

# GUIDE TO LINE 5.0 API

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b>	<b>4</b>
<b>2</b>	<b>OVERVIEW OF API</b>	<b>4</b>
2.1	Filter	4
2.2	Nomenclature	4
2.3	Support for GZIP	4
2.4	Functionalities not supported	4
<b>3</b>	<b>SECURITY</b>	<b>5</b>
3.1	Authentication – obtaining tokens	5
3.2	Authentication – token renewal	6
3.3	Request:	6
3.4	Access to endpoints – using tokens	7
3.5	More information about Bearer Authorization:	8
3.6	Authorization	8
<b>4</b>	<b>LINE APIS</b>	<b>8</b>
4.1	Alert	8
4.2	Document	9
4.3	AlertCfgtn	10
4.4	Account	10
4.5	OperationsProfessionalParticipant	11

4.6	Participant .....	13
4.7	RiskProfile .....	13
4.8	Position.....	14
4.9	Auxiliary endpoints.....	15
5	TEST TOOL.....	15
5.1	Error message structure .....	15

## 1 INTRODUCTION

The LINE 5.0 pre-trade risk control system is designed to evaluate all forms of access to B3's electronic trading platform in accordance with preset limits.

The LINE 5.0 API provides web endpoints built in the Representational State Transfer (REST) architectural style that can be used to query, register, update and remove limit settings.

## 2 OVERVIEW OF API

### 2.1 Filter

The standard used to filter endpoints in queries via the LINE 5.0 API is the query string, with parameters from the URL.

Example of filter using two parameters (param1 and param2):

```
https:// {line5.0-domain}/api/vi/{resource}?param1=valor1&param2=valor2
```

### 2.2 Nomenclature

Nomenclature follows the provisions of the document *B3's APIs – Overview*.

### 2.3 Support for GZIP

In order to reduce network bandwidth use, the LINE 5.0 API supports data compression via GZIP. The procedure for requesting compression is described in the section on Usage in the document *B3's APIs – Overview*.

### 2.4 Functionalities not supported

Version 1 of the LINE 5.0 API does not support the following functionalities:

- Pagination
- Sorting
- Field selection

## 3 SECURITY

The LINE 5.0 API uses OAuth2 to guarantee the authentication and authorization of its web endpoints, based on the use of tokens, with access solely via HTTPS.

The security architecture, implemented using OAuth2, is standard for B3's APIs and detailed in the section on Security of the document *B3's APIs - Overview*.

### 3.1 Authentication – obtaining tokens

To obtain tokens *access\_token* and *refresh\_token*, authentication via the LINE 5.0 API will be required:

Obtain Basic Authorization (HTTP GET)

Request:

Example using curl:

```
curl -X GET https://{line5.0-domain}/api/v1.0/token/authorization
```

Response:

```
{"header": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"}
```

1) Obtain tokens using Basic Authorization (HTTP POST)

2) Request:

Header Parameters

Authorization: Basic xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

content-type : application/x-www-form-urlencoded

Payload

grant\_type=password





Response:

```
[{"id":999,"code":999999999,"document":{"id":9999,"code":"123.456.789-10","name":"Fulano","broker":{"id":999,"category":"PNP"},"isProtected":false,"isBlocked":true,"name":"Fulano","isDisable":false}, {"id":999,"code":000000000,"document":{"id":0000,"code":"123.456.789-11","name":"Ciclano","broker":{"id":999,"category":"PNP"},"isProtected":true,"isBlocked":false,"name":"Ciclano","isDisable":true}]
```

### 3.5 More information about Bearer Authorization:

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

### 3.6 Authorization

The endpoints of the LINE 5.0 API have validations that use access rules. In authentication, the user consisting of login + password + trading participant code + category code is validated, and the token generated reflects this information. If the user tries to access an endpoint that requires a rule not contained in the token, the user will receive a response with an error code.

## 4 LINE APIs

The technical information required for the implementation of integrations is available in the LINE API catalogue at <https://LINE5-CERT/api/endpoints>.

### 4.1 Alert

Service for listing or updating alerts:

Method	Description
GET	Obtains system alerts
POST	Updates alert generation parameters



## 4.2 Document

Service for listing or updating information on pre-trade risk configurations for customers and specific limits that supersede those defined by the profile with which the customer is associated:

Method	Description
GET /v1.0/document/{docId}	Obtains data for a customer filtered by document number
POST /v1.0/document/{docId}	Updates the trading model of a customer identified by a document number
DELETE /v1.0/document/{docId}/lmt	Deletes limit data for a customer identified by a document number
GET /v1.0/document/{docId}/lmt	Obtains limit data for a customer identified by a document number
POST /v1.0/document/{docId}/lmt	Registers limit data for a customer identified by a document number
GET /v1.0/document/{docId}/lmt/mkta	Obtains limits in authorized markets for a customer identified by a document number
POST /v1.0/document/{docId}/lmt/mkta	Registers authorized markets for a customer identified by a document number
DELETE /v1.0/document/{docId}/lmt/mkta	Deletes authorized markets for a customer identified by a document number
GET /v1.0/document/{docId}/lmt/spxi	
POST /v1.0/document/{docId}/lmt/spxi	
DELETE /v1.0/document/{docId}/lmt/spxi	
GET /v1.0/document/{docId}/lmt/std	Obtains balance of forwards in limits for a customer identified by a document number
POST /v1.0/document/{docId}/lmt/std	Registers balance of forwards in limits for a customer identified by a document number
DELETE /v1.0/document/{docId}/lmt/std	Deletes balance of forwards in limits for a customer identified by a document number

## 4.3 AlertCfgtn

Service for listing, updating or deleting alert settings:

Method	Description
GET /v1.0/alertCfgtn	Obtains alert settings
POST /v1.0/alertCfgtn	Registers alert parameters
POST /v1.0/alertCfgtn/{alertCfgtnId}	Registers parameters for a specific alert
DELETE /v1.0/alertCfgtn/{alertCfgtnId}	Deletes specific alert setting

## 4.4 Account

Service for listing or updating limit settings and properties associated with accounts and specific limits that supersede those defined by the profile with which the account is associated:

Method	Endpoint
GET /v1.0/account	Obtains account data
GET /v1.0/account/{acctId}	Obtains data for an account identified by a code
POST /v1.0/account/{acctId}	Registers data for an account identified by a code
DELETE /v1.0/account/{acctId}/lmt	Deletes limit data for an account identified by a code
GET /v1.0/account/{acctId}/lmt	Obtains limit data for an account identified by a code
POST /v1.0/account/{acctId}/lmt	Registers limit data for an account identified by a code

GET /v1.0/account/{acctId}/lmt/mkta	Obtains authorized markets for an account identified by a code
POST /v1.0/account/{acctId}/lmt/mkta	Registers authorized markets for an account identified by a code
DELETE /v1.0/account/{acctId}/lmt/mkta	Deletes authorized markets for an account identified by a code
GET /v1.0/account/{acctId}/lmt/spxi	
POST /v1.0/account/{acctId}/lmt/spxi	
DELETE /v1.0/account/{acctId}/lmt/spxi	
GET /v1.0/account/{acctId}/lmt/std	Obtains balance of forwards for an account identified by a code
POST /v1.0/account/{acctId}/lmt/std	Registers balance of forwards for an account identified by a code
DELETE /v1.0/account/{acctId}/lmt/std	Deletes balance of forwards for an account identified by a code

## 4.5 OperationsProfessionalParticipant

Service for querying or updating pre-trade limit settings associated with traders:

Method	Endpoint
GET /v1.0/operationsProfessionalParticipant	Obtains information about traders
GET /v1.0/operationsProfessionalParticipant/{oprn sPrfalPtcptId}	Obtains information about a trader identified by a code
POST /v1.0/operationsProfessionalParticipant/{oprn sPrfalPtcptId}	Registers data for a trader identified by a code
DELETE /v1.0/operationsProfessionalParticipant/{oprn sPrfalPtcptId}/lmt	Deletes limit data for a trader identified by a code

GET /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt	Obtains limits for a trader identified by a code
POST /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt	Registers limits for a trader identified by a code
GET /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/mkta	Obtains authorized markets for a trader identified by a code
POST /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/mkta	Registers authorized markets for a trader identified by a code
DELETE /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/mkta	Deletes authorized markets for a trader identified by a code
GET /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/spxi	
POST /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/spxi	
DELETE /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/spxi	
GET /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/std	Obtains balance of forwards for a trader identified by a code
POST /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/std	Registers balance of forwards for a trader identified by a code
DELETE /v1.0/operationsProfessionalParticipant/{oprnsPrfalPtcptId}/lmt/std	Deletes balance of forwards for a trader identified by a code

## 4.6 Participant

Service for listing or updating a participant's limit settings and properties at the pre-trade risk level:

Method	Endpoint
GET /v1.0/participant/{ptcptId}	Obtains data for a participant identified by a code
POST /v1.0/participant/{ptcptId}	Registers data for a participant identified by a code

## 4.7 RiskProfile

Profiles are common definitions of types of account, customer or trader, concentrating limit settings. For example, if there are accounts that will be configured with the same limit values, it is possible to create a profile with these limit values and later associate the accounts with this profile (the same applies to customers and traders). The purpose of a profile is to minimize the work participants need to do to set limits:

Method	Endpoint
GET /v1.0/riskProfile	Obtains data for risk profiles
POST /v1.0/riskProfile	Registers risk profiles
GET /v1.0/riskProfile/{prflId}	Obtains data for a specific risk profile
POST /v1.0/riskProfile/{prflId}	Registers information for a risk profile identified by a code
DELETE /v1.0/riskProfile/{prflId}	Deletes a specific risk profile

GET /v1.0/riskProfile/{prflId}/enty	Obtains a specific risk profile for an entity
POST /v1.0/riskProfile/{prflId}/enty	Registers a risk profile for an entity
DELETE /v1.0/riskProfile/{prflId}/lmt	Deletes a limit associated with a specific profile
GET /v1.0/riskProfile/{prflId}/lmt	Obtains limit data associated with a specific profile
POST /v1.0/riskProfile/{prflId}/lmt	Registers limit data for a specific profile
GET /v1.0/riskProfile/{prflId}/lmt/mktas	Obtains authorized markets for a specific profile
POST /v1.0/riskProfile/{prflId}/lmt/mktas	Registers authorized markets for a specific profile
DELETE /v1.0/riskProfile/{prflId}/lmt/mktas	Deletes authorized markets for a specific profile
GET /v1.0/riskProfile/{prflId}/lmt/spxis	
POST /v1.0/riskProfile/{prflId}/lmt/spxis	
DELETE /v1.0/riskProfile/{prflId}/lmt/spxis	
GET /v1.0/riskProfile/{prflId}/lmt/stds	
POST /v1.0/riskProfile/{prflId}/lmt/stds	
DELETE /v1.0/riskProfile/{prflId}/lmt/stds	

## 4.8 Position

Positions are values that represent \*\*\*

Method	Endpoint
POST /v1.0/position/hstry	Obtains the history of a participant's positions
GET /vi/position/cur	
POST /v1.0/position/rank	

## 4.9 Auxiliary endpoints

Auxiliary endpoints for listing types of entity, such as types of metric, types of profile etc.

Method	Description
GET /v1.0/metricType	Types of metric
GET /vi/profileType	Types of profile
GET /v1.0/lineEntityType	Types of entity
GET /v1.0/user	Information on logged-in user

## 5 TEST TOOL

The LINE 5.0 API provides a tool to test all functionalities in the certification environment, enabling developers to navigate to endpoints, obtain their documentation and execute them, simulating a real execution. Developers can use this tool to refine their understanding of each feature and we recommend its use to test the examples available via cURL, so as to help develop their solutions. The tool is available at <https://LINE5-CERT/api/>. In production this tool will be disabled.

### 5.1 Error message structure

The error return standard used by LINE 5.0 is a JSON object containing an identifier (represented by the *logref* attribute) and a message (represented by the *message* attribute). The reported identifier will enable B3's support team to identify the problem that has occurred in the logs. The message will describe the problem in a humanly comprehensible format.

```
VndErrors[VndError[logref: <codErro>, message: <msgErro>, links: []]]
```

**codErro:** error code to be reported to B3 if support is required – problem tracking identifier.

**msgErro:** error message – description of the problem that occurred, e.g. an attempt to set a limit using the wrong metric group:

*“Only SPXI metric group can receive inheritance! Received: {Limit={brokerCode:000, profileId:000, metricGroup:10, groupId:00, instrumentId:000, symbol:"BVMF3"}}”*

## CHANGE LOG:

Date	Version	Description	Author
JUN/2017	4.1	First version	ESC
AUG/2017	4.2	Example of cURL call with GZIP	ESC
AUG/2017	4.3	Inclusion of alerts and positions	ESC
SEP/2017	4.4	Alignment of endpoints with canonical	ESC