# 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)
* [logout](#md-logout)

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)
* [get\_daily\_price\_series](#md-get_daily_price_series)
* [get\_option\_chain](#md-get_optionchain)

Orders and Trades

* [place\_order](#md-place_order)
* [modify\_order](#md-modify_order)
* [cancel\_order](#md-cancel_order)
* [exit\_order](#md-exit_order)
* [product\_convertion](#md-prd_convert)
* [get\_orderbook](#md-get_orderbook)
* [get\_tradebook](#md-get_tradebook)
* [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  
Example:

#credentials  
user = <uid>  
pwd = <password>  
factor2 = <2nd factor>  
vc = <vendor code>  
app\_key = <secret key>  
imei = <imei>  
  
ret = api.login(userid=uid, password=pwd, twoFA=factor2, vendor\_code=vc, api\_secret=app\_key, imei=imei)

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| apkversion\* |  | Application version. |
| uid\* |  | User Id of the login user |
| pwd\* |  | Sha256 of the user entered password. |
| factor2\* |  | DOB or PAN as entered by the user. (DOB should be in DD-MM-YYYY) |
| vc\* |  | Vendor code provided by noren team, along with connection URLs |
| appkey\* |  | Sha256 of uid |
| imei\* |  | Send mac if users logs in for desktop, imei is from mobile |
| addldivinf |  | Optional field, Value must be in below format: |
| ipaddr |  | Optional field |
| source | API |  |

Response Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Login Success Or failure status |
| susertoken |  | It will be present only on login success. This data to be sent in subsequent requests in jKey field and web socket connection while connecting. |
| lastaccesstime |  | It will be present only on login success. |
| spasswordreset | Y | If Y Mandatory password reset to be enforced. Otherwise the field will be absent. |
| exarr |  | Json array of strings with enabled exchange names |
| uname |  | User name |
| prarr |  | Json array of Product Obj with enabled products, as defined below. |
| actid |  | Account id |
| email |  | Email Id |
| brkname |  | Broker id |
| emsg |  | This will be present only if Login fails. |

Sample Success Response :  
{  
"request\_time": "20:18:47 19-05-2020",  
"stat": "Ok",  
"susertoken": "3b97f4c67762259a9ded6dbd7bfafe2787e662b3870422ddd343a59895f423a0",  
"lastaccesstime": "1589899727"  
}

Sample Failure Response :  
{  
"request\_time": "20:32:14 19-05-2020",  
"stat": "Not\_Ok",  
"emsg": "Invalid Input : Wrong Password"  
}

#### <a name="md-logout"></a> logout()

Terminate the session

Example:

ret = api.logout()

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | User Id of the login user |

Response Details :  
Response data will be in json format with below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Logout Success Or failure status |
| request\_time |  | It will be present only on successful logout. |
| emsg |  | This will be present only if Logout fails. |

Sample Success Response :  
{  
"stat":"Ok",  
"request\_time":"10:43:41 28-05-2020"  
}

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"emsg":"Server Timeout : "  
}

#### <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

Example:

ret = api.place\_order(buy\_or\_sell='B', product\_type='C',  
 exchange='NSE', tradingsymbol='CANBK-EQ',   
 quantity=1, discloseqty=0,price\_type='SL-LMT', price=200.00, trigger\_price=199.50,  
 retention='DAY', remarks='my\_order\_001')

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| actid\* |  | Login users account ID |
| exch\* | NSE / NFO / BSE / MCX | Exchange (Select from ‘exarr’ Array provided in User Details response) |
| tsym\* |  | Unique id of contract on which order to be placed. (use url encoding to avoid special char error for symbols like M&M) |
| qty\* |  | Order Quantity |
| prc\* |  | Order Price |
| trgprc |  | Only to be sent in case of SL / SL-M order. |
| dscqty |  | Disclosed quantity (Max 10% for NSE, and 50% for MCX) |
| prd\* | 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) |
| trantype\* | B / S | B -> BUY, S -> SELL |
| prctyp\* | LMT / MKT / SL-LMT / SL-MKT / DS / 2L / 3L |  |
| ret\* | DAY / EOS / IOC | Retention type (Show options as per allowed exchanges) |
| remarks |  | Any tag by user to mark order. |
| ordersource | MOB / WEB / TT | Used to generate exchange info fields. |
| bpprc |  | Book Profit Price applicable only if product is selected as B (Bracket order ) |
| blprc |  | Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order ) |
| trailprc |  | Trailing Price applicable only if product is selected as H and B (High Leverage and Bracket order ) |
| amo |  | Yes , If not sent, of Not “Yes”, will be treated as Regular order. |
| tsym2 |  | Trading symbol of second leg, mandatory for price type 2L and 3L (use url encoding to avoid special char error for symbols like M&M) |
| trantype2 |  | Transaction type of second leg, mandatory for price type 2L and 3L |
| qty2 |  | Quantity for second leg, mandatory for price type 2L and 3L |
| prc2 |  | Price for second leg, mandatory for price type 2L and 3L |
| tsym3 |  | Trading symbol of third leg, mandatory for price type 3L (use url encoding to avoid special char error for symbols like M&M) |
| trantype3 |  | Transaction type of third leg, mandatory for price type 3L |
| qty3 |  | Quantity for third leg, mandatory for price type 3L |
| prc3 |  | Price for third leg, mandatory for price type 3L |

Response Details :

Response data will be in json format with below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Place order success or failure indication. |
| request\_time |  | Response received time. |
| norenordno |  | It will be present only on successful Order placement to OMS. |
| emsg |  | This will be present only if Order placement fails |

Sample Success Response:  
{  
"request\_time": "10:48:03 20-05-2020",  
"stat": "Ok",  
"norenordno": "20052000000017"  
}

Sample Error Response :  
{  
"stat": "Not\_Ok",  
"request\_time": "20:40:01 19-05-2020",  
"emsg": "Error Occurred : 2 \"invalid input\""  
}

#### <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

Example:

orderno = ret['norenordno'] #from placeorder return value  
ret = api.modify\_order(exchange='NSE', tradingsymbol='CANBK-EQ', orderno=orderno,  
 newquantity=2, newprice\_type='MKT', newprice=0.00)  
## sl modification  
ret = api.modify\_order(exchange='NSE', tradingsymbol='CANBK-EQ', orderno=orderno,  
 newquantity=2, newprice\_type='SL-LMT', newprice=201.00, newtrigger\_price=200.00)

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| exch\* |  | Exchange |
| norenordno\* |  | Noren order number, which needs to be modified |
| prctyp | LMT / MKT / SL-MKT / SL-LMT | This can be modified. |
| prc |  | Modified / New price |
| qty |  | Modified / New Quantity |
| tsym\* |  | Unque id of contract on which order was placed. Can’t be modified, must be the same as that of original order. (use url encoding to avoid special char error for symbols like M&M) |
| ret | DAY / IOC / EOS | New Retention type of the order |
|  |  |  |
| trgprc |  | New trigger price in case of SL-MKT or SL-LMT |
| uid\* |  | User id of the logged in user. |
| bpprc |  | Book Profit Price applicable only if product is selected as B (Bracket order ) |
| blprc |  | Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order ) |
| trailprc |  | Trailing Price applicable only if product is selected as H and B (High Leverage and Bracket order ) |

Response Details :

Response data will be in json format with below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Modify order success or failure indication. |
| result |  | Noren Order number of the order modified. |
| request\_time |  | Response received time. |
| emsg |  | This will be present only if Order modification fails |

Sample Success Response :  
{  
"request\_time":"14:14:08 26-05-2020",  
"stat":"Ok",  
"result":"20052600000103"  
}

Sample Failure Response :  
{  
"request\_time":"16:03:29 28-05-2020",  
"stat":"Not\_Ok",  
"emsg":"Rejected : ORA:Order not found"  
}

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

cancel an order

Example:

orderno = ret['norenordno'] #from placeorder return value  
ret = api.cancel\_order(orderno=orderno)

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| norenordno\* |  | Noren order number, which needs to be modified |
| uid\* |  | User id of the logged in user. |

Response Details :

Response data will be in json format with below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Cancel order success or failure indication. |
| result |  | Noren Order number of the canceled order. |
| request\_time |  | Response received time. |
| emsg |  | This will be present only if Order cancelation fails |

Sample Success Response :  
{  
"request\_time":"14:14:10 26-05-2020",  
"stat":"Ok",  
"result":"20052600000103"  
}

Sample Failure Response :  
{  
"request\_time":"16:01:48 28-05-2020",  
"stat":"Not\_Ok",  
"emsg":"Rejected : ORA:Order not found to Cancel"  
}

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

exits a cover or bracket order

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| norenordno\* |  | Noren order number, which needs to be modified |
| prd\* | H / B | Allowed for only H and B products (Cover order and bracket order) |
| uid\* |  | User id of the logged in user. |

Response Details :

Response data will be in json format with below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Cancel order success or failure indication. |
| dmsg |  | Display message, (will be present only in case of success). |
| request\_time |  | Response received time. |
| emsg |  | This will be present only if Order cancelation fails |

#### <a name="md-prd\_convert"></a> position\_product\_conversion(exchange, tradingsymbol, quantity, new\_product\_type, previous\_product\_type, buy\_or\_sell, day\_or\_cf)

Convert a product of a position

Example:

ret = api.get\_positions()  
#converts the first position from existing product to intraday  
p = ret[0]  
ret = api.position\_product\_conversion(p['exch'], p['tsym'], p['netqty'], 'I', p['prd'], 'B', 'DAY')

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| exch\* |  | Exchange |
| tsym\* |  | Unique id of contract on which order was placed. Can’t be modified, must be the same as that of original order. (use url encoding to avoid special char error for symbols like M&M) |
| qty\* |  | Quantity to be converted. |
| uid\* |  | User id of the logged in user. |
| actid\* |  | Account id |
| prd\* |  | Product to which the user wants to convert position. |
| prevprd\* |  | Original product of the position. |
| trantype\* |  | Transaction type |
| postype\* | Day / CF | Converting Day or Carry forward position |
| ordersource | MOB | For Logging |

Response Details :

Response data will be in json format with below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Position conversion success or failure indication. |
| emsg |  | This will be present only if Position conversion fails. |

Sample Success Response :  
{  
"request\_time":"10:52:12 02-06-2020",  
"stat":"Ok"  
}

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"emsg":"Invalid Input : Invalid Position Type"  
}

#### <a name="md-get\_orderbook"></a> Order Book

List of Orders placed for the account

Example :

ret = api.get\_order\_book()  
print(ret)

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| prd | H / M / ... | Product name |

Response Details :

Response data will be in json Array of objects with below fields in case of success.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Order book success or failure indication. |
| exch |  | Exchange Segment |
| tsym |  | Trading symbol / contract on which order is placed. |
| norenordno |  | Noren Order Number |
| prc |  | Order Price |
| qty |  | Order Quantity |
| prd |  | Display product alias name, using prarr returned in user details. |
| status |  |  |
| trantype | B / S | Transaction type of the order |
| prctyp | LMT / MKT | Price type |
| fillshares |  | Total Traded Quantity of this order |
| avgprc |  | Average trade price of total traded quantity |
| rejreason |  | If order is rejected, reason in text form |
| exchordid |  | Exchange Order Number |
| cancelqty |  | Canceled quantity for order which is in status cancelled. |
| remarks |  | Any message Entered during order entry. |
| dscqty |  | Order disclosed quantity. |
| trgprc |  | Order trigger price |
| ret | DAY / IOC / EOS | Order validity |
| uid |  |  |
| actid |  |  |
| bpprc |  | Book Profit Price applicable only if product is selected as B (Bracket order ) |
| blprc |  | Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order ) |
| trailprc |  | Trailing Price applicable only if product is selected as H and B (High Leverage and Bracket order ) |
| amo |  | Yes / No |
| pp |  | Price precision |
| ti |  | Tick size |
| ls |  | Lot size |
| token |  | Contract Token |
| norentm |  |  |
| ordenttm |  |  |
| exch\_tm |  |  |
| snoordt |  | 0 for profit leg and 1 for stoploss leg |
| snonum |  | This field will be present for product H and B; and only if it is profit/sl order. |

Response data will be in json format with below fields in case of failure:

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Not\_Ok | Order book failure indication. |
| request\_time |  | Response received time. |
| emsg |  | Error message |

Sample Success Output :  
Success response :  
[  
{  
“stat” : “Ok”,  
“exch” : “NSE” ,  
“tsym” : “ACC-EQ” ,  
“norenordno” : “20062500000001223”,  
“prc” : “127230”,  
“qty” : “100”,  
“prd” : “C”,  
“status”: “Open”,  
“trantype” : “B”,  
“prctyp” : ”LMT”,  
“fillshares” : “0”,  
“avgprc” : “0”,  
“exchordid” : “250620000000343421”,  
“uid” : “VIDYA”,  
“actid” : “CLIENT1”,  
“ret” : “DAY”,  
“amo” : “Yes”  
},  
{  
“stat” : “Ok”,  
“exch” : “NSE” ,  
“tsym” : “ABB-EQ” ,  
“norenordno” : “20062500000002543”,  
“prc” : “127830”,  
“qty” : “50”,  
“prd” : “C”,  
“status”: “REJECT”,  
“trantype” : “B”,  
“prctyp” : ”LMT”,  
“fillshares” : “0”,  
“avgprc” : “0”,  
“rejreason” : “Insufficient funds”  
“uid” : “VIDYA”,  
“actid” : “CLIENT1”,  
“ret” : “DAY”,  
“amo” : “No”  
}  
]

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"emsg":"Session Expired : Invalid Session Key"  
}

#### <a name="md-get\_tradebook"></a> Trade Book

List of Trades of the account

Example:

ret = api.get\_trade\_book()  
print(ret)

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| actid\* |  | Account Id of logged in user |

Response Details :

Response data will be in json Array of objects with below fields in case of success.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Order book success or failure indication. |
| exch |  | Exchange Segment |
| tsym |  | Trading symbol / contract on which order is placed. |
| norenordno |  | Noren Order Number |
| qty |  | Order Quantity |
| prd |  | Display product alias name, using prarr returned in user details. |
| trantype | B / S | Transaction type of the order |
| prctyp | LMT / MKT | Price type |
| fillshares |  | Total Traded Quantity of this order |
| avgprc |  | Average trade price of total traded quantity |
| exchordid |  | Exchange Order Number |
| remarks |  | Any message Entered during order entry. |
| ret | DAY / IOC / EOS | Order validity |
| uid |  |  |
| actid |  |  |
| pp |  | Price precision |
| ti |  | Tick size |
| ls |  | Lot size |
| cstFrm |  | Custom Firm |
| fltm |  | Fill Time |
| flid |  | Fill ID |
| flqty |  | Fill Qty |
| flprc |  | Fill Price |
| ordersource |  | Order Source |
| token |  | Token |

Response data will be in json format with below fields in case of failure:

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Not\_Ok | Order book failure indication. |
| request\_time |  | Response received time. |
| emsg |  | Error message |

Sample Success Output :

[  
{  
"stat": "Ok",  
"norenordno": "20121300065715",  
"uid": "GURURAJ",  
"actid": "GURURAJ",  
"exch": "NSE",  
"prctyp": "LMT",  
"ret": "DAY",  
"prd": "M",  
"flid": "102",  
"fltm": "01-01-1980 00:00:00",  
"trantype": "S",  
"tsym": "ACCELYA-EQ",  
"qty": "180",  
"token": "7053",  
"fillshares": "180",  
"flqty": "180",  
"pp": "2",  
"ls": "1",  
"ti": "0.05",  
"prc": "800.00",  
"flprc": "800.00",  
"norentm": "19:59:32 13-12-2020",  
"exch\_tm": "00:00:00 01-01-1980",  
"remarks": "WC TEST Order",  
"exchordid": "6857"  
},  
{  
"stat": "Ok",  
"norenordno": "20121300065716",  
"uid": "GURURAJ",  
"actid": "GURURAJ",  
"exch": "NSE",  
"prctyp": "LMT",  
"ret": "DAY",  
"prd": "M",  
"flid": "101",  
"fltm": "01-01-1980 00:00:00",  
"trantype": "B",  
"tsym": "ACCELYA-EQ",  
"qty": "180",  
"token": "7053",  
"fillshares": "180",  
"flqty": "180",  
"pp": "2",  
"ls": "1",  
"ti": "0.05",  
"prc": "800.00",  
"flprc": "800.00",  
"norentm": "19:59:32 13-12-2020",  
"exch\_tm": "00:00:00 01-01-1980",  
"remarks": "WC TEST Order",  
"exchordid": "6858"  
}  
]

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

history an order

orderno = ret['norenordno'] #from placeorder return value  
ret = api.single\_order\_history(orderno=orderno)

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| norenordno\* |  | Noren Order Number |

Response Details :

Response data will be in json Array of objects with below fields in case of success.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Order book success or failure indication. |
| exch |  | Exchange Segment |
| tsym |  | Trading symbol / contract on which order is placed. |
| norenordno |  | Noren Order Number |
| prc |  | Order Price |
| qty |  | Order Quantity |
| prd |  | Display product alias name, using prarr returned in user details. |
| status |  |  |
| rpt |  | (fill/complete etc) |
| trantype | B / S | Transaction type of the order |
| prctyp | LMT / MKT | Price type |
| fillshares |  | Total Traded Quantity of this order |
| avgprc |  | Average trade price of total traded quantity |
| rejreason |  | If order is rejected, reason in text form |
| exchordid |  | Exchange Order Number |
| cancelqty |  | Canceled quantity for order which is in status cancelled. |
| remarks |  | Any message Entered during order entry. |
| dscqty |  | Order disclosed quantity. |
| trgprc |  | Order trigger price |
| ret | DAY / IOC / EOS | Order validity |
| uid |  |  |
| actid |  |  |
| bpprc |  | Book Profit Price applicable only if product is selected as B (Bracket order ) |
| blprc |  | Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order ) |
| trailprc |  | Trailing Price applicable only if product is selected as H and B (High Leverage and Bracket order ) |
| amo |  | Yes / No |
| pp |  | Price precision |
| ti |  | Tick size |
| ls |  | Lot size |
| token |  | Contract Token |
| norentm |  |  |
| ordenttm |  |  |
| exch\_tm |  |  |

Response data will be in json format with below fields in case of failure:

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Not\_Ok | Order book failure indication. |
| request\_time |  | Response received time. |
| emsg |  | Error message |

Sample Success Output :

[  
{  
"stat": "Ok",  
"norenordno": "20121300065716",  
"uid": "DEMO1",  
"actid": "DEMO1",  
"exch": "NSE",  
"tsym": "ACCELYA-EQ",  
"qty": "180",  
"trantype": "B",  
"prctyp": "LMT",  
"ret": "DAY",  
"token": "7053",  
"pp": "2",  
"ls": "1",  
"ti": "0.05",  
"prc": "800.00",  
"avgprc": "800.00",  
"dscqty": "0",  
"prd": "M",  
"status": "COMPLETE",  
"rpt": "Fill",  
"fillshares": "180",  
"norentm": "19:59:32 13-12-2020",  
"exch\_tm": "00:00:00 01-01-1980",  
"remarks": "WC TEST Order",  
"exchordid": "6858"  
},  
{  
"stat": "Ok",  
"norenordno": "20121300065716",  
"uid": "DEMO1",  
"actid": "DEMO1",  
"exch": "NSE",  
"tsym": "ACCELYA-EQ",  
"qty": "180",  
"trantype": "B",  
"prctyp": "LMT",  
"ret": "DAY",  
"token": "7053",  
"pp": "2",  
"ls": "1",  
"ti": "0.05",  
"prc": "800.00",  
"dscqty": "0",  
"prd": "M",  
"status": "OPEN",  
"rpt": "New",  
"norentm": "19:59:32 13-12-2020",  
"exch\_tm": "00:00:00 01-01-1980",  
"remarks": "WC TEST Order",  
"exchordid": "6858"  
},  
{  
"stat": "Ok",  
"norenordno": "20121300065716",  
"uid": "DEMO1",  
"actid": "DEMO1",  
"exch": "NSE",  
"tsym": "ACCELYA-EQ",  
"qty": "180",  
"trantype": "B",  
"prctyp": "LMT",  
"ret": "DAY",  
"token": "7053",  
"pp": "2",  
"ls": "1",  
"ti": "0.05",  
"prc": "800.00",  
"dscqty": "0",  
"prd": "M",  
"status": "PENDING",  
"rpt": "PendingNew",  
"norentm": "19:59:32 13-12-2020",  
"remarks": "WC TEST Order"  
},  
{  
"stat": "Ok",  
"norenordno": "20121300065716",  
"uid": "DEMO1",  
"actid": "DEMO1",  
"exch": "NSE",  
"tsym": "ACCELYA-EQ",  
"qty": "180",  
"trantype": "B",  
"prctyp": "LMT",  
"ret": "DAY",  
"token": "7053",  
"pp": "2",  
"ls": "1",  
"ti": "0.05",  
"prc": "800.00",  
"prd": "M",  
"status": "PENDING",  
"rpt": "NewAck",  
"norentm": "19:59:32 13-12-2020",  
"remarks": "WC TEST Order"  
}  
]

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

retrieves the holdings as a list

Example:

ret = api.get\_holdings()

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| actid\* |  | Account id of the logged in user. |
| prd\* |  | Product name |

Response Details :  
Response data will be in json format with below fields in case of Success:

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Holding request success or failure indication. |
| exch\_tsym |  | Array of objects exch\_tsym objects as defined below. |
| holdqty |  | Holding quantity |
| dpqty |  | DP Holding quantity |
| npoadqty |  | Non Poa display quantity |
| colqty |  | Collateral quantity |
| benqty |  | Beneficiary quantity |
| unplgdqty |  | Unpledged quantity |
| brkcolqty |  | Broker Collateral |
| btstqty |  | BTST quantity |
| btstcolqty |  | BTST Collateral quantity |
| usedqty |  | Holding used today |
| upldprc |  | Average price uploaded along with holdings |

Notes:  
Valuation : btstqty + holdqty + brkcolqty + unplgdqty + benqty + Max(npoadqty, dpqty) - usedqty  
Salable: btstqty + holdqty + unplgdqty + benqty + dpqty - usedqty

Exch\_tsym object:

|  |  |  |
| --- | --- | --- |
| Json Fields of object in values Array | Possible value | Description |
| exch | NSE, BSE, NFO ... | Exchange |
| tsym |  | Trading symbol of the scrip (contract) |
| token |  | Token of the scrip (contract) |
| pp |  | Price precision |
| ti |  | Tick size |
| ls |  | Lot size |

Response data will be in json format with below fields in case of failure:

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Not\_Ok | Position book request failure indication. |
| request\_time |  | Response received time. |
| emsg |  | Error message |

Sample Success Response :  
[  
{  
"stat":"Ok",  
"exch\_tsym":[  
{  
"exch":"NSE",  
"token":"13",  
"tsym":"ABB-EQ"  
}  
],  
"holdqty":"2000000",  
"colqty":"200",  
"btstqty":"0",  
"btstcolqty":"0",  
"usedqty":"0",  
"upldprc" : "1800.00"  
},  
{  
"stat":"Ok",  
"exch\_tsym":[  
{  
"exch":"NSE",  
"token":"22",  
"tsym":"ACC-EQ"  
}  
],  
"holdqty":"2000000",  
"colqty":"200",  
"btstqty":"0",  
"btstcolqty":"0",  
"usedqty":"0",  
"upldprc" : "1400.00"  
}  
]

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"emsg":"Invalid Input : Missing uid or actid or prd."  
}

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

retrieves the positions cf and day as a list

Example:

ret = api.get\_positions()  
mtm = 0  
pnl = 0  
for i in ret:  
 mtm += float(i['urmtom'])  
 pnl += float(i['rpnl'])  
 day\_m2m = mtm + pnl  
print(f'{day\_m2m} is your Daily MTM')

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| actid\* |  | Account id of the logged in user. |

Response Details :

Response data will be in json format with Array of Objects with below fields in case of success.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Position book success or failure indication. |
| exch |  | Exchange segment |
| tsym |  | Trading symbol / contract. |
| token |  | Contract token |
| uid |  | User Id |
| actid |  | Account Id |
| prd |  | Product name to be shown. |
| netqty |  | Net Position quantity |
| netavgprc |  | Net position average price |
| daybuyqty |  | Day Buy Quantity |
| daysellqty |  | Day Sell Quantity |
| daybuyavgprc |  | Day Buy average price |
| daysellavgprc |  | Day buy average price |
| daybuyamt |  | Day Buy Amount |
| daysellamt |  | Day Sell Amount |
| cfbuyqty |  | Carry Forward Buy Quantity |
| cforgavgprc |  | Original Avg Price |
| cfsellqty |  | Carry Forward Sell Quantity |
| cfbuyavgprc |  | Carry Forward Buy average price |
| cfsellavgprc |  | Carry Forward Buy average price |
| cfbuyamt |  | Carry Forward Buy Amount |
| cfsellamt |  | Carry Forward Sell Amount |
| lp |  | LTP |
| rpnl |  | RealizedPNL |
| urmtom |  | UnrealizedMTOM. |
| bep |  | Break even price |
| openbuyqty |  |  |
| opensellqty |  |  |
| openbuyamt |  |  |
| opensellamt |  |  |
| openbuyavgprc |  |  |
| opensellavgprc |  |  |
| mult |  |  |
| pp |  |  |
| prcftr |  | gn*pn/(gd*pd). |
| ti |  | Tick size |
| ls |  | Lot size |
| request\_time |  | This will be present only in a failure response. |

Response data will be in json format with below fields in case of failure:

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Not\_Ok | Position book request failure indication. |
| request\_time |  | Response received time. |
| emsg |  | Error message |

Sample Success Response :  
[  
{  
"stat":"Ok",  
"uid":"POORNA",  
"actid":"POORNA",  
"exch":"NSE",  
"tsym":"ACC-EQ",  
"prarr":"C",  
"pp":"2",  
"ls":"1",  
"ti":"5.00",  
"mult":"1",  
"prcftr":"1.000000",  
"daybuyqty":"2",  
"daysellqty":"2",  
"daybuyamt":"2610.00",  
"daybuyavgprc":"1305.00",  
"daysellamt":"2610.00",  
"daysellavgprc":"1305.00",  
"cfbuyqty":"0",  
"cfsellqty":"0",  
"cfbuyamt":"0.00",  
"cfbuyavgprc":"0.00",  
"cfsellamt":"0.00",  
"cfsellavgprc":"0.00",  
"openbuyqty":"0",  
"opensellqty":"23",  
"openbuyamt":"0.00",  
"openbuyavgprc":"0.00",  
"opensellamt":"30015.00",  
"opensellavgprc":"1305.00",  
"netqty":"0",  
"netavgprc":"0.00",  
"lp":"0.00",  
"urmtom":"0.00",  
"rpnl":"0.00",  
"cforgavgprc":"0.00"

}

]

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"request\_time":"14:14:11 26-05-2020",  
"emsg":"Error Occurred : 5 \"no data\""  
}

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

retrieves the margin and limits set

Request Details:

|  |  |  |  |
| --- | --- | --- | --- |
| 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. |

Sample Success Response :  
{  
"request\_time":"18:07:31 29-05-2020",  
"stat":"Ok",  
"cash":"1500000000000000.00",  
"payin":"0.00",  
"payout":"0.00",  
"brkcollamt":"0.00",  
"unclearedcash":"0.00",  
"daycash":"0.00",  
"turnoverlmt":"50000000000000.00",  
"pendordvallmt":"2000000000000000.00",  
"turnover":"3915000.00",  
"pendordval":"2871000.00",  
"marginused":"3945540.00",  
"mtomcurper":"0.00",  
"urmtom":"30540.00",  
"grexpo":"3915000.00",  
"uzpnl\_e\_i":"15270.00",  
"uzpnl\_e\_m":"61080.00",  
"uzpnl\_e\_c":"-45810.00"  
}

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"emsg":"Server Timeout : "  
}  
Market Info

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

Search for scrip or contract and its properties

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

Example:

exch = 'NFO'  
query = 'BANKNIFTY 30DEC CE' # multiple criteria to narrow results   
ret = api.searchscrip(exchange=exch, searchtext=query)  
  
if ret != None:  
 symbols = ret['values']  
 for symbol in symbols:  
 print('{0} token is {1}'.format(symbol['tsym'], symbol['token']))

Example 2:

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"  
 }  
 ]  
}

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| stext\* |  | Search Text |
| exch |  | Exchange (Select from ‘exarr’ Array provided in User Details response) |

Response Details :

Response data will be in json format with below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Market watch success or failure indication. |
| values |  | Array of json objects. (object fields given in below table) |
| emsg |  | This will be present only in case of errors. |

|  |  |  |
| --- | --- | --- |
| Json Fields of object in values Array | Possible value | Description |
| exch | NSE, BSE, NFO ... | Exchange |
| tsym |  | Trading symbol of the scrip (contract) |
| token |  | Token of the scrip (contract) |
| pp |  | Price precision |
| ti |  | Tick size |
| ls |  | Lot size |

Sample Success Response :

{  
"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"  
}  
]  
}

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"emsg":"No Data : "  
}

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

gets the complete details and its properties

Example:

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

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| exch |  | Exchange |
| token |  | Contract Token |

Response Details :

Response data will have below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| request\_time |  | It will be present only in a successful response. |
| stat | Ok or Not\_Ok | Market watch success or failure indication. |
| exch | NSE, BSE, NFO ... | Exchange |
| tsym |  | Trading Symbol |
| cname |  | Company Name |
| symnam |  | Symbol Name |
| seg |  | Segment |
| exd |  | Expiry Date |
| instname |  | Intrument Name |
| strprc |  | Strike Price |
| optt |  | Option Type |
| isin |  | ISIN |
| ti |  | Tick Size |
| ls |  | Lot Size |
| pp |  | Price precision |
| mult |  | Multiplier |
| gp\_nd |  | gn/gd \* pn/pd |
| prcunt |  | Price Units |
| prcqqty |  | Price Quote Qty |
| trdunt |  | Trade Units |
| delunt |  | Delivery Units |
| frzqty |  | Freeze Qty |
| gsmind |  | scripupdate Gsm Ind |
| elmbmrg |  | Elm Buy Margin |
| elmsmrg |  | Elm Sell Margin |
| addbmrg |  | Additional Long Margin |
| addsmrg |  | Additional Short Margin |
| splbmrg |  | Special Long Margin |
| splsmrg |  | Special Short Margin |
| delmrg |  | Delivery Margin |
| tenmrg |  | Tender Margin |
| tenstrd |  | Tender Start Date |
| tenendd |  | Tender End Eate |
| exestrd |  | Exercise Start Date |
| exeendd |  | Exercise End Date |
| elmmrg |  | Elm Margin |
| varmrg |  | Var Margin |
| expmrg |  | Exposure Margin |
| token |  | Contract Token |
| prcftr\_d |  | ((GN / GD) \* (PN/PD)) |

Sample Success Response :  
{  
"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": "",  
"delunt": "ACC",  
"token": "22",  
"varmrg": "40.00"  
}

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"request\_time":"10:50:54 10-12-2020",  
"emsg":"Error Occurred : 5 \"no data\""  
}

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

gets the complete details and its properties

Example:

exch = 'NSE'  
token = '22'  
ret = api.get\_quotes(exchange=exch, token=token)

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| exch |  | Exchange |
| token |  | Contract Token |

Response Details :

Response data will be in json format with below fields.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok or Not\_Ok | Watch list update success or failure indication. |
| request\_time |  | It will be present only in a successful response. |
| exch | NSE, BSE, NFO ... | Exchange |
| tsym |  | Trading Symbol |
| cname |  | Company Name |
| symname |  | Symbol Name |
| seg |  | Segment |
| instname |  | Instrument Name |
| isin |  | ISIN |
| pp |  | Price precision |
| ls |  | Lot Size |
| ti |  | Tick Size |
| mult |  | Multiplier |
| uc |  | Upper circuit limitlc |
| lc |  | Lower circuit limit |
| prcftr\_d |  | Price factor |
| token |  | Token |
| lp |  | LTP |
| o |  | Open Price |
| h |  | Day High Price |
| l |  | Day Low Price |
| v |  | Volume |
| ltq |  | Last trade quantity |
| ltt |  | Last trade time |
| bp1 |  | Best Buy Price 1 |
| sp1 |  | Best Sell Price 1 |
| bp2 |  | Best Buy Price 2 |
| sp2 |  | Best Sell Price 2 |
| bp3 |  | Best Buy Price 3 |
| sp3 |  | Best Sell Price 3 |
| bp4 |  | Best Buy Price 4 |
| sp4 |  | Best Sell Price 4 |
| bp5 |  | Best Buy Price 5 |
| sp5 |  | Best Sell Price 5 |
| bq1 |  | Best Buy Quantity 1 |
| sq1 |  | Best Sell Quantity 1 |
| bq2 |  | Best Buy Quantity 2 |
| sq2 |  | Best Sell Quantity 2 |
| bq3 |  | Best Buy Quantity 3 |
| sq3 |  | Best Sell Quantity 3 |
| bq4 |  | Best Buy Quantity 4 |
| sq4 |  | Best Sell Quantity 4 |
| bq5 |  | Best Buy Quantity 5 |
| sq5 |  | Best Sell Quantity 5 |
| bo1 |  | Best Buy Orders 1 |
| so1 |  | Best Sell Orders 1 |
| bo2 |  | Best Buy Orders 2 |
| so2 |  | Best Sell Orders 2 |
| bo3 |  | Best Buy Orders 3 |
| so3 |  | Best Sell Orders 3 |
| bo4 |  | Best Buy Orders 4 |
| so4 |  | Best Sell Orders 4 |
| bo5 |  | Best Buy Orders 5 |
| so5 |  | Best Sell Orders 5 |

Sample Success Response :  
{  
"request\_time":"12:05:21 18-05-2021",  
"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",  
"uc":"2093.95",  
"lc":"1713.25",  
"prcftr\_d":"(1 / 1 ) \* (1 / 1)",  
"token":"22",  
"lp":"0.00",  
"h":"0.00",  
"l":"0.00",  
"v":"0",  
"ltq":"0",  
"ltt":"05:30:00",  
"bp1":"2000.00",  
"sp1":"0.00",  
"bp2":"0.00",  
"sp2":"0.00",  
"bp3":"0.00",  
"sp3":"0.00",  
"bp4":"0.00",  
"sp4":"0.00",  
"bp5":"0.00",  
"sp5":"0.00",  
"bq1":"2",  
"sq1":"0",  
"bq2":"0",  
"sq2":"0",  
"bq3":"0",  
"sq3":"0",  
"bq4":"0",  
"sq4":"0",  
"bq5":"0",  
"sq5":"0",  
"bo1":"2",  
"so1":"0",  
"bo2":"0",  
"so2":"0",  
"bo3":"0",  
"so3":"0",  
"bo4":"0",  
"so4":"0",  
"bo5":"0",  
"So5":"0"  
}

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"request\_time":"10:50:54 10-12-2020",  
"emsg":"Error Occurred : 5 \"no data\""  
}

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

gets the chart date for the symbol

Example:

lastBusDay = datetime.datetime.today()  
lastBusDay = lastBusDay.replace(hour=0, minute=0, second=0, microsecond=0)  
ret = api.get\_time\_price\_series(exchange='NSE', token='22', starttime=lastBusDay.timestamp(), interval=5)

Request Details :

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| uid\* |  | Logged in User Id |
| exch\* |  | Exchange |
| token\* |  |  |
| st |  | Start time (seconds since 1 jan 1970) |
| et |  | End Time (seconds since 1 jan 1970) |
| intrv | “1”, ”3”, “5”, “10”, “15”, “30”, “60”, “120”, “240” | Candle size in minutes (optional field, if not given assume to be “1”) |

Response Details :

Response data will be in json format in case for failure.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Not\_Ok | TPData failure indication. |
| emsg |  | This will be present only in case of errors. |

Response data will be in json format in case for success.

|  |  |  |
| --- | --- | --- |
| Json Fields | Possible value | Description |
| stat | Ok | TPData success indication. |
| time |  | DD/MM/CCYY hh:mm:ss |
| into |  | Interval open |
| inth |  | Interval high |
| intl |  | Interval low |
| intc |  | Interval close |
| intvwap |  | Interval vwap |
| intv |  | Interval volume |
| v |  | volume |
| intoi |  | Interval io change |
| oi |  | oi |

Sample Success Response :  
[  
{  
"stat":"Ok",  
"time":"02-06-2020 15:46:23",  
"into":"0.00",  
"inth":"0.00",  
"intl":"0.00",  
"intc":"0.00",  
"intvwap":"0.00",  
"intv":"0",  
"intoi":"0",  
"v":"980515",  
"oi":"128702"  
},  
{  
"stat":"Ok",  
"time":"02-06-2020 15:45:23",  
"into":"0.00",  
"inth":"0.00",  
"intl":"0.00",  
"intc":"0.00",  
"intvwap":"0.00",  
"intv":"0",  
"intoi":"0",  
"v":"980515",  
"oi":"128702"  
},  
{  
"stat":"Ok",  
"time":"02-06-2020 15:44:23",  
"into":"0.00",  
"inth":"0.00",  
"intl":"0.00",  
"intc":"0.00",  
"intvwap":"0.00",  
"intv":"0",  
"intoi":"0",  
"v":"980515",  
"oi":"128702"  
},  
{  
"stat":"Ok",  
"time":"02-06-2020 15:43:23",  
"into":"1287.00",  
"inth":"1287.00",  
"intl":"0.00",  
"intc":"1287.00",  
"intvwap":"128702.00",  
"intv":"4",  
"intoi":"128702",  
"v":"980515",  
"oi":"128702"  
},  
{  
"stat":"Ok",  
"time":"02-06-2020 15:42:23",  
"into":"0.00",  
"inth":"0.00",  
"intl":"0.00",  
"intc":"0.00",  
"intvwap":"0.00",  
"intv":"0",  
"intoi":"0",  
"v":"980511",  
"oi":"128702"  
}  
]

Sample Failure Response :  
{  
"stat":"Not\_Ok",  
"emsg":"Session Expired : Invalid Session Key"  
}

#### <a name="md-get\_optionchain"></a> get\_option\_chain(exchange, tradingsymbol, strikeprice, count):

gets the chart date for the symbol

|  |  |  |  |
| --- | --- | --- | --- |
| Param | Type | Optional | Description |
| exchange | string | False | Exchange (UI need to check if exchange in NFO / CDS / MCX / or any other exchange which has options, if not don't allow) |
| tradingsymbol | string | False | Trading symbol of any of the option or future. Option chain for that underlying will be returned. (use url encoding to avoid special char error for symbols like M&M) |
| strikeprice | float | False | Mid price for option chain selection |
| count | int | True | Number of strike to return on one side of the mid price for PUT and CALL. (example cnt is 4, total 16 contracts will be returned, if cnt is is 5 total 20 contract will be returned) |

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 | False | Exchange |
| tsym | string | False | Trading Symbol of Contract |
| token | string | False | Contract token |
| optt | string | False | Option type |
| strprc | string | False | Strike Price |
| pp | string | False | Price Precision |
| ti | string | False | Tick Size |
| ls | string | False | Lot Size |

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

starts the websocket, WebSocket feed has 2 types of ticks( t=touchline d=depth)and 2 stages (k=acknowledgement, f=further change in tick).

|  |  |  |  |
| --- | --- | --- | --- |
| 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

Subscription Acknowledgement:

| Json Fields| Possible value| Description|  
| --- | --- | --- |  
| t | ok | ‘ok’ represents order update subscription acknowledgement |

Order Update subscription Updates :

| Json Fields | Possible value | Description |  
| --- | --- | --- |  
| t | om | ‘om’ represents touchline feed |  
| norenordno | | Noren Order Number |  
| uid | | User Id |  
| actid | | Account ID |  
| exch | | Exchange |  
| tsym | | Trading symbol |  
| qty | | Order quantity |  
| prc | | Order Price |  
| prd | | Product |  
| status | | Order status (New, Replaced, Complete, Rejected etc) |  
| reporttype | | Order event for which this message is sent out. (Fill, Rejected, Canceled) |  
| trantype | | Order transaction type, buy or sell |  
| prctyp | | Order price type (LMT, MKT, SL-LMT, SL-MKT) |  
| ret | | Order retention type (DAY, EOS, IOC,...) |  
| fillshares | | Total Filled shares for this order |  
| avgprc | | Average fill price |  
| fltm | | Fill Time(present only when reporttype is Fill) |  
| flid | | Fill ID (present only when reporttype is Fill) |  
| flqty | | Fill Qty(present only when reporttype is Fill) |  
| flprc | | Fill Price(present only when reporttype is Fill) |  
| rejreason | | Order rejection reason, if rejected |  
| exchordid | | Exchange Order ID |  
| cancelqty | | Canceled quantity, in case of canceled order |  
| remarks | | User added tag, while placing order |  
| dscqty | | Disclosed quantity |  
| trgprc | | Trigger price for SL orders |  
| snonum | | This will be present for child orders in case of cover and bracket orders, if present needs to be sent during exit |  
| snoordt | | This will be present for child orders in case of cover and bracket orders, it will indicate whether the order is profit or stoploss |  
| blprc | | This will be present for cover and bracket parent order. This is the differential stop loss trigger price to be entered. |  
| bpprc | | This will be present for bracket parent order. This is the differential profit price to be entered. |  
| trailprc | | This will be present for cover and bracket parent order. This is required if trailing ticks is to be enabled. |  
| exch\_tm | | This will have the exchange update time |

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

send a list of instruments to watch

t='tk' is sent once on subscription for each instrument. this will have all the fields with the most recent value  
thereon t='tf' is sent for fields that have changed.

For example  
quote event: 03-12-2021 11:54:44{'t': 'tk', 'e': 'NSE', 'tk': '11630', 'ts': 'NTPC-EQ', 'pp': '2', 'ls': '1', 'ti': '0.05', 'lp': '118.55', 'h': '118.65', 'l': '118.10', 'ap': '118.39', 'v': '162220', 'bp1': '118.45', 'sp1': '118.50', 'bq1': '26', 'sq1': '6325'}  
quote event: 03-12-2021 11:54:45{'t': 'tf', 'e': 'NSE', 'tk': '11630', 'lp': '118.45', 'ap': '118.40', 'v': '166637', 'sp1': '118.55', 'bq1': '3135', 'sq1': '30'}  
quote event: 03-12-2021 11:54:46{'t': 'tf', 'e': 'NSE', 'tk': '11630', 'lp': '118.60'}

in the example above we see first message t='tk' with all the values, 2nd message has lasttradeprice avg price and few other fields with value changed.. note bp1 isnt sent as its still 118.45  
in the next tick ( 3rd message) only last price is changed to 118.6

|  |  |  |  |
| --- | --- | --- | --- |
| Param | Type | Optional | Description |
| instruments | list | False | list of instruments [NSE\ |

Subscription Acknowledgement:

Number of Acknowledgements for a single subscription will be the same as the number of scrips mentioned in the key (k) field.

| Json Fields | Possible value | Description|  
| --- | --- | --- |  
| t | tk |‘tk’ represents touchline acknowledgement |  
| e |NSE, BSE, NFO ..|Exchange name |  
| tk |22|Scrip Token |  
| pp |2 for NSE, BSE & 4 for CDS USDINR|Price precision |  
| ts | | Trading Symbol |  
| ti | | Tick size |  
| ls | | Lot size |  
| lp | |LTP |  
| pc | |Percentage change |  
| v | | volume |  
| o | | Open price |  
| h | | High price |  
| l | | Low price |  
| c | | Close price |  
| ap | | Average trade price |  
| oi | | Open interest |  
| poi | | Previous day closing Open Interest |  
| toi | | Total open interest for underlying |  
| bq1 | | Best Buy Quantity 1 |  
| bp1 | | Best Buy Price 1 |  
| sq1 | | Best Sell Quantity 1 |  
| sp1 | | Best Sell Price 1|

TouchLine subscription Updates :  
Accept for t, e, and tk other fields may / may not be present.

| Json Fields | Possible value | Description|  
| --- | --- | --- |  
| t | tf |‘tf’ represents touchline acknowledgement |  
| e |NSE, BSE, NFO ..|Exchange name |  
| tk | 22 |Scrip Token |  
| lp | |LTP |  
| pc | |Percentage change |  
| v | | volume |  
| o | | Open price |  
| h | | High price |  
| l | | Low price |  
| c | | Close price |  
| ap | | Average trade price |  
| oi | | Open interest |  
| poi | | Previous day closing Open Interest |  
| toi | | Total open interest for underlying |  
| bq1 | | Best Buy Quantity 1 |  
| bp1 | | Best Buy Price 1 |  
| sq1 | | Best Sell Quantity 1 |  
| sp1 | | Best Sell Price 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='B',  
 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 (C) 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.