2;
creditEntryD3 - CED3;
debitEntryD0 - DED0;
debitEntryD1 - DED1;
debitEntryD2 - DED2;
debitEntryD3 - DED3;
startCreditEntryD1 - CESD1;
startCreditEntryD2 - CESD2;
startDebitEntryD1 - DESD1;
startDebitEntryD2 - DESD2;
financialLeverage - FNLE;
overdraftCredit - OVCR;
balanceCashMarketBovespa - BCMB;
balanceCashMarketBmf - BCMF;
balanceUncoveredOperation - BUOP;
balanceOptions - BOPT;
previousClosePrice - PCP;
currentInstrumentPrice - CIP;
openAvgValueToday - OAVT;
totalPurchaseAmt - TPV;
totalSaleAmt - TSV;
blocked - BCKA;
openedPurchaseAmmount - OPAM;
openedSellAmmount - OSAM;
cashLimit - CHLT;
optionLimit - OPLT;

dailyFinancialActivities - DFAT;
bspCashOperationCredit - BSCC;
bspOptionOperationCredit - BSOC;
bmfOperationCredit - BMFC;
operationEquityCredit - OEC;
financialAccountRisk - FARK;
overdraftLimit - OVLT;
overdraftAmmount - OVAM;
potencialOverdraftAmmount - POVA;
bmfBlockedMargin - BMBM;
adjustD0 - ADD0;
adjustD1 - ADD1;
bmfLimit - BMFL;
bmfExecutedBlockedMargin - BEBM;
bmfNotExecutedBlockedMargin - BNBM;
bmfPurchaseOpenMargin - BPOM;
bmfPurchaseExecutedMargin - BPEM;
bmfSellOpenMargin - BSOM;
bmfSellExecutedMargin - BSEM;
notExecutedPurchaseAmmount - NEPA;
notExecutedSellAmmount - NESA;
bmfNotExecutedPurchaseAmmount - BNEP;
bmfNotExecutedSellAmmount - BNES;
bmfUnblockedMargin - BUNM;
filledOverdraftOptionAmmount - FOOA;
notExecutedOverdraftOptionAmmount - NOOA;
xbmfMarginConsidered - BMCL;
cashOpenPurchases - CHOP;
optionsNotExecutedPurchaseAmmount - OPOP;
cashOpenSells - CHOS;
optionsNotExecutedSellAmmount - OPOS;
cashExecutedPurchases - CHEP;
optionsExecutedPurchases - OPEP;
cashExecutedSells - CHES;
optionsExecutedSells - OPES;
uniqueLimit - UNQL;
blockedValueLimit - BLVL;
cashLeveragedLimit - CHLL;
optionsLeveragedLimit - OPLL;
bmfLeveragedLimit - BMFLL;
cashPurchaseLimit - CHPL;
optionsPurchaseLimit - OPPL;
bmfPurchaseLimit - BMFPL;

cashSellLimit - CHSL;
optionsSellLimit - OPSL;
bmfSellLimit - BMFSL;
cashPurchaseAmount - CHPA;
cashSellAmount - CHSA;
optionsPurchaseAmount - OPPA;
optionsSellAmount - OPSA;
bmfPurchaseAmount - BMFPA;
bmfSellAmount - BMFSA;
initialProjectedBalance - INPRBA;
initialLimit - INLT;
initialLimitBmf - INLTB;
bslimitOperationCredit - BSOPCR;
initialStockPortfolioValue - INSPV;
initialOptionsPortfolioValue - INOPV;
initialBmfPortfolioValue - INBPV;
totalStockPortfolioValue - TSPV;
totalOptionsPortfolioValue - TOPV;
totalBmfPortfolioValue - TBPV;
totalOnlineStockPortfolioValue - TOSPV;
totalOnlineOptionsPortfolioValue - TOOPV;
totalOnlineBmfPortfolioValue - TOBPV;
openProfitLoss - OPL;
closedProfitLoss - CLPL;
openProfitLossAux - OPLX;
closedProfitLossAux - CLPLX;
openProfitLossDaytrade - OPLDT;
closedProfitLossDaytrade - CPLDT;
initialEquity - INEQ;
totalEquity - TOEQ;
xbspMarginConsidered - GNMA;
optionsMargin - OPMA;
bmfPositionAdjust - BPAD;
dayTradeAdjustment - DTAD;
bmfTradingAdjust - BTAD;
ipoBlocked - IPOB;
creditTed - CTED;
manualAdjust - MADJ;
balanceWithoutLimit - BWLT;
optionsLeverageLimit - LOLI;
treasuryDirect - TRDI;
fixedIncome - INFU;
clubs - CLFU;

funds - FNDS;
fixedIncomeConsidered - INFUC;
clubsConsidered - CLFUC;
fundsConsidered - FNDSC;
dividends - DVDS;
xbpsWarrantyTotal - BSPWT;
xbmfWarrantyTotal - BMFWT;
xbmfMarginAvailable - BMFMA;
brokerageXbmf - BGBMF;
brokerageXbps - BGBSP;
taxes - TAX;
emolument - EMOL;
accountMarginTotal - ACMT;
accountMarginUsed - ACMU;
leverage - LEVE;
quoteOptionLeverage - OPLVG;
equityLeveraged - EQLVG;
d1CreditDaytrade - D1CDT;
d1DebitDaytrade - D1DDT;
d3CreditDaytrade - D3CDT;
d3DebitDaytrade - D3DDT;
d1CreditPosition - D1CPO;
d1DebitPosition - D1DPO;
d3CreditPosition - D3CPO;
d3DebitPosition - D3DPO;
negativePositionAmount - NPA;
negativeOptPositionAmount - NOPA;
creditLimit - CRDL;
buyAveragePrice - BAVP;
sellAveragePrice - SAVP;
warrantyRequired - WREQ;
warrantyDeposited - WDEP;
profitTotalLoss - (OPL + CLPL);
balanceProjOnline - (INBA + CED0 - DED0 + CED1 - DED1 + CED2 - DED2 + CED3 - DED3);
balanceD0 - (CED0 - DED0);
balanceD1 - (CED1 - DED1);
balanceD2 - (CED2 - DED2);
balanceD3 - (CED3 - DED3);

Segue abaixo o exemplo da resposta:

```
{"code":0,"financialInformationBean":{"marketID":{"value":"XBSP"},"account":{"value":"10000"},"financialAccountRejReason":{"value":""},"text":{"value":""},"finalMarkToMarketAmount":{"value":""},"incrementalMarkToMarketAmount":{"value":""},"tradeVariationAmount":{"value"}}
```

```
:""}, "startOfDayMarkToMarketAmount": {"value": ""}, "premiumAmount": {"value": ""}, "cashResidualAmount": {"value": ""}, "cashAmount": {"value": ""}, "valueAdjustedAmount": {"value": ""}, "currentValueAmount": {"value": ""}, "marginAccountBalance": {"value": ""}, "totalBlockedBalance": {"value": ""}, "blockedMargin": {"value": ""}, "futureSettlementBalance": {"value": ""}, "blockedFromOthers": {"value": ""}, "availableWithdrawnBalance": {"value": ""}, "adjustedBalance": {"value": ""}, "initialMarginBalance": {"value": ""}, "maintainanceMarginBalance": {"value": ""}, "availableNegotiationBalance": {"value": ""}, "recompositionValue": {"value": ""}, "availableBalance": {"value": "-592855.97"}, "currentDayBalance": {"value": ""}, "entryDay0": {"value": ""}, "entryDay1": {"value": ""}, "entryDay2": {"value": ""}, "entryDay3": {"value": ""}, "financialVolume": {"value": ""}, "projectedBalance": {"value": "-952737"}, "initialBalance": {"value": "3926.03"}, "creditEntryD0": {"value": "12095170.3"}, "creditEntryD1": {"value": "22904"}, "creditEntryD2": {"value": "0"}, "creditEntryD3": {"value": "0"}, "debitEntryD0": {"value": "12691952.3"}, "debitEntryD1": {"value": "371042"}, "debitEntryD2": {"value": "5820"}, "debitEntryD3": {"value": "1997"}, "startCreditEntryD1": {"value": "22904"}, "startCreditEntryD2": {"value": "0"}, "startDebitEntryD1": {"value": "371042"}, "startDebitEntryD2": {"value": "5820"}, "financialLeverage": {"value": ""}, "overdraftCredit": {"value": ""}, "balanceCashMarketBovespa": {"value": ""}, "balanceCashMarketBmf": {"value": ""}, "balanceUncoveredOperation": {"value": ""}, "balanceOptions": {"value": ""}, "previousClosePrice": {"value": ""}, "currentInstrumentPrice": {"value": ""}, "openAvgValueToday": {"value": ""}, "totalPurchaseAmt": {"value": ""}, "totalSaleAmt": {"value": ""}, "blocked": {"value": ""}, "openedPurchaseAmmount": {"value": ""}, "openedSellAmmount": {"value": ""}, "cashLimit": {"value": ""}, "optionLimit": {"value": ""}, "dailyFinancialActivities": {"value": ""}, "bspCashOperationCredit": {"value": ""}, "bspOptionOperationCredit": {"value": ""}, "bmfOperationCredit": {"value": ""}, "operationEquityCredit": {"value": ""}, "financialAccountRisk": {"value": ""}, "overdraftLimit": {"value": ""}, "overdraftAmmount": {"value": ""}, "potencialOverdraftAmmount": {"value": ""}, "bmfBlockedMargin": {"value": ""}, "adjustD0": {"value": ""}, "adjustD1": {"value": ""}, "bmfLimit": {"value": ""}, "bmfExecutedBlockedMargin": {"value": ""}, "bmfNotExecutedBlockedMargin": {"value": ""}, "bmfPurchaseOpenMargin": {"value": "0"}, "bmfPurchaseExecutedMargin": {"value": "98837.5825"}, "bmfSellOpenMargin": {"value": "128478"}, "bmfSellExecutedMargin": {"value": "19767.5165"}, "notExecutedPurchaseAmmount": {"value": ""}, "notExecutedSellAmmount": {"value": ""}, "bmfNotExecutedPurchaseAmmount": {"value": ""}, "bmfNotExecutedSellAmmount": {"value": ""}, "bmfUnblockedMargin": {"value": ""}, "filledOverdraftOptionAmmount": {"value": ""}, "notExecutedOverdraftOptionAmmount": {"value": ""}, "xbmfMarginConsidered": {"value": "0"}, "cashOpenPurchases": {"value": "1000"}, "optionsNotExecutedPurchaseAmmount": {"value": "0"}, "cashOpenSells": {"value": "5000"}, "optionsNotExecutedSellAmmount": {"value": "0"}, "cashExecutedPurchases": {"value": "1997"}, "optionsExecutedPurchases": {"value": "0"}, "cashExecutedSells": {"value": "0"}, "optionsExecutedSells": {"value": "0"}, "uniqueLimit": {"value": "-1188342.989"}, "blockedValueLimit": {"value": ""}, "cashLeveragedLimit": {"value": "0"}, "optionsLeveragedLimit": {"value": "0"}, "bmfLeveragedLimit": {"value": "0"}, "cashPurchaseLimit": {"value": "0"}, "optionsPurchaseLimit": {"value": "0"}, "bmfPurchaseLimit": {"value": "0"}, "cashSellLimit": {"value": "0"}, "optionsSellLimit": {"value": "0"}, "bmfSellLimit": {"value": "0"}, "cashPurchaseAmmount": {"value": "1997"}, "cashSellAmmount": {"value": "0"}, "optionsPurchaseAmmount": {"value": "0"}, "optionsSellAmmount": {"value": "0"}, "bmfPurchaseAmmount": {"value": "37813747.5"}, "bmfSellAmmount": {"value": "37864275"}, "initialProjectedBalance": {"value": ""}
```

```
946813.97"},"initialLimit":{"value":"-
946813.97"},"initialLimitBmf":{"value":"0"},"bslimitOperationCredit":{"value":"0"},"initialStock
PortfolioValue":{"value":"0"},"initialOptionsPortfolioValue":{"value":"0"},"initialBmfPortfolioV
alue":{"value":"0"},"totalStockPortfolioValue":{"value":"0"},"totalOptionsPortfolioValue":{"val
ue":"0"},"totalBmfPortfolioValue":{"value":"0"},"totalOnlineStockPortfolioValue":{"value":"199
4"},"totalOnlineOptionsPortfolioValue":{"value":"0"},"totalOnlineBmfPortfolioValue":{"value":
"0"},"openProfitLoss":{"value":"3372"},"closedProfitLoss":{"value":"10100.000000001"},"open
ProfitLossAux":{"value":""},"closedProfitLossAux":{"value":""},"openProfitLossDaytrade":{"valu
e":""},"closedProfitLossDaytrade":{"value":""},"initialEquity":{"value":"-
946737.89"},"totalEquity":{"value":"-
933265.8899999999"},"xbspMarginConsidered":{"value":"0"},"optionsMargin":{"value":"0"},"b
mfPositionAdjust":{"value":"0"},"dayTradeAdjustment":{"value":""},"bmfTradingAdjust":{"valu
e":"13475.000000001"},"ipoBlocked":{"value":""},"creditTed":{"value":""},"manualAdjust":{"va
lue":""},"balanceWithoutLimit":{"value":""},"optionsLeverageLimit":{"value":""},"treasuryDirec
t":{"value":"0"},"fixedIncome":{"value":"0"},"clubs":{"value":"0"},"funds":{"value":"0"},"fixedIn
comeConsidered":{"value":"0"},"clubsConsidered":{"value":"0"},"fundsConsidered":{"value":"0
"},"dividends":{"value":"76.08"},"xbspWarrantyTotal":{"value":"0"},"xbmfWarrantyTotal":{"val
ue":"0"},"xbmfMarginAvailable":{"value":"0"},"brokerageXbmf":{"value":""},"brokerageXbsp":{"
value":""},"taxes":{"value":""},"emolument":{"value":""},"accountMarginTotal":{"value":""},"a
ccountMarginUsed":{"value":""},"leverage":{"value":"1994"},"quoteOptionLeverage":{"value":
"0"},"equityLeveraged":{"value":""},"d1CreditDaytrade":{"value":""},"d1DebitDaytrade":{"valu
e":""},"d3CreditDaytrade":{"value":""},"d3DebitDaytrade":{"value":""},"d1CreditPosition":{"val
ue":""},"d1DebitPosition":{"value":""},"d3CreditPosition":{"value":""},"d3DebitPosition":{"valu
e":""},"negativePositionAmount":{"value":"0"},"negativeOptPositionAmount":{"value":"0"},"cr
editLimit":{"value":""},"buyAveragePrice":{"value":""},"sellAveragePrice":{"value":""},"warrant
yRequired":{"value":""},"warrantyDeposited":{"value":""},"profitTotalLoss":{"value":"13472.00
0000"},"balanceProjOnline":{"value":"-948810.970000"},"balanceD0":{"value":"-
596782.000000"},"balanceD1":{"value":"-348138.000000"},"balanceD2":{"value":"-
5820.000000"},"balanceD3":{"value":"-1997.000000"}}}}
```

loginDuplicated Mensagem de monitoramento de logins duplicados. Para o funcionamento, basta utilizar o mesmo **websocket** da autenticação mudando a mensagem **JSON** de envio. O qual deverá ser passado:

Token: Valor retornado na autenticação

Module: O modulo em que o serviço se encontra, no caso negotiation;

Service: O serviço o qual está requisitando, no caso loginDuplicated;

Parameters.username: De qual usuário deve ser retornado as informações;

Parameters.dispatch: Indica se quer que o servidor envie os dados ou não, valores válidos (**true**, **false**);

Segue o exemplo de envio:

```
doSend(' {  
  "token": "e07fa786-fc49-4178-a0cf-faae5a087a18",  
  "module": "negotiation",  
  "service": "loginDuplicated",  
  "parameters": {"username": "100", "dispatch": "true"}  
});
```

Resposta: