

Updated \* -- Mandatory

## Introduction

Pi is a collection of REST APIs that provides many required capabilities to build a modern stock market investment and trading platform. Using these API endpoints, you can execute orders in real time (equities, commodities, currency), stream live market data over WebSockets, and more.

Base URL

<https://piconnect.flattrade.in/PiConnectTP>

To use these APIs, you need to register your App with us to generate your apiKey and apiSecret.

For registering your app follow below step,

1. Login to Wall <https://wall.flattrade.in>

2. Navigate to Pi in top menu bar and click on "CREATE NEW API KEY"



3. Use the following values to enter the form,

**App Name** Your App Name

**App ShortName** Short Name of your APP

**Redirect URL** URL to which we need to redirect after successful login authentication.  
Note: Code to generate the token will be sent as parameter to this URL

**Postback URL** URL to which you will be receiving order updates for the orders placed through API.

**Description** Short description about your app

## Create a new API Key

Enter your application details where you will be using the API Key.

Please ensure to provide correct redirect url without any space. We will be sending the validation today to this redirect url after the sucessful login with code and client parameters. Please refer our [documentation](#) for more details.

App name	App ShortName
Redirect URL	
Postback URL	0 / 100
Description	
<input type="checkbox"/> I read and agree to the <a href="#">terms</a>	
<b>CREATE</b>	<b>CANCEL</b>

4. Your API key is now generated

My API Keys						<a href="#">CREATE NEW API KEY</a>
Status	AppName	RedirectURL	ApiKey	SecretKey	Regenerate Key	
<span>Active</span>	YOUAPP	<a href="https://yourapp.com/redirect?">https://yourapp.com/redirect?</a>	e110d599999999e913a8f354			

## 5. Copy the APIKEY

6. Click the eye icon  to view the secret key.

## Copy secret key

My API Keys					<a href="#">CREATE NEW API KEY</a>
Status	AppName	RedirectUrl	ApiKey	SecretKey	Regenerate Key
<span>Active</span>	YOUAPP	<a href="https://yourapp.com/redirect">https://yourapp.com/redirect</a>	e110d69999999999e913a8f354	2022_49999999999999999999999999999999	<a href="#">Regenerate</a>

**Don't have an account? [Sign up](#) now!**

## LOGIN FLOW

APIKey and APISecert is used to generate a access token (JKey) that will be used in all our APIs to perform trading.

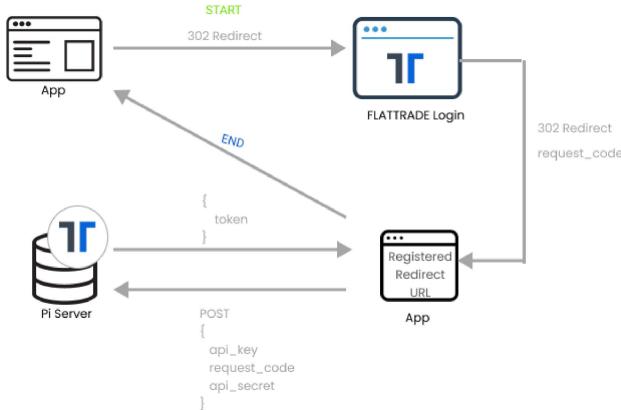
The access token generation process starts with an Authentication process initiated from a web browser. You need to authorize the program using your client ID (UCC), trading password and PAN/year of birth.

Whether your program runs on a GUI or a console, you always have to access the web browser to create an access token which then allows you to use the API.

The access token has a validity of 24 hours so you only have to generate access token once during the day.

(Note: Access tokens are cleared between 5 to 6 AM in the morning. We recommend you regenerate your access token after 6 AM).

Once you generate your access token, you can store it and bypass authentication for subsequent connects.



## Token Generation Steps

Step	Description
1	Open the Authorization URL <code>https://auth.flattrade.in/?app_key=APIKEY</code> browser Note: replace APIKEY text in this URL with the / allocated for you in the <u>above step 4</u>
2	Enter your Client id (UCC), Password, PAN/DOE submit

- 1 Open the Authorization URL  
`https://auth.flattrade.in/?app_key=APIKEY`  
 browser  
 Note: replace APIKEY text in this URL with the / allocated for you in the above step 4
- 2 Enter your Client id (UCC), Password, PAN/DOE submit

Step	Description
3	After you are authorized in our authentication process, the authorization screen will redirect to your URL with request_token ( <a href="https://yourRedirectURL.com/?request_code=requestCodeValue">https://yourRedirectURL.com/?request_code=requestCodeValue</a> )  <u>Note:</u> Redirect URL is pre-registered with us against each API Key. If you have different redirect URLs for PROD and TESTING, then each environment should have a different API Key registered with us.  <u>request_code:</u> It is an one-time code obtained during the login flow. This code's lifetime is only a few minutes, so it is meant to be exchanged for a token immediately after obtaining.
4	Call to <a href="https://authapi.flattrade.in/trade/api/authenticate">https://authapi.flattrade.in/trade/api/authenticate</a> in POST method to validate request_code and get the token. { "api_key":"xcvvwegfhgh4454646", "request_code":"xxdfddfdfdsfsf84okkdlfelfdfd", "api_secret":"sdfdsfsdfdsfXXXXXXXX" }  <b>NOTE:</b> Please refer <a href="#">this document</a> for a detailed walkthrough on how to handle the security key parameter to get the token.  <u>api_key:</u> The public API key <u>request_code:</u> It is an one-time code obtained during the login flow. This code's lifetime is only a few minutes, so it is meant to be exchanged for a token immediately after obtaining <u>api_secret:</u> SHA-256 hash of (api_key + request_code + api_secret)
5	You will get a response with following values. These can be used in appropriate end points to get more details of the user.  <u>On Success:</u> { "token":"dsfdsf84okkdlfelffdfd3454545454ss", "client":"CCODE123", "status":"Ok", "emsg":"" }

## POSTMAN COLLECTIONS

To Download the Postman Collections [!\[\]\(bd1a142de767a21e5362c595f844a4ff\_img.jpg\)](#)

To test the API in Postman, you need to define the required variable fields such as the base URL, clientid, jkey on the API specification.

Sample:  
 BaseUrl -  
<https://piconnect.flattrade.in/PiConnectTP>  
 ClientId - FT0000  
 jKey - GHUDWU53H32MTHPA536Q32WR

## ORDER AND TRADES

### PLACE ORDERS

To get place order you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/PlaceOrder>

#### QUERY PARAMETERS

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

```
# Here is a curl example !\[\]\(bd3b31712ad9bab5a241210fa6925cdd\_img.jpg\)
curl --location 'https://BaseURL/PlaceOrder' \
\ 
--header 'Content-Type: application/json' \
--data 'jData={
  "uid": "FZ00000",
  "actid": "FZ00000",
  "exch": "NSE",
  "tsym": "ACC-EQ",
  "qty": "50",
  "prc": "1400",
  "prd": "H",
  "trantype": "B",
  "prctyp": "LMT",
  "ret": "DAY"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Json Fields	Possible value	Description
uid*		Logged in User Id
actid*		Login users account ID
exch*	NSE / NFO /	Exchange (Select from 'exarr' Array

Possible value			Description
Json Fields			
	BSE / MCX		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 [If qty is junk value other than numbers].
prc*			Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0'].
trgprc			Only to be sent in case of SL / SL-M order.
dscqty*			Disclosed quantity (Max 10% for NSE, and 50% for MCX) [If dscqty is junk value other than numbers].
prd*	C - CNC/ M - NRML / H - CO / B - BO / I - MIS / F - MTF		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)

Json Fields	Possible value	Description
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
prctyp*	LMT / MKT / SL-LMT / SL-MKT	
ret*	DAY / EOS / IOC	Retention type [ret should be DAY / EOS / IOC else reject]
remarks		Any tag by user to mark order.
ordersource	API	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 )

Json Fields	Possible value	Description
amo*	Yes	The message "Invalid AMO" will be displayed if the "amo" field is not sent with a "Yes" value. If amo is not required, do not send this field.
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

Json Fields	Possible value	Description
		price type 3L
prc3		Price for third leg, mandatory for price type 3L
mkt_protection		market protection value in percentage

## 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

## MODIFY ORDER

To get Modify order you need to make a POST call  
to the following url :

<https://piconnect.flattrade.in/PiConnectTP/ModifyOrder>

Sample Success Response:

```
{
  "request_time": "10:48:03 20-05-2020",
  "stat": "Ok",
  "norenordno": "2005200000017"
}
```

Sample Error Response :

```
{
  "stat": "Not_Ok",
  "request_time": "20:40:01 19-05-2020",
  "emsg": "Error Occurred : 2 \"invalid  
input\""
}
```

## QUERY PARAMETERS

# Here is a curl example 

```
curl --location 'https://BaseURL/ModifyOrder'
\--header 'Content-Type: application/json' \
```

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

```
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "qty": "50",
    "prc": "1400",
    "prctyp": "LMT",
    "ret": "DAY",
    "norenordno": "123456789"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

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 [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0'].
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 [If qty is junk value other than numbers].
tsym*		Unque id of contract on which order was

Possible		
Json Fields	value	Description
		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 / EOS / IOC	Retention type [ret should be DAY / EOS / IOC else reject]
trgprc		New trigger price in case of SL-MKT or SL-LMT
uid*		User id of the logged in user.
bpprc		Book Profit Price applicable only if product is selected as B (Bracket order )
blprc		Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order )
trailprc		Trailing Price applicable only if product is selected as H and B (High Leverage and Bracket order )

## RESPONSE DETAILS

Response data will be in json format with below fields

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",
```

Json Fields	Possible value	Description	"emsg": "Rejected : ORA:Order not found" }
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	

## CANCEL ORDER

To get Cancel Order you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/CancelOrder>

# Here is a curl example 

```
curl --location 'https://BaseURL/CancelOrder'  
\  
--header 'Content-Type: application/json' \  
--data 'jData={  
    "uid": "FZ00000",  
    "norenordno": "123456789"  
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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 cancellation 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

To get Exit SNO Order you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/ExitSNOOrder>

# Here is a curl example 

```
curl --location
'https://BaseURL/ExitSNOOrder' \
--header 'Content-Type: application/json' \
--data 'jData={
  "uid": "FZ00000",
  "prd": "H",
  "norenordno": "123456789"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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

Possible		
Json Fields	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

Possible		
Json Fields	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

To get Order Margin you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/GetOrderMargin>

# Here is a curl example 

```
curl --location
'https://BaseURL/GetOrderMargin' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
```

## QUERY PARAMETERS

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

```

    "actid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "qty": "50",
    "prc": "1400",
    "prd": "H",
    "trantype": "B",
    "prctyp": "LMT",
    "norenordno": "123456789"
}&jKey=GHUDWU53H32MTHPA536Q32WR '

```

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 [If qty is junk value other than numbers].
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].
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

Json Fields	Possible value	Description
		product display name, for user to select, and send corresponding prd in API call)
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
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

## BASKET MARGIN

To get Basket Margin you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/GetBasketMargin>

# Here is a curl example 

```
curl --location
'https://BaseURL/GetBasketMargin' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "qty": "50",
    "prc": "1400",
    "prd": "H",
    "trantype": "B",
    "prctyp": "LMT",
    "norenordno": "123456789"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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 [If qty is junk value other than numbers].
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].
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)
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
prctyp*	LMT / MKT /	

Json Fields	Possible value	Description
	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
basketlists		Optional field. Array of json objects. (object fields given in below table)

## Json Fields

of object in  
values Array

Possible  
value

Description

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 [If qty is junk value other than numbers].
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL- MKT' with price '0' ].
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)
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
prctyp*	LMT / MKT / SL-LMT	

Json Fields of object in values Array	Possible value	Description
	/ SL- MKT	

introp_key	Optional field.
introp_exch	Optional field.

## 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 contain rejection reason.
marginused		Total margin used.
marginusedtrade		Margin used after trade.
emsg		This will be present only if Order placement fails

## ORDER BOOK

To get Order Book you need to make a POST call to the following url :

# Here is a curl example 

```
curl --location 'https://BaseURL/OrderBook' \
--header 'Content-Type: application/json' \
```

<https://piconnect.flattrade.in/PiConnect>

TP/OrderBook

```
--data 'jData={
    "uid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

Parameter	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

## 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 [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].

Sample Success Output :

Success response :

```
[
{
  "stat" : "Ok",
  "exch" : "NSE",
  "tsym" : "ACC-EQ",
  "norenordno" : "2006250000001223",
  "prc" : "127230",
  "qty" : "100",
  "prd" : "C",
  "status": "Open",
  "trantype" : "B",
  "prctyp" : "LMT",
  "fillshares" : "0",
  "avgprc" : "0",
  "exchordid" : "25062000000343421",
  "uid" : "VIDYA",
  "actid" : "CLIENT1",
  "ret" : "DAY",
  "amo" : "Yes"
},
{
  "stat" : "Ok",
  "exch" : "NSE",
  "tsym" : "ABB-EQ",
  "norenordno" : "2006250000002543",
  "prc" : "127830",
  "qty" : "50",
}
```

Json Fields	Possible value	Description	
qty*		Order Quantity [If qty is junk value other than numbers].	<pre> "prd" : "C", "status": "REJECT", "transtype" : "B", "prctyp" : "LMT", "fillshares" : "0", "avgprc" : "0", "rejreason" : "Insufficient funds" "uid" : "VIDYA", "actid" : "CLIENT1", "ret" : "DAY", "amo" : "No" } ] </pre>
prd		Display product alias name, using prarr returned in user details.	
status		Order status	
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].	
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 [If dscqty is junk value other than numbers].	
trgprc		Order trigger price	
ret*	DAY / EOS / IOC	Order validity [ret should be DAY / EOS / IOC else reject]	
uid			
actid			

Json Fields	Possible value	Description
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	The message "Invalid AMO" will be displayed if the "amo" field is not sent with a "Yes" value. If amo is not required, do not send this field.
pp		Price precision
ti		Tick size
ls		Lot size
token		Contract Token
orddttm		
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.
dname		Broker specific contract display name.

Json Fields	Possible value	Description
rorgqty	To be used in get margin from modify window.	
rorgprc	To be used in get margin from modify window.	
orgtrgprc	To be used in get margin from modify window, for H/B products only	
orgblprc	To be used in get margin from modify window, for H/B products only	
algo_name	Algo Name	
C	CUST_FIRM_C	

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

## MULTI LEG ORDER BOOK

To get Multi Leg Order Book you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/MultiLegOrderBook>

# Here is a curl example 

```
curl --location
'https://BaseURL/MultiLegOrderBook' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000"
```

## QUERY PARAMETERS

Parameter	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

## 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 [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].
qty*		Order Quantity [If qty is junk value other than numbers].
prd		Display product alias name, using prarr

Json Fields	Possible value	Description
		returned in user details.
status		Order status
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
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 [If dscqty is junk value other than numbers].
trgprc		Order trigger price
ret*	DAY / EOS / IOC	Order validity [ret should be DAY / EOS / IOC else reject]
uid		
actid		
bpprc		Book Profit Price applicable only if product is selected as B (Bracket order )
blprc		Book loss Price applicable only if

Json Fields	Possible value	Description
		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	The message "Invalid AMO" will be displayed if the "amo" field is not sent with a "Yes" value. If amo is not required, do not send this field.
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

Json Fields	Possible value	Description
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

To get Single Order History you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/SingleOrdHist>

# Here is a curl example 

```
curl --location
'https://BaseURL/SingleOrdHist' \
--header 'Content-Type: application/json' \
--data 'jData=
{"uid":"FZ00000"}&jKey=GHUDWU53H32MTHPA536Q32
WR'
```

## QUERY PARAMETERS

Parameter	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

## 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 [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].
qty*		Order Quantity [If qty is junk value other than numbers].
prd		Display product alias name, using prarr

Sample Success Output :

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

Json Fields	Possible value	Description	
		returned in user details.	
status		Order status	
rpt		Report Type (fill/complete etc)	
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].	
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 [If dscqty is junk value other than numbers].	
trgprc		Order trigger price	
ret*	DAY / EOS / IOC	Order validity [ret should be DAY / EOS / IOC else reject]	
uid			
actid			
bpprc		Book Profit Price applicable only if product is selected as B (Bracket order )	

```

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

```

Json Fields	Possible value	Description	
blprc		Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order )	<pre> "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" } ]</pre>
trailprc		Trailing Price applicable only if product is selected as H and B (High Leverage and Bracket order )	
amo*	Yes	The message "Invalid AMO" will be displayed if the "amo" field is not sent with a "Yes" value. If amo is not required, do not send this field.	
pp		Price precision	
ti		Tick size	
ls		Lot size	
token		Contract Token	
orddttm			
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

## TRADE BOOK

To get Trade Book you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/TradeBook>

```
# Here is a curl example □
curl --location 'https://BaseURL/TradeBook' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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

## 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.

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"}, {"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"}]
```

Json Fields	Possible value	Description	
norenordno		Noren Order Number	"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" }, {
qty*		Order Quantity [If qty is junk value other than numbers].	"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", "transtype": "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"
prd		Display product alias name, using prarr returned in user details.	}
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].	]
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 / EOS / IOC	Order validity [ret should be DAY / EOS / IOC else reject]	
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	

Possible		
Json Fields	value	Description
ordersource		Order Source
token		Token

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

Possible		
Json Fields	value	Description
stat	Not_Ok	Order book failure indication.
request_time		Response received time.
emsg		Error message

## POSITIONS BOOK

To get Positions Book you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/PositionBook>

```
# Here is a curl example 
curl --location
'https://BaseUrl/PositionBook' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
}'&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

Possible		
Name	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

## 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	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

Sample Success Response :

```
[
{
  "stat": "Ok",
  "uid": "POORNA",
  "actid": "POORNA",
  "exch": "NSE",
  "tsym": "ACC-EQ",
  "prarr": "C",
  "pp": "2",
  "ls": "1",
  "ti": "5.00",
  "mult": "1",
  "prcftr": "1.000000",
  "daybuyqty": "2",
  "daysellqty": "2",
  "daybuyamt": "2610.00",
  "daybuyavgprc": "1305.00",
  "daysellamt": "2610.00",
  "daysellavgprc": "1305.00",
  "cfbuyqty": "0",
  "cfsellqty": "0",
  "cfbuyamt": "0.00",
  "cfbuyavgprc": "0.00",
  "cfsellamt": "0.00",
  "cfsellavgprc": "0.00",
  "openbuyqty": "0",
  "opensellqty": "23",
  "openbuyamt": "0.00",
  "openbuyavgprc": "0.00",
  "opensellamt": "30015.00",
  "opensellavgprc": "1305.00",
  "netqty": "0",
  "netavgprc": "0.00",
  "lp": "0.00",
  "urmtom": "0.00",
  "rpn1": "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\\\""
}
```

}

Json Fields	Possible value	Description
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	
totsellavgprc	Total Sell Avg Price	
lp	LTP	
rpnl	RealizedPNL	
urmtom	UnrealizedMTOM. (Can be recalculated in LTP update : = netqty * (lp from web socket - netavgprc) * prcftr	
bep	Break even price	
openbuyqty		
opensellqty		
openbuyamt		
opensellamt		
openbuyavgprc		
opensellavgprc		
mult		
pp		
prcftr	gn*pn/(gd*pd).	
ti	Tick size	

Possible		
Json Fields	value	Description
ls		Lot size
instname		Instrument Name
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:

Possible		
Json Fields	value	Description
stat	Not_Ok	Position book request failure indication.
request_time		Response received time.
emsg		Error message

## PRODUCT CONVERSION

To get Product Conversion you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/ProductConversion>

```
# Here is a curl example □
curl --location
'https://BaseURL/ProductConversion' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "qty": "50",
    "prc": "1400",
    "prd": "H",
    "trantype": "B",
    "prctyp": "LMT",
    "prevprd": "C",
    "postype": "Day"
}'
```

## QUERY PARAMETERS

Possible		
Name	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 [If qty is junk value other than numbers].	
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.	
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
postype*	Day / CF	Converting Day or Carry forward position
ordersource	API	For Logging

## RESPONSE DETAILS

Response data will be in json format with below fields.

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"
```

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.

## PLACE GTT ORDER

To Place GTT Order Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/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.

# Here is a curl example 

```
curl --location
'https://BaseURL/PlaceGTTOrder' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "validity": "DAY",
    "qty": "50",
    "prc": "1400",
    "prd": "H",
    "transtype": "B",
    "prctyp": "LMT",
    "prevprd": "C",
    "ret": "DAY",
    "dscqty": "10"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

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

Sample Failure Response :

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

Json Fields	Possible value	Description
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
prctyp*	LMT / MKT / SL-LMT / SL-MKT / DS / 2L / 3L	
prd*	C / M / H	Product name
ret*	DAY / EOS / IOC	Retention type [ret should be DAY / EOS / IOC else reject]
actid*		Login users account ID
qty*		Order Quantity [If qty is junk value other than numbers].
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].
dscqty*		Disclosed quantity (Max 10% for NSE, and 50% for MCX) [If dscqty is junk value other than numbers].

## 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.

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

## MODIFY GTT ORDER

To Modify GTT Order Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/ModifyGTTOrder>

Json Fields	Possible value	Description
uid*		User id of the logged in user
tsym*		Trading symbol
exch*		Exchange Segment
ai_t*		Alert Type, should be original alert type, can't be modified
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 [transtype should be 'B' or 'S' else reject].
prctyp*	LMT / MKT / SL-LMT /	

```
# Here is a curl example 
curl --location
'https://BaseURL/ModifyGTTOrder' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "validity": "DAY",
    "qty": "50",
    "prc": "1400",
    "prd": "H",
    "trantype": "B",
    "prctyp": "LMT",
    "prevprd": "C",
    "ret": "DAY",
    "dscqty": "10"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

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

Sample Failure Response :

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

}

Json Fields	Possible value	Description
	SL-MKT / DS / 2L / 3L	
prd*	C / M / H	Product name
ret*	DAY / EOS / IOC	Retention type [ret should be DAY / EOS / IOC else reject]
actid*		Login users account ID
qty*		Order Quantity [If qty is junk value other than numbers].
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].
dscqty*		Disclosed quantity (Max 10% for NSE, and 50% for MCX) [If dscqty is junk value other than numbers].

## 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

Json Fields	Possible value	Description
Input		
	2) Session Expired	

## CANCEL GTT ORDER

To Cancel GTT Order Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/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.

```
# Here is a curl example Copy
curl --location
'https://BaseURL/CancelGTTOrder' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "al_id": "21041500000013"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

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

Sample Failure Response :

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

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

## GET PENDING GTT ORDER

To Get Pending GTT Order Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/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.

```
# Here is a curl example □
curl --location
'https://BaseURL/GetPendingGTTOrder' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

```
[
{
    "stat": "Ok",
    "ai_t": "LTP_A",
    "Al_id": "21041500000002",
    "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": "API",
    "d": "1900.00"
}
```

Sample Failure Response :

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

Json Fields	Possible value	Description	"emsg": "Session Expired : Invalid Session Key"
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	
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].	
prctyp	LMT / MKT / SL-LMT / SL-MKT / DS / 2L / 3L		
prd	C / M / H	Product name	
ret*	DAY / EOS / IOC	Retention type [ret should be DAY / EOS / IOC else reject]	
actid		Login users account ID	
qty*		Order Quantity [If qty is junk value other than numbers].	
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].	
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired	

## GET ENABLED GTTS

To Get Enabled GTTs Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/GetEnabledGTTS>

### REQUEST DETAILS

Parameter	Possible
Name	Description

jData*	Should send json object with fields in below list
--------	---

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

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

### RESPONSE DETAILS

Response data will have below fields.

Json	Possible
Fields	Description

stat	GTT order success or failure indication.
------	--

request_time	This will be present only in a successful response.
--------------	---

ai_ts	Array of alert types
-------	----------------------

```
# Here is a curl example Copy
curl --location
'https://BaseUrl/GetEnabledGTTS' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

```
{
  "stat": "Ok",
  "request_time": "04062021121503",
  "ai_ts": [
    {"ai_t": "ATP"},
    {"ai_t": "LTP"}
  ]
}
```

Sample Failure Response :

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

## PLACE OCO ORDER

To Place OCO Order Request you need to make a

POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/PlaceOCOOrder>

## 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*		Unique id of contract on which order to be placed. (use url encoding to avoid special char error for symbols like M&M)
exch*		Exchange
validity*	DAY or GTT	Validity
ai_t*		Alert type
exchsym		Exchange symbol
oivariable		Array Object, details given below.
place_order_params*		List of place order Params fields.
place_order_params_leg2*		List of Place order params fields for leg2.

Input Sample:

```
jData=
{"uid": "FZ00000", "ai_t": "LMT_BOS_O", "remarks": "admin", "validity": "GTT", "tsym": "ACC-EQ", "exch": "NSE", "oivariable": [{"d": "20000", "var_name": "x"}, {"d": "30000", "var_name": "y"}], "place_order_params": {"tsym": "ACC-EQ", "exch": "NSE", "trantype": "B", "prctyp": "MKT", "prd": "C", "ret": "DAY", "actid": "FZ00000", "uid": "FZ00000", "ordersource": "WEB", "qty": "1", "prc": "0"}, "place_order_params_leg2": {"tsym": "ACC-EQ", "exch": "NSE", "trantype": "S", "prctyp": "MKT", "prd": "C", "ret": "DAY", "actid": "FZ00000", "uid": "FZ00000", "ordersource": "WEB", "qty": "1", "prc": "0"}}&jKey=652c99c82d7edcd4f472869786074c90bd27df0c68635c2e53db0ed08cbea0f
```

## OIVARIABLE OBJ FORMAT

Json Fields	Possible value	Description
d*		Data to be compared with LTP
var_name*	x or y	Variable Name

## PLACE\_ORDER\_PARAMS OBJ FORMAT

Json Fields	Possible value	Description
tsym*		Trading symbol of the scrip (contract)
exch*		Exchange
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
prctyp*		Price Type
prd*		Product
ret*	DAY / EOS / IOC	Retention type [ret should be DAY / EOS / IOC else reject]
actid*		Acct Id
uid*		User Id
ordersource	MOB / WEB / TT	Used to generate exchange info fields.
remarks		Any tag by user to mark order.
qty*		Order Quantity [If qty is junk value other than numbers].
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].

Json Fields	Possible value	Description
trgprc		New trigger price in case of SL-MKT or SL-LMT

## RESPONSE DETAILS

Response data will have below fields.

Json Fields	Possible value	Description
stat		OCO orders 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": "18:56:26 08-10-2021",
  "stat": "OI created",
  "al_id": "21100800000009"
}
```

Sample Failure Response :

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

## MODIFY OCO ORDER

To Modify OCO Order Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/ModifyOCOOrder>

## REQUEST DETAILS

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

# Here is a curl example 

```
curl --location
'https://BaseURL/ModifyOCOOrder' \
--header 'Content-Type: application/json' \
--data 'jData={
  "uid": "FZ00000",
  "exch": "NSE",
  "tsym": "ACC-EQ",
  "validity": "DAY"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Json Fields	Possible value	Description
uid*		User id of the logged in user.
tsym*		Unique id of contract on which order to be placed. (use url encoding to avoid special char error for symbols like M&M)
exch*		Exchange
validity*	DAY or GTT	Validity
ai_t*		Alert type
al_id*		Alert id
exchsym		Exchange symbol
oivariable		Array Object, details given below.
place_order_params		Array Object, details given below.

## OIVARIABLE OBJ FORMAT

Json Fields	Possible value	Description
d*		Data to be compared with LTP
var_name*	x or y	Variable Name

## PLACE\_ORDER\_PARAMS OBJ FORMAT

Json Fields	Possible value	Description
tsym*		Trading symbol of the scrip (contract)
exch*		Exchange

Json Fields	Possible value	Description
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
prctyp*		Price Type
prd*		Product
ret*	DAY / EOS / IOC	Retention type [ret should be DAY / EOS / IOC else reject]
actid*		Acct Id
uid*		User Id
ordersource	MOB / WEB / TT	Used to generate exchange info fields.
remarks		Any tag by user to mark order.
qty*		Order Quantity [If qty is junk value other than numbers].
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].
trgprc		New trigger price in case of SL-MKT or SL-LMT

## RESPONSE DETAILS

Response data will have below fields.

Json Fields	Possible value	Description
stat		OCO order success or failure indication. i."stat":"OI replaced" - incase of success

Sample Success Response :

```
{
  "request_time": "11:14:52 11-10-2021",
  "stat": "OI replaced",
  "al_id": "21101100000001"
}
```

Sample Failure Response :

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

Json Fields	Possible value	Description
	ii."stat":"Invalid Oi" -	incase of failure
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	

## CANCEL OCO ORDER

To Cancel OCO Order Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/CancelOCOOrder>

### 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

```
# Here is a curl example Copy
curl --location
'https://BaseURL/CancelOCOOrder' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "al_id": "2108300000040"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :  
{

Response data will have below fields.

Json Fields	Possible value	Description
stat	OCO 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	

```
"request_time": "17:41:02 30-08-2021",
"stat": "Ok delete success"
,"al_id": "21083000000040"
}
```

Sample Failure Response :

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

## HOLDINGS AND LIMITS

### HOLDINGS

To get Holdings you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/Holdings>

### QUERY PARAMETERS

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

# Here is a curl example 

```
curl --location 'https://BaseURL/Holdings' \
--header 'Content-Type: application/json' \
--data 'jData={
"uid": "FZ00000",
"actid": "FZ00000",
"prd": "H"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

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

## RESPONSE DETAILS

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

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Holding request success or failure indication.
exch_tsym		Array of objects exch_tsym objects as defined below.
holdqty		Holding quantity
dpqty		DP Holding quantity
npoadqty		Non Poa display quantity
colqty		Collateral quantity
benqty		Beneficiary quantity
unplgdqty		Unpledged quantity
brkcolqty		Broker Collateral
btstqty		BTST quantity
btstcolqty		BTST Collateral quantity
usedqty		Holding used today
upldprc		Average price uploaded along with holdings

Notes:

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."
}
```

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

Salable: btstqty + holdqty + unplgdqty + benqty +  
 dpqty - usedqty

#### [Exch\\_tsym object:](#)

Json

Fields of  
object in

values	Possible value	Description
Array		
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

## LIMITS

To get Limits you need to make a POST call to the following url :  
<https://piconnect.flattrade.in/PiConnectTP/Limits>

```
# Here is a curl example □
curl --location 'https://BaseURL/Limits' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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.

## 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

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": "5000000000000.00",
  "pendordvallmt": "20000000000000.00",
  "turnover": "3915000.00",
  "pendordval": "2871000.00",
  "marginused": "3945540.00",
  "mtomcurper": "0.00",
  "urmtom": "30540.00",
  "grexpo": "3915000.00",
  "uzpn1_e_i": "15270.00",
  "uzpn1_e_m": "61080.00",
  "uzpn1_e_c": "-45810.00"
}
```

Sample Failure Response :

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

```

    Possible
    value      Description
}
  "emsg": "Server Timeout : "
}
```

---

**Json Fields**

---

-----**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**-----

---

Json Fields	Possible value	Description
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	
scripbskmar	Scrip basket margin	
addscripbskmrg	Additional scrip basket margin	
brokerage	Brokerage amount	
collateral	Collateral calculated based on uploaded holdings	
cash_coll	Cash Collateral	

Json Fields	Possible value	Description
grcoll		Valuation of uploaded holding pre haircut
<hr/> -----Additional Risk Limits-----		
<hr/>		
turnoverlmt		
pendordvallmt		
<hr/> -----Additional Risk Indicators-----		
<hr/>		
turnover		Turnover
pendordval		Pending Order value
<hr/> -----Margin used detailed breakup fields-----		
<hr/>		
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

Json Fields	Possible value	Description
		(FX Intraday)
rz.pnl_f_m	Current realized PNL (FX Margin)	
rz.pnl_c_i	Current realized PNL (Commodity Intraday)	
rz.pnl_c_m	Current realized PNL (Commodity Margin)	
uz.pnl_e_i	Current unrealized MTOM (Equity Intraday)	
uz.pnl_e_m	Current unrealized MTOM (Equity Margin)	
uz.pnl_e_c	Current unrealized MTOM (Equity Cash n Carry)	
uz.pnl_d_i	Current unrealized MTOM (Derivative Intraday)	
uz.pnl_d_m	Current unrealized MTOM (Derivative Margin)	
uz.pnl_f_i	Current unrealized MTOM (FX Intraday)	

Json Fields	Possible value	Description
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)	

Json Fields	Possible value	Description
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)	

Json Fields	Possible value	Description
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	

Json Fields	Possible value	Description
	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	

Json Fields	Possible value	Description
	(Commodity Intraday)	
addscripbkskmrg_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)	

Json Fields	Possible value	Description
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)	
mr_fx_u	MR fx used	
mr_sell	MR sell credit	
mr_t1sell	MR t1 sell credit	
mr_eqt_a	MR equity allocated	
mr_der_a	MR derivatives allocated	
mr_fx_a	MR fx allocated	
mr_com_a	MR commodity allocated	
request_time	request_time	
emsg	This will be present only in a failure response.	

## GET INDEX LIST

To get Index List you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/GetIndexList>

## QUERY PARAMETERS

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	Limits request success or failure indication.
values		Array Of Basket, Criteria pair.
request_time		This will be present only in a successful response.

```
# Here is a curl example □
curl --location
'https://BaseURL/GetIndexList' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "exch": "NSE"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

### 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"
    }
  ]
}
```

Possible		
Json Fields	Possible value	Description
emsg		This will be present only in case of errors.

## BASKET, CRITERIA PAIR OBJECT :

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

```

    "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

To get Top List Names you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/TopListName>

# Here is a curl example □

```

curl --location 'https://BaseURL/TopListName'
 \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "exch": "NSE"
}&jKey=GHUDWU53H32MTHPA536Q32WR'

```

## QUERY PARAMETERS

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
bskt		Basket name
crt		criteria

Sample Success Response :

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

Sample Failure Response :

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

}

## GET TOP LIST

To get Top List you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/TopList>

## QUERY PARAMETERS

Parameter	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*		bskt
crt*		criteria

## RESPONSE DETAILS

Response data will be in json format with below fields.

# Here is a curl example 

```
curl --location 'https://BaseURL/TopList' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "exch": "NSE",
    "tb": "T",
    "bskt": "NSEALL",
    "crt": "LTP"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

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",  
            "t": "2020-06-03T15:44:45.000Z"  
        }  
    ]  
}
```

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

```

    "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)

To get Time Price Data (Chart data) you need to make a POST call to the following url :

```
# Here is a curl example □
curl --location 'https://BaseURL/TPSeries' \
```

<https://piconnect.flattrade.in/PiConnect>  
TP/TPSeries

```
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "exch": "NSE",
    "token": "INFY",
    "st": "12315",
    "et": "4874564",
    "intrv": "1"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

Parameter	Possible value	Description
Name	value	
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)
intrv	1 / 3 / 5 / 10 / 15 / 30 / 60 / 120	chart intervals

## 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.

Sample Success Response :

```
[
{
    "stat": "Ok",
    "time": "02-06-2020 15:46:23",
    "into": "0.00",
    "inth": "0.00",
    "intl": "0.00",
    "intc": "0.00",
    "intvw": "0.00",
    "intv": "0",
    "into": "0",
    "v": "980515",
    "oi": "128702"
},
{
    "stat": "Ok",
```

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	

```

    "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 EOD CHART DATA

To get EOD chart data you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/EODChartData>

## QUERY PARAMETERS

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
sym*		Symbol name
from*		From date
to*		To date

## RESPONSE DETAILS

Response data will be in json format with below fields.

Json Fields	Possible value	Description
time	DD/MM/CCYY hh:mm:ss	

```

# Here is a curl example □
curl --location
'https://BaseURL/EODChartData' \
--header 'Content-Type: application/json' \
--data 'jData={
  "sym": "NSE:RELIANCE-EQ",
  "from": "1624838400",
  "to":
  "1663718400"}&jKey=GHUDWU53H32MTHPA536Q32WR'

```

Sample Success Response :

```

[
  {
    "time": "21-SEP-2022",
    "into": "2496.75",
    "inth": "2533.00",
    "intl": "2495.00",
    "intc": "2509.75",
    "ssboe": "1663718400",
    "intv": "4249172.00"
  },
  {
    "time": "15-SEP-2022",
    "into": "2495.00",
    "inth": "2509.75",
    "intl": "2495.00",
    "intc": "2533.00",
    "ssboe": "1663718400",
    "intv": "4249172.00"
  }
]

```

Json Fields	Possible value	Description
into	Interval open	
inth	Interval high	
intl	Interval low	
intc	Interval close	
ssboe	Date,Seconds in 1970 format	
intv	Interval volume	

```

    "into": "2583.00",
    "inth": "2603.55",
    "intl": "2556.75",
    "intc": "2562.70",
    "ssboe": "1663200000",
    "intv": "4783723.00"
  },
  {
    "time": "28-JUN-2021",
    "into": "2122.00",
    "inth": "2126.50",
    "intl": "2081.00",
    "intc": "2086.00",
    "ssboe": "1624838400",
    "intv": "9357852.00"
  }
]

```

## GET OPTION CHAIN

To get Option Chain you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/GetOptionChain>

## QUERY PARAMETERS

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

```

# Here is a curl example □

curl --location
'https://BaseURL/GetOptionChain' \
--header 'Content-Type: application/json' \
--data 'jData={
  "uid": "FZ00000",
  "exch": "NSE",
  "tsym": "ACC-EQ",
  "strprc": "",
  "cnt": ""
}&jKey=GHUDWU53H32MTHPA536Q32WR'

```

Json Fields	Possible value	Description
uid*	Logged in User Id	
tsym*	Trading symbol of any of the option or future. Option	

Json Fields	Possible value	Description
		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	Possible value	Description
Array		

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

## GET OPTION GREEK

To get Option greek you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/GetOptionGreek>

## QUERY PARAMETERS

Parameter	Possible value	Description
Name		

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

# Here is a curl example 

```
curl --location
'https://BaseURL/GetOptionGreek' \
--header 'Content-Type: application/json' \
--data 'jData={
    "exd": "",
    "strprc": "",
    "sptprc": "",
    "int_rate": "",
    "volatility": "",
    "optt": ""
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Json Fields	Possible value	Description
exd		Expiry Date
strprc		Strike Price
sptprc		Spot Price
int_rate		Init Rate
volatility		Volatility
optt		Option Type

## RESPONSE DETAILS

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat		Success or failure indication.
request_time		This will be present only in a successful response.
cal_price		Cal Price
put_price		Put Price
cal_delta		Cal Delta
put_delta		Put Delta
cal_gamma		Cal Gamma
put_gamma		Put Gamma
cal_theta		Cal Theta
put_theta		Put Theta
cal_rho		Cal Rho
put_rho		Put Rho
cal_vego		Cal Vego
put_vego		Put Vego

Sample Success Response :

```
{
  "request_time": "17:22:58 28-07-2021",
  "stat": "OK",
  "cal_price": "1441",
  "put_price": "0.417071",
  "cal_delta": "0.997304",
  "put_delta": "-0.002696",
  "cal_gamma": "0.000001",
  "put_gamma": "0.000001",
  "cal_theta": "-31.535015",
  "put_theta": "-31.401346",
  "cal_rho": "0.000119",
  "put_rho": "-0.016590",
  "cal_vego": "0.006307",
  "put_vego": "0.006307"
}
```

Sample Failure Response :

```
{
  "stat": "Not_Ok",
  "emsg": "Invalid Input : jData is Missing."
}
```

## EXCH MSG

To get Exch Msg you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/ExchMsg>

```
# Here is a curl example □
curl --location 'https://BaseURL/ExchMsg' \
--header 'Content-Type: application/json' \
--data 'jData=
{"uid":"FZ00000","exch":"NSE"}&jKey=GHUDWU53H
32MTHPA536Q32WR'
```

## QUERY PARAMETERS

Parameter	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)

## 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

To get Broker Msg you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/GetBrokerMsg>

```
# Here is a curl example □
curl --location
'https://BaseURL/GetBrokerMsg' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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.

Sample Success Response :

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

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

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

## SPAN CALCULATOR

To get Span Calculator you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/SpanCalc>

```
# Here is a curl example □
curl --location 'https://BaseURL/SpanCalc' \
--header 'Content-Type: application/json' \
--data 'jData={
    "actid": "FZ00000",
    "pos": ""
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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

Possible		
Json Fields	value	Description
		buyqty, sellqty

## ALERTS

### SET ALERT

To Set Alert Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/SetAlert>

#### REQUEST DETAILS

Parameter	Possible	
Name	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.

# Here is a curl example [Copy](#)

```
curl --location 'https://BaseURL/SetAlert' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "ai_t": "",
    "validity": "DAY",
    "remarks": ""
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

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

Sample Failure Response :

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

## 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

## CANCEL ALERT

To Cancel Alert Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/CancelAlert>

## REQUEST DETAILS

Parameter		
Name	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

# Here is a curl example 

```
curl --location 'https://BaseURL/CancelAlert' \
--header 'Content-Type: application/json' \
--data 'jData={ \
    "uid": "FZ00000", \
    "actid": "FZ00000" \
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

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

Sample Failure Response :

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

## 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

## MODIFY ALERT

To Modify Alert Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/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

# Here is a curl example 

```
curl --location 'https://BaseURL/ModifyAlert' \
--header 'Content-Type: application/json' \
--data 'jData={ \
    "uid": "FZ00000", \
    "actid": "FZ00000", \
    "exch": "NSE", \
    "tsym": "ACC-EQ", \
    "ai_t": "", \
    "validity": "DAY", \
    "remarks": "" \
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

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

Json Fields	Possible value	Description
exch*		Exchange Segment
ai_t*		Alert Type, should be original alert type, can't be modified.
al_id		Alert Id
validity*	DAY or GTT	Validity
d		Data to be compared with LTP
remarks*		Any message Entered during order entry.

**Sample Failure Response :**

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

## 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

## GET PENDING ALERT

To Get Pending Alert Request you need to make a POST call to the following url:  
<https://piconnect.flattrade.in/PiConnectTP/GetPendingAlert>

```
# Here is a curl example
curl --location
'https://BaseURL/GetPendingAlert' \
--header 'Content-Type: application/json'
```

## REQUEST DETAILS

Parameter	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

```
--data 'jData={
    "uid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

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"  
}
```

## GET ENABLED ALERT TYPES

To Get Enabled Alert Types Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/GetEnabledAlertTypes>

### REQUEST DETAILS

Parameter	Possible	
Name	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.
request_time		This will be present only in a successful response.
ai_ts		Array of alert types

```
# Here is a curl example Copy
curl --location
'https://BaseURL/GetEnabledAlertTypes' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

```
{
  "stat": "Ok",
  "request_time": "04062021121503",
  "ai_ts": [
    {"ai_t": "ATP"},
    {"ai_t": "LTP"},
    {"ai_t": "Perc. Change"}
  ]
}
```

Sample Failure Response :

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

## GET UNSETTLED TRADING DATE:

To Get UnSettled Trading date Request you need to make a POST call to the following url:  
<https://piconnect.flattrade.in/PiConnectTP/GetUnStledTradingDate>

## REQUEST DETAILS

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

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

## RESPONSE DETAILS

Response data will have below fields.

Json	Possible	
Fields	value	Description
stat		GTT order 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)

```
# Here is a curl example □
curl --location
'https://BaseURL/GetUnStledTradingDate' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

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:

To get max Payout Amount you need to make a POST call to the following url:  
<https://piconnect.flattrade.in/PiConnectTP/GetMaxPayoutAmount>

## REQUEST DETAILS

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

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

## RESPONSE DETAILS

Response data will have below fields.

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

```
# Here is a curl example □
curl --location
'https://BaseURL/GetMaxPayoutAmount' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

```
{
"request_time":"15:52:26 10-05-2021",
112
"stat":"Ok",
"actid":"C-GURURAJ",
"payout":21200.20
}
```

Sample Failure Response :

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

## FUNDS PAYOUT REQUEST

```
# Here is a curl example □
```

To get Funds Payout Request you need to make a POST call to the following url:  
<https://piconnect.flattrade.in/PiConnectTP/FundsPayOutReq>

## REQUEST DETAILS

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

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

```
curl --location
'https://BaseURL/FundsPayOutReq' \
--header 'Content-Type: application/json' \
--data 'jData={
    "actid": "FZ00000",
    "actid": "FZ00000",
    "payout": ""
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

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

Sample Failure Response :

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

## GET PAYIN REPORT

To get Payin Report you need to make a POST call to the following url:  
<https://piconnect.flattrade.in/PiConnectTP/GetPayinReport>

## REQUEST DETAILS

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

Json Fields	Possible value	Description
actid*		Login users account ID
from_date*		From date
to_date*		To date

## RESPONSE DETAILS

Response data will have below fields.

Json Fields	Possible value	Description
stat		success or failure indication.
actid		This will be present only in a successful response.
trans_ref_num		transaction reference number (number which defines each transaction)

```
# Here is a curl example □
curl --location
'https://BaseURL/GetPayinReport' \
--header 'Content-Type: application/json' \
--data 'jData={
    "actid": "FZ00000",
    "from_date": "",
    "to_date": ""
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

```
[
{
  "stat": "Ok",
  "actid": "GURURAJ",
  "trans_ref_num": "2021125000001",
  "tran_status": "Complete",
  "amt": "10000.00"
},
{
  "stat": "Ok",
  "actid": "GURURAJ",
  "trans_ref_num": "2021125000002",
  "tran_status": "Complete",
  "amt": "10000.00"
}
]
```

Sample Failure Response :

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

Json

Fields	Possible value	Description
tran_status	ADD_FUND_S T_COMPLETE _STR	This is used to indicate the status of transaction
amt	Amount	

## GET PAYOUT REPORT

To get Funds Payout Request you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/GetPayoutReport>

Parameter Possible

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

Json Possible

Fields	Possible value	Description
actid*		Login users account ID
from_date*		From date
to_date*		To date

## RESPONSE DETAILS

Response data will have below fields.

Json

Fields	Possible value	Description
stat	success or failure	indication.

# Here is a curl example 

```
curl --location
'https://BaseURL/GetPayoutReport' \
--header 'Content-Type: application/json' \
--data 'jData={
    "actid": "FZ00000",
    "from_date": "",
    "to_date": ""
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

```
[
{
  "stat": "Ok",
  "actid": "GURURAJ",
  "trans_ref_num": "2021127000002",
  "tran_status": "Complete",
  "amt": "-1000.00"
},
{
  "stat": "Ok",
  "actid": "GURURAJ",
  "trans_ref_num": "2021127000003",
  "tran_status": "Complete",
  "amt": "-100.00"
},
{
  "stat": "Ok",
  "actid": "GURURAJ",
  "trans_ref_num": "2021127000004",
  "tran_status": "Complete",
  "amt": "-1000.00"
},
{
  "stat": "Ok",
  "actid": "GURURAJ",
```

Json Fields	Possible value	Description
actid		This will be present only in a successful response.
trans_ref_num		transaction reference number (number which defines each transaction)
tran_status	WITHDRAW_ST_COMPLETE	This is used to indicate the status of transaction
amt		Amount

```

"trans_ref_num": "20211270000005",
"tran_status": "Complete",
"amt": "-100.00"
}
]

```

Sample Failure Response :

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

## CANCEL PAYOUT

To Cancel Payout you need to make a POST call to the following url:

<https://piconnect.flattrade.in/PiConnectTP/CancelPayout>

### 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
actid*		Login users account ID

```
# Here is a curl example □
curl --location
'https://BaseURL/CancelPayout' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
    "trans_ref_num": ""
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

```
{
"request_time": "18:59:25 12-05-2021",
"stat": "Ok",
"actid": "GURURAJ",
"tran_status": "88"
}
```

Sample Failure Response :

```
{
"stat": "Not_Ok",
"request_time": "18:58:47 12-05-2021",
"errmsg": "Error Occurred : -103 20211300000033"
```

Json Fields	Possible value	Description	is Already Canceled }
uid*		User id of the logged in user.	
trans_ref_num*		transaction reference number (number which defines each transaction)	
brkname		Broker name	

## RESPONSE DETAILS

Response data will have below fields.

Json Fields	Possible value	Description
stat		success or failure indication.
actid		This will be present only in a successful response.
tran_status		This is used to indicate the status of transaction
request_time		This will be present only in a successful response.

## WEB SOCKET API

Connect to  
wss://piconnect.flattrade.in/PiConnectWSTp/

```
# Here is a curl example
```

```
curl --location  
'wss://piconnect.flattrade.in/PiConnectWSTp/'  
--data '{  
    "t": "c",
```

## 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.

```

    "uid": "FZ00000",
    "actid": "FZ00000",
    "source": "API",
    "susertoken": "GHUDWU53H32MTHPA536Q32WR"
  }

```

## CONNECT

### REQUEST

Json Fields	Possible value	Description
t	c	'c' represents connect task
uid		User ID
actid		Account id
source	API	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

```

# Here is a curl example □

curl --location
'wss://picconnect.flattrade.in/PiConnectWSTp/'
--data '{
  "t": "t",

```

```
"k": "NSE|22#BSE|508123#NSE|10#BSE|2879"
```

```
}
```

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

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

# Here is a curl example [Copy](#)

```
curl --location
'wss://picconnect.flattrade.in/PiConnectWSTp/'
--data '{
  "t": "uk",
  "k": "NSE|22#BSE|508123#NSE|10#BSE|2879"
}'
```

### 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

# Here is a curl example 

```
curl --location
'wss://piconnect.flattrade.in/PiConnectWSTp/'
--data '{
    "t": "d",
    "k": "NSE|22#BSE|508123#NSE|10#BSE|2879"
}'
```

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",
```

Json Fields	Possible value	Description	
h	High price		"sq3": "1", "bq4": "139", "sq4": "53", "bq5": "393", "sq5": "94" }
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 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		
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		
sq2	Best Sell Quantity 2		
sq3	Best Sell Quantity 3		

Json Fields	Possible value	Description
sq4	Best Sell Quantity 4	
sq5	Best Sell Quantity 5	
sp1	Best Sell Price 1	
sp2	Best Sell Price 2	
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	

## SUBSCRIPTION DEPTH ACKNOWLEDGEMENT

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

Json Fields	Possible value	Description
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 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	
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	

Json Fields	Possible value	Description
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	
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	

## UNSUBSCRIBE DEPTH

### REQUEST

```
# Here is a curl example 
curl --location
'wss://piconnect.flattrade.in/PiConnectWSTp/'
--data '{
        "t": "ud",
        "k": "NSE|22#BSE|508123#NSE|10#BSE|2879"
      '
```

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

## RESPONSE

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

```
# Here is a curl example Copy
curl --location
'wss://piconnect.flattrade.in/PiConnectWSTp/'
--data '{
    "t": "o",
    "actid": "FZ00000"
}'
```

## SUBSCRIPTION ACKNOWLEDGEMENT

Json	Possible	
Fields	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 [If qty is junk value other than numbers].
prc*		Order Price [If prc is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price '0' ].
prd		Product
status		Order status (New, Replaced, Complete, Rejected etc)
reporttype		Order event for which this message is sent out. (Fill, Rejected, Canceled)
transtype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
prctyp		Order price type (LMT, MKT, SL-LMT, SL-MKT)
ret*	DAY / EOS / IOC	Order Retention type [ret should be DAY / EOS / IOC else reject]
fillshares		Total Filled shares for this order
avgprc		Average fill price

Json Fields	Possible value	Description
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 [If dscqty is junk value other than numbers].
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.

Json Fields	Possible value	Description
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

```
# Here is a curl example □
curl --location
'wss://piconnect.flattrade.in/PiConnectWSTp/'
--data '{
    "t": "uo"
}'
```

### RESPONSE

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

## USER DETAILS

### USER DETAILS

To get User details you need to make a POST call to the following url :  
<https://piconnect.flattrade.in/PiConnectTP/UserDetails>

```
# Here is a curl example □
curl --location 'https://BaseURL/UserDetails'
 \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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

Json feilds	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.

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"
  ]
}
```

Json Fields	Possible value	Description	
brkname	Broker id		"LMT", "SL-LMT", "SL-MKT", "DS", "2L", "3L", "4L" ], "brkname": "VIDYA", "brnchid": "VIDDU", "email": "gururaj@gmail.com", "actid": "GURURAJ", "uprev": "INVESTOR", "stat": "Ok" }
brnchid	Branch id		Sample Failure Response: { "stat": "Not_Ok", "emsg": "Session Expired : Invalid Session Key" }
email			
actid			
m_num	Mobile Number		
uprev	Always it will be an INVESTOR, other types of user not allowed to login using this API.		
access_type	Access Type		
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	

## SCRIPS

### SEARCH SCRIPS

To get Search scrips you need to make a POST call to the following url :

# Here is a curl example 

```
curl --location 'https://BaseURL/SearchScrip'
```

<https://piconnect.flattrade.in/PiConnectTP/SearchScrip>

```
\n
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "stext": "NIFTY",
    "exch": "NSE"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

## QUERY PARAMETERS

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

Json	Possible	
feilds	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	Possible	
Fields	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

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",
```

Json Fields	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

```

        "tsym": "RELBANKNAV-EQ"
    },
    {
        "exch": "NSE",
        "token": "2882",
        "tsym": "RELCAPITAL-EQ"
    },
    {
        "exch": "NSE",
        "token": "18070",
        "tsym": "RELCONSNAV-EQ"
    },
    {
        "exch": "NSE",
        "token": "18071",
        "tsym": "RELDIVNAV-EQ"
    },
    {
        "exch": "NSE",
        "token": "18072",
        "tsym": "RELGOLDNAV-EQ"
    },
    {
        "exch": "NSE",
        "token": "2885",
        "tsym": "RELIANCE-EQ"
    },
    {
        "exch": "NSE",
        "token": "15068",
        "tsym": "RELIGARE-EQ"
    },
    {
        "exch": "NSE",
        "token": "553",
        "tsym": "RELINFRA-EQ"
    },
    {
        "exch": "NSE",
        "token": "18074",
        "tsym": "RELVNV20NAV-EQ"
    }
]
}

```

Sample Failure Response :

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

## GET QUOTES

To get place order you need to make a POST call to the following url :

<https://piconnect.flattrade.in/PiConnectTP/GetQuotes>

```
# Here is a curl example 
curl \
jData={"uid":"FZ00000", "exch":"NSE",
"token":"22"}&jKey=GHUDWU53H32MTHPA536Q32WR
```

### QUERY PARAMETERS

Parameter	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

### RESPONSE DETAILS

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Watch list update success or failure indication.
request_time		It will be present only in a successful response.
exch	NSE, BSE, NFO ...	Exchange

Sample Success Response :

```
{
"request_time":"12:05:21 18-05-2021",
"stat":"OK",
,"exch":"NSE",
"tsym":"ACC-EQ",
"cname":"ACC LIMITED",
"symname":"ACC",
"seg":"EQT",
"instname":"EQ",
"isin":"INE012A01025",
"pp":"2",
"ls":"1",
"ti":"0.05",
"mult":"1",
"uc":"2093.95",
"lc":"1713.25",
"prcftr_d":"(1 / 1) * (1 / 1)",
"token":"22",
"lp":"0.00",
"h":"0.00",
"l":"0.00",
```

Json Fields	Possible value	Description	
tsym		Trading Symbol	"v":"0", "ltq":"0", "ltt":"05:30:00", "bp1":"2000.00", "sp1":"0.00", "bp2":"0.00", "sp2":"0.00", "bp3":"0.00", "sp3":"0.00", "bp4":"0.00", "sp4":"0.00", "bp5":"0.00", "sp5":"0.00", "bq1":"2", "sq1":"0", "bq2":"0", "sq2":"0", "bq3":"0", "sq3":"0", "bq4":"0", "sq4":"0", "bq5":"0", "sq5":"0", "bo1":"2", "so1":"0", "bo2":"0", "so2":"0", "bo3":"0", "so3":"0", "bo4":"0", "so4":"0", "bo5":"0", "so5":"0"
cname		Company Name	}
symname		Symbol Name	Sample Failure Response :
seg		Segment	{ "stat":"Not_Ok", "request_time":"10:50:54 10-12-2020", "emsg":"Error Occurred : 5 \"no data\""
instname		Intrument Name	}
isin		ISIN	
pp		Price precision	
ls		Lot Size	
ti		Tick Size	
mult		Multiplier	
uc		Upper circuit limit	
lc		Lower circuit limit	
prcftr_d		Price factor ((GN / GD) * (PN/PD))	
token		Token	
lp		LTP	
h		Day High Price	
l		Day Low Price	
v		Volume	
ltq		Last trade quantity	
ltt		Last trade time	
ltd	dd-mm-yy	Last Trade Date	
bp1		Best Buy Price 1	
sp1		Best Sell Price 1	
bp2		Best Buy Price 2	
sp2		Best Sell Price 2	
bp3		Best Buy Price 3	
sp3		Best Sell Price 3	
bp4		Best Buy Price 4	
sp4		Best Sell Price 4	

Json Fields	Possible value	Description
bp5	Best Buy Price 5	
sp5	Best Sell Price 5	
bq1	Best Buy Quantity 1	
sq1	Best Sell Quantity 1	
bq2	Best Buy Quantity 2	
sq2	Best Sell Quantity 2	
bq3	Best Buy Quantity 3	
sq3	Best Sell Quantity 3	
bq4	Best Buy Quantity 4	
sq4	Best Sell Quantity 4	
bq5	Best Buy Quantity 5	
sq5	Best Sell Quantity 5	
bo1	Best Buy Orders 1	
so1	Best Sell Orders 1	
bo2	Best Buy Orders 2	
so2	Best Sell Orders 2	
bo3	Best Buy Orders 3	
so3	Best Sell Orders 3	
bo4	Best Buy Orders 4	
so4	Best Sell Orders 4	
bo5	Best Buy Orders 5	
so5	Best Sell Orders 5	
und_exch	Underlying Exch seg	
und_tk	Underlying Token	
ord_msg	Order Message	
sptprc	Spot Price [ # ]	
issuecap	issue capital	
e_date	end date	

## POSTBACK / WEBHOOK

You will be receiving order updates for the orders placed through API.

### RESPONSE DETAILS

Response data will be in json format with below fields

Json Fields	Possible value	Description
norenordno	Noren (Number)	User Id
uid	Accour	Exchan
actid	Trading	Order C
exch	Order C	qty is ju
tsym	Order C	other th
qty*	Order C	number
prc*	Order F	is junk
	is junk	than nu
	than nu	"Order I
	"Order I	cannot
	cannot	prctyp :
	prctyp :	MKT' w
	MKT' w	].
prd	Produc	
status	Order s	
	Replace	
	Comple	
	etc)	
reporttype	Order e	
	which t	
	messag	

Sample:

```
{
  "norenordno": "23010500000376",
  "kdid": "1",
  "uid": "ASHWATHINV123",
  "actid": "ASHWATHINV",
  "exch": "NSE", "tsym": "ACC-EQ",
  "qty": "1",
  "rorgqty": "0",
  "ipaddr": "117.248.82.174",
  "ordenttm": "1672921211",
  "mkt_protection": "0.00",
  "sno_fillid": "",
  "trantype": "B",
  "prctyp": "LMT",
  "ret": "DAY",
  "amo": "Yes",
  "token": "22",
  "prc": "2500.00",
  "pcode": "C",
  "remarks": "",
  "status": "OPEN",
  "rpt": "New",
  "ls": "1",
  "ti": "0.05",
  "rprc": "2500.00",
  "dscqty": "0",
  "norentm": "17:50:11 05-01-2023",
  "checksum": "619521a541ff3e634ecb02147f0cb77e
               822ea415c9b79259cd5e40592a73b810"
}
```

Json Fields	Possible value	Description
		out.(Filled, Cancelled)
transtype*	B / S	B -> BU [transty be 'B' or reject].
prctyp		Order p (LMT, M LMT, SI
ret*	DAY / EOS / IOC	Order R type [re DAY / E else rej
fillshares		Total Fi for this
avgprc		Average
fltm		Fill Tim only w/ reportty
fid		Fill ID (j only w/ reportty
flqty		Fill Qty(1 only w/ reportty
flprc		Fill Pric only w/ reportty
rejreason		Order re reason,
exchordid		Exchan
cancelqty		Cancel in case cancelle
remarks		User ac while p order

Json Fields	Possible value	Description
dscqty*		Discloses quantity [If dscq is present, value of dscqty is number]
trgprc		Trigger price for SL order
snonum		This will be present if orders cover a range of orders, needs to be triggered during execution
snoordt		This will be present if orders cover a range of orders, indicates the order type or stop
blprc		This will be present if bracket order. To differerentiate loss trigger from to be entered
bpprc		This will be present if bracket order. To differerentiate price to be entered from entered
trailprc		This will be present if bracket order. To require a trailing stop

Json Fields	Possible value	Description
		ticks is enabled
exch_tm		This wi exchan time Fc mm-YY hh:MM
amo*	Yes	The me "Invalid be disp "amo" f sent wi value. I not req not sen
tm		TimeSt
kidid		Kid Id
sno_fillid		BO Seq
checksum	sha256 [ noren_order_num +noren_time_stamp+vendor_ key ]	CheckS sure ch matche any thir sendinç order u your url

## SCRIP MASTER

### Scrip Groups

NSE - Equity	<a href="#">DOWNLOAD </a>
NSE - Equity Derivatives	<a href="#">DOWNLOAD </a>
NSE - Index Derivatives	<a href="#">DOWNLOAD </a>
NSE - Currency Derivatives	<a href="#">DOWNLOAD </a>

### Scrip Groups

MCX - Commodity

DOWNLOAD 

BSE - Equity

DOWNLOAD 

BSE - Index Derivatives

DOWNLOAD 

BSE - Equity Derivatives

DOWNLOAD 

## ORDER API RATE LIMIT

Time Frame	Rate Limit
Per Second	10
Per Minute	40

## API RATE LIMIT

Time Frame	Rate Limit
Per Second	40
Per Minute	200