# Guide to learn FIX

Learning FIX (Financial Information Exchange) protocol involves understanding a few core concepts and getting hands-on practice.

### 1. ****Understand the Basics****

* **What is FIX?**: FIX is a messaging standard developed for real-time electronic trading of [financial instruments](#_Financial instruments). It’s used to facilitate communication between trading systems and financial institutions.
* **Key Components**:
  + **[Messages](#_Fix messages)**: FIX messages are structured in a specific format and are used to transmit various types of financial information.
  + **[Fields](#_Fix message Fields)**: Each message contains fields, which represent different data elements. Fields are identified by unique tags.
  + **[Sessions](#_FIX sessions)**: FIX operates over sessions that define the rules for communication between trading partners.

### 2. ****Study FIX Specifications****

* **FIX Documentation**: Review the FIX Protocol specification documents. The FIX Trading Community (FIXTC) website provides access to the official FIX specifications.
* **FIX Versions**: Familiarize yourself with different versions of FIX, such as FIX4.2, FIX4.4, and FIX5.0. The latest version will often have more features and updated standards.

### 3. ****Learn the Message Structure****

* **Message Types**: Understand various FIX message types (e.g., NewOrder, ExecutionReport, OrderCancelRequest) and their purposes.
* **Fields and Tags**: Study common fields (e.g., OrderID, Price, Quantity) and their roles within different message types.

### 4. ****Get Hands-On Experience****

* **Simulators and Tools**: Use FIX simulators or test environments to practice sending and receiving FIX messages. Tools like FIXimate or FIX Simulator can be helpful.
* **APIs and Libraries**: Explore FIX libraries and APIs in languages such as Java, C++, or Python. Libraries like QuickFIX or FIX4J can be useful for implementing FIX in your applications.

### 5. ****Build Practical Skills****

* **Develop Sample Applications**: Implement simple trading applications that use FIX messages to get practical experience.
* **Debugging and Monitoring**: Learn to debug and monitor FIX messages to understand how data flows and identify issues.

### 6. ****Join the Community****

* **Forums and Groups**: Engage with online forums or communities focused on FIX. Sites like Stack Overflow or the FIX Trading Community forum can provide valuable insights and support.
* **Training and Certification**: Consider formal training courses or certifications related to FIX protocol to deepen your knowledge and validate your skills.

### 7. ****Stay Updated****

* **Industry News**: Keep up with updates and changes in the FIX protocol by following industry news and updates from FIX Trading Community.

By following these steps, you’ll develop a solid understanding of FIX protocol and gain practical experience in working with it.

# Explanations

## Financial instruments

Financial instruments are assets that can be traded or used for financial transactions. They represent a legal agreement or contract and can be either physical or electronic. Financial instruments fall into several categories, each serving different purposes in the financial markets. Here’s an overview of the main types:

### 1. ****Equities****

* **Stocks**: Shares of ownership in a corporation. They represent a claim on the company's assets and earnings.
* **Common Stock**: Provides voting rights and potential dividends, but in the event of liquidation, it is paid after debt and preferred stockholders.
* **Preferred Stock**: Generally offers no voting rights but provides a fixed dividend and has priority over common stock in the event of liquidation.

### 2. ****Debt Instruments****

* **Bonds**: Long-term debt securities issued by corporations, governments, or other entities, paying periodic interest and returning the principal amount at maturity.
* **Notes**: Shorter-term debt instruments than bonds, with similar characteristics but typically lower interest rates.
* **Commercial Paper**: Unsecured, short-term debt issued by corporations to meet immediate financial needs.

### 3. ****Derivatives****

* **Options**: Contracts that give the holder the right, but not the obligation, to buy or sell an underlying asset at a specified price within a certain period.
* **Futures**: Contracts obligating the buyer to purchase, and the seller to sell, an asset at a predetermined future date and price.
* **Swaps**: Contracts where two parties exchange cash flows or other financial instruments, such as interest rate swaps or currency swaps.

### 4. ****Currencies****

* **Foreign Exchange (Forex)**: Trading in different currencies. The exchange rates fluctuate based on economic conditions, geopolitical events, and market sentiment.

### 5. ****Commodities****

* **Physical Commodities**: Basic goods such as oil, gold, agricultural products, etc., that are traded on commodity markets.
* **Commodity Futures**: Contracts to buy or sell a commodity at a future date and price.

### 6. ****Investment Funds****

* **Mutual Funds**: Investment vehicles pooling funds from many investors to invest in a diversified portfolio of stocks, bonds, or other securities.
* **Exchange-Traded Funds (ETFs)**: Similar to mutual funds but traded on stock exchanges like individual stocks.

### 7. ****Structured Products****

* **Collateralized Debt Obligations (CDOs)**: Complex financial instruments backed by a pool of assets, such as loans or mortgages.
* **Mortgage-Backed Securities (MBS)**: Securities backed by a collection of mortgages.

### 8. ****Hybrid Instruments****

* **Convertible Bonds**: Bonds that can be converted into a predetermined number of the issuing company’s equity shares.
* **Warrants**: Long-term options issued by companies, allowing investors to buy shares at a set price before expiration.

### Characteristics of Financial Instruments:

* **Liquidity**: The ease with which an instrument can be bought or sold without affecting its price.
* **Risk**: The potential for loss or gain associated with the instrument.
* **Return**: The income or profit generated from the instrument, such as interest, dividends, or capital gains.
* **Market**: The venue or system where the instrument is traded, such as a stock exchange or over-the-counter (OTC) market.

Understanding financial instruments is crucial for anyone involved in finance or investing, as they play a central role in financial markets and transactions.

## Fix messages

### 1. ****Message Structure****

Each FIX message consists of the following main elements:

* **Message Header**: Contains metadata about the message, such as the message type, sender, and recipient.
* **Message Body**: Contains the actual data or content of the message.
* **Message Trailer**: Includes information for message integrity and validation.

### 2. ****Message Types****

FIX messages are categorized into different types, each serving a specific purpose. Here are some common message types:

#### ****Order Management****

* **NewOrder - Single (D)**: Used to place a new order. Includes details such as order type, quantity, and price.
* **OrderCancelRequest (F)**: Request to cancel an existing order. Specifies the original order to be canceled.
* **OrderCancelReplaceRequest (G)**: Request to modify an existing order. It includes details for canceling the original order and placing a new one.

#### ****Execution Reports****

* **ExecutionReport (8)**: Provides information about the status of an order. This includes execution details, status updates, and any changes to the order.

#### ****Market Data****

* **MarketDataRequest (V)**: Request for market data. Specifies the type of data needed, such as bids, offers, or trades.
* **MarketDataSnapshotFullRefresh (W)**: Provides a complete snapshot of market data for a specific instrument.
* **MarketDataIncrementalRefresh (X)**: Provides incremental updates to market data.

#### ****Session Management****

* **Logon (A)**: Initiates a session between trading systems. Includes authentication and session parameters.
* **Logout (5)**: Ends a session. Used for terminating the connection between trading systems.
* **Heartbeat (0)**: Regularly sent to ensure that the session is active and functioning properly.

#### ****Trade Confirmation****

* **TradeCaptureReport (AE)**: Reports the details of a trade. Includes information about the parties involved, trade date, and instrument details.

### 3. ****Message Fields****

Each FIX message contains fields that convey specific pieces of information. Fields are identified by unique tags and follow a key-value format. Examples of commonly used fields include:

* **OrderID (11)**: Unique identifier for the order.
* **Price (44)**: Price of the instrument.
* **Quantity (38)**: Number of units or shares.
* **Symbol (55)**: Identifier for the financial instrument.

### 4. ****Message Flow****

FIX messages follow a sequence of interactions between parties:

1. **Initiation**: One party sends a Logon message to initiate a session.
2. **Order Placement**: The trader sends a NewOrder - Single message to place an order.
3. **Order Updates**: The system responds with an ExecutionReport to provide updates on the order status.
4. **Market Data**: Requests and responses for market data are handled using messages like MarketDataRequest and MarketDataSnapshotFullRefresh.
5. **Session Management**: Regular Heartbeat messages are exchanged to keep the session active. Logout is used to end the session.

### 5. ****Message Validation****

FIX messages are validated using several techniques:

* **Checksum**: A checksum value at the end of each message ensures data integrity. The checksum is calculated from the message content.
* **Sequence Numbers**: Each message includes a sequence number to track the order of messages and detect any missing or out-of-order messages.

### 6. ****Implementation****

To work with FIX messages:

* **Libraries and Tools**: Utilize FIX libraries such as QuickFIX or FIX4J to facilitate the creation and handling of FIX messages.
* **Simulators**: Use FIX simulators for testing and debugging your FIX implementation.
* **Documentation**: Refer to FIX Protocol specifications for detailed information on message formats and field definitions.

Understanding and implementing FIX messages require familiarity with the FIX protocol and practice in constructing and interpreting these standardized messages.

## Fix message Fields

FIX (Financial Information eXchange) messages are a standardized way of communicating trade-related information between financial institutions. The FIX protocol is widely used for real-time electronic trading and order routing.

Each FIX message is made up of fields that convey specific pieces of information. These fields are identified by unique tags and can vary depending on the type of message. Here's a general breakdown of how FIX message fields work:

### Structure of FIX Messages

**Message Header**:

* 1. Contains metadata about the message, such as its type, sender, and receiver.
  2. Key fields include:
     1. **35** (Message Type): Identifies the type of message (e.g., New Order - Single, Execution Report).
     2. **34** (Message Sequence Number): A unique sequence number for the message.
     3. **49** (SenderCompID): Identifier of the sender.
     4. **56** (TargetCompID): Identifier of the recipient.
     5. **52** (SendingTime): The time the message was sent.

**Message Body**:

* 1. Contains the main content of the message, which varies by message type.
  2. Example fields include:
     1. **11** (ClOrdID): Client order ID, a unique identifier for the order from the client.
     2. **54** (Side): Specifies the side of the order (e.g., Buy or Sell).
     3. **55** (Symbol): The symbol of the instrument being traded.
     4. **44** (Price): The price at which the order is to be executed.
     5. **38** (OrderQty): The quantity of the order.

**Message Trailer**:

* 1. Contains fields related to the end of the message, such as checksums.
  2. Key fields include:
     1. **10** (CheckSum): A checksum for message integrity.

### Example of a FIX Message

Here’s a simplified example of a FIX message for a New Order - Single:

8=FIX.4.2|9=113|35=D|34=1|49=SenderCompID|56=TargetCompID|52=20240731-12:30:00.000|11=OrderID123|54=1|55=AAPL|38=100|44=150.25|40=2|59=0|10=123|

* **8=FIX.4.2**: Indicates the FIX protocol version.
* **35=D**: Specifies a New Order - Single message.
* **34=1**: Message sequence number.
* **49=SenderCompID**: Sender's ID.
* **56=TargetCompID**: Recipient's ID.
* **52=20240731-12:30:00.000**: Sending time.
* **11=OrderID123**: Client order ID.
* **54=1**: Buy side (1 for buy, 2 for sell).
* **55=AAPL**: Symbol of the security.
* **38=100**: Quantity of the order.
* **44=150.25**: Price of the security.
* **40=2**: Order type (2 for limit).
* **59=0**: Time in force (0 for Day).
* **10=123**: Checksum for message validation.

### Field Types

1. **Required Fields**: Fields that must be included for the message to be valid.
2. **Optional Fields**: Fields that can be included but are not mandatory.
3. **Conditional Fields**: Fields that are included based on the values of other fields.

Each FIX message type has a different set of required, optional, and conditional fields, tailored to the specific needs of the trading or order routing scenario.

The FIX protocol's flexibility and standardization help ensure that different systems and firms can communicate effectively and accurately, reducing the potential for misunderstandings or errors in electronic trading.

## FIX sessions

In the context of the FIX (Financial Information eXchange) protocol, a session refers to a persistent, stateful connection between two FIX-compliant systems (such as trading platforms, brokers, or exchanges) that enables them to exchange FIX messages in a structured and reliable manner.

### Key Concepts of FIX Sessions

**Session Management**:

* 1. FIX sessions manage the state and continuity of communication between two parties. They handle the exchange of messages, track message sequence numbers, and manage reconnections.
  2. A session ensures that messages are delivered in the correct order and that both parties agree on the status and sequence of communication.

**Session Initiation and Termination**:

* 1. **Initiation**: A FIX session begins when one party (the initiator) establishes a connection with another party (the acceptor). This is typically done by sending a logon message.
  2. **Termination**: A session can end either by mutual agreement or by one party sending a logout message. Sessions may also end due to connection issues or protocol violations.

**Session Messages**:

* 1. **Logon**: The initial message sent to establish a session. It includes information like session settings, authentication details, and version numbers.
     1. Tag **35=A** (Logon)
  2. **Logout**: Sent to end a session. It can include a reason for termination.
     1. Tag **35=5** (Logout)
  3. **Heartbeat**: Sent periodically to check if the connection is still active and to ensure that the session is still open.
     1. Tag **35=0** (Heartbeat)
  4. **Resend Request**: Requested when a message is missing or not received correctly, asking for specific messages to be resent.
     1. Tag **35=2** (Resend Request)
  5. **Sequence Reset**: Used to reset the sequence numbers if there is a sequence number mismatch or error.
     1. Tag **35=4** (Sequence Reset)

**Session Parameters**:

* 1. **BeginString**: Identifies the version of the FIX protocol being used (e.g., FIX.4.2, FIX.4.4).
  2. **SenderCompID**: Identifier for the sender of the message.
  3. **TargetCompID**: Identifier for the recipient of the message.
  4. **HeartBtInt**: Interval (in seconds) at which heartbeats are sent to keep the session alive.
  5. **SessionID**: A unique identifier for the session, typically composed of the sender and target component IDs and other parameters.
  6. **MsgSeqNum**: A sequence number assigned to each message to ensure messages are processed in order.

**Session Flow**:

* 1. **Establishing a Session**: The initiator sends a Logon message, and the acceptor responds with a Logon message of its own.
  2. **Message Exchange**: Once the session is established, the parties exchange application-specific messages as needed (e.g., orders, executions).
  3. **Maintaining the Session**: Heartbeats are sent periodically, and any sequence number gaps are managed through Resend Requests and Sequence Resets.
  4. **Terminating the Session**: Either party can send a Logout message to end the session, followed by a final heartbeat to confirm the disconnection.

### Example of a FIX Logon Message

Here’s a simplified example of a FIX Logon message:

8=FIX.4.2|35=A|34=1|49=SenderCompID|56=TargetCompID|52=20240731-12:30:00.000|11=12345|54=1|34=100|49=SenderCompID|56=TargetCompID|60=20240731-12:30:00.000|10=123|

* **8=FIX.4.2**: Protocol version.
* **35=A**: Logon message type.
* **34=1**: Message sequence number.
* **49=SenderCompID**: Sender’s ID.
* **56=TargetCompID**: Recipient’s ID.
* **52=20240731-12:30:00.000**: Sending time.
* **11=12345**: Unique logon ID.
* **54=1**: Indicates the heart-beat interval (in seconds).
* **34=100**: Indicates the sequence number for this session.
* **10=123**: Checksum for message validation.

### Summary

FIX sessions are crucial for maintaining reliable, ordered, and stateful communication in financial trading. They handle message sequencing, reconnections, and session management to ensure that trading systems can communicate effectively and accurately. Understanding and properly managing FIX sessions is essential for implementing and operating FIX-based trading systems.