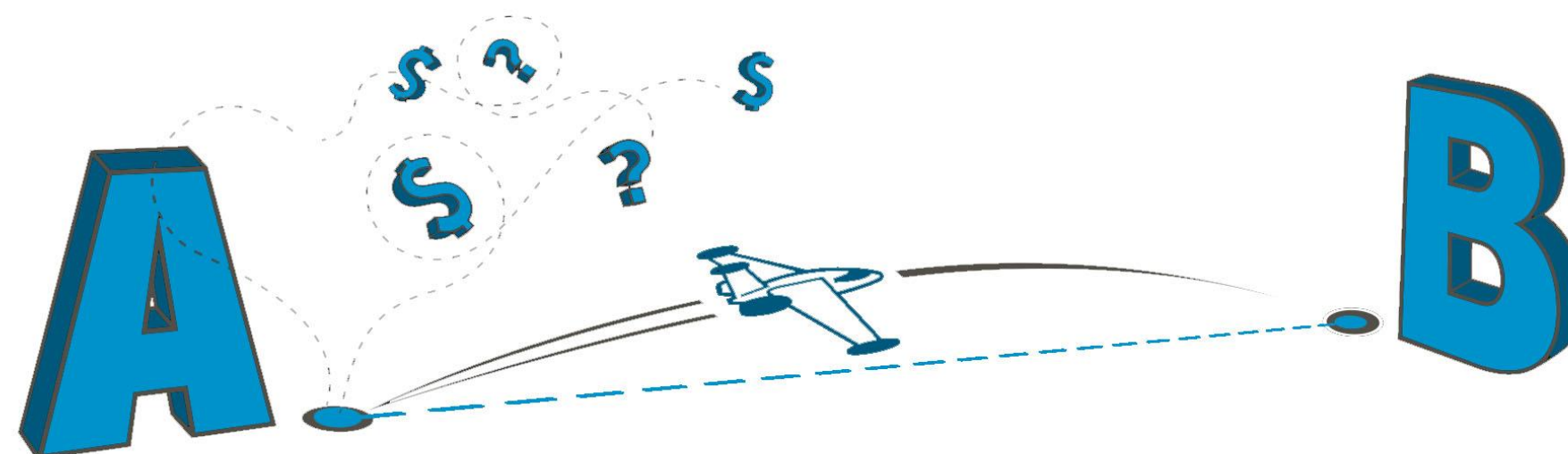# Installation Guide
Website Integration Kit v.075

TOOLS FOR BROKERS
enjoy technical perfection

Contents

Caution

1. For your server safety, we **strongly** recommend **testing** all technical solutions in a **demo** environment **first**.

   While testing, make sure that the demo environment you use is maximum similar to your live server conditions.

2. We highly appreciate your **convenience**.

   Our first goal is to make you pleased with the product and the **result** it brings.

   We are always happy to know that you enjoy using our product and can achieve your **exact tasks** by using it.

   That is why we highly recommend to **contact us**, so that our team specialists would easily **install** the product for you

   and **demonstrate** the product's operation and usage on your demo servers.

3. **Warning: Do not operate** the product until you have **read** and thoroughly **understood** this manual.

1. Introduction

1.1. Description

Website Integration Kit is an application that **connects** your website and MetaTrader4 and **provides** you with an **access** to MT4 wherever you are.

The solution includes both *Trader* and *Admin Cabinets* that enable to **request all** necessary data from MT4 server and to **run** different **actions** on it.

1.2. Product Advantages

1.2.1. Trader Cabinet

The *Trader Cabinet* is basically a **web-interface** that gives an opportunity to trader to:

- ➤ Open and manage trading **accounts**;
- ➤ Perform financial **transactions**;
- ➤ Get attached to **money managers**;
- ➤ Establish **referral links** for IB-structure.

1.2.2. Admin Cabinet

The *Admin Cabinet* is a tool that allows to perform **manager functions** you need with any internet-enabled device.

The *Admin Cabinet* provides you with:

- ➤ an ability to always **stay online** in your MT4;
- ➤ an **admin access** to your traders' accounts;
- ➤ support by **MT4 Manager**;
- ➤ an ability to **combine** it with *Ultimate Multi Account Management* (*UMAM*) and *Multilevel IB and Agent Commission* (*MLAC*) plugins.

1.2.3. WebAPI is vital for your business if

- ➤ You need a **reliable access** to your web-server and **any** of your data at **any** time;
- ➤ You want to **view** and **edit** information **as you go**;
- ➤ You need the **commands** of your web-interface to be **simple**;
- ➤ This is important for you that such a tool would be **compatible with** any web-programming language;
- ➤ The flexibility is a key aspect for your software as the flexibility makes it possible to add **any particular function** you need.

1.3. How does it work?

Website Integration Kit operates as a tunnel between your website and MetaTrader4 that connects them for **convenience** and universal **availability** of your trading system.

It requires MT4 Manager access to establish connection, and uses *TCP/IP* protocol for incoming connections.

Figure 1. WebAPI operating principle of
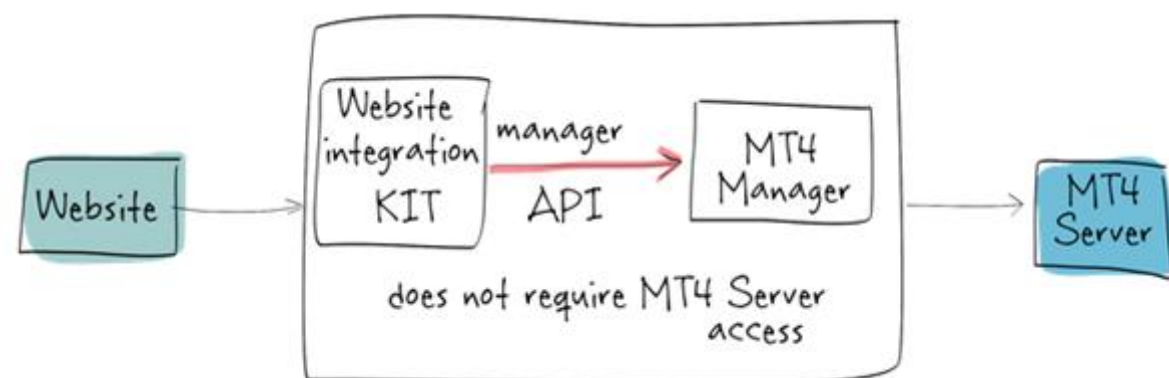
We provide a tool with a specific set of functions that can be **integrated by the broker**. It is really easy to do for an IT department or a single web developer.

The solution performs functions available in MT4 Manager. E.g., an Investor can view their trading history/balance/equity **from the website** in their *Trader Cabinet*.

*Note 1:

When testing, please, **use** *Python*, *PHP*, or any other web-programming language to check if the requests work **correctly**.

Example:

Let us assume that you currently have ***no access*** *to MT4 Manager/MT4 Administrator. However, you* ***need*** *to* ***change*** *a trader's* ***credit*** *value. Just log into your website, and* ***send*** *the* ***changecredit*** *request by* ***a single click***.

2. Installation and configuration

2.1. Installation

1) Edit *config.cfg*;
2) Edit *ipfilter.cfg*;
3) IP-list (line by line) with the connection data for **WebAPI**;
4) *Install* and *Start* service, or *Run as Application* (*.bat* file).

2.2. Configuration

The **main** configuration file for you to edit is *config.cfg.*

Please see the configuration example in **Figure 2**.

Figure 2. *config.cfg* **file example**

```
1   #server ip and port for incoming connections.
2   #(external IP of current network device + free port)
3   externalIP=127.0.0.1
4   externalPort=4444
5
6   #mt4 manager's access, nust have full rights
7   mt4server=127.0.0.1
8   mt4managerLogin=1
9   mt4managerPass=manager
10
11  #number of max. realtime connections to mt4 server
12  mt4connectionsCount=2
13
14  #security data
15  useEncryption=0
16  useHash=0
17  encryptionKey=asfas1
18  hashKey=fsdvgfygfsddsag
19
20  socketTimeout=20
```

Table 1. Parameters

externalIP	127.0.0.1 or 0.0.0.0 by default.
externalPort	free port that should be used by the application.
mt4server	IP-address of MT4 server to be used. Please do not use 443 port if your MT4 Server and WebAPI are on the same PC.
mt4managerLogin	login of a manager account on MT4 server (the manager should have certain MT4 rights).
mt4managerPass	password for MT4 manager account.

mt4connectionsCount	number of connections to MT4 server. It allows multiple incoming requests handling.
useEncryption	0 or 1: to not use (0) or use (1) encryption.
useHash	0 or 1: to not use (0) or use (1) md5 hash for requests.
encryptionKey	the key to encrypt a request.
hashKey	the key used as 'salt' (random data) in md5 hash generation.
socketTimeout	inactivity period in seconds, after which the client incoming connection will be closed.
URLEncode	0 or 1: to not use (0) or use (1) url-encoding.

2.3.   IP-filter editing

The file *ipfilter.cfg* allows incoming connections from listed IP-addresses. To allow **all** IP addresses, input '**'* as shown in **Figure 3**.

Figure 3. *ipfilter.cfg* **file example**

```
1   127.0.0.1
2   *
```

To allow **different** IP addresses, input IPs into the file **line by line**.

3. Web API requests

3.1. Requests list

This is how the complete list of requests looks like:

1. Account have trades
2. Add order
3. Attach
4. Balance fix
5. Bulk change balance
6. Bulk change credit
7. Bulk modify account
8. Bulk modify order
9. Change agent commission
10. Change balance
11. Change credit
12. Change password
13. Check password
14. Close order
15. Close order by
16. Create account
17. Create account in range
18. Create group
19. Delete account
20. Delete group swaps
21. Delete pending order
22. Delete plugin value
23. Detach
24. Get account balance
25. Get account info
26. Get account info expanded
27. Get accounts
28. Get accounts with p/l
29. Get all balances
30. Get balances operations
31. Get chart history
32. Get group
33. Get group allowed symbols
34. Get groups
35. Get history
36. Get journal
37. Get leverage and group
38. Get logins by group
39. Get margin info
40. Get margin info expanded
41. Get masters
42. Get open trades
43. Get opened trades for login
44. Get open orders
45. Get order

46. Get pending orders

47. Get p/l

48. Get plugin

49. Get slaves

50. Get symbols

51. Get trading volume

52. Get users online

53. Get version

54. Modify account

55. Modify group

56. Modify group swaps

57. Modify order

58. Modify pending order

59. Open order

60. Open pending order

61. Pumping (Subscription for news, equity level etc.)

62. Send mail

63. Server time

64. Transfer balance

65. Update plugin value

3.2. Sending requests. Example

To send a **request**:

- ✓ **connect** to the application using *externalIP* and *externalPort*,

- ✓ **write** the request to the open TCP/IP socket, and then

- ✓ **read** the **answer** from the socket.

Note 2:

no reconnection needed to send multiple requests; however, **before** sending a **new** request, you need to read the answer for your **current** one.

Request format with no encryption (**example**):

'action=actionName&param1=paramValue1&...&paramN=paramValueN& request_id=randomValue&hash=hashValue\0'

HashValue generation: *md5 (request_id + actionName + hashKey)*

Encryption: simple *XOR* encryption with looping usage of *encryptionKey*.

If you need to use '&' in your values, apply *HTMLURL* encoding to those values.

Answer format:

Examples:

1) Answer **with no** *SCSV (Semi-Colon Separasted Values)* data:
   'result=resultValue&param1=paramValue1&..&paramN=paramValueN\ n'

2) Answer **with** *SCSV* data:
   'result=resultValue&param1=paramValue1&...&paramN=paramValueN &size=sizeOfSCSVData\n scsvDataLine1\n ...

scsvDataLineN\n'

Each value on the line is **separated** with '**;**'

Table 2. Result types

1	success
0	invalid action type
-1	mt4 connection issue
-2	invalid hash (if hash used)
-3	invalid incoming parameters
-4	no valid data on mt4 server

3.3.   Requests specifics

Request parameters are those you need to send in your requests.

These include:

- ✓ **required** parameters;

- ✓ **optional** parameters.

You need to specify all **required** parameters, otherwise the request fails.

Answer Parameters is what you see as a result of sending a request.

The **SCSVLineFormat** line shows the **format** of returned data if the parameter values are separated with semicolon.

cmd (=*command*) – order operation type

Table 3. Command values

0	OP_BUY
1	OP_SELL
2	OP_BUY_LIMIT
3	OP_SELL_LIMIT

4	OP_BUY_STOP
5	OP_SELL_STOP
6	OP_BALANCE
7	OP_CREDIT

Table 4 contains numeriacal values for MT4 **Journal** modes. This might be useful for *getjournal* parameter configuration. Use the table to specify *mode* parameter value – to get certain **record types** of the Journal.

Table 4. Journal mode values

0	**Standard**: all record types except for the users connection data
1	**Logins**: records on connection to the server
2	**Trades**: trade operations messages
3	**Errors**: records on error messages
4	**Full**: all record types including the users connection data

Now, let us take a **closer look** at **each** request.

3.4.   Requests descriptions

1. **'accounthavetrades'** allows to request information on any trades on the account.

   There are **2 answer options**:

   1 – the account has some trades;

   0 – no trades.

   - ✓ **Action**: *accounthavetrades*

   - ✓ **Required** request parameters: *login*

   - ✓ **Optional** request parameters: *request_id*

- ✓ **Answer** parameters: *result, login, trades, request_id*
- ✓ **SCSVLineFormat**: not used

2. **'addorder'** adds an order to the trading history. If you set *open_time* parameter only, the order is indicated as an open one in Manager MT4. However, it has no effect on the actual balance.
   - ✓ **Action**: *addorder*
   - ✓ **Required** request parameters: *login, symbol, volume*
   - ✓ **Optional** request parameters: *cmd* (opens **buy** orders with no *cmd* specified), *sl, tp, open_price, close_price, request_id, open_time, close_time*
   - ✓ **Answer** parameters: *result, order, request_id*
   - ✓ **SCSVLineFormat**: not used

   Note that *1* set as *volume* means **1 lot**.

3. **'attach'** gets an Investor (Slave) account attached to a Money Manager (Master).
   - ✓ **Action**: *attach*
   - ✓ **Required** request parameters: *master, slave, plugin_name*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result, request_id*
   - ✓ **SCSVLineFormat**: not used

4. **'balancefix'** gets the accounts balances fixed for the specified logins.
   - ✓ **Action**: *balancefix*

- ✓ **Required** request parameters: *logins*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, request_id*
- ✓ **SCSVLineFormat**: not used

5. **'bulkchangebalance'** enables to change balance of specific accounts.
   - ✓ **Action**: *bulkchangebalance*
   - ✓ **Required** request parameters: *login, value, comment*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result, size, request_id*
   - ✓ **SCSVLineFormat**: *login; newbalance; ticket; success_flag*

6. **'bulkchangecredit'** action changes the current credit value for multiple accounts and adds a comment to each account's trading history.
   - ✓ **Action**: *bulkchangecredit*
   - ✓ **Required** request parameters: *logins, value, comment*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result, size, request_id*
   - ✓ **SCSVLineFormat**: *login; newcredit; ticket; success_flag*

7. **'bulkmodifyaccount'** allows changing general information of specified accounts. The number of accounts is not limited.
   - ✓ **Action**: *bulkmodifyaccount*
   - ✓ **Required** request parameters: *login*

- ✓ **Optional** request parameters: *name, group, agent, leverage, email, id, address, city, state, zipcode, country, phone, password_phone, comment, enable, enable_change_password, enable_read_only, send_reports, status, request_id*
- ✓ **Answer** parameters: *result, request_id*
- ✓ **SCSVLineFormat**: logins failed to get modified

8. **'bulkmodifyorder'** allows changing parameters of specified orders.
- ✓ **Action**: *bulkmodifyorder*
- ✓ **Required** request parameters: *order*
- ✓ **Optional** request parameters: *symbol, cmd, volume, open_time, open_price, closetime, close_price, sl, tp, expiration, reason, 1_conv_rate, 2_conv_rate, commission, commission_agent, storage, profit, taxes, margin_rate, comment, gw_order, gw_volume, gw_open_price_dt, gw_close_price_dt, request_id*
- ✓ **Answer** parameters: *result, request_id*
- ✓ **SCSVLineFormat**: not used

Note 1: unlike *modifyorder*, *bulkmodifyorder* is able to process **unlimited** amount of orders specified in **one** request.

Note 2: *1* set as *volume* would mean **1 lot**.

9. **'changeagentcommission'** action allows order commission size changing.
- ✓ **Action**: *changeagentcommission*

- ✓ **Required** request parameters: *order, value*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, order, request_id*
- ✓ **SCSVLineFormat**: not used

10. **'changebalance'** request enables to make a deposit/withdrawal.
- ✓ **Action**: *changebalance*
- ✓ **Required** request parameters: *login, value, comment*
- ✓ **Optional** request parameters: *freemarginlimit, request_id*
- ✓ **Answer** parameters: *result, login, newbalance, order, request_id*
- ✓ **SCSVLineFormat**: not used

Note: *order* parameter returns the **transaction ticket** indicated in the account trading history.

11. **'changecredit'** action allows changing credit value for an account with a comment added to the trading history.
- ✓ **Action**: *changecredit*
- ✓ **Required** request parameters: *login, value, comment*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, login, newcredit, order, request_id*
- ✓ **SCSVLineFormat**: not used

12. **'changepassword'** request allows changing (either trading or investor one).
- ✓ **Action**: *changepassword*
- ✓ **Required** request parameters: *login, pass, investor*

- ✓ **Optional** request parameters: none
- ✓ **Answer** parameters: *result*, *login*, *request_id*
- ✓ **SCSVLineFormat**: not used

Note: there are **two options** for *investor* parameter:

- ➢ If *0* is set, the **trading** password will be changed,
- ➢ If *1* is set, the **investor** password will be changed.

13. **'checkpassword'** action allows trading password checking.
- ✓ **Action**: *checkpassword*
- ✓ **Required** request parameters: *login*, *pass*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result*, *login*, *request_id*
- ✓ **SCSVLineFormat**: not used

14. **'closeorder'** action closes a specific order.
- ✓ **Action**: *closeorder*
- ✓ **Required** request parameters: *order*
- ✓ **Optional** request parameters: *volume*, *request_id*
- ✓ **Answer** parameters: *result*, *request_id*
- ✓ **SCSVLineFormat**: not used

Note 1: You may set a price for closing of each specified order.

Format: *order=order_id1,price1;order_id2;order_id3,price3*

Example: order=215773,3.99000;215778;215779,3.99900\0')

Note 2: *1* set as *volume* means **1 lot**.

15. **'closeorderby'** action allows closing a specific order.
- ✓ **Action**: *closeorderby*
- ✓ **Required** request parameters: *order*, *orderby*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result*, *request_id*
- ✓ **SCSVLineFormat**: not used

Note: you may specify price for any order closing in format: *order=order_id1,price1;order_id2;order_id3,price3*

Example: *order=1816683,1.99999;1816684,1.99999*

16. **'createaccount'** action is intended to create an account.
- ✓ **Action**: *createaccount*
- ✓ **Required** request parameters: *login*, *name*, *group*, *leverage*
- ✓ **Optional** request parameters: *agent*, *email*, *id*, *address*, *city*, *state*, *zipcode*, *country*, *phone*, *password*, *password_phone*, *password_investor*, *comment*, *enable*, *enable_change_password*, *enable_read_only*, *send_reports*, *status*, *request_id*
- ✓ **Answer** parameters: *result*, *login*, *request_id*
- ✓ **SCSVLineFormat**: not used

17. **'createaccountinrange'** action allows trader creating an account with a specific login. For example, if *begin=10, end=20*, the account will be created with a login within that range.

   ✓ **Action**: *createaccountinrange*

   ✓ **Required** request parameters: *name, group, begin, end*

   Optional request parameters: *login, agent, leverage, email, id, address, city, state, zipcode, country, phone, password, password_phone, password_investor, comment, enable, enable_change_password, enable_read_only, send_reports, status, request_id*

   ✓ **Answer** parameters: *result, login, request_id*

   ✓ **SCSVLineFormat**: not used

18. '**creategroup'** action creates a group with a specified name.

   ✓ **Action**: *creategroup*

   ✓ **Required** request parameters: *name*

   ✓ **Optional** request parameters: *base_group, currency, signature, company, smtp_server, smtp_login, smtp_password, support_email, templates, enable, default_leverage, default_deposit, timeout, adv_security, copies, reports, margin_call, margin_mode, margin_stopout, news, check_ie_prices, maxpositions, interestrate, use_swap,* close_reopen, *hedge_prohibited, close_fifo, margin_type, archive_period, archive_max_balance, archive_pending_period, stopout_skip_hedged, request_id*

   ✓ **Answer** parameters: *result, name, request_id*

   ✓ **SCSVLineFormat**: not used

19. **'deleteaccount'** action enables to delete an account via the website.

   ✓ **Action**: *deleteaccount*

   ✓ **Required** request parameters: *login*

   ✓ **Optional** request parameters: *request_id*

   ✓ **Answer** parameters: *result, login, request_id*

   ✓ **SCSVLineFormat**: not used

20. **'deletegroupswaps'** enables broker to delete swap size previously set for a group.

   ✓ **Action**: *deletegroupswaps*

   ✓ **Required** request parameters: *name, symbol*

   ✓ **Optional** request parameters: *request_id*

   ✓ **Answer** parameters: *result, request_id*

   ✓ **SCSVLineFormat**: not used

21. **'deletependingorder'** action removes previously placed pending order(s). Multiple orders should be separated with '**;**'.

   ✓ **Action**: *deletependingorder*

   ✓ **Required** request parameters: *order*

   ✓ **Optional** request parameters: *request_id*

   ✓ **Answer** parameters: *result, request_id*

   ✓ **SCSVLineFormat**: not used

22. **'deletepluginvalue'** enables to delete the plugin parameters.
   - ✓ **Action**: *deletepluginvalue*
   - ✓ **Required** request parameters: *name*, *param*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result*, *request_id*
   - ✓ **SCSVLineFormat**: not used

23. **'detach'** gets an Investor (Slave) detached from a Money Manager (Master) account.
   - ✓ **Action**: *detach*
   - ✓ **Required** request parameters: *slave*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result*, *request_id*
   - ✓ **SCSVLineFormat**: not used

24. **'getaccountbalance'** returns the specific account's balance.
   - ✓ **Action**: *getaccountbalance*
   - ✓ **Required** request parameters: *login*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result*, *login*, *balance*, *request_id*
   - ✓ **SCSVLineFormat**: not used

25. **'getaccountinfo'** allows requesting the account's detailed data.
   - ✓ **Action**: *getaccountinfo*
   - ✓ **Required** request parameters: *login*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result, login, name, email, group, leverage, regdate, country, state, address, phone, city, zip, enable, tradingblocked, balance, comment, enableChangePassword, free_margin, opened_orders, agent, lastdate, request_id*
   - ✓ **SCSVLineFormat**: not used

26. **'getaccountinfoex'** request enables trader to request full account information. Additional parameters are: *id*, *status*.
   - ✓ **Action**: *getaccountinfoex*
   - ✓ **Required** request parameters: *login*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result, login, name, email, group, leverage, regdate, country, state, address, phone, city, zip, enable, tradingblocked, balance, comment, enable_change_password, free_margin, opened_orders, agent, id, status, interestrate, send_reports, prevmonthbalance, prevbalance, credit, equity, margin, margin_level, taxes, timestamp, lastdate, request_id*
   - ✓ **SCSVLineFormat**: not used

27. **'getaccounts'** returns information of a specific agent and the corresponding group, with all accounts belonging to it.
   - ✓ **Action**: *getaccounts*
   - ✓ **Required** request parameters: *agent*, *group*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result*, *size*

- ✓ **SCSVLineFormat**: *login*; *name*; *email*; *group*; *leverage*; *regdate*; *country*; *state*; *adress*; *city*; *zip*; *enable*; *tradingblocked*; *balance*; *comment*; *enableChangePassword,*; *agent*; *last_access_date*; *request_id*

Note: two options for the parameter *agent* are **available**:

- ➢ If it is specified, and the *group* parameter is not, the agent groups data are returned.
- ➢ If it is not specified or *= 0*, all accounts of the specified *agent* are returned.

28. **'getaccountswithpl'** returns data on all accounts with profit/loss.
- ✓ **Action**: *getaccountswithpl*
- ✓ **Required** request parameters: *value, typeofcomparison*

  typeofcomparison parameter has the following **values**:
  - ➢ *1* – greater_than,
  - ➢ *2* – less_than,
  - ➢ *3* – equal
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, size, request_id*
- ✓ **SCSVLineFormat**: *logins*

29. **'getallbalances'** action returns all balances from your server.
- ✓ **Action**: *getallbalances*
- ✓ **Required** request parameters: none
- ✓ **Optional** request parameters: *request_id*

- ✓ **Answer** parameters: *result, size, request_id*
- ✓ **SCSVLineFormat**: *login*; *balance*

30. **'getbalancesoperations'** allows receiving information about deposits and withdrawals of account for a specific period of time.
- ✓ **Action**: *getbalancesoperations*
- ✓ **Required** request parameters: *login, from, to*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, login, size, request_id*
- ✓ **SCSVLineFormat**: *ticket*; *profit*; *open_time*; *comment*

31. **'getcharthistory'** action returns information on charts for a requested symbol and period.
- ✓ **Action**: *getcharthistory*
- ✓ **Required** request parameters: *symbol, from, to, size*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, count, request_id*
- ✓ **SCSVLineFormat**: *chart_ticks*; *time*; *ticks_amount*

32. **'getgroup'** returns the group full data.
- ✓ **Action**: *getgroup*
- ✓ **Required** request parameters: *name*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, name, currency, signature, company, smtp_server, smtp_login, smtp_password, support_email, templates, ebable, default_leverage, default_deposit, timeout, copies, report, margin_call,*

margin_mode,   margin_stopout,   news,   check_ie_prices, maxpositions,   interestrate,   use_swap,   close_reopen, hedge_prohibited,   close_fifo,   margin_type,   archive_period, archive_max_balance,   archive_pending_period, stopout_skip_hedged, request_id

- ✓ **SCSVLineFormat**: not used

33.   **'getgroupallowedsymbols'** returns the list of all symbols allowed for the requested group.

- ✓ **Action**: *getgroupallowedsymbols*
- ✓ **Required** request parameters: *group*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, count, symbols, request_id*
- ✓ **SCSVLineFormat**: not used

34.   **'getgroups'** allows requesting information on all groups on your server, such as groups amount, their names and currencies.

- ✓ **Action**: *getgroups*
- ✓ **Required** request parameters: none
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, groupsCount, size, request_id*
- ✓ **SCSVLineFormat**: *group_name; enable; currency*

35.   **'gethistory'** allows requesting all trading history within the specified time period (*Unix time* format).

- ✓ **Action**: *gethistory*
- ✓ **Required** request parameters: *login, from, to*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, users, size, request_id*
- ✓ **SCSVLineFormat**: *login; order; symbol; open_price; close_price; profit; volume; open_time; close_time; comment; commission; agent_commission; cmd; sl; tp; swap*

Note 1: in parameter *login*, accounts divided by ';'

Note 2: in parameter *login*, you may use mask '*' to get history of all accounts.

Note 3: in parameter *cmd* – order operation type (for the *cmd* parameter values, please see in **Table 3** on **p.8**).

36.   **'getjournal'** returns data from in MT4 Manager/Admin Journal.

- ✓ **Action**: *getjournal*
- ✓ **Required** request parameters: *mode, from, to*
- ✓ **Optional** request parameters: *filter, request_id*
- ✓ **Answer** parameters: *result, size, request_id*
- ✓ **SCSVLineFormat**: *time; ip; message*

Note: parameter *mode* allows getting different kinds of data recorded in the Journal, exactly like in MT4. For the parameter values, please see **Table 4** on p.**8**.

37.   **'getleverageandgroup'** returns the current leverage values of all groups on the server.

- ✓ **Action**: *getleverageandgroup*

- ✓ **Required** request parameters: none
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, size, request_id*
- ✓ **SCSVLineFormat**: *'group;name;leverage'*

38. **'getloginsbygroup'** returns *SCSV*-formatted logins list contained in the specified group.
- ✓ **Action**: *getloginsbygroup*
- ✓ **Required** request parameters: *group*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer**: *result, logins, request_id*
- ✓ **SCSVLineFormat**: not used

39. **'getmargininfo'** action requests information on margin level, balance, and equity user by user.
- ✓ **Action**: *getmargininfo*
- ✓ **Required** request parameters: *login*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, users, balance, equity, margin, freeMargin, size, request_id*
- ✓ **SCSVLineFormat**: *users; balance; equity; margin; free_margin*

Note 1: in parameter *login*, accounts divided by '*;*'

Note 2: *balance* returned by the action is equal to balance + credit (to get balance and credit **separately**, use *getmargininfoex*)

40. **'getmargininfoex'** allows getting margin level, balance, equity data. It is similar to *getmargininfo* action, however, *getmargininfoex* returns credit data separately.
- ✓ **Action**: *getmargininfoex*
- ✓ **Required** request parameters: *login*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, users, balance, credit, equity, margin, freeMargin, size, request_id*
- ✓ **SCSVLineFormat**: *users; balance; credit; equity; margin; free_margin*

Note: in parameter *login*, accounts are separated by '*;*'

41. **'getmasters'** enables to request the list of all Master accounts.
- ✓ **Action**: *getmasters*
- ✓ **Required** request parameters: none
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, mastersCount, size, request_id*
- ✓ **SCSVLineFormat**: *master_login; slaves_amount*

42. **'getopenedtrades'** returns the list of all open trades on your server.
- ✓ **Action**: *getopenedtrades*
- ✓ **Required** request parameters: no
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, trades, size, request_id*

✓ **SCSVLineFormat**: *order*; *login*; *symbol*; *cmd*; *volume*; *open_time*; *open_price*; *sl*; *tp*; *profit*; *commission*; *comment*

43. **'getopenedtradesforlogin'** request allows trader to get the list of their currently opened trades.

    ✓ **Action**: *getopenedtradesforlogin*

    ✓ **Required** request parameters: *login*

    ✓ **Optional** request parameters: *request_id*

    ✓ **Answer** parameters: *result, trades, size, request_id*

    ✓ **SCSVLineFormat**: *order*; *login*; *symbol*; *cmd*; *volume*; *open_time*; *open_price*; *sl*; *tp*; *profit*; *commission*; *comment*

44. **'getopenorders'** allows to get the list of open orders for the requested login.

    ✓ **Action**: *getopenorders*

    ✓ **Required** request parameters: *login*

    ✓ **Optional** request parameters: *request_id*

    ✓ **Answer** parameters: *result, trades, size, request_id*

    ✓ **SCSVLineFormat**: *login*; *order*; *symbol*; *open_price*; *current_price*; *profit*; *volume*; *open_time*; *close_time*; *comment*; *commission*; *agent_commission*; *cmd*; *sl*; *tp*; *swaps*

Note: in parameter *cmd*, specify the order operation type (for the *cmd* parameter values, please see **Table 3** on p.**8**).

45. **'getorder'** returns information on a specific order.

    ✓ **Action**: *getorder*

    ✓ **Required** request parameters: *order*

    ✓ **Optional** request parameters: *request_id*

    ✓ **Answer** parameters: *result, size, request_id*

    ✓ **SCSVLineFormat**: *login*; *order*; *symbol*; *open_price*; *close_price*; *profit*; *volume*; *open_time*; *close_time*; *comment*; *commission*; *agent*; *cmd*; *sl*; *tp*; *swap*

Note 1: if you request data on an **opened** order, the **current** price (the one at the moment of requesting) will be returned as *close_price*.

Note 2: in parameter *cmd*, specify the order operation type (for the *cmd* parameter values, please see **Table 3** on p.**8**).

46. **'getpendingorders'** returns data on pending orders for a specific account.

    ✓ **Action**: *getpendingorders*

    ✓ **Required** request parameters: *login*

    ✓ **Optional** request parameters: *request_id*

    ✓ **Answer** parameters: *result, trades, size, request_id*

    ✓ **SCSVLineFormat**: *order*; *login*; *symbol*; *order_type*; *open_time*; *open_price*; *sl*; *tp*; *comment*

Note: parameter *open_time* indicates the time when the order was **placed**.

47.   **'getpl'** allows requesting data on account's profit/loss.

- ✓ **Action**: *getpl*
- ✓ **Required** request parameters: *login*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, size, request_id*
- ✓ **SCSVLineFormat**: *pl_value*

48.   **'getplugin'** request returns name of the specified plugin installed on your MT4 server, and its parameters with the configured values.

- ✓ **Action**: *getplugin*
- ✓ **Required** request parameters: *name*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, name, size, request_id*
- ✓ **SCSVLineFormat**: *parameter_name; value*

49.   **'getslaves'** action returns all Investor (Slave) accounts of a requested Money Manager (Master).

- ✓ **Action**: *getslaves*
- ✓ **Required** request parameters: *master*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, slavesCount, size, logins, request_id*
- ✓ **SCSVLineFormat**: not used

50.   **'getsymbols'** request allows trader to request information on a certain symbol.

- ✓ **Action**: *getsymbols*

- ✓ **Required** request parameters: none
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, symbolsCount, size, request_id*
- ✓ **SCSVLineFormat**: *symbol_name*; *digits*; *profit_mode*; *security*; *security_type*; *swapLong*; *swapShort*; *spread*; *bid*; *ask*; *margin_mode*; *tickSize*; *tickPrice*; *initial_margin*; *contract_size*; *point*

Note 1: parameter *profit_mode* is equal to *Profit calculation* in MetaTrader (*Symbol settings –> Calculation –> Profit Calculation*). The values that return in the response, will mean: *0* = Forex; *1* = CFD; *2* = Futures, etc.

Note 2: parameter *margin_mode* is equal to parameter *Margin calculation* in MetaTrader (*Symbol settings –> Calculation –> Margin Calculation*). The values that return in the response, will mean: *0* = Forex; *1* = CFD; *2* = Futures, etc.

51.   **'gettradingvolume'** returns information on the account's trading volume for the specified period.

- ✓ **Action**: *gettradingvolume*
- ✓ **Required** request parameters: *login, from, to*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, users, size, request_id*

✓ **SCSVLineFormat**: *login*; *order*; *symbol*; *open_price*; *close_price*; *profit*; *volume*; *open_time*; *close_time*; *conversionPrice*; *comment*; *commission*; *agent_commission*; *cmd*

Note 1: in parameter *login*, accounts are separated by ';'

Note 2: in parameter *cmd*, specify the order operation type (for the *cmd* parameter values, please see **Table 3** on p.**8**).

Note 3: *gettradingvolume* does not consider **pending** orders.

52. **'getusersonline'** returns amount of all users currently logged to the server, and lists their logins and IP-addresses.

✓ **Action**: *getusersonline*

✓ **Required** request parameters: none

✓ **Optional** request parameters: *request_id*

✓ **Answer** parameters: *result, count, request_id*

✓ **SCSVLineFormat**: *login*; *ip*

53. **'getversion'** allows getting the current version of *WebAPI*.

✓ **Action**: *getversion*

✓ **Required** request parameters: none

✓ **Optional** request parameters: *request_id*

✓ **Answer** parameters: *result, version, request_id*

✓ **SCSVLineFormat**: not used

54. **'modifyaccount'** action enables trader to change any account parameters (like in *createaccount*) except for *password*, *password_investor*, and *enable_change_password*.

✓ **Action**: *modifyaccount*

✓ **Required** request parameters: *login*

✓ **Optional** request parameters: *name*, *leverage*, *group*, *agent*, *email*, *id*, *address*, *city*, *state*, *zipcode*, *country*, *phone*, *password_phone*, *comment*, *enable*, *enable_read_only*, *send_reports*, *status*, *request_id*

✓ **Answer** parameters: *result, login, request_id*

✓ **SCSVLineFormat**: not used

55. **'modifygroup'** is to specify and/or change the group settings.

✓ **Action**: *modifygroup*

✓ **Required** request parameters: *name*, *currency*

✓ **Optional** request parameters: *signature, company, smtp_server*, *smtp_login, smtp_password, support_email, templates, enable*, *default_leverage, default_deposit, timeout, adv_security*, *copies, reports, margin_call, margin_mode, margin_stopout*, *news, check_ie_prices, maxpositions, interestrate, use_swap*, *close_reopen, hedge_prohibited, close_fifo, margin_type*, *archive_period*, *archive_max_balance,archive_pending_period,stopout_skip_hedged, request_id*

✓ **Answer** parameters: *result, group, request_id*

✓ **SCSVLineFormat**: not used

56. **'modifygroupswaps'** action gives an opportunity for broker to change swaps' size for a group.
- ✓ **Action**: *modifygroupswaps*
- ✓ **Required** request parameters: *name*, *symbol*, *swap_long*, *swap_short*, *margin_divider*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result*, *request_id*
- ✓ **SCSVLineFormat**: not used

57. **'modifyorder'** action allows the open order SL/TP changing.

Note: you may set incorrect SL/TP values as well – upon such modification, the order will be closed on the specified price.
- ✓ **Action**: *modifyorder*
- ✓ **Required** request parameters: *order*, *sl*, *tp*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result*, *request_id*
- ✓ **SCSVLineFormat**: not used

58. **'modifypending'** action enables to change settings in pending orders, e.g. their SL/TP level, volume, price.
- ✓ **Action**: *modifypending*
- ✓ **Required** request parameters: *order*, *volume*, *price*
- ✓ **Optional** request parameters: *sl*, *tp*
- ✓ **Answer** parameters: *result*, *request_id*
- ✓ **SCSVLineFormat**: not used

Note: *1* set as *volume* means **1 lot**.

59. **'openorder'** action opens a trade on a specific account.
- ✓ **Action**: *openorder*
- ✓ **Required** request parameters: *login*, *symbol*, *volume*
- ✓ **Optional** request parameters: *cmd* (opens **buy** orders with no *cmd* specified), *price*, *sl*, *tp*, *comment*, *request_id*
- ✓ **Answer** parameters: *result*, *orderid*, *request_id*
- ✓ **SCSVLineFormat**: not used

Note: *100* set as *volume* means **1 lot**.

60. **'openpending'** opens a pending order on a specific account.
- ✓ **Action**: *openpending*
- ✓ **Required** req parameters: *login*, *symbol*, *cmd*, *volume*, *price*
- ✓ **Optional** request parameters: *sl*, *tp*, *comment*, *request_id*
- ✓ **Answer** parameters: *result*, *orderid*, *request_id*
- ✓ **SCSVLineFormat**: NOT used

Note: *100* set as *volume* means **1 lot**.

61. **'pumping'** action enables traders to subscribe on:
- Trades
- News
- Quotes
- Margin calls

Thus, it returns data depending on the *pumping* type specified.

- ✓ **Action**: *pumping*
- ✓ **Required** request parameters: *trades / news / quotes / margin calls* (where '/' actually means **and/or**)
- ✓ **Optional** request parameters: *request_id*
- ✓ **Request example**: *action=pumping&trades=1&quotes=1*
- ✓ **Answer** parameters: depend on the chosen info type(s). See all 4 types **answer parameters** below in **Pumped data**.
- ✓ **SCSVLineFormat**: used in all types of returned information.

Pumped data:

1) **For** *trades=1*:

trades; *login*; *order*; *symbol*; *open_price*; *close_price*; *profit*; *volume*; *open_time*; *close_time*; *comment*; *commission*; *commission_agent*; *cmd*; *sl*; *tp*; *storage*; *act_type*

Note: parameter *act_type* **values**:

- ➢ added;
- ➢ closed;
- ➢ deleted;
- ➢ updated.

2) **For** *news=1*:

news; *key*; *time*; *ctm*; *topic*; *category*; *keywords*; *body*; *readed*; *priority*; *langid*

3) **For** *quotes=1*:

quotes; *symbol*; *bid*; *ask*

4) **For** *margincalls=1*:

margincalls; *login*; *group*; *leverage*; *updated*; *balance*; *equity*; *volume*; *margin*; *margin_free*; *margin_level*; *margin_type*; *level_type*

Note: parameter *level _type* has the following **values**:

- ➢ *0* – marginlevel_ok
- ➢ *1* – marginlevel_margincall
- ➢ *2* – marginlevel_stopout

So, **each returned message** starts with trades, news, quotes, or margin calls mark.

62. **'sendmail'** allows email sending to a specific account with indicating the sender data and the message details.
   - ✓ **Action**: *sendmail*
   - ✓ **Required** request parameters:*to_login*, *from_login*, *subject*, *body*
   - ✓ **Optional** request parameters: *request_id*
   - ✓ **Answer** parameters: *result*, *request_id*
   - ✓ **SCSVLineFormat**: not used

63. **'servertime'** returns the server time set in *Unix time* format.
   - ✓ **Action**: *servertime*
   - ✓ **Required** request parameters: none

- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, time, request_id*
- ✓ **SCSVLineFormat**: not used

64. **'transferbalance'** action enables to send funds from one account to another.

- ✓ **Action**: *transferbalance*
- ✓ **Required** request parameters: *receiver_logins, receiver_comment, from_logins, from_comment, amount*
- ✓ **Optional** request parameters: *request_id*
- ✓ **Answer** parameters: *result, size, request_id*
- ✓ **SCSVLineFormat**: *login*; *balance*; *credit*; *equity*; *margin*; *margin_free*; *success_flag*

65. **'updatepluginvalue'** request allows updating the needed parameter of a specific plugin.

- ✓ **Action**: *updatepluginvalue*
- ✓ **Required** request parameters: *name, param*
- ✓ **Optional** request parameters: *value, request_id*
- ✓ **Answer** parameters: *result, request_id*
- ✓ **SCSVLineFormat**: not used

Note: *name* means full name of *.dll* file of the needed plugin.

4. Important to know

1. If you set the **same** address in External IP and MT4 Server, the **ports** are necessary to be **different**, otherwise Socket Bind Error may occur.

2. The listed requests were designed for **traders**, however, some of these can be used by **brokers** as well.

3. If you have **more than 1** request active, the priority will be on the **higher** string.

4. All **required** parameters have to be specified for **any** request.

5. Please **be attentive to** the parameter *volume* while setting it.
   1) for *addorder, openorder, openpending, 100* means **1 lot**.
   2) for *bulkmodify, modifypending*, setting *1* means **1 lot**.

5. Contact us

Hopefully, you found this user manual clear, precise, and useful.

If you **have** any questions or suggestions about the product,

– do not hesitate to contact your **personal account manager**, or our

   technical support team via:

   • **skype:** tools4brokers.support
   • **email:** support@t4b.com
   • **phone:** +44-20-8133-8385 UK,
                   +852-8125-7512 HK
   • **telegram:** @t4b_support
   • **live chat with our specialist from our website**.

Our team will also gladly consult you on how you may **adjust** the product
to your **individual** requirements.

If you have any suggestions on the manual:

– please, go ahead and write to our **technical support**, what you
   would like to see here, or what needs to be
   added/corrected/changed.
– **Thank you for your feedback!**