Tradias FIX Pricing API

Introduction

Tradias offers FIX-based APIs for its customers. This document describes the FIX protocol-based Pricing API and is written in the style of the FIX rules of engagement (RoE) document. Therefore, it represents a supplement to the official FIX specification published on the FPL website. Tradias may update this document anytime to reflect changes in our FIX Trading API.

Tradias Trading Model

Tradias is a leading player in the OTC crypto market, setting new standards for liquidity and execution quality. Operating 24/7, Tradias offers exceptional crypto asset liquidity at competitive prices with minimal slippage, which is especially beneficial for large volume trades. As a principal desk, Tradias facilitates direct trades against its OTC desk, eliminating intermediaries and ensuring efficient transactions.

With a diverse range of over 150 crypto assets available for trading, Tradias supports various fiat currencies and stablecoins, offering a comprehensive trading experience tailored to institutional clients. Our robust APIs enables clients to automate trading, manage order flows, and perform seamless reconciliation and settlement. The platform supports over 200 trades per second per client, ensuring rapid execution and outstanding performance.

Pricing at Tradias is highly competitive and confidential, allowing clients to execute their strategies with discretion. To maintain transparency, it provides market conformity checks and detailed trading reports. Operating under the regulations of a certified credit institution in Germany, Tradias ensures minimal counterparty risk, making it a trustworthy partner for institutional trading.

Settlement processes are designed to be flexible and efficient, with options for T+1 and weekly cycles for post-trade settlements. Clients can settle via fiat methods, including SEPA and international wire transfers, or through custodial wallets for crypto transactions.

By providing a fully regulated infrastructure, Tradias opens the door for institutional investors to tap into the vast potential of digital assets, complemented by additional services such as tokenization and consulting.

All orders within the Tradias OTC platform are executed as Fill or Kill (FoK) orders, ensuring that orders are either fully executed or completely rejected, thereby simplifying order management. While limit orders can be placed, only FoK limit orders are supported currently.

FIX Protocol

Session level and connectivity rules

Network Options

The connectivity to the Tradias FIX Pricing API is done over the Internet using TLS. Tradias is the server side and provides the following environments:

- Production
- Pre-Production

Tradias provides a list (usually two) of connection points (Hostname and Port) and a TLS certificate for each environment. To be accepted as a FIX client, the client's IP address must be whitelisted by Tradias. Please contact info-api@tradias.de for questions or to begin the onboarding process.

FIX Versions

The Tradias FIX Pricing API is based on the FIX.4.4 protocol, available at FIX 4.4 Specification.

Encryption Support

Tradias uses a TLS-encrypted TCP connection over the Internet. To establish a connection, Tradias needs to whitelist the client's IP address and provide the client with an SSL certificate. There is no additional encryption in the FIX messages used!

Duplicate and Resend Message Handling

The Tradias FIX Pricing API uses the normal FIX sequence number mechanism to avoid duplicates. To improve performance, the API does not use the message resend mechanism. The related FIX messages are not supported. A sequence number reset occurs each time a client connects to the server.

Start of Day / End of Day Procedures

The Tradias FIX Pricing API acts as a server and expects clients to establish a connection with it. Tradias provides at least two FIX server addresses per FIX connection. In case of a connection problem, the client should try any other addresses to re-establish the connection. The FIX sessions of the Pricing API do not use message recovery. Every time a session is established, a sequence number reset takes place. The client has to redo any subscriptions after the reconnect.

The connection time of a FIX session is 00:00 UTC until 23:59:59 UTC every day (incl. weekends and holidays). That implies a termination of the FIX sessions at midnight UTC.

Counter Party Identification

Tradias supports multiple clients, while the FIX Pricing API supports only one client per FIX session. Therefore, messages from and to different end clients in a single FIX session are prohibited.

The FIX Pricing API identifies the client and Tradias by the fields SenderCompID (49) and TargetCompID (56). Messages sent by the client to the FIX Pricing API need to fill the SenderCompID (49) with the client_id of the client (a UUID provided by Tradias during the onboarding process), and the TargetCompID (56) will be filled with the tradias_id (a UUID provided by Tradias during the onboarding process). For messages sent by Tradias to the client, Tradias fills the SenderCompID (49) with the tradias_id (a UUID provided by Tradias during the onboarding process) and the TargetCompID (56) with the client_id of the client (a UUID provided by Tradias during the onboarding process).

FIX Header and Trailer

The FIX Trading API uses the following FIX Header:

Field	Name	Type	Req	Comment
8	BeginString	String	✓	Must be first field in message (“FIX.4.4”).
9	BodyLength	Length	✓	Must be second field in message.
35	MsgType	String	✓	Must be third field in message.
49	SenderCompID	String	✓	For Messages sent by the client: client_id, otherwise tradias_id. Both are provided by Tradias during the onboarding.
56	TargetCompID	String	✓	For Messages sent by the client: tradias_id, otherwise client_id. Both are provided by Tradias during the onboarding.
34	MsgSeqNum	SeqNum	✓	Integer message sequence number.
52	SendingTime	UTCTimestamp	✓	Time of message transmission, always expressed in UTC.

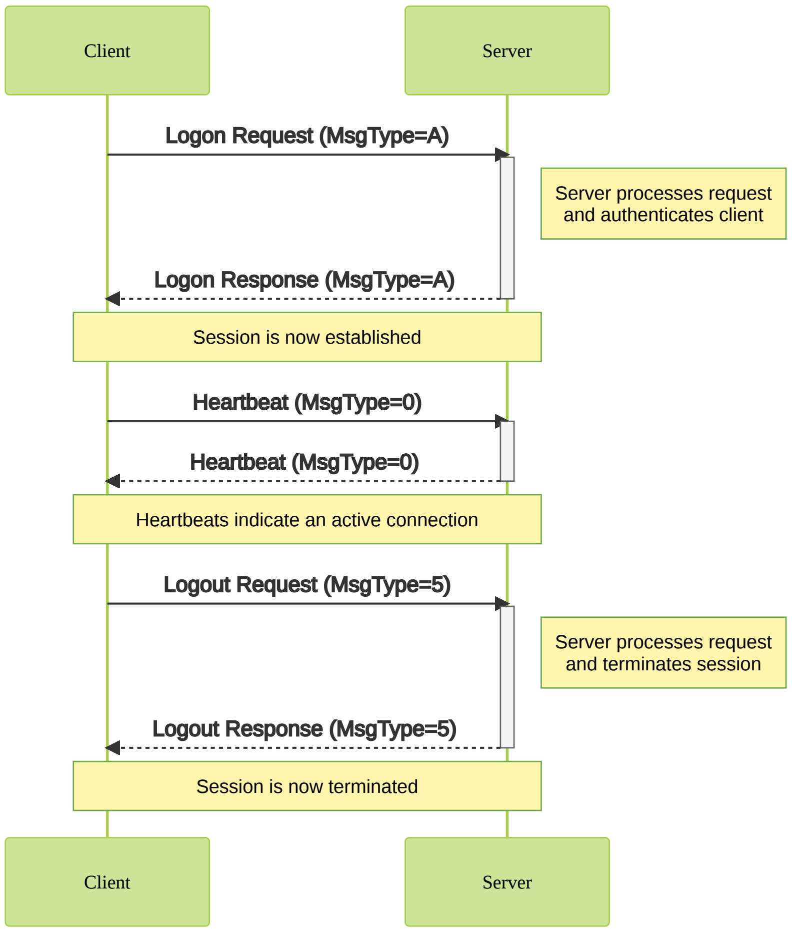
The FIX Trailer of the FIX Trading API is the following:

Field	Name	Type	Req	Comment
10	CheckSum	String	✓	Three-byte, simple checksum. See the FIX spec for details. ALWAYS LAST FIELD IN MESSAGE, i.e. serves, with the trailing <SOH>, as the End-Of-Message delimiter. Always defined as three characters.

FIX Session Level Messages

The FIX Trading API uses the following session-level messages. For further details, see the FIX spec.

The complete logic is depicted in the following diagram:


Client
Server
Logon Request (MsgType=A)
Server processes request and authenticates client
Logon Response (MsgType=A)
Session is now established
Heartbeat (MsgType=0)
Heartbeat (MsgType=0)
Heartbeats indicate an active connection
Logout Request (MsgType=5)
Server processes request and terminates session
Logout Response (MsgType=5)
Session is now terminated
Client
Server

Logon [A]

Field	Name	Type	Req	Comment

	Header	Component	✓	MsgType (35) = A
98	EncryptMethod	int	✓	Method of encryption. Must be 0 = None.
108	HeartBtInt	String	✓	Heartbeat interval (seconds). Note same value used by both sides.
141	ResetSeqNumFlag	Boolean	—	Indicates that both sides of the FIX session should reset sequence numbers.
	Trailer	Component	✓	

Heartbeat [0]

Field	Name	Type	Req	Comment
	Header	Component	✓	MsgType (35) = 0
112	TestReqID	String	—	Required when the heartbeat is the result of a Test Request message.
	Trailer	Component	✓	

Test Request [1]

Field	Name	Type	Req	Comment
	Header	Component	✓	MsgType (35) = 1
112	TestReqID	String	✓	Identifier included in Test Request message to be returned in resulting Heartbeat.
	Trailer	Component	✓	

Reject [3]

Field	Name	Type	Req	Comment
	Header	Component	✓	MsgType (35) = 3
45	RefSeqNum	SeqNum	✓	MsgSeqNum of rejected message.

371	RefTagID	int	–	The tag number of the FIX field referenced.
372	RefMsgType	String	–	The MsgType of the FIX message referenced.
373	SessionRejectReason	int	–	Code to identify reason for a session-level Reject message.
58	Text	String	–	Where possible, message to explain reason for rejection.
	Trailer	Component	✓	

Logout [5]

Field	Name	Type	Req	Comment
	Header	Component	✓	MsgType (35) = 5
58	Text	String	–	Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent.
	Trailer	Component	✓	

Failover Procedures

To make the Tradias FIX service resilient, Tradias has multiple server addresses (Hostname and Port). The client can use any of them to connect. If an address does not allow a connection, the client should try another provided address until one is possible. When the connection is reestablished and the client has completed the Logon, subscriptions need to be redone.

Definition of Security ID Conventions

In the Tradias FIX Pricing API, the crypto pair/token is identified by the tag Symbol (55). The content is the Identifier of the instrument. It consists of the asset codes of the two legs and an optional asset tenor: {leg_1_asset} “-” {leg_2_asset} [“/” {tenor}]

The tenor can be skipped if it is SP (spot).

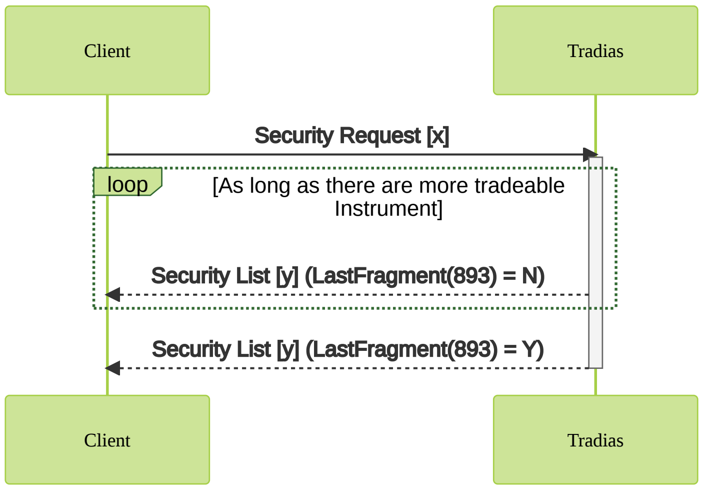
Example (Bitcoin against Euro spot): BTC-EUR/SP or BTC-EUR

Business message types

Static Data

The client can request a list of tradeable instruments with the “Security List Request” [x]. Tradias will answer with a sequence of “Security List” [y] messages containing tradeable instruments. The last “Security List” [y] message will have set LastFragment (893) to “Y”.

The complete logic is depicted in the following diagram:


Client
Tradias
Security Request [x]
loop
[As long as there are more tradeable Instrument]
Security List [y] (LastFragment(893) = N)
Security List [y] (LastFragment(893) = Y)
Client
Tradias

Security List Request [x]

Field	Name	Type	Req	Comment
	Header	Component	✓	MsgType (35) = x
320	SecurityReqID	String	✓	Unique ID of a Security Definition Request.
559	SecurityListRequestType	int	✓	Type of Security List Request being made. Valid values: 4 = All Securities
	Trailer	Component	✓	

Security List [y]

Field	Name	Type	Req	Comment
	Header	Component	✓	MsgType (35) = y

320	SecurityReqID	String	✓	Unique ID of the related Security Definition Request.
322	SecurityResponseID	String	✓	Identifier for the Security List message.
560	SecurityRequestResult		✓	Result of the Security Request identified by the SecurityReqID. Valid values: 0 = valid request 1 = invalid or unsupported request 2 = no instruments found that match selection criteria 3 = not authorized to retrieve instrument data 4 = instrument data temporarily unavailable 5 = request not supported
393	TotNoRelatedS	int	—	Used to indicate if the total number of securities being returned for this request. Used in the event that message fragmentation is required.

893	LastFragment	Boolean	—	Indicates whether this message is the last in a sequence of messages.
				Valid values:
				Y = last message
				N = not last message

146	NoRelatedSym	NumInGroup	—	Specifies the number of repeating symbols specified in this message.
55	Symbol	String	✓	Identifier of the instrument. It consists of the asset codes of the two legs and an optional asset tenor: {leg_1_asset} “-” {leg_2_asset} [“/” {tenor}] Example (Bitcoin against Euro spot): BTC-EUR/SP
969	MinPriceIncrement	float	✓	Minimum price increment for the instrument. FIX 5.0 Field

561	RoundLot	Qty	✓	Minimum size increment for the instrument.
562	MinTradeVol	Qty	—	The minimum order quantity for the instrument.
1140	MaxTradeVol	Qty	—	The maximum order quantity for the instrument. FIX 5.0 SP1 Field
	Trailer	Component	✓	

Pricing

The client can subscribe to price information through the FIX Pricing API for any tradable instrument. The subscription has to be for aggregated Bid and Offer prices. Tradias will provide prices for all agreed-upon sizes.

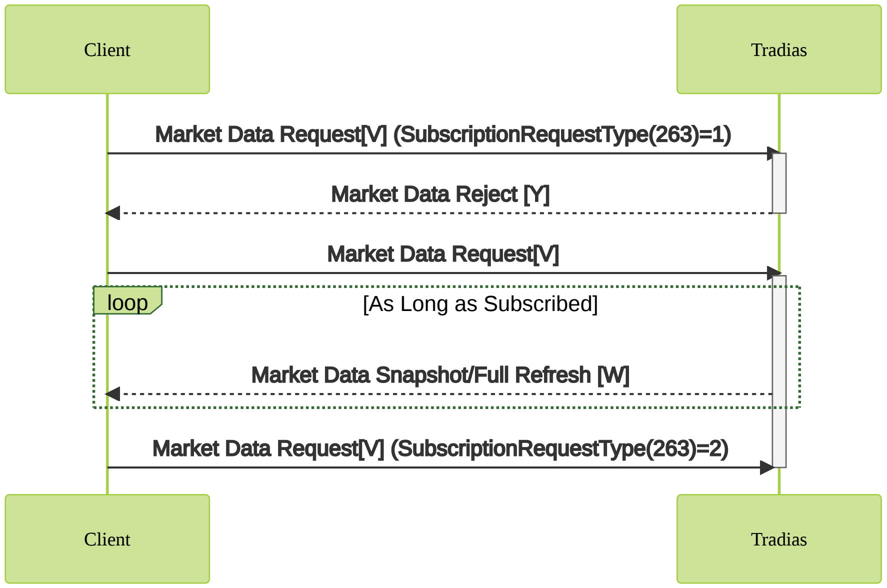
After the subscription, Tradias will send prices until the client unsubscribes from the instrument or the FIX session ends.

Multiple Instruments: The client can subscribe and unsubscribe to multiple instruments in one Market Data Request [V]. A partial unsubscription is is possible. The remaining instruments will still be subscribed and will receive updates!

The prices Tradias provides have an agreed-upon market depth. The updates sent always include bid and offer prices (if available at that moment) and all available price levels.

Price Levels: The price for an instrument is dependent on the side and the quantity. The expected execution price is the price associated with the lowest quantity in the price update message equal or larger than order size (i.e. “the price of the next higher quantity”).

If a subscription (Market Data Request [V]) is not valid or cannot be honoured, a rejection message (Market Data Reject [Y] will be the answer from Tradias. The complete logic is depicted in the following diagram:


Client
Tradias
Market Data Request[V] (SubscriptionRequestType(263)=1)
Market Data Reject [Y]
Market Data Request[V]
loop
[As Long as Subscribed]
Market Data Snapshot/Full Refresh [W]
Market Data Request[V] (SubscriptionRequestType(263)=2)
Client
Tradias

Market Data Request [V]

Field	Name	Type	Req	Comment
	Header	Component	✓	MsgType (35) = V
262	MDReqID	String	✓	Must be unique, or the ID of previous Market Data Request [V] to disable if SubscriptionRequestType (263) = 2 (Disable previous Snapshot + Updates Request).
263	SubscriptionRequestType	char	✓	SubcriptionRequestType indicates to the other party what type of response is expected. A snapshot request only asks for current information. A subscribe request asks for updates as the status changes. Unsubscribe will cancel any future update messages from the counter party. Valid values: 1 = Snapshot + Updates (Subscribe) 2 = Disable previous Snapshot + Update Request (Unsubscribe)
264	MarketDepth	int	✓	Depth of market for Book Snapshot. Valid values: 0 = Full Book

265	MDUpdateType	int	–	Required if SubscriptionRequestType (263) = 1 (Snapshot + Updates). Valid values: 0 = Full Refresh
266	AggregatedBook	Boolean	–	Specifies whether or not book entries should be aggregated. Valid values: Y = one book entry per side per price (Not specified) = Y
267	NoMDEntryTypes	NumInGrou p	✓	Number of MDEntryType (269) fields requested. Has to be 2. As fix Bid and Offer are delivered and need to be requested.
269	MDEntryType	char	✓	This is a element of the list of all the types of Market Data Entries that the client requesting the Market Data is interested in receiving. Valid values: 0 = Bid 1 = Offer Both values 0 and 1 need to be listed in this repeated field, otherwise the Market Data Request [V] will be rejected.
146	NoRelatedSym	NumInGrou p	✓	Specifies the number of repeating symbols specified in this message.
55	Symbol	String	✓	Identifier of the instrument. It consists of the asset codes of the two legs and an optional asset tenor: {leg_1_asset} “-” {leg_2_asset} [“/” {tenor}] Example (Bitcoin against Euro spot): BTC-EUR/SP
815	ApplQueueAction	int	–	Action to take to resolve an application message queue (backlog).

				Valid values: 0 = No action taken 2 = Overlay last If not specified the option is 0 (No action taken) This field is used together with the field ApplQueueMax (812) to define the merging of updates in case of more prices are available from Tradias than the client can process. The option 0 (No action taken) will do nothing an queue the next update to sent it to the client. Option 1 (Overlay last) will overwrite the previous update of the instrument with the new update. That way older updates will get lost but the newer updates are sent sooner. With the field ApplQueueMax (812) the number of updates queued for an instrument that will not be merged can be defined.
812	ApplQueueMax	int	–	Used to specify the maximum number of application messages that can be queued before a corrective action needs to take place to resolve the queuing issue. If not specified the value is 1 If the ApplQueueAction (815) is 0 (No action taken) this field has no impact, as all updates are queued and send. In case of option 1 (Overlay last) this field defines how many updates will be queued before the last one will be overwritten by the new one.
	Trailer	Component	✓	

Market Data Request Reject [Y]

Field	Name	Type	Req	Comment

	Header	Component	✓	MsgType (35) = Y
262	MDReqID	String	✓	Must refer to the MDReqID (262) of the request.
281	MdReqRejReason	char	✓	Reason for the rejection of a Market Data Request [X]. Valid values: 0 = Unknown symbol 1 = Duplicate MDReqID (262) 2 = Insufficient Bandwidth 3 = Insufficient Permissions 4 = Unsupported SubscriptionRequestType (263) 5 = Unsupported MarketDepth (264) 6 = Unsupported MDUpdateType (265) 7 = Unsupported AggregatedBook (266) 8 = Unsupported MDEntryType (269)
58	Text	String	✓	Rejection reason.
	Trailer	Component	✓	

Market Data Snapshot/Full Refresh [W]

Field	Name	Type	Req	Comment
	Header	Component	✓	MsgType (35) = W
262	MDReqID	String	✓	Must refer to the MDReqID (262) of the request.
55	Symbol	String	✓	Identifier of the instrument. It consists of the asset codes of the two legs and an optional asset tenor: {leg_1_asset} “-” {leg_2_asset} [“/” {tenor}] Example (Bitcoin against Euro spot): BTC-EUR/SP

268	NoMDEntries	NumInGroup	✓	Number of entries following.
269	MDEntryType	char	✓	This is a element of the list of all the types of Market Data Entries that the client requesting the Market Data is interested in receiving. Valid values: 0 = Bid 1 = Offer
270	MDEntryPx	Price	✓	Price of the Market Data Entry.
271	MDEntrySize	Qty	✓	Quantity represented by the Market Data Entry.
813	ApplQueueDepth	int	–	Number of updates that were queued at the time that the update was created by Tradias. If not specified the value is 0. A value of 0 means that there has no update been queued.
814	ApplQueueResolution	int	–	Resolution taken when ApplQueueDepth <813> exceeds ApplQueueMax <812> or system specified maximum queue size. Valid values: 0 = No action taken 2 = Overlay last If not specified the resolution is 0 (No action taken)
	Trailer	Component	✓	

Test scripts

To define test cases for the Tradias FIX-based API described in the text, we need to create scenarios that validate the correct behaviour of the system under various conditions. Here are some sample test cases categorized by functionality:

Connectivity and Session Management

1. **Connectivity Test**
   - **Objective**: To ensure the client can connect to the Tradias server over the internet using TLS.
   - **Steps**:
     - Attempt to establish a connection using the provided hostname and port.
     - Validate that the connection is secured using the provided TLS certificate.
   - **Expected Result**: The connection is successfully established using TLS.
2. **Whitelist IP Test**
   - **Objective**: To verify that only whitelisted IP addresses are accepted.
   - **Steps**:
     - Attempt to connect to the server using a non-whitelisted IP address.
   - **Expected Result**: Connection should be refused.
3. **Login Session Test**
   - **Objective**: To confirm successful logon and logoff procedures.
   - **Steps**:
     - Send a Logon (Message Type A) with the correct credentials.
     - Send a Logout (Message Type 5) to terminate the session.
   - **Expected Result**: Login and logout should succeed without errors.

Subscription and Market Data Handling

1. **Subscription Request Test**
   - **Objective**: To ensure that subscription requests for market data updates are correctly processed.
   - **Steps**:
     - Send a Market Data Request (Message Type V) to subscribe to an instrument.
     - Receive Market Data Snapshot/Full Refresh (Message Type W).
   - **Expected Result**: Subscription is acknowledged, and market data updates are received.
2. **Unsubscription Request Test**
   - **Objective**: To confirm the unsubscription process.
   - **Steps**:
     - Send a Market Data Request (Message Type V) to subscribe to an instrument.
     - Receive Market Data Snapshot/Full Refresh (Message Type W).
     - Send a Market Data Request (Message Type V) with unsubscription details.
   - **Expected Result**: No further market data updates are received.

Error Handling

1. **Invalid Symbol Test**
   - **Objective**: To test how the system handles requests with unknown symbols.
   - **Steps**:
     - Send a Market Data Request (Message Type V) with an invalid symbol.
   - **Expected Result**: Receive a Market Data Request Reject (Message Type Y) with a reason code for an unknown symbol.
2. **Sequence Number Mismatch Test**
   - **Objective**: To validate error handling when a sequence number mismatch occurs.
   - **Steps**:
     - Intentionally send a message with an out-of-sequence number.
   - **Expected Result**: Receive a Reject (Message Type 3) message detailing the sequence number mismatch.

Static Data Handling

1. **Instrument List Request Test**
   - **Objective**: To confirm the client can request and receive a list of tradable instruments.
   - **Steps**:
     - Send a Security List Request [x] to request tradable instruments.
     - Receive a series of "Security List" [y] messages containing tradable instruments.
   - **Expected Result**: The final "Security List" [y] message should have LastFragment (893) set to "Y", indicating the end of the list.

Failover and Recovery

1. **Server Failover Test**
   - **Objective**: To verify client failover procedures to secondary server addresses in case of primary server failure.
   - **Steps**:
     - Simulate a primary server failure.
     - Attempt to connect to a secondary server address.
   - **Expected Result**: The client should successfully connect to the secondary server and continue operations.

These test cases cover a range of scenarios that the Tradias FIX-based API may encounter during operation. They are designed to ensure the system is robust, reliable, and follows the FIX protocol standards.

Examples

Logon

Logon Request

In the context of FIX 4.4, a logon message is denoted by the MsgType (35) set to "A". Here's an example of a logon message with randomly generated UUIDs for the SenderCompID and TargetCompID:

```
8=FIX.4.4|9=133|35=A|34=1|49=123e4567-e89b-12d3-a456-426614174000|56=123e4567-e89b-12d3-a456-426614174001|52=20240813-16:20:00.000|98=0|108=30|141=Y|10=124|
```

In the above message:

- `8=FIX.4.4` specifies the FIX version.
- `9=133` is the BodyLength, which must be calculated as the character count of the entire message excluding the start (8) and end (10) tags.
- `35=A` indicates it's a logon message.
- `34=1` is the MsgSeqNum, indicating this is the first message in the sequence.
- `49=123e4567-e89b-12d3-a456-426614174000` is the SenderCompID, a randomly generated UUID for the sender.
- `56=123e4567-e89b-12d3-a456-426614174001` is the TargetCompID, a randomly generated UUID for the receiver.
- `52=20240813-16:20:00.000` is the SendingTime in UTC, formatted as YYYYMMDD-HH:MM:SS.sss.
- `98=0` represents the EncryptMethod, set to 0, indicating no encryption.
- `108=30` is the HeartBtInt, representing the heartbeat interval in seconds.
- `141=Y` indicates that the sequence number should be reset.
- `10=124` is the CheckSum, which should sum all bytes in the message modulo 256.

Please note that the BodyLength (9) and the CheckSum (10) need to be calculated correctly. The vertical bar `|` is used here for clarity but should be replaced with the ASCII character for "Start of Heading" (SOH), `0x01`.

Also, the UUIDs are just examples; in a real-world scenario, Tradias provides those during onboarding.

Logon Response

The logon response is typically a mirrored version of the logon request with some additional fields that may be specific to the session. Below is an example of how a logon response to the above message might look like, with a placeholder for the calculated checksum, assuming the server has accepted the logon request:

> 8=FIX.4.4|9=133|35=A|34=2|49=123e4567-e89b-12d3-a456-426614174001|56=123e4567-e89b-12d3-a456-426614174000|52=20240813-16:21:00.000|98=0|108=30|141=N|10=115|

In this response:

- `8=FIX.4.4` specifies the FIX version.
- `9=133` is the BodyLength, which must be calculated for the response.
- `35=A` indicates it's a logon message.
- `34=2` is the MsgSeqNum, indicating this is the second message in the sequence (the response).
- `49=123e4567-e89b-12d3-a456-426614174001` is the SenderCompID, which now contains the server's UUID, as it is the sender of the response.
- `56=123e4567-e89b-12d3-a456-426614174000` is the TargetCompID, which now contains the client's UUID, as it's the intended recipient.
- `52=20240813-16:21:00.000` is the SendingTime in UTC, formatted as YYYYMMDD-HH:MM:SS.sss, indicating the time the server sends the response.
- `98=0` represents the EncryptMethod, set to 0, indicating no encryption.
- `108=30` is the HeartBtInt, representing the heartbeat interval in seconds agreed upon.
- `141=N` indicates that the sequence number should not be reset, as this is a continuous session.
- `10=115` is the CheckSum, which should be calculated based on the content of the message.

Like with the logon request, the BodyLength (9) and the CheckSum (10) fields must be calculated accurately to validate the message. The vertical bar `|` is used here for readability but should be replaced with the SOH character in a real FIX message. The sequence number and the timestamp should be adjusted to match the actual values at the response time.

Static data request (Instrument List)

Request

Here is a sample FIX message requesting an instrument list and the corresponding response based on FIX 4.4 protocol specifications.

```
8=FIX.4.4|9=136|35=x|34=3|49=123e4567-e89b-12d3-a456-
426614174000|56=123e4567-e89b-12d3-a456-426614174001|52=20240813-
16:40:00.000|320=REQID67890|559=4|10=223|
```

In this request:

- `35=x` indicates it's a Security List Request.
- `320=REQID67890` is the SecurityReqID, a unique identifier for the security list request.
- `559=4` is the SecurityListRequestType, with the value `4` indicating a request for all securities.
- `9=136` and `10=223` are the BodyLength and CheckSum, which need to be calculated.

Response

```
8=FIX.4.4|9=294|35=y|34=4|49=123e4567-e89b-12d3-a456-
426614174001|56=123e4567-e89b-12d3-a456-426614174000|52=20240813-
16:40:05.000|320=REQID67890|322=RESPID12345|560=0|393=2|893=Y|146=2|55=BTC-
EUR|969=0.000000001|561=0.000001|562=0.000001|1140=20|55=ETH-
EUR|969=0.000000001|561=0.000001|562=0.000001|1140=2000|10=135|
```

In this response:

- `35=y` indicates it's a Security List message.
- `320=REQID67890` corresponds to the original SecurityReqID of the request.
- `322=RESPID12345` is the SecurityResponseID, a unique identifier for the security list message.
- `560=0` is the SecurityRequestResult, indicating a valid request.
- `393=2` is the TotNoRelatedSym, indicating the total number of securities returned.
- `893=Y` is the LastFragment, indicating this is the last message in a sequence.
- `146=2` is the NoRelatedSym, specifying the number of symbols in the message.
- `55=BTC-EUR` and `55=ETH-EUR` are symbols for each security.
- `969=0.0001` and `969=0.01` are MinPriceIncrement for each symbol.

- `562=100000` is the MinTradeVol for each symbol.
- `9=294` and `10=135` are the BodyLength and CheckSum, which must be calculated.

Please note that in these messages:

- `|` represents the ASCII character for "Start of Header" (SOH), which is `0x01`.
- The BodyLength (`9`) value should be the character count of the message excluding the `8`, `9`, and `10` tags.
- The CheckSum (`10`) value is the sum of all ASCII values of the characters in the message modulo 256, which should be a three-digit number.

Subscribe to Prices with Updates

This example is a FIX message for an instrument subscription request and the corresponding subscription response. These messages are based on the FIX 4.4 protocol.

Subscription Request

```
8=FIX.4.4|9=170|35=V|34=5|49=123e4567-e89b-12d3-a456-
426614174000|56=123e4567-e89b-12d3-a456-426614174001|52=20240813-
16:30:00.000|262=REQID4711|263=1|264=0|265=0|266=Y|146=1|55=BTC-EUR|10=099|
```

In this request:

- `35=V` indicates it's a Market Data Request.
- `262=REQID4711` is the MDReqID, a unique identifier for the market data request.
- `263=1` is the SubscriptionRequestType, indicating a request for a snapshot plus updates (subscribe).
- `264=0` is the MarketDepth, indicating a full book is requested.
- `265=0` is the MDUpdateType, indicating that full refreshes are required.
- `266=Y` is the AggregatedBook, indicating book entries should be aggregated.
- `146=1` is the NoRelatedSym, indicating one symbol is specified in the message.
- `55=BTC-EUR` is the Symbol specifying the instrument to subscribe to.
- `9=170` and `10=099` are the BodyLength and CheckSum, respectively, which must be calculated.

Subscription Response

```
8=FIX.4.4|9=204|35=W|34=6|49=123e4567-e89b-12d3-a456-426614174001|56=123e4567-e89b-12d3-a456-426614174000|52=20240813-16:30:05.000|262=REQID4711|55=BTC-EUR|268=2|269=0|270=54123.349563|271=5|269=1|270=54193.462953|271=5|10=221|
```

In this response:

- `35=W` indicates it's a Market Data Snapshot/Full Refresh.
- `262=REQID4711` refers to the original MDReqID of the request.
- `55=BTC-EUR` is the Symbol for which the data is being provided.
- `268=2` is the NoMDEntries, indicating the number of entries in the message.
- `269=0` and `269=1` are MDEntryType fields for bid and offer, respectively.
- `270=54123.349563` and `270=54193.462953` are MDEntryPx, representing the bid and offer prices.
- `271=5` is the MDEntrySize, representing the size for both the bid and the offer.
- `9=204` and `10=221` are the BodyLength and CheckSum, respectively, which must be calculated.

Glossary

- **OTC (Over The Counter)**: Trading directly between two parties without a centralized exchange. This method is suitable for larger trades that could impact market prices if executed publicly.
- **FIX (Financial Information Exchange)**: A communication protocol for the real-time exchange of information related to securities transactions and markets.
- **API (Application Programming Interface)**: A set of protocols and tools for building software applications, which, in this context, allows for interaction with the Tradias trading system.
- **RoE (Rules of Engagement)**: Guidelines or rules outlining how to interact with a particular FIX API.
- **FPL (FIX Protocol Limited)**: The organization responsible for maintaining and developing the FIX protocol.
- **TLS (Transport Layer Security)**: A cryptographic protocol designed to provide communications security over a computer network.
- **Production Environment**: The live environment where the actual trading happens.

- **Pre-Production Environment**: A testing environment that simulates the production environment.
- **Whitelisting**: The process of allowing certain IP addresses to access a network or service, ensuring security.
- **FIX.4.4 Protocol**: A version of the FIX protocol that defines a specific set of message formats and rules for engaging in electronic financial transactions.
- **SSL Certificate**: A digital certificate that provides authentication for a website and enables an encrypted connection.
- **Session**: The period during which a user is logged on to a system.
- **Sequence Number**: A unique identifier for messages in a sequence that helps prevent message duplication.
- **SenderCompID (Tag 49)**: A FIX message field identifying the message's sender.
- **TargetCompID (Tag 56)**: A FIX message field that identifies the message's intended recipient.
- **UUID (Universally Unique Identifier)**: A 128-bit number used to identify information in computer systems.
- **BeginString (Tag 8)**: A FIX message field that indicates the beginning of a message.
- **BodyLength (Tag 9)**: A FIX message field that represents the length of a message body.
- **MsgType (Tag 35)**: A FIX message field that defines the message type.
- **MsgSeqNum (Tag 34)**: A FIX message field that specifies the message sequence number.
- **SendingTime (Tag 52)**: A FIX message field that indicates the time a message was sent.
- **CheckSum (Tag 10)**: A FIX message field that provides a simple checksum to validate the integrity of message data.
- **Logon (Message Type A)**: A FIX session-level message used when a user is logging on to the system.
- **Heartbeat (Message Type 0)**: A FIX session-level message that confirms the connection is alive.
- **Test Request (Message Type 1)**: A FIX session-level message used to test the connection status.
- **Reject (Message Type 3)**: A FIX session-level message used to reject a message that does not comply with the protocol.
- **Logout (Message Type 5)**: A FIX session-level message used to terminate a session.
- **Symbol (Tag 55)**: A FIX message field representing the instrument's identifier.
- **Market Data Request (Message Type V)**: A FIX message type used to request market data.

- **Market Data Snapshot/Full Refresh (Message Type W)**: A FIX message used to provide a full snapshot of market data.
- **Market Data Request Reject (Message Type Y)**: A FIX message used to reject a market data request.