

## GENERAL

### Main concepts

The solution provided to the propfirm is structured by the following components:

- User Dashboard: propfirm's customers can login on this portal in order to see their metrics, open orders/positions, trading report and platforms(if applicable)
- Admin Dashboard: propfirm's managers has access to this area and it can be used for the following purposes:
  - customer support: admins can access to all information about accounts, historical orders, historical trades and so on
  - management: creation of user, trading rule and so on
- APIs:
  - Rest API: they are used in order to manage different entities, like: User, Group Universes, Trading Rules, Subscriptions and Trading Accounts.
  - Webhooks: enabled on request and they allows the propfirm to get notified on real time when there are some changes/events on the following entities:
    - Accounts: Creation, Snapshot updates and status changes
    - Positions: Overnight
    - Subscriptions: Creation, Sign of the agreement (NO\_PRO / PRO), changes of status
    - Report of a trade closed
- Important: all entities can be created, edited and deleted both from API and admin dashboard. Furthermore all the information available on user's dashboard are available also through API. It is care of the propfirm understanding what it is better to integrate on their side through API or in which scenarios it is better to use our portal.

### Architecture

We provide to the propfirm two different environments:

- staging: used for testing and development purposes. Users and accounts created on this environment are free of charge
- production: used for real users and accounts

## ENTITIES

### Organization users

On the system there is a distinguish between normal users(customers) and propfirm's employess or managers. The second one can be divided into the following roles:

- viewer: readonly user
- manager: can see accounts details and eventually close orders and positions
- administrator: full control on symbols, users, rules and so on

When a new account is bought on propfirm's side, the first step to create is an organization user, since the trading accounts must be attached to a specific organization user.

Organization users can have two different management mode. Dedicated management mode is not enabled as default and it may have an extra cost since it allows more management, iframes view and so on.

- **Shared:** users have only one credential on VolumetricaFX system which is used to join / link several and different organizations. In this scenario, the organization generates an invitation url to join its organization.
- **Dedicated:** organization is entitled to create a specific user which will only have the accounts of its organization. In this scenario, a random username will be generated and should be sent to the user in order to access the dashboard and the platform. Furthermore, Propfirm can also set a specific password on user creation if wanted, otherwise a random password will be generated

## Trading Accounts

Users can potentially have an unlimited number of accounts and we categorize the accounts on different mode (both for billing and also for an architecture prospective):

- Evaluations (mode 0): it is usually the evaluation account
- SimFunded (mode 1 ) / Funded (mode 2): it is the sim funded account where user trades in a SIM environment, but he is eligible for payouts
- Live (mode 3): it is a live account connected to a broker where user trades with live money
- Trial (mode 4): trial account, if agreed, it can be used for demo purposes
- Contest (mode 5): contest account, it is the account to make contests. We need to categorize it since they usually have a different billing
- Training (mode 100): reserved mode, all users are automatically provided with a training account where they can set custom balances, reset it and so on. It used to both test / learn the platform and also to improve personal trading skills. It is free of charge.

During its lifetime, the account can have the following status values:

- Initialized = Account is created, but not enabled yet. Account can persist this status if it has been setted a delayed start (not common used)
- Enabled = Account is enabled and ready for trading
- Challenge Success = if enabled on the risk/rule parameters, it means that the user has reach the run-up/profit target
- Challenge Failed = if enabled on the risk/rule parameters, it means that user has breached one of the risk params like for example intraday drawdown, drawdown and so on
- Disabled = The account has been manually disabled by API or propfirm's admins

Risk parameters for the account are managed by the trading rule(we see it on details later). When an account is created, it can be decided to set a dedicated trading rule(account level) with a custom drawdown and so on, or it can be used a trading rule globally defined. What it is best to use, it depends on case by case. One thing to note is that, if a global trading rule is used, **any changes to that rule will affect ALL the accounts associated** on real time.

## RISK PARAMETERS

### Symbols

By default, the solution is provided by a preset of symbols available with default margins, spreads, commissions and leverage that an account can open.

For symbol management, propfirms have different possibilities:

- changes margins and commission for all account, editing them on symbol's admin page
- use custom margins and commissions for specific trading rule or limit the symbols available: this can be done using the "Group universe" function

## Group Universes

Propfirm can create several different group universes, which overrides the list of symbol available for the accounts with custom margins and symbols.

Groups universes can be linked to many trading rules, so it means that the relationship is 1-N.

The use of group universes is not a requirement, since trading rule can also be created without any group universe associated, **but it is strongly recommended to handle margins and specific leverage based on the product type.**

## Trading Rules

As written before, trading rules can be both global defined or defined on the account level. It is the entity which controls all the risk parameters for the accounts, in particular:

- Maximum Drawdown
- Run-up / Profit target
- Intraday Drawdown
- Intraday Runup
- Maximum position loss
- Maximum position gain
- Max portfolio loss
- Max portfolio gain
- Maximum daily number of trades
- Various: minimum session numbers, overnight and overweek allowed and so on

Each of the previous params can have different settings, both in term of enable or disable, values and anchor mode, but there are some common terms which have sense to explain:

- Balance: it is the closed balance of the account, so it means that it does NOT include floating/open PnL
- Equity: it is the floating balance of the account, which includes open positions

Actions available:

- None: the risk param is not enabled for the account
- Challenge fail: if triggered, the account is immediately disabled
- Flat: all open positions are flattened, but account keeps active
- Intraday disable: all open positions are flattened, the account is disabled for the CURRENT trading session. User will be able to trade again when next session opens

Values – it can be defined both or either a money amount or a % amount and it is possible to decide(selection) which of them should be used for risk evaluation

Trading rules are cyclically checked for each account with a very low frequency in order to disable the accounts that goes outside the defined risk params.

## ADMIN DASHBOARD

Beyond the management of the entities, admins can access to all account information, which includes:

- realtime updates of accounts balance
- view of open orders and positions of specific account
- list of the historical trade for manual check of consistency or similar things

- list of the historical orders which includes the source IPs, platform and date/time. It is useful for tickets management

Furthermore, it is available a monitor page, where propfirm can see the list of all their accounts, filtered by specific trading rule, with realtime updates of balances, equities, open trades and so on.

## API

### Rest API

Swagger Docs. It contains also parameters details:

<https://staging-api.volumetricafx.com/swagger/index.html>

All the request must be executed as follow and the result is on json format:

- Endpoint:
  - Staging: <https://staging-api.volumetricafx.com/api/v2/propsite>
  - Production: <https://api.volumetricafx.com/api/v2/propsite>
- It must be included this header in the request:
  - x-api-key = *token provided by email*
- All responses message are wrapped in the following object:
  - success: *bool*
  - data: *object* - if successful (200 Status code) and expected for the API, it contains the object/entity
  - statusCode: *int* – sent only if unsuccessful (status code different to 200)
  - message: *string* - sent only if unsuccessful
  - details: list of *string* – sent only if unsuccessful, it provides extra details

On swagger there are descriptions of different endpoints available with also the description of the various params. If more details are required, feel free to reach us so we can help you and support the integration.

On this document, we analyze only the most important APIs available:

- User:
  - Dedicated user management endpoints
    - Creation - /user (POST):
      - It must be provided all the user details(name, surname and so on). They are needed for market data agreements as explained before
      - Parameters:
        - “encryptionMode” – if password is encrypted on API exchanges:
          - 0 = None encryption
          - 1 = AES\_256 encryption – key is provided via mail
        - “forceNewPassword”: it is setted at false by default. A password is ALWAYS generated if the user is new. It may be setted to true if it is required a new password on user update
      - Returns:
        - “userId:” it **must be stored** on the propfirm’s DB because it is necessary for the next calls

- password: it contains the password just created. It is encrypted with the same approach of “encryptionMode”
- IMPORTANT: each propfirm can decide if they want to send the login credentials via email to the user or not. Two different approaches can be used:
  - Propfirm stores the users’ passwords on its database and it shows them in its dashboard
  - Password is sent via email from our backend to each user and the propfirm doesn’t need to store them.
- /user/loginurl (POST): it can be used to generate a OTP url, which redirects the user to our dashboard. In this case, the user will be automatically logged in. It can be useful for putting a button on propfirm’s customer area
- Shared user management endpoint
  - Generate invitation url - /user/invite (POST):
    - It is required to provide the country and it is suggested to send also an externalId, which is an id on your side in order to avoid to generate more invite for the same user on your side
    - Returns:
      - “userId:” it **must be stored** on the propfirm’s DB because it is necessary for the next calls
      - status: the status of the invitation
      - inviteUrl: this is the url generated which should be either send or displayed somewhere on your dashboard in order to allow user to join the organization
- IFrame available:
  - Dedicated user management only:
    - /VolumetricaWebApp (most used): it generates an url which can be embedded in an iframe and shows the web trading terminal. This is a nice feature if you want to allow the user to directly trade from your dashboard without the need to login in any other portal
    - /iframe: it generates an iframe of the trading dashboard which can be embedded. It shows on realtime orders, positions, balance information and so on
    - /iframePortfolio: it generates iframe which shows only the positions and orders widget (always updated on realtime)
- Trading Accounts:
  - /tradingAccount (GET):
    - it returns the real time information for the trading account. Look at swagger for field available, they should be intuitive
  - /tradingAccount (POST):
    - it creates a new trading account.
    - Params that need an extra explanation:
      - “userId”: it is the id returned on the user creation
      - “currency”:
        - 0 = EUR
        - 1 = USD (we suggest to use this)
      - “mode”:
        - Evaluations (mode 0): it is usually the evaluation account

- SimFunded (mode 1) / Funded (mode 2): it is the sim funded account where user trades in a SIM environment, but he is eligible for payouts
- Live (mode 3): it is a live account connected to a broker where user trades with live money
- Trial (mode 4): trial account, if agreed, it can be used for demo purposes
- Contest (mode 5): contest account, it is the account to make contests. We need to categorize it since they usually have a different billing
- Training (mode 100): reserved mode, all users are automatically provided with a training account where they can set custom balances, reset it and so on. It used to both test / learn the platform and also to improve personal trading skills. It is free of charge.
- “portfolioMode”:
  - 0 = Netting mode
  - 1 = Hedging mode
- “header”: default is header is something like: DFX-CUM\_NUMBER. Setting this field, it can be overridden.
- “description”: custom description that it will be shown on the dashboard
- “disableOtherAccountsEnabled”: if true – it disables all the other account enabled
- Account expirations:
  - “expirationMode”:
    - 0 = never expires
    - 1 = use “endDate” sent
    - 2 = use “expirationDays” and it adds them from activation date/time
    - 3 = use “expirationDays” and it adds them from the first order inserted date/time
    - 4 = use “expirationDays” and it adds them from the first order executed date/time
- Trading rules:
  - Global level: it must be setted the “accountRuleId” param
  - Account level: it must me populated “accountCustomRule”, setting all the parameters
- “subscriptionDetail”:
  - if you directly manage the subscription, do NOT use it. It is used only on transparent subscription management
  - “dataFeedProducts” – list of integers: each value corresponds to a different exchange data entitlements. Values changes based on the propfirm requirements, but just to understand we make some examples:
    - 0 = CME L1
    - 1 = COMEX L1
    - 2 = CME L2
  - “platform”:
    - 0 = VOLUMETRICA\_TRADING:

- if this platform is selected, user can chose which platform to use(VolSys or VolBook) from the portal or a default platform can be setted through “volumetricaPlatform” field
    - 1 = ATAS
    - 2 = QUANTOWER
- Important fields returned:
  - “accountId” returned must be stored for next calls or for having reference on webhook notifications
  - “tradingRuleId”: if it has been setted the trading rule on the account level, it is generated a new id and it may be used later for changing params
- /tradingAccount/Enable (POST):
  - it enables a trading account
- /tradingAccount/Disable (POST):
  - it disables a trading account. If you want to force the disable, set “forceClose” to true, otherwise if it has open positions and the market is closed, an exception can be thrown
  - send also a “reason”, so the user can see it on the dashboard. It may avoid unnecessary tickets
- /tradingAccount/ChangeStatus (POST):
  - if a disable is requested and you want to force the disable, set “forceClose” to true, otherwise if it has open positions and the market is closed, an exception can be thrown
  - send also a “reason”, so the user can see it on the dashboard. It may avoid unnecessary tickets
  - “status”: new status for the account (list below)
- /tradingAccount/changeTradingRule (POST):
  - change the trading rule for the account
  - IMPORTANT: it cannot be used if the rule used by the account is on the account level
- /tradingAccount (DELETE):
  - it cannot be used on active accounts
- Important fields available on the response or webhook body received:
  - “accountId:” internal id of the account. It must be used for next calls
  - “status” – see description on entity section
    - 0 = Initialized
    - 1 = Enabled
    - 2 = ChallengeSuccess
    - 4 = ChallengeFailed
    - 8 = Disabled
  - “reason”: if account is disabled, it provides the reason of the disable. Possible values are:
    - "TRADING\_RULE\_BALANCE";
    - "TRADING\_RULE\_RUNUP";
    - "TRADING\_RULE\_MAX\_DD";
    - "TRADING\_RULE\_MAX\_INTRADAY\_DD";
    - "TRADING\_RULE\_MAX\_INTRADAY\_RUNUP";
    - "TRADING\_RULE\_MAX\_POS\_OPEN\_LOSS";

- "TRADING\_RULE\_MAX\_POS\_OPEN\_GAIN";
- "TRADING\_RULE\_MAX\_OPEN\_LOSS";
- "TRADING\_RULE\_MAX\_OPEN\_GAIN";
- "TRADING\_RULE\_MAX\_NUMBER\_TRADES";
- "TRADING\_RULE\_OVERNIGHT";
- "TRADING\_RULE\_OVERWEEK";
- Trading rules:
  - they can be created or edited also through API
  - /tradingRule (GET):
    - it returns the trading rule params
  - /tradingRule (POST):
    - create new trading rule globally defined
    - update an existing trading rule: it can be both global or account level
  - /tradingRule/validate (POST):
    - it checks if the params are correctly setted. If necessary, it must be call before the creation or update of the trading rule
  - /tradingRule/changeGroupUniverse (POST):
    - it changes the group universe associated to a trading rule

### Webhooks

We strictly suggest the integration of webhooks technology, because it both helps propfirm to get notified on realtime when something happens and our resources to have a lower overhead compared to polling all the accounts to get changes.

The architecture is structured to be consistent, which means that failed webhooks are sent until we do not receive an HTTP 200 status code, so couple of considerations:

- you do not have any risk of loosing information during update or downtime
- they are sent sequentially in order to rebuild the changes of status in a chronological order
- if you have reviewed the request, please always answer 200 OK, even if some exceptions are thrown, otherwise all other webhooks are blocked

Propfirm should provide an url endpoint where they can receive the HTTP requests. Requests will be sent using the following token:

- x-api-key: token provided via email. Usually it is the same of token used for Rest API requests

At the moment, they are available the following categories of notifications:

- 0 = Accounts:
  - Events available:
    - 0 = Created
    - 1 = Update – it can be configured to receive the following updates:
      - account status changes: it is enabled by default. It is sent a notification for example when an account switches from “Enable” to “Challenge failed”
      - balance changes: it can be enabled on request, an event it is sent each time that a transaction is closed on the account
    - 2 = Delete
- 1 = Positions
  - Events available:



- 3 = Overnight: it notifies overnight positions for the account. Some propfirm do not want to disable account as default on overnight, but managed it on their side
- 3 = Trade Report – it notifies an close transaction with PnL informations
  - it is enabled on request
  - Events available:
    - 0 = Created
- 5 = Organization User – it notifies when a user organization status changed. For example when the user accepts the invite to join the organization

For development or debug, propfirm's admins can see webhook generated on the admin portal with also the JSON body generated.

If you need some examples for specific situations, please directly ask us and we will provide JSON payloads example.

Last thing: on swagger there is an endpoint which allows you to see what it is received through webhook, in particular: `/webhook/GetModel`