



X Open Hub

xManager - Communications
Protocol



xManager

xManager - Communications Protocol

v. 1.01

Disclaimer: X Open Hub Limited is authorised and regulated by the UK Financial Conduct Authority (FRN 522157), company number 7227848. Registered trading offices are at 29 Marylebone Road, London NW1 5JX, United Kingdom.

W: www.xopenhub.pro | **Skype:** xopenhub | **T:** +44 (0)20 3394 1700 | **A:** 29 Marylebone Road, London. NW1 5JX. UK.



History of document modification:

| Date | Version | Author | Description |
|------------|---------|--------------------------------|---|
| 07.06.2013 | 0.90 | Michał Bobruk Marcin Szarek | Initial Draft |
| 12.06.2013 | 0.91 | Michał Bobruk Marcin Szarek | Draft |
| 21.08.2013 | 0.92 | Michał Bobruk Marcin Szarek | check account password, position info, order info |
| 04.10.2013 | 0.93 | Michał Bobruk Marcin Szarek | create order, modify order, delete order, modify position, close position, account instrument list, instrument info |
| 14.11.2013 | 0.94 | Michał Bobruk Marcin Szarek | change manager account password, gross/net account |
| 13.12.2013 | 0.95 | Michał Bobruk Marcin Szarek | leverage list, journal list |
| 07.02.2014 | 0.96 | Michał Bobruk Marcin Szarek | configure session, logged account list, closed trade id list, credit in/out, set push response |
| 07.03.2014 | 0.97 | Michał Bobruk Marcin Szarek | Introducing Brokers (IB), currency list |
| 06.04.2014 | 0.98 | Michał Bobruk Marcin Szarek | Multi Account Manager (MAM) |
| 30.06.2014 | 0.99 | Michał Bobruk Marcin Szarek | execution list |
| 01.02.2015 | 1.00 | Michał Bobruk Marcin Szarek | account group list position reference type list free margin rule description list free margin rule name list mirror mode list mirror book list offer name list commission pattern list |
| 17.07.2015 | 1.01 | Michał Bobruk Marcin Szarek | ib ledger details list |



Table of contents

| | |
|--|----|
| Introduction | 4 |
| 1 Definitions | 4 |
| 2 General data format | 4 |
| 3 Communication with the xManager Server | 5 |
| 4 Welcome package | 5 |
| 5 Configuring session | 5 |
| 6 Server login..... | 6 |
| 7 Ping/pong | 6 |
| 8 Getting information on manager's rights..... | 7 |
| 9 Getting the list of accounts | 7 |
| 10 Getting information about selected account..... | 8 |
| 11 Getting an account calculated data | 10 |
| 12 Update account data, create new account | 11 |
| 13 Getting logged accounts list | 14 |
| 14 Getting list of account groups information..... | 14 |
| 15 Getting information about selected account groups..... | 15 |
| 16 Open an instant position on client behalf with any price – open position correction | 16 |
| 17 Close position on client behalf – close position correction | 17 |
| 18 Create order on client behalf | 18 |
| 19 Modify order on client behalf | 18 |
| 20 Delete order on client behalf | 20 |
| 21 Modify position on client behalf | 20 |
| 22 Close position on client behalf (including partial close) | 21 |
| 23 Add ledger operation | 21 |
| 24 Add correction ledger operation and change in closed_trades..... | 22 |
| 25 Get execution list | 22 |
| 26 Get position list | 23 |
| 27 Get order margin..... | 25 |
| 28 Get order list | 25 |
| 29 Get ledger list | 27 |
| 30 Getting closed trades IDs list..... | 28 |
| 31 Getting closed trades list (openings and closings after the execution) | 28 |
| 32 Change manager account password | 29 |
| 33 Change account password | 30 |
| 34 Block/Unblock account | 30 |
| 35 Enable/Disable trade for the account | 31 |
| 36 Get ledger type list | 31 |
| 37 Get order status list..... | 32 |
| 38 Get order side list | 32 |
| 39 Get account leverage list..... | 33 |
| 40 Get order origin list | 33 |
| 41 Get instrument list | 34 |
| 42 Get country list..... | 34 |
| 43 Get instrument quote type list..... | 35 |
| 44 Get order type list | 35 |



| | | |
|-------|--|----|
| 45 | Get position reference type list..... | 36 |
| 46 | Get free margin rule description list | 36 |
| 47 | Get free margin rule name list | 37 |
| 48 | Get mirror mode list..... | 37 |
| 49 | Get mirror book list | 38 |
| 50 | Get offer name list | 38 |
| 51 | [advanced] Retrieve the value of a server variable..... | 39 |
| 52 | Set a flag switching on/off push mode..... | 39 |
| 53 | Getting account report information..... | 41 |
| 54 | Check customer password | 42 |
| 55 | Get account instrument list..... | 43 |
| 56 | Instrument information..... | 43 |
| 57 | Position information..... | 44 |
| 58 | Order information | 45 |
| 59 | Block/unblock gross account..... | 45 |
| 60 | Getting server logs | 46 |
| 61 | Introducing Brokers (IB) | 47 |
| 61.1 | Getting IB Account list | 47 |
| 61.2 | Getting IB Master Account list | 47 |
| 61.3 | Getting a list of IB Accounts for a selected IB Master | 47 |
| 61.4 | Getting a list of accounts for a selected IB | 48 |
| 61.5 | Getting a list of IB ledger details | 48 |
| 61.6 | Getting a list of commissions for a selected IB | 49 |
| 61.7 | Overwriting the list of commission for a selected IB | 50 |
| 61.8 | Updating the list of commission for a selected IB | 50 |
| 61.9 | Getting a list of instrument class IDs | 51 |
| 61.10 | Getting information about instrument class | 51 |
| 61.11 | Getting a list of commission patterns for a selected IB | 52 |
| 61.12 | Getting a list of commission patterns..... | 52 |
| 61.13 | Getting information about commission pattern..... | 52 |
| 61.14 | Creating IB commission pattern..... | 53 |
| 61.15 | Getting IB account permissions from db glossary | 54 |
| 61.16 | Getting view permissions for IB account | 54 |
| 61.17 | Adding view permissions for IB account..... | 55 |
| 61.18 | Deleting view permissions for IB account..... | 55 |
| 62 | Getting a list of available currencies | 56 |
| 63 | Multi Account Manager (MAM)..... | 56 |
| 63.1 | Getting the list of available MAM commission calculation list..... | 56 |
| 63.2 | Getting the list of commission types for MAM success fees | 57 |
| 63.3 | Getting the list of MAM management fee types | 57 |
| 63.4 | Getting the list of MAM accounts..... | 58 |
| 63.5 | Getting MAM commission | 58 |
| 63.6 | Getting the list of accounts assigned to the MAM account | 59 |
| 63.7 | Updating MAM commission | 59 |
| 63.8 | Getting allocation method for MAM | 59 |
| 63.9 | Updating MAM account allocation method | 60 |
| 63.10 | Manual calculation of success fees..... | 60 |



Introduction

This document presents information on xManager Server communication protocol.

The communication protocol of the xManager Server uses JSON format.
JSON format used by the server doesn't allow extensions (e.g. comments, other flags).

JSON format standardization document is available under the following link:

RFC 4627: <http://tools.ietf.org/html/rfc4627>

1 Definitions

The following definitions will be used in this specification document:

simple type: type, which value is itself: int, int64, string, double, bool;

decimal type: the representation of decimal type as a simple string type;

j-value (JSON value): any simple type, j-object or j-array;

j-object (JSON object): a record containing any number of named j-values (pair <string, j-value>);

j-array (JSON array): an array where each element is j-value;

j-subvalue: j-value which is a component of a j-object.

A j-object can contain *zero* elements. A j-array can have *zero* length.

The name of j-value can be an empty string.

Encoding of strings is set to UTF-8. In this format the server sends and receives data.

Definition of unix-time:

Unix time, or POSIX time, is a system for describing points in time, defined as the number of seconds elapsed since midnight Coordinated Universal Time (UTC) of January 1, 1970.

2 General data format

Each packet consists of exactly one main, unnamed j-value.

The data stream consists of consecutive j-values, with no punctuation.

The main j-value is a j-object containing exactly two j-subvalues which are j-objects.

The first j-object is named *header* and consists of at least a field type as a simple type string. This is a packet type.

The header can also optionally contain a simple type field serial int, including user information, which will be duplicated in a packet header(s) of a direct server response to the sent packet. The purpose of this field is to give a possibility to identify server responds with requests (there may be many requests of the same type simultaneously processed in the server and it is important to associate the responses with the requests). If the request header doesn't contain the field serial, the server doesn't broadcast this field in responses.

The second j-subobject of the main packet j-object is named *data* and its content is specific for a given packet type. The specifications for different types of packages are described in the next chapter.



A sample of properly defined packet in xManager Protocol:

```
{
  "header": { "type": "login" }
  "data": {
    "login": "example_manager_name",
    "password": "test"
  }
}
```

3 Communication with the xManager Server

All communication with xManager Server is SSL encrypted.

4 Welcome package

Immediately after connecting to the server, a welcome packet containing version number and compilation is sent.

packet type: **welcome**

packet body:

| name | type | description |
|-----------|--------|-------------------------------|
| version | int | server version |
| timestamp | string | server compilation time stamp |

5 Configuring session

Connection parameters can be configured by configure session packet.

packet type: **configure session**

packet body:

| name | type | description |
|-------------|------|---|
| compression | bool | protocol compression set on/off |
| separator | bool | separator set on/off (two \n signs) between JSONs |

Server responds with configure session response packet with the following format:

packet type: **configure session response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the correct login process is *success*.



Note: compressed data is sent starting from the next packet following configure session response. Compressed data is in the following format: 4 bytes for 32-bit unsigned number (little-endian) indicating the size of compressed data. Next, there are 4 bytes for 32-bit unsigned number (little-endian) indicating the size of uncompressed data. Next, there is raw compressed data in LZ4 format. Each compressed packet may contain any non-zero number of consecutive complete packets in JSON format.

6 Server login

Only limited communication is allowed prior to user login to the server. Allowed commands: *login*, *ping* and *change manager account password*. Once the client is connected to the xManager Server, the server ignores all packets sent by the client, except for *login*, *ping* or *change manager account password*.

Correct communication with the server starts by sending logging packet:

packet type: **login**

packet body:

| name | type | description | |
|----------|--------|--|----------|
| login | string | logging manager name | |
| password | string | logging manager password | |
| banner | string | any inscription/note which identifies partner application name | optional |

Server responds with *login response* packet with the following format:

packet type: **login response**

packet body:

| name | type | description |
|-----------|--------|---|
| result | string | login result |
| read_only | bool | state of field is always defined, it determines whether the user is able to modify data |

The result of the correct login process is *success*.

If the manager's password is expired, the result of login process is *password has expired*. The manager cannot log in. In order to change current password to a new one, *change manager account password* command should be used.

Any other message points an error (e.g. *invalid account*, *forbidden* or other).

7 Ping/pong

If there is a need, the xManager client can send *ping* packet to check the connection. The *ping* packet does not contain any other information.

packet type: **ping**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |



The server shall always respond with a *pong* packet.

In case when *pong* packet is not received in a certain, rational time span, the connection may be considered as inactive and the communication as disconnected/broken.

In general, the connection is broken if any packet other than allowed is sent to xManager Server.

packet type: **pong**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

8 Getting information on manager's rights

packet type: **manager rights**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

The Server responds with a packet contains the list of manager rights (i.e. permissions).

packet type: **manager rights response**

packet body:

| name | type | description |
|--------|--------------------|------------------|
| rights | j-array right_info | access rights |
| result | string | operation result |

Each entry in the j-array *rights* is a j-object with the following structure:

j-object: **right_info** :

| name | type | description |
|-------------|--------|-------------------------------------|
| id_right | int | ID of a permission from db glossary |
| name | string | permission name |
| description | string | access right description |

9 Getting the list of accounts

Request for accounts list:

packet type: **account list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Once the server receives request packet, it responds with the list of all customers' account IDs, and it is represented as j-array *accounts* of a simple type *int*.



packet type: **account list response**

packet body:

| name | type | description |
|----------|-------------|--------------------------------|
| accounts | j-array int | list of customers' account IDs |
| result | string | operation result |

10 Getting information about selected account

packet type: **account info list**

packet body:

| name | type | description |
|----------|-------------|--------------------------------|
| accounts | j-array int | list of customers' account IDs |

The server responds with a packet containing complete account descriptions.

The number of entries in the j-array does not exceed the number of entries in the *account info list* query.

packet type: **account info list response**

packet body:

| name | type | description |
|----------|-----------------------------|------------------------------|
| accounts | j-array <i>account_info</i> | complete account description |
| result | string | operation result |

Each entry in the j-array *accounts* is a j-object with the following structure:

j-object: **account_info** :

| name | type | description | |
|----------------------|-------------|--|--|
| id_account | int | account identification number, from db | |
| login | int | login | |
| name | string | name | |
| surname | string | surname | |
| group | string | name of client group, chosen from a list of groups (see: group list packet, §13) | |
| country | string | name of country, chosen from glossary (see: country list packet, §42) | |
| city | string | city | |
| currency | string | currency name of client's group | |
| balance | decimal | available money for the account | |
| credit | decimal | available credit for the account (client's credit facility) | |
| equity | decimal | account equity, at the time of response packet generated | calculated |
| margin | decimal | account margin | calculated |
| free_margin | decimal | account free margin | calculated |
| regdate | int | unix-time UTC of the account registration date | |
| address | string | address | |
| state | string | state | |
| zip_code | string | zip-code | |
| email | string | e-mail | |
| bank_account | string | bank account number | |
| ref_account | string | bank account number of agent | |
| phone | string | phone number | |
| aggregate_group | string | name of the group of related accounts | deprecated, unused |
| enabled | bool | whether account is allowed to login to system | |
| trade_enable | bool | whether trading in the system is enabled for account | |
| daily_confirmation | bool | whether daily report is sent to client | currently unused, must be set to false |
| id_number | string | personal client identification number, e.g. national identity document | |
| is_gross | bool | whether account is in gross state | |
| is_leverage | int | account leverage ID, chosen from glossary (see: leverage list packet, §39) | |
| margin_level | decimal | client margin level, percentage (%) representation | calculated |
| account_types | j-array int | account types (<i>IB</i> , <i>MAM</i>). Definition of array contents: <ul style="list-style-type: none"> regular trading account – array is empty IB – array includes number 1 MAM – array includes number 3 | |
| id_ib_account | int | IB account ID, assigned to this account | optional |
| id_master_ib_account | int | master IB account ID, assigned to this account | optional |
| id_mam_account | int | MAM account ID, assigned to this account | optional |



11 Getting an account calculated data

packet type: **account info calc list**

packet body:

| name | type | description |
|----------|-------------|--------------------------------|
| accounts | j-array int | list of customers' account IDs |

The server responds with a packet containing calculated data for a selected account.

packet type: **account info calc list response**

packet body:

| name | type | description |
|---------------|----------------------------------|-------------------------|
| accounts_calc | j-array <i>account_calc_info</i> | account calculated data |
| result | string | operation result |

Each entry in the j-array *accounts_calc_info* is a j-object with the following structure:

j-object: ***account_calc_info*** :

| name | type | description | |
|--------------|---------|--|------------|
| id_account | int | account identification number | from db |
| balance | decimal | available money for the account | from db |
| equity | decimal | account equity, at the time of response packet generated | calculated |
| margin | decimal | account margin | calculated |
| free_margin | decimal | account free margin | calculated |
| margin_level | decimal | client margin level, percentage (%) representation | calculated |



12 Update account data, create new account

It is possible to update or modify the account data. To do so, a j-object with following structure needs to be sent:

packet type: update account info

packet body:

| name | type | description | |
|----------------------|-------------|--|--|
| id_account | int | account ID (from db), which data is being updated; account identification number | |
| name | string | name | |
| surname | string | surname | |
| group | string | name of client group, chosen from a list of groups (see: group list packet, §13) | |
| city | string | city | |
| address | string | address | |
| country | string | name of country, chosen from glossary (see: country list packet, §42) | |
| zip_code | string | zip-code | |
| state | string | state | |
| phone | string | phone number | |
| email | string | e-mail | |
| bank_account | string | bank account number | |
| ref_account | string | bank account number of agent | |
| id_number | string | personal client identification number, e.g. national identity document | |
| daily_confirmation | bool | whether daily report is sent to client | currently unused, must be set to false |
| aggregate_group | string | name of the group of related accounts | deprecated, unused, must be defined as empty, i.e.: "" |
| is_gross | bool | whether account is in gross state | |
| id_leverage | int | account leverage ID, chosen from glossary (see: leverage list packet, §39) | optional |
| account_types | j-array int | account types (<i>IB</i> , <i>MAM</i>). Definition of array contents: <ul style="list-style-type: none"> regular trading account – array is empty IB – array includes number 1 MAM – array includes number 3 | |
| id_ib_account | int | IB account ID, assigned to this account | optional |
| id_master_ib_account | int | master IB account ID, assigned to this account | optional |
| id_mam_account | int | MAM account ID, assigned to this account | optional |

The server responds with a confirmation message:



packet type: **update account info response**

packet body:

| name | type | description |
|--------|--------|---------------|
| result | string | update result |

The result of the successful update is *success*. Any other result indicates an error.

To create a new account packet with a following structure need to be sent:

packet type: **create account info**

packet body:

| name | type | description | |
|----------------------|-------------|--|--|
| name | string | name | |
| surname | string | surname | |
| group | string | name of client group, chosen from a list of groups (see: group list packet, §13) | |
| country | string | name of country, chosen from glossary (see: country list packet, §42) | |
| city | string | city | |
| address | string | address | |
| state | string | state | |
| zip_code | string | zip-code | |
| email | string | e-mail | |
| bank_account | string | bank account number | optional |
| ref_account | string | bank account number of agent | optional |
| phone | string | phone number | |
| enabled | bool | whether account is allowed to login to system | |
| trade_enable | bool | whether trading in the system is enabled for account | |
| password | string | client password | |
| daily_confirmation | bool | whether daily report is sent to client | currently unused, must be set to false |
| id_number | string | personal client identification number, e.g. national identity document | |
| aggregate_group | string | name of the group of related accounts | deprecated, unused, must be defined as empty, i.e.: "" |
| is_gross | bool | whether account is in gross state | |
| id_leverage | int | account leverage ID, chosen from glossary (see: leverage list packet, §39) | optional |
| account_types | j-array int | account types (<i>IB</i> , <i>MAM</i>). Definition of array contents: <ul style="list-style-type: none">regular trading account – array is emptyIB – array includes number 1MAM – array includes number 3 | |
| id_ib_account | int | IB account ID, assigned to this account | optional |
| id_master_ib_account | int | master IB account ID, assigned to this account | optional |
| id_mam_account | int | MAM account ID, assigned to this account | optional |



The server responds with result of the operation with the following packet structure.

packet type: **create account info response**

packet body:

| name | type | description | |
|------------|--------|---|-----------|
| result | string | result of create operation | |
| id_account | int | ID of created account - if the operation was successful; account identification number | generated |
| login | int | login of created account - if the operation was successful; | generated |
| password | string | password for the created account | |

The result of the successful operation is *success*. Any other result indicates an error.

Server generates: ID number, login and password automatically.



13 Getting logged accounts list

The manager can get a list of logged accounts.

If the packet body is sent as empty (i.e. does not contain optional field), the server sends the list of all logged accounts. Otherwise the server sends the list of accounts from the given range.

packet type: **logged account list**

packet body:

| name | type | description | |
|----------|-------------|--------------------------|----------|
| accounts | j-array int | the list of accounts IDs | optional |

The server responds with the result of operation with the list of logged accounts IDs:

packet type: **logged account list response**

packet body:

| name | type | description | |
|----------|-------------|---------------------------------|--|
| result | string | operation result | |
| accounts | j-array int | the list of logged accounts IDs | |

The result of the successful operation is *success*. Any other result indicates an error.

14 Getting list of account groups information

A Manager connected to xManager Server may get a list of all account groups.

In order to get a list of account groups, the *group list* packet needs to be sent, with no other data.

packet type: **group list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Once the server receives this request it generates a list of all account groups and sends back the answer in a *group list response* packet. The packet contains a field named *groups*, which is a j-array of j-objects *group info*.

packet type: **group list response**

packet body:

| name | type | description |
|--------|---------------------------|------------------|
| groups | j-array <i>group_info</i> | customer groups |
| result | string | operation result |

Each element of the table is of the following type:



j-object: **group_info**:

| name | type | description |
|--------------------|---------|---|
| id_group | int | clients group number |
| name | string | group name |
| country | string | group country |
| currency | string | currency name of clients group |
| max_trade_nominal | decimal | maximum nominal amount of open positions from clients group, in EUR |
| interest_rate | decimal | interest rate for clients group, percentage (%) representation |
| tax | decimal | taxation, percentage (%) representation |
| initial_balance | decimal | initial balance, setting by default, for accounts being created in such a group |
| margin_call_level | int | margin call level |
| stop_out_level | int | stop out level |
| freemargin_rule | string | free margin rule |
| automation_allowed | bool | whether automation for order acceptance is allowed |

15 Getting information about selected account groups

packet type: **account group info list**

packet body:

| name | type | description |
|----------------|-------------|--------------------------------------|
| account_groups | j-array int | list of customers' account group IDs |

The server responds with a packet containing complete account group descriptions.

The number of entries in the j-array does not exceed the number of entries in the *account group info list* query.

packet type: **account group info list response**

packet body:

| name | type | description |
|----------------|----------------------------|------------------------------------|
| account_groups | j-array account_group_info | complete account group description |
| result | string | operation result |

Each entry in the j-array *account_groups* is a j-object with the following structure:



j-object: **account_group_info**:

| name | type | description | |
|-------------------------|---------|---|----------|
| id_group | int | account group identification number, from db | |
| name | string | group name | |
| country | string | group country | optional |
| currency | string | currency name of clients group | |
| max_trade_nominal | decimal | maximum nominal amount of open positions from clients group, in EUR | |
| interest_rate | decimal | interest rate for clients group, percentage (%) representation | |
| tax | decimal | taxation, percentage (%) representation | |
| initial_balance | decimal | initial balance setting by default for accounts being created in such a group | |
| margin_call_level | int | margin call level | |
| stop_out_level | int | stop out level | |
| id_freemargin_rule | int | free margin rule ID | |
| automation_allowed | bool | whether automation for order acceptance is allowed | |
| id_id_default_leverage | int | default account group leverage ID | |
| id_mirror_account | int | mirror account ID | optional |
| max_dma_nominal | decimal | maximum nominal amount of open DMA positions from clients' group, in EUR | |
| calc_swap | bool | whether swaps are calculated | |
| id_mirror_mode | int | mirror mode ID | optional |
| id_mirror_book | int | mirror book ID | optional |
| account_validity_period | int | time, after which account will be automatically disabled (in milliseconds) | |

16 Open an instant position on client behalf with any price – open position correction

For correction purpose a position may be opened on client account.



packet type: **open position correction**

packet body:

| name | type | description | |
|-----------------|---------|--|----------|
| instrument | string | instrument name | |
| id_dict_quote | int | ID of quote type, (see: quote list packet, §43) | |
| id_dict_side | int | transaction side | |
| id_account | int | client account number | |
| price | decimal | price | |
| nominal | decimal | nominal for execution | |
| calc_swap | decimal | calculated, accrued swap | optional |
| calc_commission | decimal | calculated, accrued commission | optional |
| take_profit | decimal | take profit | optional |
| stop_loss | decimal | stop loss | optional |
| margin | decimal | margin, in order to get suggestion for margin value, use: calc margin packet, §27 | |
| comment | string | comment for operation | optional |

The server responds with a confirmation message:

packet type: **open position correction response**

packet body:

| name | type | description |
|--------|--------|-------------------------------------|
| result | string | result of open correction operation |

The result of the successful update is *success*. Any other result indicates an error.

17 Close position on client behalf – close position correction

On client account a position may be closed, for correction purpose.

packet type: **close position correction**

packet body:

| name | type | description |
|-------------|---------|------------------------------------|
| id_position | int | position ID |
| nominal | decimal | nominal chosen for close operation |

The server responds with a confirmation message:

packet type: **close position correction response**

packet body:

| name | type | description |
|--------|--------|--------------------------------------|
| result | string | result of close correction operation |

The result of the successful update is *success*. Any other result indicates an error.



18 Create order on client behalf

On client account an order may be created.

packet type: **create order**

packet body:

| name | type | description | |
|------------------|---------|----------------------------------|----------|
| login | string | client login | |
| side | string | transaction side | |
| id_instrument | int | instrument id (instrument table) | |
| id_quote | int | quote type id | |
| instrument_name | string | instrument name | |
| activation_price | decimal | activation price | optional |
| comment | string | comment for operation | |
| tp | decimal | take profit price | optional |
| sl | decimal | stop loss price | optional |
| order_type | string | order type: market, limit | |
| expiry_time | int | order expiry time | optional |

Server responds with a packet:

packet type: **create order response**

packet body:

| name | type | description |
|------------|---------|------------------------|
| result | string | operation result |
| id_order | int | order id |
| open_price | decimal | transaction open price |

The result of the successful update is *success*. Any other result indicates an error.

19 Modify order on client behalf

On client account an order may be modified.



packet type: **modify order**

packet body:

| name | type | description | |
|------------------|---------|----------------------------------|----------|
| id_order | int | modified order id | |
| login | string | client login | |
| side | string | transaction side | |
| id_instrument | int | instrument id (instrument table) | |
| id_quote | int | quote type id | |
| instrument_name | string | instrument name | |
| nominal | decimal | nominal | |
| price | decimal | price | |
| activation_price | decimal | activation price | optional |
| comment | string | comment for operation | |
| tp | decimal | take profit price | optional |
| sl | decimal | stop loss price | optional |
| order_type | string | order type: market, limit | |
| expiry_time | int | order expiry time | optional |

Server responds with a packet:

packet type: **modify order response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful update is *success*. Any other result indicates an error.



20 Delete order on client behalf

On client account an order may be deleted.

packet type: **delete order**

packet body:

| name | type | description |
|----------|------|------------------|
| id_order | int | deleted order id |

Server responds with a packet:

packet type: **delete order response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful update is *success*. Any other result indicates an error.

21 Modify position on client behalf

On client account a position may be modified.

packet type: **modify position**

packet body:

| name | type | description | |
|-----------------|---------|----------------------------------|----------|
| id_position | int | modified position id | |
| login | string | client login | |
| side | string | transaction side | |
| id_instrument | int | instrument id (instrument table) | |
| id_quote | int | quote type id | |
| instrument_name | string | instrument name | |
| nominal | decimal | nominal | |
| price | decimal | price | |
| comment | string | comment for operation | |
| tp | decimal | take profit price | optional |
| sl | decimal | stop loss | optional |
| order_type | string | order type: market, limit | |
| expiry_time | int | order expiry time | optional |

Server responds with a packet:

packet type: **modify position response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |



The result of the successful update is *success*. Any other result indicates an error.

22 Close position on client behalf (including partial close)

On client account a position may be closed (also partially).

packet type: **close position**

packet body:

| name | type | description | |
|-------------|---------|-----------------------|----------|
| id_position | int | closed position id | |
| nominal | decimal | nominal | |
| price | decimal | price | |
| comment | string | comment for operation | optional |

Server responds with a packet:

packet type: **close position response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful update is *success*. Any other result indicates an error.

23 Add ledger operation

Deposits, withdrawals, corrections, credit facilities can be performed on a customer account.

packet type: **ledger operation**

packet body:

| name | type | description | |
|--------------------|---------|--|----------|
| id_account | int | account ID number | |
| login | int | customer login | |
| ledger_type | string | ledger type: <i>deposit, withdrawal, credit in, credit out, correction</i> | |
| amount | decimal | operation amount in account currency | |
| comment | string | comment for operation | optional |
| id_corrected_entry | int | corrected entry ID (in case when ledger_type is <i>correction</i>) | optional |
| instrument_name | string | instrument name | optional |
| id_dict_quote | int | quote type ID | optional |
| expdate | int | expiration date (for ledger_type <i>credit</i>) | optional |

Server responds with a packet:



packet type: **ledger operation response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is *success*. Any other result indicates an error.

24 Add correction ledger operation and change in closed_trades

packet type: **correction ledger operation**

packet body:

| name | type | description |
|-----------------|---------|---|
| id_closed_trade | int | closed trade ID |
| swap | decimal | swap correction amount in account currency |
| profit | decimal | profit correction amount in account currency |
| commission | decimal | commission correction amount in account currency |
| comment | string | operation comment, will be displayed in <i>ledgers</i> and <i>closed trades</i> |

Server responds with a packet:

packet type: **correction ledger operation response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is *success*. Any other result indicates an error.

25 Get execution list

packet type: **execution list**

packet body:

| name | type | description | |
|---------------|-------------|---|----------|
| from_datetime | int | closed beginning of the interval from which the list of executions is to be generated | optional |
| to_datetime | int | open end of the interval from which the list of executions is to be generated | optional |
| accounts | j-array int | account IDs list | optional |
| executions | j-array int | execution IDs list | optional |

There are three combination of fields, which are being operated:

1. from_datetime + to_datetime
2. from_datetime + to_datetime + accounts
3. executions



The server responds with result of the operation with following packet structure.

packet type: **execution list response**

packet body:

| name | type | description | |
|------------|------------------------|----------------------------------|----------|
| executions | j-array execution_info | array of executions | |
| result | string | operation result | |
| valid from | int | date since when data is in vault | optional |
| valid to | int | date until when data is in vault | optional |

The result of the successful update is success. Any other result indicates an error.

Each element of the array has the following structure:

j-object: **execution_info**:

| name | type | description | |
|------------------|---------|--|----------|
| id_execution | int | execution ID number | |
| executor_number | int | executor number | |
| execution_number | int | execution number | |
| id_order | int | order ID number, to which execution is added | |
| nominal | decimal | execution nominal | |
| comment | string | comment for operation | optional |
| price | decimal | execution price | |
| side | string | 'buy' or 'sell' | |
| instrument | string | instrument name | optional |
| id_dict_quote | int | quote type ID | |
| login | int | login | |
| group | string | account group name | |
| currency | string | instrument currency name | |
| datetime | int | execution time | |
| user_login | string | login of user who accepted operation | optional |

26 Get position list

packet type: **position list**

packet body:

| name | type | description | |
|-----------|-------------|----------------------|----------|
| accounts | j-array int | list of account IDs | optional |
| positions | j-array int | list of position IDs | optional |

One of the fields (accounts, positions) need to be set.

Server responds with a packet:



packet type: **position list response**

packet body:

| name | type | description |
|-----------|------------------------------|--------------------|
| positions | j-array <i>position_info</i> | array of positions |
| result | string | operation result |

Each element of the array has the following structure:

j-object: ***position_info*** :

| name | type | description | |
|-----------------|--------------------------|--|------------|
| id_position | int | position ID | |
| login | int | login | |
| name | string | name | |
| surname | string | surname | |
| group | string | account group name | |
| id_execution | int | execution ID | |
| nominal | decimal | execution nominal | |
| create_time | int | position creation time | |
| update_time | int | position updated time | |
| open_price | decimal | average open price | |
| side | string | 'buy' or 'sell' | |
| instrument | string | instrument name | |
| id_dict_quote | int | quote type ID | |
| id_account | int | account ID | |
| id_order_TP | int | ID order related TP | optional |
| id_order_SL | int | ID order related SL | optional |
| price_TP | decimal | TP price | optional |
| price_SL | decimal | SL price | optional |
| market_price | decimal | current instrument price | calculated |
| swap | decimal | swaps accrued by now | calculated |
| commission | decimal | commission accrued by now | calculated |
| profit | decimal | profit/loss at the time of response generated | calculated |
| position_margin | decimal | margin calculated for this position, for further processing and calculation of client margin level | |
| rollover | decimal | points calculated, accrued from order rollover by now | calculated |
| comment_clnt | string | customer comment | |
| currency | string | profit currency name = account currency | |
| profit_conv | j-array <i>conv_info</i> | profit/loss at the time of response generated, in currency of calculation | calculated |
| swap_conv | j-array <i>conv_info</i> | swaps calculated, accrued by now, in currency of calculation | calculated |
| commission_conv | j-array <i>conv_info</i> | commission calculated, accrued by now, in currency of calculation | calculated |
| id_order | int | ID of order which leads to position opening | |



j-object: **conv_info** :

| name | type | description |
|-------|---------|--|
| key | string | currency name |
| value | decimal | value represented in listed key currency |

27 Get order margin

packet type: **calc margin**

packet body:

| name | type | description |
|---------------|---------|-------------------|
| instrument | string | instrument name |
| id_dict_quote | int | quote type ID |
| id_account | int | account ID number |
| price | decimal | price |
| nominal | decimal | execution nominal |

Server responds with a packet:

packet type: **calc margin response**

packet body:

| name | type | description | |
|--------|---------|------------------|------------|
| margin | decimal | order margin | calculated |
| result | string | operation result | |

28 Get order list

packet type: **order list**

packet body:

| name | type | description | |
|---------------|-------------|---|----------|
| from_datetime | int | closed beginning of the interval from which the list of orders is to be generated | optional |
| to_datetime | int | open end of the interval from which the list of orders is to be generated | optional |
| accounts | j-array int | list of accounts IDs | optional |
| orders | j-array int | list of order IDs | optional |

The following combinations of fields are available:

1. from_datetime + to_datetime + accounts
2. orders

Server responds with a packet:



packet type: **order list response**

packet body:

| name | type | description | |
|------------|---------------------------|----------------------------------|----------|
| orders | j-array <i>order_info</i> | array of orders | |
| result | string | operation result | |
| valid from | int | date since when data is in vault | optional |
| valid to | int | date until when data is in vault | optional |

Each element of the array has the following structure:

j-object: **order_info** :

| name | type | description | |
|----------------------|---------|--|----------|
| id_order | int | order ID | |
| mirror_root_id_order | int | ID of root order, that this order is mirrored from | optional |
| login | int | login | |
| name | string | name | |
| surname | string | surname | |
| group | string | account group name | |
| nominal | decimal | nominal of execution | |
| order_type | string | order type: <i>'market', 'limit'</i> | |
| order_status | string | order status: <i>'pending new', 'new', 'accepted', 'rejected', 'requoted', 'canceled', 'expired', 'filled', 'partial filled'</i> | |
| filled_nominal | decimal | filled order nominal | |
| open_time | int | order adding time | |
| execution_time | int | order execution time | optional |
| price | decimal | order price | |
| activation_price | decimal | order activation price | optional |
| expiry_time | int | order expiration time | optional |
| margin | decimal | order margin | |
| order_origin | string | order origin: <i>'manual', 'auto', 'by other (dealer)'</i> | |
| slippage | int | order slippage, amount in pips | |
| main_order_id | int | main order ID (for related orders: TP/SL) | optional |
| side | string | <i>'buy', 'sell'</i> | |
| id_account | string | account ID | |
| takeprofit_price | decimal | TP price of the order | optional |
| stoploss_price | decimal | SL price of the order | optional |
| id_position | int | ID of position which this order is TP/SL | optional |
| instrument | string | instrument name | |
| id_dict_quote | int | quote type ID | |
| comment_clnt | string | customer comment | |
| comment_sys | string | system comment | |



29 Get ledger list

packet type: **ledger list**

packet body:

| name | type | description |
|---------------|-------------|--|
| from_datetime | int | closed beginning of the interval from which the ledger list is to be generated |
| to_datetime | int | open end of the interval from which the ledger list is to be generated |
| accounts | j-array int | list of accounts ID |

Server responds with a packet:

packet type: **ledger list response**

packet body:

| name | type | description | |
|------------|----------------------------|----------------------------------|----------|
| ledgers | j-array <i>ledger_info</i> | array of ledgers | |
| result | string | operation result | |
| valid from | int | date since when data is in vault | optional |
| valid to | int | date until when data is in vault | optional |

Each element of the array has the following structure:

j-object: **ledger_info** :

| name | type | description | |
|-------------------|---------|---|----------|
| id_ledger | int | ledger ID | |
| login | int | login | |
| name | string | name | |
| surname | string | surname | |
| group | string | account group name | |
| currency | string | group currency name | |
| type | string | ledger type: 'deposit', 'withdrawal', 'close trade', 'correction', 'interest', 'tax', 'swap', 'rolling', 'dividend', 'acquire right', 'spinoff', 'provision', 'credit in', 'credit out' | |
| time | int | ledger time | |
| debit | decimal | debit | |
| credit | decimal | credit | |
| id_corr_entry | int | corrected ledger ID | optional |
| comment | string | comment | optional |
| correction_amount | decimal | ledger correction amount | optional |
| expdate | int | expiration date (for ledger_type credit) | optional |



30 Getting closed trades IDs list

packet type: **closed trade id list**

packet body:

| name | type | description |
|---------------|-------------|--|
| from_datetime | int | closed beginning of the interval from which the list of closed trades IDs is to be generated |
| to_datetime | int | open end of the interval from which the list of closed trades IDs is to be generated |
| accounts | j-array int | list of account IDs |

Server responds with a packet:

packet type: **closed trade id list response**

packet body:

| name | type | description | |
|------------------|-------------|----------------------------------|----------|
| closed_trade ids | j-array int | array of closed trades IDs | |
| result | string | operation result | |
| valid from | int | date since when data is in vault | optional |
| valid to | int | date until when data is in vault | optional |

31 Getting closed trades list (openings and closings after the execution)

packet type: **closed trade list**

packet body:

| name | type | description | |
|---------------|-------------|--|----------|
| closed trades | j-array int | array of closed trades IDs | optional |
| from_datetime | int | closed beginning of the interval from which the list of closed trades is to be generated | optional |
| to_datetime | int | open end of the interval from which the list of closed trades is to be generated | optional |
| accounts | j-array int | list of account IDs | optional |

There are two combinations of fields, which are being operated:

1. closed trades
2. from_datetime + to_datetime + accounts

Server responds with a packet:



packet type: **closed trade list response**

packet body:

| name | type | description | |
|---------------|-----------------------------------|---|----------|
| closed_trades | j-array <i>closed_trades_info</i> | connection between opening and closing of transaction | |
| result | string | operation result | |
| valid from | int | date since when data is in vault | optional |
| valid to | int | date until when data is in vault | optional |

Each element of the array has the following structure:

j-object: ***closed_trades_info*** :

| name | type | description | |
|-----------------|---------|---|----------|
| id_closed_trade | int | closed trade ID | |
| id_instrument | int | instrument ID (<i>instrument</i> table) | |
| instrument | string | instrument name | |
| id_dict_quote | int | quote type ID | |
| side | string | 'buy' or 'sell'; glossary name | |
| nominal | decimal | execution nominal; settlement.nominal | |
| id_position | int | closed position ID | |
| close_id_order | int | order ID that closed the position | |
| open_id_order | int | | |
| open_time | int | open time (opening execution); execution.create_datetime | |
| close_time | int | close time (closing execution); execution.create_datetime (settlement.executionid_execution_close) | |
| open_price | decimal | open price; execution.exec_price | |
| close_price | decimal | close price; execution.exec_price (settlement.executionid_execution_close) | |
| tp | decimal | take profit price - opening execution; execution.orderid_order -> order.price | optional |
| sl | decimal | stop loss price - closing execution; execution.orderid_order -> order.price | optional |
| swap | decimal | ledger.settlementid_settlement -> ledger.debit or credit where type = 'swap' | |
| commission | decimal | ledger.settlementid_settlement -> ledger.debit or credit where type = 'commission' | |
| profit_loss | decimal | ledger.settlementid_settlement -> ledger.debit or credit where type = 'close trade' | |
| comment | string | closed trades comment | optional |
| id_account | int | account ID from db | |
| currency | string | profit currency = account currency (from the glossary) | |
| name | string | account owner name | |
| surname | string | account owner surname | |
| group_name | string | account group name | |

32 Change manager account password

Manager account password can be change.



packet type: **change manager account password**

packet body

| name | type | description |
|--------------|--------|---------------------------------------|
| login | int | manager name |
| old_password | string | old, current manager account password |
| new_password | string | new manager account password |

Server responds with a packet:

packet type: **change manager account password response**

packet body

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is success. Any other result indicates an error.

33 Change account password

Account password can be changed.

packet type: **set account password**

packet body:

| name | type | description |
|----------|--------|----------------------|
| login | int | account number |
| password | string | account new password |

Server responds with a packet:

packet type: **set account password response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is *success*. Any other result indicates an error.

34 Block/Unblock account

A manager can block or unblock the account.

packet type: **set enable account**

packet body:

| name | type | description |
|------------|------|--|
| id_account | int | customer account ID |
| enable | bool | true - <i>unblock</i> - <i>enable trade</i> , false - <i>block</i> |



Server responds with a packet:

packet type: **set enable account response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is *success*. Any other result indicates an error.

35 Enable/Disable trade for the account

A manager can enable or disable trade for the account.

packet type: **set trade_enable account**

packet body:

| name | type | description |
|--------------|------|--|
| id_account | int | customer account ID |
| trade_enable | bool | true - <i>unblock</i> , false - <i>block</i> |

Server responds with a packet:

packet type: **set trade_enable account response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is *success*. Any other result indicates an error.

36 Get ledger type list

packet type: **ledger type list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **ledger type list response**

packet body:

| name | type | description |
|--------------|---------------------------------|-----------------------|
| ledger_types | j-array <i>ledger_type_info</i> | array of ledger types |
| result | string | operation result |

Each element of the array has the following structure:



j-object: *ledger_type_info* :

| name | type | description |
|------------|--------|------------------|
| id_ledtype | int | type ID |
| name | string | ledger type name |

37 Get order status list

packet type: **order status list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **order status list response**

packet body:

| name | type | description |
|--------------|----------------------------------|-------------------------|
| order_status | j-array <i>order_status_info</i> | array of order statuses |
| result | string | operation result |

Each element of the array has the following structure:

j-object: *order_status_info* :

| name | type | description |
|----------|--------|-------------------|
| id_state | int | status ID |
| name | string | order status name |

38 Get order side list

packet type: **order side list**

pocket body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **order side list response**

pocket body:

| name | type | description |
|------------|--------------------------------|----------------------|
| order_side | j-array <i>order_side_info</i> | array of order sides |
| result | string | operation result |

Each element of the array has the following structure:



j-object: **order_side_info** :

| name | type | description |
|---------|--------|----------------------------|
| id_side | int | side ID |
| name | string | order side name (buy/sell) |

39 Get account leverage list

packet type: **leverage list**

pocket body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **leverage list response**

pocket body:

| name | type | description |
|----------|------------------------------|----------------------|
| leverage | j-array leverage_info | leverage information |
| result | string | operation result |

Each element of the array has the following structure:

j-object: **leverage_info** :

| name | type | description |
|-------------|--------|----------------------|
| id_leverage | int | id leverage |
| name | string | leverage description |

40 Get order origin list

packet type: **order origin list**

pocket body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **order origin list response**

packet body:

| name | type | description |
|--------------|----------------------------------|------------------------|
| order_origin | j-array order_origin_info | array of order origins |
| result | string | operation result |

Each element of the array has the following structure:



j-object: **order_origin_info** :

| name | type | description |
|-----------|--------|-------------------|
| id_origin | int | ID origin |
| name | string | order origin name |

41 Get instrument list

packet type: **instrument list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **instrument list response**

packet body:

| name | type | description |
|-------------|--------------------------------|------------------|
| instruments | j-array <i>instrument_info</i> | instrument info |
| result | string | operation result |

Each element of the array has the following structure:

j-object: **instrument_info** :

| name | type | description |
|---------------|--------|---|
| id_instrument | int | instrument ID (<i>instrument</i> table) |
| name | string | instrument name (<i>instrument base</i> table) |
| id_dict_quote | int | quote type ID |
| precision | int | instrument precision |

42 Get country list

packet type: **country list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **country list response**

packet body:

| name | type | description |
|-----------|-----------------------------|-------------------|
| countries | j-array <i>country_info</i> | list of countries |
| result | string | operation result |

Each element of the array has the following structure:



j-object: *country_info* :

| name | type | description |
|------------|--------|--------------|
| id_country | int | country ID |
| name | string | country name |
| code | string | country code |

43 Get instrument quote type list

packet type: **quote list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **quote list response**

packet body:

| name | type | description |
|--------|---------------------------|---------------------|
| quotes | j-array <i>quote_info</i> | quote type glossary |
| result | string | operation result |

Each element of the array has the following structure:

j-object: *quote_info* :

| name | type | description |
|---------------|--------|---------------|
| id_dict_quote | int | quote type ID |
| name | string | type name |

44 Get order type list

packet type: **order type list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **order type list response**

packet body:

| name | type | description |
|--------|--------------------------------|---------------------|
| types | j-array <i>order_type_info</i> | order type glossary |
| result | string | operation result |

Each element of the array has the following structure:



j-object: **order_type_info** :

| name | type | description |
|--------------------|--------|---------------|
| id_dict_order_type | int | order type ID |
| name | string | type name |

45 Get position reference type list

packet type: **position reference type list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **position reference type list response**

packet body:

| name | type | description |
|--------------------------|--------------------------------------|----------------------------------|
| position_reference_types | j-array position_reference_type_info | position reference type glossary |
| result | string | operation result |

Each element of the array has the following structure:

j-object: **position_reference_type_info**:

| name | type | description |
|------|--------|----------------------------|
| id | int | position reference type id |
| name | string | type name |

46 Get free margin rule description list

packet type: **free margin rule description list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **free margin rule description list response**

packet body:

| name | type | description |
|-------------------------------|--------------------------------------|---------------------------------------|
| free_margin_rule_descriptions | j-array free_margin_rule_description | free margin rule description glossary |
| result | string | operation result |

Each element of the array has the following structure:



j-object: *free_margin_rule_description*:

| name | type | description |
|------|--------|---------------------------------|
| id | int | free margin rule description id |
| name | string | type name |

47 Get free margin rule name list

packet type: **free margin rule name list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **free margin rule name list response**

packet body:

| name | type | description |
|------------------------|-------------------------------|--------------------------------|
| free_margin_rule_names | j-array free_margin_rule_name | free margin rule name glossary |
| result | string | operation result |

Each element of the array has the following structure:

j-object: *free_margin_rule_name*:

| name | type | description |
|------|--------|--------------------------|
| id | int | free margin rule name id |
| name | string | type name |

48 Get mirror mode list

packet type: **mirror mode list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **mirror mode list response**

packet body:

| name | type | description |
|--------------|---------------------|----------------------|
| mirror_modes | j-array mirror_mode | mirror mode glossary |
| result | string | operation result |

Each element of the array has the following structure:



j-object: **mirror_mode:**

| name | type | description |
|------|--------|----------------|
| id | int | mirror mode id |
| name | string | type name |

49 Get mirror book list

packet type: **mirror book list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **mirror book list response**

packet body:

| name | type | description |
|--------------|---------------------|----------------------|
| mirror_books | j-array mirror_book | mirror book glossary |
| result | string | operation result |

Each element of the array has the following structure:

j-object: **mirror_book:**

| name | type | description |
|------|--------|----------------|
| id | int | mirror book id |
| name | string | type name |

50 Get offer name list

packet type: **offer name list**

packet body:

| name | type | description |
|---------|------|-------------|
| <empty> | | |

packet type: **offer name list response**

packet body:

| name | type | description |
|-------------|--------------------|---------------------|
| offer_names | j-array offer_name | offer name glossary |
| result | string | operation result |

Each element of the array has the following structure:



j-object: **offer_name**:

| name | type | description |
|------|--------|---------------|
| id | int | offer name id |
| name | string | type name |

51 [advanced] Retrieve the value of a server variable

There is a set of set of variables in the server. It is possible to retrieve value of a variable by sending a message of type *variable*.

packet type: **variable**

packet body:

| name | type | description |
|------|--------|---------------|
| name | string | variable name |

The server responds with a following packet:

packet type: **variable response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |
| name | string | variable name |
| value | string | variable value |

The result of the successful operation is *success*. Any other result indicates an error (e.g. *forbidden*).

52 Set a flag switching on/off push mode

packet type: **set push**

packet body:

| name | type | description |
|--------|------|--------------------------|
| enable | bool | flag switching push mode |

Server responds with a packet:

packet type: **set push response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is *success*. Any other result indicates an error.

When push mode is switching on, client might get the following data packet/announcement:



packet type: **push position**

packet body:

| name | type | description | |
|----------|----------------------------------|---|----------|
| type | string | information type: 'information', 'create or modify', 'delete' | |
| position | j-object <i>position_info</i> | an object information to which push is related | optional |
| key | int | key identifying position | optional |

packet type: **push order**

packet body:

| name | type | description | |
|-------|-------------------------------|---|----------|
| type | string | information type: 'information', 'create or modify', 'delete' | |
| order | j-object <i>order_info</i> | an object information to which push is related | optional |
| key | int | key identifying order | optional |

packet type: **push execution response status**

packet body:

| name | type | description | |
|---------------------------|--|---|----------|
| type | string | information type: 'information', 'create or modify', 'delete' | |
| execution response status | j-object <i>execution_response_status</i> | an object information to which push is related | optional |
| key | int | key identifying execution | optional |

j-object: **execution_response_status** :

| name | type | description |
|----------------|--------|----------------------------------|
| id_order | int | order id number |
| id_status | int | result code (accept/reject/etc.) |
| result_code | int | error code |
| id_account | int | id of account bound to order |
| price | string | execution price |
| comment | string | comment |
| customer_value | string | customer comment |

packet type: **push account**

packet body:

| name | type | description | |
|---------|---------------------------------|---|----------|
| type | string | information type: 'information', 'create or modify', 'delete' | |
| account | j-object <i>account_info</i> | an object information to which push is related | optional |
| key | int | key identifying account | optional |



53 Getting account report information

packet type: **account report info list**

packet body:

| name | type | description |
|---------------|-------------|---|
| accounts | j-array int | list of accounts identification numbers |
| from_datetime | int | closed beginning of the interval from which the report is to be generated |
| to_datetime | int | open end of the interval from which the report is to be generated |

Server responds with a packet containing full account description. The number of entries in j-array does not exceed the number of entries w *account info list* query.

packet type: **account report info list response**

packet body:

| name | type | description | |
|-----------------|-------------------------------|----------------------------------|----------|
| accounts_report | j-array <i>account_report</i> | account report | |
| result | string | operation result | |
| valid from | int | date since when data is in vault | optional |
| valid to | int | date until when data is in vault | optional |

Each entry in j-array *accounts_report* is a j-object with a following structure:

j-object: ***account_report*** :

| name | type | description |
|-----------------|---------|--|
| id_account | int | account ID, from db |
| create_datetime | int | report time |
| name | string | name |
| surname | string | surname |
| group | string | customer group name |
| currency | string | currency glossary name |
| deposit | decimal | deposits from the last day |
| withdrawal | decimal | withdrawals from the last day |
| profit_loss | decimal | profit/loss at the end of the day: equity-(balance+credit) |
| balance | decimal | amount available for the account |
| equity | decimal | customer equity at the time of response generated |
| margin | decimal | customer margin |
| free_margin | decimal | customer free margin |
| credit | decimal | account credit (customer credit facility) |



54 Check customer password

packet type: **check account password**

packet body:

| name | type | description |
|----------|--------|---------------------------|
| login | int | customer login |
| password | string | customer account password |

Server responds with a packet:

packet type: **check account password response**

packet body

| name | type | description |
|------------|--------|-----------------------------|
| result | string | operation result |
| is_correct | bool | true when password is valid |

The result of the successful operation is *success*. Any other result indicates an error.



55 Get account instrument list

packet type: **account instrument list**

packet body:

| name | type | description |
|------------|------|---------------------|
| id_account | int | customer account ID |

Server responds with a packet:

packet type: **account instrument list response**

packet body

| name | type | description |
|-------------|--|------------------|
| instruments | j-array <i>account_instrument_info</i> | instrument info |
| result | string | operation result |

Each entry in the j-array *instruments* is a j-object with the following structure:

j-object: ***account_instrument_info*** :

| name | type | description |
|---------------|---------|---|
| id_instrument | int | instrument ID (<i>instrument</i> table) |
| name | string | instrument name (<i>instrument base</i> table) |
| id_dict_quote | int | quote type id |
| precision | int | instrument precision |
| min_lots | decimal | minimum number of lots |
| max_lots | decimal | maximum number of lots |
| step_lots | decimal | step |

56 Instrument information

Getting instrument information.

packet type: **instrument info**

packet body

| name | type | description |
|---------------|--------|---|
| id_instrument | int | instrument ID (<i>instrument</i> table) |
| name | string | instrument name (<i>instrument base</i> table) |
| id_dict_quote | int | quote type id |
| id_account | int | account id |

Server responds with a packet:



packet type: **instrument info response**

packet body

| name | type | description | |
|---------------|--------|---|------------|
| result | string | operation result | |
| id_instrument | int | instrument ID (<i>instrument</i> table) | |
| name | string | instrument name (<i>instrument base</i> table) | |
| id_dict_quote | int | quote type id | deprecated |
| id_account | int | account id | |
| bid | double | bid price | optional |
| ask | double | ask price | optional |
| mid | double | mid price | optional |
| last | int | last quote time | optional |
| digits | int | number of decimal places | deprecated |

Optional fields are returned only if there was a quote for an instrument.
The result of the successful operation is success. Any other result indicates an error.

57 Position information

Getting position information.

packet type: **position info**

packet body

| name | type | description |
|-------------|------|-------------|
| id_position | int | position ID |

Server responds with a packet:

packet type: **position info response**

packet body

| name | type | description | |
|----------|------------------------------|--------------------------------|----------|
| result | string | operation result | |
| position | j-array <i>position_info</i> | information about the position | optional |

The result of the successful operation is success. Any other result indicates an error.



58 Order information

Getting order information.

packet type: **order info**

packet body

| name | type | description |
|----------|------|-------------|
| id_order | int | order ID |

Server responds with a packet:

packet type: **order info response**

packet body

| name | type | description | |
|--------|---------------------------|-------------------|----------|
| result | string | operation result | |
| order | j-array <i>order_info</i> | order information | optional |

The result of the successful operation is *success*. Any other result indicates an error.

59 Block/unblock gross account

Setting account as gross or net.

packet type: **set gross account**

packet body

| name | type | description |
|------------|------|---|
| id_account | int | account ID |
| gross | bool | <i>true</i> – gross, <i>false</i> – net |

Server responds with a packet:

packet type: **set gross account response**

packet body

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is *success*. Any other result indicates an error.



60 Getting server logs

The manager can view server logs.

packet type: **journal list**

packet body

| name | type | description | |
|-------------------|--------|---|----------|
| from_datetime | int | closed beginning of the interval from which server logs are retrieved | |
| to_datetime | int | open end of the interval from which server logs are retrieved | |
| parameters_filter | string | string used for filtering parameters (see: j-object <i>journal_info</i>) | optional |

Server responds with a packet:

packet type: **journal list response**

packet body

| name | type | description |
|---------|-----------------------------|------------------|
| journal | j-array <i>journal_info</i> | server logs |
| result | string | operation result |

j-object: *journal_info* :

| name | type | description |
|----------------|--------|--|
| time | int | operation time log in UTC milliseconds |
| client_login | string | client login log |
| client_address | string | client address log |
| client_hash | string | session id log |
| action | string | operation type log |
| parameters | string | operation parameters log |
| result | string | operation result log |

The result of the successful operation is *success*. Any other result indicates an error.



61 Introducing Brokers (IB)

61.1 Getting IB Account list

Getting a list of IB Accounts:

packet type: **ib account list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of IDs as an array of *int*:

packet type: **ib account list response**

packet body

| name | type | description |
|----------|-------------|---------------------|
| accounts | j-array int | list of account IDs |
| result | string | operation result |

61.2 Getting IB Master Account list

Getting a list of IB Master Accounts:

packet type: **ib master account list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of IDs as an array of *int*:

packet type: **ib master account list response**

packet body

| name | type | description |
|----------|-------------|-------------------------------|
| accounts | j-array int | list of IB master account IDs |
| result | string | operation result |

61.3 Getting a list of IB Accounts for a selected IB Master

Getting a list of accounts:

packet type: **ib master filter ib account list**

packet body

| name | type | description |
|------------|------|----------------------|
| id_account | int | IB master account ID |

Server responds with a list of IDs as an array of *int*:



packet type: **ib master filter ib account list response**

packet body

| name | type | description |
|----------|-------------|---------------------|
| accounts | j-array int | list of account IDs |
| result | string | operation result |

61.4 Getting a list of accounts for a selected IB

Getting a list of accounts:

packet type: **ib filter account list**

packet body

| name | type | description |
|------------|------|---------------|
| id_account | int | IB account ID |

Server responds with a list of IDs as an array of *int*:

packet type: **ib filter account list response**

packet body

| name | type | description |
|----------|-------------|---------------------|
| accounts | j-array int | list of account IDs |
| result | string | operation result |

61.5 Getting a list of IB ledger details

Information request:

packet type: **ib ledger details list**

packet body

| name | type | description | |
|-----------------|-------------|--|----------|
| id_account | int | ID of IB account | optional |
| id_account_list | j-array int | list of IB accounts IDs | optional |
| range_from | int | closed beginning of the interval from which the ledger list is to be generated | |
| range_to | int | open end of the interval from which the ledger list is to be generated | |

There are two combinations of fields, which are being operated as a valid request:

1. id_account + range_from + range_to
2. id_account_list + range_from + range_to

Server responds with result of the operation with following packet structure:

packet type: **ib ledger details list response**

packet body

| name | type | description |
|-------------------|---------------------------|-------------------|
| ib_ledger_details | j-array ib_ledger_details | ib ledger details |
| result | string | operation result |

The result of the successful update is success. Any other result indicates an error.



Each entry in the j-array `ib_ledger_details` is a j-object with the following structure:

j-object: ***ib_ledger_details*** :

| name | type | description | |
|-----------------|---------|--------------------|----------|
| account_id | int | account ID | optional |
| surname | string | account surname | optional |
| nominal | decimal | nominal | optional |
| ledger_nominal | decimal | nominal of ledger | optional |
| instrument_name | string | instrument name | optional |
| side | integer | buy or sell | optional |
| open_price | decimal | open price | optional |
| close_price | decimal | close price | optional |
| position_id | int | ID of position | optional |
| comment | string | comment | optional |
| create_time | int | create time | |
| type | string | type of commission | |

61.6 Getting a list of commissions for a selected IB

Getting a list of commissions:

packet type: **ib commission list**

packet body

| name | type | description |
|------------|------|---------------|
| ib_account | int | IB account ID |

Server responds with a list of commissions for a certain IB, as an array of *ib_commission* j-objects.

packet type: **ib commission list response**

packet body

| name | type | description |
|--------------------|------------------------------|--------------------|
| ib_commission_list | j-array <i>ib_commission</i> | list of commission |
| result | string | operation result |

j-object: ***ib_commission*** :

| name | type | description |
|-----------------------|------|-----------------------|
| id_account_root | int | root account |
| id_account_group | int | account group ID |
| id_instrument_class | int | instrument class ID |
| id_commission_pattern | int | commission pattern ID |



61.7 Overwriting the list of commission for a selected IB

Overwriting a list of commissions:

packet type: **set ib commission list**

packet body

| name | type | description |
|----------------|------------------------------|-----------------------|
| ib_commissions | j-array <i>ib_commission</i> | list of IB commission |

j-object: *ib_commission* :

| name | type | description |
|---------------------|------|---------------------|
| name | type | description |
| id_account_root | int | root account |
| id_account_group | int | account group ID |
| id_instrument_class | int | instrument class ID |

Server responds with a packet:

packet type: **set ib commission list response**

packet body

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the correct login process is *success*.

61.8 Updating the list of commission for a selected IB

Overwriting a list of commissions:

packet type: **update ib commission list**

packet body

| name | type | description |
|----------------|------------------------------|-----------------------|
| ib_commissions | j-array <i>ib_commission</i> | list of IB commission |

Server responds with a packet:

packet type: **update ib commission list response**

packet body

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the correct login process is *success*.



61.9 Getting a list of instrument class IDs

Information request:

packet type: **instrument class list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of IDs as an array of *int*:

packet type: **instrument class list response**

packet body

| name | type | description |
|--------------------|-------------|------------------------------|
| instrument_classes | j-array int | list of instrument class IDs |
| result | string | operation result |

61.10 Getting information about instrument class

Information request:

packet type: **instrument class info list**

packet body

| name | type | description |
|--------------------|-------------|------------------------------|
| instrument_classes | j-array int | list of instrument class IDs |

Server responds with a list of information, as an array of *instrument_class_info* j-objects.

packet type: **instrument class info list response**

packet body

| name | type | description |
|--------------------|--------------------------------------|--|
| instrument_classes | j-array <i>instrument_class_info</i> | list of information about instrument class |
| result | string | operation result |

j-object: *instrument_class_info* :

| name | type | description |
|---------------------|--------|--------------------------|
| id_instrument_class | int | instrument class ID |
| name | string | name of instrument class |



61.11 Getting a list of commission patterns for a selected IB

Information request:

packet type: **ib commission pattern list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of commission pattern IDs as an array of *int*:

packet type: **ib commission pattern list response**

packet body

| name | type | description |
|---------------------|-------------|----------------------------|
| commission_patterns | j-array int | commission pattern ID list |
| result | string | operation result |

61.12 Getting a list of commission patterns

Information request:

packet type: **commission pattern list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of commission pattern IDs as an array of *int*:

packet type: **commission pattern list response**

packet body

| name | type | description |
|---------------------|-------------|----------------------------|
| commission_patterns | j-array int | commission pattern ID list |
| result | string | operation result |

61.13 Getting information about commission pattern

Request for information:

packet type: **commission pattern info list**

packet body

| name | type | description |
|---------------------|-------------|--------------------------------|
| commission_patterns | j-array int | list of commission pattern IDs |

Server responds with a list of information, as an array of *commission_pattern_info* j-objects.



packet type: **commission pattern info list response**

packet body

| name | type | description |
|---------------------|--|---|
| commission_patterns | j-array <i>commission_pattern_info</i> | list of information about commissions pattern |
| result | string | operation result |

j-object: *commission_pattern_info* :

| name | type | description | |
|-------------------------|---------|------------------------------------|----------|
| id_commission_pattern | int | commission pattern ID | |
| name | string | name of commission pattern | |
| id_currency | int | currency ID | |
| id_dict_commission | int | commission type from a db glossary | |
| transaction_value | decimal | commission value | optional |
| percent | int | commission value in % | optional |
| id_dict_commission_type | int | commission pattern type | |

61.14 Creating IB commission pattern

Request:

packet type: **create ib commission pattern**

packet body

| name | type | description |
|--------------------|--------------------------------|---|
| commission_pattern | <i>commission_pattern_info</i> | information about commission pattern that will be created |

j-object: *commission_pattern_info* :

| name | type | description | |
|-----------------------|---------|------------------------------------|------------------------------------|
| id_commission_pattern | int | commission pattern ID | note: must be set to 0 (i.e. zero) |
| name | string | name of commission pattern | |
| id_currency | int | currency ID | |
| id_dict_commission | int | commission type from a db glossary | |
| transaction_value | decimal | commission value | optional |
| percent | int | commission value in % | optional |

The server responds with the result of operation with the following packet structure:

packet type: **create ib commission pattern response**

packet body

| name | type | description |
|-----------------------|--------|-------------------------------------|
| result | string | operation result |
| id_commission_pattern | int | newly created commission pattern ID |

The result of the successful operation is success. Any other result indicates an error.

Server generates ID number automatically.



61.15 Getting IB account permissions from db glossary

Request for information:

packet type: **account right list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a full list of permission available to be granted for IB account, as an array of *account_right_info* j-objects.

packet type: **account right list response**

packet body

| name | type | description |
|----------------|-----------------------------------|--|
| account_rights | j-array <i>account_right_info</i> | list of permissions available for IB account |
| result | string | operation result |

j-object: ***account_right_info*** :

| name | type | description |
|------------------|--------|------------------------|
| id_account_right | int | permission ID |
| name | string | permission name |
| description | string | permission description |

61.16 Getting view permissions for IB account

Request:

packet type: **account get account right list**

packet body

| name | type | description |
|---------|------|-------------|
| account | int | account ID |

Server responds with packet.

packet type: **account get account right list response**

packet body

| name | type | description |
|--------------------|-------------|---------------------------------|
| account_right_list | j-array int | list of granted permissions IDs |
| result | string | operation result |



61.17 Adding view permissions for IB account

Request:

packet type: **account create account right list**

packet body

| name | type | description |
|----------------------------|--------------------------------------|-----------------------------------|
| account_account_right_list | j-array <i>account_account_right</i> | list of permissions to be created |

j-object: ***account_account_right*** :

| name | type | description |
|------------------|------|---------------|
| id_account | int | account ID |
| id_account_right | int | permission ID |

Server responds with a packet:

packet type: **account create account right list response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is success. Any other result indicates an error.

61.18 Deleting view permissions for IB account

Request:

packet type: **account delete account right list**

packet body

| name | type | description |
|----------------------------|--------------------------------------|-----------------------------------|
| account_account_right_list | j-array <i>account_account_right</i> | list of permissions to be deleted |

j-object: ***account_account_right*** :

| name | type | description |
|------------------|------|---------------|
| id_account | int | account ID |
| id_account_right | int | permission ID |

Server responds with a packet:

packet type: **account delete account right list response**

packet body:

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

The result of the successful operation is success. Any other result indicates an error.



62 Getting a list of available currencies

Request:

packet type: **currency list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a full list of available currencies, as an array of *account_right_info* j-objects.

packet type: **currency list response**

packet body

| name | type | description |
|------------|-------------------------|------------------------------|
| currencies | j-array <i>currency</i> | list of available currencies |
| result | string | operation result |

j-object: **currency** :

| name | type | description |
|------------------|--------|---------------|
| id_dict_currency | int | currency ID |
| name | string | currency name |

63 Multi Account Manager (MAM)

63.1 Getting the list of available MAM commission calculation list

packet type: **mam calculation list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of available commission calculation frequencies.

packet type: **mam calculation list response**

packet body

| name | type | description |
|-----------------|--------------------------------|--|
| mam calculation | j-array <i>mam_calculation</i> | the list of available commission calculation frequencies |
| result | string | operation result |

j-object: **mam_calculation** :

| name | type | description |
|------|--------|---------------------------------------|
| id | int | commission calculation frequency id |
| name | string | commission calculation frequency name |



63.2 Getting the list of commission types for MAM success fees

packet type: **mam success fee type list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of commission types for MAM success fees.

packet type: **mam success fee type list response**

packet body

| name | type | description |
|----------------------|-------------------------------------|---|
| mam success fee type | j-array <i>mam_success_fee_type</i> | the list of commission types for MAM success fees |
| result | string | operation result |

j-object: *mam_success_fee_type* :

| name | type | description |
|------|--------|---------------------------|
| id | int | fee calculation type id |
| name | string | fee calculation type name |

63.3 Getting the list of MAM management fee types

packet type: **mam management fee type list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of MAM management fee types.

packet type: **mam management fee type list response**

packet body

| name | type | description |
|----------------------|--|----------------------------------|
| mam success fee type | j-array <i>mam_management_fee_type</i> | the list of management fee types |
| result | string | operation result |

j-object: *mam_management_fee_type* :

| name | type | description |
|------|--------|---------------|
| id | int | fee type id |
| name | string | fee type name |



63.4 Getting the list of MAM accounts

packet type: **mam account list**

packet body

| name | type | description |
|---------|------|-------------|
| <empty> | | |

Server responds with a list of IDs as an int array.

packet type: **mam account list response**

packet body

| name | type | description |
|----------|-------------|---------------------|
| accounts | j-array int | list of account IDs |
| result | string | operation result |

63.5 Getting MAM commission

packet type: **mam commission**

packet body

| name | type | description |
|---------|------|----------------|
| account | int | MAM account id |

Server responds with a list (0 or 1) of commissions as *mam_commission* array.

packet type: **mam commission response**

packet body

| name | type | description |
|----------|-------------------------------|----------------------------------|
| accounts | j-array <i>mam_commission</i> | list of MAM accounts commissions |
| result | string | operation result |

j-object: ***mam_commission*** :

| name | type | description | |
|-------------------------------|---------|---|----------|
| id | int | commission ID | |
| account_id | string | account ID with assigned commission | |
| success_fee_percent | decimal | success fee value in % | optional |
| success_fee_calculation_id | int | "mam calculation list response" packet ID | optional |
| success_fee_type_id | int | "mam success fee type list response" packet ID | optional |
| management_fee_percent | decimal | management fee value in % | optional |
| management_fee_calculation_id | int | "mam calculation list response" packet id | optional |
| management_fee_type_id | int | "mam management fee type list response" packet id | optional |

"Success" or "management" fee should fill in the 3 related fields.



63.6 Getting the list of accounts assigned to the MAM account

packet type: **mam filter account list**

packet body

| name | type | description |
|---------|------|----------------|
| account | int | MAM account ID |

Server responds with a list of IDs as an int array.

packet type: **mam filter account list response**

packet body

| name | type | description |
|----------|-------------|---------------------|
| accounts | j-array int | list of account IDs |
| result | string | operation result |

63.7 Updating MAM commission

packet type: **update mam commission**

packet body

| name | type | description | |
|-------------------------------|---------|---|----------|
| id | int | commission ID | |
| account_id | string | account ID with assigned commission | |
| success_fee_percent | decimal | success fee value in % | optional |
| success_fee_calculation_id | int | "mam calculation list response" packet ID | optional |
| success_fee_type_id | int | "mam success fee type list response" packet ID | optional |
| management_fee_percent | decimal | management fee in % | optional |
| management_fee_calculation_id | int | "mam calculation list response" packet ID | optional |
| management_fee_type_id | int | "mam management fee type list response" packet ID | optional |

"Success" or "management" fee should fill in the 3 related fields.

Server responds with a packet:

packet type: **update mam commission response**

packet body

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

63.8 Getting allocation method for MAM

packet type: **mam account allocation**

packet body

| name | type | description |
|---------|------|----------------|
| account | int | MAM account ID |

Server responds with MAM account allocation method ID:

1. Balance



2. Free margin

Server responds with a packet:

packet type: **mam account allocation response**

packet body

| name | type | description |
|------------------------|--------|----------------------------------|
| accounts allocation id | int | MAM account allocation method ID |
| result | string | operation result |

63.9 Updating MAM account allocation method

packet type: **update mam account allocation**

packet body

| name | type | description |
|-------------------|------|----------------------------------|
| account | int | MAM account ID |
| mam_allocation_id | int | MAM account allocation method ID |

MAM account allocation method ID:

1. Balance
2. Free margin

Server responds with a packet:

packet type: **update mam account allocation response**

packet body

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |

63.10 Manual calculation of success fees

packet type: **calculate success fee for master**

packet body

| name | type | description |
|---------|------|----------------|
| account | int | MAM account ID |

Server responds with a packet:

packet type: **calculate success fee for master response**

packet body

| name | type | description |
|--------|--------|------------------|
| result | string | operation result |