
How To use GFDL WebSockets in Postman

Contents

SubscribeRealtime	5
SubscribeSnapshot	6
GetExchanges :	7
GetLastQuote	8
GetLastQuoteShort	9
GetLastQuoteShortWithClose	10
GetHistory	11
GetInstrumentsOnSearch	12
GetInstruments	13
GetInstrumentTypes	14
GetProducts	15
GetExpiryDates	16
GetOptionTypes	17
GetStrikePrices	18
GetServerInfo	19
GetLimitation	20
GetMarketMessages	21
GetExchangeMessages	22
GetLastQuoteOptionChain	23
GetExchangeSnapshot:	24
GetLastQuoteOptionGreeks:	25
GetLastQuoteArrayOptionGreeks	26
GetLastQuoteOptionGreeksChain	27

About Us

“We Are Low Latency Realtime Data Vendor Of Indian Stock Exchanges – Since 2010”

Global Financial Datafeeds LLP (GFDL) is a India-based professional Setup providing realtime data services to stock market traders – to help them take correct trading decisions. Founded in 2009 and having setup at Nashik, the company aims to revolutionize delivery of Financial Data in India.

GFDL is authorised realtime (L1) Data Vendor of NSE Data & Analytics Limited (formerly known as DotEx International Limited), the info-vending arm of [National Stock Exchange of India](#) (NSE) and [Multi Commodity Exchange of India](#) (MCX).

How To use GFDL WebSockets in Postman, Follow below sequence :

1.

Create a connection. To connect, use the endpoint given - as below :

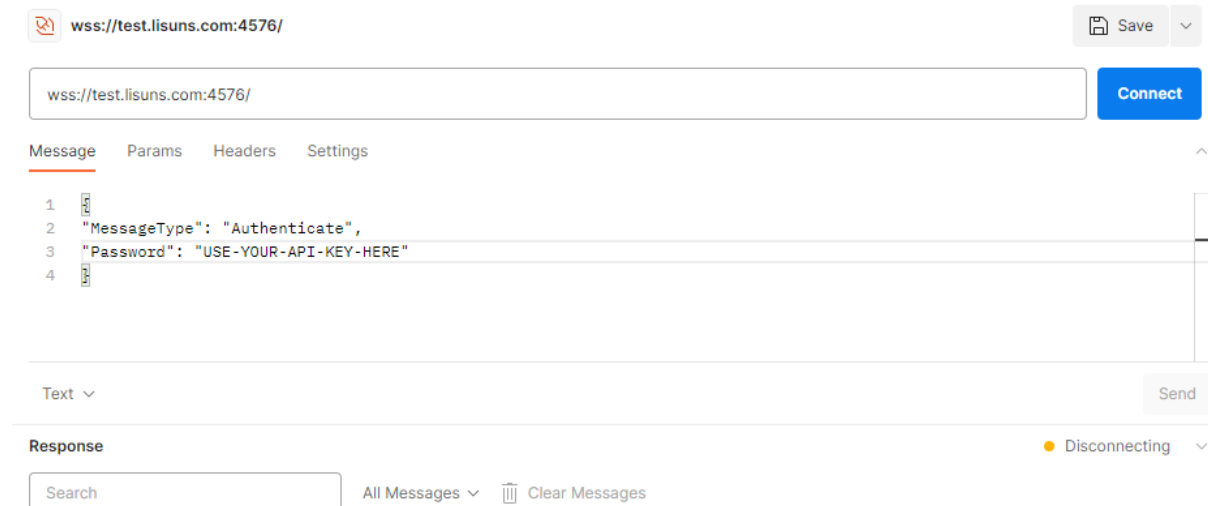
wss://test.lisuns.com:4576/

2.

First you need to authenticate. For that, you will need to send message as below :

```
{  
  MessageType: "Authenticate",  
  Password: "USE_YOUR_APIKEY_HERE"  
}
```

NOTE:DO NOT forget to replace the actual APIKey here : USE_YOUR_APIKEY_HERE

Request Body Window

The screenshot shows the Postman interface for a WebSocket connection. At the top, the URL bar contains 'wss://test.lisuns.com:4576/' with a 'Save' button to its right. Below the URL bar is a 'Connect' button. The 'Message' tab is selected, showing a JSON body with the following content:

```
1 {  
2   "MessageType": "Authenticate",  
3   "Password": "USE-YOUR-API-KEY-HERE"  
4 }
```

Below the message editor, there is a 'Text' dropdown and a 'Send' button. At the bottom, the 'Response' section is visible, showing a 'Search' input, 'All Messages' dropdown, and 'Clear Messages' button. A status indicator shows 'Disconnecting'.

Response window

The screenshot shows the Postman application interface. At the top, the URL bar displays `wss://test.lisuns.com:4576/`. Below the URL bar, the 'Message' tab is selected, showing a JSON request body:

```
{
  "MessageType": "Authenticate",
  "Password": "XXXXXXXXXXXXXXXXXXXX"
}
```

Below the request, the 'Response' tab is selected, showing a list of messages. The first message is a 'New messages' button. The second message is a JSON response:

```
{
  "Complete": true,
  "Message": "Welcome!",
  "MessageType": "AuthenticateResult"
}
```

The response is displayed in the 'JSON' tab, showing the same JSON structure. The status bar at the bottom indicates 'Connected' and 'Show Hexdump' is visible.

After authentication, you can send any requests. For example

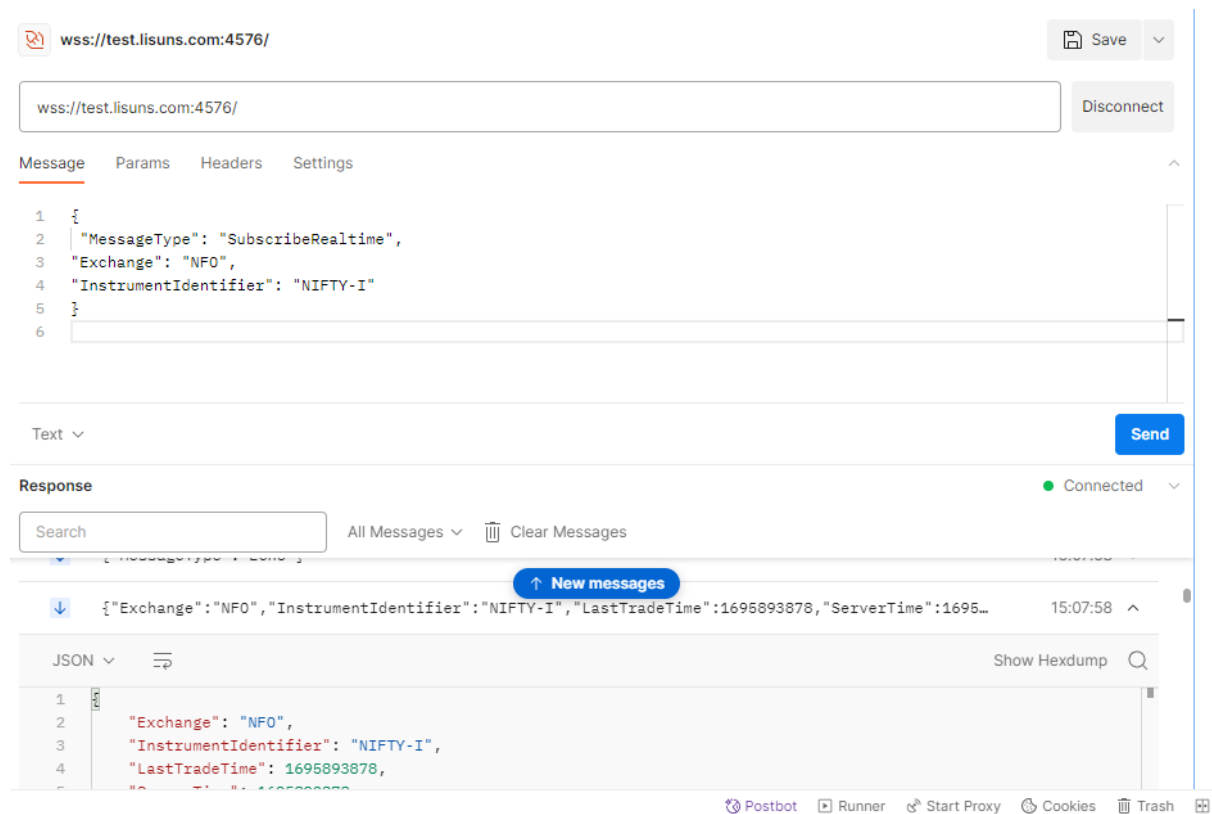
SubscribeRealtime

Subscribe to Realtime data, returns market data every second (Bid/Ask/Trade)

```
{  
  "MessageType": "SubscribeRealtime",  
  "Exchange": "NFO",  
  "InstrumentIdentifier": "NIFTY-I"  
}
```

For detailed description please refer below link.

[SubscribeRealtime](#) | [Global Datafeeds](#)



The screenshot displays a web-based WebSocket client interface. At the top, the URL bar shows `wss://test.lisuns.com:4576/`. Below the URL bar, there are tabs for "Message", "Params", "Headers", and "Settings". The "Message" tab is active, showing a JSON object: `{ "MessageType": "SubscribeRealtime", "Exchange": "NFO", "InstrumentIdentifier": "NIFTY-I" }`. To the right of the message input, there are "Save" and "Disconnect" buttons. Below the message input, there is a "Text" dropdown and a "Send" button. The "Response" section is visible, showing a "Connected" status. Below the response section, there is a "New messages" button. The response area shows a JSON object: `{ "Exchange": "NFO", "InstrumentIdentifier": "NIFTY-I", "LastTradeTime": 1695893878, "ServerTime": 1695893878, "Bid": 1695893878, "Ask": 1695893878, "Trade": 1695893878 }`. At the bottom of the interface, there are several utility buttons: "Postbot", "Runner", "Start Proxy", "Cookies", "Trash", and a grid icon.

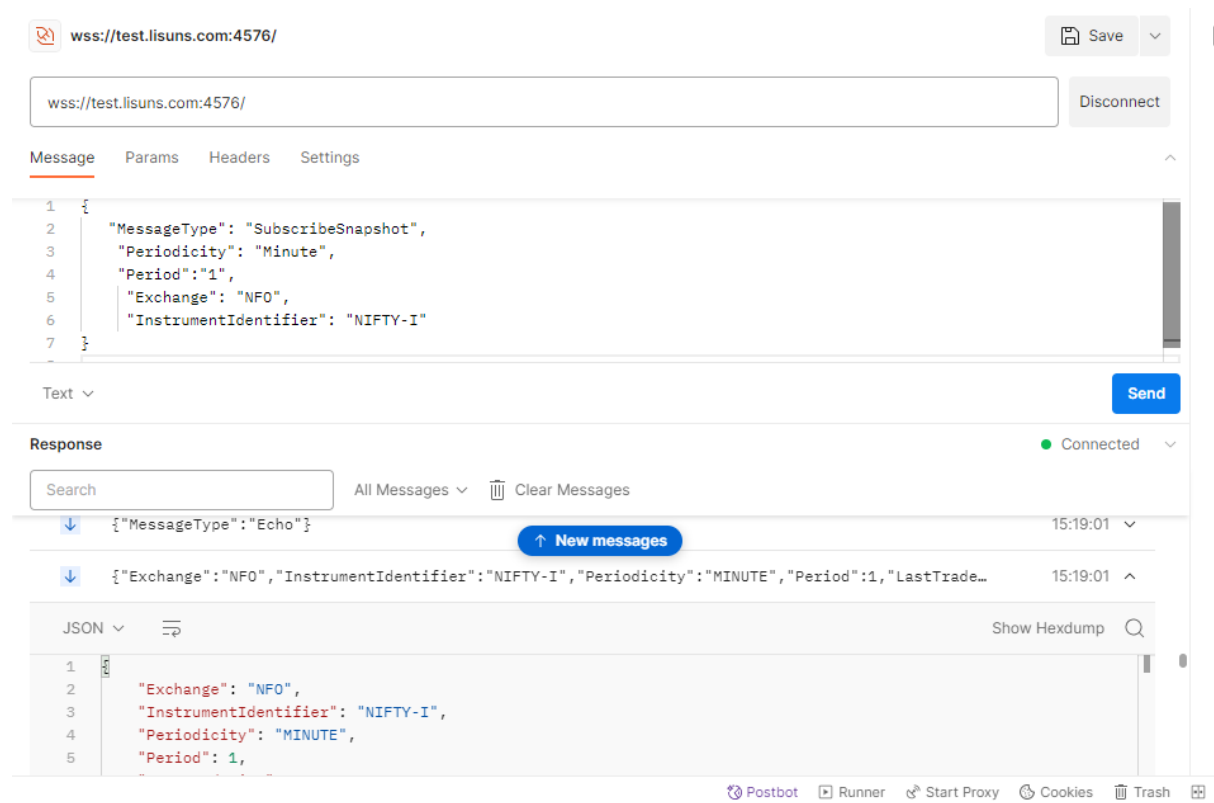
SubscribeSnapshot

Subscribe to Realtime Snapshots data, returns data snapshot as per Periodicity & Period values

```
{  
  "MessageType": "SubscribeSnapshot",  
  "Periodicity": "Minute",  
  "Period": "1",  
  "Exchange": "NFO",  
  "InstrumentIdentifier": "NIFTY-I"  
}
```

For detailed description please refer below link.

[SubscribeSnapshot | Global Datafeeds](#)



The screenshot displays a web application interface for testing WebSocket connections. The URL bar shows `wss://test.lisuns.com:4576/`. Below the URL bar, there is a 'Send' button and a 'Disconnect' button. The 'Message' tab is selected, showing a JSON message:

```
{  
  "MessageType": "SubscribeSnapshot",  
  "Periodicity": "Minute",  
  "Period": "1",  
  "Exchange": "NFO",  
  "InstrumentIdentifier": "NIFTY-I"  
}
```

. The 'Response' section shows a list of messages, including an 'Echo' message and a message with the same JSON structure as the one in the 'Message' tab. The interface also includes a 'Search' bar, a 'New messages' button, and a 'Show Hexdump' option.

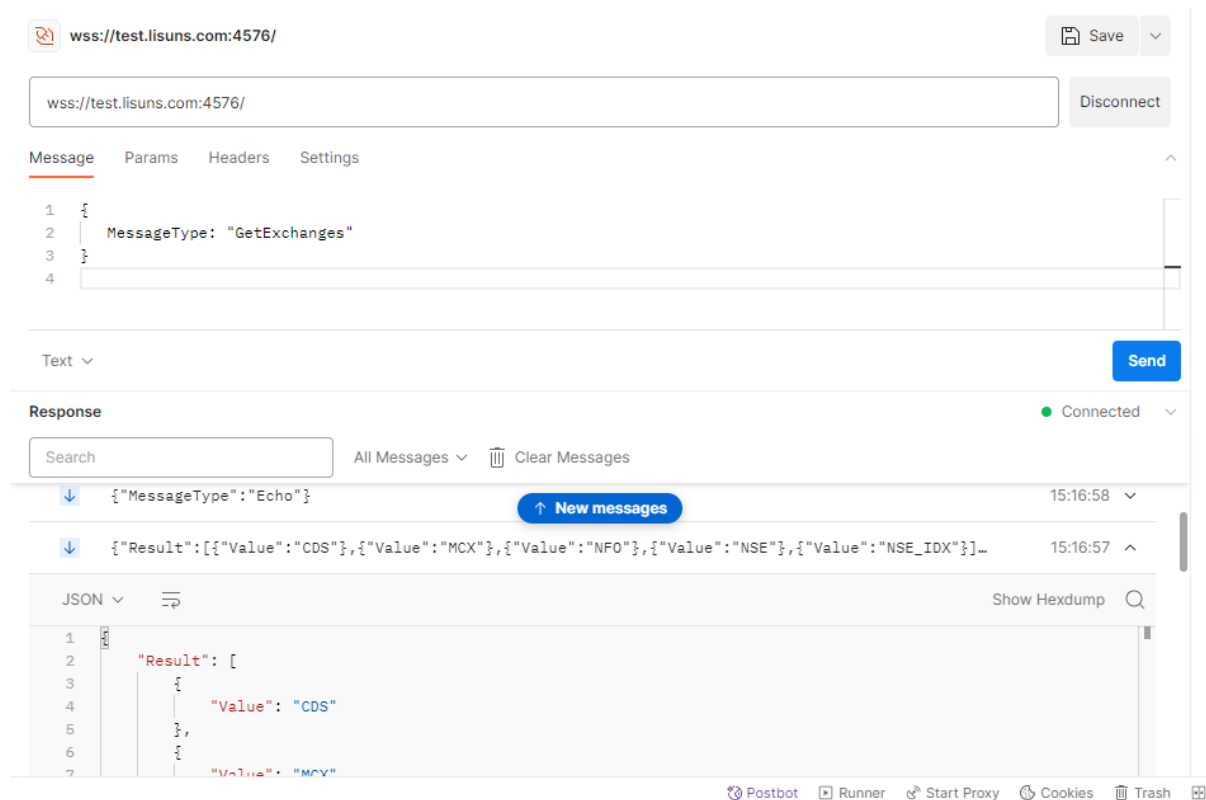
GetExchanges

Returns array of available exchanges configured for API Key

```
{  
  MessageType: "GetExchanges"  
}
```

For detailed description please refer below link.

[GetExchanges | Global Datafeeds](#)



The screenshot displays a web-based API client interface. At the top, the URL bar shows `wss://test.lisuns.com:4576/`. Below the URL bar, there are tabs for `Message`, `Params`, `Headers`, and `Settings`. The `Message` tab is active, showing a JSON message: `{ "MessageType": "GetExchanges" }`. Below the message, there is a `Text` input field and a `Send` button. The `Response` section shows a list of messages. The first message is `{ "MessageType": "Echo" }` with a timestamp of 15:16:58. The second message is `{ "Result": [{ "Value": "CDS" }, { "Value": "MCX" }, { "Value": "NFO" }, { "Value": "NSE" }, { "Value": "NSE_INDEX" }] }` with a timestamp of 15:16:57. Below the response list, there is a `JSON` tab and a `Show Hexdump` button. The `JSON` tab is active, showing the response in a formatted JSON view:

```
{  
  "Result": [  
    {  
      "Value": "CDS"  
    },  
    {  
      "Value": "MCX"  
    },  
    {  
      "Value": "NFO"  
    },  
    {  
      "Value": "NSE"  
    },  
    {  
      "Value": "NSE_INDEX"  
    }  
  ]  
}
```

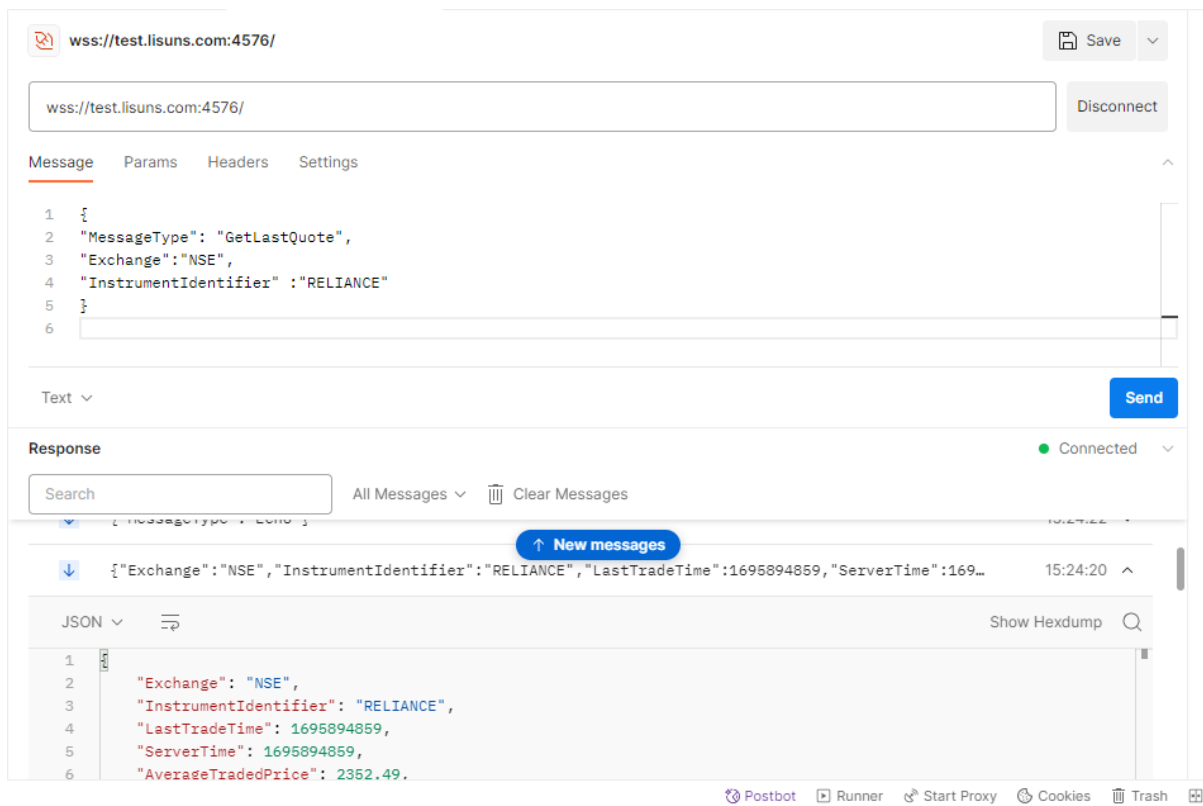
GetLastQuote

Returns LastTradePrice of Single Symbol (detailed)

```
{  
  "MessageType": "GetLastQuote",  
  "Exchange": "NSE",  
  "InstrumentIdentifier": "RELIANCE"  
}
```

For detailed description please refer below link.

[GetLastQuote | Global Datafeeds](#)



The screenshot shows a web application interface for a WebSocket connection. The address bar displays `wss://test.lisuns.com:4576/`. The interface includes a "Save" button, a "Disconnect" button, and tabs for "Message", "Params", "Headers", and "Settings". The "Message" tab is active, showing a JSON message being sent:

```
1 {  
2   "MessageType": "GetLastQuote",  
3   "Exchange": "NSE",  
4   "InstrumentIdentifier": "RELIANCE"  
5 }  
6
```

Below the message, there is a "Text" input field and a "Send" button. The "Response" section shows a "Connected" status and a search bar. A "New messages" button is visible. The response is displayed in a JSON view, showing the following data:

```
1 {  
2   "Exchange": "NSE",  
3   "InstrumentIdentifier": "RELIANCE",  
4   "LastTradeTime": 1695894859,  
5   "ServerTime": 1695894859,  
6   "AverageTradedPrice": 2352.49
```

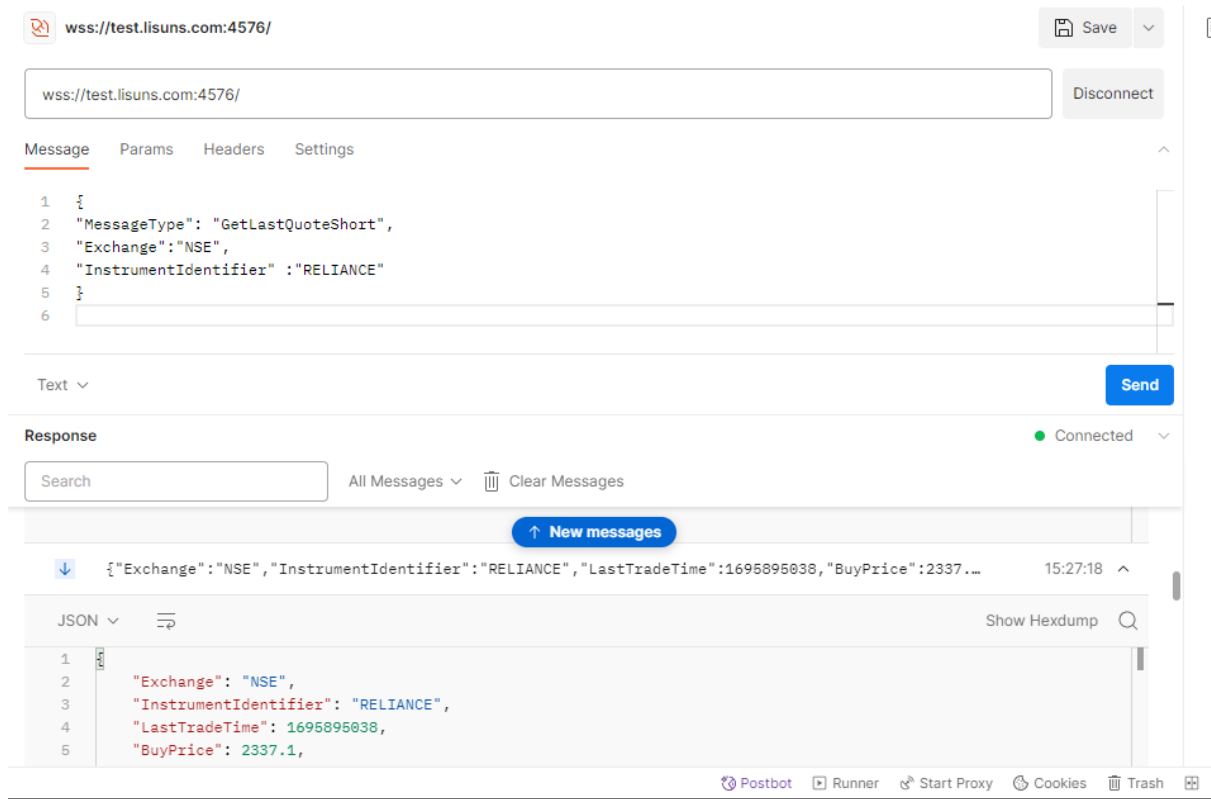

GetLastQuoteShort

Returns LastTradePrice of Single Symbol (short)

```
{  
  "MessageType": "GetLastQuoteShort",  
  "Exchange": "NSE",  
  "InstrumentIdentifier": "RELIANCE"  
}
```

For detailed description please refer below link.

[GetLastQuoteShort | Global Datafeeds](#)



The screenshot displays a web application interface for testing WebSocket messages. The top section shows the URL `wss://test.lisuns.com:4576/` and a `Save` button. Below this is a text input field containing the same URL and a `Disconnect` button. The interface is divided into tabs: `Message`, `Params`, `Headers`, and `Settings`. The `Message` tab is active, showing a JSON message being sent:

```
1 {  
2   "MessageType": "GetLastQuoteShort",  
3   "Exchange": "NSE",  
4   "InstrumentIdentifier": "RELIANCE"  
5 }  
6
```

Below the message input is a `Text` dropdown and a `Send` button. The `Response` section shows a `Connected` status. It includes a search bar, a dropdown for `All Messages`, and a `Clear Messages` button. A `New messages` button is also present. The response is displayed in a list, showing a message received at `15:27:18`:

```
{ "Exchange": "NSE", "InstrumentIdentifier": "RELIANCE", "LastTradeTime": 1695895038, "BuyPrice": 2337.1, "LastTradePrice": 2337.1 }
```

The response is shown in `JSON` format, with a `Show Hexdump` button. The bottom of the interface features a navigation bar with icons for `Postbot`, `Runner`, `Start Proxy`, `Cookies`, `Trash`, and a grid icon.

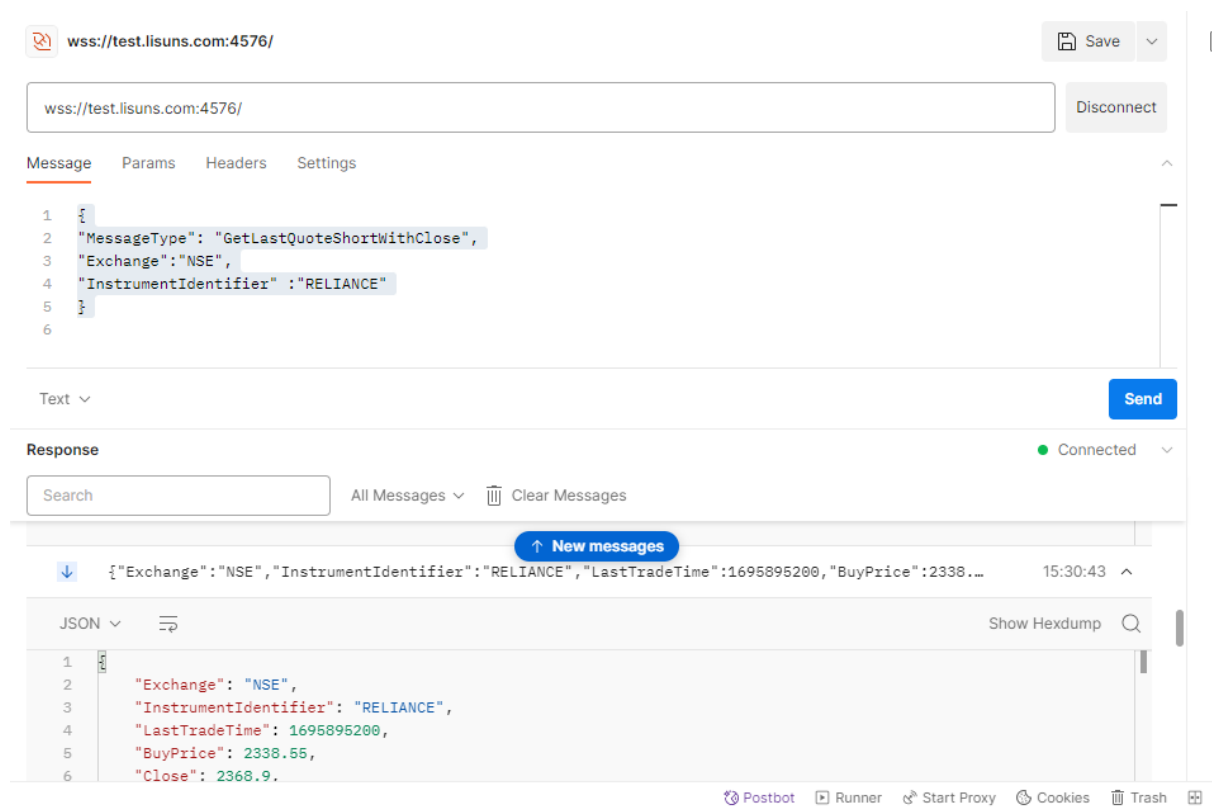
GetLastQuoteShortWithClose

Returns LastTradePrice of Single Symbol (short) with Close of Previous Day

```
{  
  "MessageType": "GetLastQuoteShortWithClose",  
  "Exchange": "NSE",  
  "InstrumentIdentifier": "RELIANCE"  
}
```

For detailed description please refer below link.

[GetLastQuoteShortWithClose | Global Datafeeds](#)



The screenshot displays a web-based WebSocket client interface. At the top, the URL `wss://test.lisuns.com:4576/` is entered in the address bar. Below the address bar, there are tabs for "Message", "Params", "Headers", and "Settings". The "Message" tab is active, showing a JSON message being sent:

```
{  
  "MessageType": "GetLastQuoteShortWithClose",  
  "Exchange": "NSE",  
  "InstrumentIdentifier": "RELIANCE"  
}
```

Below the message input area, there is a "Text" dropdown and a "Send" button. The "Response" section shows a "Connected" status. Below the response area, there is a search bar and a "Clear Messages" button. A "New messages" button is also present. The response message is displayed in a JSON view:

```
{  
  "Exchange": "NSE",  
  "InstrumentIdentifier": "RELIANCE",  
  "LastTradeTime": 1695895200,  
  "BuyPrice": 2338.55,  
  "Close": 2368.9  
}
```

At the bottom of the interface, there are several utility buttons: "Postbot", "Runner", "Start Proxy", "Cookies", "Trash", and a settings icon.

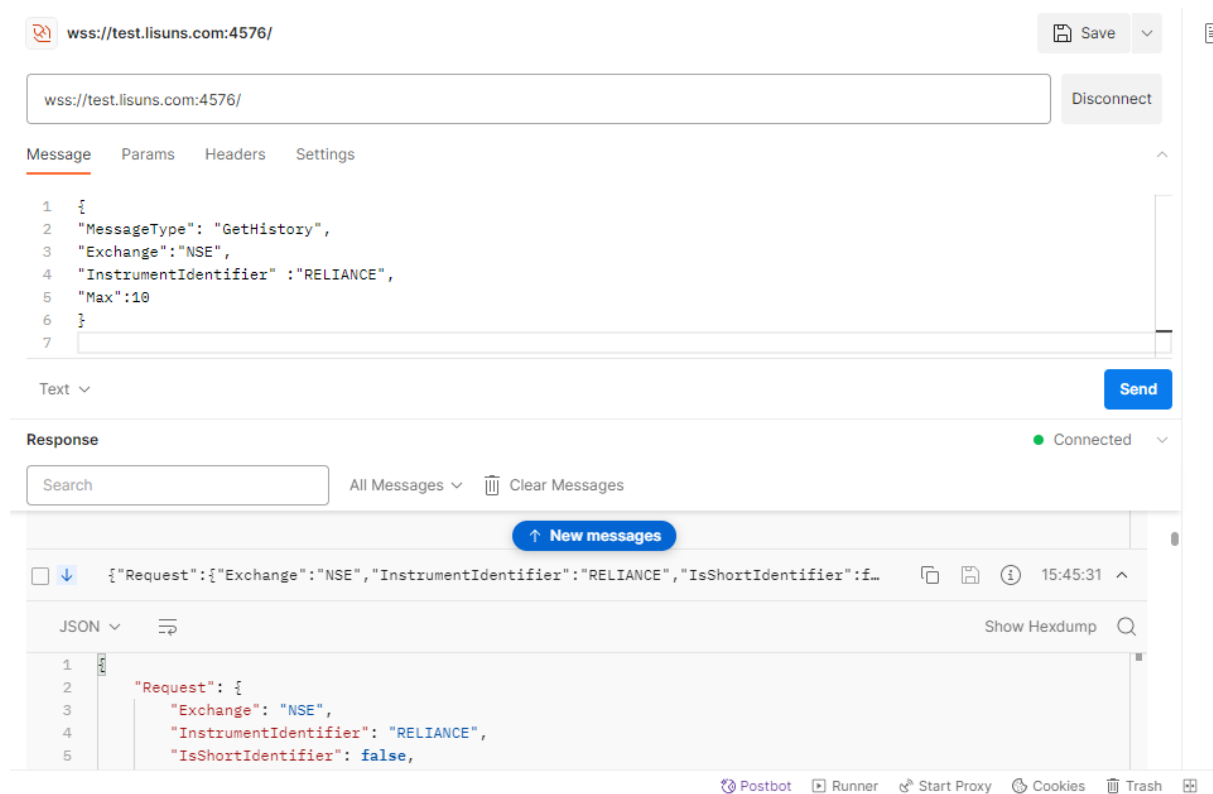
GetHistory

Returns historical data (Tick / Minute / EOD)

```
{  
  "MessageType": "GetHistory",  
  "Exchange": "NSE",  
  "InstrumentIdentifier": "RELIANCE",  
  "Max": 10  
}
```

For detailed description please refer below link.

[GetHistory | Global Datafeeds](#)



The screenshot displays a web-based WebSocket client interface. At the top, the URL bar shows `wss://test.lisuns.com:4576/`. Below the URL bar, there is a "Disconnect" button. The interface has tabs for "Message", "Params", "Headers", and "Settings". The "Message" tab is active, showing a JSON message in a text area with line numbers 1 through 7. The message is:

```
{  
  "MessageType": "GetHistory",  
  "Exchange": "NSE",  
  "InstrumentIdentifier": "RELIANCE",  
  "Max": 10  
}
```

 Below the message area, there is a "Text" dropdown and a "Send" button. The "Response" section shows a "Connected" status with a green dot. Below this, there is a search bar and a "Clear Messages" button. A "New messages" button is also present. The response area shows a list of messages, with the first one selected. The selected message is a JSON object:

```
{  
  "Request": {  
    "Exchange": "NSE",  
    "InstrumentIdentifier": "RELIANCE",  
    "IsShortIdentifier": false,  
    "Max": 10  
  }  
}
```

 The interface also includes a "Show Hexdump" button and a bottom toolbar with icons for Postbot, Runner, Start Proxy, Cookies, Trash, and a settings icon.

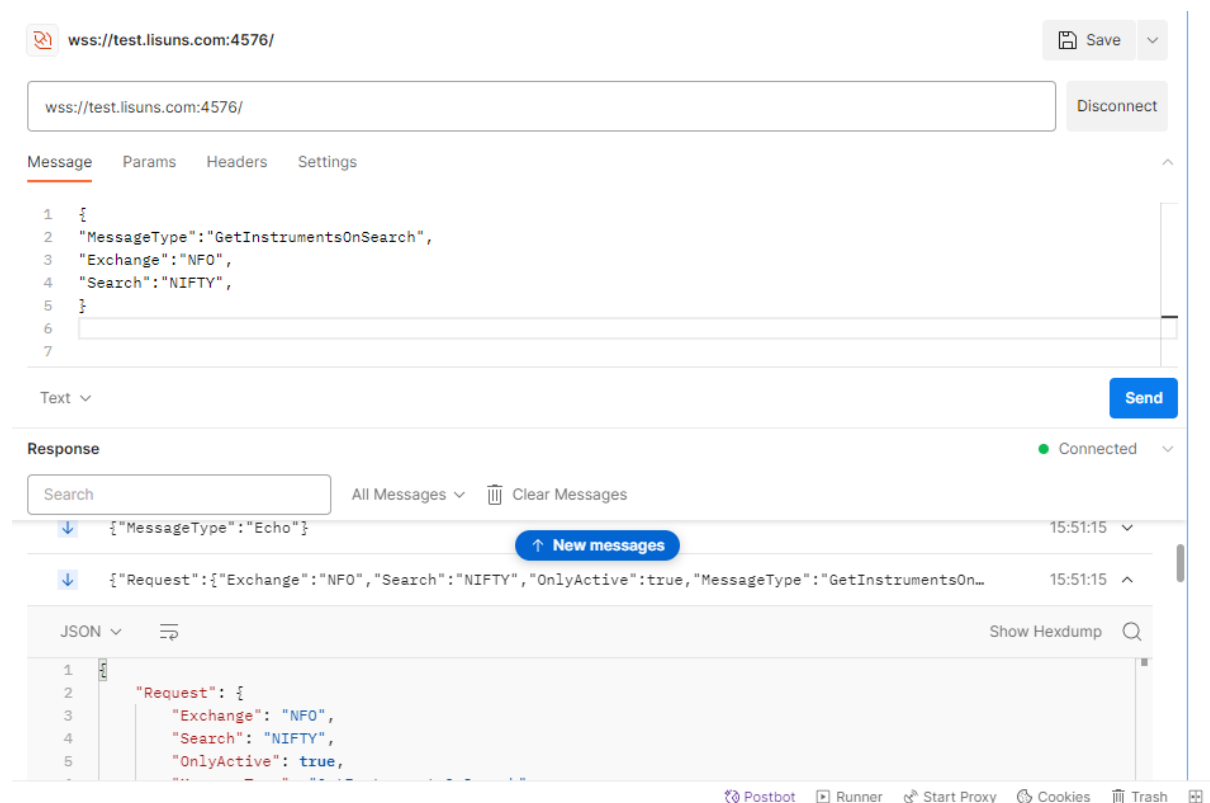
GetInstrumentsOnSearch

Returns array of max. 20 instruments by selected exchange and 'search string'

```
{  
  "MessageType": "GetInstrumentsOnSearch",  
  "Exchange": "NFO",  
  "Search": "NIFTY",  
}
```

For detailed description please refer below link.

[GetInstrumentsOnSearch | Global Datafeeds](#)



The screenshot displays a WebSocket client interface. At the top, the URL bar shows `wss://test.lisuns.com:4576/`. Below the URL bar, there are tabs for `Message`, `Params`, `Headers`, and `Settings`. The `Message` tab is active, showing a JSON message being sent:

```
{  
  "MessageType": "GetInstrumentsOnSearch",  
  "Exchange": "NFO",  
  "Search": "NIFTY",  
}
```

. A `Send` button is visible. Below the message input, there is a `Response` section with a search bar and a `Clear Messages` button. The response list shows two messages: an `Echo` message and a `GetInstrumentsOnSearch` message. The `GetInstrumentsOnSearch` message is expanded, showing its JSON structure:

```
{  
  "Request": {  
    "Exchange": "NFO",  
    "Search": "NIFTY",  
    "OnlyActive": true,  
  },  
}
```

. The interface also includes a `Connected` status indicator and a `Show Hexdump` button.

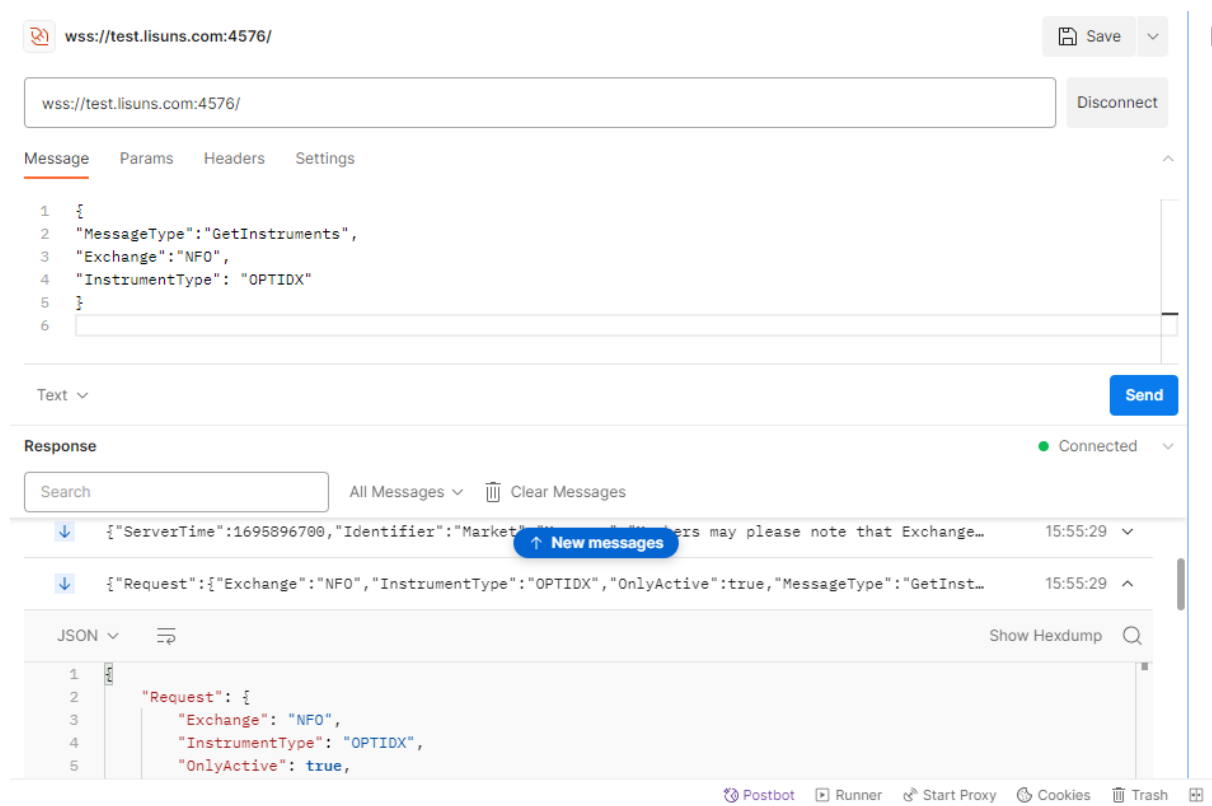
GetInstruments

Returns array of instruments by selected exchange

```
{  
  "MessageType": "GetInstruments",  
  "Exchange": "NFO",  
  "InstrumentType": "OPTIDX"  
}
```

For detailed description please refer below link.

[GetInstruments | Global Datafeeds](#)



The screenshot displays a web browser interface for a WebSocket client. The address bar shows the URL `wss://test.lisuns.com:4576/`. The interface includes a 'Message' tab with a JSON request, a 'Response' tab showing a JSON response, and a 'Send' button. The response is a JSON object with 'Request' and 'OnlyActive' fields.

```
1 {  
2   "MessageType": "GetInstruments",  
3   "Exchange": "NFO",  
4   "InstrumentType": "OPTIDX"  
5 }  
6
```

Text

Send

Response

Search All Messages Clear Messages

↑ New messages

```
1 {  
2   "Request": {  
3     "Exchange": "NFO",  
4     "InstrumentType": "OPTIDX",  
5     "OnlyActive": true,  
6   }  
7 }
```

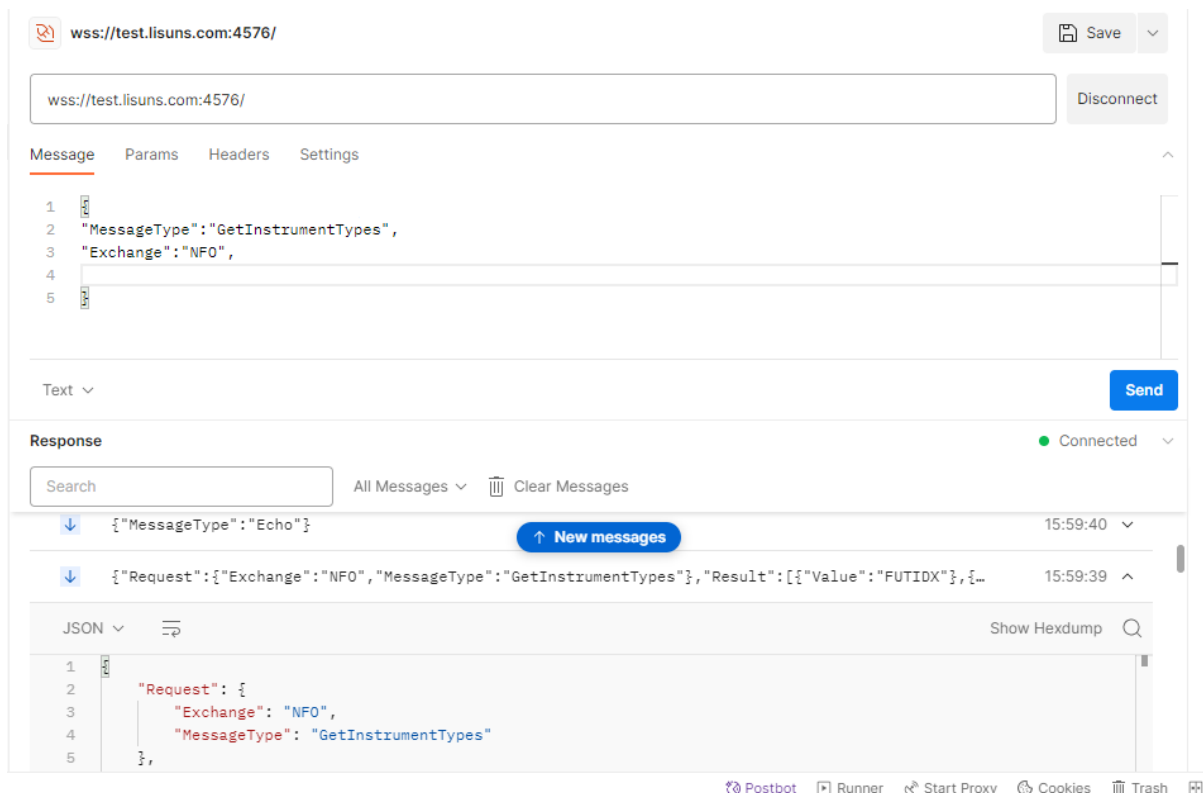
GetInstrumentTypes

Returns list of Instrument Types (e.g. FUTIDX, FUTSTK, etc.)

```
{  
  "MessageType": "GetInstrumentTypes",  
  "Exchange": "NFO",  
}
```

For detailed description please refer below link.

[GetInstrumentTypes | Global Datafeeds](#)



The screenshot displays a web-based WebSocket client interface. At the top, the connection URL is `wss://test.lisuns.com:4576/`. The interface includes tabs for Message, Params, Headers, and Settings. The Message tab is active, showing a JSON request: `{ "MessageType": "GetInstrumentTypes", "Exchange": "NFO" }`. Below the message input, there is a "Send" button. The Response section shows a list of messages. The most recent message is a response from the server: `{ "Request": { "Exchange": "NFO", "MessageType": "GetInstrumentTypes" }, "Result": [ { "Value": "FUTIDX" }, { ... } ] }`. The interface also includes a "New messages" button and a "Show Hexdump" option. At the bottom, there are links for Postbot, Runner, Start Proxy, Cookies, and Trash.

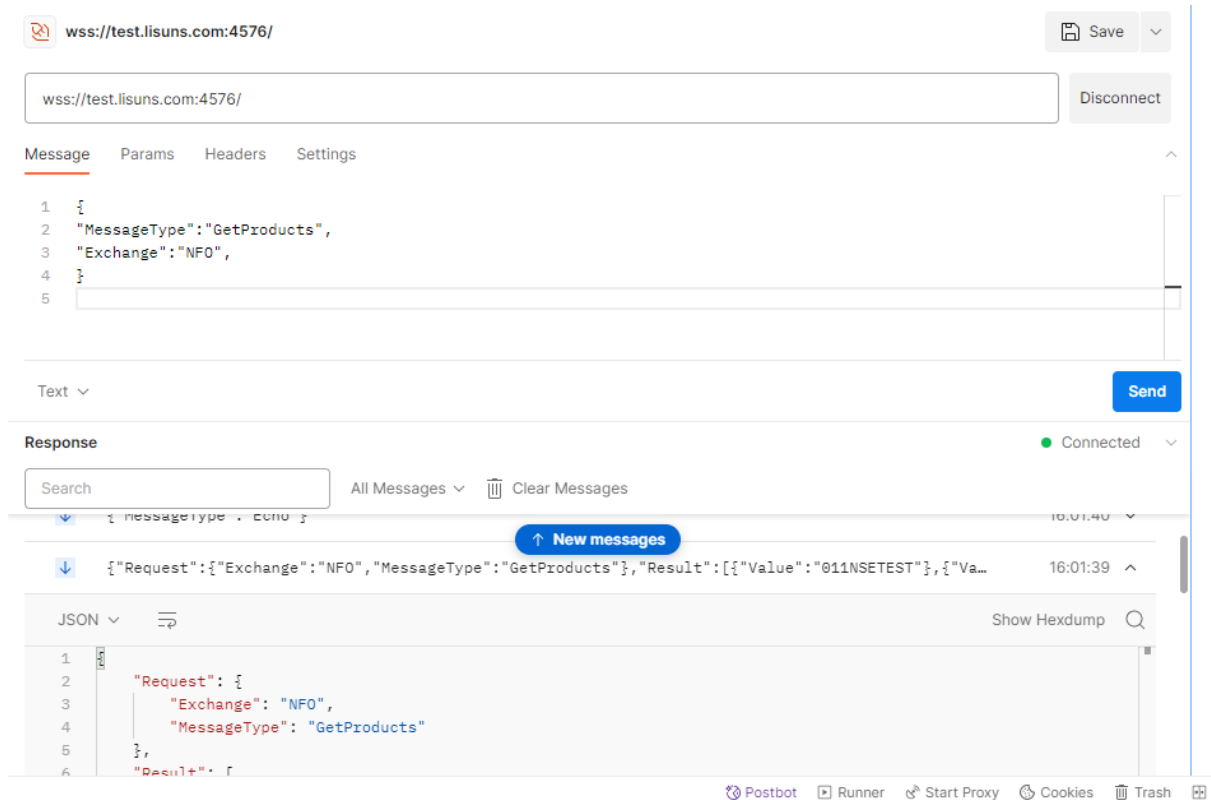
GetProducts

Returns list of Products (e.g. NIFTY, BANKNIFTY, GAIL, etc.)

```
{  
  "MessageType": "GetProducts",  
  "Exchange": "NFO",  
}
```

For detailed description please refer below link.

[GetProducts | Global Datafeeds](#)



The screenshot displays a web application for testing WebSocket messages. The address bar shows `wss://test.lisuns.com:4576/`. The interface includes a 'Save' button, a 'Disconnect' button, and tabs for 'Message', 'Params', 'Headers', and 'Settings'. The 'Message' tab is active, showing a JSON payload:

```
1 {  
2   "MessageType": "GetProducts",  
3   "Exchange": "NFO",  
4 }  
5
```

Below the message input is a 'Text' dropdown and a 'Send' button. The 'Response' section shows a 'Connected' status and a search bar. A 'New messages' button is visible. The response is displayed in JSON format:

```
1 {  
2   "Request": {  
3     "Exchange": "NFO",  
4     "MessageType": "GetProducts"  
5   },  
6   "Result": [
```

The bottom of the interface features a navigation bar with icons for Postbot, Runner, Start Proxy, Cookies, Trash, and a grid icon.

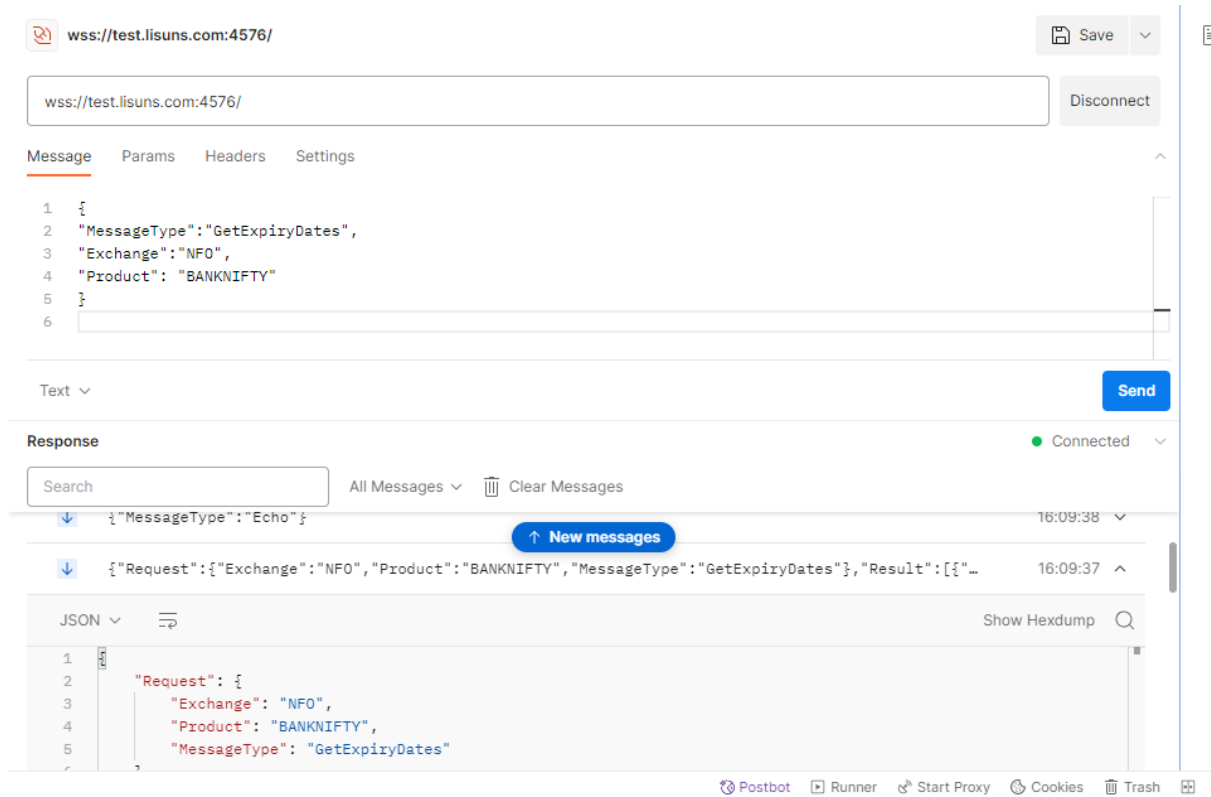
GetExpiryDates

Returns array of Expiry Dates (e.g. 25JUN2020, 30JUL2020, etc.)

```
{  
  "MessageType": "GetExpiryDates",  
  "Exchange": "NFO",  
  "Product": "BANKNIFTY"  
}
```

For detailed description please refer below link.

[GetExpiryDates | Global Datafeeds](#)



The screenshot displays a web-based WebSocket client interface. At the top, the URL `wss://test.lisuns.com:4576/` is entered in the address bar. Below the address bar, there are tabs for **Message**, **Params**, **Headers**, and **Settings**. The **Message** tab is selected, showing a JSON request:

```
1 {  
2   "MessageType": "GetExpiryDates",  
3   "Exchange": "NFO",  
4   "Product": "BANKNIFTY"  
5 }  
6
```

Below the message input, there is a **Send** button. The **Response** section shows a **Connected** status. Below the response section, there is a search bar and a list of messages. The first message is a JSON response:

```
1 {  
2   "Request": {  
3     "Exchange": "NFO",  
4     "Product": "BANKNIFTY",  
5     "MessageType": "GetExpiryDates"  
6   },  
7   "Result": [...]  
8 }
```

The interface also includes a **New messages** button and a **Show Hexdump** option. At the bottom, there are icons for **Postbot**, **Runner**, **Start Proxy**, **Cookies**, **Trash**, and a **Grid** icon.

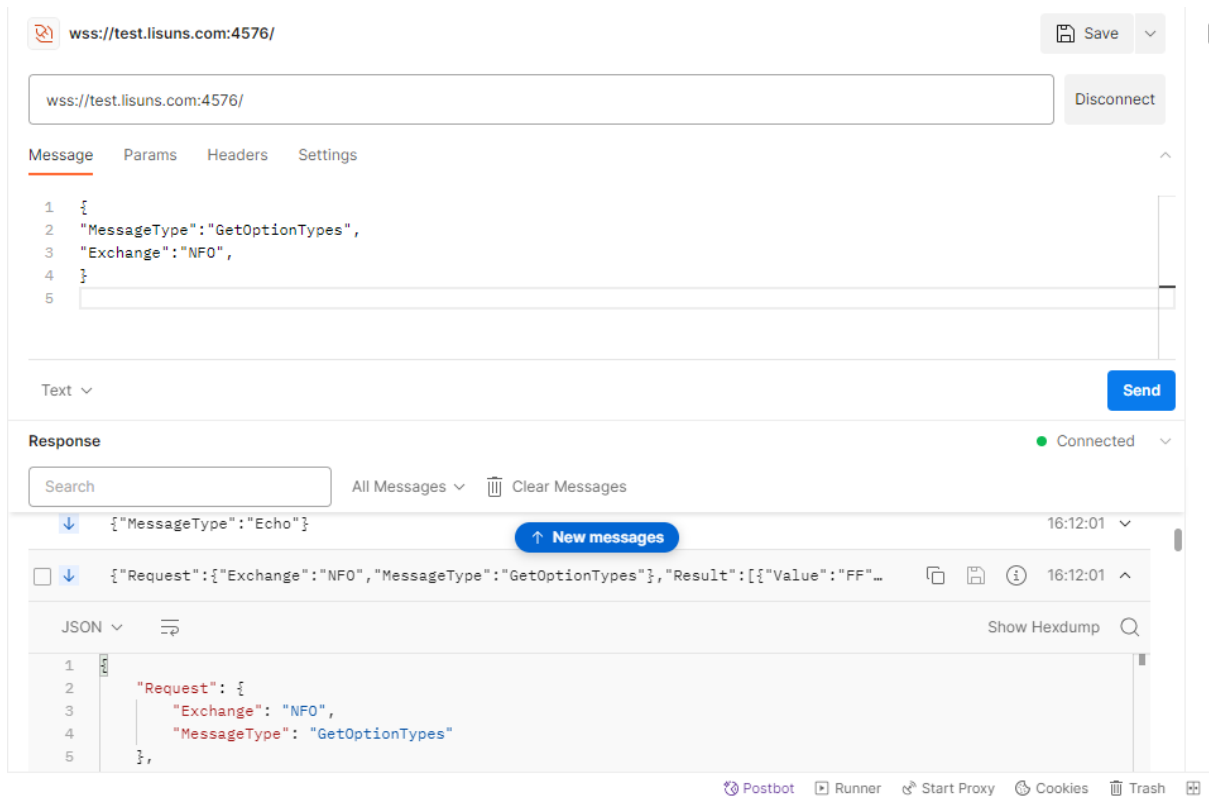
GetOptionTypes

Returns list of Option Types (e.g. CE, PE, etc.)

```
{  
  "MessageType": "GetOptionTypes",  
  "Exchange": "NFO",  
}
```

For detailed description please refer below link.

[GetOptionTypes | Global Datafeeds](#)



The screenshot displays a REST client interface for a WebSocket endpoint. The URL bar shows `wss://test.lisuns.com:4576/`. The interface includes a "Save" button and a "Disconnect" button. The "Message" tab is active, showing a JSON request:

```
{  
  "MessageType": "GetOptionTypes",  
  "Exchange": "NFO",  
}
```

. The "Response" tab is also active, showing a JSON response:

```
{  
  "Request": {  
    "Exchange": "NFO",  
    "MessageType": "GetOptionTypes"  
  },  
  "Result": [{  
    "Value": "FF"  
  }]  
}
```

. The interface includes a "Send" button and a "New messages" button. The status bar at the bottom shows "Connected" and "Show Hexdump" options.

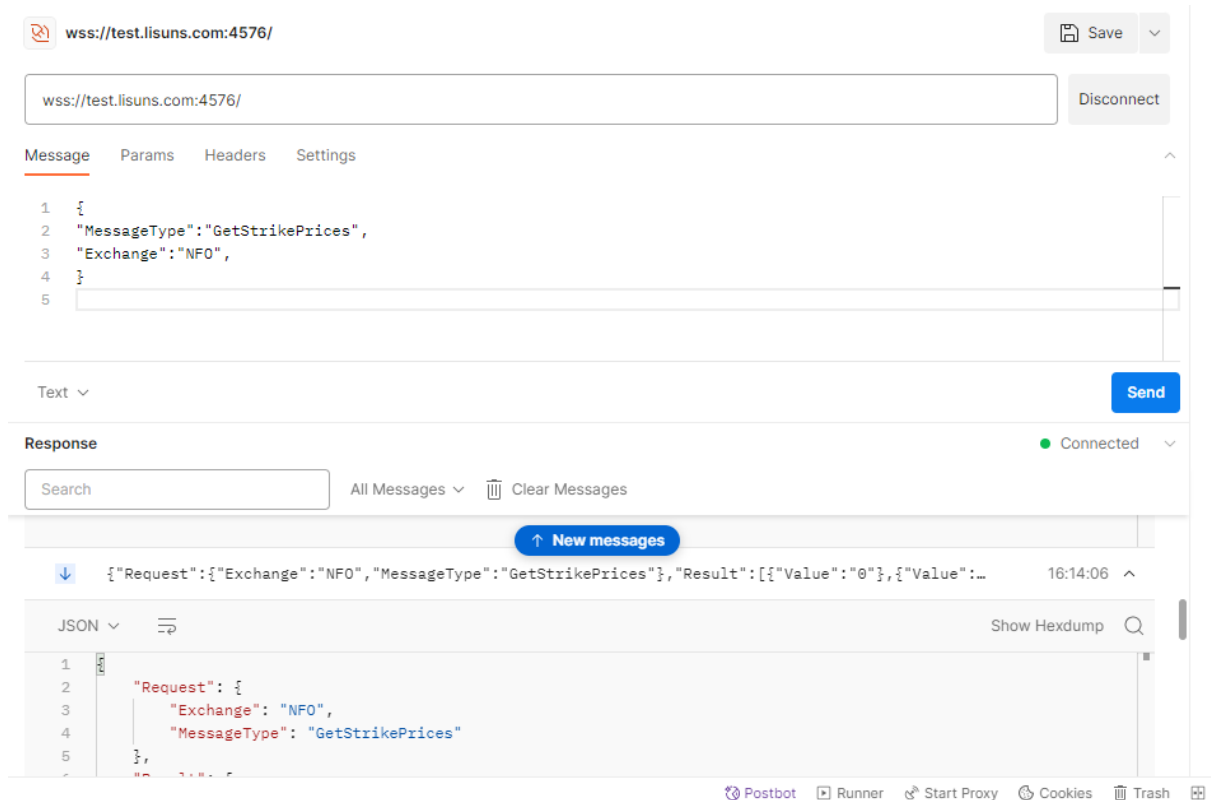
GetStrikePrices

Returns list of Strike Prices (e.g. 10000, 11000, 75.5, etc.)

```
{  
  "MessageType": "GetStrikePrices",  
  "Exchange": "NFO",  
}
```

For detailed description please refer below link.

[GetStrikePrices | Global Datafeeds](#)



The screenshot displays a web browser interface for a WebSocket client. The address bar shows the URL `wss://test.lisuns.com:4576/`. The 'Message' tab is selected, showing a JSON request:

```
{  
  "MessageType": "GetStrikePrices",  
  "Exchange": "NFO",  
}
```

. The 'Response' tab is also visible, showing a JSON response:

```
{  
  "Request": {  
    "Exchange": "NFO",  
    "MessageType": "GetStrikePrices"  
  },  
  "Result": [{  
    "Value": "0",  
    "Value": "..."  
  }]  
}
```

. The interface includes a 'Send' button and a 'Connected' status indicator. The bottom of the browser shows various extensions like Postbot, Runner, Start Proxy, Cookies, and Trash.

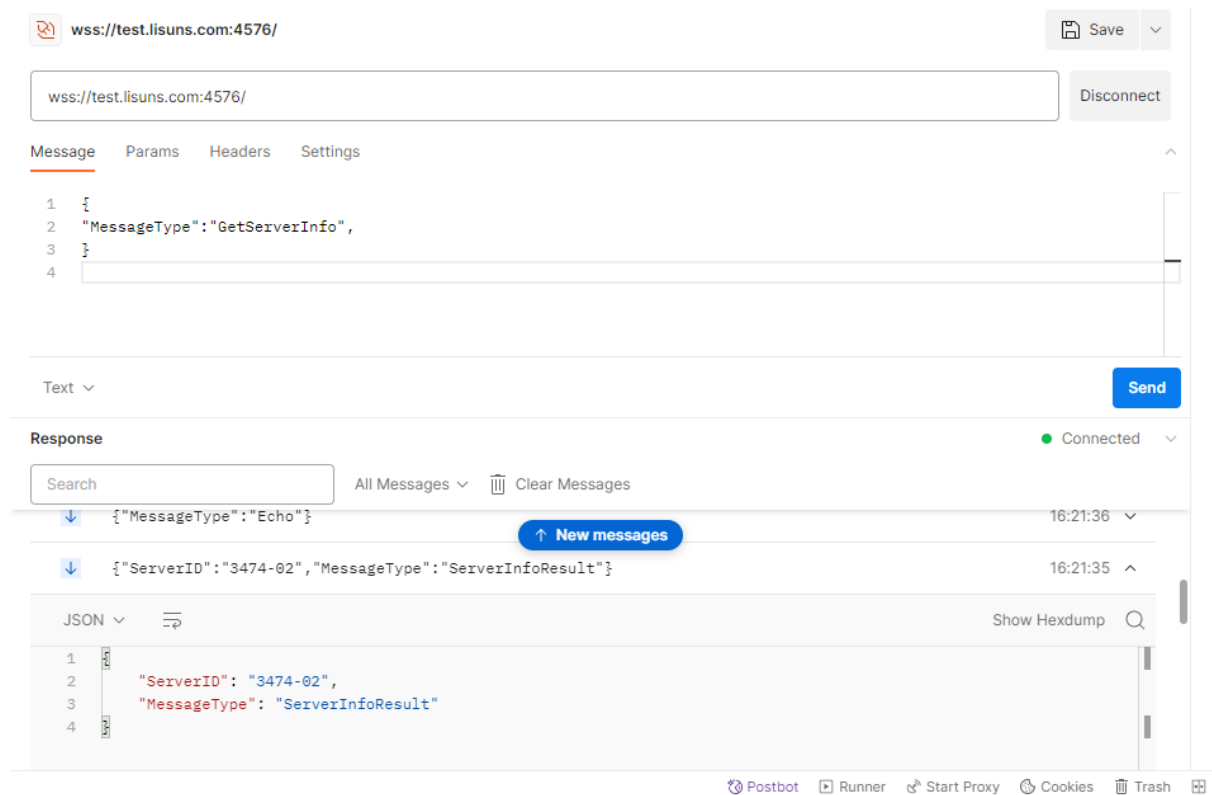
GetServerInfo

Returns information about server where connection is made

```
{  
  "MessageType": "GetServerInfo",  
}
```

For detailed description please refer below link.

[GetServerInfo | Global Datafeeds](#)



wss://test.lisuns.com:4576/ Save Disconnect

Message Params Headers Settings

```
1 {  
2   "MessageType": "GetServerInfo",  
3 }  
4
```

Text Send

Response Connected

Search All Messages Clear Messages

↓ { "MessageType": "Echo" } 16:21:36

↑ New messages

↓ { "ServerID": "3474-02", "MessageType": "ServerInfoResult" } 16:21:35

JSON Show Hexdump

```
1 {  
2   "ServerID": "3474-02",  
3   "MessageType": "ServerInfoResult"  
4 }
```

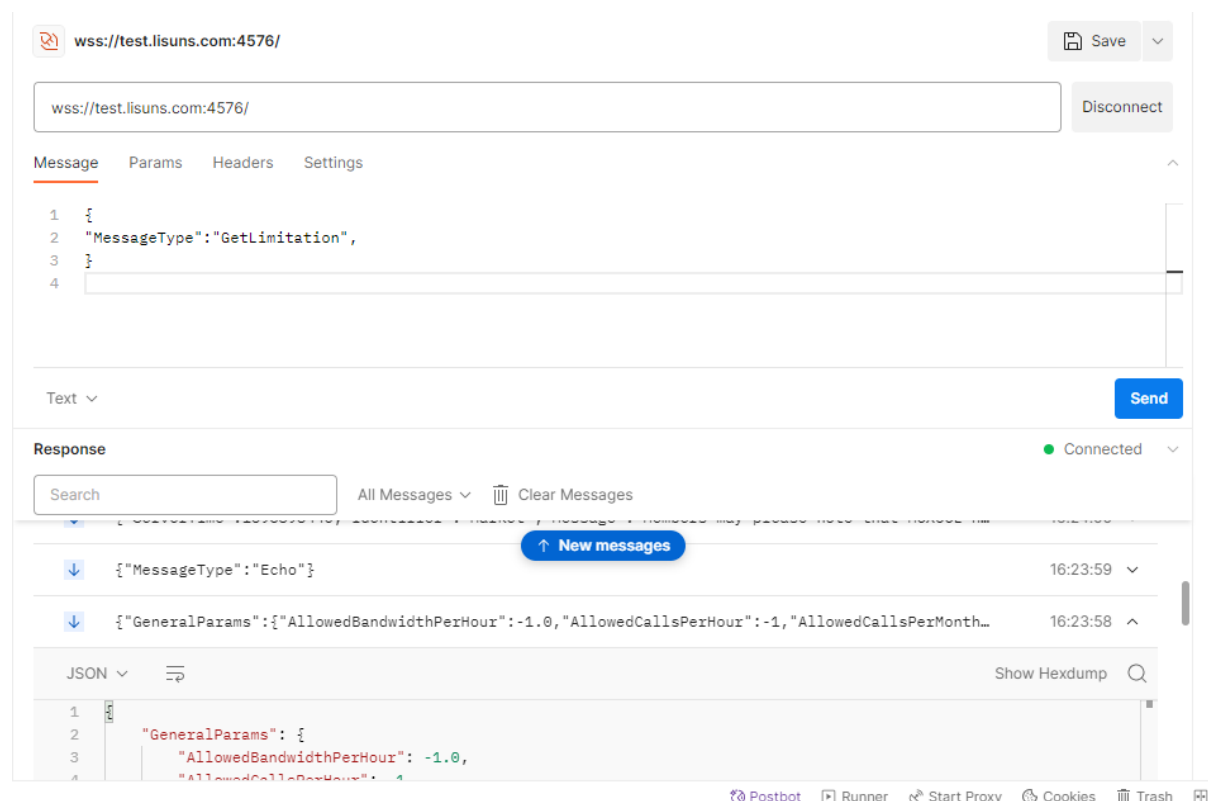
Postbot Runner Start Proxy Cookies Trash

GetLimitation

Returns user account information (e.g. which functions are allowed, Exchanges allowed, symbol limit, etc.)

```
{
  "MessageType": "GetLimitation",
}
```

For detailed description please refer below link.

[GetLimitation](#) | [Global Datafeeds](#)

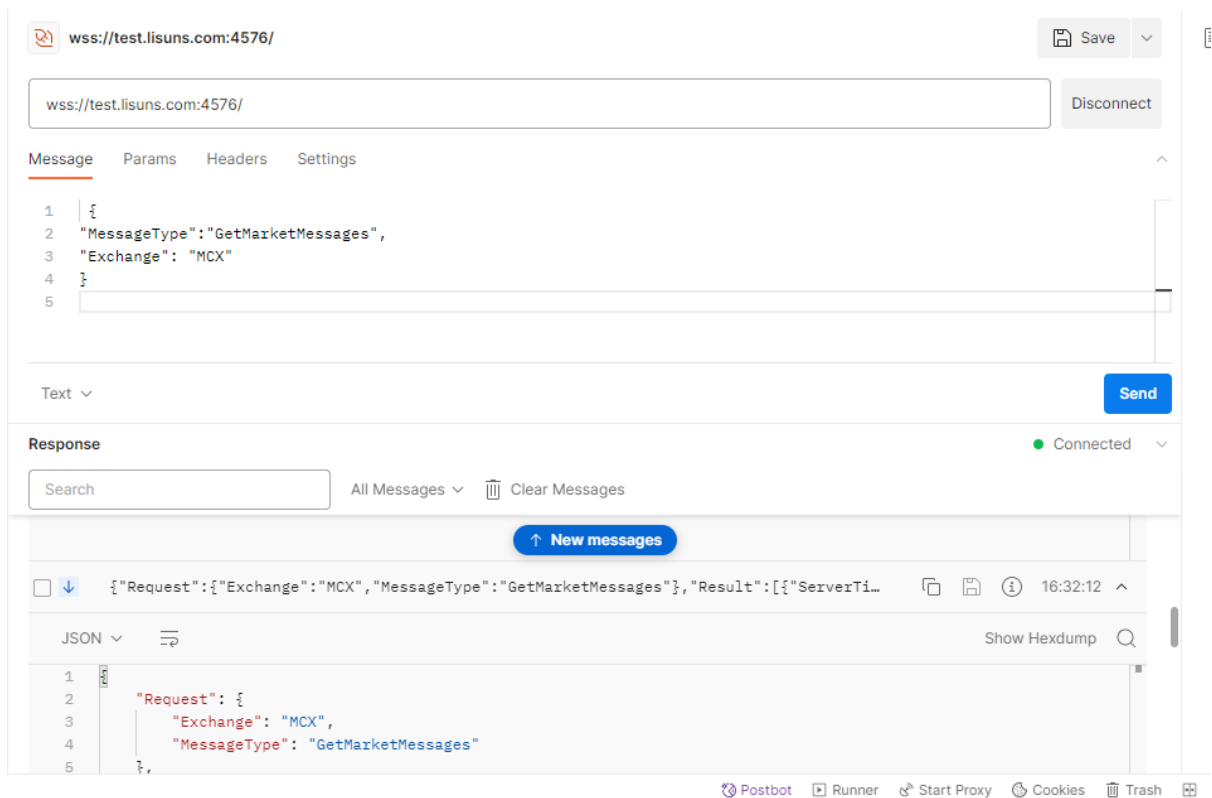
GetMarketMessages

Returns array of last messages (Market Messages) related to selected exchange

```
{  
  "MessageType": "GetMarketMessages",  
  "Exchange": "MCX"  
}
```

For detailed description please refer below link.

[GetMarketMessages | Global Datafeeds](#)



The screenshot displays a web application interface for testing WebSocket messages. At the top, there is a URL bar containing 'wss://test.lisuns.com:4576/' and a 'Send' button. Below the URL bar, there is a 'Message' tab with a text area containing the following JSON message:

```
{  
  "MessageType": "GetMarketMessages",  
  "Exchange": "MCX"  
}
```

Below the message tab, there is a 'Response' section with a search bar and a 'Clear Messages' button. The response section shows a list of messages, with the first message expanded to show its JSON content:

```
{  
  "Request": {  
    "Exchange": "MCX",  
    "MessageType": "GetMarketMessages"  
  },  
  "Result": [{  
    "ServerTime": 163212  
  }]  
}
```

The interface also includes a 'New messages' button and a 'Show Hexdump' option. At the bottom, there is a status bar with icons for Postbot, Runner, Start Proxy, Cookies, Trash, and a grid icon.

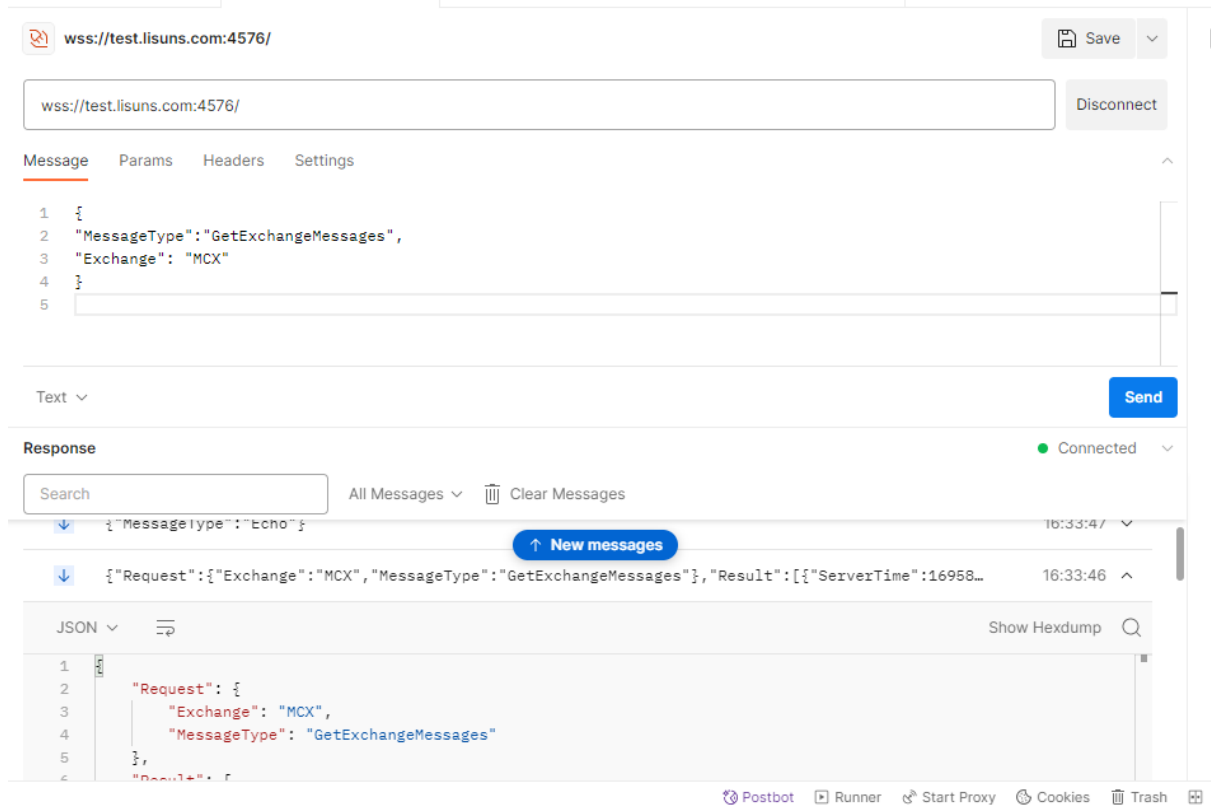
GetExchangeMessages

Returns array of last messages (Exchange Messages) related to selected exchange.

```
{  
  "MessageType": "GetExchangeMessages",  
  "Exchange": "MCX"  
}
```

For detailed description please refer below link.

[GetExchangeMessages | Global Datafeeds](#)



The screenshot displays a web application interface for testing WebSocket messages. The URL bar shows `wss://test.lisuns.com:4576/`. The 'Message' tab is active, showing a JSON request:

```
{  
  "MessageType": "GetExchangeMessages",  
  "Exchange": "MCX"  
}
```

. The 'Response' tab shows a list of messages, including a 'New messages' button. The JSON response is expanded, showing:

```
{  
  "Request": {  
    "Exchange": "MCX",  
    "MessageType": "GetExchangeMessages"  
  },  
  "Result": [  
    {  
      "ServerTime": 16958...    }  
  ]  
}
```

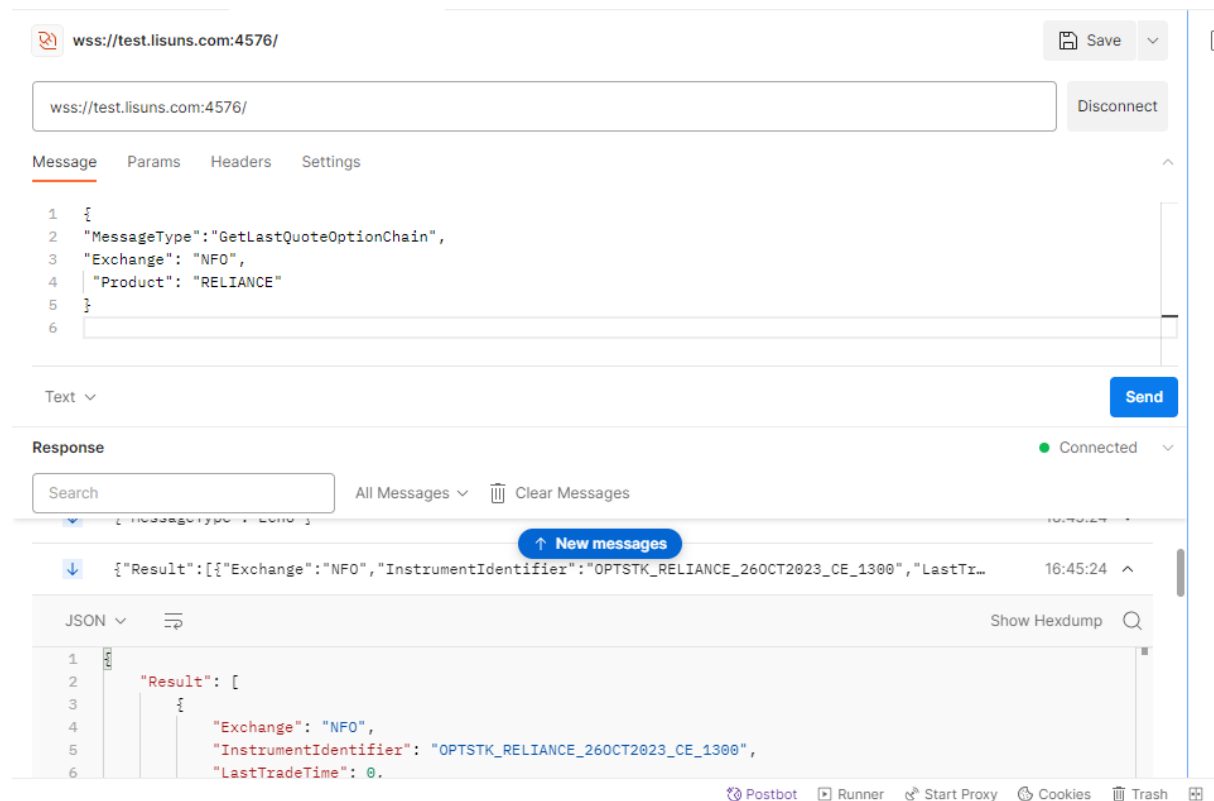
GetLastQuoteOptionChain

Returns LastTradePrice of entire OptionChain of requested underlying.

```
{  
  "MessageType": "GetLastQuoteOptionChain",  
  "Exchange": "NFO",  
  "Product": "RELIANCE"  
}
```

For detailed description please refer below link.

[GetLastQuoteOptionChain | Global Datafeeds](#)



The screenshot displays a web-based WebSocket client interface. At the top, the URL bar shows `wss://test.lisuns.com:4576/`. Below the URL bar, there are tabs for **Message**, **Params**, **Headers**, and **Settings**. The **Message** tab is active, showing a JSON message being sent:

```
1 {  
2   "MessageType": "GetLastQuoteOptionChain",  
3   "Exchange": "NFO",  
4   "Product": "RELIANCE"  
5 }  
6
```

Below the message input, there is a **Text** dropdown and a **Send** button. The **Response** section shows a green status indicator and a **Connected** label. Below this, there is a search bar and a list of messages. A message is selected, showing its timestamp as 16:45:24. The response is displayed in JSON format:

```
1 {  
2   "Result": [  
3     {  
4       "Exchange": "NFO",  
5       "InstrumentIdentifier": "OPTSTK_RELIANCE_26OCT2023_CE_1300",  
6       "LastTradeTime": 0,  
7     }  
8   ]  
9 }
```

At the bottom of the interface, there are several utility buttons: **Postbot**, **Runner**, **Start Proxy**, **Cookies**, **Trash**, and a window management icon.

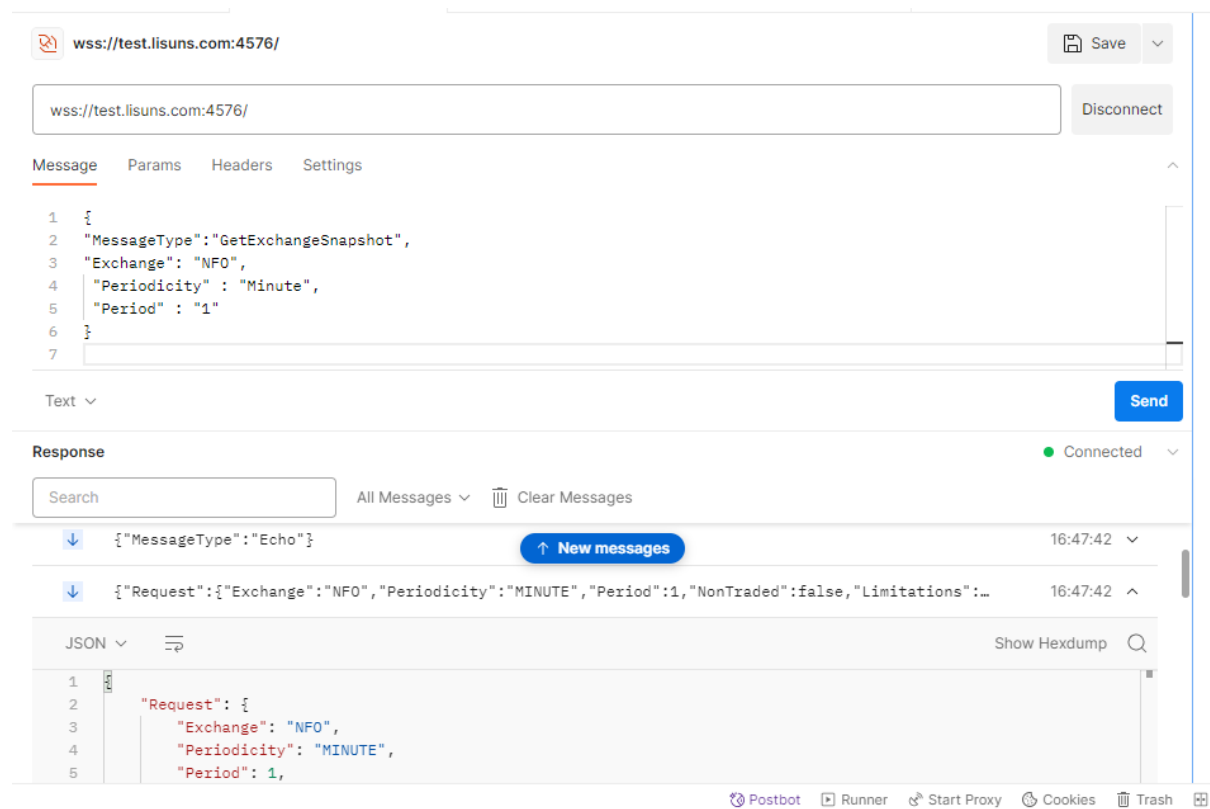
GetExchangeSnapshot

Returns entire Exchange Snapshot as per Period & Periodicity

```
{  
  "MessageType": "GetExchangeSnapshot",  
  "Exchange": "NFO",  
  "Periodicity": "Minute",  
  "Period": "1"  
}
```

For detailed description please refer below link.

[GetExchangeSnapshot | Global Datafeeds](#)



The screenshot displays a web application interface for testing WebSocket messages. The top section shows the URL `wss://test.lisuns.com:4576/` and a `Save` button. Below this is a text input field containing the same URL and a `Disconnect` button. The main area is divided into tabs: `Message`, `Params`, `Headers`, and `Settings`. The `Message` tab is active, showing a JSON message being sent:

```
1 {  
2   "MessageType": "GetExchangeSnapshot",  
3   "Exchange": "NFO",  
4   "Periodicity": "Minute",  
5   "Period": "1"  
6 }  
7
```

Below the message input is a `Text` dropdown and a `Send` button. The `Response` section shows the received message, which is a JSON object:

```
{  
  "Request": {  
    "Exchange": "NFO",  
    "Periodicity": "MINUTE",  
    "Period": "1",  
    "NonTraded": false,  
    "Limitations": ...  
  }  
}
```

The interface also includes a `Search` bar, `All Messages` dropdown, `Clear Messages` button, and a `New messages` button. The bottom status bar shows `Connected` and a `Show Hexdump` button.

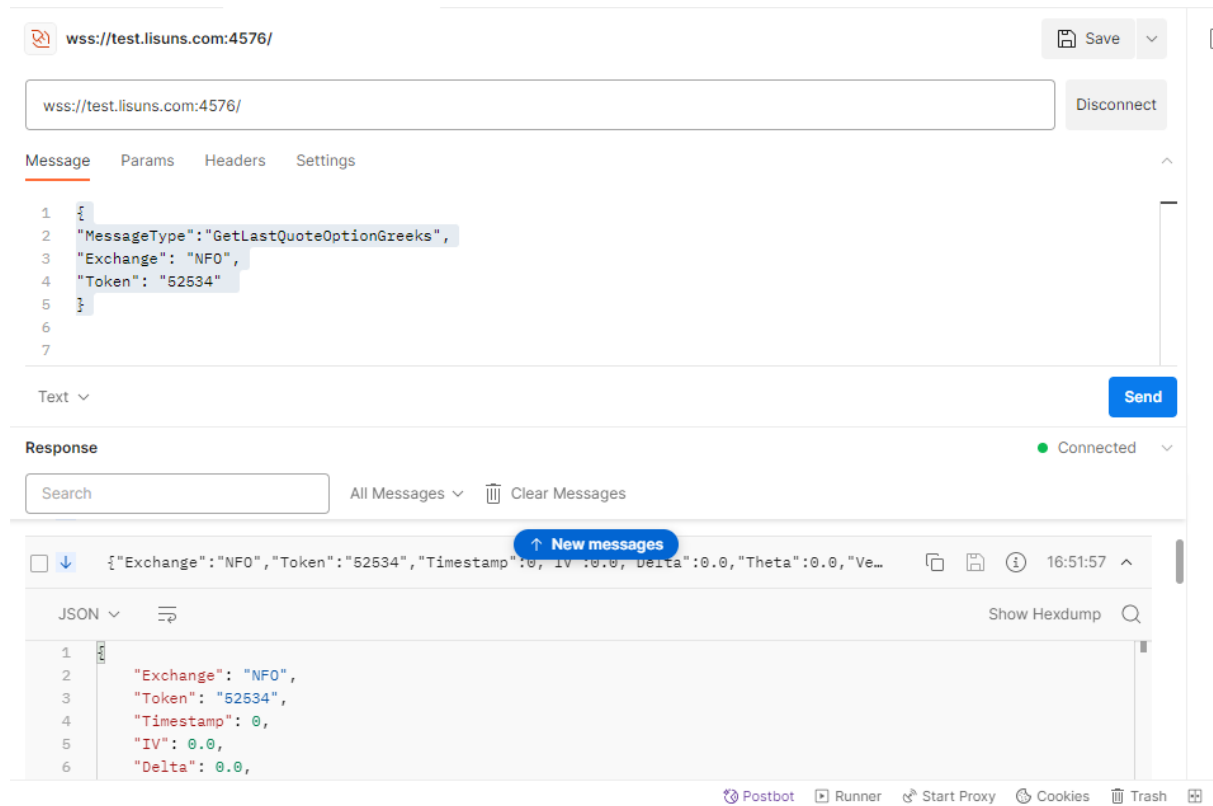
GetLastQuoteOptionGreeks

Returns Last Traded Option Greek values of Single Symbol (detailed)

```
{  
  "MessageType": "GetLastQuoteOptionGreeks",  
  "Exchange": "NFO",  
  "Token": "52534"  
}
```

For detailed description please refer below link.

[GetLastQuoteOptionGreek | Global Datafeeds](#)



The screenshot displays a web-based WebSocket client interface. At the top, the URL `wss://test.lisuns.com:4576/` is entered in the address bar. Below the address bar, there are tabs for `Message`, `Params`, `Headers`, and `Settings`. The `Message` tab is active, showing a JSON message being sent: `{ "MessageType": "GetLastQuoteOptionGreeks", "Exchange": "NFO", "Token": "52534" }`. Below the message input, there is a `Send` button. The `Response` section shows a green status indicator and the text `Connected`. Below this, there is a search bar and a list of messages. The first message is expanded, showing a JSON response: `{ "Exchange": "NFO", "Token": "52534", "Timestamp": 0, "IV": 0.0, "Delta": 0.0, "Theta": 0.0, "Vega": 0.0 }`. The interface also includes a `New messages` button and a `Show Hexdump` option.

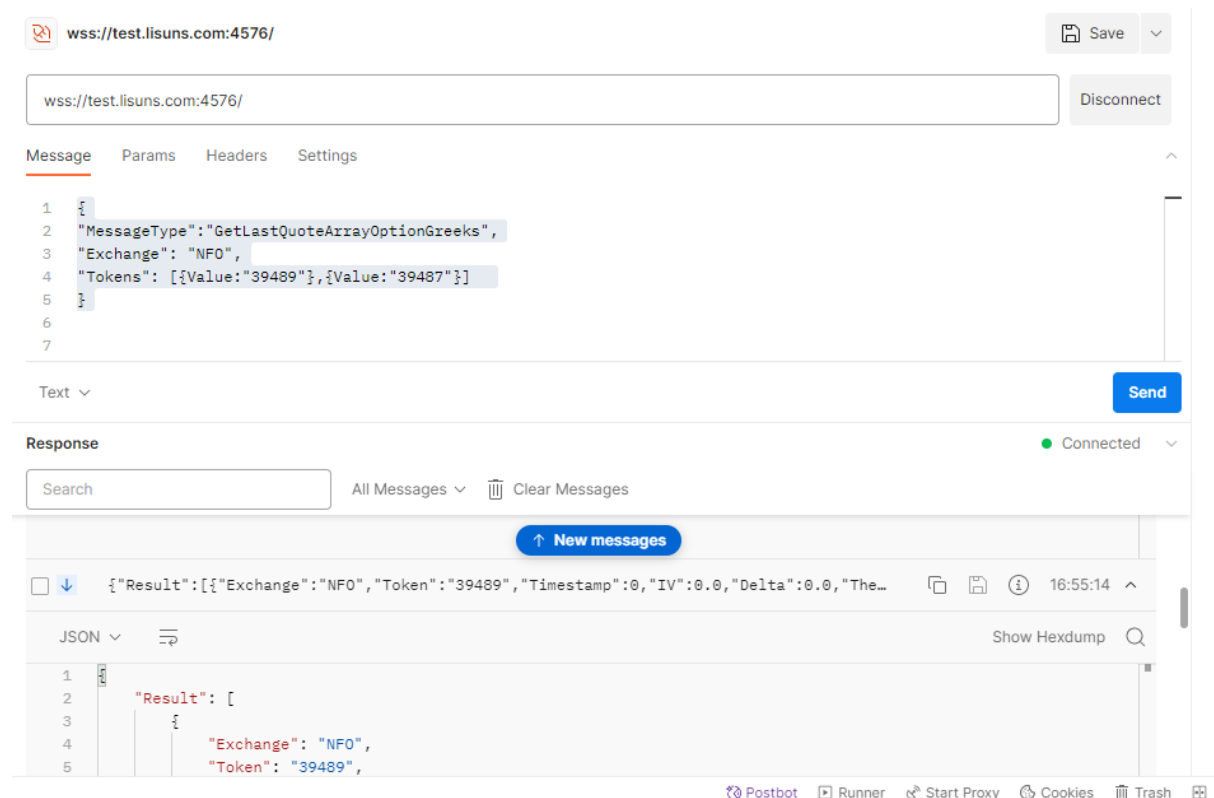
GetLastQuoteArrayOptionGreeks

Returns Last Traded Option Greek values of multiple Symbols – max 25 in single call (detailed)

```
{  
  "MessageType": "GetLastQuoteArrayOptionGreeks",  
  "Exchange": "NFO",  
  "Tokens": [{Value: "39489"}, {Value: "39487"}]  
}
```

For detailed description please refer below link.

[GetLastQuoteArrayOptionGreeks | Global Datafeeds](#)



The screenshot displays a web-based WebSocket client interface. At the top, the URL `wss://test.lisuns.com:4576/` is entered in the address bar. Below the address bar, there are tabs for "Message", "Params", "Headers", and "Settings". The "Message" tab is active, showing a JSON message being sent:

```
1 {  
2   "MessageType": "GetLastQuoteArrayOptionGreeks",  
3   "Exchange": "NFO",  
4   "Tokens": [{Value: "39489"}, {Value: "39487"}]  
5 }  
6  
7
```

Below the message, there is a "Text" input field and a "Send" button. The "Response" section shows a "Connected" status. Below the response, there is a search bar and a "Clear Messages" button. A "New messages" button is also present. The response message is displayed in a JSON view:

```
{  
  "Result": [  
    {  
      "Exchange": "NFO",  
      "Token": "39489",  
      "Timestamp": 0,  
      "IV": 0.0,  
      "Delta": 0.0,  
      "The..."  
    }  
  ]  
}
```

The interface also includes a "Show Hexdump" button and a "Postbot" button at the bottom.

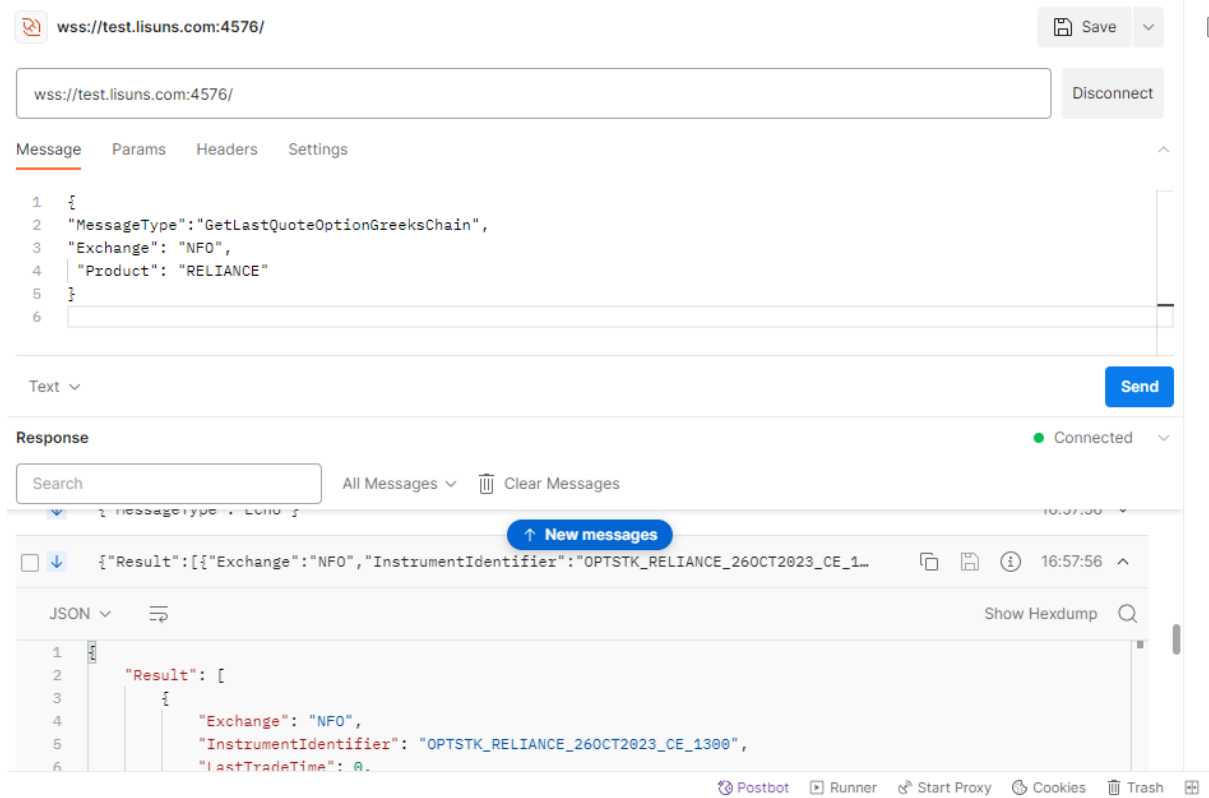
GetLastQuoteOptionGreeksChain

Returns Last Traded Option Greek values of entire OptionChain of requested underlying

```
{  
  "MessageType": "GetLastQuoteOptionGreeksChain",  
  "Exchange": "NFO",  
  "Product": "RELIANCE"  
}
```

For detailed description please refer below link.

[GetLastQuoteOptionGreeksChain | Global Datafeeds](#)



The screenshot shows a web browser interface for a WebSocket connection. The address bar displays `wss://test.lisuns.com:4576/`. The interface includes a 'Save' button, a 'Disconnect' button, and tabs for 'Message', 'Params', 'Headers', and 'Settings'. The 'Message' tab is active, showing the following JSON message:

```
1 {  
2   "MessageType": "GetLastQuoteOptionGreeksChain",  
3   "Exchange": "NFO",  
4   "Product": "RELIANCE"  
5 }  
6
```

Below the message, there is a 'Text' dropdown and a 'Send' button. The 'Response' section shows a 'Connected' status. A search bar and 'All Messages' / 'Clear Messages' options are present. A 'New messages' button is also visible. The response is displayed in JSON format:

```
1 {  
2   "Result": [  
3     {  
4       "Exchange": "NFO",  
5       "InstrumentIdentifier": "OPTSTK_RELIANCE_26OCT2023_CE_1300",  
6       "lastTradeTime": 0
```