

About ProStocks

ProStocks, a Sunlight Broking LLP venture, is a leading online brokerage firm allowing customers to invest and trade in stocks listed in Indian Stock Market. Customers can trade at BSE and NSE in segments including Equity Cash (Intraday and Delivery), Equity Derivatives (Futures and Options), Currency Derivatives (Futures and Options) and Depository Services.

We provide the trading services at a very low brokerage fee in comparison to the traditional brokers. Average customer saves between 60% to 99% brokerage with us. We are able to offer services at lower fee because of efficiently managing our expenses and upfront investment in technology to reduce the cost of transactions as volume increases. We are online broker and do not have overheads like branches, RMs, expensive advertisement etc.

We are registered broker / member of BSE (Membership ID: [6635](#)), NSE (Membership ID: 90084 [Web](#) | [Pdf](#)) and depository participant of CDSL. We are SEBI registered Stock Broker (SEBI Registration no.: [INZ000048660](#)) and Depository Participant (SEBI Registration no.: [IN-DP-280-2016](#)).

Solution :

ProStocks Rest API with Web Socket to develop Web , Mobile, Desktop and Server to Server Trading Platform.

We do provide custom made solution for high frequency Algo trading for Hedge Funds , Family Office and Institutions.

TABLE OF CONTENTS

About ProStocks	1
Login and User Details	8
Login	8
Logout	10
Forgot Password	11
Change Password	12
Set Device Pin	13
Login with Device Pin	15
Get HS Token	16
Validate HS Token	17
User Details	18
Save FCM token	21
Watch Lists	22
Get WatchList Names	22
Get WatchList	23
Search Scrips	25
Add Scrip to Watch List	29
Delete Scrip to Watch List	30
Get SecurityInfo	31
Order and Trades	35
Place Order	35
Modify Order	38
Cancel Order	40
Exit SNO Order	41
Order Margin	42
Order Book	45
Multi Leg Order Book	49

Single Order History	52
Trade Book	58
Positions Book	62
Product Conversion	66
Holdings and Limits	68
Holdings	68
Limits	71
Market Info	79
Get Index List	79
Get Top List Names	82
Get Top List	84
Get Time Price Data (Chart data)	87
Get Option Chain	90
Exch Msg	92
Get Broker Msg	93
Span Calculator	94
Alerts	96
Set Alert	96
Cancel Alert	98
Modify Alert	99
Get Pending Alert	101
Place GTT Order	103
Modify GTT Order	105
Cancel GTT Order	107
Get Pending GTT Order	108
Get UnSettled Trading date:	110
Funds	112
Get Max Payout Amount:	112
Funds Payout Request	113
Web Socket API	115
General Guidelines	115

Connect	115
Subscribe Touchline	116
Unsubscribe Touchline	118
Subscribe Depth	119
Unsubscribe Depth	126
Subscribe Order Update	127
Unsubscribe Order Update	129

IMPORTANT LINKS

<https://github.com/Prostocks>

<https://www.prostocks.com/prostocks-star-api.html>

Version History

Version	Changes	Details
1.6.0	Option Chain added	Return 10 strikes above and below the given Price, to load the option chain screen.
	Span calculator added	Provides span and exposure value for given positions
1.7.0	Change password	dmsg added in case of successful password change
	Get and Validate HS token	Get HandShake Token And Validate HS token added. Help direct login facility into third party sites (BackOffice / DP etc)
1.8.0	Span Calculator fields corrected	
	Exit order added.	
1.9.0	Order margin API added	
	/SingleOrdHist and /GetIndexList added	
	Upper ckt limit and lower ckt limit added to security info api	
	52 week high / low added to depth	
1.10.0	SaveFCMToken api added.	
	Url encoded tsym support for special chars.	In Place, Modify Order, Order Margin, Option Chain, Production conversion.
1.11.0	OrderBook	Snonum, snoordt fields added in order book

	LoginApi	Added product enabled, exchange enabled, account id, watch list, user name
	User Details	User name added.
1.12.0	User Details / Login API / Holding API	Additional fields added to User Details / Login API / Holding API
1.14.0	Order Update	Missing fields added in Order Update web socket

Login and User Details

Login

Request to be POSTed to uri : **/NorenWClientTP/QuickAuth**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list

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 vendor_key
imei*		Send mac if users logs in for desktop, imei is from mobile
ipaddr		Optional field
source	WEB / MOB	

Example:


```
curl https://starapiuat.prostocks.com/NorenWClientTP/QuickAuth \
-d "jsonData={ \"apkversion\": \"1.0.0\", \"uid\": \"VIDYA\", \"pwd\": \"s3cur3ld\", \"factor2\": \"31-08-2017\", \"imei\": \"134243434\", \"source\": \"MOB\"}"
```

Response Details :

Response data will be in json format with below fields.

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. (Redirect to force change password if message is "Invalid Input : Password Expired" or "Invalid Input : Change Password")

Sample Success Response :

```
{
  "request_time": "20:18:47 19-05-2020",
  "stat": "Ok",
  "susertoken": "3b97f4c67762259a9ded6dbd7bfaf2787e662b3870422ddd343a59895f423a0",
  "lastaccesstime": "1589899727"
}
```

Sample Failure Response :

```
{
  "request_time": "20:32:14 19-05-2020",
  "stat": "Not_Ok",
  "emsg": "Invalid Input : Wrong Password"
}
```

Logout

Request to be POSTed to uri : **/NorenWClientTP/Logout**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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 : "
}
```

Forgot Password

Request to be POSTed to uri : **/NorenWClientTP/ForgotPassword**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list

Json Fields	Possible value	Description
uid*		User Id
pan*		Pan of the user
dob*		Date of birth

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Password reset is Success Or failure status
request_time		Response received time.
emsg		This will be present only if password reset fails. ("Invalid User or User Details")

Sample Success Response :

```
{
  "request_time":"10:52:56 28-05-2020",
  "stat":"Ok"
}
```

Sample Failure Response :

```
{
  "request_time":"17:42:13 26-05-2020",
  "stat":"Not_Ok",
  "emsg":"Error Occurred : Wrong user id or user details"
}
```

Change Password

Request to be POSTed to uri : **/NorenWClientTP/Changepwd**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list

Json Fields	Possible value	Description
uid*		User Id
oldpwd*		Sha256 of old password

pwd*		New password in plain text
------	--	----------------------------

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Password reset is Success Or failure status
request_time		Response received time.
dmsg		This will be present only in case of success. Number of days to expiry will be present in same.
emsg		This will be present only if password change fails

Sample Success Response :

```
{
  "request_time":"10:20:04 27-05-2020",
  "stat":"Ok",
  "dmsg":"Password Change Success. Your new password will expire in 15"
}
```

Sample Failure Response :

```
{
  "request_time":"10:21:09 27-05-2020",
  "stat":"Not_Ok",
  "emsg":"Error Occurred : Password already used"
}
```

Set Device Pin

Request to be POSTed to uri : **/NorenWClientTP/SetPin**

Request Details :

Parameter Name	Possible value	Description
----------------	----------------	-------------

jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User Id
imei*		Imei or device unique fingerprint
source*		Access type (MOB / WEB / API)
dpin*		New pin in plain text

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	If Pin setting is Success Or failure status
request_time		This will be present only if password change succeeds.
emsg		This will be present only if password change fails

Sample Success Response :

```
{
  "request_time": "14:59:43 27-05-2020",
  "stat": "Ok"
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

}

Login with Device Pin

Request to be POSTed to uri : **/NorenWClientTP/PinAuth**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list

Json Fields	Possible value	Description
uid*		User Id
imei*		Imei or device unique fingerprint
source*		Access type (MOB / WEB / API)
dpin*		sha256 of entered device pin
vc*		Vendor code provided by noren team, along with connection URLs
appkey*		Sha256 of uid vendor_key
apkversion*		Application version number
ipaddr		global Ip of internet access

Response Details :

Response data will be in json format with below fields.

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.
request_time		Response received time.
emsg		This will be present only if Login fails.

Sample Success Response :

```
{
  "request_time":"17:01:45 27-05-2020",
  "stat":"Ok",
  "susertoken":"b0856b3f6c4bac657417fc95de3e2060567b8bd80665e0a8ab82bbde5c434936",
  "lastaccesstime":"1590579105"
}
```

Sample Failure Response :

```
{
  "request_time":"11:19:56 28-05-2020",
  "stat":"Not_Ok",
  "emsg":"Invalid Input : Mpin Invalid"
}
```

Get HS Token

Request to be POSTed to uri : **/NorenWClientTP/GetHsToken**

Request Details :

Parameter Name	Possible value	Description
----------------	----------------	-------------

jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User Id

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	If Pin setting is Success Or failure status
request_time		This will be present only if password change succeeds.
hstk		One time Token to be sent to BackOffice or third party link
emsg		This will be present only if password change fails

Validate HS Token

Request to be POSTed to uri : **/NorenWClientTP/ValidateHsToken**

(To be used only from server, Call this url from Browser / Client Side APKs)

Request Details :

Parameter Name	Possible value	Description
LoginId*		Send sLoginId received from Initiator site

token*		Key Obtained on login success.
--------	--	--------------------------------

Response Details :

Response data will be in plain text format TRUE if Token is valid and FALSE for invalid User Id or Token.

External Integration (Backoffice Url..etc) Flow:

1. Trading site will call the third party url on user clicking the specified link (eg:Back Office login)
2. Trading site will pass the User id , Token and Client ID to the the third party url
3. Third Party application/web server will make a server call to our web server using this "Validate HS Token" Url.
4. If Trading site web server says ok then Third party application will provide access to the user/client

User Details

Request to be POSTed to uri : **/NorenWClientTP/UserDetails**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
-------------	----------------	-------------

stat	Ok or Not_Ok	User details success or failure indication.
exarr		Json array of strings with enabled exchange names
orarr		Json array of strings with enabled price types for user
prarr		Json array of Product Obj with enabled products, as defined below.
brkname		Broker id
brnchid		Branch id
email		
actid		
m_num		Mobile Number
uprev		Always it will be an INVESTOR, other types of user not allowed to login using this API.
request_time		It will be present only in a successful response.
emsg		This will be present only in case of errors.

Product Obj format

Json Fields	Possible value	Description
prd		Product name
s_prdt_ali		Product display name
exch		Json array of strings with enabled, allowed exchange names

Sample Success Response:

```
{
  "request_time": "20:20:04 19-05-2020",
  "prarr": [
    { "prd": "C",
      "s_prdt_ali": "Delivery",
      "exch": ["NSE", "BSE"]
    },
    { "prd": "I",
      "s_prdt_ali": "Intraday",
      "exch": ["NSE", "BSE", "NFO"]
    },
    { "prd": "H",
      "s_prdt_ali": "High Leverage",
      "exch": ["NSE", "BSE", "NFO"]
    },
    { "prd": "B",
      "s_prdt_ali": "Bracket Order",
      "exch": ["NSE", "BSE", "NFO"]
    }
  ],
  "exarr": [
    "NSE",
    "NFO"
  ],
  "orarr": [
    "MKT",
    "LMT",
    "SL-LMT",
    "SL-MKT",
    "DS",
    "2L",
    "3L",
    "4L"
  ],
  "brkname": "VIDYA",
  "brnchid": "VIDDU",
  "email": "gururaj@gmail.com",
  "actid": "GURURAJ",
  "uprev": "INVESTOR",
  "stat": "Ok"
}
```

Sample Failure Response:

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Save FCM token

Request to be POSTed to uri : **/NorenWClientTP/SaveFMCToken**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User Id
fcmtn*		FCM token collected from device

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	If save FCM token is Success Or failure status
request_time		This will be present only if password change succeeds.
emsg		This will be present only if save token fails

Sample Success Response :

```
{
  "request_time":"14:59:43 27-05-2020",
  "stat":"Ok"
}
```

Sample Failure Response :

```
{
  "stat":"Not_Ok",
  "emsg":"Session Expired : Invalid Session Key"
}
```

Watch Lists

Get WatchList Names

Request to be POSTed to uri : **/NorenWClientTP/MWList**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	MWList success or failure indication.
values		Watch List names as a json array of strings.
request_time		It will be present only in a successful response.
emsg		This will be present only in case of errors or No WatchLists are set yet.

Sample Success Response :

```
{
  "request_time": "12:34:52 21-05-2020",
  "values": [
    "default",
    "WL"
  ],
  "stat": "Ok"
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Get WatchList

Request to be POSTed to uri : **/NorenWClientTP/MarketWatch**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list

jKey*		Key Obtained on login success.
-------	--	--------------------------------

Json Fields	Possible value	Description
uid*		Logged in User Id
wlname*		Name of the Watchlist, for which scrip list is required.

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)
request_time		It will be present only in a successful response.
emsg		This will be present only in case of errors. That is : 1) Invalid Input : Invalid WatchList Name 2) Session Expired

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 :

```
{
  "request_time": "13:25:17 21-05-2020",
  "values": [
    {
      "exch": "BSE",
      "token": "972889",
      "tsym": "915PTCIF27"
    },
    {
      "exch": "NSE",
      "token": "13",
      "tsym": "ABB-EQ"
    },
    {
      "exch": "NSE",
      "token": "22",
      "tsym": "ACC-EQ"
    }
  ],
  "stat": "Ok"
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Invalid Input : Missing uid or wlname."
}
```

Search Scripts

Request to be POSTed to uri : **/NorenWClientTP/SearchScrip**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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. That is : 1) Invalid Input 2) Session Expired

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": "RELCONSNV-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": "RELVN20NAV-EQ"
}
]
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "No Data : "
}
```

Add Scrip to Watch List

Request to be POSTed to uri : **/NorenWClientTP/AddMultiScripsToMW**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id
wlname*		Name of the Watchlist, for which scrip list is required.
scrips*		List of scrips, example format NSE 22#BSE 506734

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.
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
{
  "request_time": "13:50:40 21-05-2020",
  "stat": "Ok"
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Delete Scrip to Watch List

Request to be POSTed to uri : **/NorenWClientTP/DeleteMultiMWScrips**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id
wlname*		Name of the Watchlist, for which scrip list is required.
scrips*		List of scrips, example format NSE 22#BSE 506734

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.
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
{
  "request_time": "13:50:40 21-05-2020",
  "stat": "Ok"
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Invalid Input : Missing uid or wlname or scrips."
}
```

Get SecurityInfo

Request to be POSTed to uri : **/NorenWClientTP/GetSecurityInfo**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id

exch		Exchange
token		Contract Token

Example:

```
jData={"uid": "{{USER_ID}}", "exch": "NSE",
"token": "22"}&jKey=c180c60ec8f7870490ec27f4acbdba179a149a5e5b1d503a0682d8b72acb24b3
```

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

splbmrq		Special Long Margin
splsmrq		Special Short Margin
delmrq		Delivery Margin
tenmrq		Tender Margin
tenstrd		Tender Start Date
tenendd		Tender End Date
exestrq		Exercise Start Date
exeendd		Exercise End Date
elmmrq		Elm Margin
varmrq		Var Margin
expmrq		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": "ACC.BO",
    "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\""
}

```

Order and Trades

Place Order

Request to be POSTed to uri : **/NorenWClientTP/PlaceOrder**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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

Example:

```
curl https://starapiuat.prostocks.com/NorenWClientTP/PlaceOrder \
  -d "jsonData={\"uid\": \"VIDYA\", \"actid\": \"CLIENT1\", \"exch\": \"NSE\", \"tsym\": \"ACC-EQ\", \"qty\": \"50\", \"price\": \"1400\", \"prd\": \"H\", \"trantype\": \"B\", \"prctyp\": \"LMT\", \"ret\": \"DAY\"}" \
  -d "jKey=GHUDWU53H32MTHPA536Q32WR"
```

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\""
}
```

Modify Order

Request to be POSTed to uri : **/NorenWClientTP/ModifyOrder**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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 Quantity to Fill / Order Qty - This is the total qty to be filled for the order. Its Open Qty/Pending Qty plus Filled Shares (cumulative for the order) for the order. * Please do not send only the pending qty in this field
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.
bprrc		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"
}
```

Cancel Order

Request to be POSTed to uri : **/NorenWClientTP/CancelOrder**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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"
}
```

Exit SNO Order

Request to be POSTed to uri : **/NorenWClientTP/ExitSNOOrder**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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

Order Margin

Request to be POSTed to uri : **/NorenWClientTP/GetOrderMargin**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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.
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	

blprc		Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order)
rorgqty		Optional field. Application only for modify order, open order quantity
fillshares		Optional field. Application only for modify order, quantity already filled.
rorgprc		Optional field. Application only for modify order, open order price
orgtrgprc		Optional field. Application only for modify order, open order trigger price
norenordno		Optional field. Application only for H or B order modification
snonum		Optional field. Application only for H or B order modification

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.
remarks		This field will be available only on success.
cash		Total credits available for order
marginused		Total margin used.
emsg		This will be present only if Order placement fails

Order Book

Request to be POSTed to uri : **/NorenWClientTP/OrderBook**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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

Example:

```
curl https://starapiuat.prostocks.com/NorenWClientTP/OrderBook \
  -d "jData={\"uid\": \"VIDYA\"}" \
  -d "jKey=GHUWU53H32MTHPA536Q32WR"
```

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		Order 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
orddtm		
ordenttm		
extm		
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": "2506200000000343421",
    "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"
}
```

Multi Leg Order Book

Request to be POSTed to uri : **/NorenWClientTP/MultiLegOrderBook**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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

Example:

```
curl https://starapiuat.prostocks.com/NorenWClientTP/MultiLegOrderBook \
  -d "jData={\"uid\": \"VIDYA\"}" \
  -d "jKey=GHUUDWU53H32MTHPA536Q32WR"
```

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		Order 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
tsym2		Trading symbol of second leg, mandatory for price type 2L and 3L
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
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
fillshares2		Total Traded Quantity of 2nd Leg
avgprc2		Average trade price of total traded quantity for 2nd leg
fillshares3		Total Traded Quantity of 3rd Leg
avgprc3		Average trade price of total traded quantity for 3rd leg

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

Single Order History

Request to be POSTed to uri : **/NorenWClientTP/SingleOrdHist**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
-------------	----------------	-------------

uid*		Logged in User Id
norenordno*		Noren Order Number

Example:

```
curl https://starapiuat.prostocks.com/NorenWClientTP/SingleOrdHist \
-d "jData={\"uid\": \"VIDYA\"} \" \
-d \"jKey=GHUDWU53H32MTHPA536Q32WR\"
```

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		Order status
rpt		Report Type (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
orddtm		
ordenttm		
extm		

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",
    "discqty": "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",
    "discqty": "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",
  "discqty": "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"
  }
]

```

Trade Book

Request to be POSTed to uri : **/NorenWClientTP/TradeBook**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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

Example:

```

curl https://starapiuat.prostocks.com/NorenWClientTP/TradeBook \
-d "jData={\"uid\": \"VIDYA\", \"actid\": \"DEMO1\"}" \
-d "jKey=GHUWU53H32MTHPA536Q32WR"

```

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"
  }
]

```

Positions Book

Request to be POSTed to uri : **/NorenWClientTP/PositionBook**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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

Example:

```

curl https://starapiuat.prostocks.com/NorenWClientTP/PositionBook \
  -d "jData={\"uid\": \"VIDYA\", \"actid\": \"ACCT_1\"}" \
  -d "jKey=GHUDWU53H32MTHPA536Q32WR"

```

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. (Can be recalculated in LTP update : = netqty * (lp from web socket - netavgprc) * prcfr
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",
    "prcfr": "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",
    "rpn": "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\""
}

```

Product Conversion

Request to be POSTed to uri : **/NorenWClientTP/ProductConversion**

Request Details :

Parameter Name	Possible value	Description
----------------	----------------	-------------

jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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"
}
```

Holdings and Limits

Holdings

Request to be POSTed to uri : **/NorenWClientTP/Holdings**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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
btstqty		BTST quantity
btstcolqty		BTST Collateral quantity
usedqty		Holding used today
upldprc		Average price uploaded along with holdings

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."
}

```

Limits

Request to be POSTed to uri : **/NorenWClientTP/Limits**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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

seg	CM / FO / FX	Segment
exch		Exchange

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Limits request success or failure indication.
actid		Account id
prd		Product name
seg	CM / FO / FX	Segment
exch		Exchange
-----Cash Primary Fields-----		
cash		Cash Margin available
payin		Total Amount transferred using Payins today
payout		Total amount requested for withdrawal today
-----Cash Additional Fields-----		
brkcollamt		Prevalued Collateral Amount
unclearedcash		Uncleared Cash (Payin through cheques)

daycash		Additional leverage amount / Amount added to handle system errors - by broker.
-----Margin Utilized-----		
marginused		Total margin / fund used today
mtomcurper		Mtom current percentage
-----Margin Used components-----		
cbu		CAC Buy used
csc		CAC Sell Credits
rpnl		Current realized PNL
unmtom		Current unrealized mtom
marprt		Covered Product margins
span		Span used
expo		Exposure margin
premium		Premium used
varelm		Var Elm Margin
grexpo		Gross Exposure
grexpo_d		Gross Exposure derivative
scripskmar		Scrip basket margin

addscripskmrg		Additional scrip basket margin
brokerage		Brokerage amount
collateral		Collateral calculated based on uploaded holdings
grcoll		Valuation of uploaded holding pre haircut
-----Additional Risk Limits-----		
turnoverlmt		
pendordvallmt		
-----Additional Risk Indicators-----		
turnover		Turnover
pendordval		Pending Order value
-----Margin used detailed breakup fields-----		
rzpnl_e_i		Current realized PNL (Equity Intraday)
rzpnl_e_m		Current realized PNL (Equity Margin)
rzpnl_e_c		Current realized PNL (Equity Cash n Carry)
rzpnl_d_i		Current realized PNL (Derivative Intraday)
rzpnl_d_m		Current realized PNL (Derivative Margin)
rzpnl_f_i		Current realized PNL (FX Intraday)

rzpnl_f_m		Current realized PNL (FX Margin)
rzpnl_c_i		Current realized PNL (Commodity Intraday)
rzpnl_c_m		Current realized PNL (Commodity Margin)
uzpnl_e_i		Current unrealized MTOM (Equity Intraday)
uzpnl_e_m		Current unrealized MTOM (Equity Margin)
uzpnl_e_c		Current unrealized MTOM (Equity Cash n Carry)
uzpnl_d_i		Current unrealized MTOM (Derivative Intraday)
uzpnl_d_m		Current unrealized MTOM (Derivative Margin)
uzpnl_f_i		Current unrealized MTOM (FX Intraday)
uzpnl_f_m		Current unrealized MTOM (FX Margin)
uzpnl_c_i		Current unrealized MTOM (Commodity Intraday)
uzpnl_c_m		Current unrealized MTOM (Commodity Margin)
span_d_i		Span Margin (Derivative Intraday)
span_d_m		Span Margin (Derivative Margin)
span_f_i		Span Margin (FX Intraday)
span_f_m		Span Margin (FX Margin)
span_c_i		Span Margin (Commodity Intraday)

span_c_m		Span Margin (Commodity Margin)
expo_d_i		Exposure Margin (Derivative Intraday)
expo_d_m		Exposure Margin (Derivative Margin)
expo_f_i		Exposure Margin (FX Intraday)
expo_f_m		Exposure Margin (FX Margin)
expo_c_i		Exposure Margin (Commodity Intraday)
expo_c_m		Exposure Margin (Commodity Margin)
premium_d_i		Option premium (Derivative Intraday)
premium_d_m		Option premium (Derivative Margin)
premium_f_i		Option premium (FX Intraday)
premium_f_m		Option premium (FX Margin)
premium_c_i		Option premium (Commodity Intraday)
premium_c_m		Option premium (Commodity Margin)
varelm_e_i		Var Elm (Equity Intraday)
varelm_e_m		Var Elm (Equity Margin)
varelm_e_c		Var Elm (Equity Cash n Carry)
marprt_e_h		Covered Product margins (Equity High leverage)

marprt_e_b		Covered Product margins (Equity Bracket Order)
marprt_d_h		Covered Product margins (Derivative High leverage)
marprt_d_b		Covered Product margins (Derivative Bracket Order)
marprt_f_h		Covered Product margins (FX High leverage)
marprt_f_b		Covered Product margins (FX Bracket Order)
marprt_c_h		Covered Product margins (Commodity High leverage)
marprt_c_b		Covered Product margins (Commodity Bracket Order)
scripbskmar_e_i		Scrip basket margin (Equity Intraday)
scripbskmar_e_m		Scrip basket margin (Equity Margin)
scripbskmar_e_c		Scrip basket margin (Equity Cash n Carry)
addscripbskmrg_d_i		Additional scrip basket margin (Derivative Intraday)
addscripbskmrg_d_m		Additional scrip basket margin (Derivative Margin)
addscripbskmrg_f_i		Additional scrip basket margin (FX Intraday)
addscripbskmrg_f_m		Additional scrip basket margin (FX Margin)
addscripbskmrg_c_i		Additional scrip basket margin (Commodity Intraday)
addscripbskmrg_c_m		Additional scrip basket margin (Commodity Margin)
brkage_e_i		Brokerage (Equity Intraday)

brkage_e_m		Brokerage (Equity Margin)
brkage_e_c		Brokerage (Equity CAC)
brkage_e_h		Brokerage (Equity High Leverage)
brkage_e_b		Brokerage (Equity Bracket Order)
brkage_d_i		Brokerage (Derivative Intraday)
brkage_d_m		Brokerage (Derivative Margin)
brkage_d_h		Brokerage (Derivative High Leverage)
brkage_d_b		Brokerage (Derivative Bracket Order)
brkage_f_i		Brokerage (FX Intraday)
brkage_f_m		Brokerage (FX Margin)
brkage_f_h		Brokerage (FX High Leverage)
brkage_f_b		Brokerage (FX Bracket Order)
brkage_c_i		Brokerage (Commodity Intraday)
brkage_c_m		Brokerage (Commodity Margin)
brkage_c_h		Brokerage (Commodity High Leverage)
brkage_c_b		Brokerage (Commodity Bracket Order)
request_time		This will be present only in a successful response.

emsg		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":"200000000000000.00",
  "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

Get Index List

Request to be POSTed to uri : **/NorenWClientTP/GetIndexList**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list

jKey*		Key Obtained on login success.
-------	--	--------------------------------

Json Fields	Possible value	Description
uid*		Logged in User Id
exch*		Exchange

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	TopListNames success or failure indication.
values		Array Of Basket, Criteria pair.
request_time		This will be present only in a successful response.
emsg		This will be present only in case of errors.

Basket, Criteria pair Object :

Json Fields	Possible value	Description
idxname		Index Name
token		Index token used to subscribe

Sample Output:

```
{
  "request_time": "20:12:29 13-12-2020",
```



```
"values": [  
  {  
    "idxname": "HangSeng BeES-NAV",  
    "token": "26016"  
  },  
  {  
    "idxname": "India VIX",  
    "token": "26017"  
  },  
  {  
    "idxname": "Nifty 50",  
    "token": "26000"  
  },  
  {  
    "idxname": "Nifty IT",  
    "token": "26008"  
  },  
  {  
    "idxname": "Nifty Next 50",  
    "token": "26013"  
  },  
  {  
    "idxname": "Nifty Bank",  
    "token": "26009"  
  },  
  {  
    "idxname": "Nifty 500",  
    "token": "26004"  
  },  
  {  
    "idxname": "Nifty 100",  
    "token": "26012"  
  },  
  {  
    "idxname": "Nifty Midcap 50",  
    "token": "26014"  
  },  
  {  
    "idxname": "Nifty Realty",  
    "token": "26018"  
  }  
]
```

```

    },
  ]
}

```

Get Top List Names

Request to be POSTed to uri : **/NorenWClientTP/TopListName**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id
exch*		Exchange

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	TopListNames success or failure indication.
values		Array Of Basket, Criteria pair.
request_time		This will be present only in a successful response.

emsg		This will be present only in case of errors.
------	--	--

Basket, Criteria pair Object :

Json Fields	Possible value	Description
basket		Basket name
crt		criteria

Sample Success Response :

```
{
  "request_time": "13:08:22 03-06-2020",
  "values": [
    {
      "basket": "NSEBL",
      "crt": "VOLUME"
    },
    {
      "basket": "NSEBL",
      "crt": "LTP"
    },
    {
      "basket": "NSEBL",
      "crt": "VALUE"
    },
    {
      "basket": "NSEEQ",
      "crt": "VOLUME"
    },
    {
      "basket": "NSEEQ",
      "crt": "LTP"
    },
    {
      "basket": "NSEEQ",
      "crt": "VALUE"
    },
    {
      "basket": "NSEALL",
      "crt": "VOLUME"
    },
    {
      "basket": "NSEALL",

```

```

    "crt": "LTP"
  },
  {
    "bskt": "NSEALL",
    "crt": "VALUE"
  }
]
}

```

Sample Failure Response :

```

{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}

```

Get Top List

Request to be POSTed to uri : **/NorenWClientTP/TopList**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id
exch*		Exchange
tb*	T or B	Top or Bottom
bskt*		Basket name
crt*		criteria

Response Details :

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	TopList success or failure indication.
values		Array of top / bottom contracts object
request_time		This will be present only in a successful response.
emsg		This will be present only in case of errors.

top / bottom contracts object :

Json Fields	Possible value	Description
tsym		Trading symbol
lp		LTP
c		Previous Close price
v		volume
value		Total traded value
oi		Open interest
pc		LTP percentage change

Sample Success Response :

[

```
{
  "stat": "Ok",
  "request_time": "15:44:45 03-06-2020",
  "values": [
    {
      "tsym": "AIRAN-EQ",
      "lp": "950.00",
      "c": "915.00",
      "v": "42705",
      "value": "40185405.00",
      "oi": "0",
      "Pc": "3.83"
    },
    {
      "tsym": "SHRENIK-EQ",
      "lp": "1850.00",
      "c": "1785.00",
      "v": "206846",
      "value": "368806418.00",
      "oi": "0",
      "Pc": "3.64"
    },
    {
      "tsym": "REMSONSIND-EQ",
      "lp": "6000.00",
      "c": "5795.00",
      "v": "3948",
      "value": "22752324.00",
      "oi": "0",
      "pc": "3.54"
    },
    {
      "tsym": "AXISNIFTY-EQ",
      "lp": "106700.00",
      "c": "103301.00",
      "v": "422",
      "value": "43825544.00",
      "oi": "0",
      "Pc": "3.29"
    }
  ]
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Invalid Input : Missing uid or exch or bskt or tb or crt"
}
```

Get Time Price Data (Chart data)

Request to be POSTed to uri : **/NorenWClientTP/TPSeries**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

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)

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"
}

```

Get Option Chain

Request to be POSTed to uri : **/NorenWClientTP/GetOptionChain**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id
tsym*		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)
exch*		Exchange (UI need to check if exchange in NFO / CDS / MCX / or any other exchange which has options, if not don't allow)
strprc*		Mid price for option chain selection

cnt*		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 5 total 20 contract will be returned)
------	--	---

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. That is : 1) Invalid Input 2) Session Expired

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)
optt		Option Type
strprc		Strike price
pp		Price precision
ti		Tick size

ls		Lot size
----	--	----------

Exch Msg

Request to be POSTed to uri : **/NorenWClientTP/ExchMsg**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id
exch		Exchange (Select from 'exarr' Array provided in User Details response)

Example:

```
curl https://starapiuat.prostocks.com/NorenWClientTP/ExchMsg \
  -d "jData={\"uid\": \"VIDYA\"}" \
  -d "jKey=NSE"
```

Response Details :

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

Json Fields	Possible value	Description
stat	Ok	Whi Exch Msg success or failure indication.
exchmsg		It will be present only in a successful response.

exchtm		Exchange Time
--------	--	---------------

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

Get Broker Msg

Request to be POSTed to uri : **/NorenWClientTP/GetBrokerMsg**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id

Response Details :

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

Json Fields	Possible value	Description
stat	Ok	Broker Msg success or failure indication.
dmsg		This will be present only in case of success. Number of days to expiry will be present in same.
norentm		Noren Time

Sample Success Response :

```
[
  {
    "stat": "Ok",
    "norentm": "02-05-1975 08:48:52",
    "msgtyp": "Admin Message",
    "dmsg": "Test Msg All Message Recovery2"
  },
  {
    "stat": "Ok",
    "norentm": "02-05-1975 08:48:52",
    "msgtyp": "Admin Message",
    "dmsg": "Test Msg All Message Recovery2"
  }
]
```

Span Calculator

Request to be POSTed to uri : **/NorenWClientTP/SpanCalc**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list

Json Fields	Possible value	Description
actid*		Any Account id, preferably actual account id if sending from post login screen
pos*		Array of json objects. (object fields given in below table)

Json Fields of object in values Array	Possible value	Description
exch	NFO, CDS, MCX ...	Exchange
instname	FUTSTK, FUTIDX, OPTSTK, FUTCUR...	Instrument name
symname	USDINR, ACC, ABB, NIFTY..	Symbol name
expd	2020-10-29	YYYY-MM-DD format
optt	CE, PE	Option Type
strprc	11900.00, 71.0025	Strike price
buyqty		Buy Open Quantity
sellqty		Sell Open Quantity
netqty		Net traded quantity

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.
span		Span value
expo		Exposure margin
span_trade		Span value ignoring input fields buyqty, sellqty
expo_trade		Exposure margin ignoring input fields buyqty, sellqty

Alerts

Set Alert

Request to be POSTed to uri : **/NorenWClientTP/SetAlert**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User id of the logged in user.

tsym*		Trading symbol
exch*		Exchange Segment
ai_t*		Alert Type
validity*	DAY or GTT	Validity
d		Data to be compared with LTP
remarks*		Any message Entered during order entry.

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		alert success or failure indication.
request_time		This will be present only in a successful response.
al_id		Alert Id
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
[
  {
    "request_time":"11:22:26 08-04-2021",
    "stat":"Oi created",
    "al_id":"21040800000004"
  }
]
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Cancel Alert

Request to be POSTed to uri : **/NorenWClientTP/CancelAlert**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid		User id of the logged in user.
al_id*		Alert Id

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		alert success or failure indication.
request_time		This will be present only in a successful response.

al_id		Alert Id
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
[
  {
    "request_time": "15:03:33 08-04-2021",
    "stat": "Oi delete success",
    "al_id": "21040800000008"
  }
]
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Modify Alert

Request to be POSTed to uri : **/NorenWClientTP/ModifyAlert**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User id of the logged in user.

tsym*		Trading symbol
exch*		Exchange Segment
ai_t*		Alert Type
al_id		Alert Id
validity*	DAY or GTT	Validity
d		Data to be compared with LTP
remarks*		Any message Entered during order entry.

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		alert success or failure indication.
request_time		This will be present only in a successful response.
al_id		Alert Id
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
[
  {
    "request_time":"16:36:42 08-04-2021",
    "stat":"Oi Replaced",
    "al_id":"21040800000013"
```

```
}
]
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Get Pending Alert

Request to be POSTed to uri : **/NorenWClientTP/GetPendingAlert**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User id of the logged in user.

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		alert success or failure indication.
ai_t		Alert type

al_id		Alert Id
tsym		Trading symbol
exch		Exchange Segment
token		Contract token
remarks		Any message Entered during order entry.
validity	DAY or GTT	Validity
d		Data to be compared with LTP
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
[
  {
    "Stat": "ok",
    "ai_t": "LTP_A",
    "al_id": "21040800000008",
    "tsym": "ACC-EQ",
    "exch": "NSE",
    "token": "22",
    "remarks": "test",
    "validity": "DAY",
    "d": "95000.00"
  }
]
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Place GTT Order

Request to be POSTed to uri : **/NorenWClientTP/PlaceGTTOrder**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User id of the logged in user.
tsym*		Trading symbol
exch*		Exchange Segment
ai_t*		Alert Type
validity*	DAY or GTT	Validity
d		Data to be compared with LTP
remarks*		Any message Entered during order entry.
trantype*	B / S	B -> BUY, S -> SELL
prctyp*	LMT / MKT / SL- LMT / SL-MKT / DS / 2L / 3L	
prd*	C / M / H	Product name

ret*	DAY / EOS / IOC	Retention type (Show options as per allowed exchanges)
actid*		Login users account ID
qty*		Order Quantity
prc*		Order Price
dscqty		Disclosed quantity (Max 10% for NSE, and 50% for MCX)

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		GTT order success or failure indication.
request_time		This will be present only in a successful response.
al_id		Alert Id
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
[
  {
    "request_time":"10:02:06 15-04-2021",
    "stat":"Oi created",
    "Al_id":"21041500000010"
  }
]
```

Sample Failure Response :

```
{
```



```

"stat": "Not_Ok",
"emsg": "Session Expired : Invalid Session Key"
}

```

Modify GTT Order

Request to be POSTed to uri : **/NorenWClientTP/ModifyGTTOrder**

Json Fields	Possible value	Description
uid*		User id of the logged in user.
tsym*		Trading symbol
exch*		Exchange Segment
ai_t*		Alert Type
al_id		Alert Id
validity*	DAY or GTT	Validity
d		Data to be compared with LTP
remarks*		Any message Entered during order entry.
trantype*	B / S	B -> BUY, S -> SELL
prctyp*	LMT / MKT / SL- LMT / SL-MKT / DS / 2L / 3L	
prd*	C / M / H	Product name

ret*	DAY / EOS / IOC	Retention type (Show options as per allowed exchanges)
actid*		Login users account ID
qty*		Order Quantity
prc*		Order Price
dscqty		Disclosed quantity (Max 10% for NSE, and 50% for MCX)

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		GTT order success or failure indication.
request_time		This will be present only in a successful response.
al_id		Alert Id
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
[
  {
    "request_time":"12:15:18 15-04-2021",
    "stat":"Oi Replacedt",
    "Al_id":"21041500000008"
  }
]
```

Sample Failure Response :

```
{
  "stat":"Not_Ok",
```

```
"emsg": "Session Expired : Invalid Session Key"
}
```

Cancel GTT Order

Request to be POSTed to uri : **/NorenWClientTP/CancelGTTOrder**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid		User id of the logged in user.
al_id*		Alert Id

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		GTT order success or failure indication.
request_time		This will be present only in a successful response.
al_id		Alert Id

emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired
------	--	--

Sample Success Response :

```
[
  {
    "request_time":"12:20:01 15-04-2021",
    "stat":"Oi delete success",
    "Al_id":"21041500000013"
  }
]
```

Sample Failure Response :

```
{
  "stat":"Not_Ok",
  "emsg":"Session Expired : Invalid Session Key"
}
```

Get Pending GTT Order

Request to be POSTed to uri : **/NorenWClientTP/GetPendingGTTOrder**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User id of the logged in user.

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		alert success or failure indication.
ai_t		Alert type
al_id		Alert Id
tsym		Trading symbol
exch		Exchange Segment
token		Contract token
remarks		Any message Entered during order entry.
validity	DAY or GTT	Validity
d		Data to be compared with LTP
trantype	B / S	B -> BUY, S -> SELL
prctyp	LMT / MKT / SL-LMT / SL- MKT / DS / 2L / 3L	
prd	C / M / H	Product name
ret	DAY / EOS / IOC	Retention type (Show options as per allowed exchanges)
actid		Login users account ID
qty		Order Quantity

prc		Order Price
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :

```
[
  {
    "stat": "Ok",
    "ai_t": "LTP_A",
    "AI_id": "210415000000002",
    "tsym": "ACC-EQ",
    "exch": "NSE",
    "Token": "22",
    "Remarks": "test",
    "validity": "DAY",
    "actid": "MOHINI",
    "trantype": "B",
    "prctyp": "LMT",
    "Qty": 1,
    "Prc": "1305.00",
    "C": "C",
    "prd": "C",
    "ordersource": "MOB",
    "d": "1900.00"
  }
]
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Get UnSettled Trading date:

Request to be POSTed to uri : **/NorenWClientTP/GetUnStledTradingDate**

Request Details :

Parameter Name	Possible value	Description
----------------	----------------	-------------

jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User id of the logged in user.

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		success or failure indication.
request_time		This will be present only in a successful response.
trd_date		Array of objects (trade date as defined below)

Sample Success Response :

```
{
  "stat": "Ok",
  "Request_time": "10052021152900",
  "trd_date": [
    {
      "trd_date": "28-04-2021"
    },
    {
      "trd_date": "29-04-2021"
    }
  ]
}
```

```

    },
    {
      "trd_date": "30-04-2021"
    }
  ]
}

```

Sample Failure Response :

```

{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}

```

Funds

Get Max Payout Amount:

Request to be POSTed to uri : **/NorenWClientTP/GetMaxPayoutAmount**

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		User id of the logged in user.
actid		Login users account ID

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		success or failure indication.
request_time		This will be present only in a successful response.
actid		Account id
payout		Maximum payout amount

Sample Success Response :

```
{
  "request_time":"15:52:26 10-05-2021",
  "stat":"Ok",
  "actid":"C-GURURAJ",
  "payout":"21200.20"
}
```

Sample Failure Response :

```
{
  "stat":"Not_Ok",
  "emsg":"Session Expired : Invalid Session Key"
}
```

Funds Payout Request

Request to be POSTed to uri : /NorenWClientTP/FundsPayOutReq

Request Details :

Parameter Name	Possible value	Description
jData*		Should send json object with fields in below list

jKey*		Key Obtained on login success.
-------	--	--------------------------------

Json Fields	Possible value	Description
uid*		User id of the logged in user.
actid		Login users account ID
payout		payout amount
remarks		Any message Entered during order entry.

Response Details :

Response data will have below fields.

Json Fields	Possible value	Description
stat		Tran status
request_time		This will be present only in a successful response.
trn_id		Tran id

Sample Success Response :

```
{
  "request_time":"15:52:27 10-05-2021",
  "trn_id":"20211300000030",
  "stat":"W"
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Session Expired : Invalid Session Key"
}
```

Web Socket API

Connect to wss://starapiuat.prostocks.com/NorenWSTP

General Guidelines

- 1) As soon as connection is done, a connection request should be sent with User id and login session id.
- 2) All input and output messages will be in json format.

Connect

Request:

Json Fields	Possible value	Description
t	c	'c' represents connect task
uid		User ID
actid		Account id
source	WEB / MOB	Source should be same as login request.
susertoken		User Session Token

Response :

Json Fields	Possible value	Description
t	ck	'ck' represents connect acknowledgement
uid		User ID
s		Ok or Not_Ok(in case of invalid user id or session id)

Subscribe Touchline

Request :

Json Fields	Possible value	Description
t	t	't' represents touchline task
k		One or more scriplist for subscription. Example NSE 22#BSE 508123#NSE NIFTY

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

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 feed
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

Unsubscribe Touchline

Request:

Json Fields	Possible value	Description
t	u	'u' represents Unsubscribe Touchline
k		One or more scriplist for unsubscription. Example NSE 22#BSE 508123

Response :

Json Fields	Possible value	Description
t	uk	'uk' represents Unsubscribe Touchline acknowledgement

k		One or more scriplist for unsubscription. Example NSE 22#BSE 508123
---	--	--

Subscribe Depth

Request :

Json Fields	Possible value	Description
t	d	'd' represents depth subscription
k		One or more scriplist for subscription. Example NSE 22#BSE 508123

Subscription Depth 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	dk	'dk' represents depth 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
ltt		Last trade time
ltq		Last trade quantity
tbq		Total Buy Quantity
tsq		Total Sell Quantity
bq1		Best Buy Quantity 1
bq2		Best Buy Quantity 2
bq3		Best Buy Quantity 3
bq4		Best Buy Quantity 4
bq5		Best Buy Quantity 5
bp1		Best Buy Price 1
bp2		Best Buy Price 2
bp3		Best Buy Price 3

bp4		Best Buy Price 4
bp5		Best Buy Price 5
bo1		Best Buy Orders 1
bo2		Best Buy Orders 2
bo3		Best Buy Orders 3
bo4		Best Buy Orders 4
bo5		Best Buy Orders 5
sq1		Best Sell Quantity 1
sq2		Best Sell Quantity 2
sq3		Best Sell Quantity 3
sq4		Best Sell Quantity 4
sq5		Best Sell Quantity 5
sp1		Best Sell Price 1
sp2		Best Sell Price 2
sp3		Best Sell Price 3
sp4		Best Sell Price 4
sp5		Best Sell Price 5

so1		Best Sell Orders 1
so2		Best Sell Orders 2
so3		Best Sell Orders 3
so4		Best Sell Orders 4
so5		Best Sell Orders 5
lc		Lower Circuit Limit
uc		Upper Circuit Limit
52h		52 week high low in other exchanges, Life time high low in mcx
52l		52 week high low in other exchanges, Life time high low in mcx

Depth subscription Updates :

Json Fields	Possible value	Description
t	df	'df' represents depth feed
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
ltt		Last trade time
ltq		Last trade quantity
tbq		Total Buy Quantity
tsq		Total Sell Quantity
bq1		Best Buy Quantity 1
bq2		Best Buy Quantity 2
bq3		Best Buy Quantity 3
bq4		Best Buy Quantity 4
bq5		Best Buy Quantity 5
bp1		Best Buy Price 1
bp2		Best Buy Price 2
bp3		Best Buy Price 3

bp4		Best Buy Price 4
bp5		Best Buy Price 5
bo1		Best Buy Orders 1
bo2		Best Buy Orders 2
bo3		Best Buy Orders 3
bo4		Best Buy Orders 4
bo5		Best Buy Orders 5
sq1		Best Sell Quantity 1
sq2		Best Sell Quantity 2
sq3		Best Sell Quantity 3
sq4		Best Sell Quantity 4
sq5		Best Sell Quantity 5
sp1		Best Sell Price 1
sp2		Best Sell Price 2
sp3		Best Sell Price 3
sp4		Best Sell Price 4
sp5		Best Sell Price 5

so1		Best Sell Orders 1
so2		Best Sell Orders 2
so3		Best Sell Orders 3
so4		Best Sell Orders 4
so5		Best Sell Orders 5
lc		Lower Circuit Limit
uc		Upper Circuit Limit
52h		52 week high low in other exchanges, Life time high low in mcx
52l		52 week high low in other exchanges, Life time high low in mcx

Sample Message :

```
{
  "t": "df",
  "e": "NSE",
  "tk": "22",
  "o": "1166.00",
  "h": "1179.00",
  "l": "1145.35",
  "c": "1152.65",
  "ap": "1159.74",
  "v": "819881",
  "tbq": "120952",
  "tsq": "131730",
  "bp1": "1156.00",
  "sp1": "1156.50",
  "bp2": "1155.80",
  "sp2": "1156.55",
  "bp3": "1155.75",
  "sp3": "1156.65",
```

```

    "bp4": "1155.70",
    "sp4": "1156.70",
    "bp5": "1155.65",
    "sp5": "1156.75",
    "bq1": "4",
    "sq1": "10",
    "bq2": "67",
    "sq2": "63",
    "bq3": "83",
    "sq3": "1",
    "bq4": "139",
    "sq4": "53",
    "bq5": "393",
    "sq5": "94"
  }

```

Unsubscribe Depth

Request:

Json Fields	Possible value	Description
t	ud	'ud' represents Unsubscribe depth
k		One or more scriplist for unsubscription. Example NSE 22#BSE 508123

Response :

Json Fields	Possible value	Description
t	udk	'udk' represents unsubscribe depth acknowledgement
k		One or more scriplist for unsubscription. Example NSE 22#BSE 508123

Subscribe Order Update

Request :

Json Fields	Possible value	Description
t	o	'o' represents order update subscription task
actid		Account id based on which order updated to be sent.

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

Unsubscribe Order Update

Request:

Json Fields	Possible value	Description
t	uo	'uo' represents Unsubscribe Order update

Response :

Json Fields	Possible value	Description
t	uok	'uok' represents Unsubscribe Order update acknowledgement

Addendum 1

Please find the logic for trading symbol creation:

NSE

- Exchange file name: **nnf_security**
- Delimiter: |
- Trading Symbol: **Column2-Column3**

NFO / CDS

- Exchange file name: **contract / cd_contract**
- Delimiter: |
- 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* CE
 - **Note:**
 - *SymbolName* - Column4
 - *ExpDate* - (Column7 + **315532800**) to be converted from long to date format
 - *OptionType* - Column9
 - *InstrumentName* - Column3
 - *Multiplier* - Column52
 - *DecimalPrecision* - Column10
 - *StrikePrice* - Column8 needs to be divided by ($Multiplier + 10^{DecimalPrecision}$)

BSE

- Exchange file name: **SCRIP.TXT**
- Delimiter: |
- Trading Symbol: **Column3**

Addendum 2

NSE Files are available in FTP provided by NSE .

FTP Path is <ftp://ftp.connect2nse.com/>

Equity FTP Login Name FTPGUEST and same as PWD and folder name is COMMON

FNO FTP Login Name FAOGUEST and same as PWD and folder name is FAOCOMMON

CDS FTP Login Name CDSGUEST and same as PWD and folder name is CDSCOMMON

S r N o	FTP Login Name and Password	FTP Path	File Name	Time to download	Purpose
1	Login Name FTPGUEST and same as PWD	ftp://ftp.connect2nse.com/common	Red_Flag_List_09012020082539.csv	File Name contain Trading Date. 0830 AM onwards every 5 min till found for current date	This file contain list of scrip which are likely to go in breach list in near future for NRE Client for buy. No restriction of sale.
2	Login Name FTPGUEST and same as PWD	ftp://ftp.connect2nse.com/common	Breach_List_09012020082539.csv	On Trading Day 0830 am onwards every 5 min. File Name contain Trading Date	This file contain list of scrip in which no further buy is permitted to NRE. No restriction on Sale.
3	Login Name FTPGUEST and same as PWD	ftp://ftp.connect2nse.com/common	CB_Bhavcopy08012020.CSV	File Name contain Trading date, which has passed. 0615 PM onwards every 60 MIN till found for Next Trading Date .	This file contain Closing price of listed Bonds along with trade date. This is Bhav Copy for Bonds
4	Login Name FTPGUEST and same as PWD	ftp://ftp.connect2nse.com/common/Bhavcopy	nupr080120.zip	File Name is ddmmyy.ZIP. Download from 0530PM onwards for trade date and prefixed by nupr	This file contain various prices related to today is equity market session. Closing price of Bonds , SME , Equity, Board Meeting , Book Closure, Corp Announcement etc. Part of the information can be use in displaying Detailed Scrip info for particular scrip in Trading Application
5	Login Name FTPGUEST and same as PWD	ftp://ftp.connect2nse.com/common/Bhavcopy	fo08042021.zip	File Name is ddmmyy.ZIP. Download from 0530PM onwards for trade date and prefixed by fao	File Contain Closing prices for Futures and Options Segment for each contract traded on the date of file
6	Login Name FTPGUEST and same as PWD	ftp://ftp.connect2nse.com/common/ntneat	security.gz	File Name is security.gz . Download from 0530PM onwards for trade date . Time and date must be greater than Trading date and 5 PM IST. E.g If downloading on Friday for Trading date of Next Monday than date and time of file creation must be greater than friday's date and time 0500 PM IST	This is Scrip Master for NSE Equity Segment and Underlying Scrip master for Equity Futures and Options Segment. File has very rich data and lot of decision support data can be extracted. file do we have next Trading day's Upper and Lower Circuit also Called Daily Price Range (DPR). File also have indication which scrip allowed in pre open , which scrip are Trade to Trade and which Scrip are ASM and GSM.
7	Login Name FTPGUEST and same as PWD	ftp://ftp.connect2nse.com/common/ntneat	contract.gz	File Name is contract.gz . Download from 0530PM onwards for trade date . Time and date must be greater than Trading date and 6 PM IST. E.g If downloading on Friday for Trading date of Next Monday than date and time of file creation must be greater than friday's date and time 0500 PM IST	This is Scrip Master for NSE Equity Futures and Option Segment and also called FNO Contract Master .