Live Market Feed

Real-time Market Data across exchanges and segments can now be availed on your system via WebSocket. WebSocket provides an efficient means to receive live market data. WebSocket keeps a persistent connection open, allowing the server to push real-time data to your systems.

All Dhan platforms work on these same market feed WebSocket connections that deliver lightning fast market data to you.

> You can establish upto five WebSocket connections per user with 5000 instruments on each connection.

All request messages over WebSocket are in JSON whereas all response messages over WebSocket are in Binary. You will require WebSocket library in any programming language to be able to use Live Market Feed along with Binary converter.

Using DhanHQ Libraries for WebSockets ›

Establishing Connection

To establish connection with DhanHQ WebSocket for Market Feed, you can to the below endpoint using WebSocket library.

```
wss://api-feed.dhan.co?version=2&token=eyxxxxx&clientId=100xxxxxxx&authType=2
```

Query Parameters

Field	Description
version *required*	`2` for DhanHQ v2
token *required*	Access Token generated via Dhan
clientId *required*	User specific identification generated by Dhan
authType *required*	`2` by Default

Adding Instruments

You can subscribe upto 5000 instruments in a single connection and receive market data packets. For subscribing, this can be done using JSON message which needs to be send over WebSocket connection.

Note

You can only send upto 100 instruments in a single JSON message. You can send multiple messages over a single connection to subscribe to all instruments and receive data.

Request Structure

```
{
  "RequestCode" : 15,
  "InstrumentCount" : 2,
  "InstrumentList" : [
    {
      "ExchangeSegment" : "NSE_EQ",
      "SecurityId" : "1333"
    },
    {
      "ExchangeSegment" : "BSE_EQ",
      "SecurityId" : "532540"
    }
  ]
}
```

Parameters

Field	Type	Description

RequestCode *required*	int	Code for subscribing to particular data mode. Refer to feed request code to subscribe to required data mode
InstrumentCount *required*	int	No. of instruments to subscribe from this request
InstrumentList.ExchangeSegment *required*	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
InstrumentList.SecurityId *required*	string	Exchange standard ID for each scrip. Refer here

Keeping Connection Alive

To keep the WebSocket connection alive and prevent it from closing, the server side uses **Ping-Pong** module. Server side sends ping every 10 seconds to the client server (in this case, your system) to maintain WebSocket status as open.

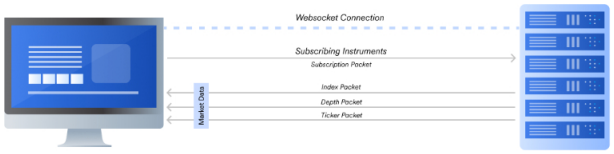
> In case the client server does not respond for more than 40 seconds, the connection is closed from server side and you will have to reestablish connection.

Market Data

The market feed data is sent as structured binary packet which is shared at super fast speed.

DhanHQ Live Market Feed is real-time data and there are three modes in which you can receive the data, depending on your use case:

- Ticker Data
- Quote Data
- Full Data



> **Understanding Binary Message**
> Binary messages consist of sequences of bytes that represent the data. This contrasts with text messages, which use character encoding (e.g., UTF-8) to represent data in a readable format. Binary messages require parsing to extract the relevant information.
>
> The reason for us to choose binary messages over text or JSON is to have compactness, speed and flexibility on data to be shared at lightning fast speed. All responses from Dhan Market Feed consists of Response Header and Payload. Header for every response message remains the same with different feed response code, while the payload can be different.

Response Header

The response header message is of 8 bytes which will remain same as part of all the response messages. The message structure is given as below.

Bytes	Type	Size	Description
`1`	[ ] byte	`1`	Feed Response Code can be referred in Annexure
`2-3`	int16	`2`	Message Length of the entire payload packet
`4`	[ ] byte	`1`	Exchange Segment can be referred in Annexure
`5-8`	int32	`4`	Security ID - can be found here

Ticker Packet

This packet consists of Last Traded Price (LTP) and Last Traded Time (LTT) data across segments.

Bytes	Type	Size	Description
`0-8`	[ ] array	`8`	Response Header with code `2` Refer to enum for values
`9-12`	float32	`4`	Last Traded Price of the subscribed instrument
`13-16`	int32	`4`	Last Trade Time

Prev Close

Whenever any instrument is subscribed for any data packet, we also send this packet which has Previous Day data to make it easier for day on day comparison.

Bytes	Type	Size	Description
0-8	[ ] array	8	Response Header with code 6 Refer to enum for values
9-12	float32	4	Previous day closing price
13-16	int32	4	Open Interest - previous day

Quote Packet

This data packet is for all instruments across segments and exchanges which consists of complete trade data, along with Last Trade Price (LTP) and other information like update time and quantity.

Bytes	Type	Size	Description
0-8	[ ] array	8	Response Header with code 4 Refer to enum for values
9-12	float32	4	Latest Traded Price of the subscribed instrument
13-14	int16	2	Last Traded Quantity
15-18	int32	4	Last Trade Time (LTT)
19-22	float32	4	Average Trade Price (ATP)
23-26	int32	4	Volume
27-30	int32	4	Total Sell Quantity
31-34	int32	4	Total Buy Quantity
35-38	float32	4	Day Open Value
39-42	float32	4	Day Close Value - only sent post market close
43-46	float32	4	Day High Value
47-50	float32	4	Day Low Value

OI Data

Whenever you subscribe to Quote Data, you also receive Open Interest (OI) data packets which is important for Derivative Contracts.

Bytes	Type	Size	Description
0-8	[ ] array	8	Response Header with code 5 Refer to enum for values
9-12	int32	4	Open Interest of the contract

Full Packet

This data packet is for all instruments across segments and exchanges which consists of complete trade data along with Market Depth and OI data in a single packet.

Bytes	Type	Size	Description
0-8	[ ] array	8	Response Header with code 8 Refer to enum for values
9-12	float32	4	Latest Traded Price of the subscribed instrument
13-14	int16	2	Last Traded Quantity
15-18	int32	4	Last Trade Time (LTT)
19-22	float32	4	Average Trade Price (ATP)
23-26	int32	4	Volume
27-30	int32	4	Total Sell Quantity
31-34	int32	4	Total Buy Quantity

35-38	int32	4	Open Interest in the contract (for Derivatives)
39-42	int32	4	Highest Open Interest for the da (only for NSE_FNO)
43-46	int32	4	Lowest Open Interest for the day (only for NSE_FNO)
47-50	float32	4	Day Open Value
51-54	float32	4	Day Close Value - only sent post market close
55-58	float32	4	Day High Value
59-62	float32	4	Day Low Value
63-162	Market Depth Structure	100	5 packets of 20 bytes each for each instrument in below provided structure

Each of these 5 packets will be received in the following packet structure:

Bytes	Type	Size	Description
1-4	int32	4	Bid Quantity
5-8	int32	4	Ask Quantity
9-10	int16	2	No. of Bid Orders
11-12	int16	2	No. of Ask Orders
13-16	float32	4	Bid Price
17-20	float32	4	Ask Price

Feed Disconnect

If you want to disconnect WebSocket, you can send below JSON request message via the connection.

```
{
  "RequestCode" : 12
}
```

In case of WebSocket disconnection from server side, you will receive disconnection packet, which will have disconnection reason code.

- If more than 5 websockets are established, then the first socket will be disconnected with `805` with every additional connection.

Bytes	Type	Size	Description
0-8	[ ] array	8	Response Header with code `50` Refer to enum for values
9-10	int16	2	Disconnection message code - here

You can find detailed Disconnection message code description here.