

Market Data — API WebSocket · Pacote PRO

Cedro Technologies · Documentação para o cliente · Versão 2026.1

1. Sobre este documento

Este documento descreve **exclusivamente** a API **WebSocket** do Cedro Market Data no **Pacote PRO**. Ele é autocontido: a sequência de conexão, o envelope das mensagens, todos os serviços disponíveis e os limites do pacote estão aqui.

Todo o conteúdo técnico deste documento reflete a especificação oficial da API WebSocket da Cedro. Onde um dado depende da configuração da sua conta — como o endereço do servidor —, isso está sinalizado explicitamente em vez de assumido.

2. O que é a API WebSocket

A API WebSocket estabelece uma **conexão bilateral persistente** entre a sua aplicação e o servidor de market data da Cedro (chamado **Web Feeder**), sobre TCP. Diferente de um modelo de requisição e resposta, aqui você **assina** os dados desejados uma vez e o servidor passa a **enviar as atualizações continuamente**, enquanto a conexão estiver aberta.

As mensagens trafegam em **JSON**, no padrão WebSocket — o que torna a integração direta em navegadores, Node.js, Python e praticamente qualquer stack moderna, sem necessidade de implementar um parser de protocolo proprietário.

Quando este é o canal certo

- Telas que precisam de **atualização em tempo real**: um ticker que pisca a cada negócio, uma fita que rola sozinha, uma watchlist que se atualiza sem recarregar.
- Aplicações web de cotações, plataformas de investimento, *home brokers* e aplicativos de trading.
- Cenários em que você quer **controlar a frequência de entrega** por ativo — cada assinatura aceita um parâmetro `delay` em milissegundos.

1



Av. Anselmo Alves dos Santos, 1111
Uberlândia-MG



www.cedrotech.com



contato@cedrotech.com



(34) 3239-0003

Cobertura de dados

O recurso de cotação é entregue em streaming, disponível para os mercados **B3 e Moedas**.

Mercados contratados no Pacote PRO: Bovespa e BM&F. A permissão por mercado é uma característica da sua conta — confirme com o time comercial se precisar de cobertura adicional.

3. Limites e serviços do Pacote PRO

3.1 Limite de ativos

Item	Limite contratado
Ativos assinados simultaneamente	100 ativos

O limite conta os ativos com assinatura **ativa** na sua conta em um dado momento — ou seja, aqueles para os quais você enviou uma mensagem com `subscribetype: "1"` e ainda não cancelou. Cancelar a assinatura de um ativo (`subscribetype: "0"`) libera uma vaga imediatamente.

📌 **Gerencie o seu conjunto ativo.** Com 100 vagas, a prática recomendada é assinar apenas os ativos efetivamente visíveis na tela do usuário e cancelar a assinatura ao sair dela. A seção 9.1 traz um padrão de implementação para isso.

3.2 Serviços incluídos

Serviço	service	O que entrega
Cotação	quote	Preço, variação, melhor compra/venda e mais de 100 campos, em streaming
Livro completo (Book L2)	book	Livro de ofertas oferta a oferta, com corretora
Livro agregado (Book L2)	aggregatedBook	Livro somado por nível de preço
Negócios do dia	quoteTrade	Fita de negócios em tempo real

2



Av. Anselmo Alves dos Santos, 1111
Uberlândia-MG



www.cedrotech.com



contato@cedrotech.com



(34) 3239-0003

Referência comercial: R\$ 579,90/mês, cobrados anualmente.

4. Conexão e autenticação

4.1 Endpoint

A conexão é aberta em:

```
ws://{endereço_do_servidor}/ws
```

! O **endereço do servidor** (host e porta) é definido para a sua conta e informado pela Cedro na ativação do contrato. Confirme com o suporte o endereço correto de homologação e de produção antes de iniciar a integração.

Use os eventos padrão do WebSocket — `onopen`, `onmessage`, `onerror` — e envie mensagens com `websocket.send(...)`. É obrigatório **autenticar antes de assinar qualquer dado**.

4.2 Login

Assim que a conexão abrir, envie a mensagem de login:

```
{
  "module": "login",
  "service": "authentication",
  "parameters": { "login": "<SEU_LOGIN>", "password": "<SUA_SENHA>" }
}
```

Em caso de sucesso, a resposta chega no evento `onmessage`:

```
{ "success": true, "token": "<token-da-sessão>" }
```

O campo `token` identifica a sua sessão e deve ser enviado em **todas** as requisições de dados seguintes (seção 5).

Exemplo completo em JavaScript:



```
const wsUri = "ws://{endereço_do_servidor}/ws";
const websocket = new WebSocket(wsUri);
let token = null;

websocket.onopen = () => {
  websocket.send(JSON.stringify({
    module: "login",
    service: "authentication",
    parameters: { login: process.env.CEDRO_LOGIN, password: process.env.CEDRO_PASSWORD }
  }));
};

websocket.onmessage = (evt) => {
  const msg = JSON.parse(evt.data);
  if (msg.success && msg.token) {
    token = msg.token;          // guarde o token da sessão
    // a partir daqui, assine os ativos desejados
    return;
  }
  // demais mensagens: dados de mercado (seção 6)
};

websocket.onerror = (evt) => console.error("Erro no WebSocket:", evt);
```

⚠ **O navegador nunca deve se conectar diretamente em produção com credenciais expostas.** Qualquer pessoa que inspecione o tráfego do navegador enxerga login e senha. Centralize a conexão num backend — ver seção 9.4.

4.3 Registro de FEES por conta

Para que o servidor registre o FEES (uso/cobrança) das requisições de cotação por conta, inclua o campo `account` nos parâmetros do login:



```
{
  "module": "login",
  "service": "authentication",
  "parameters": { "login": "<SEU_LOGIN>", "password": "<SUA_SENHA>", "account": "<CONTA>" }
}
```

Ao pedir a difusão de um ativo, o servidor então grava a conta e o mercado ao qual aquele ativo pertence. Isso é relevante se a sua integração atende múltiplos clientes ou contas finais e você precisa de relatório de uso segmentado.

5. Envelope das mensagens

Toda mensagem trocada é um **objeto JSON**. O login usa `module / service / parameters` (seção 4.2); as requisições de dados acrescentam `token` e `parameterGet` :

Campo	Descrição
<code>token</code>	Token da sessão, retornado no login — obrigatório em toda requisição de dados
<code>module</code>	Módulo alvo. Para dados de mercado: <code>quotes</code> . Para login: <code>login</code>
<code>service</code>	Serviço desejado — ver a tabela da seção 3.2
<code>parameterGet</code>	Ativo alvo, ex.: <code>petr4</code>
<code>parameters</code>	Objeto com os parâmetros específicos do serviço (detalhados na seção 6)

Parâmetros comuns a todos os serviços de dados:

Parâmetro	Descrição
<code>subsribetype</code>	"1" = iniciar a assinatura · "0" = parar
<code>delay</code>	Intervalo mínimo entre envios, em milissegundos



❑ O parâmetro `delay` é a sua principal ferramenta de controle de volume. Um valor baixo entrega mais atualizações e consome mais banda e CPU do cliente; um valor mais alto agrupa as mudanças. Para uma watchlist de acompanhamento, valores na casa de 1000 ms costumam ser suficientes; para um ticker sensível a cada tick, valores menores fazem sentido.

ⓘ **Atenção à grafia.** O parâmetro se chama `subsribetype` — exatamente como escrito na especificação da API. Corrigir a grafia para `subscribetype` faz a requisição não ser reconhecida.

6. Referência de serviços

Todos os serviços abaixo ficam sob `module: "quotes"` e exigem o `token` obtido no login.

6.1 Cotação — serviço `quote`

Assina a cotação de um ativo em tempo real.

```
{
  "token": "<token-da-sessão>",
  "module": "quotes",
  "service": "quote",
  "parameterGet": "petr4",
  "parameters": { "subsribetype": "1", "filter": "2,3,4", "delay": "400" }
}
```

Parâmetros:

Parâmetro	Descrição
<code>subsribetype</code>	"1" = iniciar a cotação · "0" = parar
<code>filter</code>	Lista separada por vírgula dos índices desejados (ver seção 8.5). Ex.: 2 =último preço, 3 =melhor compra, 4 =melhor venda
<code>delay</code>	Intervalo mínimo entre envios, em milissegundos

Resposta (`type: "QuoteType"`) — pares `índice → valor`:



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

⚠ **Decimal com vírgula.** Os valores numéricos chegam como **string com vírgula decimal** ("13,32"), não como número JSON. Converta antes de usar em cálculos — é um dos pontos que mais gera erro silencioso na integração.

📌 **Use o filter.** Pedir apenas os índices que a sua tela realmente exibe reduz drasticamente o volume de mensagens. Uma watchlist típica precisa de algo como "2,3,4,21,9" (último preço, compra, venda, variação e volume) — não dos 100+ campos disponíveis.

6.2 Livro completo — serviço `book`

Assina o livro de ofertas completo (oferta a oferta) de um ativo.

```
{
  "token": "<token-da-sessão>",
  "module": "quotes",
  "service": "book",
  "parameterGet": "petr4",
  "parameters": { "subsribetype": "1", "filter": "10", "delay": "5000" }
}
```

Parâmetros: `filter` = número de linhas do book; `subsribetype` = "1" / "0"; `delay` em milissegundos.

Resposta (`type`: "BookSnapshotType") — objeto `book` com `S` (símbolo) e as listas `A` e `B` :

```
{ "book": { "S": "petr4",
  "B": [ { "Q": 2900, "P": 12.83, "C": 3, "FQ": "2.9K", "T": "L" } ],
  "A": [ { "Q": 100, "P": 12.82, "C": 16, "FQ": "100", "T": "L" } ] },
  "type": "BookSnapshotType", "parameter": "petr4" }
```

Campo	Significado
S	Símbolo do ativo

7



Av. Anselmo Alves dos Santos, 1111
Uberlândia-MG



www.cedrotech.com



contato@cedrotech.com



(34) 3239-0003

Campo	Significado
Q	Quantidade da oferta
P	Preço
C	Corretora
FQ	Quantidade formatada (ex.: 2.9K)
T	Tipo da oferta: L =Limitada, O =preço de Abertura, X =melhor preço, M =preço de Mercado

⚠ **Atenção à convenção de letras.** Conforme a especificação oficial da API, **A = Compra (bids)** e **B = Venda (asks)** — invertido em relação ao que as iniciais em português sugeririam. No exemplo acima, a melhor compra (A) é 12,82 e a melhor venda (B) é 12,83. Trocar as colunas é o erro clássico na primeira implementação do book.

6.3 Livro agregado — serviço `aggregatedBook`

Assina o livro de ofertas agregado por nível de preço — o formato mais indicado para renderizar um DOM de tela.

```
{
  "token": "<token-da-sessão>",
  "module": "quotes",
  "service": "aggregatedBook",
  "parameterGet": "petr4",
  "parameters": { "subsribetype": "1", "delay": "5000" }
}
```

Sem o parâmetro `filter`, o serviço retorna por padrão **5 linhas** de compra e venda, agregadas por nível de preço.

Resposta (`type: "AggregatedBookType"`): mesmo formato do serviço `book`, mas cada linha traz apenas `Q`, `P` e `FQ` — sem `C` e `T`, já que a agregação é por nível de preço e não por oferta individual. A mesma convenção `A` = compra / `B` = venda se aplica.



6.4 Negócios do dia — serviço `quoteTrade`

Assina a fita de negócios do dia de um ativo.

```
{
  "token": "<token-da-sessão>",
  "module": "quotes",
  "service": "quoteTrade",
  "parameterGet": "petr4",
  "parameters": { "subsribetype": "1", "quantidade": "2", "delay": "600" }
}
```

Parâmetros: `quantidade` = número de negócios por linha; `subsribetype` = "1" / "0"; `delay` em milissegundos.

Resposta (`type`: "BusinessBookType") — objeto `quoteTrade` com `M` (ativo) e `L` (lista de negócios):

```
{ "quoteTrade": { "M": "PETR4", "L": [
  { "QT": 1400, "P": 27.47, "PCB": 115, "T": "Sep 30, 2019 4:11:03 PM", "PCL": 308, "CDA":
    "V" } ] },
  "type": "BusinessBookType", "parameter": "PETR4" }
```

Campo	Significado
QT	Quantidade do negócio
P	Preço
PCB	Corretora compradora
PCL	Corretora vendedora
T	Data/hora do negócio
CDA	Agressor (A =comprador, V =vendedor) — ver seção 8.3



7. Notas técnicas

Pontos que dependem da configuração da sua conta ou de confirmação com o suporte da Cedro no momento da integração:

- **Host e porta** do endpoint `/ws`, em homologação e em produção, são definidos por conta.
- **Permissão por mercado** (RealTime × Delay) é uma característica da sua conta — ver seção 9.5.
- A semântica dos campos `PCB`, `PCL` e `CDA` do serviço `quoteTrade` foi documentada a partir de exemplos de resposta reais. Antes de apoiar uma decisão crítica de negócio nesses campos, confirme a interpretação com o suporte.

8. Referência de códigos (enums)

8.1 Códigos de mercado

Usado para interpretar o índice `44` da cotação.

Código	Mercado	Observação
1	Bovespa	Ações, units, ETFs à vista (B3)
2	Dow Jones	Índice/ativos norte-americanos
3	BM&F	Futuros e derivativos B3 (WIN, WDO, DI, etc.)
4	Índices	Índices em geral
5	Money	Câmbio/indicadores monetários
7	Forex	Mercado de câmbio
8	Indicators	Indicadores econômicos
10	Nyse	New York Stock Exchange
12	Nasdaq	Nasdaq
13	CFD	Contratos por diferença
30	Bitcoin	Criptomoedas
44	Datagro	Commodities/agro (provedor Datagro)

10



Av. Anselmo Alves dos Santos, 1111
Uberlândia-MG



www.cedrotech.com



contato@cedrotech.com



(34) 3239-0003

Código	Mercado	Observação
45	INews	Notícias
52	Amex	American Stock Exchange
64	Tesouro Direto	Títulos públicos
76	Nyse FMV	Fair Market Value — NYSE
77	Nasdaq FMV	Fair Market Value — Nasdaq
78	Amex FMV	Fair Market Value — Amex

8.2 Tipos de ativo

Usado para interpretar o índice 45 da cotação.

Código	Tipo	Exemplo
1	Ativo à vista	PETR4, VALE3
2	Opção	PETRJ93
3	Índice	IBOV
4	Commodity	Café, boi
5	Moeda	DOL
6	Termo	Ações a termo
7	Futuro	WINFUT, WDOFUT
8	Leilão	—
9	Bônus	—
10	Fracionário	PETR4F
13	ETF	BOVA11
20	Corp	Renda fixa corporativa
21	SECLOAN (aluguel BTB)	Banco de Títulos
22	Tesouro Direto	Tesouro Selic 2029



8.3 Agressor do negócio

Usado para interpretar o campo `CDA` do serviço `quoteTrade`.

Valor	Significado
A	Comprador (agressor de compra)
V	Vendedor (agressor de venda)
0	Não direto
1	Direto
2	RLP (<i>Retail Liquidity Provider</i>)
3	RFQ (<i>Request for Quote</i>)
4	Midpoint Trade
5	Opening Price (leilão)
6	Point in Time Auction
I	Indefinido

□ Na prática, A / V é o valor mais frequente — costuma colorir a linha da fita (verde quando o comprador agrediu, vermelho quando foi o vendedor).

8.4 Fase e status do instrumento

Referência para os índices de status da cotação (campos 67, 84, 88 e 96).

Campo	Valor	Significado
Status do instrumento (67)	101	Normal
Status do instrumento (67)	102	Leilão
Status do instrumento (67)	105	Suspenso
Status do instrumento (67)	118	Congelado
Status do ativo (84)	0	Normal
Status do ativo (84)	1	Congelado



Campo	Valor	Significado
Status do ativo (84)	2	Suspenso
Status do ativo (84)	3	Leilão
Status do ativo (84)	4	Inibido
Fase do grupo (88)	P	Pré-abertura
Fase do grupo (88)	A	Abertura (sessão normal)
Fase do grupo (88)	N	Fechamento
Fase do grupo (88)	F	Final
Fase do grupo (88)	N0	Fechado
Tipo de mercado (96)	RT	RealTime
Tipo de mercado (96)	D	Delay
Tipo de mercado (96)	EOD	End of Day

8.5 Tabela de índices da cotação

Os índices usados no parâmetro `filter` e devolvidos em `values` (seção 6.1) vão de 0 a 138:

Índice	Significado	Tipo
0	Horário da última modificação	HHMMSS
1	Data da última modificação	YYYYMMDD
2	Preço do último negócio	Float
3	Melhor oferta de compra	Float
4	Melhor oferta de venda	Float
5	Horário do último negócio	HHMMSS
6	Quantidade do negócio atual	Inteiro
7	Quantidade do último negócio	Inteiro
8	Quantidade de negócios realizados	Inteiro
9	Volume acumulado dos negócios (papéis)	Inteiro



Índice	Significado	Tipo
10	Volume financeiro dos negócios	Float
11	Maior preço do dia	Float
12	Menor preço do dia	Float
13	Preço de fechamento do dia anterior	Float
14	Preço de abertura	Float
15	Horário da melhor oferta de compra	HHMMSS
16	Horário da melhor oferta de venda	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	Float
20	Volume da melhor oferta de venda	Inteiro
21	Variação (%)	Float
36	Preço de fechamento da última semana	Float
37	Preço de fechamento do último mês	Float
38	Preço de fechamento do último ano	Float
39	Preço de abertura do dia anterior	Float
40	Maior preço do dia anterior	Float
41	Menor preço do dia anterior	Float
42	Média	Float
43	VHDDaily	Float
44	Código do mercado (seção 8.1)	Inteiro
45	Código do tipo do ativo (seção 8.2)	Inteiro
46	Lote padrão	Inteiro
47	Descrição do ativo	String(10)
48	Nome de classificação	String



Índice	Significado	Tipo
49	Forma de cotação	Inteiro
50	Intraday Date	YYYYMMDDHHMMSS
51	LastTrade Date	YYYYMMDDHHMMSS
52	Descrição abreviada do ativo	String
53	Identificador do negócio cancelado	String
54	Data do último negócio	YYYYMMDD
56	Sentido das ofertas não atendidas ao preço de abertura	Char
57	Quantidade não atendida ao preço de abertura	Inteiro
58	Horário programado para abertura do papel	HHMMSS
59	Horário reprogramado para abertura do papel	HHMMSS
60	Código da corretora da melhor oferta de compra	Inteiro
61	Código da corretora da melhor oferta de venda	Inteiro
62	Código da corretora da última compra	Inteiro
63	Código da corretora da última venda	Inteiro
64	Data do vencimento (opções)	YYYYMMDD
65	Expirado (flag)	Char
66	Número total de papéis	Inteiro
67	Status do instrumento (seção 8.4)	Inteiro
72	Tipo da opção (A =Americana, E =Europeia, 0 =não existe)	Char
74	Direção da opção (P =venda/put, C =compra/call)	Char
81	Símbolo do ativo pai (para opções)	String
82	Preço teórico de abertura	Float
83	Quantidade teórica	Inteiro
84	Status do ativo (seção 8.4)	Inteiro
85	Preço de exercício (opções)	Float



Índice	Significado	Tipo
86	Diff (preço atual – anterior)	Float
87	Data do "previous"	YYYYMMDD
88	Fase do grupo do ativo (seção 8.4)	Char
89	Média do dia anterior	Float
90	Intervalo de margem (mercado BTC)	Float
94	Volume médio nos últimos 20 dias	Float
95	Market capitalization	Float
96	Tipo de mercado — RT/D/EOD (seção 8.4)	String(2/3)
97	Variação do fechamento em uma semana	Float
98	Variação do fechamento em um mês	Float
99	Variação do fechamento em um ano	Float
100	Quantidade de contratos em aberto	Inteiro
101	Número de dias úteis até o vencimento	Inteiro
102	Número de dias corridos até o vencimento	Inteiro
103	Ajuste do dia	Float
104	Ajuste do dia anterior	Float
105	SecurityId	String
106	TickDirection	String(2)
107	Tunnel Upper Limit (Hard Limit)	Float
108	Tunnel Lower Limit (Hard Limit)	Float
109	TradingPhase	String(2)
110	TickSize	Float
111	Volume mínimo de negociação do instrumento	Inteiro
112	Intervalo mínimo para incrementos de preço	Float
113	Quantidade mínima em uma oferta	Inteiro



Índice	Significado	Tipo
114	Quantidade máxima em uma oferta	Inteiro
115	Número único de identificação do instrumento	Inteiro
116	Moeda utilizada no preço (BRL/EUR/USD)	String
117	SecurityType (1003=ON, 1107=Non tradable Index, 1124=Fixed Income ETF)	Inteiro
118	Security Sub Type (FUT, OPT, SPOT, SOPT, FOPT, DTERM)	String(32)
119	Produto associado ao instrumento	Inteiro
120	Mês e ano de vencimento	YYYYMM
121	Preço de exercício da opção	Float
122	Moeda do preço de exercício da opção	String
123	Multiplicador do contrato	Inteiro
124	Código do tipo de preço do instrumento	Inteiro
125	Horário em que o instrumento não é mais negociável	HHMMSSmmm
126	Grupo ao qual o ativo pertence	String
127	Ajuste atual em taxa	Double
128	Ajuste anterior em taxa	Double
129	Data do ajuste atual em taxa	YYYYMMDD
130	Número de saques até a data de vencimento	Inteiro
134	Variação do volume da hora vs. média (20 dias)	Float
135	Variação do volume até a hora vs. média (20 dias)	Float
136	Código do setor do ativo	Inteiro
137	Código do subsetor do ativo	Inteiro
138	Código do segmento do ativo (-1 se não existir)	Inteiro

i Em caso de dúvida sobre um índice específico não coberto por esta tabela, confirme com o suporte da Cedro.



9. Guias práticos

9.1 Gerenciar o limite de 100 ativos

O padrão que funciona bem é tratar as assinaturas como um **conjunto controlado por um único componente** do seu backend:

1. Mantenha um `Set` com os ativos atualmente assinados.
2. Antes de assinar um novo ativo, verifique se o conjunto já atingiu **100**.
3. Se atingiu, cancele a assinatura do ativo menos relevante (por exemplo, o que saiu da tela há mais tempo) enviando a mesma mensagem com `subsribetype: "0"`.
4. Só então envie a nova assinatura.

```
const assinados = new Set();
const LIMITE = 100;

function assinar(ativo, filtro) {
  if (assinados.has(ativo)) return;
  if (assinados.size >= LIMITE) {
    const antigo = assinados.values().next().value; // política simples: FIFO
    cancelar(antigo);
  }
  websocket.send(JSON.stringify({
    token, module: "quotes", service: "quote", parameterGet: ativo,
    parameters: { subsribetype: "1", filter: filtro, delay: "1000" }
  }));
  assinados.add(ativo);
}

function cancelar(ativo) {
  websocket.send(JSON.stringify({
    token, module: "quotes", service: "quote", parameterGet: ativo,
    parameters: { subsribetype: "0" }
  }));
  assinados.delete(ativo);
}
```



9.2 Montar uma watchlist

1. Conecte e autentique (seção 4.2), guardando o `token`.
2. Para cada ativo, envie o serviço `quote` com `subsribetype: "1"` e o `filter` com os índices que a tela usa — por exemplo, `"2,3,4,21,9"` (último preço, compra, venda, variação, volume).
3. Cada mensagem em `onmessage` traz um objeto `values` com os índices pedidos — atualize a linha correspondente da lista. Lembre-se de converter a vírgula decimal.
4. Para parar, reenvie a mesma mensagem com `subsribetype: "0"`.

9.3 Renderizar um book (livro de ofertas)

1. Conecte e autentique.
2. Envie o serviço `aggregatedBook` — o mais indicado para um DOM de tela, pois já vem agregado por nível de preço.
3. A resposta traz `book.A` (**compra**) e `book.B` (**venda**) — atenção à convenção invertida descrita na seção 6.2.
4. Renderize `A` na coluna esquerda e `B` na direita, ordenando por preço.

9.4 Fita de negócios

1. Conecte e autentique.
2. Envie o serviço `quoteTrade` com `subsribetype: "1"` e a `quantidade` de negócios desejada por atualização.
3. Cada mensagem traz `quoteTrade.L`, a lista dos negócios mais recentes; use `CDA` para colorir a linha (`A` = comprador → verde, `V` = vendedor → vermelho).

9.5 Centralizar a conexão (boas práticas)

Não conecte o navegador diretamente ao Web Feeder em produção com credenciais no cliente. Centralize a conexão num **backend**, que:

- autentica **uma vez** e mantém o `token` da sessão;
- concentra o controle do limite de ativos num único lugar (seção 9.1);
- retransmite os dados para as telas dos usuários finais por um canal próprio da sua aplicação — por exemplo, um WebSocket seu ou *Server-Sent Events*.



Além da segurança, esse desenho evita multiplicar conexões e assinaturas por usuário ou por aba do navegador — o que consumiria o limite de ativos muito mais rápido do que o necessário.

Reconexão: trate `onerror` e o fechamento da conexão com uma política de reconexão com *backoff* progressivo. Ao reconectar, refaça o login, obtenha um novo `token` e **reenvie todas as assinaturas** — elas não sobrevivem à queda da conexão.

9.6 Dados atrasados (Delay) e tempo real

- Contas de homologação/trial normalmente vêm configuradas em **Delay** (tipicamente ~15 minutos). A permissão por mercado é uma característica da conta — confirme com o suporte.
- **Fora do horário de pregão**, o dado pode ser da última sessão — isso é esperado, não é erro.
- **Durante o pregão**, mesmo em Delay os dados continuam chegando — é o melhor horário para validar a integração.
- O índice `96` da cotação informa o regime do dado naquele momento: `RT`, `D` ou `EOD` (seção 8.4).

A migração para RealTime contínuo depende da permissão contratada — consulte o time comercial.

10. Glossário

- **Ativo (*symbol*)**: instrumento negociável (ex.: `PETR4`, `WINFUT`, `BOVA11`).
- **Cotação**: conjunto de preços e dados de negociação de um ativo.
- **Book / Livro de ofertas / DOM**: lista de ofertas de compra e venda por nível de preço.
- **Book agregado**: livro somado por nível de preço (o formato do serviço `aggregatedBook`).
- **Negócio / Trade**: uma execução (preço × quantidade) entre um comprador e um vendedor.
- **Fita**: a sequência cronológica de negócios realizados (serviço `quoteTrade`).
- **Agressor**: o lado que atacou o preço e fechou o negócio.
- **Token de sessão**: identificador retornado no login, obrigatório em toda requisição de dados.



- **Assinar / cancelar assinatura:** começar (`subsribetype: "1"`) ou parar (`subsribetype: "0"`) de receber atualizações de um serviço.
- **Snapshot:** o estado inicial de um dado, antes das atualizações contínuas.
- **Índice (da cotação):** o número que identifica cada campo no objeto `values` — ver seção 8.5.
- **Delay / RealTime / EOD:** dado atrasado / em tempo real / de fim de dia.
- **FEES:** registro de uso das requisições de cotação por conta, ativado pelo parâmetro `account` no login.
- **Web Feeder:** o servidor de market data da Cedro que atende este canal.

11. Suporte e credenciais

As credenciais de acesso (`login` e `password`) e o **endereço do servidor** são fornecidos pela Cedro na ativação do contrato. Elas são pessoais e intransferíveis: guarde-as em variáveis de ambiente ou em um cofre de segredos, nunca no código-fonte ou no front-end.

Canal	Contato
E-mail	comercial@cedrotech.com
Telefone	(34) 3239-0003
Site	www.cedrotech.com

Para dúvidas técnicas sobre este documento ou sobre a integração, use o mesmo canal de contato.

