

## Considerações Gerais

---

A B3 disponibiliza APIs para permitir, de forma ágil, interação com o mercado.

APIs (Application Programming Interface) possibilitam aos participantes desenvolverem ferramentas para automatizar processos de integrações com aplicações mobile, sites, machine-to-machine, além de se colocar como alternativa aos padrões de integração existentes.

Construídas no padrão REST (Representational state transfer), consistem em um conjunto de URL's (Uniform Resource Locator) que viabilizam a realização de consultas, cadastros, atualizações e remoções executadas por sistemas.

As informações aqui presentes são de ordem geral e devem ser aplicadas a toda e qualquer API publicada pela B3. Cada sistema / negócio publicará documento com orientações complementares para atender as especificidades de uso de suas APIs.

As APIs da B3 são desenvolvidas no padrão REST, gerando uma resposta para cada requisição recebida de seus participantes. Algumas podem suportar requisições simultâneas, respeitando as configurações de Throttle configuradas, para atender simultaneamente todos os clientes.

Devido as características supracitadas, é necessário que as aplicações implementadas pelos clientes controlem as respostas recebidas da API, para garantir que suas demandas foram atendidas. Este alerta se faz necessário especialmente em casos de dependência entre requisições, para evitar que uma chamada que depende do resultado de outra seja processada antes desta primeira.

## Fluxos de Negócio

---

As APIs podem ser consumidas em uma ordem específica para atender a um fluxo de negócio. A indicação sobre a sequência de uso estará disponível, se adequado, na documentação específica.

## Padrões de nomenclatura

---

Nome da API:

Formato: [Entidade/Negócio][Complemento]

O Assunto Canônico é assunto ou processo de negócio do escopo da API. Deve ser escrito no idioma inglês e sem preposições.

**Exemplo: Account, Party, Derivative, Equity, SubAccount.**

O complemento especifica ou detalha o assunto canônico, de acordo com a necessidade.

**Exemplo: PartySynchronize**

A primeira letra de cada palavra deve ser maiúscula e as demais minúsculas

## Padrões de Endpoints

---

O endpoint deve conter a seguinte estrutura:

**Http://<sisistema>.b3.com.br/<identificador>/<versão>/<nomeAPI>/<parametros>**

Elemento	Descrição
<sisistema>	Nome do sistema que expõe a API. ex.: <a href="http://gps.b3.com.br/apis/v1.0/index/">http://gps.b3.com.br/apis/v1.0/index/</a>
<identificador>	Identificador utilizado opcionalmente pelo sistema provedor ex.: <a href="http://gps.b3.com.br/apis/v1.0/index/">http://gps.b3.com.br/apis/v1.0/index/</a>
<versao>	Indica a versão da API ex.: <a href="http://gps.b3.com.br/apis/v1.0/index/">http://gps.b3.com.br/apis/v1.0/index/</a>  O versionamento deve refletir mudanças no comportamento da API, seja por implementação, mudança de funcionalidade ou manutenções menores como bug-fix por exemplo.
<nomeAPI>	Nome da api exposta ex.: <a href="http://gps.b3.com.br/apis/v1.0/index">http://gps.b3.com.br/apis/v1.0/index</a>
<parâmetros>	Parâmetros adicionais de composição ou de filtro do recurso ex.: <a href="http://gps.b3.com.br/apis/v1.0/index/IPCA">http://gps.b3.com.br/apis/v1.0/index/IPCA</a>  os parâmetros suportam valores de filtragem e podem, neste caso serem compostos de acordo com o exemplo abaixo:  ex.: <a href="http://gps.b3.com.br/apis/v1.0/index/IPCA/1">http://gps.b3.com.br/apis/v1.0/index/IPCA/1</a> => 1 é filtro que indica que o retorno será o valor acumulado do IPCA em 12 meses

## Padrões de uso

---

Entre as formas de uso de APIs, a B3 oferece a possibilidade de:

- Compactação via GZIP (<http://www.gzip.org/>)
  - A compactação, quando disponível será informada na documentação do sistema que disponibiliza o recurso.
- Múltiplos formatos de dados
  - As APIs da B3, trafegam, por padrão, dados no formato Json. Outros formatos quando disponíveis, (XML, CSV, etc.), serão informados na documentação específica.
- VPN / Internet
  - As APIs da B3 podem ser disponibilizadas em meio público (Internet) ou com acesso restrito (VPN). Os catálogos publicados são segregados para atender estas condições.
- Segurança
  - Requisitos de segurança serão aplicados considerando a natureza da informação entregue pela API dentro de um fluxo de negócio. O tópico “Segurança” deste documento oferece informações adicionais.
- Throttling
  - Para otimizar a utilização das APIs e considerando as necessidades dos clientes, as APIs da B3 implementam o recurso de Throttling, que consiste em limitar a taxa de requisições que um cliente pode enviar aos endpoint's. Os valores configurados para este controle de requisições ficam a critério da [B]<sup>3</sup>.  
Esta funcionalidade fará a medição por serviço, por usuário logado. As requisições serão contabilizadas globalmente, ou seja, levando em consideração todas as requisições por todas as conexões efetuadas.

## Política de versionamento

---

As APIs da B3 adotam um número decimal (X.X) para identificar a versão.

A parte inteira é sequencial e indica uma alteração de comportamento ou adição/subtração de funcionalidades.

Ex: <http://gps.b3.com.br/apis/v1.0/index>

A parte decimal é sequencial e indica alterações de menor impacto como correções de ordem de informações, obrigatoriedade de atributos e outros ajustes que não causem impacto significativo nas funcionalidades da API

Ex: <http://gps.b3.com.br/apis/v1.1/index>

## Segurança

---

A B3 implementará os mecanismos de autenticação por intermédio do OAuth2 em conjunto com OpenId.

**OAuth2** é um framework de autorização aberto, poderoso e flexível que pode ser usado para proteção de aplicações e APIs REST. Com OAuth2 é possível prover um mecanismo padronizado para Identity Management, em que todos os componentes do sistema possam interagir de um modo seguro, usualmente onde uma aplicação cliente necessita de acesso a um recurso protegido, agindo no lugar do usuário.

**OpenID Connect** é uma camada de identidade simples criada sobre o protocolo OAuth 2.0. OAuth 2.0 define os mecanismos para obter e usar tokens de acesso para acessar recursos protegidos, mas eles não definem os métodos padrão para fornecer informações de identidade. O OpenID Connect implementa a autenticação como uma extensão para o processo de autorização do OAuth 2.0. Ele fornece informações sobre o usuário final na forma de um id\_token que verifica a identidade do usuário e fornece informações básicas de perfil sobre o usuário.

Informações adicionais sobre as tecnologias podem ser obtidas em:

- <https://oauth.net/2/>
- <http://openid.net/connect/>
- <https://hueniverse.com/>

## Verbos

---

As APIs utilizam, por padrão os verbos **GET, POST, PUT e DELETE** como definidos na RFC 7231 (<https://tools.ietf.org/html/rfc7231>)

As exceções serão tratadas na documentação de cada API, quando couber.

## Códigos http de retorno utilizados

---

As mensagens de erro obedecerão ao definido pelas RFCs 6585, 7231 e 7235 e serão retornadas pelo servidor de aplicação ou pela própria aplicação no caso dos códigos 401 e 404.

O seguinte subset deve ser adotado:

Código HTTP	Descrição
200	Success
301	Moved Permanently
400	Bad Request

## APIs B3

401	Unauthorized
404	Not Found
408	Request Timeout
415	Unsupported media type
421	Misdirected Request
422	Unprocessable Entity
429	Too Many Requests
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

### Tratamento de erros

---

As APIs incluirão, por padrão, área para registro do resultado das requisições.

- Um pedido de inclusão pode, por exemplo, retornar um Id
- Uma consulta paginada pode receber o endereço da próxima página
- Uma alteração pode não ser concluída por conta de uma regra de negócio.

Para atender estas e outras situações análogas, o componente BusinessStatus será retornado com os códigos necessários para análise do requisitante.

Detalhes sobre o comportamento do componente podem ser encontrados na documentação técnica das APIs (Swagger)

### Exemplos de chamadas

---

Os exemplos abaixo baseados em cURL não devem ser entendidos como recomendação de uso de tecnologia vez que a implementação de clientes pode ser feita em uma gama extensa de linguagens.

Chamada utilizando GET

```
curl -i \  
-H "Accept:application/json" \  
-X GET https://{line5.0-domain}/api/v1.0/token/authorization
```

Chamada utilizando DELETE

```
curl -i \  
-H "Accept:application/json" \  
-X DELETE https://{line5.0-domain}/api/v1.0/account/<acctId>/lmt
```

Chamada utilizando POST

```
curl -v \  
-H "Authorization: Bearer <Access-Token>" \  
-X POST https://{line5.0-domain}/api/v1.0/account/<acctId>/lmt \  
-d '{  
  "sfd" : "12345",  
  "sdp" : "23456",  
  "rmkt" : "34567",  
  "irmkt" : "45678"  
'
```