

API - IMERCADO

Capture, Allocation and Give-up

Summary

1. Version log	3
2. Introduction	4
3. HTTP - API – iMercado verbs	5
4. Endpoint	5
5. iMercado API's Addresses	5
6. Swagger	5
7. How to Send (POST) and Receive (GET)	6
7.1 General Considerations	6
7.2 General Message Structure	6
7.3 About "TechnicalInformation"	6
7.4 Send (POST)	7
7.5 Receive (GET)	8
7.6 Query Parameter (GET)	9
8. API – iMercado Allocation	9
8.1 imb.a500.01 - TradeLegNotification	9
8.2 imb.a501.01 - TradeNotificationResponse	10
8.3 imb.a502.01 - SecuritiesAllocationInstruction	10
8.4 imb.a503.01 – SecuritiesLotAllocationInstruction	11
8.5 imb.a504.01 – SecuritiesAllocationInstructionStatusAdvice	11
8.6 imb.a505.01 – SecuritiesAllocationInstructionCancellation	11
8.7 imb.a506.01 TradeLegNotificationCancellation	11
8.8 imb.a507.01 – InvoiceNotification	11
8.9 imb.a508.01 – ETFTradeNotification	12
8.10 imb.a509.01 – InformGiveUpPendingApproval	12
8.11 imb.a510.01 – ResponseGiveUpAcceptanceOrRejection	12
8.12 imb.a521.01 – FinancialValueApprovalResponse	12
8.13 imb.a522.01 – SecuritiesAllocatedNetAmountNotification	13
9. Trade Flows	13
10. API Structures	13
11. Security	13
11.1 Mutual SSL (Two Way SSL)	13
11.2 Certificate Pinning	14
11.3 AWS Security Token (Amazon Web Services)	14

1. Version log

Date	Version	Description
28/09/2018	1.0	First Draft
09/10/2018	1.1	Change to the 5.0 section of this document.
08/01/2019	1.2	<ol style="list-style-type: none">1. Change to the “How to Send (POST) and Receive (GET)” and “Security” sections.2. Change to the “Swagger”.

2. Introduction

This document describes the definitions for the APIs that B3 has developed for use in iMercado capture, allocation and give-up flows. This document deals specifically with the API Web (REST) communication interface via internet.

Web-based Application Programming Interfaces (APIs) allow iMercado participants to develop applications for the automation of communication processes, using the iMercado messenger catalogue with machine-to-machine applications for participants that do not have access to the RCCF and RCB networks.

Built to the Representational State Transfer (REST) standard, they are comprised of a group of Uniform Resource Locators (URLs) that enable collection and posting of messages that are executed by systems. The information here is specific to the capture, allocation and give-up process and must be applied to every relevant API.

3. HTTP - API – iMercado verbs

For iMercado APIs, only the GET (to recover messages) and POST (to send messages) verbs will be used. The HTTP DELETE and PUT verbs are not supported.

4. Endpoint

Message endpoint code is set out in the iMercado catalogue. The URIs of each API are as follows.

API	URI
imb.a500.01 - TradeLegNotification	TradeLegNotification
imb.a501.01 - TradeNotificationResponse	TradeNotificationResponse
imb.a502.01 - SecuritiesAllocationInstruction	SecuritiesAllocationInstruction
imb.a503.01 - SecuritiesLotAllocationInstruction	SecuritiesLotAllocationInstruction
imb.a504.01 - SecuritiesAllocationInstructionStatusAdvice	SecuritiesAllocationInstructionStatusAdvice
imb.a505.01 - SecuritiesAllocationInstructionCancellation	SecuritiesAllocationInstructionCancellation
imb.a506.01 - TradeLegNotificationCancellation	TradeLegNotificationCancellation
imb.a507.01 - InvoiceNotification	InvoiceNotification
imb.a508.01 - ETFTradeNotification	ETFTradeNotification
imb.a509.01 - InformGiveUpPendingApproval	InformGiveUpPendingApproval
imb.a510.01 - ResponseGiveUpAcceptanceOrRejection	ResponseGiveUpAcceptanceOrRejection
imb.a521.01 - FinancialValueApprovalResponse	FinancialValueApprovalResponse
imb.a522.01 - SecuritiesAllocatedNetAmountNotification	SecuritiesAllocatedNetAmountNotification

Base URI (all URIs begin with): **/imercado/api/v1.0/**

Example of a full URL for production environment, referring to “imb.a500.01 – TradeLegNotification” message:

<https://api-mercado.b3.com.br/imercado/api/v1.0/TradeLegNotification>

5. iMercado API’s Addresses

B3 has two environments, one dedicated to participants certifications and the other one is the production environment.

The connection addresses referring to each environment are as follows:

- Certification: <https://api-mercado-cert.b3.com.br>
- Production: <https://api-mercado.b3.com.br>

6. Swagger

iMercado API’s technical documentation available in Swagger at: http://www.b3.com.br/en_us/project-t-2/documentos/

See: **Swagger iMercado**

7. How to Send (POST) and Receive (GET)

7.1 General Considerations

B3's iMercado Messenger system maintains control over the participants' sent and received messages, whereby each message receives a control number called SeqNum (SequenceNumber). The SeqNum (SequenceNumber) control is executed by endpoint. It is restarted (defined to 1) daily, as the iMercado Messenger is always daily, and therefore messages for days other than T0 cannot be recovered.

7.2 General Message Structure

All Webservices responses are delivered in JSON format, using an ISO 20022 based pattern.

Each message contains the following structure:

```
{
  "TechnicalInformation": {
  },
  "ApplicationHeader": {
  },
  "<Specific Business Data>:" {
  }
}
```

Where:

- TechnicalInformation = Contains technical information referring to the message transmission. In this section, some values will be set, such as SequenceNumber (only sent by B3), possibility of duplicity indicator and counterpart FIX session.
- ApplicationHeader = Contains some information that assists the message processing, as: Unique Id, message code, date/time reference, origin and destination participant.
- Specific business data = In this section there will be data referring to the transaction/business according to the iMercado's catalogue, as: TradeLegNotification, TradeNotificationResponse.

7.3 About “TechnicalInformation”

Below, there is a list of fields contained in the section “TechnicalInformation” and its instructions to set the values:

- SeqNum: This field contains the message sequence number set by B3. Only B3 can fill this field.

- **PossDupFlag:** This field contains the possibility of duplicity indicator. B3 and the participant can fill this field. If the message is being sent for the first time, expects its value to be “false”. In disaster recovery scenarios, the sender system may not be able to know if the in-process message was sent and may try a resend. If this happens, the next send attempts must be made changing its value to ‘true’
- **OnBehalfOfCompId:** This field contains the name of the sender participant FIX session. Only B3 can fill this field. The receiver participant must store this data to inform it through the “DeliverToCompId” field in the next messages on that same trade flow that will be generated in response.
- **DeliverToCompId:** This field contains the name of the receiver participant FIX session in which the message must be delivered. Only the sender participant can fill this field. This field must be filled with the value got from the “OnBehalfOfCompId” field, from a previous message, in the same trade flow.
- **GatewayName:** This field contains the name of the FIX Gateway that hosts the FIX session informed in OnBehalfOfCompId and DeliverToCompId fields. B3 and participants can fill this field

7.4 Send (POST)

Use the POST verb to send messages. Send only one message a time in each POST request, as the messenger system does not allow messages by lot in the same request.

If B3 receives the message successfully, the HTTP code 200 (OK) will be returned, containing in the request body (in TechnicalInformation section) the SeqNum field, with the sequence number that was assigned by B3, referring to the sent message, as below:

```
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
  }
}
```

```
}
}
```

If there is a validation error of the received message the system will return the error code and the response message description, as below:

```
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 Receive (GET)

From time to time, the participant will make requests to the endpoints in accordance with the trade flow to recover available messages (pooling technique). Messages are available for API calls throughout the day

B3 uses a technique known as “Long Polling” to keep the HTTP connection open (to the limit time) even when there are no available messages at the moment that the request is made. When messages are available, B3 will return them in the same connection.

The participant **must keep only one HTTP connection per endpoint** to the GET operations.

The B3 iMercado Messenger system stores the SeqNum of the last message sent as a response to the participant’s request and always updates it in every “request/response”, if it is the first time that message is

being made available to the participant. The control of the last SeqNum will not be updated when messages that have already been provided are requested.

The sequence number is informed in each message SeqNum field (in TechnicalInformation section).

The B3 iMercado messaging system will respond to each GET request by the participant by sending a message lot up to the maximum number

7.6 Query Parameter (GET)

The calls and the parameter treatment will be executed by "QueryString".

Endpoints that allow the GET method will have the following filter options (parameters):

Parameter (GET)	Parameter Value	Remarks
<ul style="list-style-type: none"> SeqNumBegin SeqNumEnd 	Missing	A missing parameter and a missing parameter value returns all the day's "new" messages. The system controls the messages sent and returns only the pending Send messages, complying with the maximum limit defined in configuration.
	SeqNumBegin = "x" SeqNumEnd = "y"	Returns all the day's messages in the designated range, complying with the maximum limit defined in configuration.
	SeqNumEnd = "y"	If only a SeqNumEnd parameter is declared, an error will be returned. In this case, the SeqNumBegin parameter will be mandatory.
	SeqNumBegin = "x"	If only the SeqNumBegin parameter is declared, only the maximum limit defined in configuration will be returned.

8. API – iMercado Allocation

The necessary technical information to implement API integration and use is available in the API – Overview document.

8.1 imb.a500.01 - TradeLegNotification

API that returns executed trades. This process will occur through the following trade processes at the Clearinghouse:

1. Trade capture by the Participant's system directly in the Master account or Subaccount;

2. Inclusion of allocation by the Full Trading Participant or Settlement Participant, from an account held in their own name to a Master account or Subaccount (e.g.: Capture account allocation inclusion to a Master account);
3. Trade capture by the Participant's system directly in the Master account or Subaccount with give-up link (Give-up).

Verb	Description
GET	Returns all the trades executed in a Master account or Subaccount.

8.2 imb.a501.01 - TradeNotificationResponse

API to accept/reject captured trades.

Verb	Description
POST	Accepts or rejects a trade executed by the Full Trading Participant

8.3 imb.a502.01 - SecuritiesAllocationInstruction

API to request inclusion of information on allocation by trade (identification of an end customer's account and other information necessary for settlement) and posting trade give-ups.

Verb	Description
POST	Requests inclusion of allocation from a Master account to a Subaccount by trade.

8.4 imb.a503.01 – SecuritiesLotAllocationInstruction

API to request inclusion of information on allocation by lot (identification of an end customer's account and other information necessary for settlement).

Verb	Description
POST	Requests inclusion of allocation from a Master account to a Subaccount by lot.

8.5 imb.a504.01 – SecuritiesAllocationInstructionStatusAdvice

API that sends:

- Status of the requested allocation by trade or lot.
 - Status of the allocation exclusion.
- Give-up status.
- Gross financial value of the allocation

Verb	Description
GET	Returns the status of the allocation, allocation exclusion, give-up and gross financial value by allocation.

8.6 imb.a505.01 – SecuritiesAllocationInstructionCancellation

API that requests trade allocation exclusion

Verb	Description
POST	Requests allocation exclusion

8.7 imb.a506.01 TradeLegNotificationCancellation

API that sends the trade cancellation.

Verb	Description
GET	Returns all the cancelled trades.

8.8 imb.a507.01 – InvoiceNotification

API that returns brokerage bill information after close of trade executed in the Participant's system.

Verb	Description
GET	Returns brokerage bill information.

8.9 imb.a508.01 – ETFTradeNotification

API that returns Primary ETF Payment/Redemption transactions and ETF Price/Settlement update executed in the Primary Market.

Verb	Description
GET	Returns Primary ETF Payment/Redemption transactions and ETF Price/Settlement update executed in the Primary Market.

8.10 imb.a509.01 – InformGiveUpPendingApproval

API that returns execution of a give-up trade that must be accepted or rejected by the Manager.

Verb	Description
GET	Returns the execution of a give-up trade that must be accepted or rejected by the Manager

8.11 imb.a510.01 – ResponseGiveUpAcceptanceOrRejection

API to accept/reject the give-up.

Verb	Description
POST	Accepts or Rejects a give-up

8.12 imb.a521.01 – FinancialValueApprovalResponse

API to validate net financial value by trade/allocation

Verb	Description
POST	Accepts or rejects net financial value by trade/allocation

8.13 imb.a522.01 – SecuritiesAllocatedNetAmountNotification

API that sends the net financial value per trade.

Verb	Description
GET	Returns the net financial value per trade.

9. Trade Flows

The APIs will be consumed in a specific order to meet a trade flow. Designation of a specific use sequence is available in the document [iMercado – Capture, Allocation and Give-up](#).

10. API Structures

The APIs' JSON structure is available in the [iMercado – Capture, Allocation and Give-up](#) document in catalogue form. It should be noted that the JSON structure uses the names defined in the Message Item column. Please note that this document establishes the mandatory nature of each field.

11. Security

This chapter contains information related to Information Security.

B3 has defined as a security model to exposed APIs the use of Mutual SSL (Two Way SSL) with certificate pinning and AWS security Tokens to send requests.

11.1 Mutual SSL (Two Way SSL)

To ensure security in the communication channel between client and server, all iMercado APIs uses HTTPS protocol (Hyper Text Transfer Protocol Secure), which implies the usage of certificate by the server to ensure its authenticity and encryption of the transmitted data. With Mutual SSL (Two Way SSL), the client also must use the certificate to server while SSL handshaking, ensuring mutual authenticity.

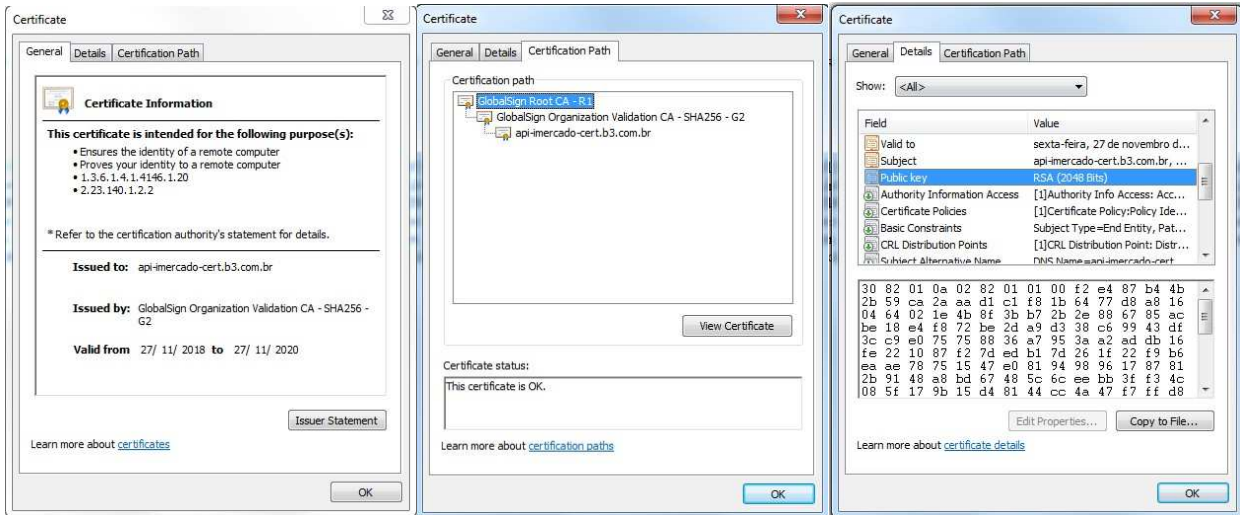
The client certificate will be provided by B3 along with the password used to protect the private key

Below there is an example of connection using Mutual SSL with curl command:

```
curl --cert ./client.cer:passwordprovidedbyB3 --key client.key \
--request GET https://api-mercado-cert.b3.com.br/healthcheck
```

11.2 Certificate Pinning

To consume the APIs is required to implement the root certificate pinning of the connecton address. The images below show how to obtain the information to the certificate pinning implementation.



11.3 AWS Security Token (Amazon Web Services)

The AWS default security token increases the security in the client-server communication, preventing request tampering in attacks such as “man in the middle”.

To generate the security token, the elements that compose the request are concatenated in one string, then the HMAC (Hash Message Authentication Code) calculation is made, following the RFC 2104 HMAC-SHA1 pattern and using the keyword (secret).

The request must come with an Authorization header containing the identification key (API key) and the token, separated by “:”, as below:

```

GET /imercado/api/v1.0/TradeLegNotification HTTP/1.1
Host: api-mercado-dev.internalenv.corp
Content-Type: application/json
Authorization: AWS f341f39c-ae77-4c64-a89e-a1cb8f7620c2:ePyj/UnXsU9nu0ZBjRc2MJH3tmA=
    
```

Diagram illustrating the Authorization header structure:

- The **Api Key** is the portion of the string before the colon: `f341f39c-ae77-4c64-a89e-a1cb8f7620c2:ePyj/UnXsU9nu0ZBjRc2MJH3tmA=`
- The **Signature** is the portion of the string after the colon: `UnXsU9nu0ZBjRc2MJH3tmA=`

For this example, the used secret was: 7b9e79e5-b8fb-4f47-9c8e-1ccbba45006e

Use the request data above to validate your implementation.

The API key and secret for request signing in production and certification environments will be provided by B3.

The link below contains the pattern complete specification:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/RESTAuthentication.html>



B3.COM.BR