



**API - IMERCADO**

**Alocação Renda Fixa Balcão**

*Sumário*

<b>1. Histórico de Versões</b>	<b>4</b>
<b>2. Introdução</b>	<b>5</b>
<b>3. Verbos HTTP - API – iMercado</b>	<b>6</b>
<b>4. Endpoint</b>	<b>6</b>
<b>5. Endereços da API do iMercado</b>	<b>6</b>
<b>6. Swagger</b>	<b>6</b>
<b>7. Funcionamento Envio (POST) e Recepção (GET)</b>	<b>7</b>
7.1 Considerações Gerais	7
7.2 Estrutura geral das mensagens	7
7.3 Sobre a seção “TechnicalInformation”	7
7.4 Envio (POST)	8
7.5 Recepção (GET)	9
7.6 Parâmetro de Consulta (GET)	10
<b>8. API – iMercado Renda Fixa</b>	<b>10</b>
8.1 OTCIncomeTradeCaptured	10
8.2 OTCIncomeAllocation	10
8.3 OTCIncomeAllocationStatus	10
8.4 OTCIncomeAllocationAcceptanceOrRejection	11
8.5 OTCIncomeTradeCancellation	11
8.6 OTCIncomeTradeCancellationApproval	11
<b>9. Fluxos de Negócio</b>	<b>11</b>
<b>10. Estrutura das APIs</b>	<b>11</b>
<b>11. Segurança</b>	<b>11</b>
11.1 Mutual SSL (Two Way SSL)	11
11.2 Pinagem de Certificado	12
11.3 Token JWT no padrão OAuth 2.0	12



### 1. Histórico de Versões

Data	Versão	Descrição
07/05/2021	1.0	Versão inicial
07/07/2021	1.1	Correções no índice 11
19/08/2021	1.2	Inclusão no índice 11.3 - Category_ID.

## 2. INTRODUÇÃO

Este documento descreve o conjunto de definições de APIs desenvolvidas pela B3 que serão utilizadas nos fluxos de captura e alocação de negócios de Balcão no iMercado. Esse documento trata-se especificamente da interface de comunicação API Web (REST) via internet.

As APIs (Application Programming Interface) WEB possibilitam aos participantes do iMercado desenvolverem aplicações para automatizar processos de comunicação, utilizando o catálogo da mensageria iMercado, com aplicações machine-to-machine, para participantes que não possuem acesso as redes RCCF e RCB.

Construídas no padrão REST (Representational state transfer), consistem em um conjunto de URL's (Uniform Resource Locator) que viabilizam a realização de coleta e envio de mensagens executadas por sistemas. As informações aqui presentes são específicas do processo Captura e Alocação de Renda Fixa e devem ser aplicadas a toda e qualquer API desde assunto.

### 3. Verbos HTTP - API – iMercado

Para as APIs do iMercado serão utilizados somente os verbos GET (para recuperar mensagens) e POST (Para enviar mensagens). Os verbos HTTP DELETE, PUT e HEAD não são suportados.

### 4. Endpoint

Os endpoints são por código de mensagem de acordo com o catálogo do iMercado. Seguem abaixo as URIs de cada API.

OTCIncomeTradeCaptured	URI
OTCIncomeTradeCaptured	trade
OTCIncomeAllocation	allocation
OTCIncomeAllocation	allocation-status
OTCIncomeAllocationAcceptanceOrRejection	allocation-approval
OTCIncomeTradeCancellation	trade-cancellation
OTCIncomeTradeCancellationApproval	trade-cancellation-approval

URI base (todas as URLs iniciam com): /api/otc-fixed-income-allocation/v1/

Exemplo de uma URL completa, para o ambiente de produção, referente a mensagem "OTCIncomeTradeCaptured":

<https://api-mercado.b3.com.br/api/otc-fixed-income-allocation/v1/trade>

### 5. Endereços da API do iMercado

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

Seguem os endereços para conexão referente cada ambiente:

- Certificação: <https://api-mercado-cert.b3.com.br>
- Produção: <https://api-mercado.b3.com.br>

### 6. Swagger

A documentação técnica detalhadas das APIs do iMercado estão disponíveis no Swagger em:

[http://clientes.b3.com.br/pt\\_br/produtos-e-servicos/sobre/solucao-de-pos-negociacao-fluxo-iii-titulos-publicos.htm](http://clientes.b3.com.br/pt_br/produtos-e-servicos/sobre/solucao-de-pos-negociacao-fluxo-iii-titulos-publicos.htm)

Ver Item: ["Swagger iMercado – Alocação Renda Fixa Balcão"](#)

## 7. Funcionamento Envio (POST) e Recepção (GET)

### 7.1 Considerações Gerais

O sistema de mensageria B3 iMercado mantém um controle das mensagens enviadas e recebidas dos participantes, no qual cada mensagem recebe um número de controle chamado de SeqNum (SequenceNumber). O controle do SeqNum (SequenceNumber) é feito por endpoint. Este é reiniciado (definido para 1) diariamente, pois a messageira iMercado é sempre diária, portanto não será possível recuperar mensagens de dias diferentes de D+0.

### 7.2 Estrutura geral das mensagens

As mensagens são trafegadas nos endpoints das APIs do iMercado em formato JSON usando o padrão baseado na ISO 20022.

Cada mensagem contém a seguinte estrutura:

```
{
  "TechnicalInformation": {
  },
  "ApplicationHeader": {
  },
  "<Dados Específicos de Negócio>": {
  }
}
```

Onde:

- TechnicalInformation= Contém informações técnicas referentes a transmissão da mensagem. Nessa seção serão indicados valores como: Número de sequência (somente enviado pela B3), indicador de possibilidade de duplicidade e sessão FIX da contraparte.
- ApplicationHeader= Contém informações que facilitam o processamento da mensagem como: Identificador único, código da mensagem, data/hora de referência, participante de origem e destino etc.
- <Dados Específicos de Negócio>= Nessa seção haverá dados referentes ao negócio/transação de acordo com o catálogo do iMercado, como por exemplo: TradeLegNotification, TradeNotificationResponse entre outros

### 7.3 Sobre a seção “TechnicalInformation”

Segue abaixo a lista dos campos contidos na seção “TechnicalInformation” e com as instruções sobre o preenchimento:

- SeqNum: Esse campo contém o número de sequência da mensagem atribuído pela B3. Somente a B3 preenche esse campo.
- PossDupFlag: Esse campo contém o indicador de possibilidade de duplicidade da mensagem. A B3 e o participante preenchem esse campo. Se a mensagem está sendo enviada pela primeira vez é

esperado que o valor desse campo contenha o valor "false". Em cenários de recuperação de falha o sistema remetente pode não ser capaz de reconhecer se a mensagem em processamento foi enviada e pode efetuar uma nova tentativa, nesse caso, as próximas tentativas de envio devem ser realizadas alterando o valor desse campo para "true".

- **OnBehalfOfCompId:** Esse campo contém o nome de sessão FIX do participante de origem. Somente a B3 preenche esse campo. O participante destinatário deve guardar esse dado para informá-lo, por meio do campo "DeliverToCompId", nas próximas mensagens do mesmo fluxo de negócio geradas em resposta.
- **DeliverToCompId:** Esse campo contém o nome de sessão FIX do participante de destino que a mensagem deverá ser entregue. Somente o participante remetente preenche esse campo. O participante (de origem) deve preencher esse campo com o valor obtido em uma mensagem anterior, do mesmo fluxo de negócio, do campo "OnBehalfOfCompId".
- **GatewayName:** Esse campo contém o nome do FIX Gateway que hospeda a sessão FIX informada nos campos OnBehalfOfCompId e DeliverToCompId. A B3 e o participante preenchem esse campo.

### 7.4 Envio (POST)

Utilizar o verbo POST para enviar mensagens. Enviar somente uma mensagem por vez em cada requisição POST, pois o sistema de mensageria não permite o envio de mensagens em lote em uma mesma requisição.

Caso a mensagem seja recebida com sucesso pela B3, será retornado o código HTTP 200 (OK) contendo no corpo da requisição o campo SeqNum (na seção TechnicalInformation) com o número de sequência, atribuído pela B3, referente a mensagem enviada, conforme ilustrado abaixo:

```
HTTP/1.1 200 OK
Content-Length: 170
Date: Fri, 04 Jan 2019 14:17:50 GMT
{
  "TechnicalInformation": {
    "SeqNum": 5,
    "PossDupFlag": null,
    "OnBehalfOfCompId": null,
    "DeliverToCompId": null,
    "GatewayName": null
  }
}
```



## API – IMERCADO: ALOCAÇÃO RENDA FIXA BALCÃO

Caso haja algum erro na validação da mensagem recebida o sistema retornará o código do erro e a descrição na mensagem de resposta, conforme ilustrado abaixo.

```
HTTP/1.1 400 Bad Request
Content-Length: 738
Date: Fri, 04 Jan 2019 14:17:50 GMT
{
  "TechnicalInformation": {
    "SeqNum": null,
    "PossDupFlag": null,
    "OnBehalfOfCompId": null,
    "DeliverToCompId": null,
    "GatewayName": null
  },
  "BusinessStatusError": {
    "BusinessStatus": {
      "code": "EBVMF0311",
      "description": "Mensagem nao autorizada",
      "comprehensiveDescription": "Mensagem nao autorizada",
      "dateTime": "2019-01-04T14:17:50"
    },
    "ErrorResponse": {
      "http": 400,
      "code": "EBVMF0311",
      "text": "O participante 7090 de categoria 39 nao esta autorizado a enviar a mensagem imb.a501.01 pelo endpoint /imercado/api/v1.0/TradeNotificationResponse,
      "developerText": "Por favor, entrar em contato com a B3,
      "moreInfo": null
    }
  }
}
```

### 7.5 Recepção (GET)

O participante deverá realizar requisições aos endpoints, de acordo com o fluxo de negócio, para recuperar as mensagens disponíveis (técnica de pooling). As mensagens ficam disponíveis ao longo do dia para as chamadas das APIs.

A B3 utiliza a técnica conhecida como "Long Polling" para manter a conexão HTTP aberta (até o tempo limite) quando no momento da requisição não há mensagens disponíveis. Quando houver mensagens disponíveis, a B3 as retornará na mesma conexão.

O participante **deve manter somente uma conexão HTTP por endpoint** para as operações GET.

O sistema da mensageria B3 iMercado armazena o SeqNum da última mensagem enviada em resposta a solicitação do participante e sempre atualiza esse em todos os "requests/responses", desde que seja a primeira vez que a mensagem esteja sendo disponibilizada ao participante. Quando for solicitado mensagens que já foram disponibilizadas anteriormente o controle do último SeqNum não será atualizado.

O número de sequência é informado no campo SeqNum (na seção TechnicalInformation) de cada mensagem.

Para cada requisição GET realizado pelo participante o sistema de mensageria B3 iMercado responderá enviando um lote de mensagens até o número máximo.

### 7.6 Parâmetro de Consulta (GET)

As chamadas e a tratativa de parâmetros serão realizadas por "QueryString".

Os endpoints que permitirem o método GET terão a possibilidade dos seguintes filtros (parâmetros):

Parâmetro (GET)	Valor Parâmetro	Observações
<ul style="list-style-type: none"> <li>SeqNumBegin</li> <li>SeqNumEnd</li> </ul>	Ausente	Ausência do parâmetro e do valor do parâmetro retorna todas as mensagens "novas" do dia. O sistema controla as mensagens enviadas e retorna apenas as pendentes de envio, respeitando o limite máximo definido pela B3.
	SeqNumBegin = "x"	Retorna todas as mensagens do dia, no range indicado, respeitando o limite máximo.
	SeqNumEnd = "y"	
	SeqNumEnd = "y"	Caso seja "declarado" somente o parâmetro SeqNumEnd, será retornado um erro. Nesse caso o parâmetro SeqNumBegin deve ser obrigatório
SeqNumBegin = "x"	Caso seja "declarado" somente o parâmetro SeqNumBegin, será retornado o limite máximo.	

## 8. API – iMercado Renda Fixa

### 8.1 OTIncomeTradeCaptured

API que contém informações relativas à captura de negócios de operações originadas do Balcão.

Verbo	Descrição
GET	Fornecer informações relativas à captura de negócio de operações originadas do Balcão.

### 8.2 OTIncomeAllocation

API que contém instruções para alocação ou exclusão de alocação.

Verbo	Descrição
POST	Informa dados para a instrução de alocação ou de exclusão de alocação.

### 8.3 OTIncomeAllocationStatus

API que retorna o status da alocação.

Verbo	Descrição
-------	-----------

## API – IMERCADO: ALOCAÇÃO RENDA FIXA BALCÃO

GET	Fornece informações relativas ao status das alocações
-----	---

### 8.4 OTIncomeAllocationAcceptanceOrRejection

API que contém informações sobre aprovação de alocação.

Verbo	Descrição
POST	Informa aprovação ou rejeição da alocação

### 8.5 OTIncomeTradeCancellation

API para pedido de cancelamento de negócio.

Verbo	Descrição
POST	Solicita o pedido de cancelamento do negócio

### 8.6 OTIncomeTradeCancellationApproval

API que contém informações para aprovação de alocação.

Verbo	Descrição
POST	Informa a aprovação do cancelamento do negócio

## 9. 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 está disponível no documento [iMercado – Alocação Renda Fixa](#).

## 10. Estrutura das APIs

A estrutura JSON das APIs está disponível no documento [iMercado – Alocação Renda Fixa](#) em forma de catálogo. Vale destacar que a estrutura JSON segue a nomenclatura definida na coluna *Message Item*. É importante ressaltar que a obrigatoriedade de cada campo também está definida neste documento.

## 11. Segurança

Neste capítulo temos as informações relacionadas à Segurança da Informação.

A B3 definiu como modelo de segurança para as APIs expostas o uso de Mutual SSL (Two Way SSL) com pinagem de certificado e tokens de assinatura JWT (JSON Web Token) para as requisições enviadas.

### 11.1 Mutual SSL (Two Way SSL)

Para que o canal de comunicação entre cliente e servidor seja seguro, para todas as APIs do iMercado, é utilizado protocolo HTTPS (Hyper Text Transfer Protocol Secure), o que implica a apresentação de certificado pelo servidor para garantir sua autenticidade e criptografia dos dados trafegados.

## API – IMERCADO: ALOCAÇÃO RENDA FIXA BALCÃO

Com o uso do Mutual SSL (Two Way SSL) o cliente também deverá apresentar certificado ao servidor durante o handshake SSL, garantindo assim mútua autenticidade.

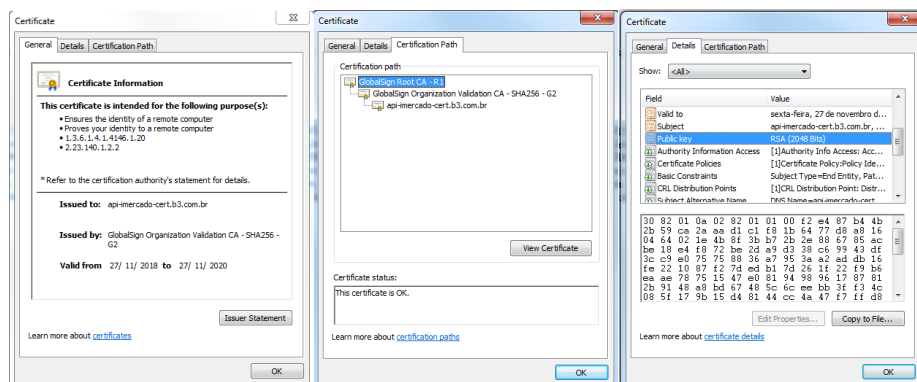
O certificado de cliente será fornecido pela B3 juntamente com a senha utilizada para proteger a chave privada.

Abaixo exemplo de conexão utilizando mutual SSL com o comando curl

```
curl --cert ./client.cer:senhafornecidapelaB3 --key client.key \  
--request GET https://api-mercado-cert.b3.com.br/healthcheck
```

### 11.2 Pinagem de Certificado

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



### 11.3 Token JWT no padrão OAuth 2.0

Para consumir a API, o usuário deverá realizar uma requisição de um token JWT - ele garante que o usuário está autorizado a seguir. Ele precisará informar as seguintes informações na requisição do token JWT no padrão OAuth 2.0:

**HTTP:** POST

**HOST:** <https://api-mercado-cert.b3.com.br/>

**Caminho:** /api/oauth/token

**Cabeçalho:**

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

Category\_ID: (Identificador da categoria do cliente **contratada**)**Comentado [RSJ1]:** Incluído esse campo no cabeçalho**Parâmetros do Corpo:***grant\_type* = client\_credentials (Esse valor é fixo - igual para todas as requisições)*client\_id* = c96rr10-dcf5-4231-ertert-cf886b8318fa (Aqui vai o client id que o usuário recebeu no pacote de acesso)*client\_secret* = wertwert-68cf-4a59-b51b-67ee6f820a77 (Aqui vai o client secret que o usuário recebeu no pacote de acesso)

O usuário deverá informar o certificado digital que recebeu no pacote de acesso, utilizando a técnica de Mutual SSL. Abaixo veja um exemplo de requisição completa de token JWT com Mutual SSL utilizando CURL - mas você pode implementar a requisição na linguagem que se sente mais confortável!

```
curl --cert ./certificado.cer:senhadocertificado --key chave.key \
--insecure --header "Content-Type: application/x-www-form-urlencoded" \
-d "grant_type=client_credentials&client_id=dc96rr10-dcf5-4231-ertert-
cf886b8318fa&client_secret=wertwert68cf-4a59-b51b-67ee6f820a77" \
-X POST https://api-mercado-cert.b3.com.br/api/oauth/token \
```

A resposta da requisição será:

```
{
"access_token": "eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiI5OTkiLCAiZXhwIjoiMTU2MDAwNzgyOSIsICJpc3MiOiJDTj1JbnRlcm1lZCBZJ0aWZpY2F0ZSAtIEpXVCxPVT1HQ01DLE89QjMgU0EsTD1TYW8gUGF1bG8sUz1TUcxDPUJSIn0.JSWxLOVX6xWmeLAhwdTP2xKt2eK3JAJ2oB6lYn06PROQdpaCk8E_CaS-
xGc2xz9iBEenLTxZTrfdhyYstkBv90fLXVQnhEVFhFXLq2Ov-
xWAO_DFPeGLXzy5_7WOpbZ3oKbjJ1XVxbCZnoDIt3VRZGNHAYiS8dZJzxV0n9D8qa_HhtZhOJNbH0ynhP
yoE8qMULvgJQ5DzjXlvk2mP-
KWlfhoY9CQnGseqTjrQWOj2kmYsQ9yCgAW6DRxB7LvTOavUk/vjOga7hmeXk9-
kunaqcu5EMpWzHeiFNGFhY1U4XhFtPZSIWzu23d6wDq5U0ZAWoV3Sw396d-hA3_cah9_hKQw",
"token_type": "Bearer",
"expires_in": "8640",
```

## API – IMERCADO: ALOCAÇÃO RENDA FIXA BALCÃO

```
"scope": "resource.WRITE resource.READ"
```

```
}
```

Os campos são:

- **access\_token**: é o token JWT. Ele deverá ser informado no cabeçalho das requisições às APIs.
- **expires\_in**: a validade do token é de XXX segundos. Após esse tempo, o token JWT não será mais válido, e o usuário deverá requisitar um novo token para consumir as APIs.
- **token\_type**: especifica o authentication schema utilizado pela API, por exemplo, Basic, Bearer, AWS, entre outros. Para as APIs de ofertas e negócios, será utilizado sempre o schema Bearer, conforme RFC:

<https://tools.ietf.org/html/rfc6750>

Para consumir a API, o **access\_token** deve ser enviado no padrão Bearer, através de um cabeçalho Authorization. Exemplo de chamada:

**HTTP:** POST

**HOST:** <https://api-mercado-cert.b3.com.br/>

**Caminho:** [api/mercado/allocacao/v1.0/TradeLegNotification](https://api-mercado-cert.b3.com.br/api/mercado/allocacao/v1.0/TradeLegNotification)

**Cabeçalho:** Authorization: Bearer {{access\_token}}



**B3.COM.BR**