WEB SOCKETS

Programmer's Guide

Version 1.4

FUNDAMENTAL
INTERACTIONS

2020 February

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

DISCLAIMER AND RIGHTS GRANTED

The information contained in this document describing Fundamental Interactions implementation of the WebSocket protocol and any other information or documentation related to *Fundamental Interactions WebSocket API Specifications* message form are provided "as is", and neither Fundamental Interactions nor any of its affiliates makes any representation or warranty, express or implied, as to the contents of the *Fundamental Interactions WebSocket API Specifications* and each such person specifically disclaims any warranty of originality, accuracy, completeness, merchantability, fitness for a particular purpose or that it is error-free. By using the *Fundamental Interactions WebSocket API Specifications*, you are agreeing to assume the entire risk associated with its use.

The Fundamental Interactions shall have no liability for damages of any kind arising in any manner out of or in connection with your use of, or your inability to use, the *Fundamental Interactions WebSocket API Specifications*, whether direct, indirect, incidental, special or consequential arising under contract or otherwise, whether Fundamental Interactions or any of its affiliates has been advised of, or otherwise might have anticipated the possibility of, such damages.

Please note that Fundamental Interactions will require prior certification of any system designed to connect to its trading and market information as well as your participation from time to time in connection with the testing of changes or upgrades to F*undamental Interactions WebSocket API Specifications*.

The information contained in the *Fundamental Interactions WebSocket API Specifications* is proprietary and confidential information belonging to Fundamental Interactions and Fix Protocol Limited and/or their respective licensors or service providers, as applicable. Copyright and trade-mark rights and any other intellectual property in the *Fundamental Interactions WebSocket API Specifications* belong to the Fundamental Interactions, FIX Protocol Limited and/or its licensors or service providers, as applicable. Your permitted use of the *Fundamental Interactions WebSocket API Specifications*, in whole or in part, is limited to the non-exclusive, nontransferable, revocable, non-assignable, personal right for you only to build a connection between your systems and Fundamental Interactions trading system.

Contact Information:

Fundamental Interactions
147 West 26th Street, 3rd Floor
New York, NY 10001
support@fundamentalinteractions.com

2 Overview

The information in this Fundamental Interactions WebSocket document describes the adaptation of the standard for vendors and subscribers to communicate with the Fundamental Interactions quotation and execution platform.

The order entry API is a layer on top of the market data API. Clients of the order entry web socket session can subscribe for market data, and vice versa. A key benefit of using the web socket API is that the FI backend will automatically push existing account activities and any corresponding updates to the user upon login (order/trade/position). The update messages associated with these are listed in this document.

The web socket order API is in JSON message format. The JSON string is communicated via secure web socket ("wss://"), so messages coming in and out are encrypted and protected by the underlying secure socket layer. Additional FI security measures exist, such as the login process, where the client is required to provide a password for verification. The client will need to encrypt the password string using the methods below described below. Other from this process, every else is straightforward.

3 LOGIN

Once client establishes the web socket connection, the client has 30 seconds to complete the login process. Otherwise, the socket will be closed by FI.

3.1 CHALLENGE

First client needs to send a challenge request to the system.

3.1.1 SAMPLE REQUEST CHALLENGE

```
{“type”:”challenge”}
```

In response, FI backend will send the customer a challenge response.

3.1.2 SAMPLE REQUEST RESPONSE TO CHALLANGE

```
{“result”:”OK”,”type”:”challenge”,”key”:”MIGfMA0GCSqGSIb3DQEBAQUAA4GNA
DCBiQKBgQCvBXTe278Lwg2MoI7iGKolSYuF+sNFKrsZplxCN9x0kItU3KIf8+1q60ILLwLewCEf7foxzp
Wp32j9YYU9vNBghuJ7BHcDYTffTRcv+QdNno491j701Hq7DIw13AGCQQTRcnfclvblnytIEWoQsiUv
PJcdiWgqJIX3IQGA47a+uwIDAQAB”}
```

3.2 KEY FIELD

The response contains a “key” field, which contains a public key string the client must use to encrypt the password field. The “key” field is a 64 base ASCII print out of the binary byte array for the corresponding public key of asymmetric public/private key pair. FI backend will hold the private key while sending clients the public key. Client will first convert the ASCII back to binary byte array. Then use that binary byte array to create a “RSA” public key instance. Afterwards, client will use that RSA public key instance to encrypt info required in this order entry API, e.g. password encryption.

3.2.1 JAVA SAMPLE

The sample code below is a java-based example of how to recreate a public key using key value from the challenge response and using that public key to encrypt a password (“test123”). The output binary array is then be translated into ASCII string output, which can be used in “pass” field for the login message.

```
byte[] array = javax.xml.bind.DatatypeConverter.parseBase64Binary(key);
KeyFactorykf = KeyFactory.getInstance("RSA");
publicKey = kf.generatePublic(new X509EncodedKeySpec(array));
Cipher = Cipher.getInstance("RSA");
cipher.init(Cipher.ENCRYPT_MODE, publicKey);
byte[] result = cipher.doFinal ("test123".getBytes("UTF-8"));
String output = javax.xml.bind.DatatypeConverter.printBase64Binary(result);
```


After encryption of password, the next step is to send in the login request.

The string in the "pass" field is from the output result in sample above

```
{"type":"login","userid":"USER","pass":"s7UW26iGE/iVfk2ihPYIcyzRqZRi/Ztb23UNMomf3xrBzGK
UHKzfNwZe5PIR/0zvfevYvkJnKLQVhR4U9/kObD/Ir0z6mBfLLgFwEcRm08jYI/nk7lDU+W32PqduT
OCThlkXYueQslK54vR9rKvMs="}
```

In response, FI backend will validate the user/password.

Below is a sample of failure and success response. If login fails, the FI backend will terminate the socket.

3.2.2 SAMPLE LOGIN SUCCESS

```
{"result":"invalid user/password","type":"login"}
```

3.2.3 SAMPLE LOGIN FAILURE

```
{"result":"OK","type":"login"}
```

4 MARKET DATA

After login, the following market data subscription is available to the client:

4.1 SUBSCRIPTION REQUEST MARKET DATA

The client may subscribe for the following types:

```
{
        "request":[
                {
                "msg":"book",
                "security":"BTCUSD",
                "dest":"GDMI
                }],
        "type":"subscribe"
}
```

4.1.1 TABLE OF MESSAGE TYPES

Market Data Type	Message
Internal Market Data	"msg":"ibook"
External Market Data	"msg":"book"
Trade Information	"msg":"trade"
Individual Symbol	"msg":"itrade"

"security":Individual symbol, for internal book feed ("ibook"/"itrade"), you can use "*"
"dest": required for external market, "GDAX"/"GEMI"/"BINA"

The internal market data, where you can use "*" to get all subscription.

"book" is for external market data, and you must provide each individual symbol in subscription.

"trade"/"itrade" is for corresponding external/internal last sale

4.1.2 SUBSCRIPTION RESPONSE MARKET DATA

{“security”:”BTCUSD”,”books”:[{“side”:”S”,”act”:”U”,”src”:”GEMI”,”pric
e”:7014.19,”qty”:0.02014047,”id”:6988929507302077441,”key”:”NA”},{“act”:”R”,”id”
:6988929507302197249},{“act”:”R”,”id”:6988929507302034433},{“act”:”R”,”id”:69889
29507301808128},{“side”:”B”,”act”:”U”,”src”:”GDAX”,”price”:7011,”qty”:0.11097064
,”id”:6988929507302013952,”key”:”NA”}],”type”:”book”}

4.1.3 TABLE OF FIELD DEFINITIONS

Field Name	Definition
security	symbol name
side	“B” for buy “S” for sell
act	“A”, add entry, “U” update existing entry”, “R” remove existing entry”
id	unique id for each entry
qty	Quantity of the entry
ioi	Boolean value, true or false, optional field, if not present, is false This is to indicate the quote is a IOI. IOI is used for negotiated market, it is not included in BBO calculation. IOI cannot be executed, but user can send counter order to initiator of IOI to start negotiation.
key	For external market, since it is aggregation, always
	for internal market, key is the order id, which corresponding to tag 37 in Execution Report message when you enter an order in fix or “refno” in web socket order messages

4.2 WEBSOCKET MARKET DATA TEST APP

Below is a link to the test application on the web to verify your results.

- http://testing.fi-edition.com/
- Password: NP99

5 TRADING API

After login, the following trading APIs functions are available to the client:

- “adduser”
- “deposit”
- “withdraw”
- “querydeposit”
- “querytrade”
- “logout”
- “getsymbols”
- “addorder”
- “cancelorder”

5.1 MESSAGE UPDATES FROM FI TO CLIENTS

FI will send current open orders and position info to the client upon login. If client places a new order, FI will send client an order update automatically. If there is trade, client will receive a trade update and corresponding position updates.

5.2 ADD USERS

Administrator account can use this command to add additional users. Ordinary user can use this command to update password.

Here is a request sample, again “pass” field needs to be encrypted using pubic key from challenge response.

5.2.1 SAMPLE RESUQEST ADD USERS

{“type”:”adduser”,”userid”:”testuser@gmail.com”,”pass”:”SZ7DMkMnU5qBszWkABPHsUjkPKol
cmWXbSdaydjsMOJG/CnujgQhm4vRWWr6BQmH9SsU7fFeheGYatTIL+n0yC+C0TqaOipSpC7rDSa
pveEsqHJTPgXYXeOb34wj76IHdv44UwT2Vr9Db+t6Q0K1j
wITy7/CxOqpYRea5g7Zt7A=”}

5.2.2 SAMPLE RESPONSES SUCCESS OR FAILURE

{“result”:”password
issue”,”userid”:”testuser@gmail.com”,”type”:”adduser”,”pass”:”SZ7DMkMnU5qBszWkABPHsUjk
PKolcmWXbSdaydjsMOJG/CnujgQhm4vRWWr6BQmH9SsU7fFeheGYatTIL+n0yC+C0TqaOipSpC
7rDSapveEsqHJTPgXYXeOb34wj76IHdv44UwT2Vr9Db+t6Q0K1j
wITy7/CxOqpYRea5g7Zt7A=”}

{“result”:”OK”,”userid”:”testuser@gmail.com”,”type”:”adduser”,”pass”:”ay5bjq6rONPIjDoEwHd
y6TZQzyzdcivSJQ6ReNdmAEQofuTC5WYnl1i5Qf7SA5c7owpN6IaODZkyb+WAN/gFylHzcSrmSme
DX9ZsuR/1AOEOeqCjlkgfEuLoaLdItYGcxOQoCDJwOl9sN81SbYxlurqq10ZLk53B9W+gAcwamZY=”
}

5.3 DEPOSIT& WITHDRAW

Administrator can make deposit or withdraw from client accounts.

5.3.1 SAMPLE REQUEST DEPOSIT & WITHDRAWAL

{“type”:”deposit”, “userid”:”0002”, “security”:”BTC”,”amount”:0.01}
{“type”:”withdraw”, “userid”:”0002”, “security”:”BTC”,”amount”:0.001}

For withdraw, you can provide optional input field “externaccount” for crypto blockchain transaction.

5.3.2 SAMPLE RESPONSE SUCCESSS OR FAILURE

{“amount”:0.01,”result”:”account
not found”,”userid”:”0002”,”security”:”BTC”,”type”:”deposit”}

{“amount”:0.01,”result”:”OK”,”userid”:”0004”,”security”:”BTC”,”type”:”deposit”}
{“amount”:0.001,”result”:”OK”,”userid”:”0004”,”security”:”BTC”,”type”:”withdraw”}

On return, if there is commission charged for action, “comm” field will show you how much you were charged.

5.4 QUERY DEPOSIT

5.4.1 SAMPLE REQUEST DEPOSIT

{“type”:”querydeposit”, “userid”:”0002”, “fromtime”:”1555522543000”}

Only administrator can specify “userid” field.

- “fromtime” field is optional, if set, it is to specify cutofftime to query

“comm”, optional, if there is commission for corresponding action, it will show you how much you were charged.

5.4.2 SAMPLE RESPONSE ON SUCCESS OR FAILURE

- “total_rec” tells you how many records there are.

For each record, there is “rec_no”, this is just for counting. Total number of records received should match value in “total_rec” field. It should be sort by “time”.

If there are more than 100 records, system can send you multiple response, each will have100 records.

{“result”:”OK”,”userid”:”0003”,”type”:”querydeposit”,”total_rec”:0}

{“result”:”OK”,”data”:[{“amount”:”0.001”,”time”:”1555522584928”,”secur
ity”:”BTC”,”rec_no”:1,”type”:”withdraw”},{“amount”:”0.01”,”time”:”1555522555008”
,”security”:”BTC”,”rec_no”:2,”type”:”deposit”}],”userid”:”0004”,”type”:”querydep
osit”,”total_rec”:2}

5.5 QUERY TRADE

5.5.1 SAMPLE REQUEST TRADE

{“type”:”querytrade”, “fromtime”:”1555522543000”}

This is to query the trades for a user.

Only administrator can specify “userid” field.

- “fromtime” field is optional, if set, it is to specify cutofftime to query
- “security” field is optional, if set, it is to return trades for that security only.

5.5.2 SAMPLE RESPONSE ON SUCCESS OR FAILURE

- “total_rec” tells you how many records there are.

For each record, there is “rec_no”, this is just for counting. Total number of records received should match value in “total_rec” field. It should be sort by “time”.

If there are more than 100 records, system can send you multiple response, each will have100 records.

{“result”:”OK”,”userid”:”0003”,”type”:”querytrade”,”total_rec”:0}
{“result”:”OK”,”data”:[{“refno”:”5GQBE9BCZOO0A3”,”category”:”CROX”,”ex
ecqty”:”10”,”execprice”:”30”,”traderefno”:”5GQBE9BCZOO0A32_5GQBE9BCZOO2”,”side”:
“B”,”trdtime”:”1557323619303”,”rec_no”:1,”security”:”MSFT”,”type”:”sale”},{“refn
o”:”5GQBE9BCZN4GA7”,”category”:”CROX”,”execqty”:”10”,”execprice”:”100.5”,”trader
efno”:”5GQBE9BCZN4GA72_5GQBE9BCZN4I”,”side”:”B”,”trdtime”:”1557323603427”,”rec_n
o”:2,”security”:”ZVZZT”,”type”:”sale”}],”type”:”querytrade”,”total_rec”:2}

5.6 QUERY ORDER

5.6.1 SAMPLE REQUEST TRADE

{“refno”:”5GS08WTIE8CWA0”,”security”:”BTCUSD”,”type”:”queryorder”}

Must provide “security” and “refno” field (that is FI’s order refno)

{“result”:”OK”,”refno”:”5GS08WTIE8CWA0”,”data”:{“updtime”:”15575052350
85”,”tif”:”GTC”,”execamount”:”10000”,”qty”:”1000”,”security”:”BTCUSD”,”type”:”or
der”,”clientorderid”:”ID1”,”liveqty”:”900”,”category”:”BRUT”,”refno”:”5GS08WTIE8
CWA0”,”price”:”100”,”execqty”:”100”,”side”:”S”},”security”:”BTCUSD”,”type”:”quer
yorder”}

{“result”:”order not found”,”refno”:”5GS08WTIE8CWA0”,”security”:”BTCUS
D”,”type”:”queryorder”}

5.7 LOGOUT

Sample request examples:

{“type”:”logout”}

This is for clean logout. Once backend receives this message, it will immediately terminate the web socket connection.

5.8 GETSYMBOLS

Sample request examples:

{“type”:”getsymbols”}
{“result”:”OK”,”data”:[{“pair”:false,”tradingsymbol”:false,”security”:
“USD”,”fractionbase”:100},{“pair”:false,”tradingsymbol”:false,”security”:”GROUP1
6”,”fractionbase”:1},{“pair”:false,”tradingsymbol”:false,”security”:”ETH”,”fract
ionbase”:10000},{“pair”:false,”tradingsymbol”:false,”security”:”COIN”,”fractionb
ase”:10},{“pair”:false,”tradingsymbol”:false,”security”:”CAD”,”fractionbase”:100
},{“pair_second”:”USD”,”pair”:true,”pair_first”:”BTC”,”tradingsymbol”:true,”secu
rity”:”BTCUSD”,”fractionbase”:1000000},{“pair_second”:”USD”,”pair”:true,”pair_fi
rst”:”BTCT”,”tradingsymbol”:true,”security”:”BTCTUSD”,”fractionbase”:1000000},{“
pair”:false,”tradingsymbol”:false,”security”:”BTCETH”,”fractionbase”:100000},{“p
air”:false,”tradingsymbol”:false,”security”:”BTC”,”fractionbase”:1000000}],”type
“:”getsymbols”}

This returns list of symbols defined in backend.

In your request, you can optionally set “security” for symbol and system will only return that symbol definition back to you. You can also set “tradingsymbolonly” to “Y” and system will return a list of permitted trading symbols back to you.

- “fractionbase” means how many decimal points precisions that system support, 1000 means for this instrument, system support decimal point up to 0.001
- “pair”, true/false, if symbol is a crypto pair, it is true, also, symbol will set “pair_first” and “pair_second” value to tell you which corresponding instruments are in this trading pair.
- “tradingsymbol”, true/false, if symbol is a tradable symbol.
- “fiat”, true/false, if symbol is crypto pair and has fiat in it or symbol itself is fiat.

5.9 TRADEHISTORY

Sample request examples:

{“type”:”tradehistory”,”security”:”BTCUSD”},
{“result”:”OK”,”data”:[{“lastprice”:”11
1”,”starttime”:”1556194354163”,”lastttime”:”1556194368559”,”volume”:”2”,”highpri
ce”:”125”,”blocktime”:”1556194320000”,”rec_no”:1,”numoftrades”:”2”,”startprice”:
“125”,”lowprice”:”111”},{“lastprice”:”135”,”starttime”:”1556194413020”,”lastttim
e”:”1556194413020”,”volume”:”1”,”highprice”:”135”,”blocktime”:”1556194380000”,”r
ec_no”:2,”numoftrades”:”1”,”startprice”:”135”,”lowprice”:”135”},{“lastprice”:”13
5”,”starttime”:”1556194529756”,”lastttime”:”1556194538364”,”volume”:”2”,”highpri
ce”:”160”,”blocktime”:”1556194500000”,”rec_no”:3,”numoftrades”:”2”,”startprice”:
“160”,”lowprice”:”135”},{“lastprice”:”140”,”starttime”:”1556194560524”,”lastttim
e”:”1556194560524”,”volume”:”1”,”highprice”:”140”,”blocktime”:”1556194560000”,”r
ec_no”:4,”numoftrades”:”1”,”startprice”:”140”,”lowprice”:”140”}],”security”:”BTC
USD”,”type”:”tradehistory”,”total_rec”:4}

Must set “security” field for the symbol that you query.

Optional field “extradays”, by default, system return last 24-hour trade info back to you. If you need extra history, set “extradays”, for example to “3” for 3 extra previous day history info.

The query returns the trade history in 1-minute block. If that minute has trade, a corresponding block will be return with summary of that minute high/low/first/last trade price as well as volume and num of trades info.

There can be multiple responses to one query if there are more than 100 entries in response. Each response will carry maximum of 100 entries. “total_rec” tells how many records there should be, and “rec_no” tell you current num of record.

5.10 ADD ORDER

Ordinary account can place orders from his own account. Alternatively, the user can pass his own order-id in “clientorderid”, which will be echoed back in corresponding “order”,” sale” update.

If order is successfully placed, the response will contain an FI assigned “refno”. The corresponding “refno” is also echoed in corresponding “order”/” sale” update. If a client needs to cancel an existing order, they must provide “refno” field in cancel request.

Available Options:

Field Name	Definition
side	Required, possible values “B”/“S”,
security	Required, instrument to trade, such as “BTCUSD”, “ETHEUR”, instruments that are allowed in exchange, must set
price	Required for most, price to trade, must set for limit order
qty	Required for most, amount of qty to trade, must set unless it is a market order, where you can set “amount” instead.
alo	Optional, true/false, default is false, if set to true, it means the order is for Adding Liquidity Only
dest	Optional, where this order should go,
tif	Optional, “IOC”/”GTC”/”GTT”/”FOK”, default is “GTC”, if “GTT” if “GTT”, must set “exptime” field
negotiate	Optional, default is false, if set to true, the order is a negotiated order, you must also specify “brokers” field where to send. For negotiated order, only send limit order.
ioi	Optional, default is false, if set to true, the order is an IOI for quote, the order itself cannot be executed.
brokers	Optional, default is not set. Only used when “negotiate” is set to true, it has format of “FIRM1,FIRM2,FIRM3...”. Note if you send same order to multiple firms, and if all firms try to execute your order, the maximum execqty is the your order qty.
displayqty	Optional, default is not set. If set, it is for reserve order, which will shows display qty of your order
ordertype	Optional, “LMT”/”MKT”, default is “LMT” and “price” field is required for limited order. For “MKT” order, specify “amount” field for corresponding currency amount, or if it is sell market order, you can still use “qty” field instead of “amount” field
stoptype	optional, “TSTOP”/”STOP”, default is not set, “STOP” is for stop order, “TSTOP” is for tailing stop order. For “STOP”, need to specify “stopprice”, for “TSTOP”, need to specify “traildelta”.
pegtype	optional, “MID”/”PRI”, default is not set, “MID” is to set midpoint peg order, “PRI” is to set primary peg order

5.10.1 SAMPLE REQUEST ADD ORDER

```
{“type”:”addorder”,”security”:”BTCUSD”,”side”:”B”,”price”:1.01,”qty”:0.1,”userid”:”ID1”}
```

To send an negotiated order (if system allows)

{"type":"addorder","clientorderid":"ID2", "price":
100,"side":"B","negotiate":true,"qty":0.1,"security":"BTCUSD",
"brokers":"LEHM,NITE"}

5.10.2 SAMPLE RESPONSES ADD ORDER

{“clientorderid”:”ID1”,”result”:”OK”,”refno”:”5FS8TX5EDN0WA0”,”price”:
1.01,”side”:”B”,”qty”:0.1,”security”:”BTCUSD”,”type”:”addorder”}

{“clientorderid”:”ID1”,”result”:”not enough fund”,”price”:
1.01,”side”:”B”,”qty”:0.1,”security”:”BTCUSD”,”type”:”addorder”}

5.11 CANCEL ORDER

Here is an example to cancel order, you must provide “refno” field, which is FI assigned order id.

5.11.1 SAMPLE REQUEST CANCEL ORDER

{“type”:”cancelorder”,”security”:”BTCUSD”,”refno”:”5FSDIAGCE768A0”}

5.11.2 SAMPLE RESPONSES CANCEL ORDER

{“result”:”OK”,”refno”:”5FSDIAGCE768A0”,”security”:”BTCUSD”,”type”:”cancelorder”}

{“result”:”order not
found”,”refno”:”5FSDIAGCE768A0”,”security”:”BTCUSD”,”type”:”cancelorder”}

5.12 Decline ORDER

Here is an example to decline an incoming negotiated order, you must provide “refno” field, which is FI assigned order id.

5.11.1 SAMPLE REQUEST Decline ORDER

{“type”:”declineorder”,”security”:”BTCUSD”,”refno”:”5FSDIAGCE768A0”}

5.13 Counter ORDER

To counter an incoming negotiated order, you must provide “refno” field, which is FI assigned order id from the incoming negotiated order, you also must provide side, qty, price from your side of view.

Once new counter is accepted, the original negotiated order is cancelled, the original initiator will receive an incoming negotiated order and you will get an new outgoing negotiated order message.

5.11.1 SAMPLE REQUEST Counter ORDER

{"type":"counterorder",”refno":"5OLOCZ06XT5DA0","price":200,"side":"S","qty":0
.1,"security":"BTCUSD”}

5.14 Execute ORDER

To execute an incoming negotiated order, you must provide “refno” field, which is FI assigned order id from the incoming negotiated order, you also must provide side, qty, price from your side of view.

If your execute request is accepted, both sides of negotiated order will receive trade update, just as normal order execution be would.

5.11.1 SAMPLE REQUEST EXECUTEORDER

{"type":"executeorder",”refno":"5OLOCZ06XT5DA0","price":200,"side":"S","qty":0
.1,"security":"BTCUSD”}

5.12 ORDER

“liveqty” field tells you much quantity is still open, if liveqty is 0, that means is the order is done.

“inbound”, true:false, only for negotiated order. True means someone send you an order, false means you sends someone an order

If client provides “clientorderid” in “addorder”, it will be echoed back in “order” update. Message shows that order is open, second message shows that order has 500 shares executed. The third messages show that order is canceled.

5.12.1 SAMPLE MESSAGES ORDER FIRST

{“liveqty”:”1000”,”refno”:”5FS7WOXHHU4WA3”,”category”:”STAGE”,”execqty
“:”0”,”price”:”30.5”,”execamount”:”0”,”qty”:”1000”,”clientorderid”:”ID123”,”security”:”MSFT”,
“type”:”order”}

{“liveqty”:”500”,”refno”:”5FS7WOXHHU4WA3”,”category”:”STAGE”,”execqty”
:”500”,”price”:”30.5”,”execamount”:”15250”,”qty”:”1000”,”clientorderid”:”ID123”,”security”:”
MSFT”,”type”:”order”}

{“liveqty”:”0”,”refno”:”5FS7WOXHHU4WA3”,”category”:”STAGE”,”execqty”:”
500”,”price”:”30.5”,”execamount”:”15250”,”qty”:”1000”,”clientorderid”:”ID123”,”security”:”MS
FT”,”type”:”order”}

5.13 SALE

- “refno” is a unique key for each trade,
- “ismaker”, true/false, if true, if liquidity provider
- “commsecurity”, optional, if there is commission charged and commision is charged on the receiving side of crypto pair, this field will contain corresponding instrument, say you buy BTCUSD and is charged a commission, “commsecurity” will be set to “BTC”, if sell BTCUSD, “commsecurity” will be set to “USD”.
- “comm”, optional, indicate corresponding commission charged related to “commsecurity”.

If client provides “clientorderid” in “addorder”, it will be echoed back in “sale” update.

5.13.1 SAMPLE TRADE UPDATES

{“refno”:”5FS7WOXHHU4WA3”,”category”:”CROX”,”execqty”:”30.5”,”execpric
e”:”30.5”,”traderefno”:”5FS7WOXHHU4WA32_5FS7WOXHHU4Y”,”clientorderid”:”ID123”,”secu
rity”:”MSFT”,”type”:”sale”}

5.14 POSITIONS

Upon login, client will receive initial positions and continuous updates afterwards.

5.14.1 SAMPLE POSITION MESSAGE

{“curpos”:”500”,”security”:”MSFT”,”type”:”position”}

5.15 QUERY POSITIONS

Query user position.

5.15.1 SAMPLE QUERY MESSAGE

{“userid”: “TEST1”, “type”:”querypos”}

“userid”,”firm”, superuser can provide “firm”,”userid” field.

“security” is optional, if input does not provide, query will get all position info for user.

5.15.2 SAMPLE QUERY RESPONSE MESSAGE

{“result”:”OK”,”firm”:”TEST”,”data”:[{“curpos”:”300”,”rec_no”:1,”secur
ity”:”ZVZZT”,”type”:”position”},{“curpos”:”300”,”rec_no”:2,”security”:”MSFT”,”ty
pe”:”position”}],”userid”:”TEST1”,”type”:”querypos”,”total_rec”:2}

{“result”:”no position”,”firm”:”TEST,”userid”:”TEST1”,”type”:”querypos”}

5.16 REGISTER WALLET

“registerwallet” This API registers a wallet from blockchain or from third party integration (such as Bitgo). If wallet does not exist for this user, system will create a new wallet. If corresponding wallet already exists, system will return existing wallet address back.

5.16.1 SAMPLE REGISTER WALLET MESSAGE

“security”, required

“userid”, optional, for superuser, should specify “userid” to indicate which account this registration is for.

{“security”:”TETH”,”type”:”registerwallet”}

5.16.2 SAMPLE REGISTER WALLET RESPONSE

{“result”:”cannot create wallet”,”security”:”TETH”,”type”:”registerwallet”}
{“result”:”OK”,”security”:”TETH”,”type”:”registerwallet”,”wallet”:”5d1
55a674ed9068e0376a5580fdadb54”}

5.17 QUERY WALLET

“querywallet” This command queries any existing wallets for the account. For superuser, it should set “account” field. If you specify “security” field, only corresponding symbol’s wallet will be returned. If no “security” field is specified, system will return all the wallets under the account.

5.17.1 SAMPLE QUERY WALLET MESSAGE

““account”, optional

“security”, optional

{ “security”:”TEST” ,”type”:”querywallet”}

5.17.2 SAMPLE QUERY WALLET RESPONSE

{“result”:”OK”,”data”:[{“userid”:”TEST123”,”security”:”TETH”,”type”:
“walletreg”,”wallet”:”5d155a674ed9068e0376a5580fdadb54”},{“userid”:”TEST123”,”
security”:”TBTC”,”type”:”walletreg”,”wallet”:”5d13cef2c4d18c95039e83b2a1e49b03”}
],”type”:”querywallet”}

{“result”,”no wallet”, “security”:”TEST” ,”type”:”querywallet”}

5.18 QUERY MULTI ORDERS

5.18.1 SAMPLE REQUEST MULTI ORDERS

{“type”:”querymultiorders”, “fromtime”:”1555522543000”}

This is to query the orders for a user.

Only administrator can specify “userid” field.

- “fromtime” field is optional, if set, it is to specify cutofftime to query
- “security” field is optional, if set, it is to return trades for that security only.

5.18.2 SAMPLE RESPONSE

{“result”:”OK”,”data”:[{“updtime”:”1562593041099”,”liveqty”:”0”,”refno
“:”5IF6Q7Y8LI8WA0”,”category”:”STAGE”,”execqty”:”0”,”price”:”100.59”,”side”:”B”,
“execamount”:”0”,”qty”:”100”,”rec_no”:1,”security”:”ZVZZT”,”type”:”order”},{“upd
time”:”1562593042339”,”liveqty”:”0”,”refno”:”5IF6Q7Y8LGPKA7”,”category”:”STAGE”,
“execqty”:”0”,”price”:”100.58”,”side”:”B”,”execamount”:”0”,”qty”:”100”,”rec_no”:
2,”security”:”ZVZZT”,”type”:”order”},{“updtime”:”1562593042339”,”liveqty”:”0”,”r
efno”:”5IF6Q7Y8LGPJA7”,”category”:”STAGE”,”execqty”:”0”,”price”:”100.57”,”side”:
“B”,”execamount”:”0”,”qty”:”100”,”rec_no”:3,”security”:”ZVZZT”,”type”:”order”},{
“updtime”:”1562593042339”,”liveqty”:”0”,”refno”:”5IF6Q7Y8LGPIA7”,”category”:”STA
GE”,”execqty”:”0”,”price”:”100.56”,”side”:”B”,”execamount”:”0”,”qty”:”100”,”rec_
no”:4,”security”:”ZVZZT”,”type”:”order”},{“updtime”:”1562593042339”,”liveqty”:”0
“,”refno”:”5IF6Q7Y8LGPHA7”,”category”:”STAGE”,”execqty”:”0”,”price”:”100.55”,”si
de”:”B”,”execamount”:”0”,”qty”:”100”,”rec_no”:5,”security”:”ZVZZT”,”type”:”order
“},{“updtime”:”1562593042339”,”liveqty”:”0”,”refno”:”5IF6Q7Y8LGPGA7”,”category”:
“STAGE”,”execqty”:”0”,”price”:”100.54”,”side”:”B”,”execamount”:”0”,”qty”:”100”,”
rec_no”:6,”security”:”ZVZZT”,”type”:”order”},{“updtime”:”1562593042339”,”liveqty
“:”0”,”refno”:”5IF6Q7Y8LGPFA7”,”category”:”STAGE”,”execqty”:”0”,”price”:”100.53”
,”side”:”B”,”execamount”:”0”,”qty”:”100”,”rec_no”:7,”security”:”ZVZZT”,”type”:”o
rder”},{“updtime”:”1562593042339”,”liveqty”:”0”,”refno”:”5IF6Q7Y8LGPEA7”,”catego
ry”:”STAGE”,”execqty”:”0”,”price”:”100.52”,”side”:”B”,”execamount”:”0”,”qty”:”10
0”,”rec_no”:8,”security”:”ZVZZT”,”type”:”order”},{“updtime”:”1562593042339”,”liv
eqty”:”0”,”refno”:”5IF6Q7Y8LGPDA7”,”category”:”STAGE”,”execqty”:”0”,”price”:”100
.51”,”side”:”B”,”execamount”:”0”,”qty”:”100”,”rec_no”:9,”security”:”ZVZZT”,”type
“:”order”},{“updtime”:”1562593042332”,”liveqty”:”0”,”refno”:”5IF6Q7Y8LGPCA7”,”ca
tegory”:”STAGE”,”execqty”:”0”,”price”:”100.5”,”side”:”B”,”execamount”:”0”,”qty”:
“100”,”rec_no”:10,”security”:”ZVZZT”,”type”:”order”}],”type”:”querymultiorders”,
“total_rec”:10}

5.19 RETRY CRYPTO DEPOSIT

Only super user can use it.

“blockhash”, required, this is the blockchain hash from the blockchain deposit transaction

“security”, required, corresponding crypto currency name

5.19.1 SAMPLE REQUEST RETRY CRYPTO DEPOSIT

{““type”:”querymultiorders”}

5.19.2 SAMPLE RESPONSE

{“amount”:”0.0001”,”result”:”OK”,”userid”:”BITTEST”,”blockhash”:”1c9a3
c059a17be3c6964e3be75ad05d34d44007c4536e7fa761c0f28dc6ce438”,”security”:”TBTC”,”
type”:”retrycryptodeposit”}

{“result”:”deposit already exist”,”blockhash”:”1c9a3c059a17be3c6964e3b
e75ad05d34d44007c4536e7fa761c0f28dc6ce438”,”security”:”TBTC”,”type”:”retrycrypto
deposit”}

6.0 SYMBOL REGISTRATION

6.10 SYMBOL REGISTRATION MESSAGE

“registersymbol”, only super user can use this command.

“security”, symbol, required, such as BTCUSD

“description”, optional, description of what this instrument is.

“fractionbase”, optional, 10000 mean 0.0001 minimum fraction

“pairseppos”, optional, specify crypto pair’s separation position for BTCUSD is 3

“istradingsymbol”, Boolean, optional, indicates if this is a trading instrument, default is set to no

- For example, “BTCUSD” should be a trading instrument, “BTC”, however, is not.

“pricepreciision”, optional, specify the price precision, 100 for 0.01

“minipriceincr”, optional, minimum price increment, in combined with pricepreciision

- For example, 5 for “minipriceincr” and 100 for “pricepreciision” means minimum increment is

“isfiat”, Boolean, optional, indicates if this is a fiat based instrument, default is no

“clearinginstrument”, optional, indicate underlying clearing instrument

“commissioninstrument”, optional, indicate underlying commissioninstrument.

“comm”, optional, indicates underlying commission rate, 5 means 5%.

“externalprecision”, optional, specify external custodian (such as Bitgo) precision

6.11 SAMPLE REQUEST SYMBOL REGISTRATION

{"type":"registersymbol","security":"WS001","description":"test symbol ws 01","fractionbase":"100","istradingsymbol":true},

6.12 SAMPLE RESPONSE

{"result":"OK","security":"WS001","fractionbase":"100","description":"test symbol ws 01","type":"registersymbol","istradingsymbol":true}

Notes

FUNDAMENTAL INTERACTIONS

Sales & Development Office
147 W. 26th Street. 300
New York, NY 10001
Office: (212) 725-3509 | Corporate: 212-845-9077
sales@fundamentalinteractions.Com

Support Office
Hudson Street, Hoboken, NJ 07030
Support Phone: 888-851-1369
support@fundamentalinteractions.com

Development Office
Development Office
1500 District Avenue, 2nd Floor, Burlington,
MA 01803

Europe Development & Support Office
Kharkov, Ukraine
61000