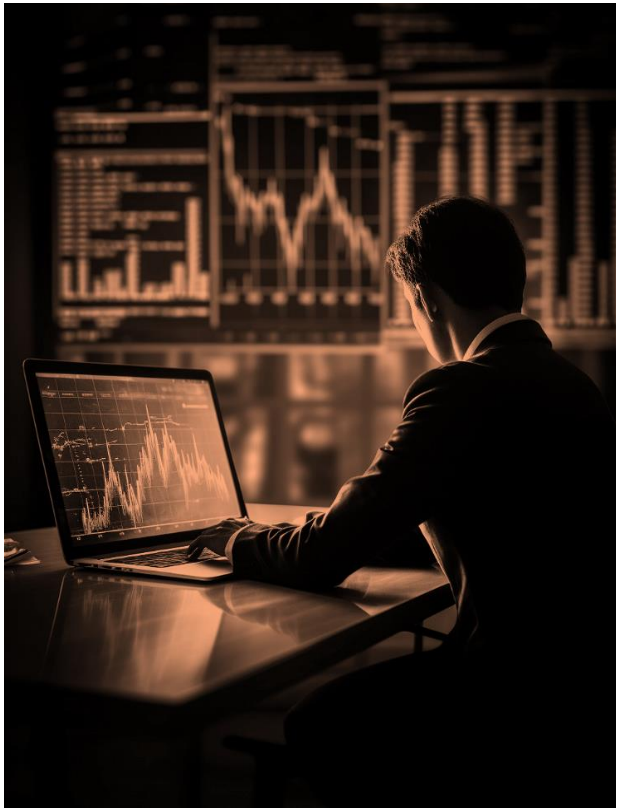
Global Datafeeds



WEBSOCKET API PYTHON

SAMPLE FOR ALL FUNCTIONS

PREFACE

Welcome to the documentation for our Python WebSocket API functions. This guide is designed to provide developers with a thorough understanding of how each function works, detailing their unique functionalities and expected responses. Whether you are new to Python or looking to deepen your knowledge, this documentation will serve as a comprehensive resource to help you effectively.

What This Documentation Includes

This detailed descriptions of each Python function, includes:

- Function Purpose: An explanation of what the function does and when to use it.
- Parameters: A list of parameters the function accepts, with descriptions.
- Return Values: Information on what the function returns.
- Example Usage: Practical code examples demonstrating how to use the function.
- Error Handling: Guidance on how to handle errors related to the function.

Contents

WEBSOCKET API PYTHON SAMPLE FOR ALL FUNCTIONS

https://globaldatafeeds.in/global-datafeeds-apis/global-datafeeds-apis/api-code-samples/websockets-python/

SR. NO	FUNCTION NAME & SNIPPET	DESCRIPTION	VALUES	COMMENTS	RESPONSE
1	*Authenticate* *def Authenticate(ws): print("Authenticating...")* *ws.send('{"MessageType":"Authenticate","Password":"' + apikey + '"}')*	This function authenticates the user and sends response accordingly.	Not Applicable	This function returns Authentication message.	{"Complete":true,"Message":"Welcome!","MessageType":"AuthenticateResult"}"

Phone: +91 7219764002 / +91 9175744002

Email: developer@globaldatafeeds.in

2	*SubscribeRealtime* *(Read More)* *def SubscribeRealtime(ws):* *Exchange = "NFO"* *InstIdentifier = "NIFTY-I"* *Unsubscribe = "false"* *strMessage = '{"MessageType":"SubscribeRealtime","Exchange":"' + Exchange + '","Unsubscribe":"' + Unsubscribe + '","InstrumentIdentifier":"' + InstIdentifier + '"}'* *print('Message : ' + strMessage)* *ws.send(strMessage)*	This function pushes realtime data every second for the subscribed instrument from server.	Exchange InstrumentIdentifier LastTradeTime ServerTime AverageTradedPrice BuyPrice BuyQty Close High Low LastTradePrice LastTradeQty Open OpenInterest QuotationLot SellPrice SellQty TotalQtyTraded Value PreOpen PriceChange PriceChangePercentage OpenInterestChange MessageType	• To see this function in action, you should run it during market hours.(server will push new data whenever available) • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to subscribe to data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • For this function, you will need to send this request only once and subscribe for the data. • Please note: If there is internet disconnection for any reason, you will need to subscribe to all the instruments again - to receive the data. • Unsubscribe is Optional parameter. By default subscribes to Realtime data. If [true], instrumentIdentifier is unsubscribed.	{"Exchange":"NFO","InstrumentIdentifier":"NIFTY-I","LastTradeTime":1719549311,"ServerTime":1719549312,"AverageTradedPrice":24194.09,"BuyPrice":24200.0,"BuyQty":3700,"Close":24095.85,"High":24240.0,"Low":24102.1,"LastTradePrice":24202.15,"LastTradeQty":1575,"Open":24151.0,"OpenInterest":14226975,"QuotationLot":25.0,"SellPrice":24202.15,"SellQty":2125,"TotalQtyTraded":2121350,"Value":51324132821.5,"PreOpen":false,"PriceChange":106.3,"PriceChangePercentage":0.44,"OpenInterestChange":298750,"MessageType":"RealtimeResult"}

3	*SubscribeSnapshot* *(Read More)* *def SubscribeSnapshot(ws):* *ExchangeName = "NFO"* *InstIdentifier = "NIFTY-I"* *Periodicity = "MINUTE"* *Period = 1* *Unsubscribe = "false"* *strMessage = '{"MessageType":"SubscribeSnapshot","Exchange":"' + ExchangeName + '","InstrumentIdentifier":"' + InstIdentifier + '","Period":' + f'{Period}' + ',"Periodicity":"' + Periodicity + '","Unsubscribe":"' + Unsubscribe + '"}'* *print(strMessage)* *ws.send(strMessage)*	This function pushes realtime Snapshots as per "Period" and "Periodicity" mentioned. For eg, if Periodicity is "Minute" and "Period" is 1 then server pushes the data of the instrument whenever 1 minute completes.	Exchange InstrumentIdentifier Periodicity LastTradeTime TradedQty OpenInterest Open High Low Close MessageType	• To see this function in action, you should run it during market hours.(server will push new data whenever available) • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to subscribe to data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • periodicity=MINUTE: Specifies that the data should be minute. • period=1: Specifies the time period (1 min). Acceptable values are : MINUTE: 1 , 2 , 3 , 4 , 5 , 6 , 10 , 12 , 15 , 20, 30 HOUR: 1 , 2 , 3 , 4 , 6 , 8 , 12 • For this function, you will need to send this request only once and subscribe for the snapshot data. • Please note: If there is internet disconnection for any reason, you will need to subscribe to all the instruments again - to receive the data. • Unsubscribe is Optional parameter. By default subscribes to Realtime data. If [true], instrumentIdentifier is unsubscribed.	{"Exchange":"NFO","InstrumentIdentifier":"FUTIDX_NIFTY_25JUL2024_XX_0","Periodicity":"MINUTE","Period":1,"LastTradeTime":1719811260,"TradedQty":5225,"OpenInterest":14337725,"Open":24152.55,"High":24154.2,"Low":24148.4,"Close":24150.0,"MessageType":"RealtimeSnapshotResult"}

4	*GetLastQuote* *(Read More)* *def GetLastQuote(ws):* *ExchangeName = "NFO"* *InstIdentifier = "NIFTY-I"* *isShortIdentifier = "false"* *strMessage = '{"MessageType":"GetLastQuote","Exchange":"' + ExchangeName + '","isShortIdentifier":"' + isShortIdentifier + '","InstrumentIdentifier":"' + InstIdentifier + '"}'* *ws.send(strMessage)*	This function return the data of SINGLE SYMBOL - whenever requested.	Exchange InstrumentIdentifier LastTradePrice ServerTime AverageTradedPrice BuyPrice BuyQty Close High Low LastTradePrice LastTradeQty Open OpenInterest QuotationLot SellPrice SellQty TotalQtyTraded Value PreOpen PriceChange PriceChangePercentage OpenInterestChange MessageType	• You will need to send these requests EVERY TIME when you need latest data. • exchange=MCX: Specifies the exchange as MCX. Please see available exchanges Here • If you want to subscribe to data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • isShortIdentifier= true(Optional parameter) Functions will use short instrument identifier format if set as [true].By default function will use long instrument identifier format. Eg. Long instrument identifier FUTCOM_CRUDEOIL_17JUL2024_0	{"Exchange":"MCX","InstrumentIdentifier":"CRUDEOIL-I","LastTradeTime":1719814531,"ServerTime":1719814536,"AverageTradedPrice":6836.49,"BuyPrice":6843.0,"BuyQty":4,"Close":6805.0,"High":6845.0,"Low":6829.0,"LastTradePrice":6844.0,"LastTradeQty":0,"Open":6835.0,"OpenInterest":4503,"QuotationLot":100.0,"SellPrice":6845.0,"SellQty":5,"TotalQtyTraded":822,"Value":561960000.0,"PreOpen":false,"PriceChange":39.0,"PriceChangePercentage":0.57,"OpenInterestChange":20,"MessageType":"LastQuoteResult"}

5	*GetLastQuoteShort* *(Read More)* *def GetLastQuoteShort(ws):* *ExchangeName = "NFO"* *InstIdentifier = "NIFTY-I"* *isShortIdentifier = "false"* *strMessage = '{"MessageType":"GetLastQuoteShort","Exchange":"' + ExchangeName + '","isShortIdentifier":"' + isShortIdentifier + '","InstrumentIdentifier":"' + InstIdentifier + '"}'* *ws.send(strMessage)*	This function returns single record of realtime data of single symbol. Contains limited fields in response	Exchange InstrumentIdentifier LastTradeTime BuyPrice LastTradePrice SellPrice MessageType	• You will need to send these requests EVERY TIME when you need latest data. • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to subscribe to data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • isShortIdentifier= true(Optional parameter) Functions will use short instrument identifier format if set as [true].By default function will use long instrument identifier format. Eg. Long instrument identifier FUTIDX_NIFTY_25JUL2024_XX_0	{"Exchange":"NFO","InstrumentIdentifier":"NIFTY-I","LastTradeTime":1719818712,"BuyPrice":24199.55,"LastTradePrice":24200.2,"SellPrice":24202.6,"MessageType":"LastQuoteShortResult"}

Phone: +91 7219764002 / +91 9175744002
Email: developer@globaldatafeeds.in www.globaldatafeeds.in

6	***GetLastQuoteShortWithClose*** *(Read More)* *def GetLastQuoteShortWithClose(ws):* *ExchangeName = "NFO"* *InstIdentifier = "NIFTY-I"* *isShortIdentifier = "false"* *strMessage = '{"MessageType":"GetLastQuoteShortWithClose","Exchange":"' + ExchangeName + '","isShortIdentifier":"' + isShortIdentifier + '","InstrumentIdentifier":"' + InstIdentifier + '"}'* *ws.send(strMessage)*	This function returns single record of realtime data of single symbol. Contains limited fields in response	Exchange InstrumentIdentifier LastTradeTime BuyPrice Close LastTradePrice SellPrice MessageType	• You will need to send these requests EVERY TIME when you need latest data. • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to subscribe to data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • isShortIdentifier= true(Optional parameter) Functions will use short instrument identifier format if set as [true].By default function will use long instrument identifier format. Eg. Long instrument identifier FUTIDX_NIFTY_25JUL2024_XX_0	{"Exchange":"NFO","InstrumentIdentifier":"NIFTY25JUL24FUT","LastTradeTime":1719819664,"BuyPrice":24194.3,"Close":24132.25,"LastTradePrice":24194.3,"SellPrice":24194.35,"MessageType":"LastQuoteShortWithCloseResult"}

7	*GetLastQuoteArray* *(Read More)* *def GetLastQuoteArray(ws):* *ExchangeName = "NFO"* *isShortIdentifiers = "false"* *InstrumentIdentifiers = '[{"Value":"NIFTY-I"},{"Value":"BANKNIFTY-I"}]'* *strMessage = '{"MessageType":"GetLastQuoteArray","Exchange":"' + ExchangeName + '","isShortIdentifier":"' + isShortIdentifiers + '","InstrumentIdentifiers":' + InstrumentIdentifiers + '}'* *ws.send(strMessage)*	This functions return the data of MULTIPLE SYMBOLS (MAX 25 IN SINGLE CALL) - whenever requested. Contains many fields in response	Exchange InstrumentIdentifier ServerTime AverageTradedPrice BuyPrice BuyQty Close High Low LastTradePrice LastTradeQty Open OpenInterest QuotationLot SellPrice SellQty TotalQty TradedValue PreOpen PriceChange PriceChangePercentage OpenInterestChange MessageType	• You will need to send these requests EVERY TIME when you need latest data. • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to subscribe to data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • isShortIdentifier= true(Optional parameter) Functions will use short instrument identifier format if set as [true].By default function will use long instrument identifier format. Eg. Long instrument identifier FUTIDX_NIFTY_25JUL2024_XX_0	{"Result":[{"Exchange":"NFO","InstrumentIdentifier":"NIFTY-I","LastTradeTime":1719821380,"ServerTime":1719821380,"AverageTradedPrice":24151.16,"BuyPrice":24212.0,"BuyQty":25,"Close":24132.25,"High":24220.0,"Low":24082.0,"LastTradePrice":24212.05,"LastTradeQty":25,"Open":24125.0,"OpenInterest":14491600,"QuotationLot":25.0,"SellPrice":24212.15,"SellQty":100,"TotalQtyTraded":2941950,"Value":71051505162.0,"PreOpen":false,"PriceChange":79.8,"PriceChangePercentage":0.33,"OpenInterestChange":333100,"MessageType":"LastQuoteResult"}, {"Exchange":"NFO","InstrumentIdentifier":"BANKNIFTY-I","LastTradeTime":1719821380,"ServerTime":1719821380,"AverageTradedPrice":52588.36,"BuyPrice":52748.2,"BuyQty":15,"Close":52529.7,"High":52798.0,"Low":52301.0,"LastTradePrice":52750.0,"LastTradeQty":0,"Open":52524.0,"OpenInterest":2551650,"QuotationLot":15.0,"SellPrice":52750.0,"SellQty":45,"TotalQtyTraded":1244385,"Value":65440166358.6,"PreOpen":false,"PriceChange":220.3,"PriceChangePercentage":0.42,"OpenInterestChange":98730,"MessageType":"LastQuoteResult"}, {"Exchange":"NFO","InstrumentIdentifier":"RELIANCE-I","LastTradeTime":1719821379,"ServerTime":1719821380,"AverageTradedPrice":3144.3,"BuyPrice":3132.85,"BuyQty":500,"Close":3151.45,"High":3171.5,"Low":3126.4,"LastTradePrice":3132.8,"LastTradeQty":0,"Open":3140.0,"OpenInterest":30387250,"QuotationLot":250.0,"SellPrice":3133.2,"SellQty":250,"TotalQtyTraded":4871000,"Value":15315885300.0,"PreOpen":false,"PriceChange":-18.65,"PriceChangePercentage":0.59,"OpenInterestChange":32750,"MessageType":"LastQuoteResult"}],"MessageType":"LastQuoteArrayResult"}

8	*GetLastQuoteArrayShort* *[(Read More)]* *def GetLastQuoteArrayShort(ws):* *ExchangeName = "NFO"* *isShortIdentifiers = "false"* *InstrumentIdentifiers = '[{"Value":"NIFTY-I"},{"Value":"BANKNIFTY-I"}]'* *strMessage = '{"MessageType":"GetLastQuoteArrayShort","Exchange":"' + ExchangeName + '","isShortIdentifier":"' + isShortIdentifiers + '","InstrumentIdentifiers":' + InstrumentIdentifiers + '}'* *ws.send(strMessage)*	This function returns array of realtime data of multiple symbols. Contains limited fields in response	Exchange InstrumentIdentifier LastTradeTime BuyPrice LastTradePrice SellPrice MessageType	• You will need to send these requests EVERY TIME when you need latest data. • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to subscribe to data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • isShortIdentifier= true(Optional parameter) Functions will use short instrument identifier format if set as [true].By default function will use long instrument identifier format. Eg. Long instrument identifier FUTIDX_NIFTY_25JUL2024_XX_0	{"Result":[{"Exchange":"NFO","InstrumentIdentifier":"NIFTY25JUL24FUT","LastTradeTime":1719823485,"BuyPrice":24210.0,"LastTradePrice":24213.5,"SellPrice":24211.4,"MessageType":"LastQuoteShortResult"},{"Exchange":"NFO","InstrumentIdentifier":"BANKNIFTY20JULFUT","LastTradeTime":0,"BuyPrice":0.0,"LastTradePrice":0.0,"SellPrice":0.0,"MessageType":"LastQuoteShortResult"},{"Exchange":"NFO","InstrumentIdentifier":"RELIANCE20JULFUT","LastTradeTime":0,"BuyPrice":0.0,"LastTradePrice":0.0,"SellPrice":0.0,"MessageType":"LastQuoteShortResult"}],"MessageType":"LastQuoteArrayShortResult"}

9	*GetLastQuoteArrayShortWithClose* *(Read More)* *def GetLastQuoteArrayShortWithClose(ws):* *ExchangeName = "NSE"* *isShortIdentifiers = "false"* *InstrumentIdentifiers = '[{"Value":"WIPRO"},{"Value":"RELIANCE"}]'* *strMessage = '{"MessageType":"GetLastQuoteArrayShortWithClose","Exchange":"' + ExchangeName + '","isShortIdentifier":"' + isShortIdentifiers + '","InstrumentIdentifiers":' + InstrumentIdentifiers + '}'* *ws.send(strMessage)*	This function returns array of realtime data of multiple symbols. Contains limited fields in response	Exchange InstrumentIdentifier LastTradeTime BuyPrice Close LastTradePrice SellPrice MessageType	• You will need to send these requests EVERY TIME when you need latest data. • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to subscribe to data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • isShortIdentifier= true(Optional parameter) Functions will use short instrument identifier format if set as [true].By default function will use long instrument identifier format. Eg. Long instrument identifier FUTIDX_NIFTY_25JUL2024_XX_0	[{"Exchange":"NSE","InstrumentIdentifier":"SBIN","LastTradeTime":1720175393,"BuyPrice":859.75,"Close":859.75,"LastTradePrice":859.75,"SellPrice":0.0,"MessageType":"LastQuoteShortWithCloseResult"},{"Exchange":"NSE","InstrumentIdentifier":"RELIANCE","LastTradeTime":1720176757,"BuyPrice":3177.25,"Close":3177.25,"LastTradePrice":3177.25,"SellPrice":0.0,"MessageType":"LastQuoteShortWithCloseResult"},{"Exchange":"NSE","InstrumentIdentifier":"AXISBANK","LastTradeTime":1720175165,"BuyPrice":1287.05,"Close":1287.05,"LastTradePrice":1287.05,"SellPrice":0.0,"MessageType":"LastQuoteShortWithCloseResult"}],"MessageType":"LastQuoteArrayShortWithCloseResult"}

10	*GetSnapshot* *(Read More)* *def GetSnapshot(ws):* *ExchangeName = "NSE"* *Periodicity = "MINUTE"* *Period = 1* *InstrumentIdentifiers = '[{"Value":"WIPRO"},{"Value":"RELIANCE"}]'* *isShortIdentifiers = "false"* *strMessage = '{"MessageType":"GetSnapshot","Exchange":"' + ExchangeName + '","Periodicity":"' + Periodicity + '","Period":' + f'{Period}' + ',"isShortIdentifiers":"' + isShortIdentifiers + '","InstrumentIdentifiers":' + InstrumentIdentifiers + '}'* *ws.send(strMessage)*	This function returns latest single snapshot of MULTIPLE SYMBOLS (MAX 25 in single call) as per "Periodicity" & "Period" mentioned. For example, if Periodicity is "Minute" and "Period" is 1 then server will return the data of the requested instrument(s) whenever 1 minute completes.	Exchange InstrumentIdentifier LastTradeTime TradedQty OpenInterest Open High Low Close MessageType	• You will need to send these requests EVERY TIME when you need latest data. • exchange=MCX: Specifies the exchange as Multi Commodity Exchange. Please see available exchanges Here • If you want to get data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • For this function, you will need to send this request only once and subscribe for the data.	{"Result":[{"InstrumentIdentifier":"NATURALGAS20JULFUT","Exchange":"MCX","LastTradeTime":0,"TradedQty":0,"OpenInterest":0,"Open":0.0,"High":0.0,"Low":0.0,"Close":0.0,"TokenNumber":null},{"InstrumentIdentifier":"CRUDEOIL20JULFUT","Exchange":"MCX","LastTradeTime":0,"TradedQty":0,"OpenInterest":0,"Open":0.0,"High":0.0,"Low":0.0,"Close":0.0,"TokenNumber":null}],"MessageType":"SnapshotResult"}

11	*GetHistory* *(Read More)* *def GetHistory(ws):* *ExchangeName = "NSE"* *InstIdentifier = "SBIN"* *Periodicity = "MINUTE"* *Period = 1* *# From = 1615919400* *# To = 1621171693* *Max = 100* *UserTag = "User1"* *isShortIdentifier = "true"* *strMessage =* *'{"MessageType":"GetHistory","Exchange":"' + ExchangeName + '","InstrumentIdentifier":"' + InstIdentifier + '","Periodicity":"' + Periodicity + '","Period":"' + str(Period) + '","Max":"' + str(Max) + '","UserTag":"' + UserTag + '","isShortIdentifier":"' + isShortIdentifier + '"}'* *print('Message : ' + strMessage)* *ws.send(strMessage)*	This function returns historical data of SINGLE SYMBOL as per "Periodicity" & "Period" mentioned This is a very powerful function which supports many optional parameters to download full / incremental data	**For DAY & MINUTE** LastTradeTime QuotationLot TradedQty OpenInterest Open High Low Close **For TICK** LastTradeTime LastTradePrice QuotationLot TradedQty OpenInterest BuyPrice BuyQty SellPrice SellQty	• exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to get the data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • periodicity=DAY: Specifies that the data should be daily. • period=1: Specifies the time period (1 day). Acceptable values are : MINUTE: 1 , 2 , 3 , 4 , 5 , 6 , 10 , 12 , 15 , 20, 30 HOUR: 1 , 2 , 3 , 4 , 6 , 8 , 12 • For this function, you will need to send this request only once and get history data.	DAY RESPONSE {"Request":{"Exchange":"NFO","InstrumentIdentifier":"NIFTYI","IsShortIdentifier":false,"From":0,"To":0,"Max":10,"Periodicity":"DAY","Period":1,"UserTag":null,"AdjustSplits":true,"MessageType":"GetHistory"},"Result":[{"LastTradeTime":1719923400,"QuotationLot":25,"TradedQty":5297200,"OpenInterest":14603925,"Open":24241.0,"High":24280.0,"Low":24122.0,"Close":24203.15},{"LastTradeTime":1719837000,"QuotationLot":25,"TradedQty":4080225,"OpenInterest":14590225,"Open":24125.0,"High":24225.2,"Low":24082.0,"Close":24206.25},...{"LastTradeTime":1719405000,"QuotationLot":25,"TradedQty":5723300,"OpenInterest":6613600,"Open":23731.25,"High":23906.8,"Low":23676.0,"Close":23867.95}],"MessageType":"HistoryOHLCResult"} -- MINUTE RESPONSE {"Request":{"Exchange":"NFO","InstrumentIdentifier":"NIFTYI","IsShortIdentifier":false,"From":0,"To":0,"Max":10,"Periodicity":"MINUTE","Period":1,"UserTag":null,"AdjustSplits":true,"MessageType":"GetHistory"},"Result":[{"LastTradeTime":1723784940,"QuotationLot":25,"TradedQty":12300,"OpenInterest":11372250,"Open":24260.0,"High":24273.75,"Low":24260.0,"Close":24270.0},..,{"LastTradeTime":1723784460,"QuotationLot":25,"TradedQty":59725,"OpenInterest":11371550,"Open":24242.55,"High":24258.15,"Low":24233.0,"Close":24254.45},{"LastTradeTime":1723784400,"QuotationLot":25,"TradedQty":62475,"OpenInterest":11371550,"Open":24259.35,"High":24262.35,"Low":24241.15,"Close":24241.35}],"MessageType":"HistoryOHLCResult"} ------- -- TICK RESPONSE : {"Request":{"Exchange":"NFO","InstrumentIdentifier":"NIFTYI","IsShortIdentifier":false,"From":0,"To":0,"Max":10,"Periodicity":"TICK","Period":1,"UserTag":null,"AdjustSplits":true,"MessageType":"GetHistory"},"Result":[{"LastTradeTime":1723785033,"LastTradePrice":24275.5,"QuotationLot":25,"TradedQty":500,"OpenInterest":11359150,"BuyPrice":24274.1,"BuyQty":75,"SellPrice":24275.5,"SellQty":25},...{"LastTradeTime":1723785020,"LastTradePrice":24275.0,"QuotationLot":25,"TradedQty":300,"OpenInterest":11359150,"BuyPrice":24274.1,"BuyQty":100,"SellPrice":24275.0,"SellQty":475},{"LastTradeTime":1723785018,"LastTradePrice":24275.0,"QuotationLot":25,"TradedQty":425,"OpenInterest":11359150,"BuyPrice":24273.55,"BuyQty":25,"SellPrice":24275.0,"SellQty":300}],"MessageType":"HistoryTickResult"}

12	*GetHistory (Allow Only After Market Data)* *(Read More)* *def GetHistory(ws):* *ExchangeName = "NSE"* *InstIdentifier = "SBIN"* *Periodicity = "MINUTE"* *Period = 1* *# From = 1615919400* *# To = 1621171693* *Max = 100* *UserTag = "User1"* *isShortIdentifier = "true"* *strMessage = '{"MessageType":"GetHistory","Exchange":"' + ExchangeName + '","InstrumentIdentifier":"' + InstIdentifier + '","Periodicity":"' + Periodicity + '","Period":"' + str(Period) + '","Max":"' + str(Max) + '","UserTag":"' + UserTag + '","isShortIdentifier":"' + isShortIdentifier + '"}'* *print('Message : ' + strMessage)* *ws.send(strMessage)*	This function returns historical data of SINGLE SYMBOL as per "Periodicity" & "Period" mentioned. IMP: For this function, User needs to enable ***Allow Only After Market Data*** *for Key from GFDL Team*	**For DAY & MINUTE** LastTradeTime QuotationLot TradedQty OpenInterest Open High Low Close **For TICK** LastTradeTime LastTradePrice QuotationLot TradedQty OpenInterest BuyPrice BuyQty SellPrice SellQty	• This function returns historical data (Tick / Minute / EOD) till **Previous working day.** • This function is useful for the users / service providers who want to provide services like backtesting – as they do not need live / current day's data. • To receive current day's historical data via this function, you will need to send request after market is closed. • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here • If you want to get the data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here • periodicity=DAY: Specifies that the data should be daily. • period=1: Specifies the time period (1 day). Acceptable values are : MINUTE: 1 , 2 , 3 , 4 , 5 , 6 , 10 , 12 , 15 , 20, 30 HOUR: 1 , 2 , 3 , 4 , 6 , 8 , 12 • For this function, you will need to send this request only once and get history data.	DAY RESPONSE {"Request":{"Exchange":"NFO","InstrumentIdentifier":"NIFTYI","IsShortIdentifier":false,"From":0,"To":0,"Max":10,"Periodicity":"DAY","Period":1,"UserTag":null,"AdjustSplits":true,"MessageType":"GetHistory"},"Result":[{"LastTradeTime":1719923400,"QuotationLot":25,"TradedQty":5297200,"OpenInterest":14603925,"Open":24241.0,"High":24280.0,"Low":24122.0,"Close":24203.15},{"LastTradeTime":1719837000,"QuotationLot":25,"TradedQty":4080225,"OpenInterest":14590225,"Open":24125.0,"High":24225.2,"Low":24082.0,"Close":24206.25},..{"LastTradeTime":1719405000,"QuotationLot":25,"TradedQty":5723300,"OpenInterest":6613600,"Open":23731.25,"High":23906.8,"Low":23676.0,"Close":23867.95}],"MessageType":"HistoryOHLCResult"} -- MINUTE RESPONSE {"Request":{"Exchange":"NFO","InstrumentIdentifier":"NIFTYI","IsShortIdentifier":false,"From":0,"To":0,"Max":10,"Periodicity":"MINUTE","Period":1,"UserTag":null,"AdjustSplits":true,"MessageType":"GetHistory"},"Result":[{"LastTradeTime":1723784940,"QuotationLot":25,"TradedQty":12300,"OpenInterest":11372250,"Open":24260.0,"High":24273.75,"Low":24260.0,"Close":24270.0},..,{"LastTradeTime":1723784460,"QuotationLot":25,"TradedQty":59725,"OpenInterest":11371550,"Open":24242.55,"High":24258.15,"Low":24233.0,"Close":24254.45},{"LastTradeTime":1723784400,"QuotationLot":25,"TradedQty":62475,"OpenInterest":11371550,"Open":24259.35,"High":24262.35,"Low":24241.15,"Close":24241.35}],"MessageType":"HistoryOHLCResult"} --TICK RESPONSE RESPONSE: {"Request":{"Exchange":"NFO","InstrumentIdentifier":"NIFTYI","IsShortIdentifier":false,"From":0,"To":0,"Max":10,"Periodicity":"TICK","Period":1,"UserTag":null,"AdjustSplits":true,"MessageType":"GetHistory"},"Result":[{"LastTradeTime":1723785033,"LastTradePrice":24275.5,"QuotationLot":25,"TradedQty":500,"OpenInterest":11359150,"BuyPrice":24274.1,"BuyQty":75,"SellPrice":24275.5,"SellQty":25},..{"LastTradeTime":1723785020,"LastTradePrice":24275.0,"QuotationLot":25,"TradedQty":300,"OpenInterest":11359150,"BuyPrice":24274.1,"BuyQty":100,"SellPrice":24275.0,"SellQty":475},{"LastTradeTime":1723785018,"LastTradePrice":24275.0,"QuotationLot":25,"TradedQty":425,"OpenInterest":11359150,"BuyPrice":24273.55,"BuyQty":25,"SellPrice":24275.0,"SellQty":300}],"MessageType":"HistoryTickResult"}

13	*GetExchanges* *(Read More)* *def GetExchanges(ws):* *strMessage = '{"MessageType":"GetExchanges"}'* *ws.send(strMessage)*	This function returns array of Exchanges allowed for your API Key	It can be any of the below: NFO NSE NSE_IDX BFO BSE BSE_IDX BSE_DEBT CDS MCX	• User will get list of exchanges i.e. enabled for API Key • If you want to get the data of multiple symbols, you will need to send 1 request each - for each symbol. Please see symbol naming conventions Here	{"Result":[{"Value":"BFO"},{"Value":"BSE"},{"Value":"BSE_DEBT"},{"Value":"BSE_IDX"},{"Value":"CDS"},{"Value":"MCX"},{"Value":"NFO"},{"Value":"NSE"},{"Value":"NSE_IDX"}],"MessageType":"ExchangesResult"}

14	*GetInstrumentsOnSearch function* *(Read More)* *def GetInstrumentsOnSearch(ws):* *ExchangeName = "NFO"* *Search = "NIFTY"* *Product = "NIFTY"* *InstrumentType = "FUTIDX"* *# OptionType = "PE"* *# Expiry = "29APR2021"* *# StrikePrice = 10000* *# OnlyActive = "TRUE"* *strMessage = '{"MessageType":"GetInstrumentsOnSearch","Exchange":"' + ExchangeName + '","Search":"' + Search + '"}'* *ws.send(strMessage)*	This function returns array of max. 20 instruments by selected exchange and 'search string'	Identifier Name Expiry StrikePrice Product PriceQuotationUnit OptionType ProductMonth UnderlyingAsset UnderlyingAssetExpiry IndexName TradeSymbol QuotationLot Description TokenNumber LowPriceRange HighPriceRange	• User will get list of instruments from selected Exchange • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here •instrumentType -FUTIDX Specifies the instrument as Future Index. •Search= NIFTY, specifies the symbol that need to be search.	{"Request":{"Exchange":"NFO","Search":"NIFTY","InstrumentType":"FUTIDX","OnlyActive":true,"MessageType":"GetInstrumentsOnSearch"},"Result":[{"Identifier":"FUTIDX_NIFTY_25JUL2024_XX_0","Name":"FUTIDX","Expiry":"25Jul2024","StrikePrice":0.0,"Product":"NIFTY","PriceQuotationUnit":"","OptionType":"XX","ProductMonth":"25Jul2024","UnderlyingAsset":"","UnderlyingAssetExpiry":"","IndexName":"","TradeSymbol":"NIFTY25JUL24FUT","QuotationLot":25.0,"Description":"","TokenNumber":"35007","LowPriceRange":21782.85,"HighPriceRange":26623.5},{"Identifier":"FUTIDX_NIFTYNXT50_26JUL2024_XX_0","Name":"FUTIDX","Expiry":"26Jul2024","StrikePrice":0.0,"Product":"NIFTYNXT50","PriceQuotationUnit":"","OptionType":"XX","ProductMonth":"26Jul2024","UnderlyingAsset":"","UnderlyingAssetExpiry":"","IndexName":"","TradeSymbol":"NIFTYNXT5026JUL24FUT","QuotationLot":10.0,"Description":"","TokenNumber":"35239","LowPriceRange":64718.35,"HighPriceRange":79100.2},{"Identifier":"FUTIDX_NIFTY_29AUG2024_XX_0","Name":"FUTIDX","Expiry":"29Aug2024","StrikePrice":0.0,"Product":"NIFTY","PriceQuotationUnit":"","OptionType":"XX","ProductMonth":"29Aug2024","UnderlyingAsset":"","UnderlyingAssetExpiry":"","IndexName":"","TradeSymbol":"NIFTY29AUG24FUT","QuotationLot":25.0,"Description":"","TokenNumber":"35415","LowPriceRange":21887.15,"HighPriceRange":26751.0},{"Identifier":"FUTIDX_NIFTYNXT50_30AUG2024_XX_0","Name":"FUTIDX","Expiry":"30Aug2024","StrikePrice":0.0,"Product":"NIFTYNXT50","PriceQuotationUnit":"","OptionType":"XX","ProductMonth":"30Aug2024","UnderlyingAsset":"","UnderlyingAssetExpiry":"","IndexName":"","TradeSymbol":"NIFTYNXT5030AUG24FUT","QuotationLot":10.0,"Description":"","TokenNumber":"35078","LowPriceRange":64989.0,"HighPriceRange":79430.95},{"Identifier":"FUTIDX_NIFTY_26SEP2024_XX_0","Name":"FUTIDX","Expiry":"26Sep2024","StrikePrice":0.0,"Product":"NIFTY","PriceQuotationUnit":"","OptionType":"XX","ProductMonth":"26Sep2024","UnderlyingAsset":"","UnderlyingAssetExpiry":"","IndexName":"","TradeSymbol":"NIFTY26SEP24FUT","QuotationLot":25.0,"Description":"","TokenNumber":"35000","LowPriceRange":21989.95,"HighPriceRange":26876.6},{"Identifier":"FUTIDX_NIFTYNXT50_27SEP2024_XX_0","Name":"FUTIDX","Expiry":"27Sep2024","StrikePrice":0.0,"Product":"NIFTYNXT50","PriceQuotationUnit":"","OptionType":"XX","ProductMonth":"27Sep2024","UnderlyingAsset":"","UnderlyingAssetExpiry":"","IndexName":"","TradeSymbol":"NIFTYNXT5027SEP24FUT","QuotationLot":10.0,"Description":"","TokenNumber":"35004","LowPriceRange":65759.55,"HighPriceRange":80372.8}],"MessageType":"InstrumentsOnSearchResult"}

15	*GetInstruments* *(Read More)* *def GetInstruments(ws):* *ExchangeName = "NFO"* *# InstrumentType = "FUTSTK"* *# Product = "NIFTY"* *# OptionType = "PE"* *# Expiry = "29AUG2024"* *# StrikePrice = 10000* *# OnlyActive = "TRUE"* *strMessage = '{"MessageType":"GetInstruments","Exchange":"' + ExchangeName + '"}'* *ws.send(strMessage)*	This function returns array of instruments by selected exchange	Identifier Name Expiry StrikePrice Product PriceQuotationUnit OptionType ProductMonth UnderlyingAsset UnderlyingAssetExpiry IndexName TradeSymbol QuotationLot Description TokenNumber LowPriceRange HighPriceRange Messagetype	• Pease Note: Huge data of several MB is returned if GetInstruments is called without any optional arguments (NSE & NFO). It is strongly advised that users build a local symbol cache at their end and refresh with our server only "on need basis". This will result in faster response time and smoother experience for endusers • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	{"Request":{"Exchange":"NFO","InstrumentType":"FUTIDX","Product":"BANKNIFTY","Expiry":"31JUL2024","OnlyActive":true,"MessageType":"GetInstruments"},"Result":[{"Identifier":"FUTIDX_BANKNIFTY_31JUL2024_XX_0","Name":"FUTIDX","Expiry":"31Jul2024","StrikePrice":0.0,"Product":"BANKNIFTY","PriceQuotationUnit":"","OptionType":"XX","ProductMonth":"31Jul2024","UnderlyingAsset":"","UnderlyingAssetExpiry":"","IndexName":"","TradeSymbol":"BANKNIFTY31JUL24FUT","QuotationLot":15.0,"Description":"","TokenNumber":"35165","LowPriceRange":47131.65,"HighPriceRange":57605.35}],"MessageType":"InstrumentsResult"}
16	*GetInstrumentsTypes* *(Read More)* *def GetInstrumentTypes(ws):* *ExchangeName = "NFO"* *strMessage = '{"MessageType":"GetInstrumentTypes","Exchange":"' + ExchangeName + '"}'* *ws.send(strMessage)*	This function returns list of Instrument Types. Supported Exchange Values : NFO, NSE, NSE_IDX, CDS, MCX. Mandatory Parameter	It can be any of the below: FUTIDX FUTSTK OPTIDX OPTSTK	• InstrumentType : Optional argument to filter the search by products like FUTIDX, FUTSTK, OPTIDX, OPTSTK, FUTCUR, FUTCOM, etc. • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	{"Request":{"Exchange":"NFO","MessageType":"GetInstrumentTypes"},"Result":[{"Value":"FUTIDX"},{"Value":"FUTSTK"},{"Value":"OPTIDX"},{"Value":"OPTSTK"}],"MessageType":"InstrumentTypesResult"}

17	*GetProducts* *(Read More)* *def GetProducts(ws):* *ExchangeName = "NFO"* *strMessage = '{"MessageType":"GetProducts","Exchange":"' + ExchangeName + '"}'* *ws.send(strMessage)*	This function returns list of Products (e.g. NIFTY, BANKNIFTY, GAIL, etc.)	NA	• List of all Products from selected exchange like: NIFTY BANKNIFTY MIDCPNIFTY BAJFINANCE , etc • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	[{"Value":"011NSETEST"},{"Value":"021NSETEST"},{"Value":"031NSETEST"},{"Value":"041NSETEST"},{"Value":"051NSETEST"},{"Value":"061NSETEST"},{"Value":"071NSETEST"},{"Value":"081NSETEST"},{"Value":"091NSETEST"},{"Value":"101NSETEST"},{"Value":"111NSETEST"},{"Value":"121NSETEST"},{"Value":"131NSETEST"},{"Value":"141NSETEST"},{"Value":"151NSETEST"},{"Value":"161NSETEST"},{"Value":"171NSETEST"},..... .... many more.
18	*GetExpiryDates* *(Read More)* *def GetExpiryDates(ws):* *ExchangeName = "NFO"* *strMessage = '{"MessageType":"GetExpiryDates","Exchange":"' + ExchangeName + '"}'* *ws.send(strMessage)*	This function returns array of Expiry Dates	25JUL2024 01AUG2024 29AUG2024 26SEP2024	• List of all expiry dates for selected exchange will be returned • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	{"Request":{"Exchange":"NFO","Product":"NIFTY","MessageType":"GetExpiryDates"},"Result":[{"Value":"04JUL2024"},{"Value":"11JUL2024"},{"Value":"18JUL2024"},{"Value":"25JUL2024"},{"Value":"01AUG2024"},{"Value":"29AUG2024"},{"Value":"26SEP2024"},{"Value":"26DEC2024"},{"Value":"27MAR2025"},{"Value":"26JUN2025"},{"Value":"24DEC2025"},{"Value":"25JUN2026"},{"Value":"31DEC2026"},{"Value":"24JUN2027"},{"Value":"30DEC2027"},{"Value":"29JUN2028"},{"Value":"28DEC2028"},{"Value":"28JUN2029"}],"MessageType":"ExpiryDatesResult"}
19	*GetOptionTypes* *(Read More)* *def GetOptionTypes(ws):* *ExchangeName = "NFO"* *strMessage = '{"MessageType":"GetOptionTypes","Exchange":"' + ExchangeName + '"}'* *ws.send(strMessage)*	This function returns list of Option Types (e.g. CE, PE, etc.)	It can be any of the below: FF XX CE PE	• User will get list of option types from selected Exchange • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	{"Exchange":"NFO","MessageType":"GetOptionTypes"},"Result":[{"Value":"FF"},{"Value":"XX"},{"Value":"CE"},{"Value":"PE"}],"MessageType":"OptionTypesResult"}

20	*GetStrikePrices* *(Read More)* *def GetStrikePrices(ws):* *ExchangeName = "NFO"* *strMessage = '{"MessageType":"GetStrikePrices", "Exchange":"' + ExchangeName + '"}'* *ws.send(strMessage)*	This function returns list of Strike Prices (e.g. 10000, 11000, 75.5, etc.)	NA	•List of all strike prices for selected exchange will be returned • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	{"Value":"159000"},{"Value":"160000"},{"Value":"161000"},{"Value":"162000"},{"Value":"163000"},{"Value":"164000"},{"Value":"165000"},{"Value":"166000"},{"Value":"167000"},{"Value":"168000"},{"Value":"169000"},{"Value":"170000"},{"Value":"171000"},{"Value":"172000"},{"Value":"173000"},{"Value":"174000"},{"Value":"176000"},{"Value":"178000"},{"Value":"180000"},{"Value":"182000"}
21	*GetServerInfo* *(Read More)* *def GetServerInfo(ws):* *strMessage = '{"MessageType":"GetServerInfo"}'* *ws.send(strMessage)*	This function returns the server endpoint where user is connected	NA	• User will get Server Endpoint details.	{"ServerID":"EF89 -18","MessageType":"ServerInfoResult"}

22	*GetLimitation* *(Read More)* *def GetLimitation(ws):* *strMessage = '{"MessageType":"GetLimitation"}'* *ws.send(strMessage)*	This function returns user account information (functions allowed, Exchanges allowed, symbol limit, etc.)	List of all information of user account like AllowedBandwidthPerHour AllowedCallsPerHour AllowedCallsPerMonth AllowedBandwidthPerMonth ExpirationDate AllowedExchanges Functions allowed, etc	•If User wish to get the limitations for his subscription can run the GetLimitation function	{"GeneralParams":{"AllowedBandwidthPerHour":-1.0,"AllowedCallsPerHour":-1,"AllowedCallsPerMonth":-1,"AllowedBandwidthPerMonth":-1.0,"ExpirationDate":1722277799,"Enabled":true},"AllowedExchanges":[{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"BFO"},{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"BSE"},{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"BSE_DEBT"},{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"BSE_IDX"},{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"CDS"},{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"MCX"},{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"NFO"},{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"NSE"},{"AllowedInstruments":-1,"DataDelay":0,"ExchangeName":"NSE_IDX"}],"AllowedFunctions":[{"FunctionName":"GetExchangeMessages","IsEnabled":true},{"FunctionName":"GetExchangeSnapshot","IsEnabled":true},{"FunctionName":"GetExchangeSnapshotAfterMarket","IsEnabled":true},{"FunctionName":"GetHistory","IsEnabled":true},{"FunctionName":"GetHistoryGreeks","IsEnabled":true},{"FunctionName":"GetLastQuote","IsEnabled":true},{"FunctionName":"GetLastQuoteArray","IsEnabled":true},{"FunctionName":"GetLastQuoteArrayOptionGreeks","IsEnabled":true},{"FunctionName":"GetLastQuoteArrayShort","IsEnabled":true},{"FunctionName":"GetLastQuoteOptionChain","IsEnabled":true},{"FunctionName":"GetLastQuoteOptionChainWithGreeks","IsEnabled":true},{"FunctionName":"GetLastQuoteOptionGreeks","IsEnabled":true},{"FunctionName":"GetLast

					QuoteShort","IsEnabled":true},{"FunctionName":"GetMarketMessages","IsEnabled":true},{"FunctionName":"GetSnapshot","IsEnabled":true},{"FunctionName":"GetTopGainersLosers","IsEnabled":true},{"FunctionName":"SubscribeGainersLosers","IsEnabled":true},{"FunctionName":"SubscribeOptionChain","IsEnabled":true},{"FunctionName":"SubscribeOptionChainGreeks","IsEnabled":true},{"FunctionName":"SubscribeRealtime","IsEnabled":true},{"FunctionName":"SubscribeRealtimeGreeks","IsEnabled":true},{"FunctionName":"SubscribeSnapshot","IsEnabled":true}],"HistoryLimitation":{"TickEnabled":true,"DayEnabled":true,"WeekEnabled":true,"MonthEnabled":true,"MaxEOD":100000,"MaxIntraday":66,"Hour_2Enabled":true,"Hour_3Enabled":true,"Hour_4Enabled":true,"Hour_6Enabled":true,"Hour_8Enabled":true,"Hour_12Enabled":true,"Minute_3Enabled":true,"Minute_4Enabled":true,"Minute_6Enabled":true,"Minute_12Enabled":true,"Minute_20Enabled":true,"MaxTicks":15,"Hour_1Enabled":true,"Minute_1Enabled":true,"Minute_2Enabled":true,"Minute_5Enabled":true,"Minute_10Enabled":true,"Minute_15Enabled":true,"Minute_30Enabled":true},"GetSnapshotLimitation":{"DayEnabled":true,"Hour_1Enabled":true,"Minute_1Enabled":true,"Minute_2Enabled":true,"Minute_5Enabled":true,"Minute_10Enabled":true,"Minute_15Enabled":true,"Minute_30Enabled":true},"ExchangeSnapshotLimitation":{"DayEnabled":true,"Hour_1Enabled":true,"Minute_1Enabled":true,"Minute_2Enabled":true,"Minute_5Enabled":true,"Minute_10Enabled":true,"Minute_15Enabled":true,"Minute_30Enabled":true},"ExchangeSnapshotInstrumentTypeLimitation":null,"SubscribeSnapshotLimitation":{"Hour_1Enabled":true,"Minute_1Enabled":true,"Minute_2Enabled":false,"Minute_5Enabled":true,"Minute_10Enabled":false,"Minute_15Enabled":true,"Minute_30Enabled":false},"MessageType":"LimitationResult"}

23	*GetMarketMessages* *(Read More)* *def GetMarketMessages(ws):* *ExchangeName = "MCX"* *strMessage = '{"MessageType":"GetMarketMessages","Exchange":"' + ExchangeName + '"}'* *ws.send(strMessage)*	This function returns array of last messages (Market Messages) related to selected exchange	ServerTime SessionID MarketType Exchange	•If User wish to get messages from Market for selected Exchange the Latest messages will be fetched Eg. Market session 1, Market session 2, etc • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	{"Request":{"Exchange":"MCX","MessageType":"GetMarketMessages"},"Result":[{"ServerTime":1719944990,"SessionID":1,"MarketType":"Special Session Close\r","Exchange":"MCX","MessageType":"MarketMessagesItemResult"},{"ServerTime":1719977340,"SessionID":1,"MarketType":"Special Session Close\r","Exchange":"MCX","MessageType":"MarketMessagesItemResult"},{"ServerTime":1719977400,"SessionID":2,"MarketType":"Normal Market Open","Exchange":"MCX","MessageType":"MarketMessagesItemResult"}],"MessageType":"MarketMessagesResult"}
24	*GetExchangeMessages* *(Read More)* *def GetExchangeMessages(ws):* *ExchangeName = "MCX"* *strMessage = '{"MessageType":"GetExchangeMessages","Exchange":"' + ExchangeName + '"}'* *ws.send(strMessage)*	This function returns array of last messages (Exchange Messages) related to selected exchange	Exchange ServerTime Identifier Message MessageType	•If User wish to get messages from selected Exchange the Latest messages will be fetched • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	{"Request":{"Exchange":"MCX","MessageType":"GetExchangeMessages"},"Result":[{"ServerTime":1719976500,"Identifier":"Market","Message":"Special Session Open - Session 1\r","MessageType":"ExchangeMessagesItemResult"},{"ServerTime":1719977340,"Identifier":"Market","Message":"Special Session Closed - Session 1\r","MessageType":"ExchangeMessagesItemResult"},{"ServerTime":1719977400,"Identifier":"Market","Message":"The Normal market has Opened for trading - Session 2","MessageType":"ExchangeMessagesItemResult"}],"MessageType":"ExchangeMessagesResult"}

25	*GetLastQuoteOptionChain* *(Read More)* *def GetLastQuoteOptionChain(ws):* *Exchange = "NFO"* *Product = "NIFTY"* *Expiry = "11JUL2024"* *# OptionType = "CE"* *StrikePrice = 24000* *strMessage = '{"MessageType":"GetLastQuoteOptionChain","Exchange":"' + Exchange + '","Product":"' + Product + '"}'* *ws.send(strMessage)*	This function returns OptionChain data in realtime	Exchange InstrumentIdentifier LastTradeTime ServerTime AverageTradedPrice BuyPrice BuyQty Close High Low LastTradePrice LastTradeQty Open OpenInterest QuotationLot SellPrice SellQty TotalQtyTraded Value PreOpen PriceChange PriceChangePercentage OpenInterestChange MessageType	• To see this function in action, you should run it during market hours.(server will push new data whenever available) • exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	[{"Exchange":"NFO","InstrumentIdentifier":"OPTIDX_NIFTY_11JUL2024_CE_24000","LastTradeTime":1720171597,"ServerTime":1720171597,"AverageTradedPrice":306.72,"BuyPrice":285.3,"BuyQty":550,"Close":353.2,"High":364.35,"Low":250.0,"LastTradePrice":285.2,"LastTradeQty":0,"Open":250.0,"OpenInterest":1066375,"QuotationLot":25.0,"SellPrice":285.95,"SellQty":325,"TotalQtyTraded":6136725,"Value":1882256292.0,"PreOpen":false,"PriceChange":-68.0,"PriceChangePercentage":-19.25,"OpenInterestChange":116525,"MessageType":"LastQuoteResult"},{"Exchange":"NFO","InstrumentIdentifier":"OPTIDX_NIFTY_11JUL2024_PE_24000","LastTradeTime":1720171597,"ServerTime":1720171597,"AverageTradedPrice":45.67,"BuyPrice":45.6,"BuyQty":1950,"Close":42.8,"High":64.8,"Low":33.25,"LastTradePrice":45.7,"LastTradeQty":750,"Open":51.4,"OpenInterest":5815875,"QuotationLot":25.0,"SellPrice":45.75,"SellQty":2625,"TotalQtyTraded":40100650,"Value":1831396685.5,"PreOpen":false,"PriceChange":2.9,"PriceChangePercentage":6.78,"OpenInterestChange":1781150,"MessageType":"LastQuoteResult"}],"MessageType":"LastQuoteArrayResult"}

26	*GetExchangeSnapshot* *(Read More)* *def GetExchangeSnapshot(ws):* *Exchange = "NFO"* *Periodicity = "Minute"* *Period = 1* *From: 1719894600* *To: 1719896400* *strMessage = '{"MessageType":"GetExchangeSnapshot","Exchange":"' + Exchange + '","Period":' + f'{Period}' + ',"Periodicity":"' + Periodicity + '"}'* *ws.send(strMessage)*	This function returns entire Exchange Snapshot in realtime This function can return maximum 5 snapshots in single call. You will need to use "From" and "To" parameters to control the dataset required	InstrumentIdentifier Exchange LastTradeTime TradedQty OpenInterest Open High Low Close TokenNumber	• exchange=NFO: Specifies the exchange as NSE Futures and Options. Please see available exchanges Here	{"Request":{"Exchange":"NFO","Periodicity":"MINUTE","Period":1,"From":1719894600,"To":1719896400,"NonTraded":false,"Limitations":[],"Delay":0,"MessageType":"GetExchangeSnapshot"},"Result":[{"LastTradeTime":1719896400,"Result":[{"InstrumentIdentifier":"OPTSTK_GMRINFRA_25JUL2024_PE_88","Exchange":"NFO","LastTradeTime":1719896400,"TradedQty":11250,"OpenInterest":675000,"Open":0.9,"High":0.9,"Low":0.9,"Close":0.9,"TokenNumber":"101445"}, ...{"InstrumentIdentifier":"OPTIDX_NIFTY_11JUL2024_PE_22750","Exchange":"NFO","LastTradeTime":1719896160,"TradedQty":200,"OpenInterest":25325,"Open":6.05,"High":6.05,"Low":6.0,"Close":6.0,"TokenNumber":"35689"},{"InstrumentIdentifier":"OPTIDX_BANKNIFTY_03JUL2024_PE_40000","Exchange":"NFO","LastTradeTime":1719896160,"TradedQty":11610,"OpenInterest":907950,"Open":0.3,"High":0.3,"Low":0.25,"Close":0.3,"TokenNumber":"50825"}]}],"MessageType":"ExchangeSnapshotResult"}