

# Pre-Trade Risk Websocket



**FUNDAMENTAL  
INTERACTIONS**

2020 January

# TABLE OF CONTENTS

<b>2</b>	<b>Overview.....</b>	<b>1</b>
<b>3</b>	<b>LOGIN.....</b>	<b>2</b>
3.1	CHALLENGE.....	2
3.1.1	SAMPLE REQUEST CHALLENGE.....	2
3.1.2	SAMPLE REQUEST RESPONSE TO CHALLENGE.....	2
3.2	KEY FIELD.....	2
3.2.1	JAVA SAMPLE.....	2
3.2.2	SAMPLE LOGIN SUCCESS.....	3
3.2.3	SAMPLE LOGIN FAILURE.....	3
3.3	MESSAGES FROM SERVER TO CLIENT.....	3
3.3.1	"risk_setting".....	3
3.3.2	"current_risk".....	4
3.4	MESSAGES FROM CLIENT TO SERVER.....	5
3.4.1	"account_update".....	5
3.4.2	"history_query".....	6



# 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 *Fundamental 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](mailto: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 CHALLENGE

```
{"result":"OK","type":"challenge","key":"MIGfMA0GCSqGSIb3DQEBAQUAA4GNA  
DCBiQKBgQCvBXTe278Lwg2Mol7iGKoLSYuF+sNFKrsZplxCN9x0kItU3KIf8+1q60ILLwLewCEf7foxzp  
Wp32j9YYU9vNBghuJ7BHcDYTffTRcv+QdNno491j701Hq7Dlw13AGCQQTRcnfclvblnytlIEWoQsiUv  
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/0zvfeyYvkJnKLQVhR4U9/kObD/Ir0z6mBfLLgFwEcRm08jYI/nk7lDU+W32PqduT
OCThlkXYueQsIk54vR9rKvMs="}

```

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"}

```

## 3.3 MESSAGES FROM SERVER TO CLIENT

For risk update, client automatically receive two types of messages "risk\_settings" and "current\_risk"

### 3.3.1 "risk\_setting"

If there is a db risk record, client will receive a "risk\_setting" message for corresponding settings.

“firm” and “trader” combination is the key.

If admin makes additional change to risk setting, you will receive an update also.

```
{
  "washcheck": "",
  "maxopenorders": "",
  "shortsale_monitor": "",
  "maxordershares": "",
  "totalallowamount": "",
  "type": "risk_setting",
  "maxpossible": "",
  "marginlimit": "",
  "SpeedChk": "",
  "symbolfilter": "",
  "optiontotalallowqty": "",
  "equitymaxordershares": "",
  "DupOrd": "",
  "percentagelimit": "",
  "optionmaxdollaramount": "",
  "cumnetotvalue": "",
  "trader": "COUNTRY",
  "active_account": "",
  "optiontotalallowamount": "",
  "fx_user": "N",
  "firm": "LEHM",
  "maxdollaramount": "",
  "maxnumpos": "",
  "totalallowqty": "",
  "optionmarginlimit": "",
  "optionmaxordersshares": "",
  "equitymaxdollaramount": "",
  "haltcheck": "",
  "isocheck": "",
  "blockoc": ""
}
```

### 3.3.2 "current\_risk"

System will send out continuous updates about the current risk values for this account. This update reflect risk values that account is using.

```
{
  "totalpendingorders": "0",
  "totalrejectedorders": "0",
  "currentoptioncumnetqty": "",
  "type": "current_risk",
  "maxdailyqty": "0",
  "currentoptionspeed": "",
  "cumnetqty": "",
  "maxspeed2": "0",
  "currentcumopenqty": "0",
  "optiontotalallowqty": "",
  "currentoptiontotalallowqty": "",
  "totaltrades": "0",
  "rbhvalue": "0.0",
  "cumopenqty": "",
  "currentcumnetotvalue": "0.0",
  "leveragebpthreshold": "0.0",
  "optioncumopenvalue": "",
  "maxmargin": "0.0",
  "rbhthreshold": "0.0",
  "optioncumopenqty": "",
  "totalallowqtyo": "0",
  "optionmaxspeed": "",
  "cumshortvalue": "",
  "totalallowqtyf": "0",
  "totalallowqtye": "0",
  "totalallowamounto": "0.0",
  "totalallowamounte": "0.0",
  "cumnetvalue": "",
  "currentspeed": "0",
  "currentoptioncumnetvalue": "",
  "totalallowamountf": "0.0",
  "currentoptioncumopenqty": "",
  "optionmarginlimit": "",
  "currentoptionmaxdollaramount": "",
  "currentspeed2": "0",
  "currentcumshortvalue": "0.0",
  "currentcumopenvalue": "0.0",
  "currentoptiontotalallowamount": "",
  "maxspeed": "0",
  "totalorders": "0",
  "cumshortqty": "",
  "leveragebpvalue": "0.0",
  "currentmargin": "0.0",
  "currentoptioncumshortvalue": "",
  "currentoptioncumopenvalue": "",
  "currentoptioncumlongvalue": "",
  "optioncumlongqty": "",
  "currenttotalallowqtyo": "0",
  "currentcumnetqty": "0",
  "currentqty": "0",
  "cumopenvalue": "",
  "optioncumnetqty": "",
  "currentoptionmargin": "",
  "currentamount": "0.0",
  "currentoptioncumlongqty": "",
  "optionmaxdollaramount": "",
  "trader": "0002",
  "currenttotalallowqtye": "0",
  "optiontotalallowamount": "",
  "optioncumnetvalue": "",
  "cumnetotvaluethreshold": "0.0",
  "currenttotalallowqtyf": "0",
  "currentcumshortqty": "0",
  "currentoptionma"
}
```

```

xordershares":"","cumlongvalue":"","currenttotalallowamountf":"0.0","firm":"LEHM
","optioncumshortqty":"","optioncumlongvalue":"","currenttotalallowamounto":"0.0
","currentcumlongqty":"0","currentcumnetvalue":"0.0","numofriskrejection":"0","o
ptioncumshortvalue":"","currentoptioncumshortqty":"","cumlongqty":"","accountche
ck":"","optionmaxordershares":"","lastmintrades":"0","currenttotalallowamounte":
"0.0","currentcumlongvalue":"0.0","maxdailyamount":"0.0","lastminorders":"0"}
received: {"totalpendingorders":"0","totalrejectedorders":"0","currentoptioncumn
etqty":"","type":"current_risk","maxdailyqty":"0","currentoptionspeed":"","cumne
tqty":"","maxspeed2":"0","currentcumopenqty":"0","optiontotalallowqty":"","curre
ntoptiontotalallowqty":"","totaltrades":"0","rbhvalue":"0.0","cumopenqty":"","cu
rrentcumnetotvalue":"0.0","leveragebpthreshold":"0.0","optioncumopenvalue":"","m
axmargin":"0.0","rbhthreshold":"0.0","optioncumopenqty":"","totalallowqtyo":"0",
"optionmaxspeed":"","cumshortvalue":"","totalallowqtyf":"0","totalallowqtye":"0"
,"totalallowamounto":"0.0","totalallowamounte":"0.0","cumnetvalue":"","currentsp
eed":"0","currentoptioncumnetvalue":"","totalallowamountf":"0.0","currentoptionc
umopenqty":"","optionmarginlimit":"","currentoptionmaxdollaramount":"","curren
tspeed2":"0","currentcumshortvalue":"0.0","currentcumopenvalue":"0.0","currentopti
ontotalallowamount":"","maxspeed":"0","totalorders":"0","cumshortqty":"","levera
gebvalue":"0.0","currentmargin":"0.0","currentoptioncumshortvalue":"","curren
toptioncumopenvalue":"","currentoptioncumlongvalue":"","optioncumlongqty":"","curr
enttotalallowqtyo":"0","currentcumnetqty":"0","currentqty":"0","cumopenvalue":"","
","optioncumnetqty":"","currentoptionmargin":"","currentamount":"0.0","currentopt
ioncumlongqty":"","optionmaxdollaramount":"","trader":"COUNTRY","currenttotalall
owqtye":"0","optiontotalallowamount":"","optioncumnetvalue":"","cumnetotvaluethr
eshold":"0.0","currenttotalallowqtyf":"0","currentcumshortqty":"0","currentoptio
nmaxordershares":"","cumlongvalue":"","currenttotalallowamountf":"0.0","firm":"L
EHM","optioncumshortqty":"","optioncumlongvalue":"","currenttotalallowamounto":
"0.0","currentcumlongqty":"0","currentcumnetvalue":"0.0","numofriskrejection":"0"
,"optioncumshortvalue":"","currentoptioncumshortqty":"","cumlongqty":"","account
check":"","optionmaxordershares":"","lastmintrades":"0","currenttotalallowamount
e":"0.0","currentcumlongvalue":"0.0","maxdailyamount":"0.0","lastminorders":"0"}

```

## 3.4 Messages from client to server

### 3.4.1 "account\_update"

If web user is permissioned to send this, he can use this message to change risk settings

"firm" and "trader" fields are required.

Here is example of request and response from server



```
{ "firm": "LEHM", "trader": "0004", "type": "account_update", "marginlimit": "7000" }
{ "result": "OK", "firm": "LEHM", "trader": "0004", "type": "account_update", "marginlimit": "7000" }
```

You can use fields from “risk\_setting” message to set corresponding risk parameters.

### 3.4.2 “history\_query”

This is to query history of what had been changed on this account

“firm” and “trader” fields are required.

Example of request and response from server

```
{ "firm": "LEHM", "trader": "0004", "type": "history_query" }
{ "result": "OK", "firm": "LEHM", "data": [ { "time": "1561149282630", "data": "Web user TEST changed marginlimit from 6000.0 to 7000.0", "rec_no": 1, "trader": "0004" }, { "time": "1561149366850", "data": "Web user TEST changed marginlimit from 7000.0 to 8000.0", "rec_no": 2, "trader": "0004" } ], "trader": "0004", "type": "history_query", "total_rec": 2 }
```

“total\_rec” tells you how many records should be returned. If there are more than 100 records, you will receive multiple updates. “rec\_no” tells you position of record that you received. Last “rec\_no” should be equal to “total\_rec”, that is also when you know you get all response.

## Notes

A series of horizontal dotted lines for taking notes, contained within a rounded rectangular frame.



## 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](mailto:sales@fundamentalinteractions.com)

### Support Office

Hudson Street, Hoboken, NJ 07030  
Support Phone: 888-851-1369  
[support@fundamentalinteractions.com](mailto:support@fundamentalinteractions.com)

### Development Office

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

### Europe Development & Support Office

Kharkov, Ukraine  
61000