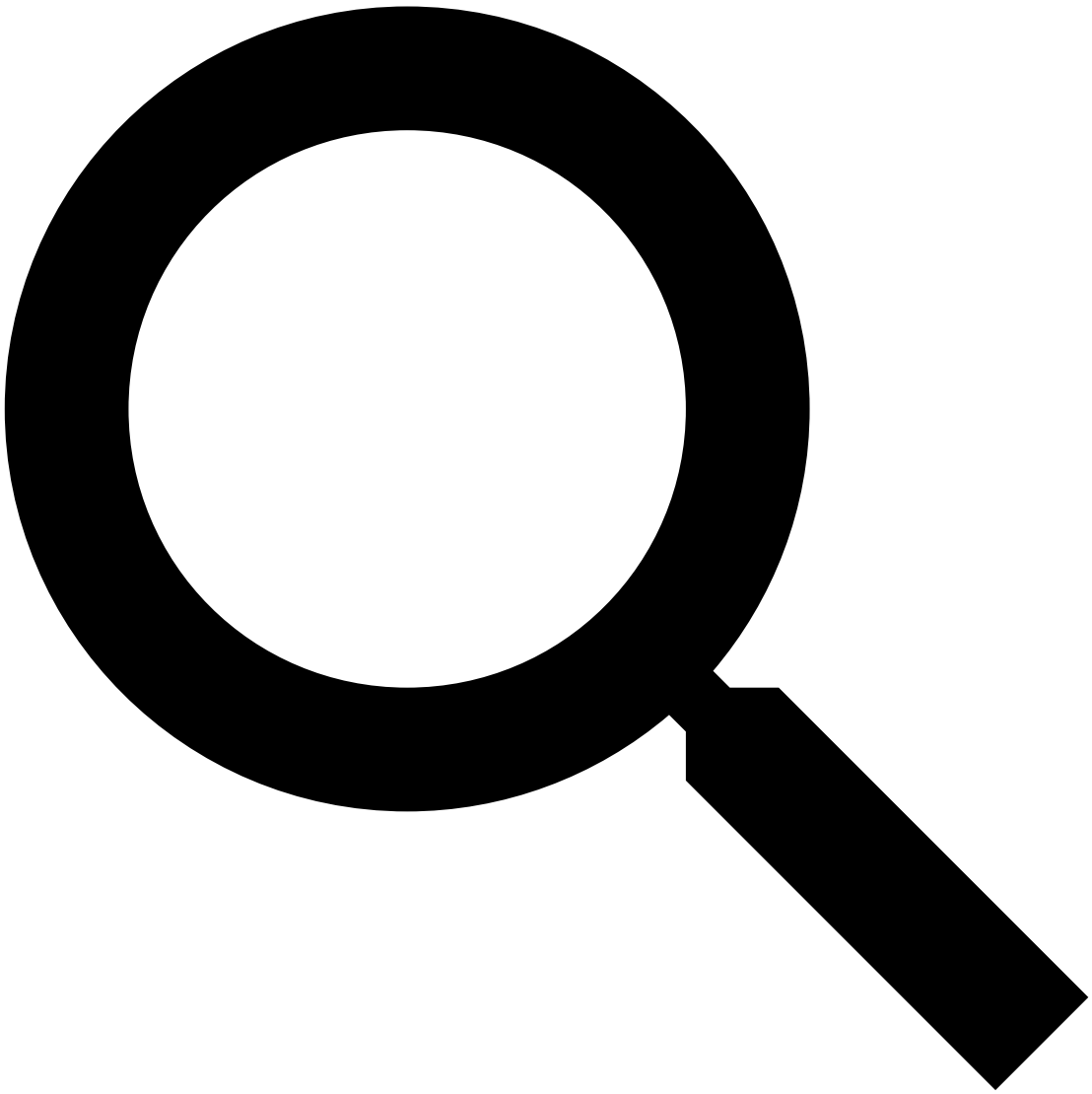
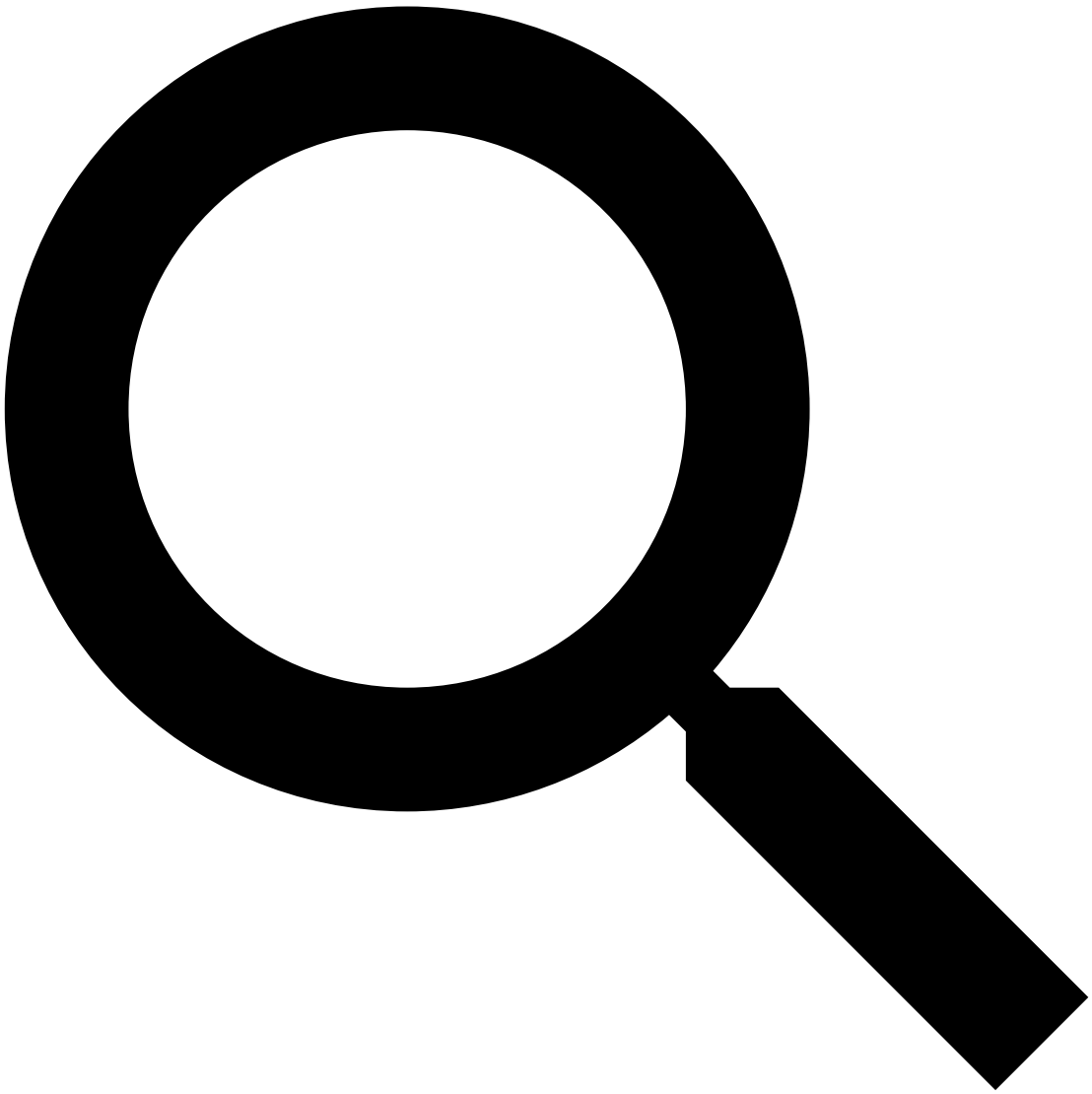
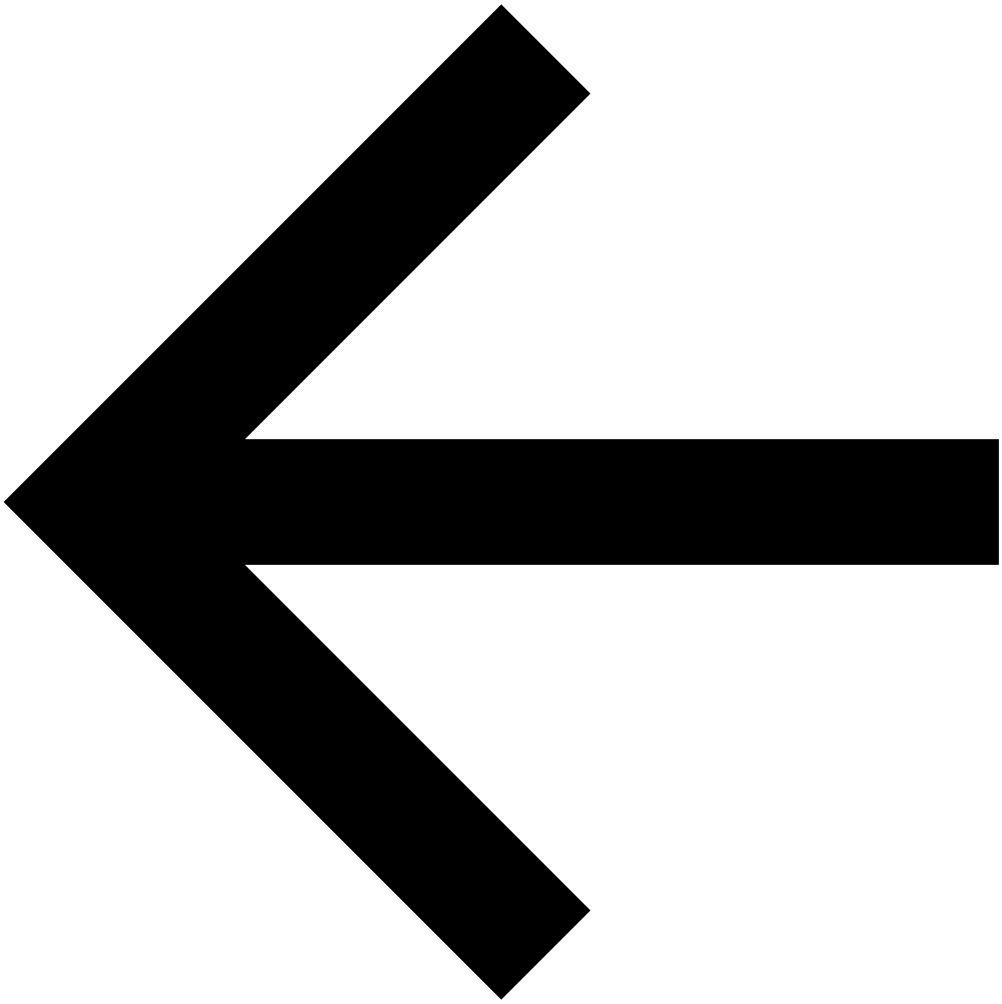


Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☐ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - Postback
  - Live Order Update
- ☒ Data APIs
  Data APIs
  - Market Quote

- Live Market Feed
- ☐ 20 Market Depth 20 Market Depth
  Table of contents
  - Establishing Connection
    - Adding Instruments
    - Keeping Connection Alive
  - 20-Level Depth Packet
    - Response Header
    - Depth Packet
  - Feed Disconnect

Table of contents

- Establishing Connection
  - Adding Instruments
  - Keeping Connection Alive
- 20-Level Depth Packet
  - Response Header
  - Depth Packet
- Feed Disconnect
- Historical Data
- Option Chain
- Annexure
- Instrument List

20 Market Depth

Level 3 data includes market depth upto 20 levels - this shows complete picture of the market movements and it is streamed real-time via websockets.

This data can be used to detect demand supply zones, outside of 5 level market depth and build trading systems to detect market movements.

> Only NSE Equity and Derivatives segments are enabled for 20 Level Market Depth.

Similar to Live Market Feed, all request messages over WebSocket are in JSON whereas all response messages over WebSocket are in Binary.

Establishing Connection

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

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

Query Parameters

Field	Description
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 50 instruments in a single connection and receive market data packets. For subscribing, this can be done using JSON message which needs to be sent over WebSocket connection.

Note

You can send all 50 instruments in a single JSON message. You can send multiple messages over a single connection as well to subscribe to all instruments in parts and receive data.

Request Structure

```
{
    "RequestCode" : 23,
    "InstrumentCount" : 2,
    "InstrumentList" : [
        {
            "ExchangeSegment" : "NSE_EQ",
            "SecurityId" : "1333"
        },
        {
            "ExchangeSegment" : "NSE_FNO",
            "SecurityId" : "532540"
        }
    ]
}
```

Parameters

Field	Type	Description
RequestCode *required*	int	Code for subscribing to particular data mode. `23` for 20 Level Market Depth. 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.

20-Level Depth Packet

The market depth data is sent as structured binary packet. It will require parsing to readable format to extract the relevant information.

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 12 bytes which will remain as part of the response message. The message structure is given as below.

Bytes	Type	Size	Description
1-2	int16	2	Message Length of the entire payload packet
3	[ ] byte	1	Feed Response Code can be referred in Annexure
4	[ ] byte	1	Exchange Segment can be referred in Annexure
5-8	int32	4	Security ID - can be found here
9-12	uint32	4	Message Sequence (to be ignored)

Depth Packet

Depth Data Packet for 20 level market depth is structured differently from 5 level depth. Over here, you will receive the bid (sell) and ask (buy) data packets separately, each containing 20 packets of 16 bytes each.

Bytes	Type	Size	Description
			Response Header
			41 for Bid Data (Buy)

0-12	[ ] array	12	51 for Ask Data (Sell) Refer to enum for values
13-332	Bid/Ask Depth Structure	320	20 packets of 16 bytes each for each instrument in below provided structure

Each of these 20 packets will be received in the following packet structure:

Bytes	Type	Size	Description
1-8	float64	8	Price
9-12	uint32	4	Quantity
13-16	uint32	4	No. of Orders

Note

Whenever 20 level depth packets are sent on the connection, they are stacked one after another in a single message. For example, if two instruments are subscribed, then the first instrument's Bid packet followed by Ask packet of that instrument is added and then the second instrument's bid and ask packets in same sequence. To handle this, you can break down the packet on the basis of length.

Feed Disconnect

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

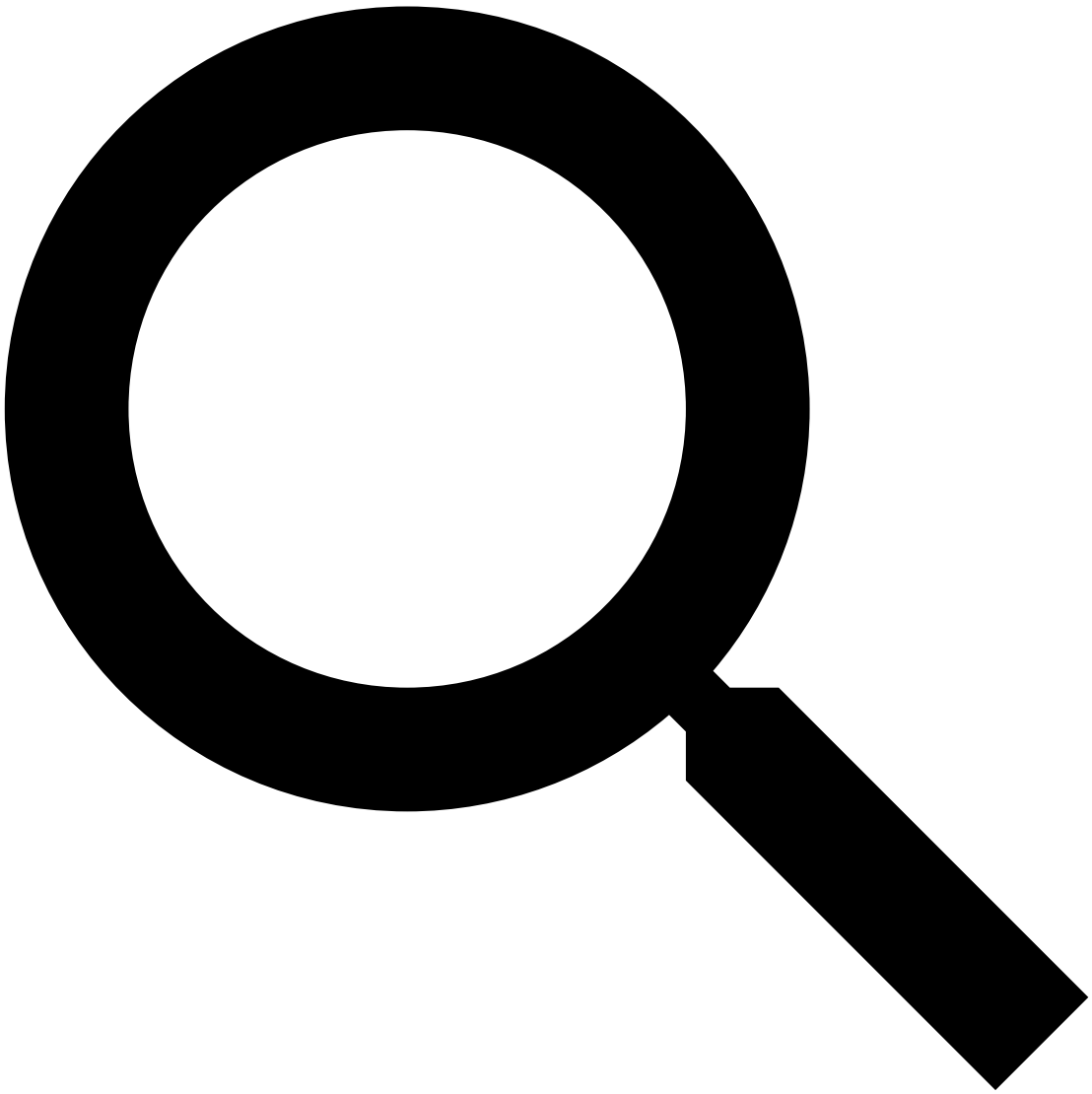
```
{
    "RequestCode" : 12
}
```

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

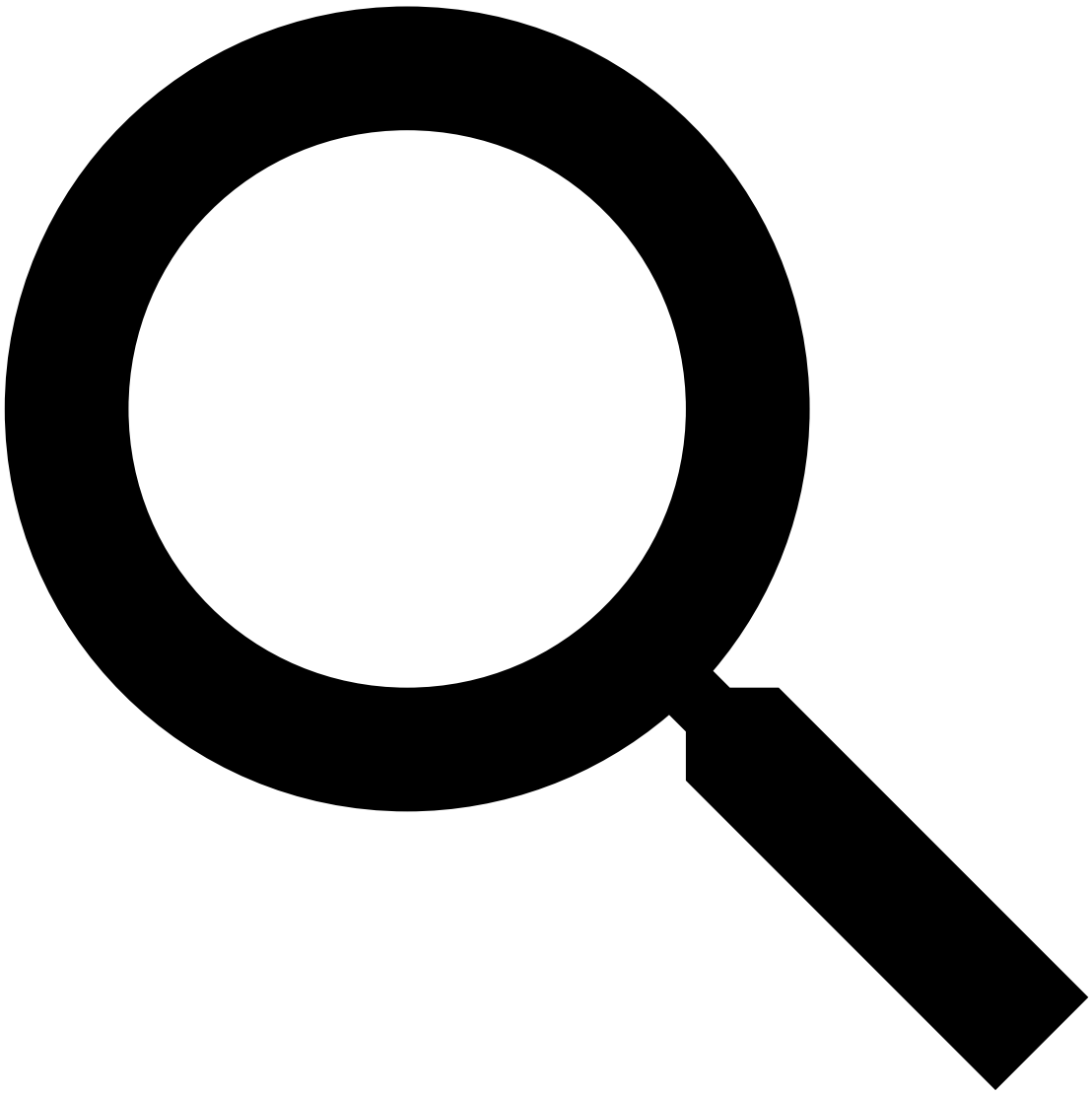
- If more than 5 websockets are established, then the first socket will be disconnected with 805 with every additional connection.

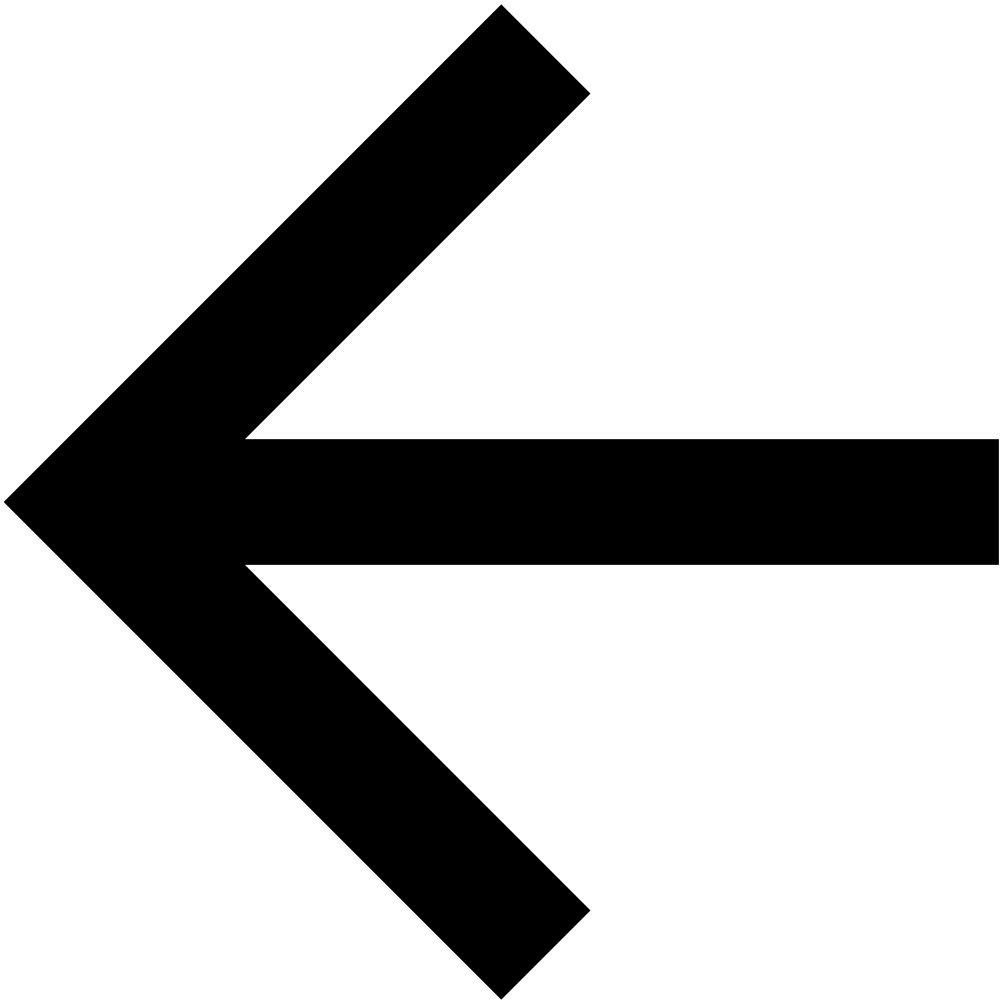
Bytes	Type	Size	Description
0-12	[ ] array	8	Response Header with code 50 Refer to enum for values
13-14	int16	2	Disconnection message code - here

You can find detailed Disconnection message code description here.



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☐ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - Postback
  - Live Order Update
- ☐ Data APIs
  Data APIs
  - Market Quote

- Live Market Feed
- 20 Market Depth
- Historical Data
- Option Chain
- ☐ Annexure Annexure

Table of contents
- Exchange Segment
- Product Type
- Order Status
- After Market Order time
- Expiry Code
- Instrument
- Feed Request Code
- Feed Response Code
- Trading API Error
- Data API Error

Table of contents

- Exchange Segment
- Product Type
- Order Status
- Instrument List
- After Market Order time
- Expiry Code
- Instrument
- Feed Request Code
- Feed Response Code
- Trading API Error
- Data API Error

Annexure

Exchange Segment

Attribute	Exchange Segment	enum
IDX_I	Index	Index Value 0
NSE_EQ	NSE	Equity Cash 1
NSE_FNO	NSE	Futures & Options 2
NSE_CURRENCY	NSE	Currency 3
BSE_EQ	BSE	Equity Cash 4
MCX_COMM	MCX	Commodity 5
BSE_CURRENCY	BSE	Currency 7
BSE_FNO	BSE	Futures & Options 8

Product Type

CO & BO product types will be valid only for Intraday.

Attribute	Detail
CNC	Cash & Carry for equity deliveries
INTRADAY	Intraday for Equity, Futures & Options
MARGIN	Carry Forward in Futures & Options
CO	Cover Order
BO	Bracket Order

Order Status

Attribute	Detail
TRANSIT	Did not reach the exchange server
PENDING	Awaiting execution
CLOSED	Used for Super Order, once both the entry and exit orders are placed
TRIGGERED	Used for Super Order, if Target or Stop Loss leg is triggered
REJECTED	Rejected by broker/exchange
CANCELLED	Cancelled by user
PART_TRADED	Partial Quantity traded successfully
TRADED	Executed successfully

After Market Order time

Attribute	Detail
PRE_OPEN	AMO pumped at pre-market session
OPEN	AMO pumped at market open
OPEN_30	AMO pumped 30 minutes after market open
OPEN_60	AMO pumped 60 minutes after market open

Expiry Code

Attribute	Detail
0	Current Expiry/Near Expiry
1	Next Expiry
2	Far Expiry

Instrument

Attribute	Detail
INDEX	Index
FUTIDX	Futures of Index
OPTIDX	Options of Index
EQUITY	Equity
FUTSTK	Futures of Stock
OPTSTK	Options of Stock
FUTCOM	Futures of Commodity
OPTFUT	Options of Commodity Futures
FUTCUR	Futures of Currency
OPTCUR	Options of Currency

Feed Request Code

Attribute	Detail
11	Connect Feed
12	Disconnect Feed
15	Subscribe - Ticker Packet
16	Unsubscribe - Ticker Packet
17	Subscribe - Quote Packet
18	Unsubscribe - Quote Packet
21	Subscribe - Full Packet
22	Unsubscribe - Full Packet
23	Subscribe - 20 Level Market Depth
24	Unsubscribe - 20 Level Market Depth

Feed Response Code

Attribute	Detail
1	Index Packet
2	Ticker Packet
4	Quote Packet

5	OI Packet
6	Prev Close Packet
7	Market Status Packet
8	Full Packet
50	Feed Disconnect

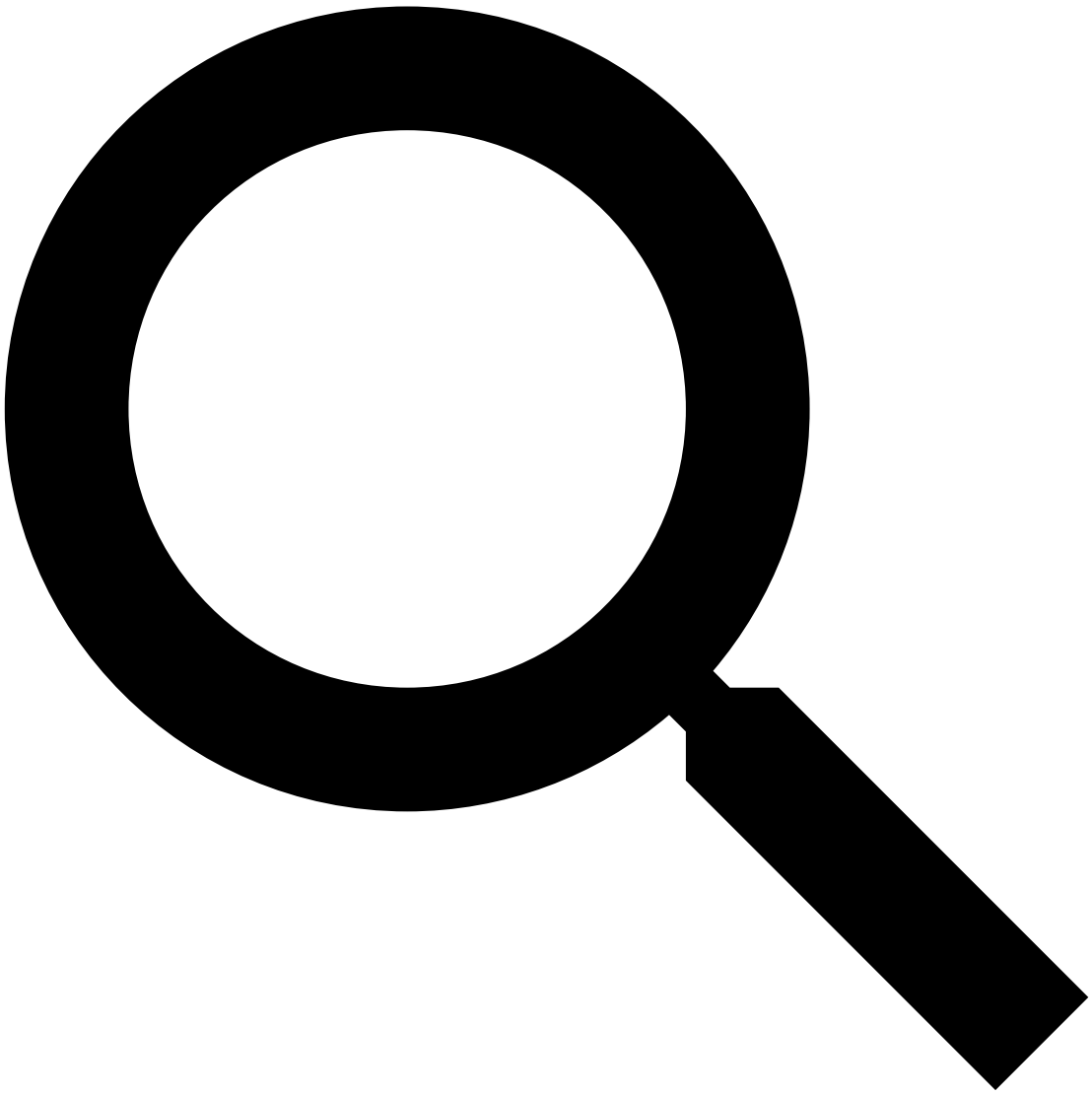
Trading API Error

Type	Code	Message
Invalid Authentication	DH-901	Client ID or user generated access token is invalid or expired.
Invalid Access	DH-902	User has not subscribed to Data APIs or does not have access to Trading APIs. Kindly subscribe to Data APIs to be able to fetch Data.
User Account	DH-903	Errors related to User's Account. Check if the required segments are activated or other requirements are met.
Rate Limit	DH-904	Too many requests on server from single user breaching rate limits. Try throttling API calls.
Input Exception	DH-905	Missing required fields, bad values for parameters etc.
Order Error	DH-906	Incorrect request for order and cannot be processed.
Data Error	DH-907	System is unable to fetch data due to incorrect parameters or no data present.
Internal Server Error	DH-908	Server was not able to process API request. This will only occur rarely.
Network Error	DH-909	Network error where the API was unable to communicate with the backend system.
Others	DH-910	Error originating from other reasons.

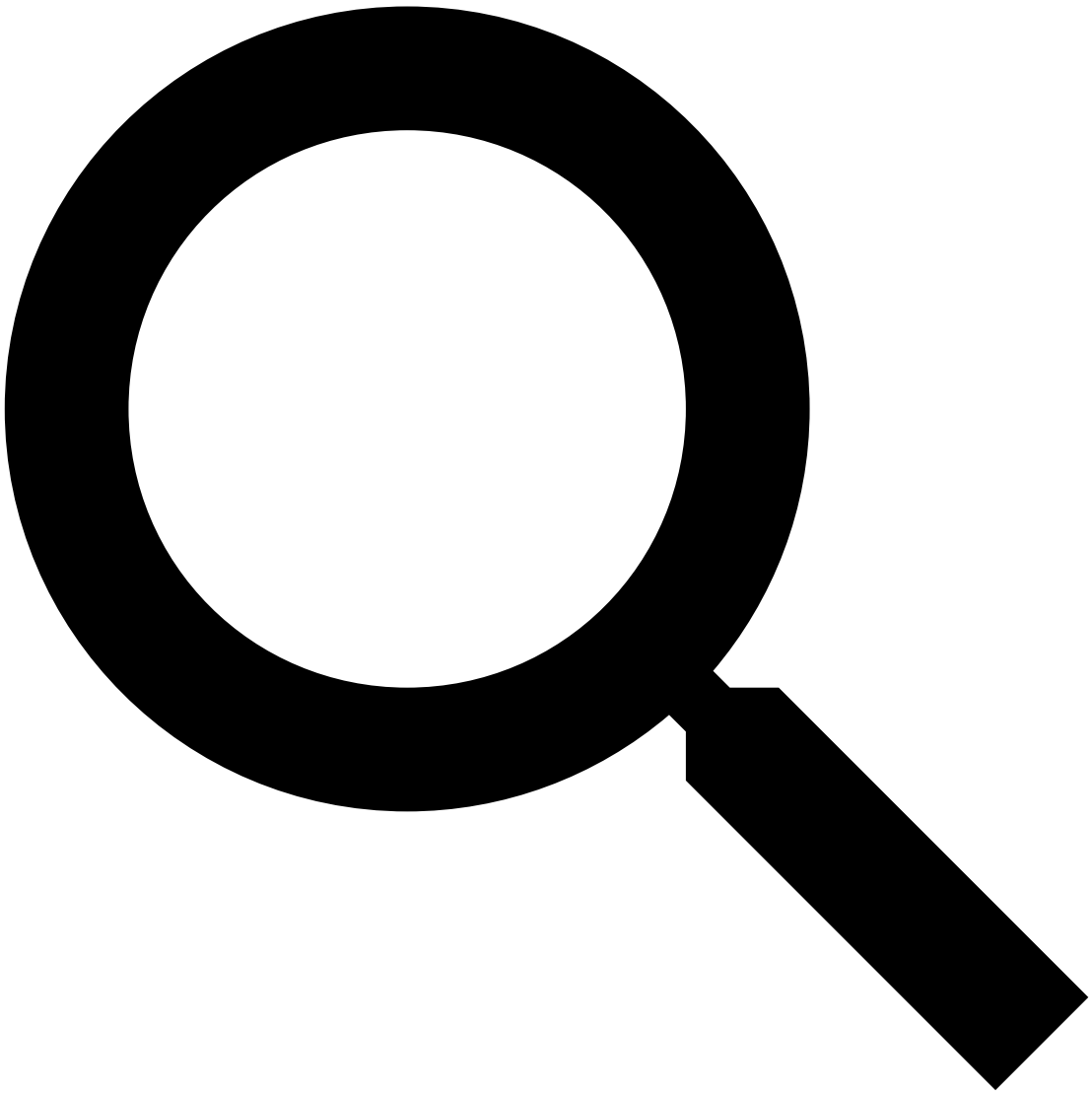
Data API Error

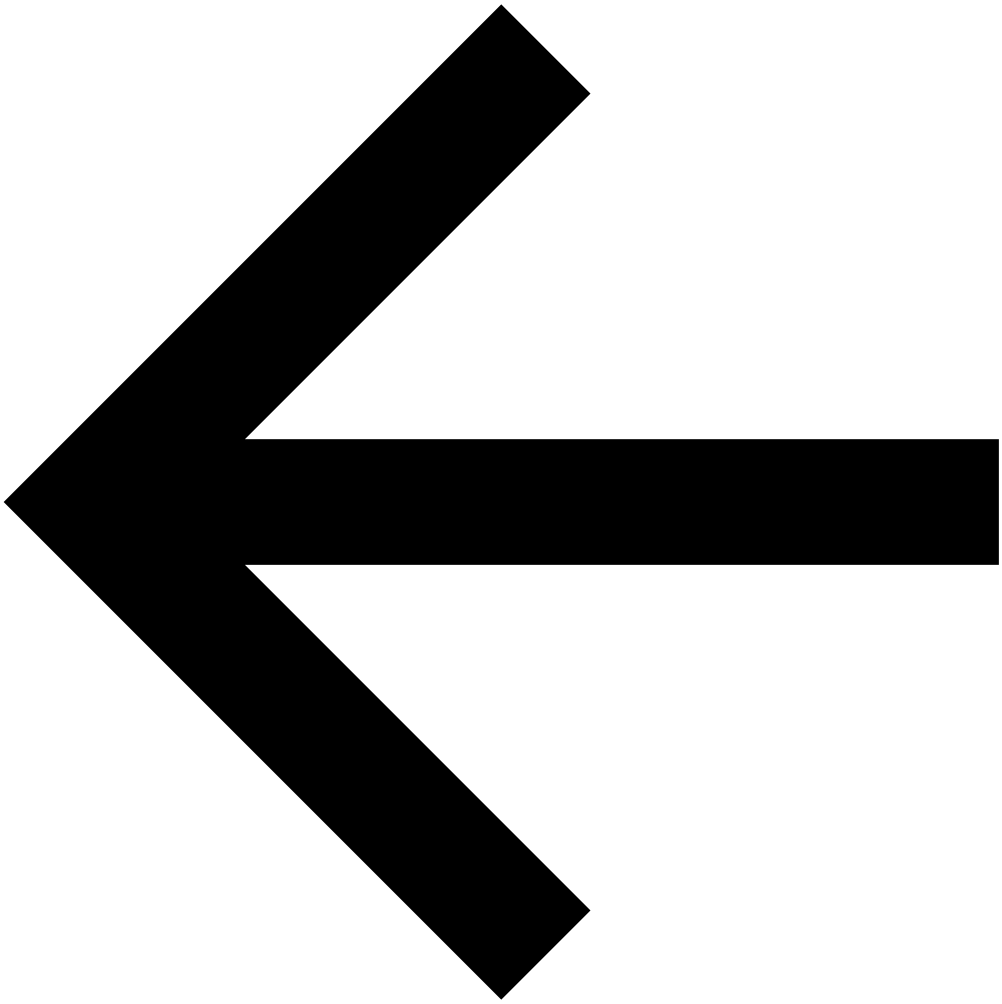
Code	Description
800	Internal Server Error
804	Requested number of instruments exceeds limit
805	Too many requests or connections. Further requests may result in the user being blocked.
806	Data APIs not subscribed
807	Access token is expired
808	Authentication Failed - Client ID or Access Token invalid
809	Access token is invalid
810	Client ID is invalid
811	Invalid Expiry Date
812	Invalid Date Format
813	Invalid SecurityId
814	Invalid Request

Ver 2.0 / API Documentation
Funds



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☒ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - ☐ Funds Funds
    Table of contents
    - Margin Calculator
    - Fund Limit
  - Statement
  - Postback
  - Live Order Update

- ☐ Data APIs
  Data APIs
  - Market Quote
  - Live Market Feed
  - 20 Market Depth
  - Historical Data
  - Option Chain
- Annexure
- Instrument List
- Margin Calculator
- Fund Limit

Table of contents

Funds

Users can get details about the fund requirements or available funds (with margin requirements) in their Trading Account.

POST	/margincalculator	Margin requirement for any order
GET	/fundlimit	Retrieve trading account fund information

Margin Calculator

Fetch span, exposure, var, brokerage, leverage, available margin values for any type of order and instrument that you want to place.

```
  curl --request POST \
--url https://api.dhan.co/v2/margincalculator \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'access-token: ' \
--data '{Request JSON}'
```

Request Structure

◉
cURL

```
{
    "dhanClientId": "1000000132",
    "exchangeSegment": "NSE_EQ",
    "transactionType": "BUY",
    "quantity": 5,
    "productType": "CNC",
    "securityId": "1333",
    "price": 1428,
    "triggerPrice": 1427,
}
```

Parameters

Field	Type	Description
dhanClientId *required*	string	User specific identification generated by Dhan
exchangeSegment *required*	enum string	Exchange & Segment `NSE_EQ` `NSE_FNO` `BSE_EQ` `BSE_FNO` `MCX_COMM`
transactionType *required*	enum string	The trading side of transaction `BUY` `SELL`
quantity *required*	int	Number of shares for the order
productType *required*	enum string	Product type `CNC` `INTRADAY` `MARGIN` `MTF` `CO` `BO`
securityId *required*	string	Exchange standard id for each scrip. Refer **here**
price *required*	float	Price at which order is placed
triggerPrice *conditionally required*	float	Price at which the order is triggered, in case of SL-M & SL-L

Response Structure

```
{
    "totalMargin": 2800.00,
    "spanMargin": 1200.00,
    "exposureMargin": 1003.00,
    "availableBalance": 10500.00,
    "variableMargin": 1000.00,
    "insufficientBalance": 0.00,
    "brokerage": 20.00,
    "leverage": "4.00"
}
```

Parameters

Field	Type	Description
totalMargin	float	Total Margin required for placing the order successfully
spanMargin	float	SPAN margin required
exposureMargin	float	Exposure margin required
availableBalance	float	Available amount in trading account
variableMargin	float	VAR or Variable margin required
insufficientBalance	float	Insufficient amount in trading account (Available Balance - Total Margin)
brokerage	float	Brokerage charges for executing order
leverage	string	Margin leverage provided for the order as per product type

Fund Limit

Get all information of your trading account like balance, margin utilised, collateral, etc.

```
curl --request GET \
  --url https://api.dhan.co/v2/fundlimit \
  --header 'Content-Type: application/json' \
  --header 'access-token: JWT'
```

Request Structure

No Body

Response Structure

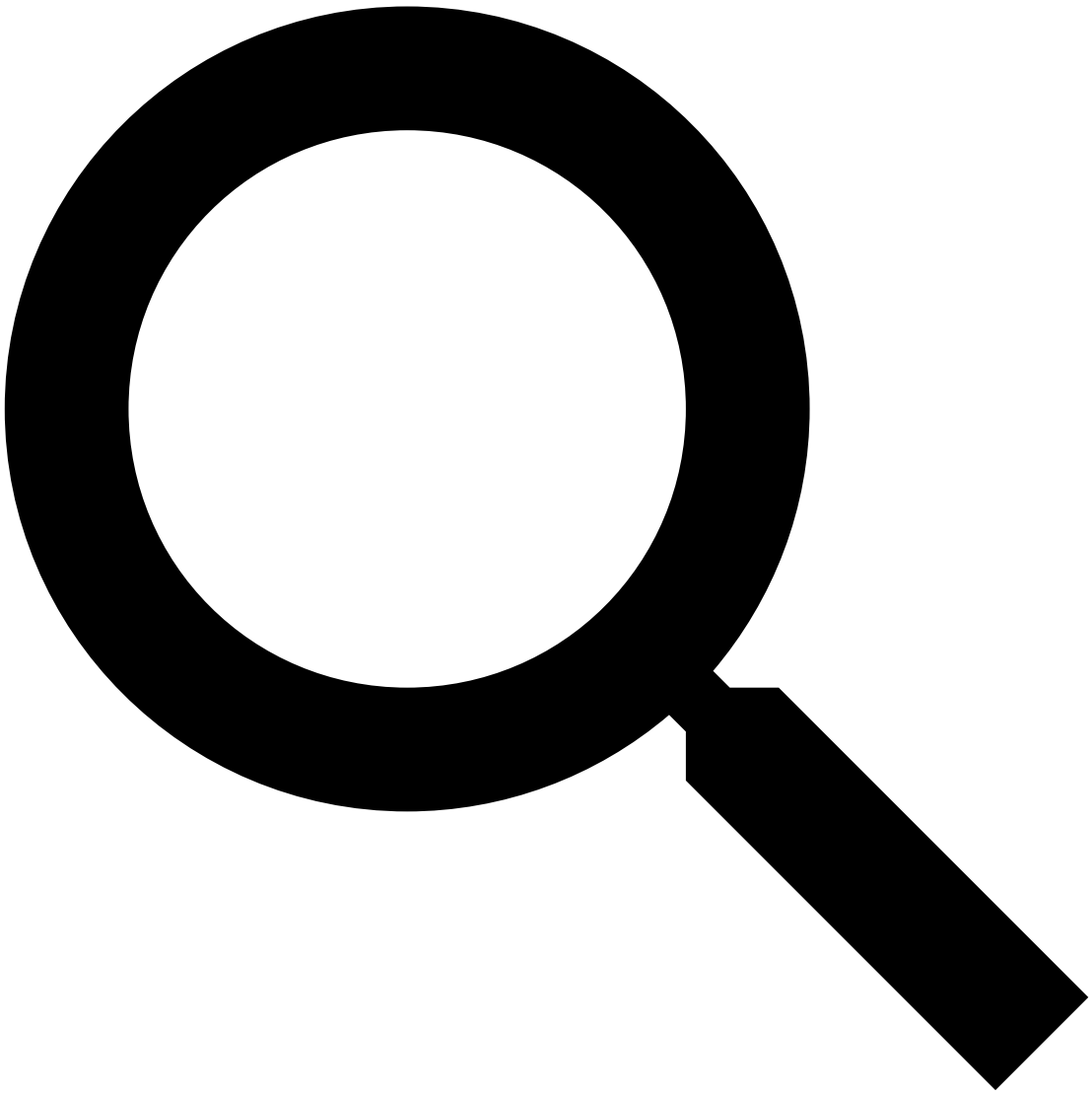
```
{
    "dhanClientId":"1000000009",
    "availabelBalance": 98440.0,
    "sodLimit": 113642,
    "collateralAmount": 0.0,
    "receiveableAmount": 0.0,
    "utilizedAmount": 15202.0,
    "blockedPayoutAmount": 0.0,
    "withdrawableBalance": 98310.0
}
```

Parameters

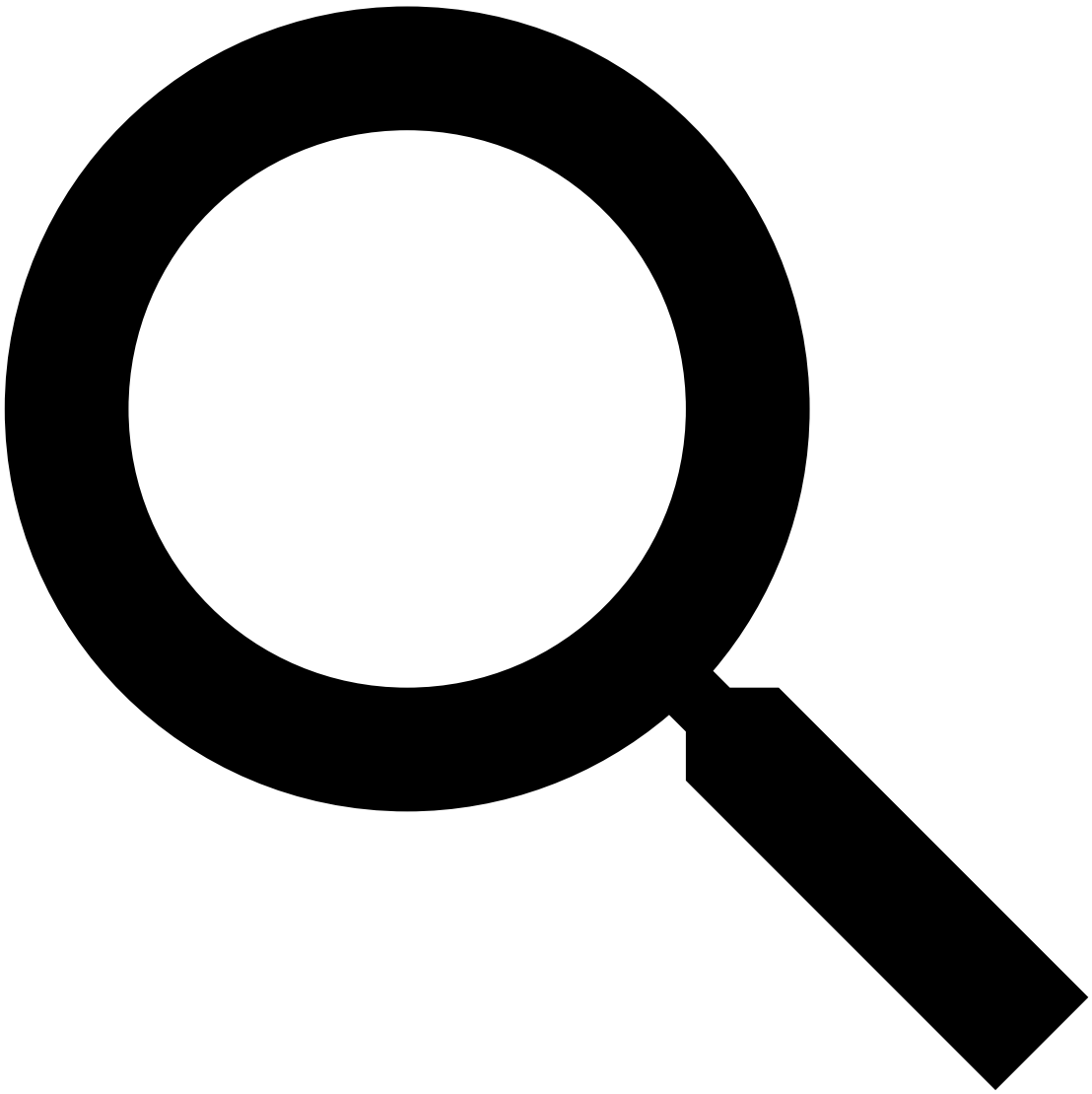
Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
availabelBalance	float	Available amount to trade
sodLimit	float	Start of the day balance in account
collateralAmount	float	Amount received against collateral
receiveableAmount	float	Amount available against selling deliveries
utilizedAmount	float	Amount utilised in the day
blockedPayoutAmount	float	Amount blocked against payout request
withdrawableBalance	float	Amount available to withdraw in bank account

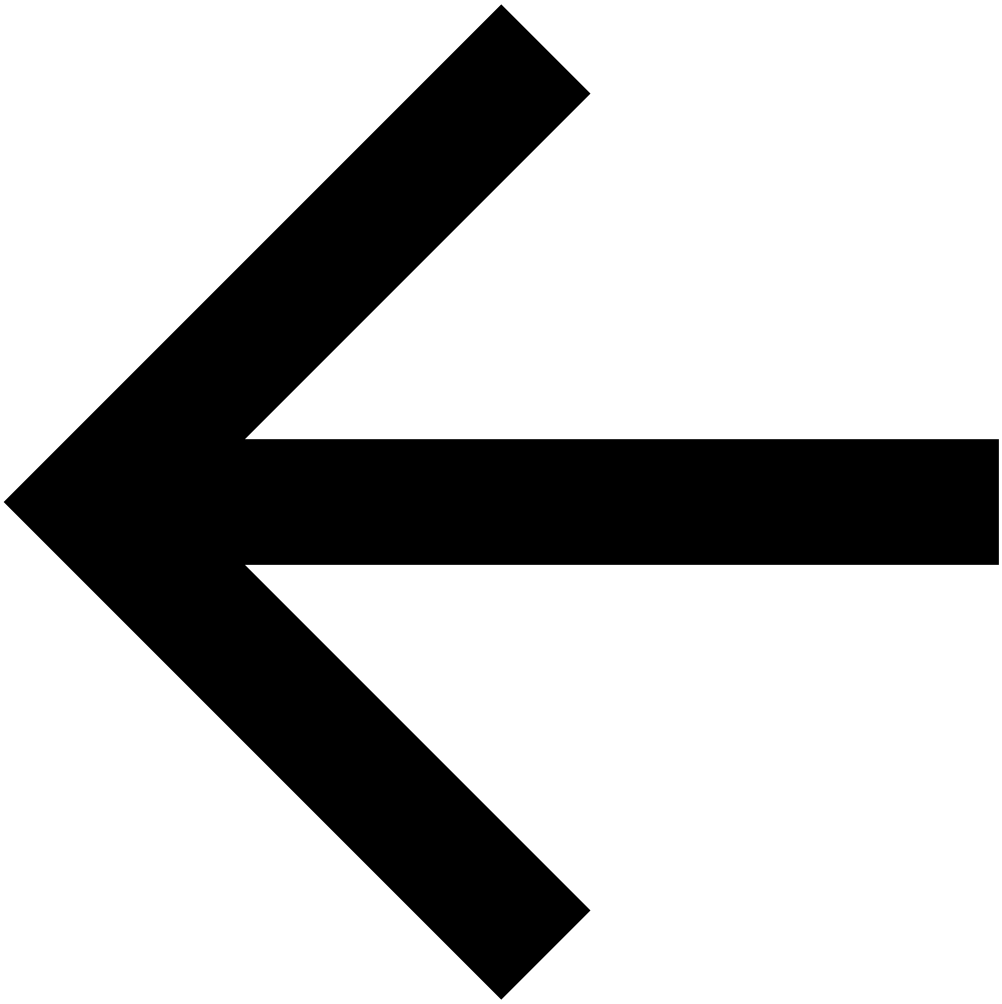
Note: For description of enum values, refer Annexure

Ver 2.0 / API Documentation
Instrument List



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☐ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - Postback
  - Live Order Update
- ☐ Data APIs
  Data APIs
  - Market Quote

- Live Market Feed
- 20 Market Depth
- Historical Data
- Option Chain

- Annexure
- ☐ Instrument List Instrument List
  Table of contents
  - Segmentwise List
  - Column Description

Table of contents

- Segmentwise List
- Column Description

Instrument List

You can fetch instrument list for all instruments which can be traded via Dhan by using below URL:

Compact:

```
https://images.dhan.co/api-data/api-scrip-master.csv
```

Detailed:

```
https://images.dhan.co/api-data/api-scrip-master-detailed.csv
```

This fetches list of instruments as CSV with Security ID and other important details which will help you build with DhanHQ APIs.

Segmentwise List

You can fetch detailed instrument list for all instruments in a particular exchange and segment by passing the same in parameters as below:

```
curl --location 'https://api.dhan.co/v2/instrument/{exchangeSegment}' \
```

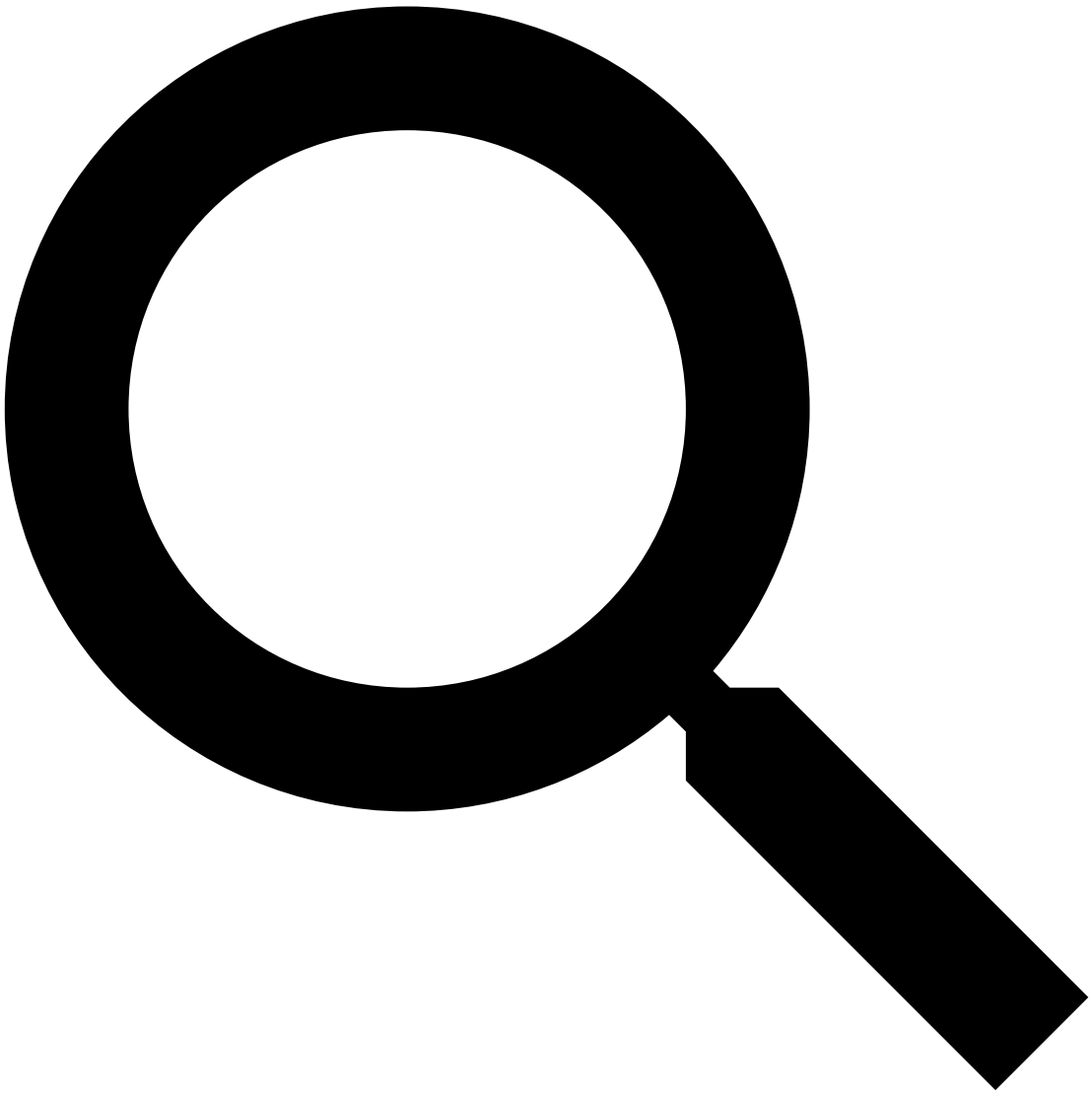
> This helps to fetch instrument list of only one particular `exchangeSegment` at a time. The mapping of the same can be found here.

Column Description

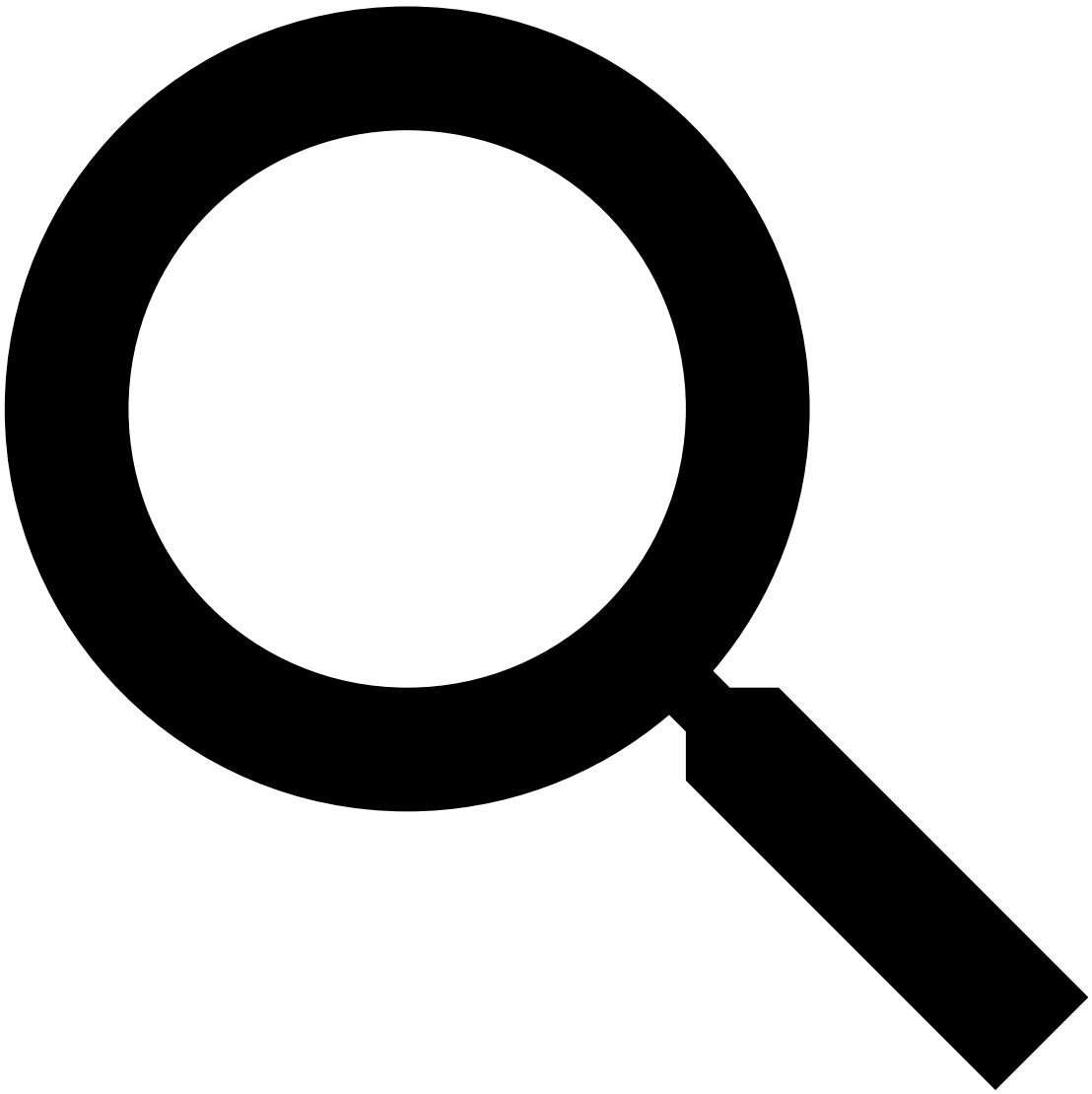
Detailed	Compact	Description
`tag`	`tag`	
`EXCH_ID`	`SEM_EXM_EXCH_ID`	Exchange `NSE BSE MCX`
`SEGMENT`	`SEM_SEGMENT`	Segment `C` - Currency `D` - Derivatives `E` - Equity `M` - Commodity
		International Securities Identification Number(ISIN) - 12-digit

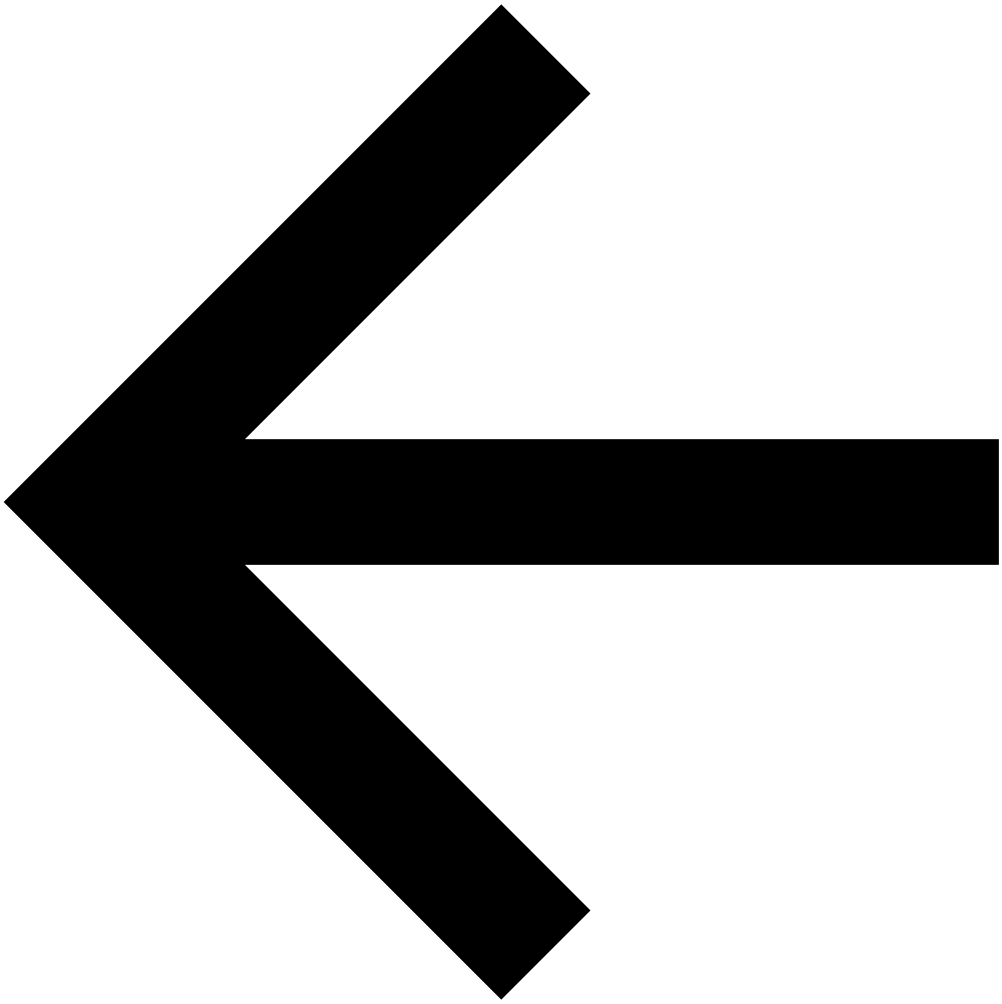
ISIN	-	alphanumeric code unique for instruments
INSTRUMENT	SEM_INSTRUMENT_NAME	Instrument defined by Exchange - defined here
removed	SEM_EXPIRY_CODE	Expiry Code (applicable in case of Futures Contract) - defined here
UNDERLYING_SECURITY_ID	-	Security ID of underlying instrument (applicable in case of derivative contracts)
UNDERLYING_SYMBOL	-	Symbol of underlying instrument (applicable in case of derivative contracts)
SYMBOL_NAME	SM_SYMBOL_NAME	Symbol name of instrument
removed	SEM_TRADING_SYMBOL	Exchange trading symbol of instrument
DISPLAY_NAME	SEM_CUSTOM_SYMBOL	Dhan display symbol name of instrument
INSTRUMENT_TYPE	SEM_EXCH_INSTRUMENT_TYPE	In addition to `INSTRUMENT` column, instrument type is defined by exchange adding more details about instrument
SERIES	SEM_SERIES	Exchange defined series for instrument
LOT_SIZE	SEM_LOT_UNITS	Lot Size in multiples of which instrument is traded
SM_EXPIRY_DATE	SEM_EXPIRY_DATE	Expiry date of instrument (applicable in case of derivative contracts)
STRIKE_PRICE	SEM_STRIKE_PRICE	Strike Price of Options Contract
OPTION_TYPE	SEM_OPTION_TYPE	Type of Options Contract CE - Call PE - Put
TICK_SIZE	SEM_TICK_SIZE	Minimum decimal point at which an instrument can be priced
EXPIRY_FLAG	SEM_EXPIRY_FLAG	Type of Expiry (applicable in case of option contracts) M - Monthly Expiry W - Weekly Expiry
BRACKET_FLAG	-	Bracket order status N - Not available Y - Allowed
COVER_FLAG	-	Cover order status N - Not available Y - Allowed
ASM_GSM_FLAG	-	Flag for instrument is ASM or GSM N - Not in ASM/GSM R - Removed from block Y - ASM/GSM
ASM_GSM_CATEGORY	-	Category of instrument in ASM or GSM NA in case of no surveillance
BUY_SELL_INDICATOR	-	Indicator to show if Buy and Sell is allowed in instrument A if both Buy/Sell is allowed
BUY_CO_MIN_MARGIN_PER	-	Buy cover order minimum margin requirement (in percentage)
SELL_CO_MIN_MARGIN_PER	-	Sell cover order minimum margin requirement (in percentage)
BUY_CO_SL_RANGE_MAX_PERC	-	Buy cover order maximum range for stop loss leg (in percentage)
SELL_CO_SL_RANGE_MAX_PERC	-	Sell cover order maximum range for stop loss leg (in percentage)
BUY_CO_SL_RANGE_MIN_PERC	-	Buy cover order minimum range for stop loss leg (in percentage)
SELL_CO_SL_RANGE_MIN_PERC	-	Sell cover order minimum range for stop loss leg (in percentage)
BUY_BO_MIN_MARGIN_PER	-	Buy bracket order minimum margin requirement (in percentage)
SELL_BO_MIN_MARGIN_PER	-	Sell bracket order minimum margin requirement (in percentage)
BUY_BO_SL_RANGE_MAX_PERC	-	Buy bracket order maximum range for stop loss leg (in percentage)
SELL_BO_SL_RANGE_MAX_PERC	-	Sell bracket order maximum range for stop loss leg (in percentage)
BUY_BO_SL_RANGE_MIN_PERC	-	Buy bracket order minimum range for stop loss leg (in percentage)
SELL_BO_SL_MIN_RANGE	-	Sell bracket order minimum range for stop loss leg (in percentage)
BUY_BO_PROFIT_RANGE_MAX_PERC	-	Buy bracket order maximum range for target leg (in percentage)
SELL_BO_PROFIT_RANGE_MAX_PERC	-	Sell bracket order maximum range for target leg (in percentage)
BUY_BO_PROFIT_RANGE_MIN_PERC	-	Buy bracket order minimum range for target leg (in percentage)
SELL_BO_PROFIT_RANGE_MIN_PERC	-	Sell bracket order minimum range for target leg (in percentage)
MTF_LEVERAGE	-	MTF Leverage available (in x multiple) for eligible `EQUITY` instruments

Ver 2.0 / API Documentation
Live Market Feed



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☐ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - Postback
  - Live Order Update
- ☒ Data APIs
  Data APIs
  - Market Quote

- ☐ Live Market Feed Live Market Feed
  Table of contents
  - Establishing Connection
    - Adding Instruments
    - Keeping Connection Alive
  - Market Data
    - Response Header
    - Ticker Packet
      - Prev Close
    - Quote Packet
      - OI Data
    - Full Packet
      - Feed Disconnect
    - 20 Market Depth
    - Historical Data
    - Option Chain
  - Annexure
  - Instrument List
  - Full Packet
- Feed Disconnect

Table of contents

- Establishing Connection
  - Adding Instruments
  - Keeping Connection Alive
- Market Data
  - Response Header
  - Ticker Packet
    - Prev Close
  - Quote Packet
    - OI Data
  - Full Packet
    - Feed Disconnect
  - 20 Market Depth
  - Historical Data
  - Option Chain
- Annexure
- Instrument List
  - Full Packet
- Feed Disconnect

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	`2` by Default

required

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
}
```

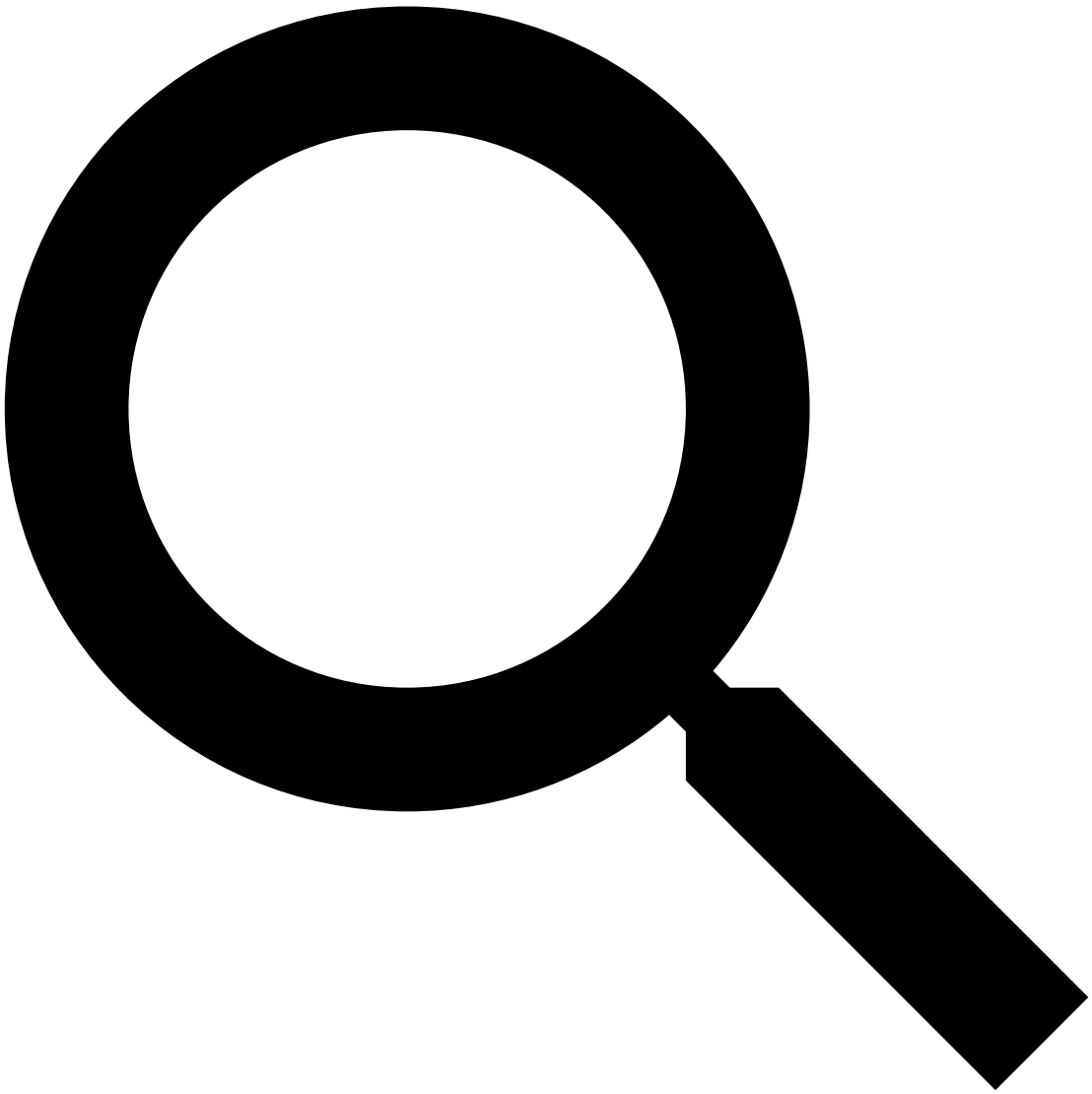
In case of WebSocket disconnection from server side, you will receive disconnection packet, which will have disconnection reason code.

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

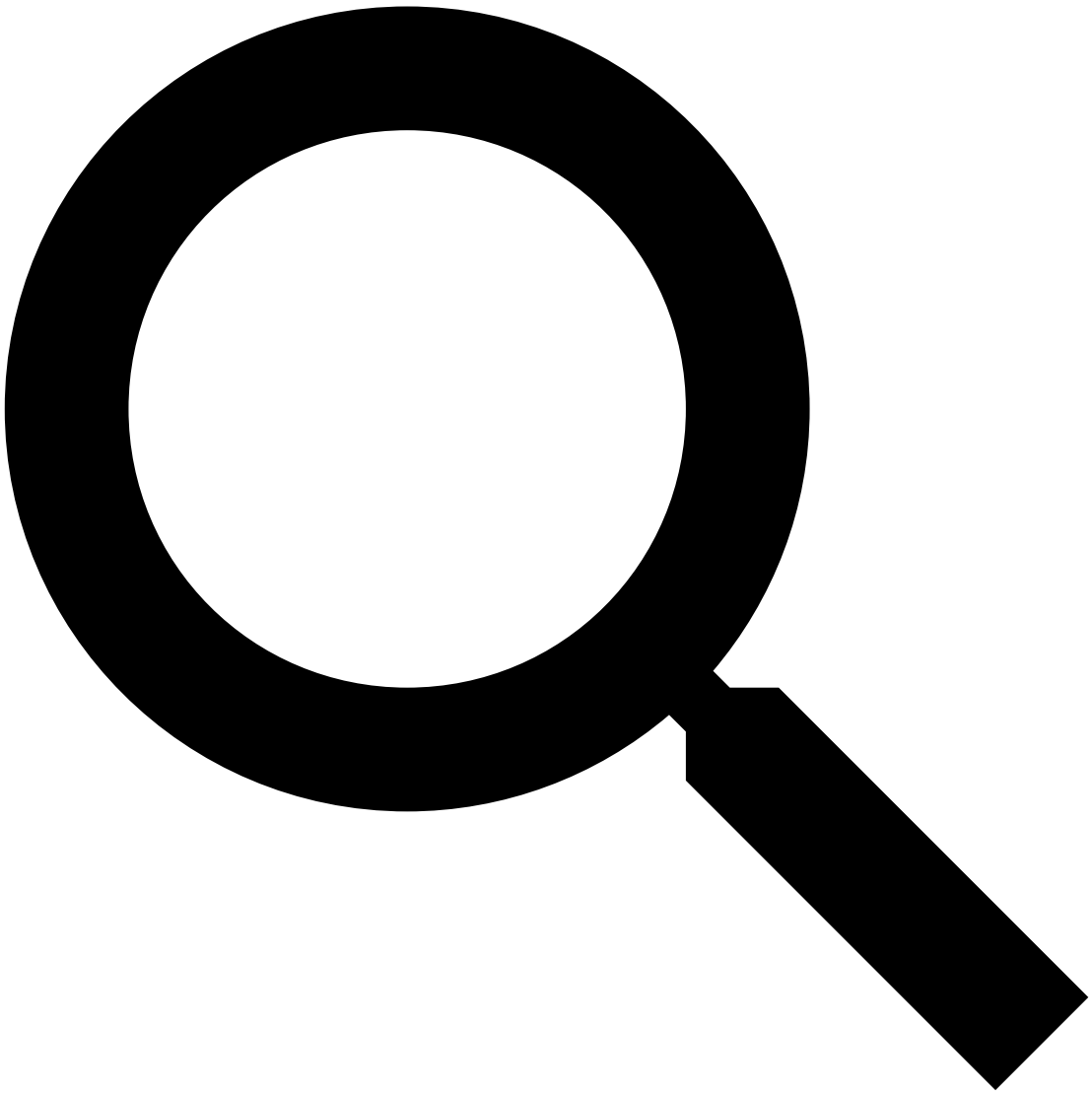
Bytes	Type	Size	Description
0-8	[ ] array	8	Response Header with code 50 Refer to enum for values
9-10	int16	2	Disconnection message code - here

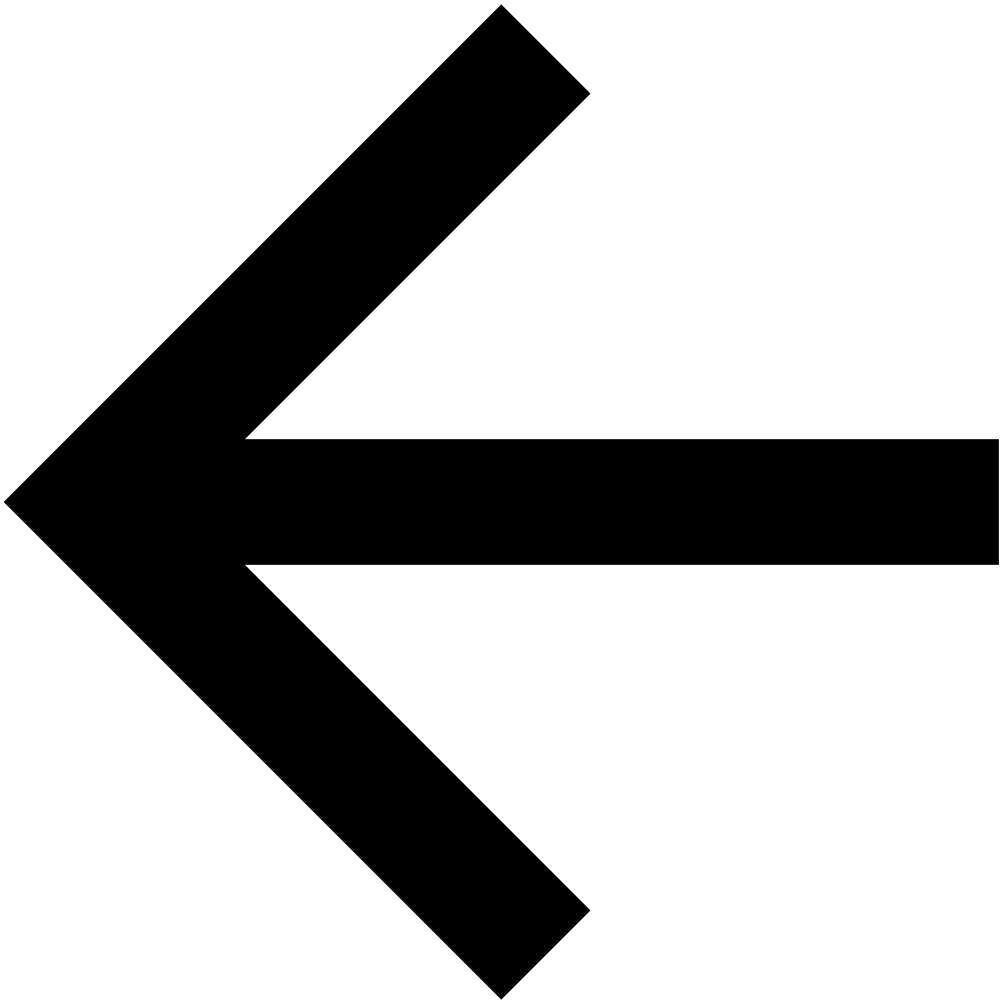
You can find detailed Disconnection message code description here.

Ver 2.0 / API Documentation
Live Order Update



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☒ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - Postback
  - ☐ Live Order Update Live Order Update
    Table of contents
    - Establishing Connection
      - For Individual

- For Partners
- Order Update
- ☐ Data APIs
  Data APIs
  - Market Quote
  - Live Market Feed
  - 20 Market Depth
  - Historical Data
  - Option Chain
- Annexure
- Instrument List

Table of contents

- Establishing Connection
  - For Individual
  - For Partners
- Order Update

Live Order Update

Realtime order updates of all your orders can be received directly via WebSocket in your system. Once you connect to the WebSocket and authorise, all order updates in your acount will get reflected real time via the stream.

With this update stream, you can know about status, traded price, quantity and other details about your orders.

The messages sent over this WebSocket will be JSON.

Establishing Connection

To establish connection with DhanHQ Live Order Update, you can connect to the below endpoint using WebSocket library.

```
wss://api-order-update.dhan.co
```

While establishing connection, you need to send Authorisation Message for connection.

For Individual

You can receive order updates for all orders placed via your account, irrespective of the platform via which it was placed.

Authorisation message structure

```
{
    "LoginReq":{
        "MsgCode": 42,
        "ClientId":"1000000001",
```

```
        "Token":"JWT"
    },
    "UserType: "SELF"
}
```

Parameters

Field	Type	Description
LoginReq *required*	{}, string	JSON for adding Client ID and Access Token
MsgCode *required*	int	Message Code for getting Order Updates `42` by default
ClientId *required*	string	User specific identification generated by Dhan
Token *required*	string	Access Token generated for user
UserType *required*	string	`SELF` for individual users

For Partners

Platforms can receive order updates originating for all users connected to their platform/app for which Partner Login module needs to be used.

Authorisation message structure

```
{
    "LoginReq":{
        "MsgCode": 42,
        "ClientId": "partner_id"
    },
    "UserType: "PARTNER",
    "Secret": "partner_secret"
}
```

Parameters

Field	Type	Description
LoginReq *required*	{}, string	JSON for adding Client ID and Access Token
MsgCode *required*	int	Message Code for getting Order Updates `42` by default
ClientId *required*	string	`partner_id` generated for the partner
UserType *required*	string	`PARTNER` for partner platforms
Secret *required*	string	`partner_secret` generated for the partner

Order Update

Order Update messages are sent via WebSocket in below structure.

Structure

```
{
    "Data": {
        "series": "EQ",
        "goodTillDaysDate": "2024-09-11",
        "instrumentType": "EQ",
        "refLtp": 13.21,
        "tickSize": 0.01,
        "algoId": "0",
        "multiplier": 1
        "Exchange": "NSE",
        "Segment": "E",
        "Source": "N",
        "SecurityId": "14366",
```

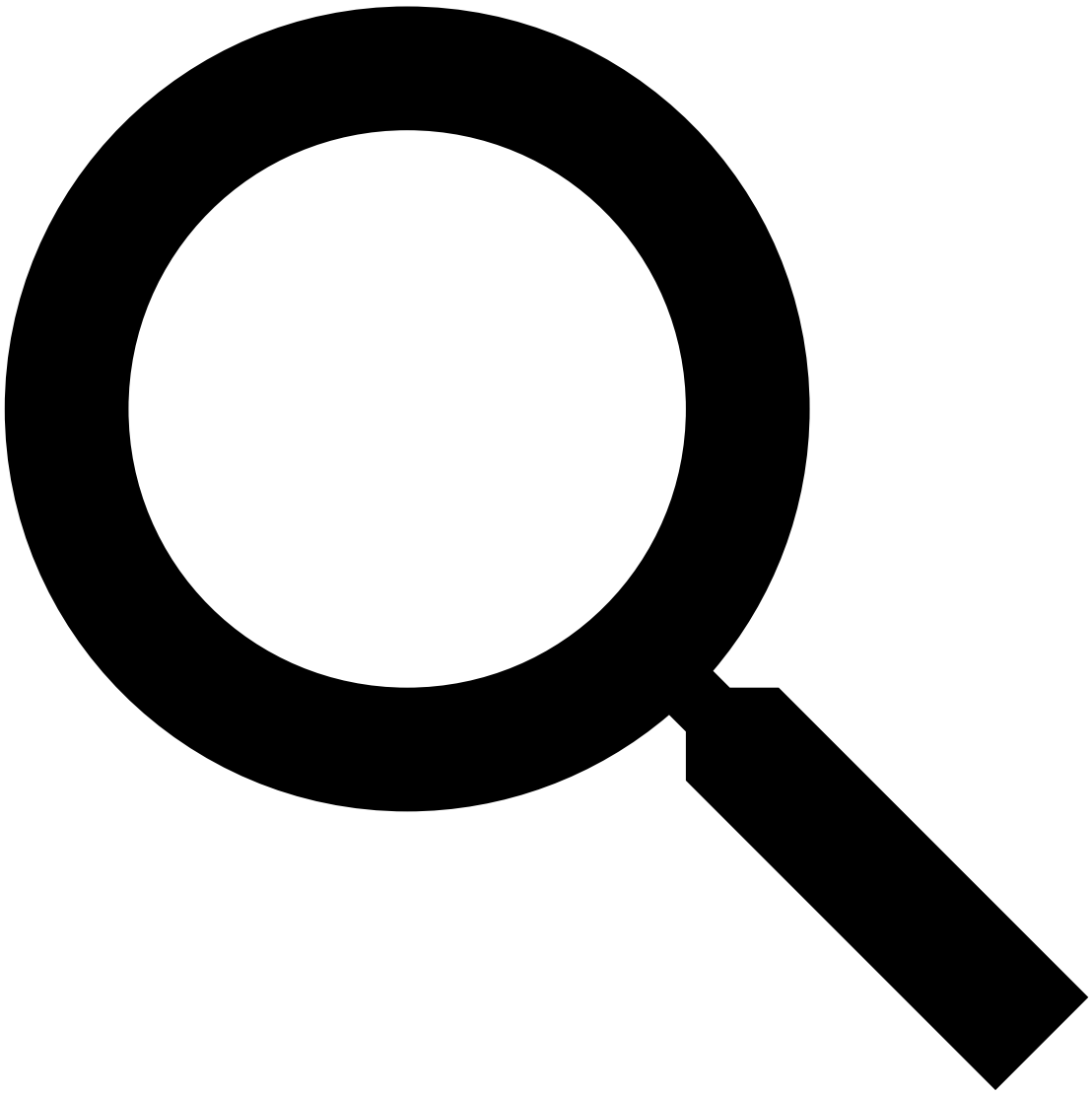
```
        "ClientId": "1000000001",
        "ExchOrderNo": "1400000000404591",
        "OrderNo": "1124091136546",
        "Product": "C",
        "TxnType": "B",
        "OrderType": "LMT",
        "Validity": "DAY",
        "DiscQuantity": 1,
        "DiscQtyRem": 1,
        "RemainingQuantity": 1,
        "Quantity": 1,
        "TradedQty": 0,
        "Price": 13,
        "TriggerPrice": 0,
        "TradedPrice": 0,
        "AvgTradedPrice": 0,
        "AlgoOrdNo": ,
        "OffMktFlag": "0",
        "OrderDateTime": "2024-09-11 14:39:29",
        "ExchOrderTime": "2024-09-11 14:39:29",
        "LastUpdatedTime": "2024-09-11 14:39:29",
        "Remarks": "NR",
        "MktType": "NL",
        "ReasonDescription": "CONFIRMED",
        "LegNo": 1,
        "Instrument": "EQUITY",
        "Symbol": "IDEA",
        "ProductName": "CNC",
        "Status": "Cancelled",
        "LotSize": 1,
        "StrikePrice": ,
        "ExpiryDate": "0001-01-01 00:00:00",
        "OptType": "XX",
        "DisplayName": "Vodafone Idea",
        "Isin": "INE669E01016",
        "Series": "EQ",
        "GoodTillDaysDate": "2024-09-11",
        "RefLtp": 13.21,
        "TickSize": 0.01,
        "AlgoId": "0",
        "Multiplier": 1,
        "CorrelationId": "",
        "Remarks": "Super Order"
    },
    "Type": "order_alert"
}
```


Parameters

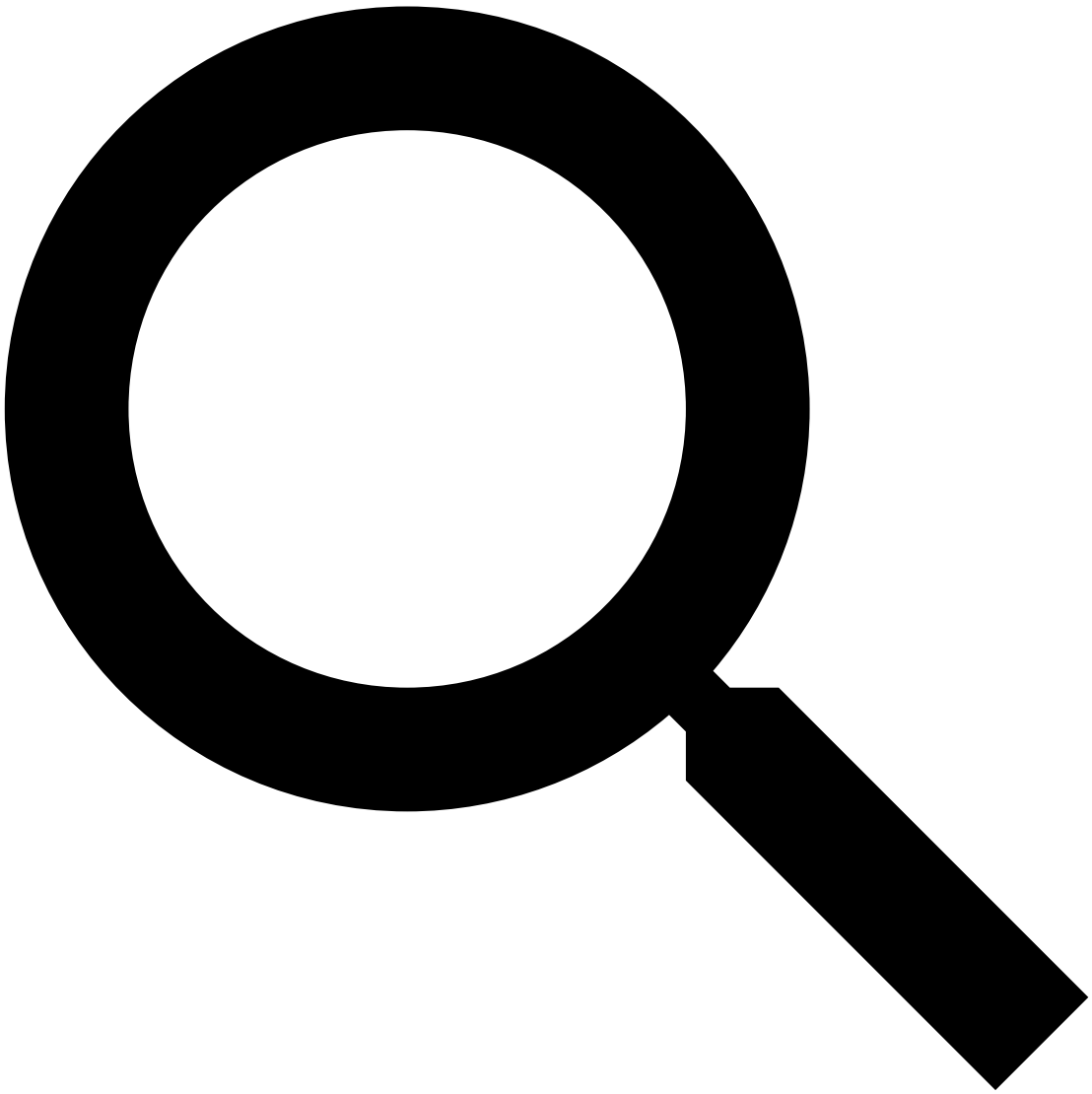
Field	Type	Description
Exchange	string	Exchange in which order is placed
Segment	string	Segment for which order is placed
Source	string	Platform via which order is placed - `P` for API Orders
SecurityId	string	Exchange standard ID for each scrip. Refer **here**
ClientId	string	User specific identification generated by Dhan
ExchOrderNo	string	Order specific identification generated by Exchange
OrderNo	string	Order specific identification generated by Dhan
Product	enum string	Product type of trade `C` for CNC, `I` for INTRADAY, `M` for MARGIN, `F` for MTF , `V` for CO, `B` for BO
TxnType	enum string	The trading side of transaction `B` for Buy `S` for Sell
OrderType	enum string	Order Type `LMT` for Limit `MKT` for Market `SL` for Stop Loss `SLM` for Stop Loss
Validity	enum string	Validity of Order `DAY` `IOC`
DiscQuantity	int	Number of shares visible
DiscQtyRem	int	Disclosed quantity pending for execution
RemainingQuantity	int	Quantity pending for execution
Quantity	int	Total order quantity placed
TradedQty	int	Actual quantity executed on exchange

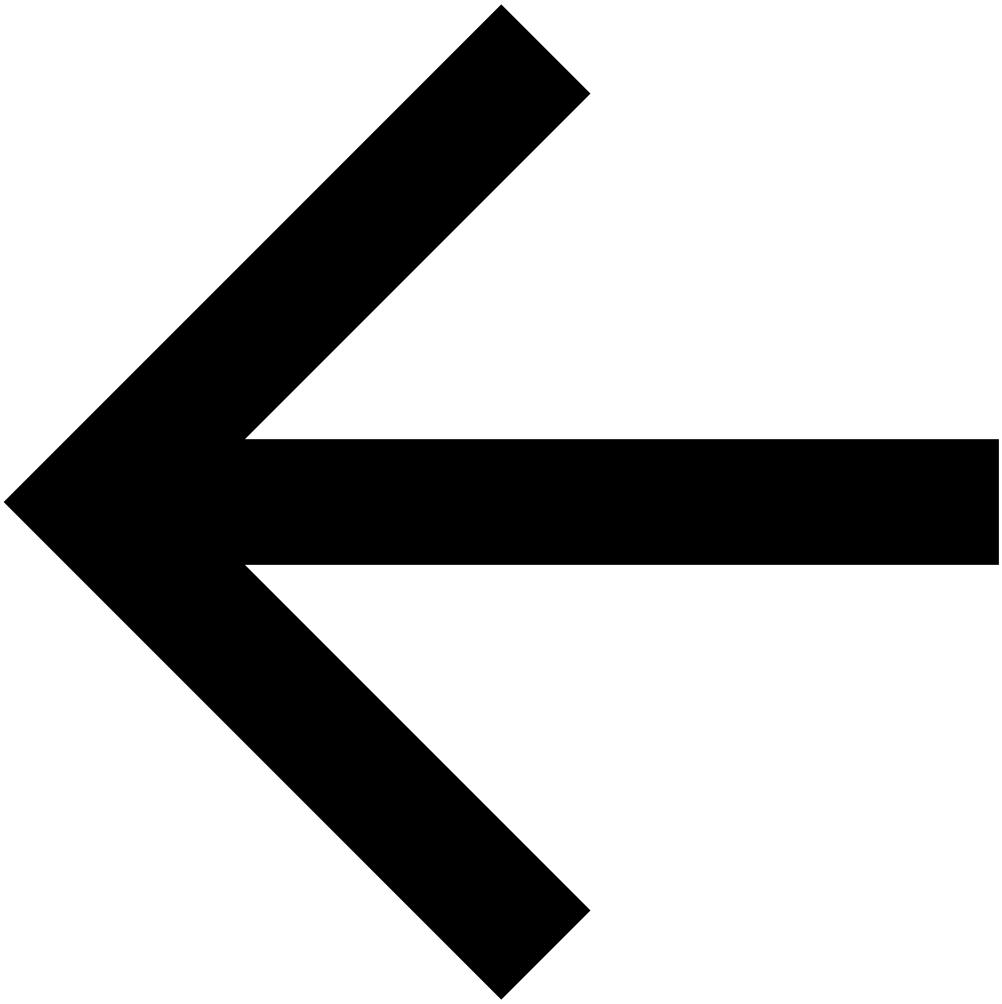
Price	float	Price at which order is placed
TriggerPrice	float	Price at which order is triggered, for SL-M, SL-L, CO & BO
TradedPrice	float	Price at which trade of an order is executed
AvgTradedPrice	float	Average trade price of an order (this will be different from `Traded Price` in case of partial execution)
AlgoOrdNo	float	Entry leg order number to track Target and Stop Loss leg (in case of `BO` and `CO`)
OffMktFlag	string	`1` in case of AMO order else `0`
OrderDateTime	string	Time at which the order is received by Dhan
ExchOrderTime	string	Time at which order is placed on Exchange
LastUpdatedTime	string	Last update time of any order modification or trade
Remarks	string	Additional remarks sent along while placing order
MktType	string	`NL` for Normal Market `AU`, `A1` and `A2`for Auction Market
ReasonDescription	string	Order rejection reason
LegNo	int	`1` for Entry Leg `2` for Stop Loss Leg `3` for Target Leg
Instrument	string	Instrument in which order is placed - here
Symbol	string	Symbol in which order is placed - Refer here
ProductName	string	Product type of the order placed - here
Status	enum string	Last updated status of the order `TRANSIT` `PENDING` `REJECTED` `CANCELLED` `TRADED` `EXPIRED`
LotSize	int	Lot Size in case of Derivatives
StrikePrice	float	Strike Price in which order is placed in Option contract
ExpiryDate	string	Expiry Date of the contract in which order is placed
OptType	string	`CE` or `PE` in case of Option contract
DisplayName	string	Name of instrument in which order is placed - Refer here
Isin	string	ISIN of the instrument in which order is placed
Series	string	Exchange series of the instrument
GoodTillDaysDate	string	Order validity in case of Forever Order
RefLtp	float	LTP at time of order update
TickSize	float	Tick size of the instrument
AlgoId	string	Exchange ID for special order types
Multiplier	int	In case of commodity and currency contracts
CorrelationId	string	The user/partner generated id for tracking back
Remarks	string	`Super Order` if the order is part of super order

Note: For description of enum values, refer Annexure



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☐ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - Postback
  - Live Order Update
- ☒ Data APIs
  Data APIs
  - ☐ Market Quote Market Quote

Table of contents
- Ticker Data
- OHLC Data
- Market Depth Data
- Live Market Feed
- 20 Market Depth
- Historical Data
- Option Chain
- Annexure
- Instrument List

Table of contents

- Ticker Data
- OHLC Data
- Market Depth Data

Market Quote

This API gives you snapshots of multiple instruments at once. You can fetch LTP, Quote or Market Depth of instruments via API which sends real time data at the time of API request.

POST /marketfeed/ltp	Get ticker data of instruments
POST /marketfeed/ohlc	Get OHLC data of instruments
POST /marketfeed/quote	Get market depth data of instruments

Info

You can fetch upto 1000 instruments in single API request with rate limit of 1 request per second.

Ticker Data

Retrieve LTP for list of instruments with single API request

```
curl --request POST \
--url https://api.dhan.co/v2/marketfeed/ltp \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'access-token: JWT' \
--header 'client-id: 1000000001' \
--data '{}'
```

Header

Header	Description

access-token *required*	Access Token generated via Dhan
client-id *required*	User specific identification generated by Dhan

Request Structure

```
  {
"NSE_EQ":[11536],
"NSE_FNO":[49081,49082]
}
```

Parameters

Field	Field Type	Description
Exchange Segment ENUM *required*	array	Security ID - can be found here

Response Structure

```
{
    "data": {
        "NSE_EQ": {
            "11536": {
                "last_price": 4520
            }
        },
        "NSE_FNO": {
            "49081": {
                "last_price": 368.15
            },
            "49082": {
                "last_price": 694.35
            }
        }
    },
    "status": "success"
}
```

Parameters

Field	Type	Description
last_price	float	LTP of the Instrument

OHLC Data

Retrieve the Open, High, Low and Close price along with LTP for specified list of instruments.

```
  curl --request POST \
--url https://api.dhan.co/v2/marketfeed/ohlc \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'access-token: JWT' \
--header 'client-id: 1000000001' \
--data '{}'
```

Header

Header	Description
access-token *required*	Access Token generated via Dhan
client-id *required*	User specific identification generated by Dhan

Request Structure

```
  {
"NSE_EQ":[11536],
"NSE_FNO":[49081,49082]
}
```

Parameters

Field	Field Type	Description
Exchange Segment ENUM *required*	array	Security ID - can be found here

Response Structure


```
{
    "data": {
        "NSE_EQ": {
            "11536": {
                "last_price": 4525.55,
                "ohlc": {
                    "open": 4521.45,
                    "close": 4507.85,
                    "high": 4530,
                    "low": 4500
                }
            }
        },
        "NSE_FNO": {
            "49081": {
                "last_price": 368.15,
                "ohlc": {
                    "open": 0,
                    "close": 368.15,
                    "high": 0,
                    "low": 0
                }
            },
            "49082": {
                "last_price": 694.35,
                "ohlc": {
                    "open": 0,
                    "close": 694.35,
                    "high": 0,
                    "low": 0
                }
            }
        }
    },
    "status": "success"
}
```


Parameters

Field	Type	Description
last_price	float	LTP of the Instrument
ohlc.open	float	Market opening price of the day
ohlc.close	float	Market closing price of the day
ohlc.high	float	Day High price
ohlc.low	float	Day Low price

Market Depth Data

Retrieve full details including market depth, OHLC data, Open Interest and Volume along with LTP for specified instruments.

```
  curl --request POST \
--url https://api.dhan.co/v2/marketfeed/quote \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'access-token: JWT' \
--header 'client-id: 1000000001' \
--data '{}'
```

Header

Header	Description
access-token *required*	Access Token generated via Dhan

client-id *required*	User specific identification generated by Dhan

Request Structure

```
  {
    "NSE_FNO":[49081]
}
```

Parameters

◉
cURL

Field	Field Type	Description
Exchange Segment ENUM *required*	array	Security ID - can be found here

Response Structure

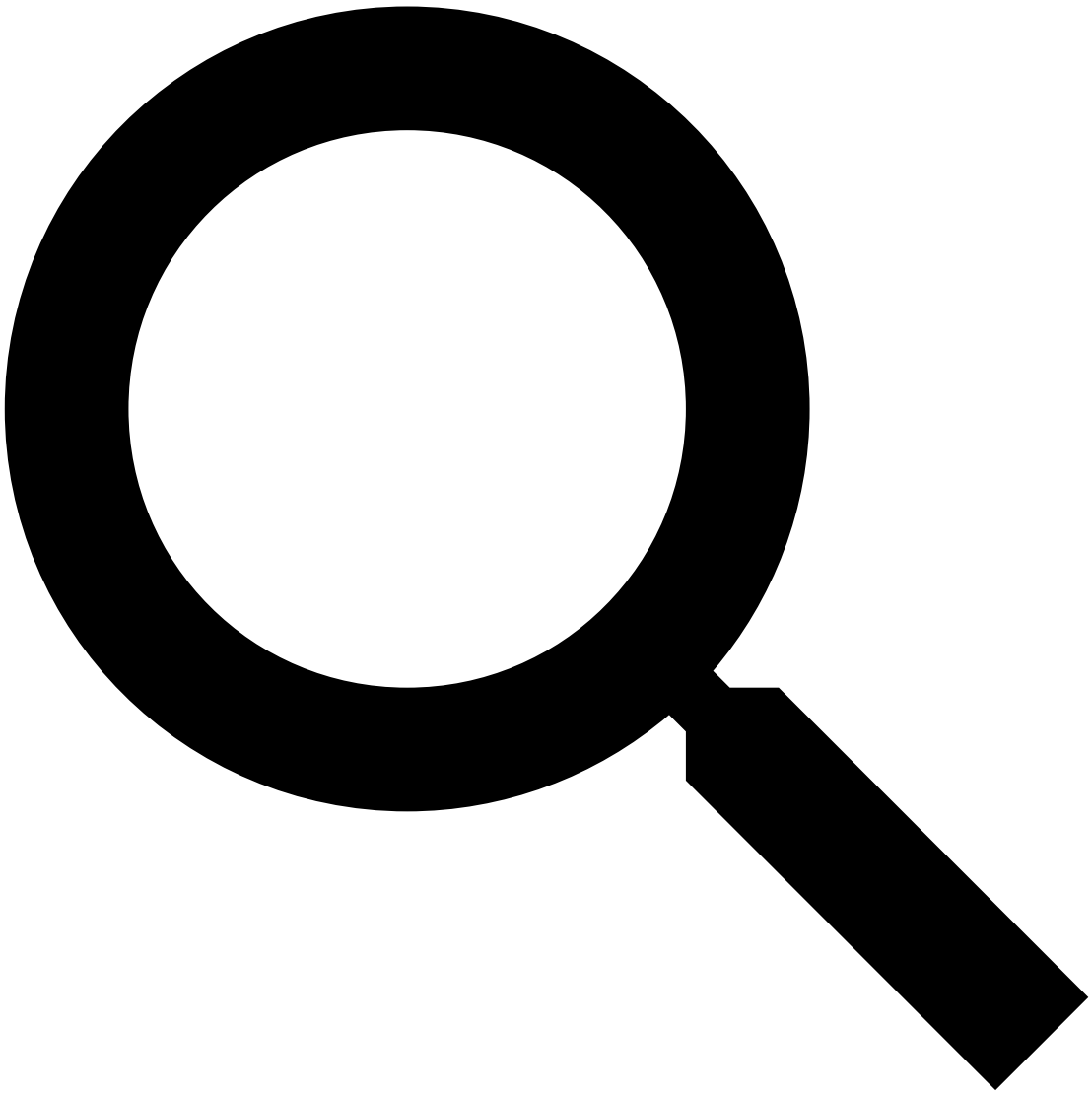

```
{
    "data": {
        "NSE_FNO": {
            "49081": {
                "average_price": 0,
                "buy_quantity": 1825,
                "depth": {
                    "buy": [
                        {
                            "quantity": 1800,
                            "orders": 1,
                            "price": 77
                        },
                        {
                            "quantity": 25,
                            "orders": 1,
                            "price": 50
                        },
                        {
                            "quantity": 0,
                            "orders": 0,
                            "price": 0
                        },
                        {
                            "quantity": 0,
                            "orders": 0,
                            "price": 0
                        },
                        {
                            "quantity": 0,
                            "orders": 0,
                            "price": 0
                        }
                    ],
                    "sell": [
                        {
                            "quantity": 0,
                            "orders": 0,
                            "price": 0
                        },
                        {
                            "quantity": 0,
                            "orders": 0,
                            "price": 0
                        },
                        {
                            "quantity": 0,
                            "orders": 0,
                            "price": 0
                        },
                        {
                            "quantity": 0,
                            "orders": 0,
                            "price": 0
                        },
                        {
                            "quantity": 0,
                            "orders": 0,
```

```
                        "price": 0
                    }
                ]
            },
            "last_price": 368.15,
            "last_quantity": 0,
            "last_trade_time": "01/01/1980 00:00:00",
            "lower_circuit_limit": 48.25,
            "net_change": 0,
            "ohlc": {
                "open": 0,
                "close": 368.15,
                "high": 0,
                "low": 0
            },
            "oi": 0,
            "oi_day_high": 0,
            "oi_day_low": 0,
            "sell_quantity": 0,
            "upper_circuit_limit": 510.85,
            "volume": 0
        }
    }
},
"status": "success"
}
```

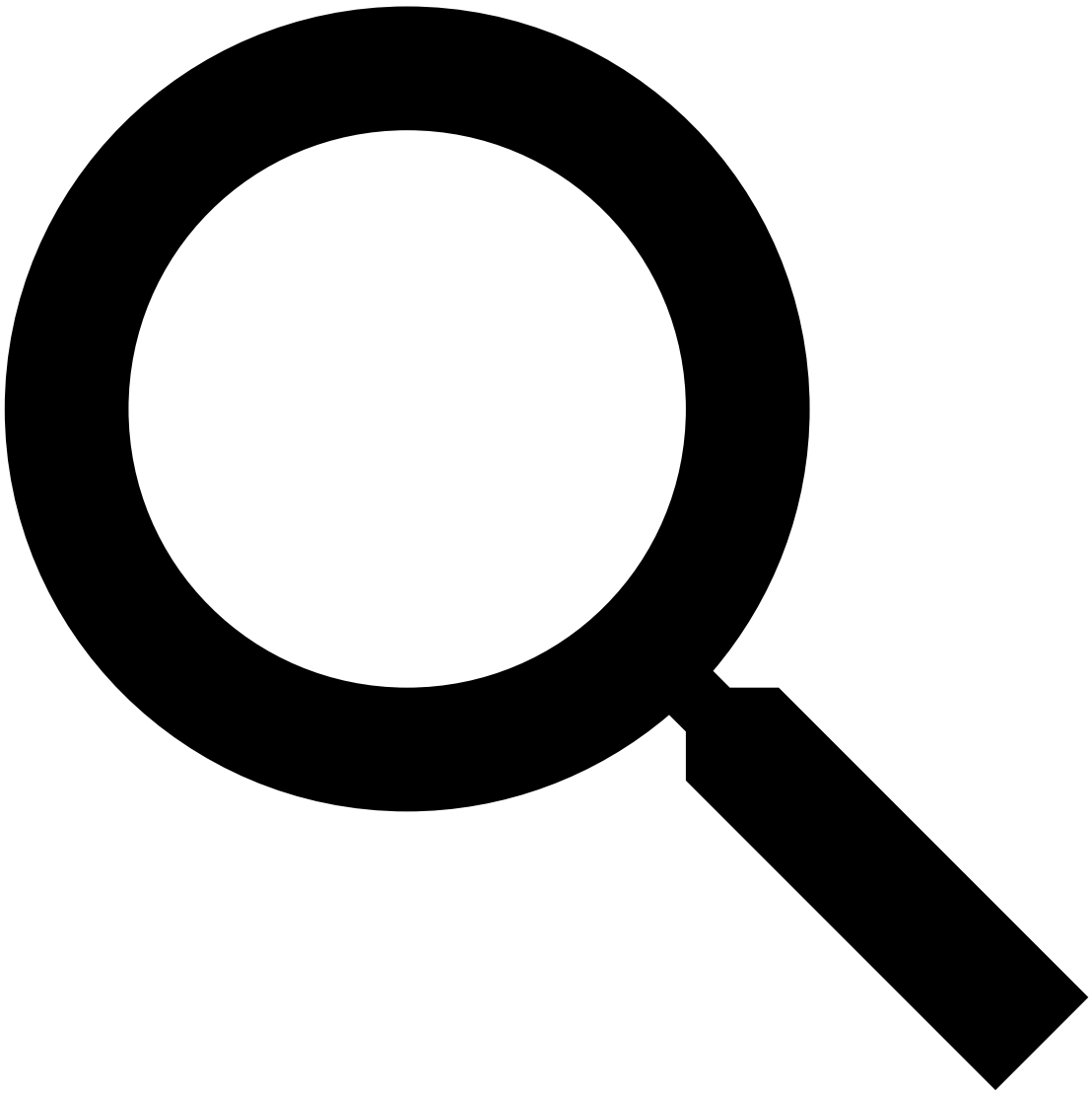

Parameters

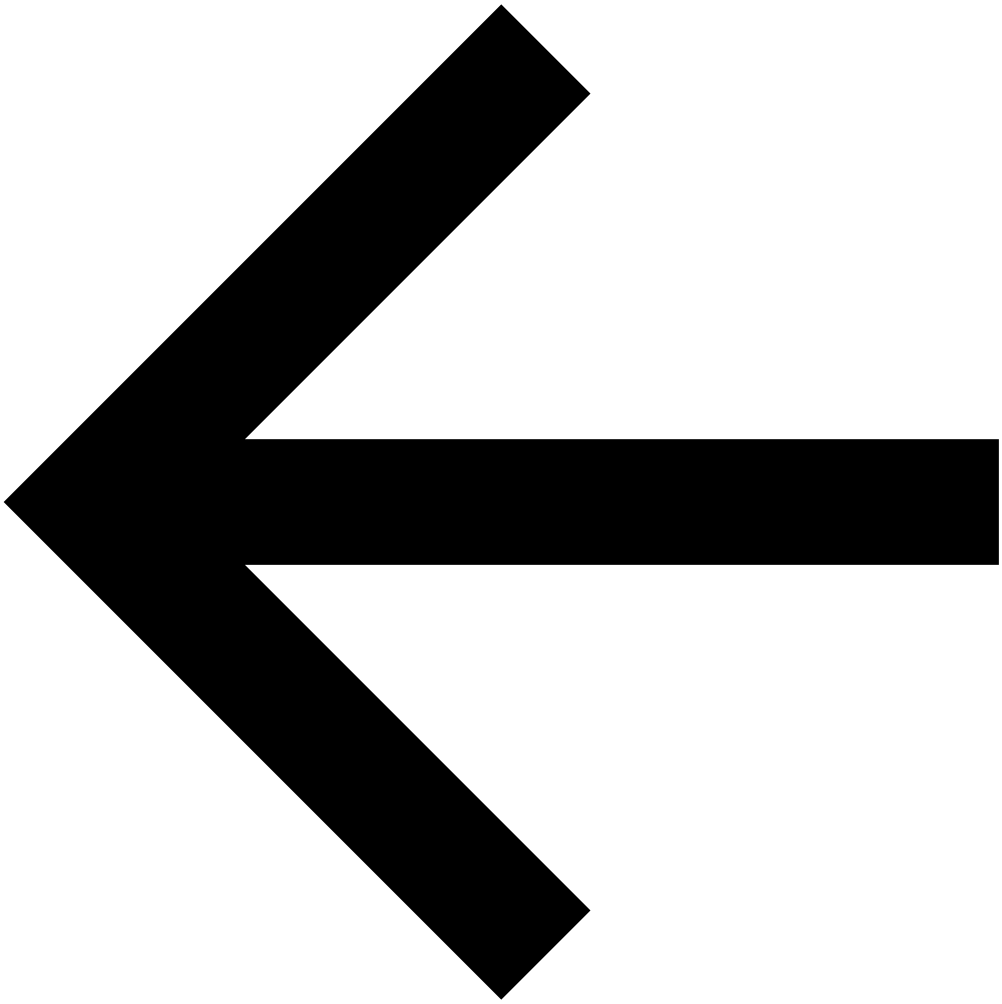
Field	Type	Description
average_price	float	Volume weighted average price of the day
buy_quantity	int	Total buy order quantity pending at the exchange
sell_quantity	int	Total sell order quantity pending at the exchange
depth.buy.quantity	int	Number of quantity at this price depth
depth.buy.orders	int	Number of open BUY orders at this price depth
depth.buy.price	float	Price at which the BUY depth stands
depth.sell.quantity	int	Number of quantity at this price depth
depth.sell.orders	int	Number of open SELL orders at this price depth
depth.sell.price	float	Price at which the SELL depth stands
last_price	float	Last traded price
last_quantity	int	Last traded quantity
last_trade_time	string	Last traded quantity
lower_circuit_limit	float	Current lower circuit limit
upper_circuit_limit	float	Current upper circuit limit
net_change	float	Absolute change in LTP from previous day closing price
volume	int	Total traded volume for the day
oi	int	Open Interest in the contract (for Derivatives)
oi_day_high	int	Highest Open Interest for the day (only for NSE_FNO)
oi_day_low	int	Lowest Open Interest for the day (only for NSE_FNO)
ohlc.open	float	Market opening price of the day
ohlc.close	float	Market closing price of the day
ohlc.high	float	Day High price
ohlc.low	float	Day Low price

Note: For description of enum values, refer Annexure



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☐ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - Postback
  - Live Order Update
- ☒ Data APIs
  Data APIs
  - Market Quote

- Live Market Feed
- 20 Market Depth
- Historical Data
- ☐ Option Chain Option Chain
  Table of contents
  - Option Chain
  - Expiry List
- Annexure
- Instrument List

Table of contents

- Option Chain
- Expiry List

Option Chain

This API gives entire Option Chain of any Option Instrument, across exchanges and segments - for NSE, BSE and MCX traded options. With Option Chain, you get OI, greeks, volume, top bid/ask and price data of all strikes of a particular underlying.

POST /optionchain	Get Option Chain of any instrument
POST /optionchain/expirylist	Expiry List for Options of Underlying

Info

You can call Option Chain API once every 3 seconds. This will give you latest updated data.

Option Chain

Retrieve real-time Option Chain across exchanges for all underlying. You can fetch Open Interest (OI), Greeks, Volume, Last Traded Price, Best Bid/Ask and Implied Volatility (IV) across all strikes for any underlying.

```
curl --location 'https://api.dhan.co/v2/optionchain' \
--header 'access-token: JWT' \
--header 'client-id: ClientId' \
--header 'Content-Type: application/json' \
--data '{}'
```

Header

Header	Description
access-token	Access Token generated via Dhan

required

client-id *required*	User specific identification generated by Dhan

Request Structure

```
  {
"UnderlyingScrip":13,
"UnderlyingSeg":"IDX_I",
"Expiry":"2024-10-31"
}
```

Parameters

Field	Field Type	Description
UnderlyingScri *required*	int	Security ID of Underlying Instrument - can be found here
UnderlyingSeg	enum string	Exchange & segment of underlying for which data is to be fetched - here
Expiry	string	Expiry Date of Option, for which Option Chain is requested. List of active expiries can be fetched from here

Response Structure


```
{
    "data": {
        "last_price": 24964.25,
        "oc": {
            .
            .
            "25000.000000": {
                "ce": {
                    "greeks": {
                        "delta": 0.52546,
                        "theta": -12.88756,
                        "gamma": 0.00136,
                        "vega": 12.98931
                    },
                    "implied_volatility": 8.945204889199001,
                    "last_price": 125.05,
                    "oi": 5962675,
                    "previous_close_price": 190.45,
                    "previous_oi": 3939375,
                    "previous_volume": 831463,
                    "top_ask_price": 124.9,
                    "top_ask_quantity": 1000,
                    "top_bid_price": 124,
                    "top_bid_quantity": 100,
                    "volume": 84202625
                },
                "pe": {
                    "greeks": {
                        "delta": -0.48099,
                        "theta": -10.56587,
                        "gamma": 0.00092,
                        "vega": 13.00105
                    },
                    "implied_volatility": 13.321804909313869,
                    "last_price": 165,
                    "oi": 5059700,
                    "previous_close_price": 153.6,
                    "previous_oi": 4667700,
                    "previous_volume": 1047989,
                    "top_ask_price": 165,
                    "top_ask_quantity": 375,
                    "top_bid_price": 164.05,
                    "top_bid_quantity": 50,
                    "volume": 81097175
                }
            }
            .
            .
            .
        }
}
```

}

Parameters

Field	Type	Description
data.last_price	float	LTP of the Underlying
data.oc	array	Option Chain Array - Strike Wise
data.oc.{strike}	array	Strike Price for Underlying
data.oc.{strike}.ce	array	Call Option data of particular strike
data.oc.{strike}.pe	array	Put Option data of particular strike

Call/Put Option Data

Field	Type	Description
greeks.delta	float	Measures the change of option's premium based on every 1 rupee change in underlying
greeks.theta	float	Measures measures how quickly an option's value decreases over time
greeks.gamma	float	Rate of change in an option's delta in relation to the price of the underlying asset
greeks.vega	float	Measures the change of option's premium in response to a 1% change in implied volatility
implied_volatility	float	Value of expected volatility of a stock over the life of the option
last_price	float	Last Traded Price of the Option Instrument
oi	int	Open Interest of the Option Instrument
previous_close_price	float	Previous day close price
previous_oi	int	Previous day Open Interest
previous_volume	int	Previous day volume
top_ask_price	float	Current best ask price available
top_ask_quantity	int	Quantity available at current best ask price
top_bid_price	float	Current best bid price available
top_bid_quantity	int	Quantity available at current best bid price
volume	int	Day volume for Option Instrument

Expiry List

Retrieve dates of all expiries of any underlying, for which Options Instruments are active.

```
curl --request POST \
--url https://api.dhan.co/v2/optionchain/expirylist \
--header 'Content-Type: application/json' \
--header 'access-token: JWT' \
--header 'client-id: 1000000001' \
--data '{}'
```

Header

Header	Description
access-token *required*	Access Token generated via Dhan
client-id *required*	User specific identification generated by Dhan

Request Structure

```
{
"UnderlyingScrip":13,
"UnderlyingSeg":"IDX_I"
}
```

Parameters

Field	Field Type	Description
UnderlyingScri *required*	int	Security ID of Underlying Instrument - can be found here
UnderlyingSeg	enum string	Exchange & segment of underlying for which data is to be fetched - here

Response Structure

```
{
    "data": [
        "2024-10-17",
        "2024-10-24",
        "2024-10-31",
        "2024-11-07",
        "2024-11-14",
        "2024-11-28",
        "2024-12-26",
        "2025-03-27",
        "2025-06-26",
        "2025-09-25",
        "2025-12-24",
        "2026-06-25",
        "2026-12-31",
        "2027-06-24",
        "2027-12-30",
        "2028-06-29",
        "2028-12-28",
        "2029-06-28"
    ],
    "status": "success"
}
```

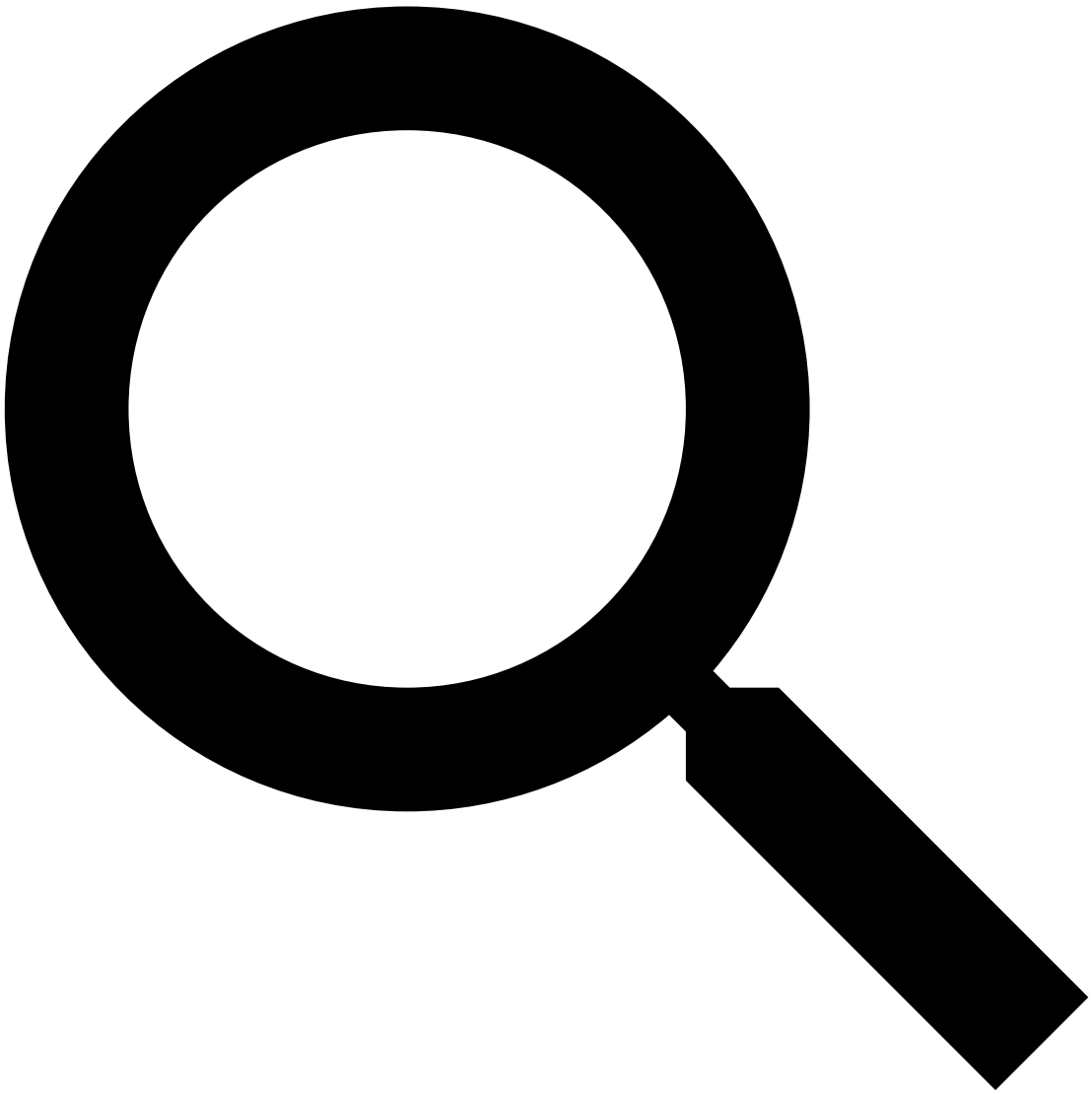

Parameters

Field	Type	Description
data[]	array	All expiry dates of underlying in `YYYY-MM-DD`

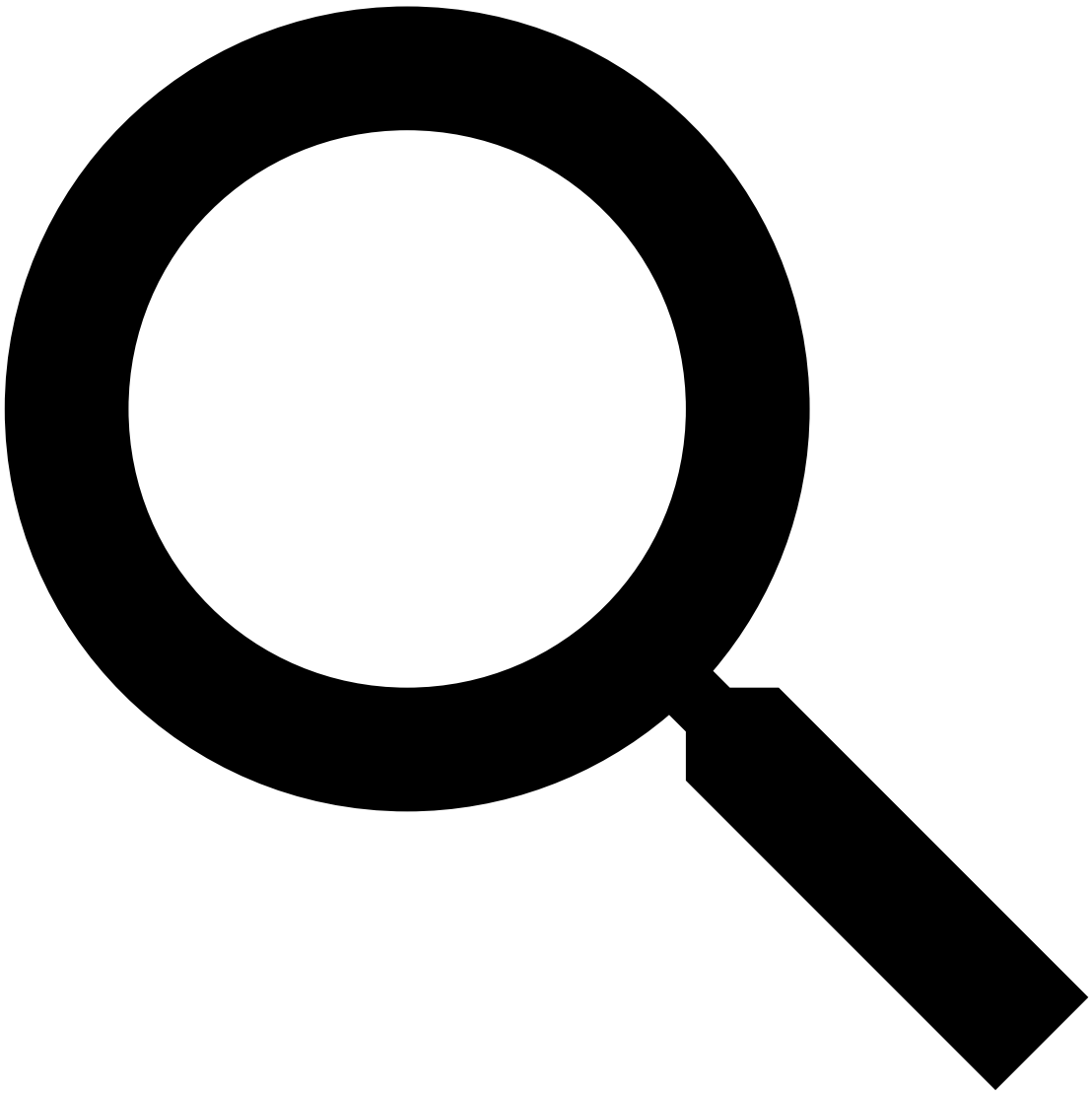
Note

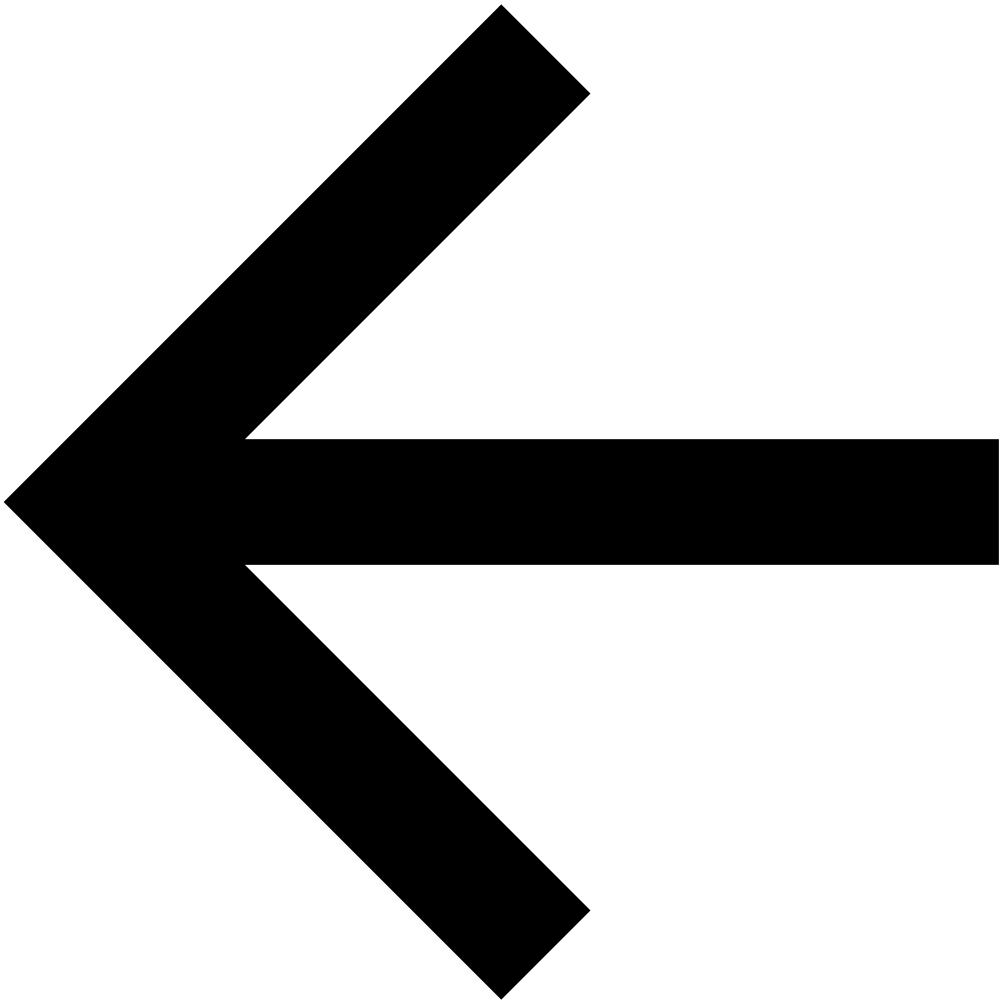
The rate limit applicable for Option Chain API is at 1 request per 3 second. This is because OI data gets updated slow, compared to LTP or other data parameter.

Note: For description of enum values, refer Annexure



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☒ Trading APIs
  Trading APIs
  - ☐ Orders Orders
    Table of contents
    - Order Placement
    - Order Modification
    - Order Cancellation
    - Order Slicing
    - Order Book
    - Get Order by Order Id
    - Get Order by Correlation Id
    - Trade Book
    - Trades of an Order
  - Super Order
  - Forever Order

- Portfolio
- EDIS
- Trader's Control
- Funds
- Statement
- Postback
- Live Order Update

- Data APIs
  - Market Quote
  - Live Market Feed
  - 20 Market Depth
  - Historical Data
  - Option Chain

Table of contents

Data APIs

- Order Placement
- Order Modification
- Order Cancellation
- Order Slicing
- Order Book
- Annexure
- Get Order by Order Id
- Instrument List
- Get Order by Correlation Id
- Trade Book
- Trades of an Order

Orders

The order management API lets you place a new order, cancel or modify the pending order, retrieve the order status, trade status, order book & tradebook.

POST	/orders	Place a new order
PUT	/orders/{order-id}	Modify a pending order
DELETE	/orders/{order-id}	Cancel a pending order
POST	/orders/slicing	Slice order into multiple legs over freeze limit
GET	/orders	Retrieve the list of all orders for the day
GET	/orders/{order-id}	Retrieve the status of an order
GET	/orders/external/{correlation-id}	Retrieve the status of an order by correlation id
GET	/trades	Retrieve the list of all trades for the day
GET	/trades/{order-id}	Retrieve the details of trade by an order id

Order Placement

The order request API lets you place new orders.

```
  curl --request POST \
--url https://api.dhan.co/v2/orders \
--header 'Content-Type: application/json' \
--header 'access-token: JWT' \
--data '{Request JSON}'
```

Request Structure

```
{
  "dhanClientId":"1000000003",
  "correlationId":"123abc678",
  "transactionType":"BUY",
  "exchangeSegment":"NSE_EQ",
  "productType":"INTRADAY",
  "orderType":"MARKET",
  "validity":"DAY",
```

```
    "securityId":"11536",
    "quantity":"5",
    "disclosedQuantity":"",
    "price":"",
    "triggerPrice":"",
    "afterMarketOrder":false,
    "amoTime":"",
    "boProfitValue":"",
    "boStopLossValue": ""
}
```

Parameters

Field	Type	Description
dhanClientId *required*	string	User specific identification generated by Dhan
correlationId	string	The user/partner generated id for tracking back.
transactionType *required*	enum string	The trading side of transaction `BUY` `SELL`
exchangeSegment *required*	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
productType *required*	enum string	Product type `CNC` `INTRADAY` `MARGIN` `MTF` `CO` `BO`
orderType *required*	enum string	Order Type `LIMIT` `MARKET` `STOP_LOSS` `STOP_LOSS_MARKET`
validity *required*	enum string	Validity of Order `DAY` `IOC`
securityId *required*	string	Exchange standard ID for each scrip. Refer **here**
quantity *required*	int	Number of shares for the order
disclosedQuantity	int	Number of shares visible (Keep more than 30% of quantity)
price *required*	float	Price at which order is placed
triggerPrice *conditionally required*	float	Price at which the order is triggered, in case of SL-M & SL-L
afterMarketOrder *conditionally required*	boolean	Flag for orders placed after market hours
amoTime *conditionally required*	enum sting	Timing to pump the after market order `PRE_OPEN` `OPEN` `OPEN_30` `OPEN_60`
boProfitValue *conditionally required*	float	Bracket Order Target Price change
boStopLossValue *conditionally required*	float	Bracket Order Stop Loss Price change

Response Structure

```
{
    "orderId": "112111182198",
    "orderStatus": "PENDING",
}
```

Parameters

Field	Type	Description
orderId	string	Order specific identification generated by Dhan
orderStatus	enum string	Last updated status of the order `TRANSIT` `PENDING` `REJECTED` `CANCELLED` `TRADED` `EXPIRED`

Order Modification

Using this API one can modify pending order in orderbook. The variables that can be modified are price, quantity, order type & validity. The user has to mention the desired value in fields.

```
  curl --request PUT \
--url https://api.dhan.co/v2/orders/{order-id} \
--header 'Content-Type: application/json' \
--header 'access-token: JWT' \
```

```
--data '{Request JSON}'
```

Request Structure

```
{
  "dhanClientId":"1000000009",
  "orderId":"112111182045",
  "orderType":"LIMIT",
  "legName":"",
  "quantity":"40",
  "price":"3345.8",
  "disclosedQuantity":"10",
  "triggerPrice":"",
  "validity":"DAY"
}
```

Parameters

Field	Type	description
dhanClientId *required*	string	User specific identification generated by Dhan
orderId *required*	string	Order specific identification generated by Dhan
orderType *required*	enum string	Order Type `LIMIT MARKET STOP_LOSS STOP_LOSS_MARKET`
legName *conditionally required*	enum string	In case of BO & CO, which leg is modified `ENTRY_LEG TARGET_LEG STOP_LOSS_LEG`
quantity *conditionally required*	int	Quantity to be modified
price *conditionally required*	float	Price to be modified
disclosedQuantity	int	Number of shares visible (if opting keep >30% of quantity)
triggerPrice *conditionally required*	float	Price at which the order is triggered, in case of SL-M & SL-L
validity *required*	enum string	Validity of Order `DAY IOC`

Response Structure

```
{
    "orderId": "112111182045",
    "orderStatus": "TRANSIT"
}
```

Parameters

Field	Type	Description
orderId	string	Order specific identification generated by Dhan
orderStatus	enum string	Last updated status of the order `TRANSIT PENDING REJECTED CANCELLED TRADED EXPIRED`

Order Cancellation

Users can cancel a pending order in the orderbook using the order id of an order. There is no body for request and response for this call. On successful completion of request ‘202 Accepted’ response status code will appear.

```
curl --request DELETE \
--url https://api.dhan.co/v2/orders/{order-id} \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Request Structure

No Body

Response Structure

```
{
"orderId": "112111182045",
"orderStatus": "CANCELLED"
}
```

Parameters

Field	Type	Description
orderId	string	Order specific identification generated by Dhan
orderStatus	enum string	Last updated status of the order `TRANSIT PENDING REJECTED CANCELLED TRADED EXPIRED`

Order Slicing

This API helps you slice your order request into multiple orders to allow you to place over freeze limit quantity for F&O instruments.

◉○
cURLPython

```
curl --request POST \
--url https://api.dhan.co/v2/orders/slicing \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
--data '{Request JSON}'
```

```
dhan.place_slice_order(data)
```

Request Structure

```
    {
      "dhanClientId":"1000000003",
      "correlationId":"123abc678",
      "transactionType":"BUY",
      "exchangeSegment":"NSE_EQ",
      "productType":"INTRADAY",
      "orderType":"MARKET",
      "validity":"DAY",
      "securityId":"11536",
      "quantity":"5",
      "disclosedQuantity":"",
      "price":"",
      "triggerPrice":"",
      "afterMarketOrder":false,
      "amoTime":"",
      "boProfitValue":"",
      "boStopLossValue": ""
  }
```

Parameters

Field	Type	Description	
dhanClientId *required*	string	User specific identification generated by Dhan	
correlationId	string	The user/partner generated id for tracking back.	
transactionType *required*	enum string	The trading side of transaction `BUY SELL`	
exchangeSegment *required*	enum string	Exchange & Segment	Exchange Segment of instrument to be subscribed as found in Annexure
productType *required*	enum string	Product type `CNC INTRADAY MARGIN MTF CO BO`	
orderType *required*	enum string	Order Type `LIMIT MARKET STOP_LOSS STOP_LOSS_MARKET`	
validity *required*	enum string	Validity of Order `DAY IOC`	
securityId *required*	string	Exchange standard ID for each scrip. Refer **here**	
quantity	int	Number of shares for the order	

required		
disclosedQuantity	int	Number of shares visible (Keep more than 30% of quantity)
price *required*	float	Price at which order is placed
triggerPrice *conditionally required*	float	Price at which the order is triggered, in case of SL-M & SL-L
afterMarketOrder *conditionally required*	boolean	Flag for orders placed after market hours
amoTime *conditionally required*	enum sting	Timing to pump the after market order `PRE_OPEN OPEN OPEN_30 OPEN_60`
boProfitValue *conditionally required*	float	Bracket Order Target Price change
boStopLossValue *conditionally required*	float	Bracket Order Stop Loss Price change

Response Structure


```
[
    {
        "orderId": "552209237100",
        "orderStatus": "TRANSIT"
    },
    {
        "orderId": "552209237100",
        "orderStatus": "TRANSIT"
    }
]
```


Parameters

Field	Type	Description
orderId	string	Order specific identification generated by Dhan
orderStatus	string	Order Type `TRANSIT PENDING REJECTED CANCELLED TRADED EXPIRED CONFIRM`

Order Book

This API lets you retrieve an array of all orders requested in a day with their last updated status.

```
  curl --request GET \
--url https://api.dhan.co/v2/orders \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Request Structure

No Body

Response Structure


```
[
    {
        "dhanClientId": "1000000003",
        "orderId": "112111182198",
        "correlationId":"123abc678",
        "orderStatus": "PENDING",
        "transactionType": "BUY",
        "exchangeSegment": "NSE_EQ",
        "productType": "INTRADAY",
        "orderType": "MARKET",
        "validity": "DAY",
        "tradingSymbol": "",
```

```
        "securityId": "11536",
        "quantity": 5,
        "disclosedQuantity": 0,
        "price": 0.0,
        "triggerPrice": 0.0,
        "afterMarketOrder": false,
        "boProfitValue": 0.0,
        "boStopLossValue": 0.0,
        "legName": ,
        "createTime": "2021-11-24 13:33:03",
        "updateTime": "2021-11-24 13:33:03",
        "exchangeTime": "2021-11-24 13:33:03",
        "drvExpiryDate": null,
        "drvOptionType": null,
        "drvStrikePrice": 0.0,
        "omsErrorCode": null,
        "omsErrorDescription": null,
        "algoId": "string"
        "remainingQuantity": 5,
        "averageTradedPrice": 0,
        "filledQty": 0
    }
]
```

Parameters

Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
orderId	string	Order specific identification generated by Dhan
correlationId	string	The user/partner generated id for tracking back
orderStatus	enum string	Last updated status of the order `TRANSIT PENDING REJECTED CANCELLED PART_TRADED TRADED EXPIRED`
transactionType	enum string	The trading side of transaction `BUY SELL`
exchangeSegment	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
productType	enum string	Product type of trade `CNC INTRADAY MARGIN MTF CO BO`
orderType	enum string	Order Type `LIMIT MARKET STOP_LOSS STOP_LOSS_MARKET`
validity	enum string	Validity of Order `DAY IOC`
tradingSymbol	string	Refer Trading Symbol in Tables
securityId	string	Exchange standard ID for each scrip. Refer **here**
quantity	int	Number of shares for the order
disclosedQuantity	int	Number of shares visible
price	float	Price at which order is placed
triggerPrice	float	Price at which order is triggered, for SL-M, SL-L, CO & BO
afterMarketOrder	boolean	The order placed is AMO ?
boProfitValue	float	Bracket Order Target Price change
boStopLossValue	float	Bracket Order Stop Loss Price change
legName	enum string	Leg identification in case of BO `ENTRY_LEG TARGET_LEG STOP_LOSS_LEG`
createTime	string	Time at which the order is created
updateTime	string	Time at which the last activity happened
exchangeTime	string	Time at which order reached at exchange
drvExpiryDate	int	For F&O, expiry date of contract
drvOptionType	enum string	Type of Option `CALL PUT`
drvStrikePrice	float	For Options, Strike Price
omsErrorCode	string	Error code in case the order is rejected or failed
omsErrorDescription	string	Description of error in case the order is rejected or failed
algoId	string	Exchange Algo ID for Dhan
remainingQuantity	integer	Quantity pending at the exchange to be traded (quantity - filledQty)
averageTradedPrice	integer	Average price at which order is traded
filledQty	integer	Quantity of order traded on Exchange

Get Order by Order Id

Users can retrieve the details and status of an order from the orderbook placed during the day.

```
curl --request GET \
--url https://api.dhan.co/v2/orders/{order-id} \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Request Structure

No Body

Response Structure

```
{
    "dhanClientId": "1000000003",
    "orderId": "112111182198",
    "correlationId":"123abc678",
    "orderStatus": "PENDING",
    "transactionType": "BUY",
    "exchangeSegment": "NSE_EQ",
    "productType": "INTRADAY",
    "orderType": "MARKET",
    "validity": "DAY",
    "tradingSymbol": "",
    "securityId": "11536",
    "quantity": 5,
    "disclosedQuantity": 0,
    "price": 0.0,
    "triggerPrice": 0.0,
    "afterMarketOrder": false,
    "boProfitValue": 0.0,
    "boStopLossValue": 0.0,
    "legName": ,
    "createTime": "2021-11-24 13:33:03",
    "updateTime": "2021-11-24 13:33:03",
    "exchangeTime": "2021-11-24 13:33:03",
    "drvExpiryDate": null,
    "drvOptionType": null,
    "drvStrikePrice": 0.0,
    "omsErrorCode": null,
    "omsErrorDescription": null,
    "algoId": "string"
    "remainingQuantity": 5,
    "averageTradedPrice": 0,
    "filledQty": 0
}
```

Parameters

Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
orderId	string	Order specific identification generated by Dhan
correlationId	string	The user/partner generated id for tracking back
orderStatus	enum string	Last updated status of the order `TRANSIT PENDING REJECTED CANCELLED PART_TRADED TRADED EXPIRED`
transactionType	enum string	The trading side of transaction `BUY SELL`
exchangeSegment	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
productType	enum string	Product type of trade `CNC INTRADAY MARGIN MTF CO BO`
orderType	enum string	Order Type `LIMIT MARKET STOP_LOSS STOP_LOSS_MARKET`
validity	enum string	Validity of Order `DAY IOC`
tradingSymbol	string	Refer Trading Symbol in Tables
securityId	string	Exchange standard ID for each scrip. Refer **here**
quantity	int	Number of shares for the order
disclosedQuantity	int	Number of shares visible
price	float	Price at which order is placed
triggerPrice	float	Price at which order is triggered, for SL-M, SL-L, CO & BO

afterMarketOrder	boolean	The order placed is AMO ?
boProfitValue	float	Bracket Order Target Price change
boStopLossValue	float	Bracket Order Stop Loss Price change
legName	enum string	Leg identification in case of BO `ENTRY_LEG TARGET_LEG STOP_LOSS_LEG`
createTime	string	Time at which the order is created
updateTime	string	Time at which the last activity happened
exchangeTime	string	Time at which order reached at exchange
drvExpiryDate	int	For F&O, expiry date of contract
drvOptionType	enum string	Type of Option `CALL PUT`
drvStrikePrice	float	For Options, Strike Price
omsErrorCode	string	Error code in case the order is rejected or failed
omsErrorDescription	string	Description of error in case the order is rejected or failed
algoId	string	Exchange Algo ID for Dhan
remainingQuantity	integer	Quantity pending at the exchange to be traded (quantity - filledQty)
averageTradedPrice	integer	Average price at which order is traded
filledQty	integer	Quantity of order traded on Exchange

Get Order by Correlation Id

In case the user has missed order id due to unforeseen reason, this API retrieves the order status using a tag called correlation id specified by users themselve.

```
  curl --request GET \
--url https://api.dhan.co/v2/orders/external/{correlation-id} \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Request Structure

No Body

Response Structure

```
{
    "dhanClientId": "1000000003",
    "orderId": "112111182198",
    "correlationId":"123abc678",
    "orderStatus": "PENDING",
    "transactionType": "BUY",
    "exchangeSegment": "NSE_EQ",
    "productType": "INTRADAY",
    "orderType": "MARKET",
    "validity": "DAY",
    "tradingSymbol": "",
    "securityId": "11536",
    "quantity": 5,
    "disclosedQuantity": 0,
    "price": 0.0,
    "triggerPrice": 0.0,
    "afterMarketOrder": false,
    "boProfitValue": 0.0,
    "boStopLossValue": 0.0,
    "legName": ,
    "createTime": "2021-11-24 13:33:03",
    "updateTime": "2021-11-24 13:33:03",
    "exchangeTime": "2021-11-24 13:33:03",
    "drvExpiryDate": null,
    "drvOptionType": null,
    "drvStrikePrice": 0.0,
    "omsErrorCode": null,
    "omsErrorDescription": null,
    "algoId": "string"
    "remainingQuantity": 5,
    "averageTradedPrice": 0,
    "filledQty": 0
}
```

Parameters

Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
orderId	string	Order specific identification generated by Dhan
correlationId	string	The user/partner generated id for tracking back
orderStatus	enum string	Last updated status of the order `TRANSIT PENDING REJECTED CANCELLED PART_TRADED TRADED EXPIRED`
transactionType	enum string	The trading side of transaction `BUY SELL`
exchangeSegment	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
productType	enum string	Product type of trade `CNC INTRADAY MARGIN MTF CO BO`
orderType	enum string	Order Type `LIMIT MARKET STOP_LOSS STOP_LOSS_MARKET`
validity	enum string	Validity of Order `DAY IOC`
tradingSymbol	string	Refer Trading Symbol in Tables
securityId	string	Exchange standard ID for each scrip. Refer **here**
quantity	int	Number of shares for the order
disclosedQuantity	int	Number of shares visible
price	float	Price at which order is placed
triggerPrice	float	Price at which order is triggered, for SL-M, SL-L, CO & BO
afterMarketOrder	boolean	The order placed is AMO ?
boProfitValue	float	Bracket Order Target Price change
boStopLossValue	float	Bracket Order Stop Loss Price change
legName	enum string	Leg identification in case of BO `ENTRY_LEG TARGET_LEG STOP_LOSS_LEG`
createTime	string	Time at which the order is created
updateTime	string	Time at which the last activity happened
exchangeTime	string	Time at which order reached at exchange
drvExpiryDate	int	For F&O, expiry date of contract
drvOptionType	enum string	Type of Option `CALL PUT`
drvStrikePrice	float	For Options, Strike Price
omsErrorCode	string	Error code in case the order is rejected or failed
omsErrorDescription	string	Description of error in case the order is rejected or failed
algoId	string	Exchange Algo ID for Dhan
remainingQuantity	integer	Quantity pending at the exchange to be traded (quantity - filledQty)
averageTradedPrice	integer	Average price at which order is traded
filledQty	integer	Quantity of order traded on Exchange

Trade Book

This API lets you retrieve an array of all trades executed in a day.

```
  curl --request GET \
--url https://api.dhan.co/v2/trades \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Request Structure

No Body

Response Structure

```
[
    {
        "dhanClientId": "1000000009",
        "orderId": "112111182045",
        "exchangeOrderId": "15112111182045",
        "exchangeTradeId": "15112111182045",
        "transactionType": "BUY",
        "exchangeSegment": "NSE_EQ",
        "productType": "INTRADAY",
        "orderType": "LIMIT",
        "tradingSymbol": "TCS",
```

```
        "securityId": "11536",
        "tradedQuantity": 40,
        "tradedPrice": 3345.8,
        "createTime": "2021-03-10 11:20:06",
        "updateTime": "2021-11-25 17:35:12"
        "exchangeTime": "2021-11-25 17:35:12",
        "drvExpiryDate": null,
        "drvOptionType": null,
        "drvStrikePrice": 0.0
    }
]
```

Parameters

Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
orderId	string	Order specific identification generated by Dhan
exchangeOrderId	string	Order specific identification generated by exchange
exchangeTradeId	string	Trade specific identification generated by exchange
transactionType	enum string	The trading side of transaction `BUY SELL`
exchangeSegment	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
productType	enum string	Product type of trade `CNC INTRADAY MARGIN MTF CO BO`
orderType	enum string	Order Type `LIMIT MARKET STOP_LOSS STOP_LOSS_MARKET`
tradingSymbol	string	Refer Trading Symbol in Tables
securityId	string	Exchange standard ID for each scrip.Refer **here**
tradedQuantity	int	Number of shares executed
tradedPrice	float	Price at which trade is executed
createTime	string	Time at which the order is created
updateTime	string	Time at which the last activity happened
exchangeTime	string	Time at which order reached at exchange
drvExpiryDate	int	For F&O, expiry date of contract
drvOptionType	enum string	Type of Option `CALL PUT`
drvStrikePrice	float	For Options, Strike Price

Trades of an Order

Users can retrieve the trade details using an order id. Often during partial trades or Bracket/ Cover Orders, traders get confused in reading trade from tradebook.The response of this API will include all the trades generated for a particular order id.

◉○
cURLPython

```
curl --request GET \
--url https://api.dhan.co/v2/trades/{order-id} \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

```
dhan.get_trade_book(order_id)
```

Request Structure

No Body

Response Structure

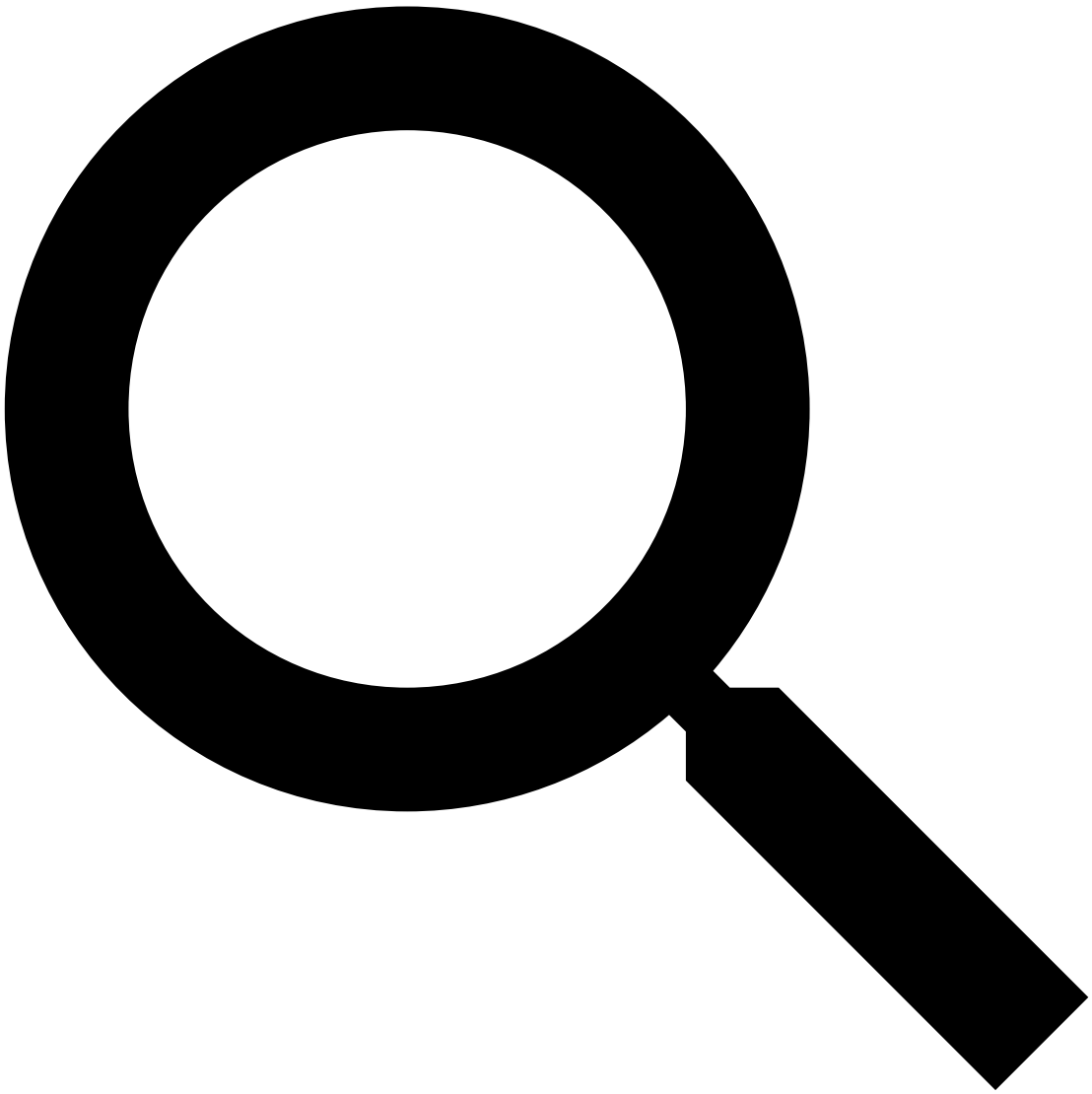
```
{
    "dhanClientId": "1000000009",
    "orderId": "112111182045",
    "exchangeOrderId": "15112111182045",
    "exchangeTradeId": "15112111182045",
    "transactionType": "BUY",
    "exchangeSegment": "NSE_EQ",
    "productType": "INTRADAY",
```

```
    "orderType": "LIMIT",
    "tradingSymbol": "TCS",
    "securityId": "11536",
    "tradedQuantity": 40,
    "tradedPrice": 3345.8,
    "createTime": "2021-03-10 11:20:06",
    "updateTime": "2021-11-25 17:35:12",
    "exchangeTime": "2021-11-25 17:35:12",
    "drvExpiryDate": null,
    "drvOptionType": null,
    "drvStrikePrice": 0.0
}
```

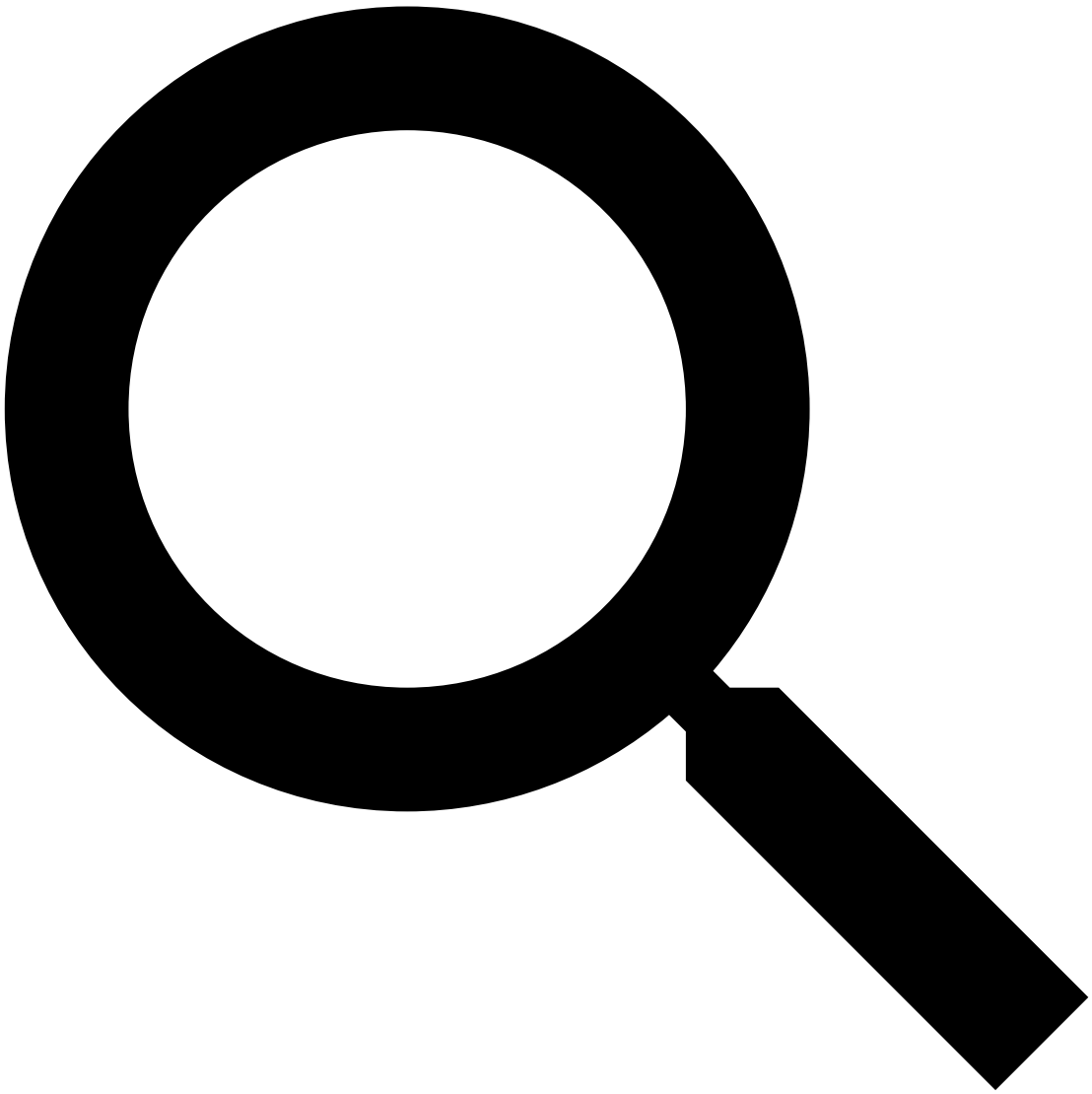

Parameters

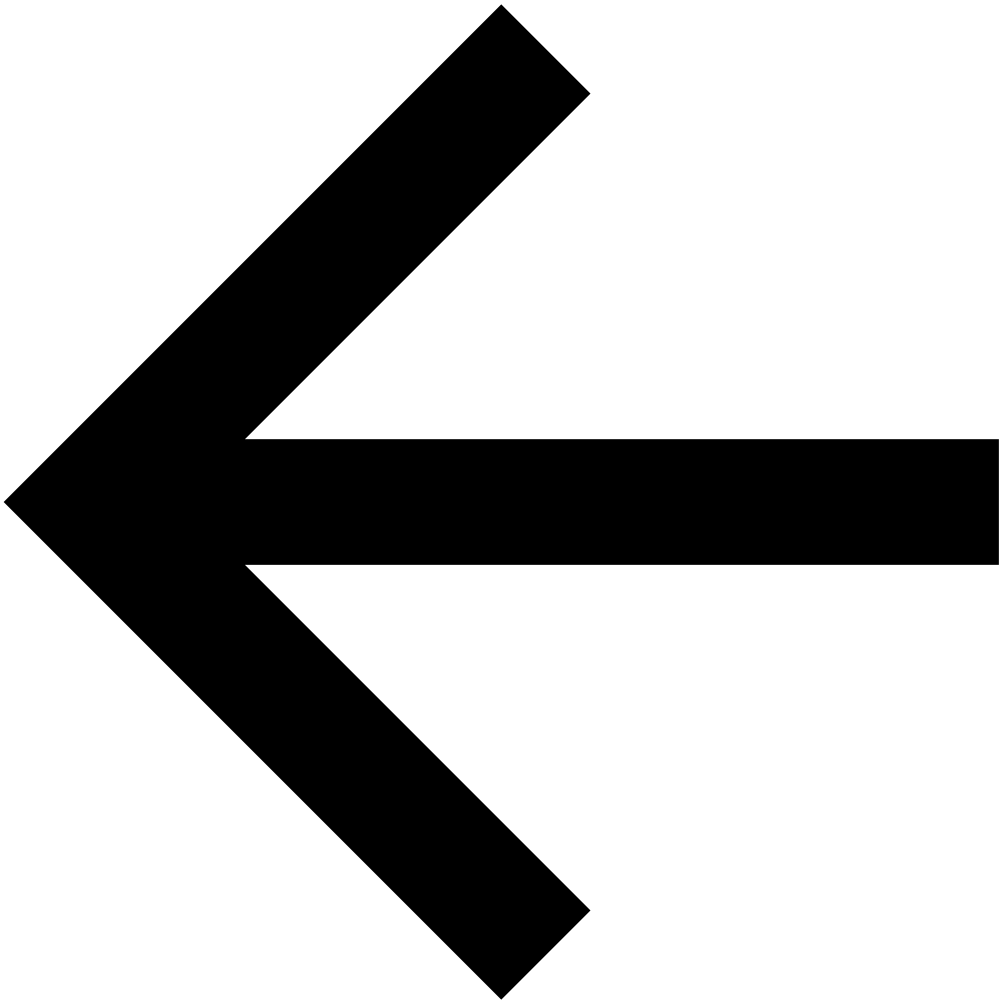
Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
orderId	string	Order specific identification generated by Dhan
exchangeOrderId	string	Order specific identification generated by exchange
exchangeTradeId	string	Trade specific identification generated by exchange
transactionType	enum string	The trading side of transaction `BUY` `SELL`
exchangeSegment	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
productType	enum string	Product type of trade `CNC` `INTRADAY` `MARGIN` `MTF` `CO` `BO`
orderType	enum string	Order Type `LIMIT` `MARKET` `STOP_LOSS` `STOP_LOSS_MARKET`
tradingSymbol	string	Refer Trading Symbol in Tables
securityId	string	Exchange standard ID for each scrip. Refer **here**
tradedQuantity	int	Number of shares executed
tradedPrice	float	Price at which trade is executed
createTime	string	Time at which the order is created
updateTime	string	Time at which the last activity happened
exchangeTime	string	Time at which order reached at exchange
drvExpiryDate	int	For F&O, expiry date of contract
drvOptionType	enum string	Type of Option `CALL` `PUT`
drvStrikePrice	float	For Options, Strike Price

Note: For description of enum values, refer Annexure



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☒ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - ☐ Portfolio Portfolio
    Table of contents
    - Holdings
    - Positions
    - Convert Position
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - Postback

- Live Order Update
- ☐ Data APIs
  Data APIs
  - Market Quote
  - Live Market Feed
  - 20 Market Depth
  - Historical Data
  - Option Chain

Table of contents

- Annexure
- Instrument List
- Holdings
- Positions
- Convert Position

Portfolio

This API lets you retrieve holdings and positions in your portfolio.

GET	/holdings	Retrieve list of holdings in demat account
GET	/positions	Retrieve open positions
POST	/positions/convert	Convert intraday position to delivery or delivery to intraday

Holdings

Users can retrieve all holdings bought/sold in previous trading sessions. All T1 and delivered quantities can be fetched.

```
curl --request GET \
--url https://api.dhan.co/v2/holdings \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Request Structure

No Body

Response Structure

```
[
    {
    "exchange": "ALL",
    "tradingSymbol": "HDFC",
    "securityId": "1330",
    "isin": "INE001A01036",
    "totalQty": 1000,
    "dpQty": 1000,
    "t1Qty": 0,
    "availableQty": 1000,
    "collateralQty": 0,
    "avgCostPrice": 2655.0
    }
```

]

Parameters

Field	Type	Description
exchange	enum string	Exchange
tradingSymbol	string	Refer Trading Symbol at Page No
securityId	string	Exchange standard ID for each scrip. Refer **here**
isin	string	Universal standard ID for each scrip
totalQty	int	Total quantity
dpQty	int	Quantity delivered in demat account
t1Qty	int	Quantity pending delivered in demat account
availableQty	int	Quantity available for transaction
collateralQty	int	Quantity placed as collateral with broker
avgCostPrice	float	Average Buy Price of total quantity

Positions

Users can retrieve a list of all open positions for the day. This includes all F&O carryforward positions as well.

```
  curl --request GET \
--url https://api.dhan.co/v2/positions \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Request Structure

No Body

Response Structure

```
[
    {
    "dhanClientId": "1000000009",
    "tradingSymbol": "TCS",
    "securityId": "11536",
    "positionType": "LONG",
    "exchangeSegment": "NSE_EQ",
    "productType": "CNC",
    "buyAvg": 3345.8,
    "buyQty": 40,
    "costPrice": 3215.0,
    "sellAvg": 0.0,
    "sellQty": 0,
    "netQty": 40,
    "realizedProfit": 0.0,
    "unrealizedProfit": 6122.0,
    "rbiReferenceRate": 1.0,
    "multiplier": 1,
    "carryForwardBuyQty": 0,
    "carryForwardSellQty": 0,
    "carryForwardBuyValue": 0.0,
    "carryForwardSellValue": 0.0,
    "dayBuyQty": 40,
    "daySellQty": 0,
    "dayBuyValue": 133832.0,
    "daySellValue": 0.0,
    "drvExpiryDate": "0001-01-01",
    "drvOptionType": null,
    "drvStrikePrice": 0.0.
    "crossCurrency": false
    }
]
```

Parameters

Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
tradingSymbol	string	Refer Trading Symbol in Tables
securityId	string	Exchange standard id for each scrip. Refer **here**

positionType	enum string	Position Type `LONG SHORT CLOSED`
exchangeSegment	enum string	Exchange & Segment `NSE_EQ NSE_FNO NSE_CURRENCY BSE_EQ BSE_FNO BSE_CURRENCY MCX_COMM`
productType	enum string	Product type `CNC INTRADAY MARGIN MTF CO BO`
buyAvg	float	Average buy price mark to market
buyQty	int	Total quantity bought
costPrice	int	Actual Cost Price
sellAvg	float	Average sell price mark to market
sellQty	int	Total quantities sold
netQty	int	buyQty - sellQty = netQty
realizedProfit	float	Profit or loss booked
unrealizedProfit	float	Profit or loss standing for open position
rbiReferenceRate	float	RBI mandated reference rate for forex
multiplier	int	Multiplying factor for currency F&O
carryForwardBuyQty	int	Carry forward F&O long quantities
carryForwardSellQty	int	Carry forward F&O short quantities
carryForwardBuyValue	float	Carry forward F&O long value
carryForwardSellValue	float	Carry forward F&O short value
dayBuyQty	int	Quantities bought today
daySellQty	int	Quantities sold today
dayBuyValue	float	Value of quantities bought today
daySellValue	float	Value of quantities sold today
drvExpiryDate	int	For F&O, expiry date of contract
drvOptionType	enum string	Type of Option `CALL PUT`
drvStrikePrice	float	For Options, Strike Price
crossCurrency	boolean	Check for non INR currency pair

Convert Position

Users can convert their open position from intraday to delivery or delivery to intraday.

```
curl --request POST \
--url https://api.dhan.co/v2/positions/convert \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header 'access-token: JWT' \
--data '{}'
```

Request Structure

```
{
    "dhanClientId": "1000000009",
    "fromProductType":"INTRADAY",
    "exchangeSegment":"NSE_EQ",
    "positionType":"LONG",
    "securityId":"11536",
    "tradingSymbol":"",
    "convertQty":"40",
    "toProductType":"CNC"
}
```

Parameters

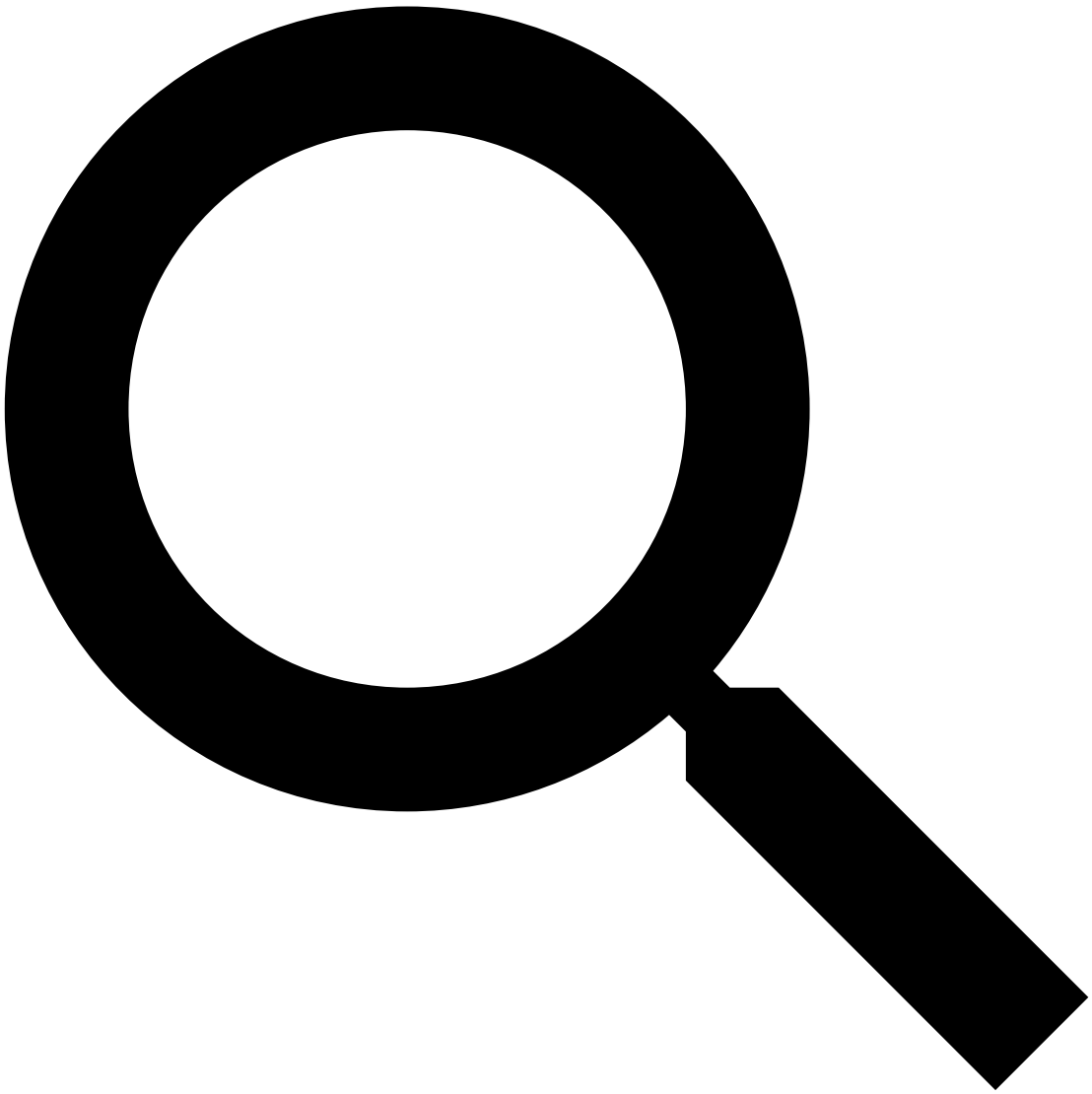
Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
fromProductType	enum string	Refer Trading Symbol in Tables `CNC INTRADAY MARGIN CO BO`
exchangeSegment	enum string	Exchange & segment in which position is created - here
positionType	enum string	Position Type `LONG SHORT CLOSED`
securityId	string	Exchange standard id for each scrip. Refer **here**
tradingSymbol	string	Refer Trading Symbol in Tables
convertQty	int	No of shares modification is desired

toProductType	enum string	Desired product type `CNC` `INTRADAY` `MARGIN` `CO` `BO`

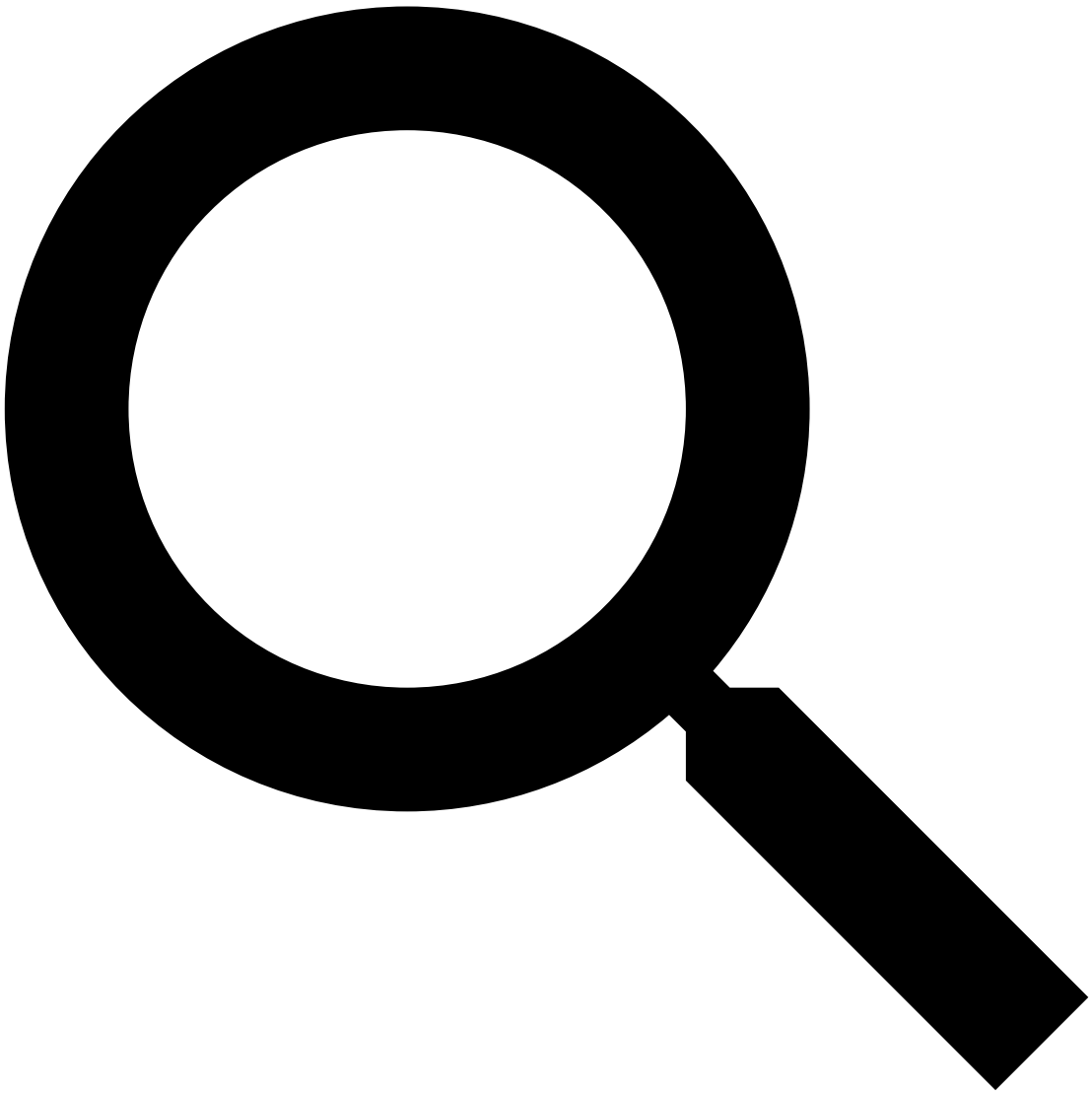
Response Structure

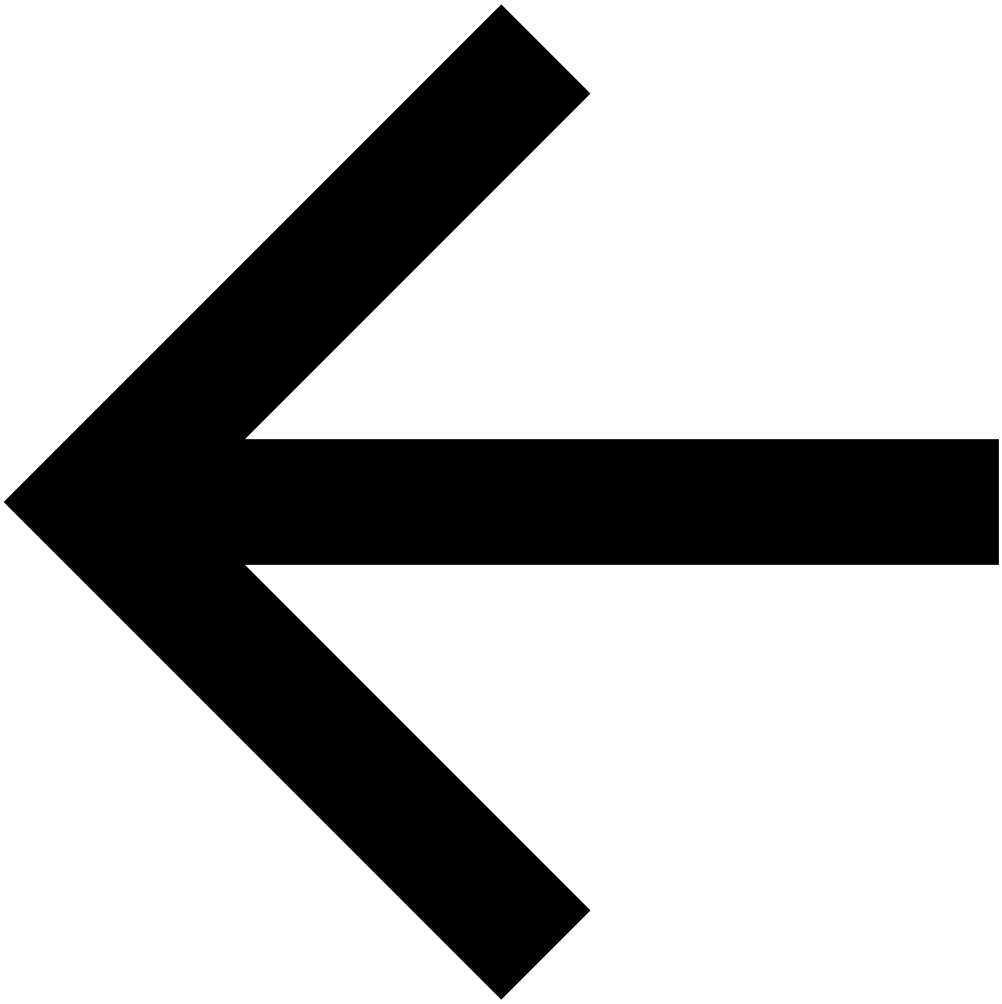
`202 Accepted`

Note: For description of enum values, refer Annexure



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☒ Trading APIs
  Trading APIs
  - Orders
  - Super Order
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement
  - ☐ Postback Postback
    Table of contents
    - Postback Payload
    - Setting up Postback
  - Live Order Update

- ☐ Data APIs
  Data APIs
  - Market Quote
  - Live Market Feed
  - 20 Market Depth
  - Historical Data
  - Option Chain
- Annexure
- Instrument List
- Postback Payload
- Setting up Postback

Table of contents

Postback

Postback API or Webhooks uses a `POST` request with **JSON payload** to the Postback URL. This JSON payload contains order update in case of change in status (`TRANSIT`, `PENDING`, `REJECTED`, `CANCELLED`, `TRADED` or `EXPIRED`) or whenever order is modified or partially filled.

This Postback API is on **access token** level i.e. all trades originating from one particular access token will be sent to that particular Webhook URL. This makes it optimal for individual developers.

Postback Payload

The JSON payload is sent as a raw HTTP POST body in below structure.

Structure


```
{
    "dhanClientId": "1000000003",
    "orderId": "112111182198",
    "correlationId":"123abc678",
    "orderStatus": "PENDING",
    "transactionType": "BUY",
    "exchangeSegment": "NSE_EQ",
    "productType": "INTRADAY",
    "orderType": "MARKET",
    "validity": "DAY",
    "tradingSymbol": "",
    "securityId": "11536",
    "quantity": 5,
    "disclosedQuantity": 0,
    "price": 0.0,
    "triggerPrice": 0.0,
    "afterMarketOrder": false,
    "boProfitValue": 0.0,
    "boStopLossValue": 0.0,
    "legName": ,
    "createTime": "2021-11-24 13:33:03",
    "updateTime": "2021-11-24 13:33:03",
    "exchangeTime": "2021-11-24 13:33:03",
    "drvExpiryDate": null,
    "drvOptionType": null,
    "drvStrikePrice": 0.0,
```

```
    "omsErrorCode": null,
    "omsErrorDescription": null
}
```

Parameters

Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
orderId	string	Order specific identification generated by Dhan
correlationId	string	The user/partner generated id for tracking back
orderStatus	enum string	Last updated status of the order `TRANSIT PENDING REJECTED CANCELLED TRADED EXPIRED`
transactionType	enum string	The trading side of transaction `BUY SELL`
exchangeSegment	enum string	Exchange & Segment `NSE_EQ NSE_FNO NSE_CURRENCY BSE_EQ MCX_COMM`
productType	enum string	Product type of trade `CNC INTRADAY MARGIN MTF CO BO`
orderType	enum string	Order Type `LIMIT MARKET STOP_LOSS STOP_LOSS_MARKET`
validity	enum string	Validity of Order `DAY IOC`
tradingSymbol	string	Refer Trading Symbol in Tables
securityId	string	Exchange standard id for each scrip. Refer **here**
quantity	int	Number of shares for the order
disclosedQuantity	int	Number of shares visible
price	float	Price at which order is placed
triggerPrice	float	Price at which order is triggered, for SL-M, SL-L, CO & BO
afterMarketOrder	boolean	The order placed is AMO ?
boProfitValue	float	Bracket Order Target Price change
boStopLossValue	float	Bracket Order Stop Loss Price change
legName	enum string	Leg identification in case of BO `ENTRY_LEG TARGET_LEG STOP_LOSS_LEG`
createTime	string	Time at which the order is created
updateTime	string	Time at which the last activity happened
exchangeTime	string	Time at which order reached at exchange
drvExpiryDate	int	For F&O, expiry date of contract
drvOptionType	enum string	Type of Option `CALL PUT`
drvStrikePrice	float	For Options, Strike Price
omserroeCode	string	Error code in case the order is rejected or failed
omsErrorDescription	string	Description of error in case the order is rejected or failed

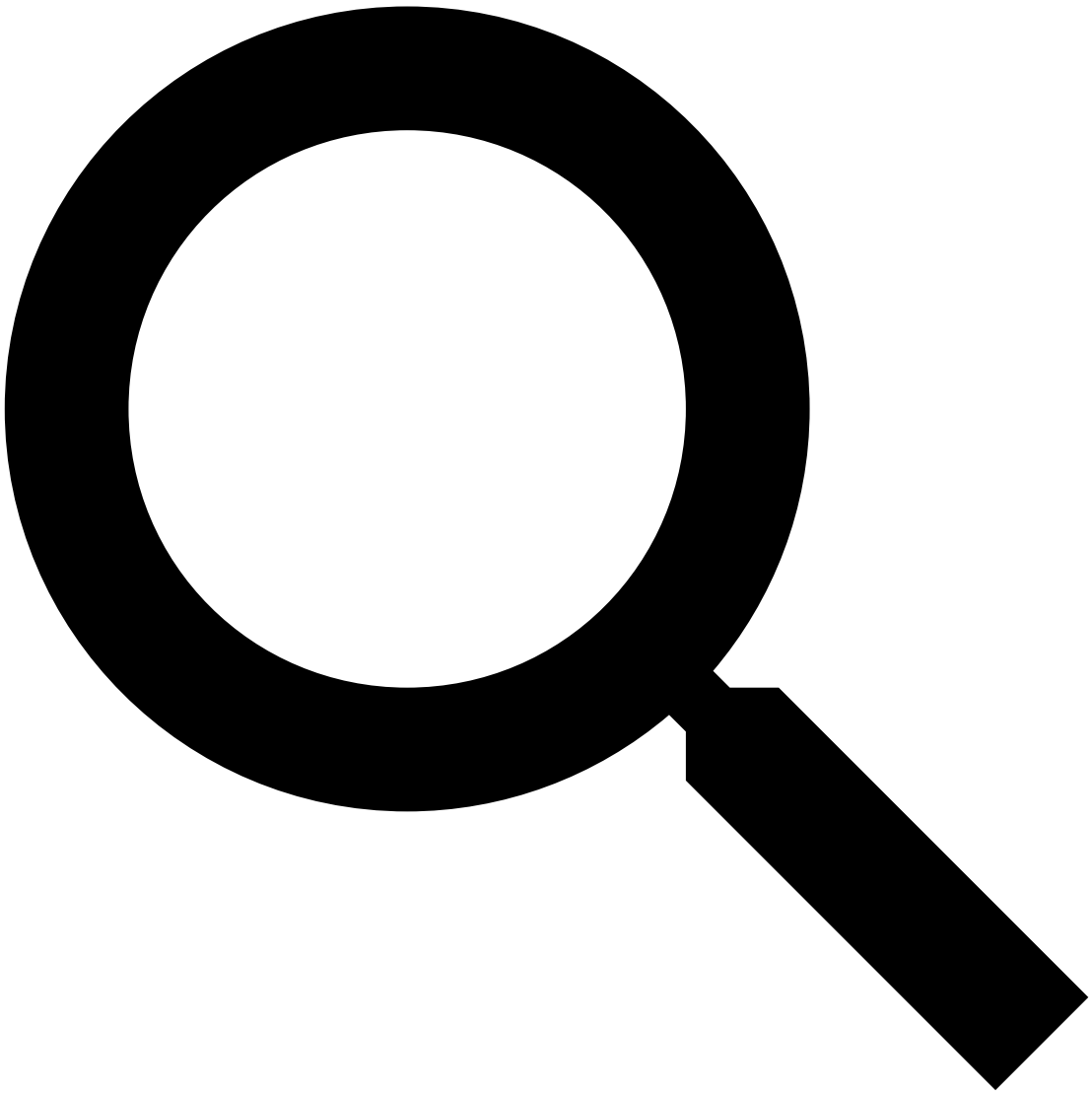
Setting up Postback

To set up Postback API, you will need to provide a unique Postback URL to receive callbacks. You will need to follow the steps below to set up Postback URL:

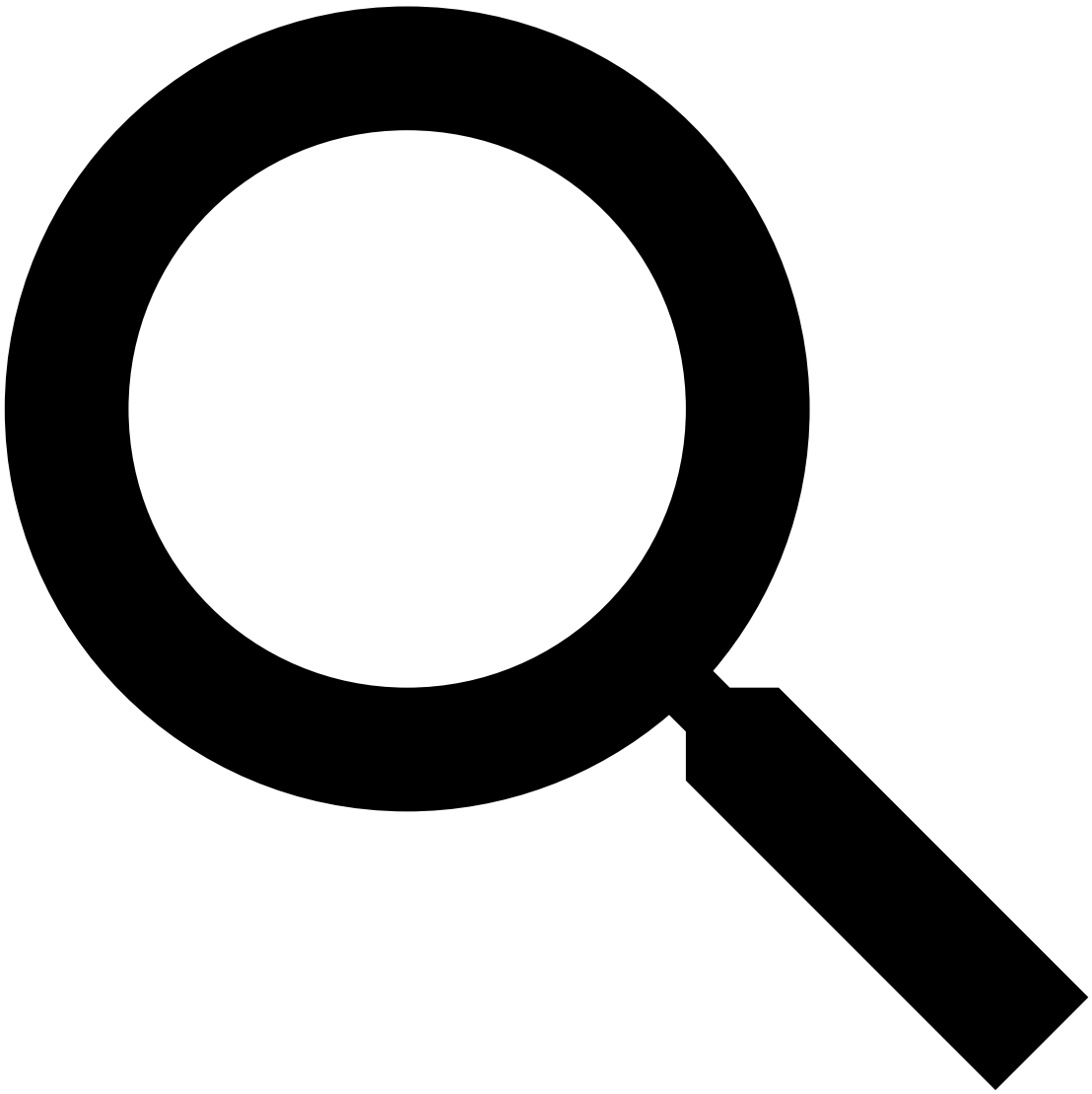
- While generating access token on web.dhan.co, enter your URL in the 'Postback URL' field.
- Click on 'Generate' to successfully set Postback and generate a new token.

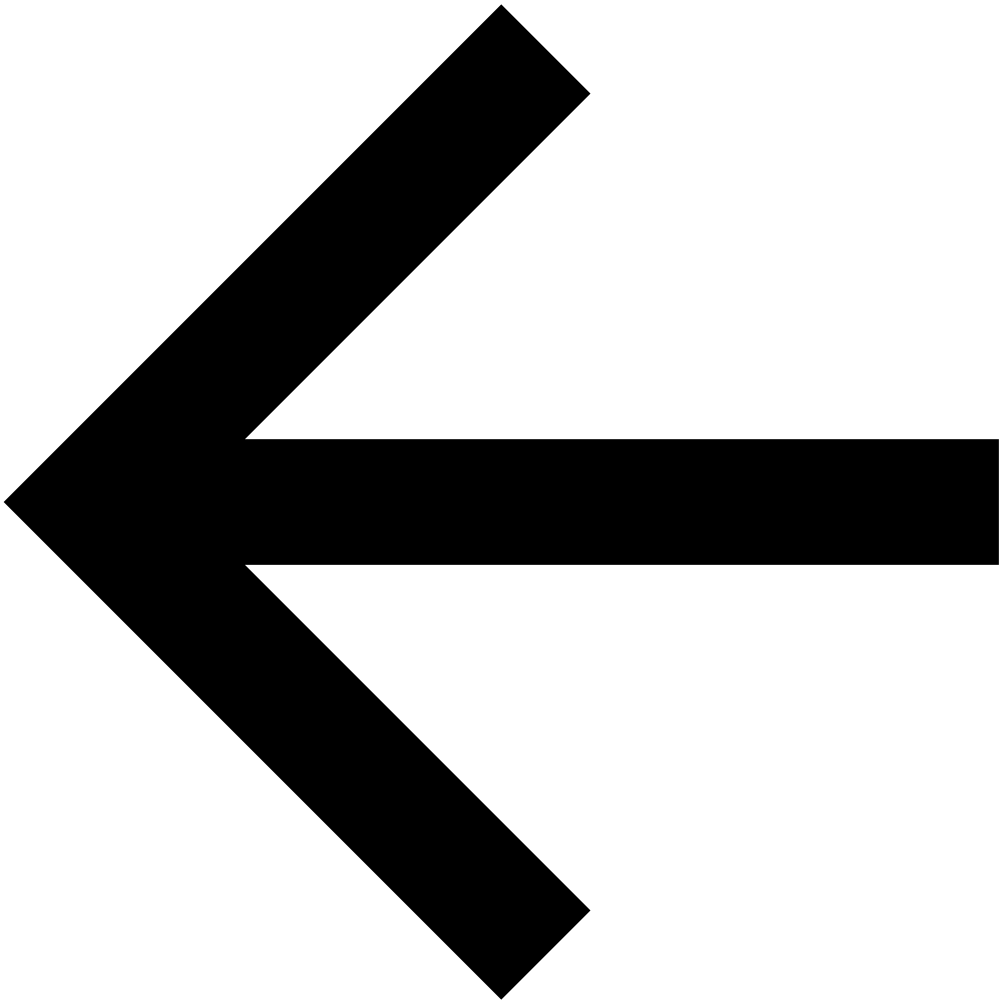
Important: You will not receive postback calls if Postback URL is set to `localhost` or `127.0.0.1`.

> Note: To receive Postback originating for all orders placed from a platform/app, Partner Login module needs to be used.



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- ☐ Releases Releases

Table of contents

  - Version 2.2
    - New Features
    - Improvements
    - Breaking Changes
  - Version 2.1
    - New Features
    - Improvements
  - Version 2
    - New Features
    - Improvements
    - Breaking Changes
    - Bug Fixes
- ☐ Trading APIs

Trading APIs

- Orders
- Super Order
- Forever Order
- Portfolio
- EDIS
- Trader's Control
- Funds
- Statement
- Postback
- Live Order Update

Table of contents

- Version 2.2
- ☐ Data APIs
  - New Features
  - Improvements
  - Breaking Changes

Data APIs

- Version 2.1
  - Market Quote
  - Live Market Feed
  - 20 Market Depth
  - Historical Data
  - Option Chain

New Features
Improvements

- Version 2
- Annexure
- Instrument List
  - Breaking Changes
  - Bug Fixes

Release Notes

Version 2.2

Date: Friday Mar 07 2025

We are adding a new order type on Dhan and which is available on v2 of DhanHQ API. This order type is called Super Order. This along with a major update to Historical Data APIs is added. You can now fetch upto last 5 years of Intraday Historical Data (minutewise) and also OI data for futures and options instruments.

New Features

- **Super Orders**
  Super Orders are a new order type which allows you to combine multiple orders for entry and exit into single order. You can enter a position and place target and stop loss orders for the same along with the option to trail your stop loss. This combines the benefits of a bracket order and a trailing stop, and is available across all exchanges and segments - Super Order.

- **User Profile**
  User Profile API is built to give a status check about different information related to user's account. This includes token validity, active segments, Data API subscription status and validity, and different user configurations like DDPI status and MTF enablement - here.

Improvements

- **Intraday Historical Data**
  Intraday Historical Data is now available for last 5 years of data. This is available for all NSE, BSE and MCX instruments. Along with increase in time range, we have also added OI data for futures and options instruments. There is `oi` parameter added to the API. Also, the `from_date` and `to_date` has option to pass IST time as well to fetch particular number of bars only. You can head over to documentation for updates in fields - here.

- **Daily Historical Data**
  Daily Historical Data has added OI data for futures and options instruments. There is `oi` parameter added to the API which is *optional* and can be used to fetch OI data - here.

- **`CorrelationId` on Live Order Update**
  Live Order Update now has two additional keys called 'CorrelationId' and 'Remarks' - here.

Breaking Changes

- **Changes in Rate Limit**
  Rate limits have been increased for Data APIs which includes Historical Data. There are no rate limits on minute and hourly time frames. You can request upto 1,00,000 requests in a day and seconds timeframe are limited to 5 requests per second - Rate Limit.

Version 2.1

Date: Monday Jan 06 2025

This add-on to DhanHQ v2 comes with 20 level market depth (Level 3 data) for APIs. Along with that, this also covers Option Chain API, which was released in between and smaller enhancements.

New Features

- **20 Market Depth**
  You can get real-time streaming of 20 level market depth, for all NSE instruments with 20 Market Depth. It is delivered via websockets and can be used to detect demand-supply zones and build your systems on top of it.

- **Option Chain**
  Dhan's Advanced Option Chain is made available on a single API request, for any underlying. With this, you get OI, greeks, volume, top bid/ask and price data of all strikes of any Option Instrument, across exchanges and segments - for NSE, BSE and MCX traded options - Option Chain API.

Improvements

- **'expiryCode' in Daily Historical Data**
  Daily Historical Data now has expiryCode as an "*Optional*" field - Daily Historical Data API.

Version 2

Date: Monday Sep 15 2024

DhanHQ v2 extends execution capability with live order updates and forever orders on superfast APIs. Along with this, we also released Market Quote APIs, built on top of Live Market Feed which can be integrated with ease. We have also introduced improvements, bug fixes and increased stability with new version.

New Features

- **Market Quote**
  Fetch LTP, Quote (with OI) and Market Depth data directly on API, for upto 1000 instruments at once with Market Quote API.

- **Forever Orders**
  Place, modify and manage your Forever Orders, including single and OCO orders to manage risk and trade efficiently with Forever Order API.

- **Live Order Update**
  Order Updates are sent in real time via websockets, which will update order status of all your orders placed via any platform - Live Order Update.

- **Margin Calculator**
  Margin Calculation simplifies order placement by providing details about required margin and available balances before placing order - Margin Calculator API.

Improvements

- **Intraday Historical Data**
  Intraday Historical Data now provides OHLC with Volume data for last 5 trading days across timeframes such as 1 min, 5 min, 15 min, 25 min and 60 min - Intraday Historical Data API.

- **GET Order APIs**
  `filledQty`, `remainingQuantity` and `averageTradedPrice` is available as part of all GET Order APIs, which makes it simpler to fetch post execution details of an order. We have also added `PART_TRADED` as a flag on `orderStatus` which will be clear distinction for partially traded orders.

- **Live Market Feed**
  You can now authorise Live Market Feed via Query Parameters and subscribe/unsubscribe to instruments via JSON

messages on websockets with this version. Also, `FULL` packet is now available which will gives LTP, Quote, OI and Market Depth data in a single packet.

Breaking Changes

- **Order Placement**
  Deprecated non-mandatory request keys including `tradingSymbol`, `drvExpiryDate`, `drvOptionType` and `drvStrikePrice` from Order Placement API endpoints. Along with this, pre-open AMO orders can also be placed now with `PRE_OPEN` tag.

- **Order Modification**
  `quantity` field needs to be placed order quantity instead of pending order quantity. Earlier, for Order Modification API, in case of partial execution, user needed to pass pending order quantity, which led to errors due to simultaneous execution on exchange or need to call GET Trade APIs as well. `quantity` and `price` fields are conditionally required for modification.

  `quantity` field in Order Modification

- **Daily Historical Data**
  `symbol` is replaced with `securityId` as key in Daily Historical Data, making it simple for users to fetch data everywhere with Security ID itself - Daily Historical Data API.

- **Error Messages**
  Now error messages are categorised with DH-900 series which helps you self debug on level of error - Error Codes.

- **Security ID List Mapping**
  Security ID List is now comprehensive with tag changes as well. Check new mappings and description - Security ID List.

- **Epoch time introduced instead of Julian time in Historical Data APIs** - Timestamp in Daily Historical Data API and Intraday Historical Data API is now Epoch or UNIX time and with key `timestamp`.

- **`Market Depth` deprecated as mode in Live Market Feed**
  `Market Depth` mode is now replaced with `FULL` packet which has combined data of Quote, OI and market depth in single packet, enabling ease in subscribing and fetching data.

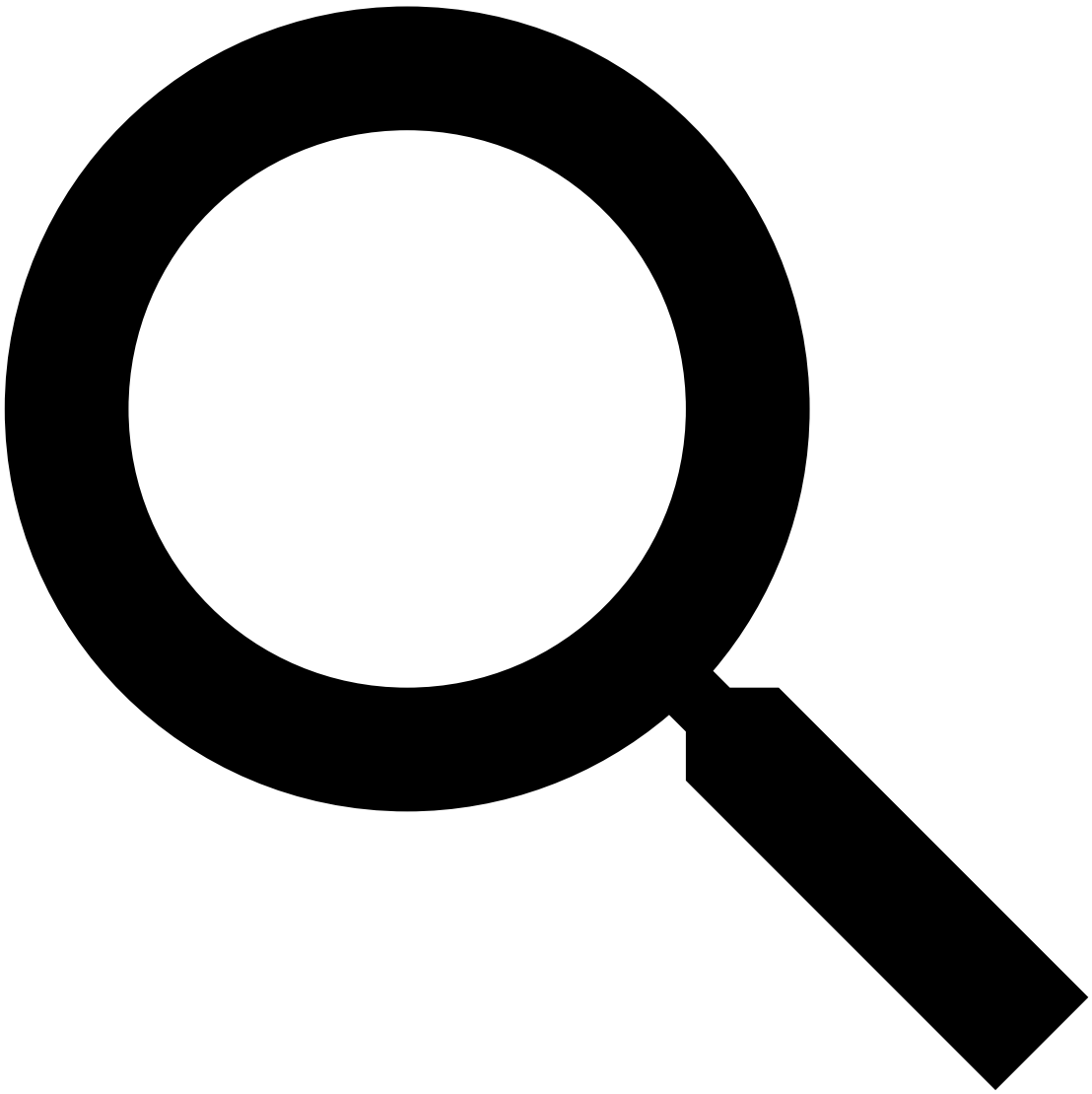
- **Change in endpoint for Trade History and Kill Switch**
  New endpoint for Trade History is `/trades`, making it common with other Trade book APIs. For Kill Switch, the new endpoint as per nomenclature is `killswitch`.

Bug Fixes

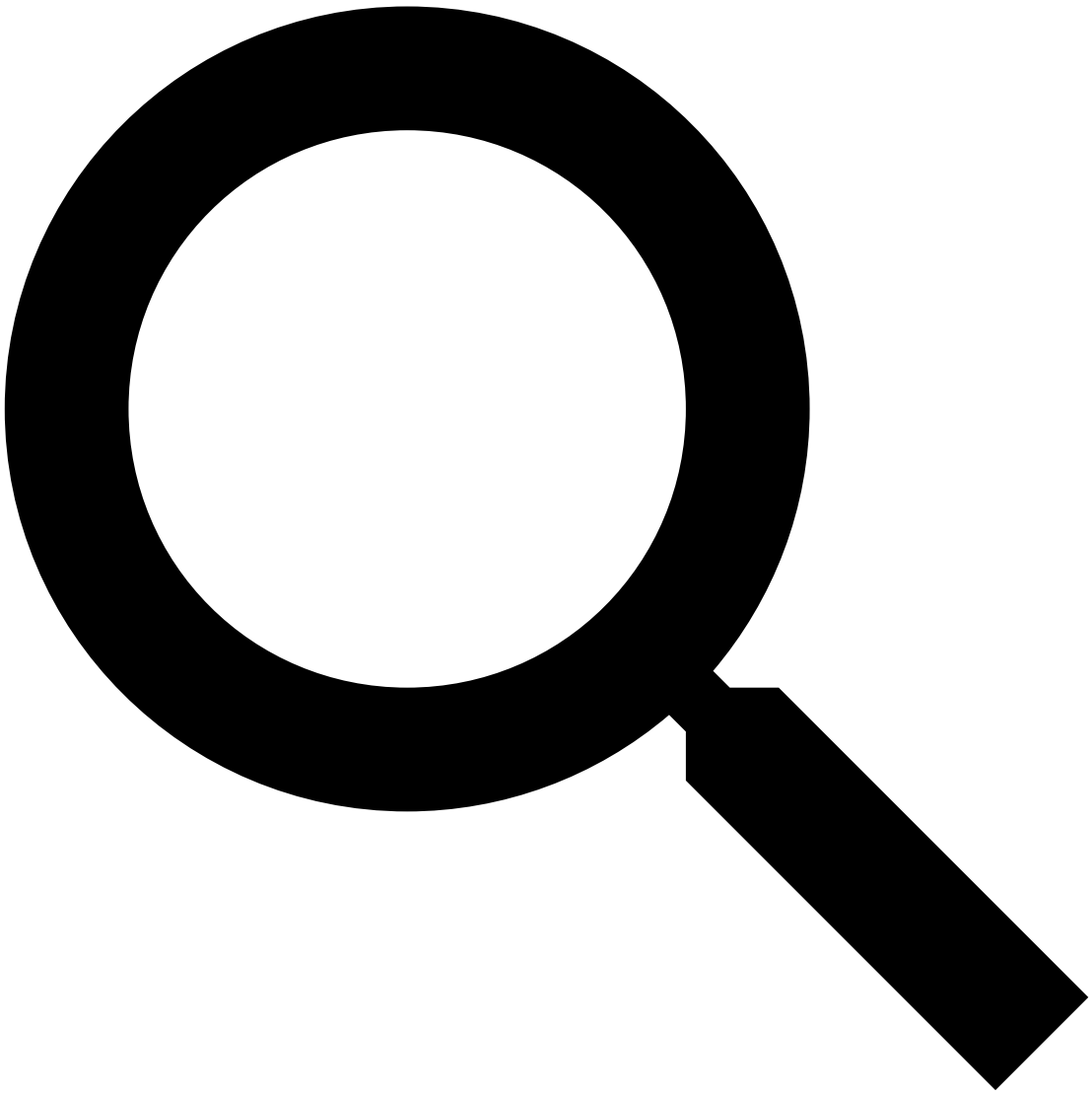
- **`realizedProfit` and `unrealizedProfit` in Positions API**
  You can now get realtime values of `realizedProfit` and `unrealizedProfit` on Positions API.

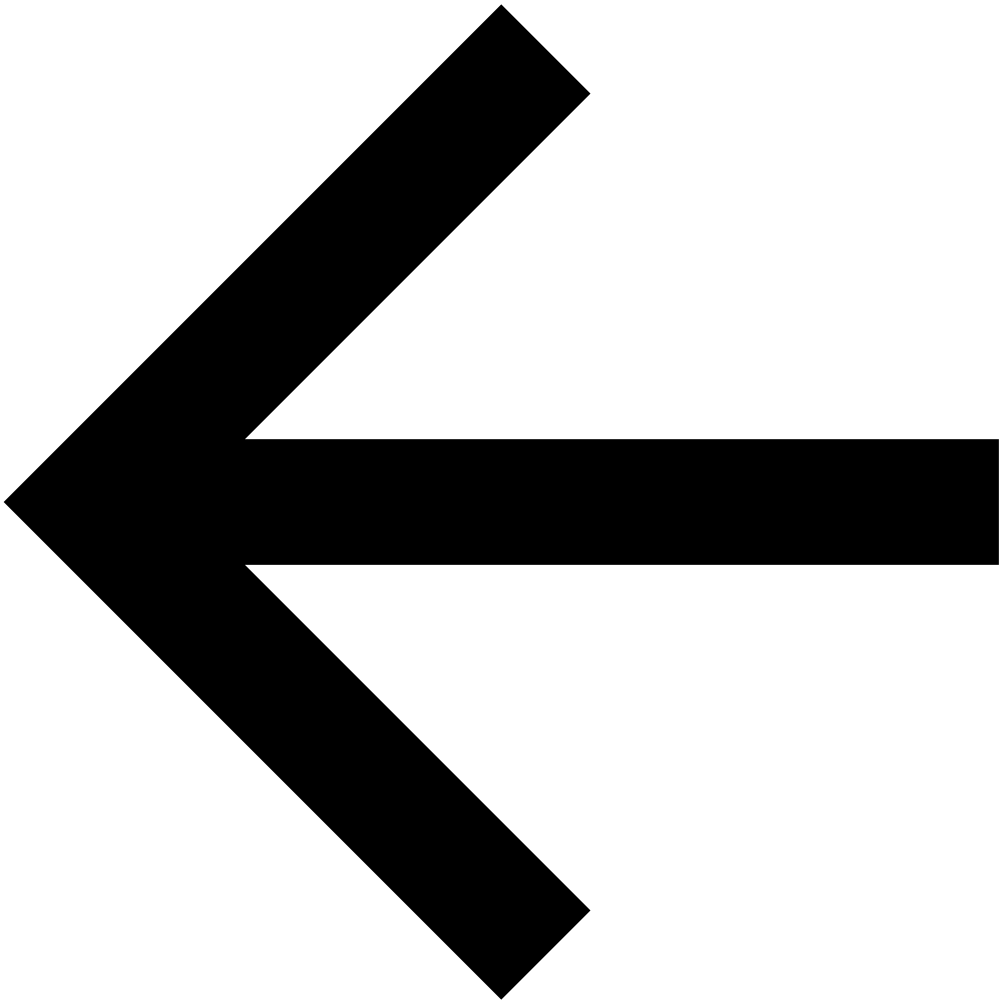
- **Target leg modification in Order Modification API**
  `TARGET_LEG` was not getting modified with Order Modification API which is fixed now.

Ver 2.0 / API Documentation
Super Order



Search





Type to start searching

Ver 2.0 / API Documentation

- Introduction
- Releases
- ☒ Trading APIs
  Trading APIs
  - Orders
  - ☐ Super Order Super Order
    Table of contents
    - Place Super Order
    - Modify Super Order
    - Cancel Super Order
    - Super Order List
  - Forever Order
  - Portfolio
  - EDIS
  - Trader's Control
  - Funds
  - Statement

- Postback
- Live Order Update
- ☐ Data APIs

Data APIs
- Market Quote
- Live Market Feed
- 20 Market Depth
- Historical Data
- Option Chain

Table of contents

- Annexure
- Instrument List
- Place Super Order
- Modify Super Order
- Cancel Super Order
- Super Order List

Super Order

Super orders are built for smart execution of your trades. They are a collection of orders clubbed into single order request, which includes entry leg, target leg and stop loss leg along with the option to add trailing stop loss.

This particular set of APIs can be used to create, modify and cancel super orders. You can place super orders across all exchanges and segments.

POST	/super/orders	Create a new super order
PUT	/super/orders/{order-id}	Modify a pending super order
DELETE	/super/orders/{order-id}/{order-leg}	Cancel a pending super order leg
GET	/super/orders	Retrieve the list of all super orders

Place Super Order

The super order request API lets you place new super orders. You can place a combination of orders using this API, wether that be entry leg, target leg and stop loss leg.

This order type is available across segments and exchanges. You can place intraday, carry forward or even MTF orders via this order type.

```
curl --request POST \
--url https://api.dhan.co/v2/super/orders \
--header 'Content-Type: application/json' \
--header 'access-token: JWT' \
--data '{Request JSON}'
```

Request Structure

```
{
"dhanClientId": "1000000003",
"correlationId": "123abc678",
"transactionType": "BUY",
"exchangeSegment": "NSE_EQ",
"productType": "CNC",
"orderType": "LIMIT",
"securityId": "11536",
```

```
    "quantity": 5,
    "price": 1500,
    "targetPrice": 1600,
    "stopLossPrice": 1400,
    "trailingJump": 10
}
```

Parameters

Field	Type	Description
dhanClientId *required*	string	User specific identification generated by Dhan
correlationId	string	The user/partner generated id for tracking back
transactionType *required*	enum string	The trading side of transaction `BUY` `SELL`
exchangeSegment *required*	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
productType *required*	enum string	Product type `CNC` `INTRADAY` `MARGIN` `MTF`
orderType *required*	enum string	Order Type `LIMIT` `MARKET`
securityId *required*	string	Exchange standard ID for each scrip. Refer **here**
quantity *required*	int	Number of shares for the order
price *required*	float	Price at which order is placed
targetPrice *required*	float	Target Price for the Super Order
stopLossPrice *required*	float	Stop Loss Price for the Super Order
trailingJump *required*	float	Price Jump by which Stop Loss should be trailed

Response Structure

```
{
    "orderId": "112111182198",
    "orderStatus": "PENDING",
}
```

Parameters

Field	Type	Description
orderId	string	Order specific identification generated by Dhan
orderStatus	enum string	Last updated status of the order `TRANSIT` `PENDING` `REJECTED`

Modify Super Order

This API can be used to modify any leg of a Super Order till it is in `PENDING` or `PART_TRADED` state.

Note

Order Entry Leg `ENTRY_LEG` can help modify the entire super order and can only be modified when the order status is `PENDING` or `PART_TRADED`. Once the entry order status is `TRADED`, only `TARGET_LEG` and `STOP_LOSS_LEG` price and trail jump can be modified.

```
  curl --request PUT \
  --url https://api.dhan.co/v2/super/orders/{order-id} \
  --header 'Content-Type: application/json' \
  --header 'access-token: JWT' \
  --data '{Request JSON}'
```

Request Structure

◉○○
Entry LegTargetStop Loss

```
{
    "dhanClientId":"1000000009",
    "orderId":"112111182045",
    "orderType":"LIMIT",
    "legName":"ENTRY_LEG",
    "quantity":"40",
    "price":"1300",
    "targetPrice": 1450,
    "stopLossPrice": 1350,
    "trailingJump": 20
}
```

```
{
    "dhanClientId":"1000000009",
    "orderId":"112111182045",
    "legName":"TARGET_LEG",
    "targetPrice": 1450
}
```

```
{
    "dhanClientId":"1000000009",
    "orderId":"112111182045",
    "legName":"STOP_LOSS_LEG",
    "stopLossPrice": 1350,
    "trailingJump": 20
}
```


Parameters

Field	Type	description
dhanClientId *required*	string	User specific identification generated by Dhan
orderId *required*	string	Order specific identification generated by Dhan
orderType *conditionally required*	enum string	Order Type `LIMIT` `MARKET`
legName *required*	enum string	`ENTRY_LEG` - Entire Super Order can be modified, only when main order status is `PENDING` or `PART_TRADED` `TARGET_LEG` `STOP_LOSS_LEG`
quantity *conditionally required*	int	Quantity to be modified - only for `ENTRY_LEG`
price *conditionally required*	float	Price to be modified - only for `ENTRY_LEG`
targetPrice *conditionally required*	float	Target Price to be modified - `ENTRY_LEG` or `TARGET_LEG`
stopLossPrice *conditionally required*	float	Stop Loss Price to be modified - `ENTRY_LEG` or `STOP_LOSS_LEG`
trailingJump *conditionally required*	float	Stop Loss Price jump to be modified - `ENTRY_LEG` or `STOP_LOSS_LEG` If trailing jump is not added or passed as `0`, it will be cancelled

Response Structure


```
{
    "orderId": "112111182045",
    "orderStatus": "TRANSIT"
}
```


Parameters

Field	Type	Description
orderId	string	Order specific identification generated by Dhan
orderStatus	enum string	Last updated status of the order `TRANSIT` `PENDING` `REJECTED` `TRADED`

Cancel Super Order

Users can cancel a pending/active super order using the order ID. There is no body for request and response for this call. On successful completion of request ‘202 Accepted’ response status code will appear.

```
curl --request DELETE \
--url https://api.dhan.co/v2/super/orders/{order-id}/{order-leg} \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Path Parameters

Field	Description	Example
order-id *required*	Order ID of the Order being cancelled	11211182198
order-leg *required*	Order Leg to be cancelled	`ENTRY_LEG` `TARGET_LEG` `STOP_LOSS_LEG`

> Note: Cancelling main order ID cancels all legs. If particular target or stop loss leg is cancelled, then the same cannot be added again.

Response Structure

```
{
"orderId": "112111182045",
"orderStatus": "CANCELLED"
}
```

Parameters

Field	Type	Description
orderId	string	Order specific identification generated by Dhan
orderStatus	enum string	Last updated status of the order `TRANSIT` `PENDING` `REJECTED` `CANCELLED`

Super Order List

This API lets you retrieve an array of all super orders placed in a day with their last updated status. This is a special order book which only consists of Super Orders, where the target and stop loss orders are nested under the main entry order leg. Individual legs of each super order can also be found in the main order book with their Order ID.

```
curl --request GET \
--url https://api.dhan.co/v2/super/orders \
--header 'Content-Type: application/json' \
--header 'access-token: JWT'
```

Request Structure

No Body

Response Structure

```
[
    {
        "dhanClientId": "1100003626",
        "orderId": "5925022734212",
        "correlationId": "string",
        "orderStatus": "PENDING",
        "transactionType": "BUY",
        "exchangeSegment": "NSE_EQ",
        "productType": "CNC",
        "orderType": "LIMIT",
        "validity": "DAY",
        "tradingSymbol": "HDFCBANK",
        "securityId": "1333",
        "quantity": 10,
        "remainingQuantity": 10,
        "ltp": 1660.95,
```

```
        "price": 1500,
        "afterMarketOrder": false,
        "legName": "ENTRY_LEG",
        "exchangeOrderId": "11925022734212",
        "createTime": "2025-02-27 19:09:42",
        "updateTime": "2025-02-27 19:09:42",
        "exchangeTime": "2025-02-27 19:09:42",
        "omsErrorDescription": "",
        "averageTradedPrice": 0,
        "filledQty": 0,
        "legDetails": [
            {
                "orderId": "5925022734212",
                "legName": "STOP_LOSS_LEG",
                "transactionType": "SELL",
                "totalQuatity": 0,
                "remainingQuantity": 0,
                "triggeredQuantity": 0,
                "price": 1400,
                "orderStatus": "PENDING",
                "trailingJump": 10
            },
            {
                "orderId": "5925022734212",
                "legName": "TARGET_LEG",
                "transactionType": "SELL",
                "remainingQuantity": 0,
                "triggeredQuantity": 0,
                "price": 1550,
                "orderStatus": "PENDING",
                "trailingJump": 0
            }
        ]
    }
]
```


Parameters

Field	Type	Description
dhanClientId	string	User specific identification generated by Dhan
orderId	string	Order specific identification generated by Dhan
correlationId	string	The user/partner generated id for tracking back
orderStatus	enum string	Last updated status of the order `TRANSIT PENDING CLOSED REJECTED CANCELLED PART_TRADED TRADED`
transactionType	enum string	The trading side of transaction `BUY SELL`
exchangeSegment	enum string	Exchange Segment of instrument to be subscribed as found in Annexure
productType	enum string	Product type of trade `CNC INTRADAY MARGIN MTF`
orderType	enum string	Order Type `LIMIT MARKET`
validity	enum string	Validity of Order `DAY`
tradingSymbol	string	Refer Trading Symbol in Tables
securityId	string	Exchange standard ID for each scrip. Refer **here**
quantity	int	Number of shares for the order
remainingQuantity	int	Quantity pending execution
ltp	float	Last Traded Price of the instrument
price	float	Price at which order is placed
afterMarketOrder	boolean	If the order is placed after market
legName	enum string	Leg identification in case of BO `ENTRY_LEG TARGET_LEG STOP_LOSS_LEG`
trailingJump	float	Price Jump by which Stop Loss should be trailed
exchangeOrderId	string	Exchange generated ID for the order
createTime	string	Time at which the order is created
updateTime	string	Last updated time of the order
exchangeTime	string	Time at which order was sent to the exchange
omsErrorDescription	string	Description of error in case the order is rejected or failed
remainingQuantity	integer	Quantity pending at the exchange to be traded (quantity - filledQty)
averageTradedPrice	integer	Average price at which order is traded
filledQty	integer	Quantity of order traded on Exchange
triggeredQuantity	integer	Quantity of Stop Loss or Target legs which has been placed on Exchange

legDetails	[]array	Array of Leg Details

Note

There are two order status updates that needs to be considered. `CLOSED` is used when the ENTRY_LEG and one of either TARGET_LEG or STOP_LOSS_LEG is also triggered for entire quantity. `TRIGGERED` is present for TARGET_LEG and STOP_LOSS_LEG which indicates which of the two is actually triggered and then `triggeredQuantity` can be referred to check the placed quantity.

Note: For description of enum values, refer Annexure