# ShoonyaApi

Api used to connect to Shoonya OMS

## Build

to build this package and install it on your server please use

pip install -r requirements.txt

## API

ShoonyaApi

* [login](#md-login)

Symbols

* [searchscrip](#md-searchscrip)
* [get\_security\_info](#md-get_security_info)
* [get\_quotes](#md-get_quotes)
* [get\_time\_price\_series](#md-get_time_price_series)

Orders and Trades

* [place\_order](#md-place_order)
* [modify\_order](#md-modify_order)
* [cancel\_order](#md-cancel_order)
* [exit\_order](#md-exit_order)
* [get\_orderbook](#md-get_orderbook)
* [get\_singleorderhistory](#md-get_singleorderhistory)

Holdings and Limits

* [get\_holdings](#md-get_holdings)
* [get\_positions](#md-get_positions)
* [get\_limits](#md-get_limits)

Websocket API

* [start\_websocket](#md-start_websocket)
* [subscribe](#md-subscribe)
* [unsubscribe](#md-unsubscribe)

Example

* [getting started](#md-example-basic)
* [Market Functions](#md-example-market)
* [Orders and Trade](#md-example-orders)

#### <a name=“md-login”></a> login(userid, password, twoFA, vendor\_code, api\_secret, imei)

connect to the broker, only once this function has returned successfully can any other operations be performed

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| userid | string | False | user credentials |
| password | string | False | password encrypted |
| twoFA | string | False | dob/pan |
| vendor\_code | string | False | vendor code shared |
| api\_secret | string | False | your secret |
| imei | string | False | imei identification |

#### <a name=“md-place\_order”></a> place\_order(buy\_or\_sell, product\_type,exchange, tradingsymbol, quantity, discloseqty, price\_type, price=0.0, trigger\_price=None, retention=‘DAY’, amo=‘NO’, remarks=None)

place an order to oms

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| buy\_or\_sell | string | False | B -> BUY, S -> SELL |
| product\_type | string | False | C / M / H Product name (Select from ‘prarr’ Array provided in User Details response, and if same is allowed for selected, exchange. Show product display name, for user to select, and send corresponding prd in API call) |
| exchange | string | False | Exchange NSE / NFO / BSE / CDS |
| tradingsymbol | string | False | Unique id of contract on which order to be placed. (use url encoding to avoid special char error for symbols like M&M |
| quantity | integer | False | order quantity |
| discloseqty | integer | False | order disc qty |
| price\_type | string | False | PriceType enum class |
| price | integer | False | Price in paise, 100.00 is sent as 10000 |
| trigger\_price | integer | False | Price in paise |
| retention | string | False | DAY / IOC / EOS |
| amo | string | True | Flag for After Market Order, YES/NO |
| remarks | string | True | client order id or free text |

#### <a name=“md-modify\_order”></a> modify\_order(orderno, exchange, tradingsymbol, newquantity,newprice\_type, newprice, newtrigger\_price, amo):

modify the quantity pricetype or price of an order

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| orderno | string | False | orderno to be modified |
| exchange | string | False | Exchange NSE / NFO / BSE / CDS |
| tradingsymbol | string | False | Unique id of contract on which order to be placed. (use url encoding to avoid special char error for symbols like M&M |
| newquantity | integer | False | new order quantity |
| newprice\_type | string | False | PriceType enum class |
| newprice | integer | False | Price in paise, 100.00 is sent as 10000 |
| newtrigger\_price | integer | False | Price in paise |

#### <a name=“md-cancel\_order”></a> cancel\_order(orderno)

cancel an order

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| orderno | string | False | orderno with status open |

#### <a name=“md-exit\_order”></a> exit\_order(orderno)

exits a cover or bracket order

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| orderno | string | False | orderno with status open |
| prd | string | False | Allowed for only H and B products (Cover order and bracket order) |

#### <a name=“md-get\_singleorderhistory”></a> single order history(orderno)

history an order

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| orderno | string | False | orderno |

#### <a name=“md-get\_holdings”></a> get\_holdings(product\_type)

retrieves the holdings as a list

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| product\_type | string | True | retreives the delivery holdings or for a given product |

#### <a name=“md-get\_positions”></a> get\_positions()

retrieves the positions cf and day as a list

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| No Parameters |  |  |  |

#### <a name=“md-get\_limits”></a> get\_limits

retrieves the margin and limits set

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| product\_type | string | True | retreives the delivery holdings or for a given product |
| segment | string | True | CM / FO / FX |
| exchange | string | True | Exchange NSE/BSE/MCX |

the response is as follows,

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| stat | Ok or Not\_Ok | False | Limits request success or failure indication. |
| actid | string | True | Account id |
| prd | string | True | Product name |
| seg | string | True | Segment CM / FO / FX |
| exch | string | True | Exchange |
| -------------------------Cash Primary Fields------------------------------- |  |  |  |
| cash | string | True | Cash Margin available |
| payin | string | True | Total Amount transferred using Payins today |
| payout | string | True | Total amount requested for withdrawal today |
| -------------------------Cash Additional Fields------------------------------- |  |  |  |
| brkcollamt | string | True | Prevalued Collateral Amount |
| unclearedcash | string | True | Uncleared Cash (Payin through cheques) |
| daycash | string | True | Additional leverage amount / Amount added to handle system errors - by broker. |
| -------------------------Margin Utilized---------------------------------- |  |  |  |
| marginused | string | True | Total margin / fund used today |
| mtomcurper | string | True | Mtom current percentage |
| -------------------------Margin Used components--------------------- |  |  |  |
| cbu | string | True | CAC Buy used |
| csc | string | True | CAC Sell Credits |
| rpnl | string | True | Current realized PNL |
| unmtom | string | True | Current unrealized mtom |
| marprt | string | True | Covered Product margins |
| span | string | True | Span used |
| expo | string | True | Exposure margin |
| premium | string | True | Premium used |
| varelm | string | True | Var Elm Margin |
| grexpo | string | True | Gross Exposure |
| greexpo\_d | string | True | Gross Exposure derivative |
| scripbskmar | string | True | Scrip basket margin |
| addscripbskmrg | string | True | Additional scrip basket margin |
| brokerage | string | True | Brokerage amount |
| collateral | string | True | Collateral calculated based on uploaded holdings |
| grcoll | string | True | Valuation of uploaded holding pre haircut |
| -------------------------Additional Risk Limits--------------------------- |  |  |  |
| turnoverlmt | string | True |  |
| pendordvallmt | string | True |  |
| -------------------------Additional Risk Indicators--------------------------- |  |  |  |
| turnover | string | True | Turnover |
| pendordval | string | True | Pending Order value |
| -------------------------Margin used detailed breakup fields------------------------- |  |  |  |
| rzpnl\_e\_i | string | True | Current realized PNL (Equity Intraday) |
| rzpnl\_e\_m | string | True | Current realized PNL (Equity Margin) |
| rzpnl\_e\_c | string | True | Current realized PNL (Equity Cash n Carry) |
| rzpnl\_d\_i | string | True | Current realized PNL (Derivative Intraday) |
| rzpnl\_d\_m | string | True | Current realized PNL (Derivative Margin) |
| rzpnl\_f\_i | string | True | Current realized PNL (FX Intraday) |
| rzpnl\_f\_m | string | True | Current realized PNL (FX Margin) |
| rzpnl\_c\_i | string | True | Current realized PNL (Commodity Intraday) |
| rzpnl\_c\_m | string | True | Current realized PNL (Commodity Margin) |
| uzpnl\_e\_i | string | True | Current unrealized MTOM (Equity Intraday) |
| uzpnl\_e\_m | string | True | Current unrealized MTOM (Equity Margin) |
| uzpnl\_e\_c | string | True | Current unrealized MTOM (Equity Cash n Carry) |
| uzpnl\_d\_i | string | True | Current unrealized MTOM (Derivative Intraday) |
| uzpnl\_d\_m | string | True | Current unrealized MTOM (Derivative Margin) |
| uzpnl\_f\_i | string | True | Current unrealized MTOM (FX Intraday) |
| uzpnl\_f\_m | string | True | Current unrealized MTOM (FX Margin) |
| uzpnl\_c\_i | string | True | Current unrealized MTOM (Commodity Intraday) |
| uzpnl\_c\_m | string | True | Current unrealized MTOM (Commodity Margin) |
| span\_d\_i | string | True | Span Margin (Derivative Intraday) |
| span\_d\_m | string | True | Span Margin (Derivative Margin) |
| span\_f\_i | string | True | Span Margin (FX Intraday) |
| span\_f\_m | string | True | Span Margin (FX Margin) |
| span\_c\_i | string | True | Span Margin (Commodity Intraday) |
| span\_c\_m | string | True | Span Margin (Commodity Margin) |
| expo\_d\_i | string | True | Exposure Margin (Derivative Intraday) |
| expo\_d\_m | string | True | Exposure Margin (Derivative Margin) |
| expo\_f\_i | string | True | Exposure Margin (FX Intraday) |
| expo\_f\_m | string | True | Exposure Margin (FX Margin) |
| expo\_c\_i | string | True | Exposure Margin (Commodity Intraday) |
| expo\_c\_m | string | True | Exposure Margin (Commodity Margin) |
| premium\_d\_i | string | True | Option premium (Derivative Intraday) |
| premium\_d\_m | string | True | Option premium (Derivative Margin) |
| premium\_f\_i | string | True | Option premium (FX Intraday) |
| premium\_f\_m | string | True | Option premium (FX Margin) |
| premium\_c\_i | string | True | Option premium (Commodity Intraday) |
| premium\_c\_m | string | True | Option premium (Commodity Margin) |
| varelm\_e\_i | string | True | Var Elm (Equity Intraday) |
| varelm\_e\_m | string | True | Var Elm (Equity Margin) |
| varelm\_e\_c | string | True | Var Elm (Equity Cash n Carry) |
| marprt\_e\_h | string | True | Covered Product margins (Equity High leverage) |
| marprt\_e\_b | string | True | Covered Product margins (Equity Bracket Order) |
| marprt\_d\_h | string | True | Covered Product margins (Derivative High leverage) |
| marprt\_d\_b | string | True | Covered Product margins (Derivative Bracket Order) |
| marprt\_f\_h | string | True | Covered Product margins (FX High leverage) |
| marprt\_f\_b | string | True | Covered Product margins (FX Bracket Order) |
| marprt\_c\_h | string | True | Covered Product margins (Commodity High leverage) |
| marprt\_c\_b | string | True | Covered Product margins (Commodity Bracket Order) |
| scripbskmar\_e\_i | string | True | Scrip basket margin (Equity Intraday) |
| scripbskmar\_e\_m | string | True | Scrip basket margin (Equity Margin) |
| scripbskmar\_e\_c | string | True | Scrip basket margin (Equity Cash n Carry) |
| addscripbskmrg\_d\_i | string | True | Additional scrip basket margin (Derivative Intraday) |
| addscripbskmrg\_d\_m | string | True | Additional scrip basket margin (Derivative Margin) |
| addscripbskmrg\_f\_i | string | True | Additional scrip basket margin (FX Intraday) |
| addscripbskmrg\_f\_m | string | True | Additional scrip basket margin (FX Margin) |
| addscripbskmrg\_c\_i | string | True | Additional scrip basket margin (Commodity Intraday) |
| addscripbskmrg\_c\_m | string | True | Additional scrip basket margin (Commodity Margin) |
| brkage\_e\_i | string | True | Brokerage (Equity Intraday) |
| brkage\_e\_m | string | True | Brokerage (Equity Margin) |
| brkage\_e\_c | string | True | Brokerage (Equity CAC) |
| brkage\_e\_h | string | True | Brokerage (Equity High Leverage) |
| brkage\_e\_b | string | True | Brokerage (Equity Bracket Order) |
| brkage\_d\_i | string | True | Brokerage (Derivative Intraday) |
| brkage\_d\_m | string | True | Brokerage (Derivative Margin) |
| brkage\_d\_h | string | True | Brokerage (Derivative High Leverage) |
| brkage\_d\_b | string | True | Brokerage (Derivative Bracket Order) |
| brkage\_f\_i | string | True | Brokerage (FX Intraday) |
| brkage\_f\_m | string | True | Brokerage (FX Margin) |
| brkage\_f\_h | string | True | Brokerage (FX High Leverage) |
| brkage\_f\_b | string | True | Brokerage (FX Bracket Order) |
| brkage\_c\_i | string | True | Brokerage (Commodity Intraday) |
| brkage\_c\_m | string | True | Brokerage (Commodity Margin) |
| brkage\_c\_h | string | True | Brokerage (Commodity High Leverage) |
| brkage\_c\_b | string | True | Brokerage (Commodity Bracket Order) |
| peak\_mar | string | True | Peak margin used by the client |
| request\_time | string | True | This will be present only in a successful response. |
| emsg | string | True | This will be present only in a failure response. |

#### <a name=“md-searchscrip”></a> searchscrip(exchange, searchtext):

search for scrip or contract and its properties

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| exchange | string | True | Exchange NSE / NFO / BSE / CDS |
| searchtext | string | True | Search Text ie partial or complete text ex: INFY-EQ, INF… |

the response is as follows,

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| stat | string | True | ok or Not\_ok |
| values | string | True | properties of the scrip |
| emsg | string | False | Error Message |

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| exch | string | True | Exchange NSE / NFO / BSE / CDS |
| tsym | string | True | Trading Symbol is the readable Unique id of contract/scrip |
| token | string | True | Unique Code of contract/scrip |
| pp | string | True | price precision, in case of cds its 4 ie 100.1234 |
| ti | string | True | tick size minimum increments of paise for price |
| ls | string | True | Lot Size |

#### <a name=“md-get\_security\_info”></a> get\_security\_info(exchange, token):

gets the complete details and its properties

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| exchange | string | True | Exchange NSE / NFO / BSE / CDS |
| token | string | True | token number of the contract |

the response is as follows,

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| stat | string | True | ok or Not\_ok |
| values | string | True | properties of the scrip |
| emsg | string | False | Error Message |

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| exch | string | True | Exchange NSE / NFO / BSE / CDS |
| tsym | string | True | Trading Symbol is the readable Unique id of contract/scrip |
| cname | string | True | Company Name |
| symnam | string | True | Symbol Name |
| seg | string | True | Segment |
| exd | string | True | Expiry Date |
| instname | string | True | Instrument Name |
| strprc | string | True | Strike Price |
| optt | string | True | Option Type |
| isin | string | True | ISIN |
| ti | string | True | Tick Size |
| ls | string | True | Lot Size |
| pp | string | True | Price Precision |
| mult | string | True | Multiplier |
| gp\_nd | string | True | GN/GD \* PN/PD |
| prcunt | string | True | Price Units |
| prcqqty | string | True | Price Quote Qty |
| trdunt | string | True | Trade Units |
| delunt | string | True | Delivery Units |
| frzqty | string | True | Freeze Qty |
| gsmind | string | True | GSM indicator |
| elmbmrg | string | True | ELM Buy Margin |
| elmsmrg | string | True | ELM Sell Margin |
| addbmrg | string | True | Additional Long Margin |
| addsmrg | string | True | Additional Short Margin |
| splbmrg | string | True | Special Long Margin |
| splsmrg | string | True | Special Short Margin |
| delmrg | string | True | Delivery Margin |
| tenmrg | string | True | Tender Margin |
| tenstrd | string | True | Tender Start Date |
| tenendd | string | True | Tender End Date |
| exestrd | string | True | Exercise Start Date |
| exeendd | string | True | Exercise End Date |
| elmmrg | string | True | ELM Margin |
| varmrg | string | True | VAR Margin |
| expmrg | string | True | Exposure Margin |
| token | string | True | Contract Token |
| prcftr\_d | string | True | ((GN / GD) \* (PN/PD)) |

#### <a name=“md-get\_quotes”></a> get\_quotes(exchange, token):

gets the complete details and its properties

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| exchange | string | True | Exchange NSE / NFO / BSE / CDS |
| token | string | True | token number of the contract |

the response is as follows,

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| stat | string | True | ok or Not\_ok |
| values | string | True | properties of the scrip |
| emsg | string | False | Error Message |

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| exch | string | True | Exchange NSE / NFO / BSE / CDS |
| tsym | string | True | Trading Symbol is the readable Unique id of contract/scrip |
| cname | string | True | Company Name |
| symname | string | True | Symbol Name |
| seg | string | True | Segment |
| instname | string | True | Instrument Name |
| isin | string | True | ISIN |
| pp | string | True | Price precision |
| ls | string | True | Lot Size |
| ti | string | True | Tick Size |
| mult | string | True | Multiplier |
| uc | string | True | Upper circuit limitlc |
| lc | string | True | Lower circuit limit |
| prcftr\_d | string | True | Price factor((GN / GD) \* (PN/PD)) |
| token | string | True | Token |
| lp | string | True | LTP |
| o | string | True | Open Price |
| h | string | True | Day High Price |
| l | string | True | Day Low Price |
| v | string | True | Volume |
| ltq | string | True | Last trade quantity |
| ltt | string | True | Last trade time |
| bp1 | string | True | Best Buy Price 1 |
| sp1 | string | True | Best Sell Price 1 |
| bp2 | string | True | Best Buy Price 2 |
| sp2 | string | True | Best Sell Price 2 |
| bp3 | string | True | Best Buy Price 3 |
| sp3 | string | True | Best Sell Price 3 |
| bp4 | string | True | Best Buy Price 4 |
| sp4 | string | True | Best Sell Price 4 |
| bp5 | string | True | Best Buy Price 5 |
| sp5 | string | True | Best Sell Price 5 |
| bq1 | string | True | Best Buy Quantity 1 |
| sq1 | string | True | Best Sell Quantity 1 |
| bq2 | string | True | Best Buy Quantity 2 |
| sq2 | string | True | Best Sell Quantity 2 |
| bq3 | string | True | Best Buy Quantity 3 |
| sq3 | string | True | Best Sell Quantity 3 |
| bq4 | string | True | Best Buy Quantity 4 |
| sq4 | string | True | Best Sell Quantity 4 |
| bq5 | string | True | Best Buy Quantity 5 |
| sq5 | string | True | Best Sell Quantity 5 |
| bo1 | string | True | Best Buy Orders 1 |
| so1 | string | True | Best Sell Orders 1 |
| bo2 | string | True | Best Buy Orders 2 |
| so2 | string | True | Best Sell Orders 2 |
| bo3 | string | True | Best Buy Orders 3 |
| so3 | string | True | Best Sell Orders 3 |
| bo4 | string | True | Best Buy Orders 4 |
| so4 | string | True | Best Sell Orders 4 |
| bo5 | string | True | Best Buy Orders 5 |
| so5 | string | True | Best Sell Orders 5 |

#### <a name=“md-get\_time\_price\_series”></a> get\_time\_price\_series(exchange, token, starttime, endtime):

gets the chart date for the symbol

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| exchange | string | True | Exchange NSE / NFO / BSE / CDS |
| token | string | True | token number of the contract |
| starttime | string | True | Start time (seconds since 1 jan 1970) |
| endtime | string | True | End Time (seconds since 1 jan 1970) |

the response is as follows,

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| stat | string | True | ok or Not\_ok |
| values | string | True | properties of the scrip |
| emsg | string | False | Error Message |

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| time | string | True | DD/MM/CCYY hh:mm:ss |
| into | string | True | Interval Open |
| inth | string | True | Interval High |
| intl | string | True | Interval Low |
| intc | string | True | Interval Close |
| intvwap | string | True | Interval vwap |
| intv | string | True | Interval volume |
| v | string | True | volume |
| inoi | string | True | Interval oi change |
| oi | string | True | oi |

#### <a name=“md-start\_websocket”></a> start\_websocket()

starts the websocket

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| subscribe\_callback | function | False | callback for market updates |
| order\_update\_callback | function | False | callback for order updates |
| socket\_open\_callback | function | False | callback when socket is open (reconnection also) |
| socket\_close\_callback | function | False | callback when socket is closed |

#### <a name=“md-subscribe\_orders”></a> subscribe\_orders()

get order and trade update callbacks

#### <a name=“md-subscribe”></a> subscribe([instruments])

send a list of instruments to watch

| **Param** | **Type** | **Optional** | **Description** |
| --- | --- | --- | --- |
| instruments | list | False | list of instruments [NSE|22,CDS|1] |

#### <a name=“md-unsubscribe”></a> unsubscribe()

send a list of instruments to stop watch

## <a name=“md-example-basic”></a> Example - Getting Started

First configure the endpoints in the api\_helper constructor.  
Thereon provide your credentials and login as follows.

from api\_helper import ShoonyaApiPy

import logging

#enable dbug to see request and responses

logging.basicConfig(level=logging.DEBUG)

#start of our program

api = ShoonyaApiPy()

#credentials

user = '< user id>'

u\_pwd = '< password >'

factor2 = 'second factor'

vc = 'vendor code'

app\_key = 'secret key'

imei = 'uniq identifier'

ret = api.login(userid=user, password=pwd, twoFA=factor2, vendor\_code=vc, api\_secret=app\_key, imei=imei)

print(ret)

## <a name=“md-example-market”></a> Example Symbol/Contract : Example\_market.py

This Example shows API usage for finding scrips and its properties

### Search Scrips

The call can be made to get the exchange provided token for a scrip or alternately can search for a partial string to get a list of matching scrips  
Trading Symbol:

SymbolName + ExpDate + ‘F’ for all data having InstrumentName starting with FUT  
SymbolName + ExpDate + ‘P’ + StrikePrice for all data having InstrumentName starting with OPT and with OptionType PE  
SymbolName + ExpDate + ‘C’ + StrikePrice for all data having InstrumentName starting with OPT and with OptionType C  
For MCX, F to be ignored for FUT instruments

api.searchscrip(exchange='NSE', searchtext='REL')

This will reply as following

{

"stat": "Ok",

"values": [

{

"exch": "NSE",

"token": "18069",

"tsym": "REL100NAV-EQ"

},

{

"exch": "NSE",

"token": "24225",

"tsym": "RELAXO-EQ"

},

{

"exch": "NSE",

"token": "4327",

"tsym": "RELAXOFOOT-EQ"

},

{

"exch": "NSE",

"token": "18068",

"tsym": "RELBANKNAV-EQ"

},

{

"exch": "NSE",

"token": "2882",

"tsym": "RELCAPITAL-EQ"

},

{

"exch": "NSE",

"token": "18070",

"tsym": "RELCONSNAV-EQ"

},

{

"exch": "NSE",

"token": "18071",

"tsym": "RELDIVNAV-EQ"

},

{

"exch": "NSE",

"token": "18072",

"tsym": "RELGOLDNAV-EQ"

},

{

"exch": "NSE",

"token": "2885",

"tsym": "RELIANCE-EQ"

},

{

"exch": "NSE",

"token": "15068",

"tsym": "RELIGARE-EQ"

},

{

"exch": "NSE",

"token": "553",

"tsym": "RELINFRA-EQ"

},

{

"exch": "NSE",

"token": "18074",

"tsym": "RELNV20NAV-EQ"

}

]

}

### Security Info

This call is done to get the properties of the scrip such as freeze qty and margins

api.get\_security\_info(exchange='NSE', token='22')

The response for the same would be

{

"request\_time": "17:43:38 31-10-2020",

"stat": "Ok",

"exch": "NSE",

"tsym": "ACC-EQ",

"cname": "ACC LIMITED",

"symname": "ACC",

"seg": "EQT",

"instname": "EQ",

"isin": "INE012A01025",

"pp": "2",

"ls": "1",

"ti": "0.05",

"mult": "1",

"prcftr\_d": "(1 / 1 ) \* (1 / 1)",

"trdunt": "ACC.BO",

"delunt": "ACC",

"token": "22",

"varmrg": "40.00"

}

### Subscribe to a live feed

Subscribe to a single token as follows

api.subscribe('NSE|13')

Subscribe to a list of tokens as follows

api.subscribe(['NSE|22', 'BSE|522032'])

First we need to connect to the WebSocket and then subscribe as follows

feed\_opened = False

def event\_handler\_feed\_update(tick\_data):

print(f"feed update {tick\_data}")

def open\_callback():

global feed\_opened

feed\_opened = True

api.start\_websocket( order\_update\_callback=event\_handler\_order\_update,

subscribe\_callback=event\_handler\_feed\_update,

socket\_open\_callback=open\_callback)

while(feed\_opened==False):

pass

# subscribe to a single token

api.subscribe('NSE|13')

#subscribe to multiple tokens

api.subscribe(['NSE|22', 'BSE|522032'])

## <a name=“md-example-orders”></a> Example - Orders and Trades : example\_orders.py

### Place Order

Place a Limit order as follows

api.place\_order(buy\_or\_sell='B', product\_type='C',

exchange='NSE', tradingsymbol='INFY-EQ',

quantity=1, discloseqty=0,price\_type='LMT', price=1500, trigger\_price=None,

retention='DAY', remarks='my\_order\_001')

Place a Market Order as follows

api.place\_order(buy\_or\_sell='B', product\_type='C',

exchange='NSE', tradingsymbol='INFY-EQ',

quantity=1, discloseqty=0,price\_type='MKT', price=0, trigger\_price=None,

retention='DAY', remarks='my\_order\_001')

Place a StopLoss Order as follows

api.place\_order(buy\_or\_sell='B', product\_type='C',

exchange='NSE', tradingsymbol='INFY-EQ',

quantity=1, discloseqty=0,price\_type='SL-LMT', price=1500, trigger\_price=1450,

retention='DAY', remarks='my\_order\_001')

Place a Cover Order as follows

api.place\_order(buy\_or\_sell='B', product\_type='H',

exchange='NSE', tradingsymbol='INFY-EQ',

quantity=1, discloseqty=0,price\_type='LMT', price=1500, trigger\_price=None,

retention='DAY', remarks='my\_order\_001', bookloss\_price = 1490)

Place a Bracket Order as follows

api.place\_order(buy\_or\_sell='B', product\_type='H',

exchange='NSE', tradingsymbol='INFY-EQ',

quantity=1, discloseqty=0,price\_type='LMT', price=1500, trigger\_price=None,

retention='DAY', remarks='my\_order\_001', bookloss\_price = 1490, bookprofit\_price = 1510)

### Modify Order

Modify a New Order by providing the OrderNumber

api.modify\_order(exchange='NSE', tradingsymbol='INFY-EQ', orderno=orderno,

newquantity=2, newprice\_type='LMT', newprice=1505)

### Cancel Order

Cancel a New Order by providing the Order Number

api.cancel\_order(orderno=orderno)

### Subscribe to Order Updates

Connecting to the Websocket will automatically subscribe and provide the order updates in the call back as follows  
Note: Feed and Order updates are received from the same websocket and needs to be connected once only.

feed\_opened = False

def event\_handler\_order\_update(order):

print(f"order feed {order}")

def open\_callback():

global feed\_opened

feed\_opened = True

api.start\_websocket( order\_update\_callback=event\_handler\_order\_update,

subscribe\_callback=event\_handler\_feed\_update,

socket\_open\_callback=open\_callback)

while(feed\_opened==False):

pass

## Author

Kumar Anand

## License

Copyright © 2021 Kambala Solutions Pvt Ltd- All Rights Reserved  
Copying of this file, via any medium is strictly prohibited.  
Proprietary and confidential.  
All file transfers are logged.