Fortex

Confidential

xCloud6 Web API Specification

Document version 3.0

Date: Sep 02, 2021

Fortex, Inc.
203 Redwood Shores Parkway, Suite 640
Redwood Shores, CA 94065

Disclaimer

Contents

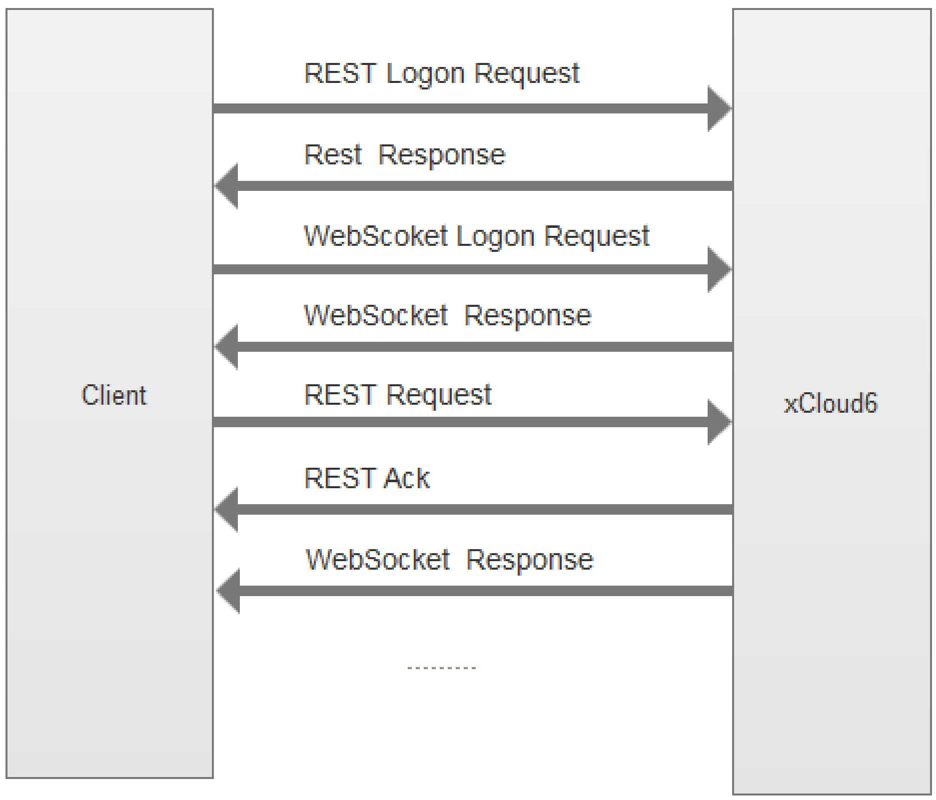
Ⅰ. Work Flow

Client may need to send a Login message via RESTful API to login first, if everything works then the server will respond a token, then client may immediately use this token to send another Login message through WebSocket.

After login successfully for both RESTful and WebSocket, client can send a request through RESTful API and expect an immediate ACK response from RESTful API to tell the request is succeeded or failed. Then the subsequent responses or updates will be sent to client through the WebSocket connection.

The token will be kept alive after the WebSocket is disconnected for a few minutes, you could use the token to re-login the WebSocket in a short time, or you can start over again from RESTful Logon message.

All messages are encrypted via SSL (HTTPS or WSS).



II. Supported Message Type

Category	Message Type
Session	RESTful Login
	WebSocket Login
Account	Query Account Info
	Query Position Info
Quote	Subscribe Quote
	Query History Chart
Trade	Query Open Orders Info
	Create New Order Request
	Create Order Cancel Request
	Create Order Cancel and Replace Request
	Query Tickets Request
News	Subscribe/Unsubscribe News Request
	Subscribe/Unsubscribe Calendar Request
Others	Save Profile
	Query Symbol List
	Query Forward Standard Tenors

III. Message Definition

All JSON keys and values are case sensitive unless there is an explicitly declared exemption.

Sever may response an ACK message for all RESTful messages except the Login message, it means the server has received the message from client and the client can expect a result from the WebSocket connection.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be "Ack"	Ack
ReferenceMsg	Y	object	The original message from the client	

1. Session

1) RESTful Login

Client needs to login first before sending any other message.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be "Login"	Login
UserInfo		Y			
➔	Login	Y	string	The account name you got from Fortex	USER1
➔	password	Y	string	The password for the account	PWD1

Sample: {"MT":"Login","UserInfo":{"login":" USER1","password":"PWD1"}}

2) WebSocket Login:

Client may establish a WebSocket connection right after the RESTful login is succeeded and send in a WebSocket login message as soon as possible. Server sends all updates to client through the WebSocket connection.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be "Login"	Login
UserInfo	Y			

➔	Login	Y	string	The account name you got from Fortex	USER1
➔	password	Y	string	The password for the account	PWD1
Tok		Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9

Sample:
{"MT":"Login","UserInfo":{"login":"USER1","password":"PWD1"},"Tok":"525197be1b9c44c98a2aef5d39f66ce9"}

2. Account

1) Account info

Client can query basic account information via sending a GetAcctInfo request.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be “GetAcctInfo”	GetAcctInfo
Tok		Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
UserInfo					
➔	login	Y	string	Which account you want to query	USER1

Sample:
{“MT”:”GetAcctInfo”,”Tok”:”525197be1b9c44c98a2aef5d39f66ce9”,"UserInfo":{"login":"USER1"}}

If everything is correct then the server will respond a message which is defined as the following

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be “GetAcctInfo”	GetAcctInfo
login	Y	string	Who sent the query request	USER1
Tok	Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9

LinkedAccts		N		If this object is missing, then it means the account doesn't link with other account.	
➔	LoginCate		int	The Logon account category setting	0
➔	Login		string	The linked account name	USER2
AcctSetting					
➔	viewOnly	Y	bool	View only or not, view only account cannot place any order	false
➔	LiqdMrgnRt	Y	double	Liquidation margin ratio	0.01
➔	method	Y	string	Standard RESTful method	PUT
➔	enable	Y	int	Account enabled or not	1
➔	mntnMrgnrt	Y	double	Maintenance margin Ratio	0.02
➔	lastOrdId	Y	int	The last order's ID	1234
➔	minQtyOnOrdEntry	Y	int	The required minimum quantity for a new order	1
➔	mrgnRt	Y	double	Initial margin ratio	0.04
➔	Credit	Y	bool	Is it a credit account or not	false
➔	Login	Y	string	The account name	USER123
AcctVal					
➔	dayCumQty	Y	int	Cumulated quantity of today	10000
➔	depoDraw	Y	double	Deposit/withdraw value	0
➔	numTickets	Y	int	Number of tickets	1
➔	bodBal	Y	double	BOD Balance	1.23E10
➔	clsPL	Y	double	Closed P&L	-197.5145
➔	method	Y	string	Standard RESTful method	PUT
➔	mntnMrgn	Y	double	Maintenance margin	3624553.565
➔	Commission	Y	double	Commission	0
➔	Liqdmrgn	Y	double	Liquidation Margin	1812276.55285
➔	Bal	Y	double	Account's current balance	1.0045E10
➔	login	Y	string	This account info belong to The Logon account name	USER1
➔	rqdMrgn	Y	double	Required Margin	751255.15551
CfgCommon					
➔	srvExecMode	Y	string	POSITION or TICKET mode	POSITION
➔	srvTz	Y	int	Server's time zone	-240
➔	srvCurTime	Y	string	Server's current time	20181019-04:40:57

UserInfo				
➔ Login	Y	string	The account which subscribe	USER1
Profile	Y	string	The account's previously saved profile	

Sample:
:{"LinkedAccts":[{"loginCate":0,"login":"USER1"},{"loginCate":0,"login":"USER2"}],"Tok":"4ecf3ccc793e4d60a7e97b05e47f9990","AcctSetting":{"viewonly":false,"liqdMrgnRt":0,"method":"PUT","enable":1,"mntnMrgnRt":1,"lastOrdId":26,"minQtyOnOrdEntry":1,"mrgnRt":1,"credit":true,"login":"USER1"},"AcctVal":{"dayCumQty":40000,"depoDraw":0,"numTickets":4,"bodBal":1.0E9,"clsPL":0,"method":"PUT","mntnMrgn":41150.296963549,"commission":0,"liqdMrgn":0,"bal":1.0E9,"login":"USER1","rqdMrgn":41150.296963549},"MT":"GetAcctInfo","CfgCommon":{"srvExecMode":"POSITION","srvTz":-420,"srvCurTime":"20181026-02:59:21"},"UserInfo":{"login":"USER1"},"Login":"USER1", "Profile":null }

2) Symbols List

Client can query basic symbols information by sending a SymListReq request.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be “SymListReq”	
TOK	Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9

Sample:
{"MT":" SymListReq","Tok":"525197be1b9c44c98a2aef5d39f66ce9"}

If everything is correct then the server will respond a message which is defined as the following

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be “SymListReq”	SymListReq
Login	Y	string	Who sent the query request	USER1
TOK	Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
CfgSymAry	Y		The symbol array setting	

➔	fwd	N		Only the forward have this setting	
	➔ dec	N	int	The forward decimal setting	8
➔	qtyDec	Y	int	The fx symbol decimal setting	3
➔	bodBid	Y	double	The BOD bid price	0.9288
➔	dec	Y	int	The symbol allow decimal point	5
➔	Method	Y	string	The server processing method: PUT = Update NEW = New DEL = Delete	PUT
➔	Pip	Y	double	The pip value point value 0.0001 mean 1 pip	4
➔	Sym	Y	string	Symbol,FX symbol must be in “CUR1/CUR2” format	AUD/CAD
➔	bodAsk	Y	double	BOD Ask	0.92889
➔	contractSz	N	int	the contract size value,	1
➔	type	Y	int	The symbol asset type: 3 = FX 7 = CFD 5 = Metal	3

Sample:
{"Tok":"663ff6bd8250474cb4785e5b24b15d33","CfgSymAry":[{"fwd":{"dec":8},"qtyDec":3,"bodBid":0.928819999999999,"dec":5,"method":"PUT","pip":4,"sym":"AUD/CAD","bodAsk":0.928899999999999,"type":3},……,{"qtyDec":0,"bodBid":2894.36,"dec":2,"baseCCY":"USD","method":"PUT","pip":1,"sym":"US500","bodAsk":2894.71,"contractSz":"1","type":7}],"MT":"SymListReq","Login":"USER1"}

3) Positions Info

Client can query basic positions information by sending a NetPosReq request.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be “NetPosReq”	NetPosReq
TOK	Y	string	The token from the RESTful	525197be1b9c44c98

			login response message	a2aef5d39f66ce9
UserInfo				
➔ login	Y	string	Which account you want to query	USER2

Sample:
{"MT":"NetPosReq","Tok":"4bc668b3959b4125a38db3b8ba67bc57","UserInfo":{"login":"USER2"}}

If everything is correct then the server will respond a message which is defined as the following

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be “NetPosReq”	NetPosReq
Login		Y	string	Who sent the query request	USER1
TOK		Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
PositionAry					
➔	sellDate	Y	string	settle Date	
➔	clsPL	Y	double	Close P&L	0
➔	Method	Y	string	The server processing method: PUT = Update NEW = New DEL = Delete	PUT
➔	OpenAvgPx	Y	double	All position avg price	0.94523
➔	Sym	Y	string	FX symbol must be in “CUR1/CUR2” format	AUD/CAD
➔	qty	Y	double	The symbol position value	2.1500-E7
➔	Commission	Y	double	The symbol commission	0
➔	secType	Y	string	time in force, 1=GTC, 3=IOC, 4=FOK	“”
➔	openRefPx	Y	double	base ccy	1.296
➔	Type	Y	int	Position type 1 = buy open 2 = sell open 3 = buy close 4 = sell close	2

➔	acct	Y	string	The position account	USER2
UserInfo		Y			
➔	Login	Y	string	The account which subscribe	USER2

Sample
{"Tok":"525197be1b9c44c98a2aef5d39f66ce9","MT":"NetPosReq","PositionAry":[{"settDate":"","clsPL":0,"method":"PUT","openAvgPx":1.13527,"sym":"EUR/USD","qty":10000,"commission":0,"secType":"","openRefPx":1,"type":2,"acct":"USER2"}],"UserInfo":{"login":"USER2"},"Login":"USER1"}

3. Quote

1) Subscribe Quote Update

Client can query basic Quote information by sending a QReq request.

JSON Key			Required	Value Type	Comment	Value Example
MT			Y	string	Must be "QReq"	QReq
Tok			Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
QReq			Y			
➔	sym		Y	string	must be entered symbols, support subscription of muti symbols	AUD/CAD
	➔	name	N	string	The forward symbols	EUR/USD
	➔	secType	N	string	Must be “FORWARD”	FORWARD
	➔	tenor	N	string	The forward symbol subscribes this tenor price.	1W
➔	sub_type		Y	int	this option means to subscribe or unsubscribe symbol. 1 = subscribe 2 = unsubscribe	1
➔	quote_type		Y	int	this option is means the subscribe quote type: 1 = L1 2 = L2 3= L1+L2	2

Sample:
{"MT":"QReq","QReq":{"sym":["AUD/CAD",{"name":"EUR/USD","secType":"FORWARD","tenor":"1W"},"USD/JPY"],"sub_type":1,"quote_type":1},"Tok":"525197be1b9c44c98a2aef5d39f66ce9"}

If everything is correct then the server will respond a message which is defined as the following, and there are 2 types of quote update. One is quote update. The other is the Market Depth quote update.

Quote update:

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be “Q”	Q
Q		Y		L1 Quote update info	
➔	a	Y	double	Ask price	0.929
➔	b	Y	double	Bid price	0.92913
➔	s	Y	string	symbol	AUD/CAD
➔	t	Y	string	the quote update time	20181019-10:33:16.552
➔	vDt	N	string	the forward vale date	20181030
➔	tnr	N	string	the forward tenor value	1W
➔	afp	N	double	ask forward point	0.000382
➔	bfp	N	double	bid forward point	0.000338
➔	sTp	N	string	Must be "FORWARD"	FORWARD

Sample:
{"Q":{"a":1.13279,"b":1.13275,"s":"EUR/USD","t":"20181116-10:32:50.372","afp":"","bfp":""},"MT":"Q"}
{"Q":{"a":1.133451,"b":1.13337,"s":"EUR/USD","t":"20181116-10:31:13.418","vDt":"20181127","tnr":"1W","afp":"0.000611","bfp":"0.00058","sTp":"FORWARD"},"MT":"Q"}

Market Depth quote update:

JSON Key			Required	Value Type	Comment	Value Example
MT			Y	string	Must be “MD”	MD
MD			N			
➔	a		Y		Ask price array	
	➔	p	Y	double	Ask price	1.133411
	➔	s	Y	double	the price qty value	1000

	➔	fp	N	double	ask forward point	0.000611
	➔	m	Y	string	The price MMID value	BOA
➔	b		Y		Bid price array	
	➔	p	Y	double	Bid price	1.13333
	➔	s	Y	double	the price qty value	50000
	➔	fp	N	double	Bid forward point	0.00058
	➔	m	Y	string	The price MMID value	BOA
➔	s		Y	string	symbol	EUR/USD
➔	t		Y	string	the quote update time	
➔	vDt		N	string	the forward vale date	
➔	tnr		N	string	the forward tenor value	
➔	sTP		N	string	Must be "FORWARD"	

Simple:

{"MT":"MD","MD":{"a":[{"p":1.13308,"aon":false,"s":50000,"m":"BOA"},{"p":1.13308,"aon":false,"s":10000,"m":"BOA"}],"b":[{"p":1.13304,"aon":true,"s":1000000,"m":"BOAA-AON"},{"p":1.13304,"aon":false,"s":50000,"m":"BOA"},{"p":1.13304,"aon":false,"s":10000,"m":"BOA"}],"s":"EUR/USD","t":"20181116-10:29:04.951"}}
{"MT":"MD","MD":{"a":[{"p":1.133411,"aon":false,"s":10000,"fp":"0.000611","m":"BOA"},{"p":1.133411,"aon":false,"s":50000,"fp":"0.000611","m":"BOA"}],"b":[{"p":1.13333,"aon":false,"s":10000,"fp":"0.00058","m":"BOA"},{"p":1.13333,"aon":false,"s":50000,"fp":"0.00058","m":"BOA"}],"s":"EUR/USD","t":"20181116-10:31:00.472","vDt":"20181127","tnr":"1W","sTp":"FORWARD"}}

2) SubScribe Chart History

Client can query basic Chart information by sending a ChartHisReq request.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be “ChartHisReq”	ChartHisReq
TOK		Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
ChartHisReq		Y			
➔	Sym	Y	string	Symbol,FX symbol must be in “CUR1/CUR2” format	EUR/USD
➔	Type	Y	int	The chart data type: 0=daily,	1

				1=minute	
➔	Days	Y	int	how many days data before the "endTime"	2
➔	reqId	Y	int	your own unique ID to identify this request, number only, it will be in the returning message.	1
➔	endTime	Y	string	To (ending at) what time(GMT0, YYYYMMDD-HH:MM:SS).	20181019-15:51:45

Sample
{"MT":"ChartHisReq","ChartHisReq":{"sym":"EUR/USD","type":1,"days":2,"reqId":1,"endTime":"20181019-15:51:45"},"Tok":"73bbc3f43c63450cb3b0837188af4c

If everything is correct then the server will respond a message which is defined as the following,

JSON Key		Required	Value Type	Comment	Value Example
TOK		Y	string	The token from the RESTful login response message	5a34f70edafb46cd99d3f326c57eb9aa
ChartDataAry		Y			
➔	c	Y	double	Close price	1.13429
➔	t	Y	string	Time	2018-10-26 11:36:00
➔	v	Y	int	volume	0
➔	h	Y	double	High price	1.13455
➔	l	Y	double	Low price	1.13429
➔	o	Y	double	open	1.134505
ChartHisReq		Y			
➔	symbol	Y	string	Symbol,FX symbol must be in “CUR1/CUR2” format	EUR/USD
➔	days	Y	string	how many days data before the "endTime"	2
➔	endTime	Y	string	To (ending at) what time (GMT0, YYYYMMDD-HH:MM:SS)	20181026-19:38:52
➔	type	Y	int	The chart data type: 0=daily, 1=minute	1

➔	reqId	Y	int	the request ID that subscribe	1

Sample:
{"Tok":"5a34f70edafb46cd99d3f326c57eb9aa","ChartDataAry":[{"c":1.13429,"t":"2018-10-26 11:36:00","v":0,"h":1.13455,"l":1.13429,"o":1.134505},......{"c":1.135925,"t":"2018-10-25 17:54:00","v":0,"h":1.13606,"l":1.13588,"o":1.13605}],"ChartHisReq":{"sym":"EUR/USD","days":"2","endTime":"20181026-19:38:52","type":"1","reqId":"1"}}

4. Trade

1) Subscribe Open Orders info

Client can query Open order information by sending a OpenOrdReq request.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	must be “OpenOrdReq”	OpenOrdReq
TOK		Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
UserInfo		Y			
➔	Login	Y	string	Which account you want to query.	USER1

Sample
{"MT":"OpenOrdReq","Tok":"74bea1308a254fc2821a8a1384cea892","UserInfo":{"login":"USER1"}}

If everything is correct then the server will respond a message which is defined as the following,

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	must be " OpenOrdReq "	OpenOrdReq
Login	Y	string	Who sent the query request	USER1
Tok	Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
OpenOrdAry	Y			

➔	tktType	Y	string	Fortex Ticket Type 0=to open, 1=to close, -1=not available	0
➔	ftxClOrdId	Y	string	Fortex close order ID	USER1:USER2:20
➔	txTime	Y	string	Transact Time	20181025-02:22:56
➔	execInst	Y	string	ExecInst, “G” = All or none – AON, “u” or empty = means allowing partial fill.	u
➔	sym	Y	string	FX symbol must be in “CUR1/CUR2” format	AUD/CAD
➔	px	Y	string	The trade prices. if the type =1, the px should be “0” If the type =U, the px is mean the Upper Limit price	0.2
➔	ordUser	Y	string	The order User	USER1
➔	ordQty	Y	string	The amount requested to trade.	10000000
➔	secType	Y	string	security type option: FOR =Forex. FORWARD =Forward	FOR
➔	tktId	Y	String	The ticket ID	
➔	tif	Y	string	Time in Force 1=GTC, 3=IOC, 4=FOK	1
➔	clOrdId	Y	string	Close Order ID	20
➔	OrgClOrdId	Y	string	Fortex Org close Order ID	USER1:USER2:20
➔	sl	Y	string	Stop Loss	0
➔	tag	Y	string	The order Comment tag. If trade by webtrader, the default value is “XCLOUD6”	XCLOUD6
➔	execType	Y	string	ExecNew = '0', ExecPartialfill = '1', ExecFill = '2', ExecDoneforDay = '3', ExecCanceled = '4',	0

				ExecReplace = '5', ExecPendingCancel = '6', ExecStopped = '7', ExecRejected = '8', ExecSuspended = '9', ExecPendingNew = 'A', ExecCalculated = 'B', ExecExpired = 'C', ExecRestated = 'D', ExecPendingReplace = 'E', ExecTrade = 'F', ExecTradeCorrect = 'G', ExecTradeCancel = 'H', ExecOrderStatus = 'I', ExecAllocation = 'J', ExecDistribution = 'K	
➔	ordstatus	Y	string	Order status 0=open, 1=partially closed, 2=closed	0
➔	side	Y	string	1=BUY 2=SELL	1
➔	lvQty	Y	string	The amount remaining – always 0 for a fill.	0
➔	method	Y	string	Must be “PUT”	PUT
➔	execDst	Y	string	execution destination The default value is “INTX”	INTX
➔	execId	Y	string	Same as the order ID	USER1:USER2:20
➔	avgPx	Y	string	Rate on the trade.	0.2
➔	tp	Y	string	Take profit	0
➔	filledQty	Y	string	The amount that’s been done, equal to the quantity on a fill.	0
➔	acct	Y	string	The order trade account	USER2
➔	ordtype	Y	string	order type, 1=Market, 2=Limit, 3=Stop, U=Threshold	2
➔	refTktId	Y	string	Fortex Reference Ticket No.	20
➔	stoppx	Y	string	price 2, when the type=U, the	0

				px2 mean the Lower Limit price	
➔	ftxClOrdIdFillNo	Y	string	FORTEX CL ORDER ID FILL NO:	USER1:USER2:20
➔	cumQty	Y	string	The amount that's been done, equal to the quantity on a fill.	100000
UserInfo		Y			
➔	Login	Y	string	The Open order info belongs to this account	USER2

Sample:
{"Tok":"9d99ea011ab4423bb6772938842af7fb","OpenOrdAry":[{"ftxClOrdId":"USER1:USER2:20","tickType":"0","txTime":"20181025-02:22:56","execInst":"u","ftxOrgClOrdId":"USER1:USER2:20","ftxRefTickNo":"20","sym":"AUD/CAD","px":"0.200000000000000","secType":"FOR","type":"2","ftxTickNo":"20","tif":"1","clOrdId":"20","sl":"0.0","tag":"FORTEX","execType":"0","orgUser":"AUTOTEST01","side":"1","lvQty":"10000.00000000","method":"PUT","px2":"0.0","execDst":"INTX","execId":"USER1:USER2:20","avgPx":"0.200000000000000","qty":"10000.00000000","tp":"0.0","filledQty":".00000000","user":"USER1","acct":"AUTOTEST02","ftxClOrdIdFillNo":"AUTOTEST01:AUTOTEST02:20:0:(null)","ordstatus":"0"}],"MT":"OpenOrdReq","UserInfo":{"login":"USER2"},"Login":"USER1"}

2) Query Tickets Request

Client can query Ticket Request information by sending a GetTicketsrequest.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be "GetTickets"	GetTickets
TOK		Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
GetTikcets					
➔	inclOp	Y	bool	Include open tickets or not	true
➔	inclCls	Y	bool	Include close tickets or not	true
➔	clsFrom	Y	string	Closed tickets from what time, Format is "YYY/MM/DD"	2015/12/25

➔	clsTo	Y	string	Closed tickets to what time, Format is “YYY/MM/DD”	2016/01/25

Sample:
{"MT":"GetTickets","Tok":"e5654bf1376c4e219de7abe1bd0ded60","GetTickets":{"inclOp":true,"inclCls":true,"clsFrom":"2015/12/25","clsTo":"2016/01/25"}}

If everything is correct then the server will respond a message which is defined as the following,

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be “GetTickets”	GetTickets
Login		Y	string	Who sent the query request	USER1
TOK		Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
GetTickets		Y			
➔	inclOp	Y	bool	Include open tickets or not	true
➔	inclCls	Y	bool	Include close tickets or not	true
➔	clsFrom	Y	string	Closed tickets from what time, Format is “YYY/MM/DD”	2015/12/25
➔	clsTo	Y	string	Closed tickets to what time, Format is “YYY/MM/DD”	2016/01/25
NetTicketAry		Y			
➔	Side	Y	string	1 or B =BUY 2 or S=SELL	S
➔	method	Y	string	Must be “PUT””	PUT
➔	cmsn	Y	double	commission for ticket open	0
➔	clsCnvRt	Y	double	optional, conversion rate for clsPx	0
➔	opQty	Y	double	remaining open quantity	3000
➔	sym	Y	string	FX symbol must be in “CUR1/CUR2” format	EUR/USD
➔	clsPx	Y	double	optional, closing price	0
➔	opCnvRt	Y	double	conversion rate for opPx	1
➔	clsQty	Y	double	optional, how much has been closed	0
➔	opPx	Y	double	open price	1.1370

➔	acct	Y	string	The Ticket info belongs this account	USER1
➔	status	Y	int	Order status 0=open, 1=partially closed, 2=closed	0

Sample:
{"Tok":"43051990efbb48f994a58d1318d27c50","GetTickets":{"clsTo":"2016/01/25","inclCls":true,"inclOp":true,"clsFrom":"2015/12/25"},"MT":"GetTickets","NetTicketAry":[{"side":"S","method":"PUT","Cmsn":0,"clsCnvRt":0,"opQty":30000,"sym":"EUR/USD","clsPx":0,"opCnvRt":1,"clsQty":0,"opPx":1.1370666666666667,"acct":"USER1","status":0},{"side":"B","method":"PUT","Cmsn":0,"clsCnvRt":0,"opQty":10000,"sym":"AUD/CAD","clsPx":0,"opCnvRt":1.31107,"clsQty":0,"opPx":0.9227699999999999,"acct":"AUTOTEST01","status":0}],"Login":"USER1"}

3) Create New Order Request

Client can create new order by sending an OrdReq request.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be “OrdReq”	OrdReq
TOK		Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
OrdReq		Y			
➔	acct	Y	String	The Trade account name	USER1
➔	sym	Y	string	Symbol,FX symbol must be in “CUR1/CUR2” format	EUR/USD
➔	secType	Y	string	security type option: FOR - this order is for forex trade FORWARD - this order is for forward trade.	FORWARD
➔	side	Y	string	1=BUY 2=SELL	1
➔	qty	Y	string	Order quantity	10000
➔	px	Y	string	The trade price. if the type =1, the px should be “0”	0

				If the type =U, the px means the Upper Limit price	
➔	type	Y	string	order type, 1=Market, 2=Limit, 3=Stop U=Threshold	1
➔	tif	Y	string	Time in Force 1=GTC, 3=IOC, 4=FOK	1
➔	sl	Y	string	Stop loss price	0
➔	tp	Y	string	Take profit price	0
➔	execDst	Y	string	execution destination The default value is "INTX"	INTX
➔	minQty	Y	stirng	The account min quantity	0
➔	stopPx	Y	string	Stop price, when eh type=3, then the order use this price to trade	0
➔	qtyRsrv	Y	string	order quantity reserved	0
➔	maxShow	Y	string	Max show	50000
➔	execBrk	Y	string	Execution broker	""
➔	execInst	Y	string	"G" = All or none – AON, "u" or empty = means allowing partial fill.	"u"
➔	Px2	Y	string	price 2, when the type=U, the px2 means the Lower Limit price	"0"
➔	txTime	Y	string	The order transaction time	20180931-21:24:31.488
➔	handlInst	Y	string	HandlInst	1
➔	prNAgc	Y	string	order principal Agency	true
➔	slpg	N	string	Slippage, only Limit order support it. (Not work)	0
➔	tktType	Y	string	The Ticket Type: 0=open 1=close -1=not available	0
➔	tktNo	Y	string	the ticket#ID	0

➔	refTktNo	Y	string	when ticket type is close, then it's the ref ticket id.	0
➔	settDate	N	string	If the secType=FORWARD, need to send the settlement date to server.	20180919

Sample:

{
    "MT": "OrdReq",
    "OrdReq": {
        "acct": " USER_1001",
        "sym": "EUR/USD",
        "secType": "FOR",
        "side": "1",
        "qty": "1",
        "px": "0.888",
        "type": "2",
        "tif": "1",
        "sl": "0",
        "tp": "0",
        "execDst": "INTX",
        "minQty": "0",
        "stopPx": "0",
        "qtyRsrv": "0",
        "maxShow": "50000",
        "execBrk": "",
        "execInst": "",
        "px2": "0",
        "txTime": "20210715-3:24:59.811",
        "handlInst": "1",
        "prnAgc": "true",
        "slpg": "0",
        "tktType": "0",
        "tktNo": "0",
        "refTktNo": "0"
    },
    "Tok": "0f9211a3466f4fd187b0dd2f87ce5b84"
}

If everything is correct then the server will respond 3 message which is defined as the following. These 3 messages are the Enter execution Report, Order ACK message and Order Trade Result message.

REST.POST: URL=/WEBTRADER/rest, Data={"MT":"OrdReq","OrdReq":{"acct":"FORTEX_TINA","sym":"EUR/USD","secType":"FOR","side":"1","qty":"1","px":"0.888","type":"2","tif":"1","sl":"0","tp":"0","execDst":"INTX","minQty":"0","stopPx":"0","qtyRsrv":"0","maxShow":"50000","execBrk":"","execInst":"","px2":"0","txTime":"20210715-3:24:59.811","handlInst":"1","prnAgc":"true","slpg":"0","tktType":"0","tktNo":"0","refTktNo":"0"},"Tok":"0f9211a3466f4fd187b0dd2f87ce5b84"}
REST.Resp: {"ReferenceMsg":{"Tok":"0f9211a3466f4fd187b0dd2f87ce5b84","MT":"OrdReq","OrdReq":{"tktType":"0","txTime":"20210715-3:24:59.811","sym":"EUR/USD","execInst":"","handlInst":"1","px":"0.888","secType":"FOR","type":"2","maxShow":"50000","tif":"1","refTktNo":"0","clOrdId":"279","execBrk":"","sl":"0","side":"1","slpg":"0","prnAgc":"true","px2":"0","tktNo":"0","execDst":"INTX","qty":"1","qtyRsrv":"0","tp":"0","acct":"FORTEX_TINA","stopPx":"0","minQty":"0"}},"MT":"Ack"}
WebSkt Recv: {"MT":"ExecRp","OrdInfo":{"tktType":"-1","ftxClOrdId":"","txTime":"20210715-03:40:32","execInst":"u","sym":"EUR/USD","px":"0.888","ordUser":"FORTEX_TINA","ordQty":"1.0","secType":"FOR","ftxTickExecSeqNo":"","origOrdUser":"","tktId":"","refPx":"","tif":"1","clOrdId":"279","origClOrdId":"","settDate":"","execBrk":"","sl":"0","tag":"XC.ORD.USER.FORTEX_TINA","dealCcy":"","execType":"A","ordStatus":"0","side":"1","lvQty":"","method":"NEW","lastPx":"","cumQty":"0","px2":"0","execDst":"INTX","fillSeqNo":"","instType":"","lastQty":"","avgPx":"","tp":"0","filledQty":"","acct":"FORTEX_TINA","ordType":"2","refTktId":"","stopPx":"","ftxClOrdIdFillNo":""},"Login":"FORTEX_TINA"}

- Enter Execution Report:

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	String	Must be “ExecRp”	ExecRp
ExecRp		Y	string	Execution Report, the tag is always is null	null
OrdInfo				The original message from the client	
➔	ftxClOrdId	Y	String	Fortex close order ID	“”
➔	tktType	Y	String	Fortex Ticket Type 0=to open, 1=to close, -1=not available	-1
➔	txTime	Y	String	TransactTime	20181022-03:18:41
➔	execInst	Y	String	“G” = All or none – AON, “u” or empty = means allowing partial fill.	u
➔	sym	Y	String	FX symbol must be in “CUR1/CUR2” format	EUR/USD
➔	px	Y	Double	The trade prices. if the type =1, the px should be “0” If the type =U, the px means the Upper Limit price	0
➔	orduser	Y	String	logon to Trade Server Account	TWS_USER2
➔	ordQty	Y	String	The amount requested to trade.	10000
➔	secType	Y	String	security type option: FOR =Forex. FORWARD =Forward	10000
➔	ftxTickExecSeqNo	N	String	FortexTicketExecutionSeqNo	“”
➔	orgOrdUser	Y	String	Fortex order orig User	“”

➔	tktId	N	String	Fortex Ticket No	“”
➔	refPx	N	Double	Fortex reference Price	“”
➔	tif	Y	String	Time in Force 1=GTC, 3=IOC, 4=FOK	1
➔	clOrdId	Y	String	Closer Order ID	883
➔	OrgClOrdId	N	String	Fortex Org close Order ID	“”
➔	settDate	N	String	If the secType=FORWARD, need to send the settlement date to server.	""
➔	execBrk	N	String	Execution broker	“”
➔	sl	Y	String	Stop Loss	“”
➔	tag	N	String	The order Comment tag. If trade by webtrader, the default value is “XCLOUD6”	XCLOUD6
➔	execType	N	String	ExecNew = '0', ExecPartialfill = '1', ExecFill = '2', ExecDoneforDay = '3', ExecCanceled = '4', ExecReplace = '5', ExecPendingCancel = '6', ExecStopped = '7', ExecRejected = '8', ExecSuspended = '9', ExecPendingNew = 'A', ExecCalculated = 'B', ExecExpired = 'C', ExecRestated = 'D', ExecPendingReplace = 'E', ExecTrade = 'F', ExecTradeCorrect = 'G', ExecTradeCancel = 'H', ExecOrderStatus = 'I', ExecAllocation = 'J', ExecDistribution = 'K',	“”
➔	ordStatus	Y	string	Order status 0=open, 1=partially closed, 2=closed	0

➔	side	Y	String	1=BUY 2=SELL	1
➔	lvQty	N	String	The amount remaining – always 0 for a fill.	“”
➔	method	Y	String	The server processing method: PUT = Update NEW = New DEL = Delete	PUT
➔	lastPx	N	String	The price for this execution.	“”
➔	cumQty	N	String	The amount that’s been done, equal to the quantity on a fill.	0
➔	Px2	Y	String	price 2, when the type=U, the px2 means the Lower Limit price	0
➔	execDst	Y	String	execution destination The default value is “INTX”	INTX
➔	execId	N	String	Same as OrderID.	“”
➔	fillSeqNo	Y	String	OrderFillSeqno	“”
➔	instType	Y	String	OrderInstType	“”
➔	lastQty	N	String	Amount traded for this execution	“”
➔	avgpx	N	String	Rate on the trade.	“”
➔	tp	Y	String	Take profit	0
➔	filledQty	Y	String	The amount that’s been done, equal to the quantity on a fill.	“”
➔	acct	Y	String	Trade Account	USER1
➔	ordType	Y	String	order type, 1=Market, 2=Limit, 3=Stop, U=Threshold	1
➔	refTktId	N	String	Fortex Reference Ticket No.	“”
➔	stoppx	Y	String	Stop price	0
Login		Y	string	The trade account name	USER1

Sample:

{

```
"ExecRp": null,
"MT": "ExecRp",
"OrdInfo": {
    "ftxClOrdId": "",
    "tickType": "-1",
    "txTime": "20181022-03:18:41",
    "execInst": "u",
    "ftxOrgClOrdId": "",
    "ftxRefTickNo": "",
    "sym": "EUR/USD",
    "px": "0.008",
    "secType": "FOR",
    "ftxTickExecSeqNo": "",
    "type": "2",
    "ftxTickNo": "",
    "refPx": "",
    "tif": "1",
    "clOrdId": "883",
    "settDate": "",
    "execBrk": "",
    "sl": "0",
    "tag": "XCLOUD6",
    "execType": "",
    "orgUser": "",
    "side": "1",
    "lvQty": "",
    "method": "NEW",
    "lastPx": "",
    "cumQty": "0",
    "px2": "0",
    "execDst": "INTX",
    "execId": "",
    "fillSeqNo": "",
    "instType": "",
    "lastQty": "",
    "avgPx": "",
    "qty": "10000.0",
    "tp": "0",
    "filledQty": "",
    "user": "TWS_USER2",
    "acct": "USER1",
    "stopPx": "",
    "ftxClOrdIdFillNo": "",
```

"status": "0"
},
"Login": "USER1"
}

- Order ACK Execution Report

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be “ExecRp””	ExecRp
ExecRp	Y	string	Execution Report, the tag is always is null	Null
OrdInfo		string	The original message from the client	
Login	N	string	The Trade account	USER1

Sample:

{"ExecRp":null,"MT":"ExecRp",”OrdInfo":{"ftxClOrdId":"","tickType":"-1","txTime":"20181217-06:16:37","execInst":"u","ftxOrgClOrdId":"","ftxRefTickNo":"","sym":"EUR/USD","px":"0","secType":"FOR","ftxTickExecSeqNo":"","type":"1","ftxTickNo":"","refPx":"","tif":"1","clOrdId":"921","settDate":"","execBrk":"","sl":"0","tag":"XCLOUD6","execType":"","orgUser":"","side":"2","lvQty":"","method":"NEW","lastPx":"","cumQty":"0","px2":"0","execDst":"INTX","execId":"","fillSeqNo":"","instType":"","lastQty":"","avgPx":"","qty":"10000.0","tp":"0","filledQty":"","user":"TWS_USER2","acct":"CROTT_TEST","stopPx":"","ftxClOrdIdFillNo":"","status":"0"}, "Login":"USER1"}

- Order Trade Result Execution Report:

  - Reject Execution Report

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be “ExecRp”	ExecRp
ExecRp		Y	string		
➔	txTime	Y	string	Transact Time	20181028-18:41:17
➔	sym	Y	string	FX symbol must be in “CUR1/CUR2” format, for example: USD/JPY.	EUR/ISD
➔	px	Y	double	The trade prices. if the type =1, the px should be “0” If the type =U, the px means the Upper Limit price	0

➔	orduser	Y	string	The order user	TWS_DEMO
➔	ordQty	N	string	The amount requested to trade.	0
➔	SecType	Y	string	security type option, Default is FOR FOR =Forex. FORWARD =Forward	“”
➔	OrdId	Y	string	Order ID	869
➔	orgOrdUser	Y	string	Fortex order orig User	“”
➔	tktId	Y	string	Ticket ID, same as the order id	869
➔	ClOrdId	Y	string	Close order ID	869
➔	orgClordId	Y	string	Same as order ID	“”
➔	settDate	N	string	the settlement date to server.	""
➔	Commission		double	Commission	0
➔	execType		string	ExecNew = '0', ExecPartialfill = '1', ExecFill = '2', ExecDoneforDay = '3', ExecCanceled = '4', ExecReplace = '5', ExecPendingCancel = '6', ExecStopped = '7', ExecRejected = '8', ExecSuspended = '9', ExecPendingNew = 'A', ExecCalculated = 'B', ExecExpired = 'C', ExecRestated = 'D', ExecPendingReplace = 'E', ExecTrade = 'F', ExecTradeCorrect = 'G', ExecTradeCancel = 'H', ExecOrderStatus = 'I', ExecAllocation = 'J', ExecDistribution = 'K',	8
➔	ordStatus	Y	string	Order Status	8
➔	side	Y	string	Trade side 1=BUY 2=SELL	

➔	lvQty	Y	double	The amount remaining – always 0 for a fill.	10000
➔	method	Y	string	The server processing method: PUT = Update NEW = New DEL = Delete	NEW
➔	lastPx	Y	double	The price for this execution.	0
➔	cumQty	Y	double	The amount that's been done, equal to the quantity on a fill.	0
➔	ErrInfo	Y			
	➔ code	Y	string	Code to identify reason:	108
	➔ desc	Y	string	The err detail message	Incorrect To Open Or To Close
➔	execId	Y	string	Trade server execute ID	1540777277
➔	lastQty	Y	double	Amount traded for this execution	0
➔	avgPx	Y	double	Rate on the trade.	0
➔	convRate	Y	double	Conversion Rate	0
➔	acct	Y	string	Trade Account	USER1
➔	ordType	Y	string	order type, 1 = Market, 2 = Limit, 3 = Stop U = Threshold	2
➔	refTktID	Y	string	Reference ticket ID,	“”
➔	stopPx	Y	double	Stop price, when eh type=3, then the order uses this price to trade	0
OrdInfo		Y			
➔	clOrdID	Y	string	Same as Order ID	869
➔	method	Y	string	The server processing method: PUT = Update NEW = New DEL = Delete	DEL
➔	acct	Y	string	The Trade account	USER1
Login		Y	string	The Trade account	USER1

Sample:

{"ExecRp":{"txTime":"20181028-18:41:17","sym":"EUR/USD","px":0,"ordUser":"TWS_DEMO2","ordQty":0,"ordId":"869","tktId":"","clOrdId":"869","ordStat":8,"settDate":"","commission":0,"secType":0,"execType":"8","lvQty":10000,"orgUser":"","side":2,"method":"NEW","lastPx":0,"cumQty":0,"orgClOrdId":"","ErrInfo":{"code":108,"desc":"Incorrect To Open Or To Close"},"execId":"1540777277","lastQty":0,"avgPx":0,"convRate":0,"acct":"USER1","ordType":"2","refTktId":"","stopPx":0},"MT":"ExecRp","OrdInfo":{"clOrdId":"869","method":"DEL","acct":"USER1"},"Login":"USER1"}

- Filled Account update info Execution Report
  If the order is filled. the server will response the account update info to client.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be “ExecRp””	ExecRp
AcctVal					
➔	dayCumQty	Y	double	Total transaction qty on that day	10000
➔	depoDraw	Y	double	Deposit/ withdraw value	0
➔	unmTickets	Y	double	The number of the tickets	1
➔	bodBal	Y	double	BOD Balance	1.23E10
➔	clsPL	Y	double	Close P&L	-197.5145
➔	method	Y	String	Always be “PUT”	PUT
➔	mntnMrgn	Y	double	Maint. Margin	3624553.565
➔	Commission	Y	double	Commission	0
➔	Liqdmrgn	Y	double	Liqudation Margin	1812276.55285
➔	Bal	Y	double	Balance	1.0045E10
➔	Login	Y	string	The Trade account name	USER1
➔	rqdMrgn	Y	double	Required Margin	751255.15551
ExecRp		Y			
➔	txTime	Y	String	TransactTime	20181024-23:15:23
➔	sym	Y	String	FX symbol must be in “CUR1/CUR2” format, for example: USD/JPY.	EUR/USD
➔	px	Y	double	The trade prices. string type =1, the px should be “0” double If the type =U, the px means the Upper Limit price.	0
➔	ordUser	Y	String	Logon Trade server account	TWS_USER2
➔	ordQty	Y	Double	The amount requested to	0

				trade.	
➔	secType	Y	String	security type option,Default is FOR FOR =Forex. FORWARD =Forward	0
➔	ordId	Y	String	Order ID	842
➔	origOrdUser	Y	string	The order user	
➔	tktId	Y	string	Ticket ID	842
➔	clordId	Y	string	Same as Order ID	842
➔	orgClOrdId	Y	string	FORTEXORIGCLORDERID:	“”
➔	settDate	Y	string	Settlement Date	20181024
➔	commission	Y	double	Commission charged for this execution.	0
➔	execType	Y	string	ExecNew = '0', ExecPartialfill = '1', ExecFill = '2', ExecDoneforDay = '3', ExecCanceled = '4', ExecReplace = '5', ExecPendingCancel = '6', ExecStopped = '7', ExecRejected = '8', ExecSuspended = '9', ExecPendingNew = 'A', ExecCalculated = 'B', ExecExpired = 'C', ExecRestated = 'D', ExecPendingReplace = 'E', ExecTrade = 'F', ExecTradeCorrect = 'G', ExecTradeCancel = 'H', ExecOrderStatus = 'I',	F
➔	ordStatus	Y	String	0=open, 1=partially closed, 2=closed	0
➔	side	Y	string	1 = Buy 2 = Sell	2
➔	lvQty	Y	string	The amount remaining – always 0 for a fill.	0
➔	method	Y	string	refer to name space Method	NEW

➔	lastpx	Y	double	The price for this execution.	1.14134
➔	cumQty	Y	double	The amount that's been done, equal to the quantity on a fill.	10000
➔	execId	Y	string	The trade order id	TWS_USER2:USER1:842_15404481231
➔	lastQty	Y	double	Amount traded for this execution.	10000
➔	avgpx	Y	double	Rate on the trade.	
➔	convrate	Y	double	Conversion Rate	1
➔	acct	Y	string	the trade account name	USER1
➔	ordType	Y	string	order type, 1=Market, 2=Limit, 3=Stop U=Threshold	1
➔	refTktId	Y	string	Reference ticket ID	842
➔	stopPx	Y	double	Stop price, when eh type=3, then the order use this price to trade	0
OrderInfo					
➔	clOrdid	Y	string	Closed Order ID	
➔	method	Y	string	refer to name space Method	DEL
➔	acct	Y	srig	The Trade Account	USER1
PositionAry		Y			
➔	settDate	N	string	Settlement Date	“”
➔	clsPL	Y	string	Closed P&L	0
➔	clsRefPx	N	string	the close conversion rate to USD	
➔	method	Y	string	refer to namespace Method	PUT
➔	openAvgPx	Y	double	the open average price	1.14134
➔	sym	Y	string	FX symbol must be in “CUR1/CUR2” format, for example: USD/JPY.	EUR/USD
➔	qty	Y	string	The symbols total position quantity	10000
➔	Commission	Y	double	Commission	0
➔	secType	Y	String	security type option,Default is FOR	0

				FOR =Forex. FORWARD =Forward	
➔	openRefPx	Y	string	the open conversion rate to USD	1
➔	type	Y	string	order type, 1=Market, 2=Limit, 3=Stop, U=Threshold	
➔	clsAvgPx	N	double	the close average price	
➔	acct	Y	string	The Trade Account	USER1
TicketAry		Y			
➔	side	Y	string	B=buy, S=sell	S
➔	method	Y	string	refer to name space Method	PUT
➔	opQty	Y	double	remaining open quantity	10000
➔	clsTm	N	string	closing time	
➔	sym	Y	string	FX symbol must be in "CUR1/CUR2" format, for example: USD/JPY.	EUR/USD
➔	clsCmsn	N	double	commission for ticket close	
➔	Cmsn	Y	double	optional, commission for ticket close	0
➔	tktId	Y	string	Ticket id	
➔	OpTm	Y	string	open time	
➔	subId	Y	string	"id"+"subId" could be a unique key to identify a ticket.	
➔	opCmsn	Y	double	commission for ticket open	
➔	clsCnvRt	Y	double	optional, conversion rate for clsPx	0
➔	clsPx	Y	double	optional, closing price	0
➔	opCnvRt	Y	double	conversion rate for opPx	1
➔	clsQty	Y	double	optional, how much has been closed	0
➔	opPx	Y	double	open price	1.14134
➔	acct	Y	String	Trade Account	USER1
➔	status	Y	string	Order status 0=open,	0

				1=partially closed, 2=closed	
Login		Y	string	The Trade Account	USER1

Sample:

{"AcctVal":{"dayCumQty":10000,"depoDraw":0,"numTickets":1,"bodBal":999999.89,"clsPL":-238.94902,"method":"PUT","mntnMrgn":586.428578587,"commission":0,"liqdMrgn":586.428578587,"bal":999760.94098,"login":"USER1","rqdMrgn":586.428578587},"ExecRp":{"txTime":"20181218-01:31:30","sym":"EUR/USD","px":0,"ordUser":"TWS_USER2","ordQty":0,"secType":0,"ordId":"1144","origOrdUser":"","tktId":"1144","clOrdId":"1144","origClOrdId":"","settDate":"20181218","commission":0,"execType":"F","ordStatus":2,"side":2,"lvQty":0,"method":"NEW","lastPx":1.13436,"cumQty":10000,"execId":"TWS_USER2:USER1:1144_15451146901","lastQty":10000,"avgPx":1.13436,"convRate":1,"acct":"USER1","ordType":"1","refTktId":"1144","stopPx":0},"MT":"ExecRp","OrdInfo":{"clOrdId":"1144","method":"DEL","acct":"USER1"},"PositionAry":[{"clsPL":0,"method":"PUT","openAvgPx":1.158254902,"sym":"EUR/USD","qty":41000,"commission":0,"openRefPx":1,"type":1,"acct":"USER1"},{"clsPL":-238.94902,"clsRefPx":1,"method":"PUT","openAvgPx":1.158254902,"sym":"EUR/USD","qty":10000,"commission":0,"openRefPx":1,"type":3,"clsAvgPx":1.13436,"acct":"USER1"}],"TicketAry":[{"side":"B","method":"PUT","opQty":41000,"clsTm":"0000/00/00-00:00:00","sym":"EUR/USD","clsCmsn":0,"tktId":"EUR/USD","opTm":"0000/00/00-00:00:00","subId":"0","opCmsn":0,"clsCnvRt":0,"clsPx":0,"opCnvRt":1,"clsQty":10000,"opPx":1.158254902,"user":"USER1","acct":"USER1","status":0}],"Login":"USER1"}

4) Create Order Cancel Request

Client can create an order cancel request by sending a OrdCxlReq request.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	String	Must be “OrdCxlReq”	OrdCxlReq
TOK		Y	String	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
OrdCxlReq					
➔	acct	Y	String	The Trade account name	USER1
➔	orgUser	Y	String	The cloud server connects to Fortex orig user	TWS_USER2
➔	orgClOrdId	Y	String	Same as order ID	895
➔	type	Y	String	Order type, 1=Market,	2

				2=Limit, 3=Stop, U=Threshold	
➔	side	Y	String	1=BUY, 2=SELL	1
➔	qty	Y	String	quantity	10000
➔	sym	Y	string	FX symbol must be in “CUR1/CUR2” format	EUR/USD
➔	secType	Y	String	security type option: FOR =Forex. FORWARD =Forward	FOR
➔	execDst	Y	String	execution destination The default value is “INTX”	INTX
➔	txTime	Y	string	The trade time	20181023-14:34:52.320

Sample:


```
{
    "MT": "OrdCxlReq",
    "OrdCxlReq": {
        "user": "USER_1001",
        "acct": " USER_1001",
        "orgUser": " USER_1001",
        "orgClOrdId": "2451",
        "type": "2",
        "side": "1",
        "qty": "10000",
        "sym": "EUR/USD",
        "secType": "FOR",
        "execDst": "INTX",
        "txTime": "20210715-15:07:26.250",
        "orgOrdTag": "XC.ORD.USER_1001"
    },
    "Tok": "a99c03ea1cc84f488e90e90c5c59fc12"
}
```


"orgOrdTag" ==open order tag

If everything is correct then the server will respond some messages which are defined as the following. These messages are Enter Execution Report, the Cancel Order Rejected report and Cancel Result Execution Report.

- Enter Execution Report

The Enter message is used for the client and server ACK message.
Please refer the Create new order request. (4.3.2)

- Cancel Order Rejected Report.

If the cancel order is rejected. The server will respond rejected message which is defined as following.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	String	Must be "OrdCxlRej"	OrdCxlRej
OrdCxlRej		Y		Cancel Order info	
➔	cxlRejRespTo	Y	String	Cancel Reject	1
➔	clOrdId	Y	string	The close order ID,same as order id	871
➔	ordStat	Y	String	Order status	0
➔	txTime	Y	String	TransactTime	20181028-18:42:55
➔	orgClOrdId	Y	String	The Org Cancel Order ID	123
➔	OrdId	Y	String	The new order ID	871
➔	acct	Y	String	The Order Trade account	USER1
➔	cxlRejRsn	Y	String	Cancel Reject Reason	52
Login		Y	string	The Trade account	USER2
ErrInfo		Y			
➔	code	Y	string	Code to identify reason	52
➔	desc	Y	string	The err detail message	Order Not Found

Sample:
{"OrdCxlRej":{"cxlRejRespTo":"1","clOrdId":"871","ordStat":0,"txTime":"20181028-18:42:55","orgClOrdId":"123","ordId":"871","acct":"USER1","cxlRejRsn":"52"},"MT":"OrdCxlRej","Login":"USER2","ErrInfo":{"code":52,"desc":"Order Not Found"}}

- Cancel Order Execution Report.

If everything is correct, the cancel order cancels successfully, then the server will respond the message as below.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	String	Must be "ExecRp"	ExecRp
ExecRp		Y	String	Execution Report, include the orders info	
➔	txTime	Y	String	TransactTime	20181028-18:41:17

➔	sym	U	String	FX symbol must be in "CUR1/CUR2" format, for example: USD/JPY.	EUR/USD
➔	px	Y	double	The trade prices. if the type =1, the px should be "0" If the type =U, the px means the Upper Limit price	0
➔	orduser	Y	String	The order user	TWS_USER2
➔	ordQty	N	Double	The amount requested to trade.	0
➔	OrdId	Y	String	Order ID	869
➔	tktId	Y	String	Ticket ID, same as the order id	869
➔	ClOrdId	Y	String	Close order ID	869
➔	ordStat	Y	String	Order Status	4
➔	settDate	N	String	If the secType=FORWARD, need to send the settlement date to server.	""
➔	Commission		Double	Commission	0
➔	secType	Y	string	security type option, Default is FOR FOR =Forex. FORWARD =Forward	""
➔	execType		string	ExecNew = '0', ExecPartialfill = '1', ExecFill = '2', ExecDoneforDay = '3', ExecCanceled = '4', ExecReplace = '5', ExecPendingCancel = '6', ExecStopped = '7', ExecRejected = '8', ExecSuspended = '9', ExecPendingNew = 'A', ExecCalculated = 'B', ExecExpired = 'C', ExecRestated = 'D', ExecPendingReplace = 'E', ExecTrade = 'F', ExecTradeCorrect = 'G',	8

				ExecTradeCancel = 'H', ExecOrderStatus = 'I', ExecAllocation = 'J', ExecDistribution = 'K',	
➔	leavQty	Y	Double	The amount remaining – always 0 for a fill.	10000
	orgUser	Y	String	Fortex order orig User	“”
➔	side	Y	Int	Trade side 1=BUY 2=SELL	
➔	method	Y	String	The server processing method: PUT = Update NEW = New DEL = Delete	NEW
➔	lastPx	Y	Double	The price for this execution.	0
➔	cumQty	Y	Double	The amount that’s been done, equal to the quantity on a fill.	0
➔	orgClordId	Y	String	Same as order ID	“”
➔	execId	Y	String	Trade server execute ID	1540777277
➔	lastQty	Y	Double	Amount traded for this execution	0
➔	avgPx	Y	Double	Rate on the trade.	0
➔	convRate	Y	Double	Conversion Rate	0
➔	acct	Y	string	Trade Account	USER1
➔	ordType	Y	String	order type, 1 = Market, 2 = Limit, 3 = Stop U = Threshold	2
➔	refTktID	Y	String	Reference ticket ID,	“”
➔	stopPx	Y	string	Stop price, when eh type=3, then the order use this price to trade	0
OrdInfo					
➔	clOrdID	Y	string	Same as Order ID	869
➔	method	Y	String	The server processing method: PUT = Update	DEL

				NEW = New DEL = Delete	
➔	acct	Y	String	The Trade account	USER1
Login		Y	String	The Trade account	USER1

5) Create Order Cancel and Replace Request

Client can create an order cancel and replace request by sending a OrdCxlRepReq request.

JSON Key		Required	Value Type	Comment	Value Example
MT		Y	string	Must be "OrdCxlRepReq"	OrdCxlRepReq
TOK		Y	String	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
OrdCxlRepReq					
➔	orgUser	Y	String	The cloud server connects to Fortex orig user	TWS_USER2
➔	user	Y	String	The Order User	TWS_USER2
➔	acct	Y	String	The trade account	USER1
➔	orgClOrdId	Y	string	Same as order ID	895
➔	type	Y	string	Order type, 1=Market, 2=Limit, 3=Stop, U=Threshold	1
➔	side	Y	String	1=BUY 2=SELL	1
➔	qty	Y	Double	quantity	10000
➔	sym	Y	String	The FX symbol	0
➔	secType	Y	String	security type option: FOR =Forex. FORWARD =Forward	FOR
➔	cumQty	Y	Double	The amount that's been done, equal to the quantity on a fill.	""

➔	px	Y	Double	The trade prices. if the type =1, the px should be "0" If the type =U, the px means the Upper Limit price	1.234
➔	stoppx	Y	double	Stop price, when eh type=3, then the order uses this price to trade	0
➔	px2	Y	Double	price 2, when the type=U, the px2 mean the Lower Limit price	0
➔	tif	Y	String	Time in Force 1=GTC, 3=IOC, 4=FOK	1
➔	execInst	Y	String	ExecInst, "G" = All or none – AON, "u" or empty = means allowing partial fill.	G
➔	handInst	Y	String	HandlInst	1
➔	execDst	Y	String	execution destination. The default value is "INTX"	INTX
➔	txTime	Y	String	The trade time	20181025-2:25:35.722
➔	tktType	N	String	The Ticket Type: 0=open 1=close -1=not available	""
➔	tktNo	N	String	the ticket#ID	""
➔	refTktNo	N	String	when ticket type is close, then it's the ref ticket id.	""
➔	sl	N	Double	Stop Loss	0
➔	tp	N	Double	Take Profit	0

Sample:


```
{
   "MT": "OrdCxlRepReq",
   "OrdCxlRepReq": {
      "orgUser": "TWS_DEMO2",
      "user": "TWS_DEMO2",
      "acct": " USER_1001",
      "orgClOrdId": "2637",
```

"type": "2",
"side": "1",
"qty": "2",
"sym": "EUR/USD",
"secType": "FOR",
"cumQty": "FOR",
"px": "0.02",
"stopPx": "0",
"px2": "0",
"tif": "1",
"execInst": "G",
"handInst": "1",
"execDst": "INTX",
"txTime": "20210719-4:43:43.479",
"tktType": "",
"tktNo": "",
"refTktNo": "",
"sl": "",
"tp": ""
},
"Tok": "d1e09bb703074fe29c5db75426dace81"
}

If everything is correct then the server will respond some messages which are defined as the following. These messages are Enter Execution Report and Replace Result Execution Report.

- Enter Execution Report

The Enter message is used for the client and server ACK message.
Please refer the Create new order request. (4.3.2)

- Replace Result Execution Report.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	String	Must be “ExecRp”	ExecRp
ExecRp	Y	String	Please refer the cancel Result exectuiton report.	
OrdInfo	Y	String	Please refer the enter execution report.	
Login	Y	String	The Trade account	USER1

5. News

1) News Request

Client can query the news information via sending a SubsNews request.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	String	Must be “SubsNews”	SubsNews
TOK	Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
Subscribe	Y	string	True = subscribe False = unsubscribe	true

Sample:
{"MT":"SubsNews","Subscribe":true,"Tok":"525197be1b9c44c98a2aef5d39f66ce9"}

If everything is correct then the server will respond a message which is defined as the following,

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be “News”	News
KeyWdL	Y	string	The related currency keyword	"USD/JPY","JPY","Japan"
Src	Y	string	This source of this news	
Id	Y	String	The identity of this message	“119841”
HdLn	Y	String	The news header line	
Body	Y	String	The news details, could be HTML	
DateTime	Y	string	When the message is published	20181029-00:52:00

Sample:
{"KeyWdL":["USD/JPY","JPY","Japan"],"Src":"FxWire","MT":"News","Id":"1119841","HdLn":"TOKYO_S NIKKEI SHARE AVERAGE OPENS UP 0.66 PCT AT 21,323.61","Body":"TOKYO_S NIKKEI SHARE AVERAGE OPENS UP 0.66 PCT AT 21,323.61","DateTime":"20181029-00:52:00"}

2) Calendar Request

Client can query the calendar information by sending a SubsCalendar request.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be “SubsCalendar”	SubsCalendar
TOK	Y	string	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
Subscribe	Y	string	True = subscribe False = unsubscribe	true

Sample:
{"MT":"SubsCalendar”,"Subscribe":true,"Tok":"525197be1b9c44c98a2aef5d39f66ce9"}

If everything is correct then the server will respond a message which is defined as the following,

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	string	Must be” Calendar”	Calendar
EvtCata	Y	string	The calendar message sort type, only support the economics now.	economics
KeyWdL	Y	string	The related currency keyword	["JPN"],
ActualVal	Y	string	The actual published value	
Src	Y	string	This source of this calendar	“Benzinga”
PreVal	Y	string	The previous value	“1016900000000.000”
EvtDateTime	Y	string	When the event will happen. Format is YYYMMDD-HH:MM:SS	"20181024-23:50:00"
ForecastVal	Y	String	Forecast Value	
Importance	Y	String	0-N, the bigger the more important	“2”
UpdtDtTm	Y	String	When the calendar is updated.Format is YYYMMDD-HH:MM:SS	"20181024-08:47:46"

ProdFrom	Y	String	The Calendar from this time.	"2018-"
desc	Y	String	The event description /Detail	
Id	Y	String	The identity of this message	"5bcafd974309aa03ffde7b52"
Evtname	Y	string	The event Title	"Foreign Bonds Buying"

Sample:
{"EvtCata":"economics","KeyWdL":["JPN"],"ActualVal":"","Src":"Benzinga","PreVal":"1016900000000.000","MT":"Calendar","EvtDateTime":"20181024-23:50:00","ForecastVal":"","Importance":"2","UpdtDtTm":"20181024-08:47:46","PrdFrom":"2018-","Desc":"","Id":"5bcafd974309aa03ffde7b52","EvtName":"Foreign Bonds Buying"}

6. Others

1) Save Profile

Client can save the client profile by sending a SaveProfile request.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	String	Must be “SaveProfile”	SaveProfile
TOK	Y	String	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
Profile	Y	string	The profile setting that need to save	"keyInt":123,"keyStr":"abc"

Sample:
{"MT":"SaveProfile","Profile":{"keyInt":123,"keyStr":"abc"},"Tok":"525197be1b9c44c98a2aef5d39f66ce9"}

If everything is correct, the client profile will be saved on the server side. Then when client logout/logon, the page will load the last saved profile.

2) Query Forward Standard Tenors

Client can query the forward standard tenors information by sending a

SubsFwdStdTenors request.

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	String	Must be "SubsFwdStdTenors"	SubsFwdStdTenors
TOK	Y	String	The token from the RESTful login response message	525197be1b9c44c98a2aef5d39f66ce9
Subscribe	Y	string	True = subscribe False = unsubscribe	true

Sample:

{"MT":"SubsFwdStdTenors","Subscribe":true,"Tok":"525197be1b9c44c98a2aef5d39f66ce9"}

If everything is correct then the server will respond a message which is defined as the following,

JSON Key	Required	Value Type	Comment	Value Example
MT	Y	String	Must be "FwdStdTenors"	FwdStdTenors
AllsymFwdstdTenors	Y	String	All symbols Forward tenors' info	
FwdStdTenorsdef	Y	string	The forward tenors	

Sample:

{"MT":"FwdStdTenors","AllSymFwdStdTenors":{"AUD/JPY":{"TD":"20181025","3W":"20181119","2W":"20181112","1W":"20181105","6M":"20190430","1Y":"20191031","TM":"20181026","3M":"20190129","2M":"20181228","SN":"20181030","1M":"20181129"},......,"NOK/SEK":{"TD":"20181025","3W":"20181119","2W":"20181112","1W":"20181105","6M":"20190430","1Y":"20191031","TM":"20181026","3M":"20190129","2M":"20181228","SN":"20181030","1M":"20181129"}},"FwdStdTenorsDef":{"TD":"Today","3W":"Spot + 3 Weeks","2W":"Spot + 2 Weeks","1W":"Spot + 1 Week","6M":"Spot + 6 Months","1Y":"Spot + 1 Year","TM":"Tomorrow","3M":"Spot + 3 Months","2M":"Spot + 2 Months","SN":"Today + 3 Days","1M":"Spot + 1 Month"}}