Pi *I* docs

Version: 1.19.0

Last Updated: 25th Jun, 2024

Don't have an account? Sign up now!

Usage terms

☐ 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.

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"

My API Keys

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 reciving 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

☐ I read and agree to the terms

CREATE CANCEL

4. Your API key is now generated

My API Keys CREATE NEW API KEY

Status	AppName		ApiKey	SecretKey	Regenerate Key
ca	YOURAPP	https://yourapp.com/redirect?	e110d69999999999e913a8f354 <T>		

5. Copy the APIKEY

6. Click the eye icon <•> to view the secret key. Copy secret key

My API Keys

E3 YOURAPP https://yourapp.com/redirect? e110d69999999999e913a8f354

Rowsperpage

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 a web browser. You need to authorize the program using your client ID (UCC), trading password and PAN/year of birth.

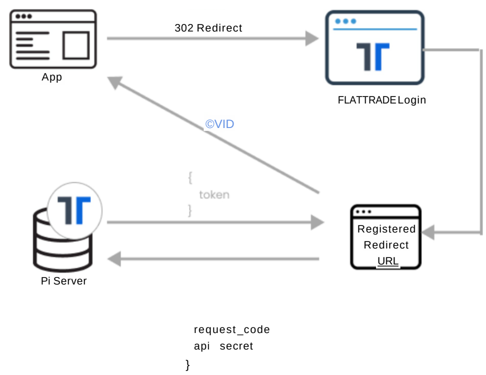
Whether your program runs on a GUI or a console, you always have to access the web browser to create an access token which then allows you to use the API.

The access token has a validity of 24 hours so you only have to generate access tokenonce 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).

Base URL

https ://piconnect.flattrade .in/PiConnectTP

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 https ://auth.flattrade.in/ ?app_key=APIKEY in browser Note: replace APIKEY text in this URL with the Apikey allocated for you in the <u>above step</u> 4
2	Enter your Client id (UCC), Password, PAN/DOB and submit
3	After you are authorized in our authentication portal, the authorization screen will redirect to your URL with request-token (https://yourRedirectURL.com/?request_code=requestCodeValue) <u>Note:</u> Redirect URL is pre-registered with us against each API Key. If you have different redirect URL for PROD and TESTING, then each environment should have different API Key registered with us <u>reauest_code:</u> It is an one-time code obtained during the login flow. This code's lifetime is only a few minutes and it is meant to be exchanged for a token immediately after obtaining

4	Call to https://authapi.flattrade.in/trade/apitoken in POST method to validate request_code and get the token
	{
	"api_key":"xcwwegfhgh4454646", "request_code":"xxdfddfdfdsfdsf84okkdlfelfdfdfd345fsf",
	"api_secret":"sdfdsfsdfdsfXXXXXXX"
	}
	NOTE: Please refer this document for a detailed walkthrough on how to handle the security key parameter to get the token.
	api_key:The public API key
	<u>reauest_code:</u> It is an one-time code obtained during the login flow. This code's lifetime is only a few minutes and it is meant to be exchanged for a token immediately after obtaining
	api_secret:SHA-256 hash of (apLkey + request-token + apLsecret)

5	You will get a response with following values. This token can be used in appropriate end points to get more details of the user
	<u>On Sucess:</u>
	{
	"token":"dsfdsf84okkdlfelfdfdfd3454545454ssdfsf",
	"client":"CCODE123",
	"status":"Ok",
	"emsg":""
	}

POSTMAN COLLECTIONS

To Download the Postman Collections ♦

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.

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)

```
# Here is a curl example  IQ

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

		unique lu ui cui iii act uii wniu,ii uiuei Luuc paceu. (uoc un encoding to avoid special char error for symbols like M&M)
qty*		Order Quantity [If qty is junk value other than numbers].
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price 'O'].
trgpre		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)
trantype*	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.
bppre		Book Profit Price applicable only if product is selected as B (Bracket order)
blpre		Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order)
trailpre		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.
tsym2		Trading symbol of second leg, mandatory for price type 2L and 3L (use url encoding to avoid special char error for symbols like M&M)
trantype2		Transaction type of second leg, mandatory for price type 2L and 3L
qty2		Quantity for second leg, mandatory for price type 2L and 3L
prc2		Price for second leg, mandatory for price type 2L and 3L
tsym3		Trading symbol of third leg, mandatory for price type 3L (use url encoding to avoid special char error for symbols like M&M)
trantype3		Transaction type of third leg, mandatory for price type 3L
qty3		Quantity for third leg, mandatory for price type 3L
prc3		Price for third leg, mandatory for price type 3L
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	OkorNot_Ok	Place order success or failure indication.
request_time		Response received time
norenordno		It will be present only on successful Order placement to OMS.
emsg		This will be present only if Order placement fails

Sample Success Response:


```
{
"request_time": "10:48:03 20-05-2020",
"stat": "Ok",
"norenordno": "20052000000017"
}
```


Sample Error Response :


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


MODIFY ORDER

To get Modify order you need to make a POST call to the following url :
https://piconnect.flattrade.in/PiConnectTP/ModifyOrder

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
exch*		Exchange
norenordno*		Noren order number, which needs to be modified
prctyp	LMT / MKT / SL-MKT/SL-LMT	This can be modified
pre*		Modified / New price [If pre is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price 'O'].
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 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)

```
# Here is a curl example IQ
curl --location 1https://BaseURL/ModifyOrder' \
--header 'Content-Type: application/json' \
--data 1jData={
        "uid": "FZ00000",
        "actid": "FZ00000",
        "exch": "NSE",
        "tsym": "ACC-EQ",
        "qty": "50",
        "pre": "1400",
        "prctyp": "LMT",
        "ret": "DAY",
        "norenordno": "123456789"
    }&j Key=GHUDWU53H32MTHPA536Q32WR'
```

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)
blpre		Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order)
trailpre		Trailing Price applicable only if product is selected as H and B (High Leverage and Bracket order)

RESPONSE DETAILS

Response data will be in json format with below fields

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Modify order success or failure indication.
result		Noren Order number of the order modified.
requestjime		Response received time
emsg		This will be present only if Order modification fails

Sample Success Response :

```
{
"requestJ: ime    14:14:08 26-05-2020",
"stat" : "Ok",
"result" : "20052600000103"
}
```

Sample Failure Response :

```
{
"requestJ: ime" : "16:03:29 28-05-2020",
"stat": "Not_Ok",
"emsg": "Rejected : 0RA:0rder not found"
}
```

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 IQ

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.
requestjime		Response received time
emsg		This will be present only if Order cancelation fails

Sample Success Response :

```
{
"requestJ: ime" : "14:14:10 26-05-2020",
"stat": "Ok",
"result" : "20052600000103"
}
```

Sample Failure Response :

```
{
"requestJ: ime    16:01:48 28-05-2020",
"stat" : "Not_Ok",
"emsg": "Rejected : 0RA:0rder 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 IQ

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.

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

RESPONSE DETAILS

Response data will be in json format with below fields

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Cancel order success or failure indication.
dmsg		Display message, (will be present only in case of success).
requestjime		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 IQ

curl --location 'https://BaseURL/GetOrderMargin' \
--header 'Content-Type: application/json' \
--data 'jData={
    ■■uid": "FZ00000",
    "actid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "qty": "50",
    "pre": "1400",
    "prd": "H",
    "trantype": "B",
    "pretyp": "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].
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if pretyp = 'MKT/ SL-MKT" with price '0'].
trgpre		Only to be sent in case of SL / SL-M order.
prd*	C / M / H	Product name (Select from 'prarr' Array provided in User Details response, and if same is allowed for selected, exchange. Show product display name, for user to select, and send corresponding prd in API call)
trantype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or "S' else reject].
pretyp*	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.
rorgpre		Optional field. Application only for modify order, open order price
orgtrgpre		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	OkorNot_Ok	Place order success or failure indication.
requestjime		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 IQ

curl --location 'https://BaseURL/GetBasketMargin' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "qty": "50",
    "pre": "1400",
    "prd": "H",
    "trantype": "B",
    "pretyp": "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].

		be zero" [if pretyp = 'MKT/ SL-MKT with price 'O'].
trgpre		Only to be sent in case of SL / SL-M order.
prd*	C / M / H	Product name (Select from 'prarr' Array provided in User Details response, and if same is allowed for selected, exchange. Show product display name, for user to select, and send corresponding prd in API call)
trantype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
pretyp*	LMT / MKT / SL-LMT / SL-MKT	
blpre		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.
rorgpre		Optional field. Application only for modify order, open order price
orgtrgpre		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].
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if pretyp = 'MKT/ SL-MKT' with price 'O'].
trgpre		Only to be sent in case of SL / SL-M order.
prd*	C / M / H	Product name (Select from 'prarr' Array provided in User Details response, and if same is allowed for selected, exchange. Show product display name, for user to select, and send corresponding prd in API call)
trantype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
pretyp*	LMT / MKT / SL-LMT / 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	OkorNot_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 :
https://piconnect.flattrade.in/PiConnectTP/OrderBook

```
# Here is a curl example  IQ

curl --location 'https://BaseURL/OrderBook' \
--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
prd	H / M / ...	Product name

RESPONSE DETAILS

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

```
Sample Success Output :
Success response :
[
{
"stat" : "Ok",
"exch" : "NSE" ,
"tsym" : "ACC-EQ" ,
"norenordno" : "20062500000001223",
"pre" : "127230",
```

Json Fields	Possible value	Description
stat	Ok or	Order book success or failure indication.

	Not_Ok	
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 returned in user details.
status		Order status
trantype*	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 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.
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

```
"qty" : "100",
"prd" : "C",
"status": "Open",
"trantype" : "B",
"prctyp" : "LMT",
"fillshares" : "0",
"avgprc" : "0",
"exchordid" : "250620000000343421",
"uid" : "VIDYA",
"actid" : "CLIENT1",
"ret" : "DAY",
"amo" : "Yes"
L
{
"stat" : "Ok",
"exch" : "NSE" ,
"tsym" : "ABB-EQ" ,
"norenordno" : "20062500000002543",
"prc" : "127830",
"qty" : "50",
"prd" : "C",
"status": "REJECT",
"trantype" : "B",
"prctyp" : "LMT",
"fillshares" : "0",
"avgprc" : "0",
"re jreason" : "Insufficient funds"
"uid" : "VIDYA",
"actid" : "CLIENT1",
"ret" : "DAY",
"amo" : "No"
}
1
Sample Failure Response :
{
"stat":"Not_Ok",
"emsg"/'Session Expired : Invalid Session Key"
}
```

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 IQ

curl --location 'https://BaseURL/MultiLegOrderBook' \
--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
prd	H/M/...	Product name

RCOrUINOC UC IMILO

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 'O'].
qty*		Order Quantity [If qty is junk value other than numbers].
prd		Display product alias name, using prarr returned in user details.
status		Order status
trantype*	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 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
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 IQ

curl --location 'https://BaseURL/SingleOrdHisf \
--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
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
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if prctyp = 'MKT/ SL-MKT' with price 'O'].
qty*		Order Quantity [If qty is junk value other than numbers].
prd		Display product alias name, using prarr returned in user details.
status		Order status
rpt		Report Type (fill/complete etc)
trantype*	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
avgpre		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].
trgpre		Order trigger price
ret*	DAY / EOS / IOC	Order validity [ret should be DAY / EOS / IOC else reject]
uid		
actid		
bppre		Book Profit Price applicable only if product is selected as B (Bracket order)
blpre		Book loss Price applicable only if product is selected as H and B (High Leverage and Bracket order)
trailpre		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

Sample Success Output :


```
[
{
"stat": "Ok","norenordno": "20121300065716",
"uid": "DEMOI",
"actid": "DEMOI",
"exch": "NSE",
"tsym": "ACCELYA-EQ",
"qty": "180",
"trantype": "B",
"prctyp": "LMT",
"ret": "DAY",
"token": "7053",
"pp": "2",
"ls": "1",
"ti": "0.05",
"pre": "800.00",
"avgpre": "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"
k
{
"stat": "Ok",
"norenordno": "20121300065716",
"uid": "DEMOI",
"actid": "DEMOI",
"exch": "NSE",
"tsym": "ACCELYA-EQ",
"qty": "180",
"trantype": "B",
"prctyp": "LMT",
"ret": "DAY",
"token": "7053",
"pp": "2",
"ls": "1",
"ti": "0.05",
"pre": "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"
k
{
"stat": "Ok",
"norenordno": "20121300065716",
"uid": "DEMOI",
"actid": "DEMOI",
"exch": "NSE",
"tsym": "ACCELYA-EQ",
"qty": "180",
"trantype": "B",
"prctyp": "LMT",
"ret": "DAY",
"token": "7053",
"pp": "2",
"ls": "1",
"ti": "0.05",
"pre": "800.00",
"dscqty": "0",
"prd": "M",
"status": "PENDING",
"rpt": "PendingNew",
"norentm": "19:59:32 13-12-2020",
"remarks": "WC TEST Order"
k
{
"stat": "Ok",
"norenordno": "20121300065716",
"uid": "DEMOI",
"actid": "DEMOI",
"exch": "NSE",
"tsym": "ACCELYA-EQ",
"qty": "180",
"trantype": "B",
"prctyp": "LMT",
"ret": "DAY",
"token": "7053",
"pp": "2",
"ls": "1",
"ti": "0.05",
"pre": "800.00",
"prd": "M",
"status": "PENDING",
"rpt": "NewAck",
"norentm": "19:59:32 13-12-2020",
"remarks": "WC TEST Order"
}
]
```


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   IQ

curl --location 'https://BaseURL/TradeBook' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
>&jKey=GHUDWU53H32MTHPA536032WR'
```

QUERY PARAMETERS

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.
norenordno		Noren Order Number
qty*		Order Quantity [If qty is junk value other than numbers].
prd		Display product alias name, using prarr returned in user details.
trantype*	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
ordersource		Order Source
token		Token

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

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

Sample Success Output :


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


POSITIONS BOOK

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  IQ

curl --location 'https://BaseURL/PositionBook' \
--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	Position book success or failure indication.
exch		Exchange Segment
tsym		Trading symbol / contract.

Sample Success Response :


```
[
{
"stat":"Ok",
"uid":"P00RNA",
"actid":"P00RNA",
"exch":"NSE",
"tsym":"ACC-EQ",
"prarr":"C",
"pp":"2",
"ls":"1",
"ti":"5.00",
"mult":"1",
"prcftr":"1.000000",
"daybuyqty":"2",
```

uid	User Id
actid	Account Id
prd	Product name to be shown.
netqty	Net Position quantity
netavgprc	Net position average price
daybuyqty	Day Buy Quantity
daysellqty	Day Sell Quantity
daybuyavgprc	Day Buy average price
daysellavgprc	Day buy average price
daybuyamt	Day Buy Amount
daysellamt	Day Sell Amount
cfbuyqty	Carry Forward Buy Quantity
cforgavgprc	Original Avg Price
cfsellqty	Carry Forward Sell Quantity
cfbuyavgprc	Carry Forward Buy average price
cfsellavgprc	Carry Forward Buy average price
cfbuyamt	Carry Forward Buy Amount
cfsellamt	Carry Forward Sell Amount
totsellavgprc	Total Sell Avg Price
IP	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).
	Tick size
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:

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


```
■■daysellqty":"2",
"daybuyamt":"2610.00",
"daybuyavgprc":"1305.00",
"daysellamt":"2610.00",
"daysellavgprc":"1305.00",
"cfbuyqty":"0",
"cfsellqty" :"0",
"cfbuyamt":"0.00",
"cfbuyavgprc":"0.00",
"cfsellamt":"0.00",
"cfsellavgprc":"0.00",
"openbuyqty":"0",
"opensellqty":"23",
"openbuyamt":"0.00",
"openbuyavgprc":"0.00",
"opensellamt":"30015.00",
"opensellavgprc":"1305.00",
"netqty":"0",
"netavgprc":"0.00",
"lp": "0.00",
"urmtom":"0. 00",
"rpnl":"0.00",
"cforgavgprc":"0.00"
}
]
```


Sample Failure Response :


```
{
"stat":"Not_Ok",
"request_time":"14:14:11 26-05-2020",
"emsg":"Error Occurred : 5 \"no data\""
}
```


PRODUCT CONVERSION

To get Product Conversion you need to make a POST call to the following url :
https://piconnect.flattrade.in/PiConnectTP/ProductConversion

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
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.
trantype*	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

```
# Here is a curl example |Q

curl --location 'https://BaseURL/ProductConversion' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "actid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "qty": "50",
    "pre": "1400",
    "prd": "H",
    "trantype": "B",
    "pretyp": "LMT",
    "prevprd": "C",
    "postype": "Day"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

RESPONSE DETAILS

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Position conversion success or failure indication.
emsg		This will be present only if Position conversion fails.

Sample Success Response :

```
{
"request_time":"10:52:12 02-06-2020",
"stat":"Ok"
}
```

Sample Failure Response :

```
{
"stat":"Not_Ok",
"emsg":"Invalid Input :  Invalid Position Type"
}
```

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.
trantype*	B/S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
pretyp*	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].
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if pretyp = 'MKT/ SL-MKT' with price 'O'].
dscqty*		Disclosed quantity (Max 10% for NSE, and 50% for MCX) [If dscqty is junk value other than numbers].

```
# Here is a curl example  IQ

curl --location 'https://BaseURL/PlaceGTTOrder' \
--header 'Content-Type: application/json" \
--data 1jData={
    "uid": "FZ00000",
    "exch": "NSE",
    "tsym": "ACC-EQ",
    "validity": "DAY",
    "qty": "50",
    "pre": "1400",
    "prd": "H",
    "trantype": "B",
    "pretyp": "LMT”,
    "prevprd": "C",
    "ret": "DAY",
    "dscqty": "10"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

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

Sample Success Response :

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

Sample Failure Response :

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

MODIFY GTT ORDER

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
aljd		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].
pretyp*	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].

```
# Here is a curl example  IQ

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",
    "pre": "1400",
    "prd": "H",
    "trantype": "B",
    "pretyp": "LMT",
    "prevprd": "C",
    "ret": "DAY",
    "dscqty": "10"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

Sample Success Response :

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

pre*	Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if pretyp = 'MKT/ SL-MKT' with price 'O'].
dscqty*	Disclosed quantity (Max 10% for NSE, and 50% for MCX) [If dscqty is junk value other than numbers].

```
]
```

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		GTT order success or failure indication.
request_time		This will be present only in a successful response.
aljd		Alert Id
emsg		This will be present only in case of errors. That is : 1) Invalid 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

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

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

Sample Success Response :

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

Sample Failure Response :

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

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.
aljd		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

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

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

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",
"pretyp":"LMT",
"Qty":l,
"Pre":"1305.00",
"C":"C",
"prd":"C",
"ordersource":"API",
"d": "1900.00"
}
]
```

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.
ai_t		Alert type
aljd		Alert Id
tsym		Trading symbol
exch		Exchange Segment
token		Contract token
remarks		Any message Entered during order entry.
validity	DAY or GTT	Validity
d		Data to be compared with LTP
trantype*	B/S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
pretyp	LMT / MKT/ SL-LMT /	

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].
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if pretyp = 'MKT/ SL-MKT' with price 'O'].
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

Here is a curl example IQ

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

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		GTT order success or failure indication.
request_time		This will be present only in a successful response.
ai_ts		Array of alert types

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

Input Sample: IQ

```
jData={"uid":"FZ00000","ai_t" : "LMT_B0S_0","remarks": "admn","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", "pretyp": "MKT","prd" : "C",
"ret" : "DAY","actid": "FZ00000","uid" : "FZ00000", "ordersource": "WEB","qty" : "1",
"pre" : "0"}, "place_order_params_leg2": {"tsym": "ACC-EQ", "exch": "NSE", "trantype": "S",
"pretyp": "MKT","prd" : "C", "ret" : "DAY","actid": "FZ00000","uid" : "FZ00000",
"ordersource": "WEB","qty" : "1",
"pre" : "0"}}&jKey=652c99c82d7edcd4f472869786074c90bd27dfd0c68635c2e53db0ed08cbea0f
```

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.

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
trantype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
pretyp*		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	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].
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if pretyp = 'MKT/ SL-MKT' with price 'O'].
trgpre		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":"01 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.

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.

```
# Here is a curl example [Q

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

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
trantype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
pretyp*		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	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].
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if pretyp = 'MKT/ SL-MKT' with price 'O'].

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 ii."stat":"Invalid Oi" - incase of failure
request_time		This will be present only in a successful response.
aljd		Alert Id
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

Sample Success Response :


```
{
"request-time":"11:14:52 11-10-2021",
"stat":"01 replaced",
"al_id":"21101100000001"
}
```


Sample Failure Response :


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


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

Here is a curl example IQ

```
curl --location 'https://BaseURL/CancelOCOOrder' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "al_id": "21083000000040"
}&jKey=GHUDWU53H32MTHPA536Q32WR'
```

RESPONSE DETAILS

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.
aljd		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":"17:41:02 30-08-2021",
"stat":"Oi delete success"
,"al_id":"21083000000040"
}
```


Sample Failure Response :


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


HOLDINGSAND 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.

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

Here is a curl example IQ

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

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

Sample Success Response :


```
[
{
"stat":"Ok",
"exch_tsym":[
{
"exch":"NSE",
"token":"13",
"tsym":"ABB-EQ"
}
J,
"holdqty":"2000000",
"colqty":"200",
"btstqty":"0",
"btstcolqty":"0",
"usedqty":"0",
"upldprc" : "1800.00"
L
{
"stat":"Ok",
"exch tsym":[
```

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

Notes:

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

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

Exch_tsym object:

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

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

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

```
{
"exch":"NSE",
"token":"ZZ",
"tsym":"ACC-EQ"
}
],
"holdqty":"Z000000",
"colqty":"200",
"btstqty":"0",
"btstcolqty":"0",
"usedqty":"0",
"upldprc" : "1400.00"
}
]
Sample Failure Response :
{
"stat":"Not_Ok",
"emsg”:"Invalid Input : Missing uid or actid or prd."
}
```

LIMITS

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

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.

```
# Here is a curl example  IQ

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

RESPONSE DETAILS

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Limits request success or failure indication.
actid		Account id
prd		Product name
seg	CM / FO / FX	Segment
exch		Exchange
---------------------Cash Primary Fields-		
cash		Cash Margin available
payin		Total Amount transferred using Payins today
payout		Total amount requested for withdrawal today
---------------------Cash Additional Fields-		
brkcollamt		Prevalued Collateral Amount
unclearedcash		Uncleared Cash (Payin through cheques)
daycash		Additional leverage amount / Amount added to handle system errors - by broker.
---------------------Margin Utilized---------------------------		
marginused		Total margin / fund used today
mtomcurper		Mtom current percentage
---------------------Margin Used components-----------------		
cbu		CAC Buy used
esc		CAC Sell Credits
rpnl		Current realized PNL
unmtom		Current unrealized mtom

```
Sample Success Response :
(
"request_time":"18:07:31 29-05-2020",
"stat":"Ok",
"cash”:"1500000000000000.00",
"payin":"0.00",
"payout":"©.00",
"brkcollamt":"0.00",
"unclearedcash":"0.00”,
"daycash":"0.00",
"turnoverlmt":"50000000000000.00",
"pendordvallmt":"2000000000000000.00",
"turnover":"3915000.00",
"pendordval":"Z871000.00",
"marginused":"3945540.00",
"mtomcurper":"0.00",
"urmtom":"30540.00",
"grexpo":"3915000.00",
"uzpnl_e_i":"15Z70.00",
"uzpnl_e_m":"61080.00",
"uzpnl_e_c":"-45810.00"
}
Sample Failure Response :
{
"stat":"Not_Ok",
"emsg":"Server Timeout :  "
}
```

marprt	Covered Product margins
span	Span used
expo	Exposure margin
premium	Premium used
varelm	Var Elm Margin
grexpo	Gross Exposure
greexpo_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
grcoll	Valuation of uploaded holding pre haircut

----------------------Additional Risk Limits—

turnoverlmt	
pendordvallmt	

----------------------Additional Risk Indicators

turnover	Turnover
pendordval	Pending Order value

-Margin used detailed breakup fields

rzpnLeJ	Current realized PNL (Equity Intraday)
rzpnl_e_m	Current realized PNL (Equity Margin)
rzpnl_e_c	Current realized PNL (Equity Cash n Carry)
rzpnLdJ	Current realized PNL (Derivative Intraday)
rzpnl_d_m	Current realized PNL (Derivative Margin)
rzpnLfJ	Current realized PNL (FX Intraday)
rzpnl_f_m	Current realized PNL (FX Margin)
rzpnLeJ	Current realized PNL (Commodity Intraday)
rzpnl_c_m	Current realized PNL (Commodity Margin)
uzpnLeJ	Current unrealized MTOM (Equity Intraday)
uzpnl_e_m	Current unrealized MTOM (Equity Margin)
uzpnl_e_c	Current unrealized MTOM (Equity Cash n Carry)
uzpnLdJ	Current unrealized MTOM (Derivative Intraday)
uzpnl_d_m	Current unrealized MTOM (Derivative Margin)
uzpnLfJ	Current unrealized MTOM (FX Intraday)
uzpnl_f_m	Current unrealized MTOM (FX Margin)
uzpnLeJ	Current unrealized MTOM (Commodity Intraday)
uzpnl_c_m	Current unrealized MTOM (Commodity Margin)
span_dj	Span Margin (Derivative Intraday)
span_d_m	Span Margin (Derivative Margin)
spanJJ	Span Margin (FX Intraday)
span_f_m	Span Margin (FX Margin)
span_c_i	Span Margin (Commodity Intraday)
span_c_m	Span Margin (Commodity Margin)
expo_d_i	Exposure Margin (Derivative Intraday)
expo_d_m	Exposure Margin (Derivative Margin)
expo_f_i	Exposure Margin (FX Intraday)
expo_f_m	Exposure Margin (FX Margin)
expo_c_i	Exposure Margin (Commodity Intraday)
expo_c_m	Exposure Margin (Commodity Margin)
premium_d_i	Option premium (Derivative Intraday)
premium_d_m	Option premium (Derivative Margin)
premium_fj	Option premium (FX Intraday)
premium_f_m	Option premium (FX Margin)
premium_c_i	Option premium (Commodity Intraday)
premium_c_m	Option premium (Commodity Margin)
varelm_e_i	Var Elm (Equity Intraday)
varelm_e_m	Var Elm (Equity Margin)
varelm_e_c	Var Elm (Equity Cash n Carry)
marprt_e_h	Covered Product margins (Equity High leverage)
marprt_e_b	Covered Product margins (Equity Bracket Order)
marprt_d_h	Covered Product margins (Derivative High leverage)
marprt_d_b	Covered Product margins (Derivative Bracket Order)
marprt_f_h	Covered Product margins (FX High leverage)
marprt_f_b	Covered Product margins (FX Bracket Order)
marprt_c_h	Covered Product margins (Commodity High leverage)
marprt_c_b	Covered Product margins (Commodity Bracket Order)
scripbskmar_e_i	Scrip basket margin (Equity Intraday)
scripbskmar_e_m	Scrip basket margin (Equity Margin)
scripbskmar_e_c	Scrip basket margin (Equity Cash n Carry)

addscripbskmrg_ dj	Additional scrip basket margin (Derivative Intraday)
addscripbskmrg_ d_m	Additional scrip basket margin (Derivative Margin)
addscripbskmrg_f J	Additional scrip basket margin (FX Intraday)
addscripbskmrg_f _m	Additional scrip basket margin (FX Margin)
addscripbskmrg_ cj	Additional scrip basket margin (Commodity Intraday)
addscripbskmrg_c_m	Additional scrip basket margin (Commodity Margin)
brkage_e_i	Brokerage (Equity Intraday)
brkage_e_m	Brokerage (Equity Margin)
brkage_e_c	Brokerage (Equity CAC)
brkage_e_h	Brokerage (Equity High Leverage)
brkage_e_b	Brokerage (Equity Bracket Order)
brkage_d_i	Brokerage (Derivative Intraday)
brkage_d_m	Brokerage (Derivative Margin)
brkage_d_h	Brokerage (Derivative High Leverage)
brkage_d_b	Brokerage (Derivative Bracket Order)
brkageJJ	Brokerage (FX Intraday)
brkage_f_m	Brokerage (FX Margin)
brkage_f_h	Brokerage (FX High Leverage)
brkage_f_b	Brokerage (FX Bracket Order)
brkage_c_i	Brokerage (Commodity Intraday)
brkage_c_m	Brokerage (Commodity Margin)
brkage_c_h	Brokerage (Commodity High Leverage)
brkage_c_b	Brokerage (Commodity Bracket Order)
mr_fx_u	MR fx used
mr_sell	MR sell credit
mr_t1sell	MRt1 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

```
# Here is a curl example  IQ

curl --location 'https://BaseURL/GetIndexList' \
--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	Limits request success or failure indication.
values		Array Of Basket, Criteria pair.
request_time		This will be present only in a successful response.
emsg		This will be present only in case of errors.

BASKET, CRITERIA PAIR OBJECT :

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

Sample Output:


```
{
"request_time": "20:12:29 13-12-2020",
"values": [
{
"idxname": "HangSeng BeES-NAV",
"token": "26016"
k
{
"idxname": "India VIX",
"token": "26017"
k
{
"idxname": "Nifty 50",
"token": "26000"
k
{
"idxname": "Nifty IT",
"token": "26008"
k
{
"idxname": "Nifty Next 50",
"token": "26013"
k
{
"idxname": "Nifty Bank",
"token": "26009"
k
{
"idxname": "Nifty 500",
"token": "26004"
k
(
"idxname": "Nifty 100",
```

```
"token": "26012"
k
{
"idxname": "Nifty Midcap 50",
"token": "26014"
k
{
"idxname": "Nifty Realty",
"token": "26018"
k
1
}
```

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 IQ

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.
requestjime		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"
k
{
"bskt":"NSEBL",
"crt": "LTP"
k
{
"bskt":"NSEBL",
"crt" : "VALUE"
k
{
"bskt" : "NSEEQ",
"crt": "VOLUME"
k
{
"bskt":"NSEEQ",
"crt": "LTP"
k
{
"bskt":"NSEEQ",
"crt" : "VALUE"
k
{
"bskt":"NSEALL",
"crt": "VOLUME"
k
{
"bskt" : "NSEALL",
"crt" :" LTP"
k
{
"bskt":"NSEALL",
"crt":"VALUE"
}
]
}
Sample Failure Response :
{
"stat" : "Not_0k",
"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

```
# Here is a curl example (Q

curl --location 'https://BaseURL/TopLisf \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "exch": "NSE",
    "tb": "T",
    "bskt": "NSEALL",
    "crt": "LTP"
}&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
tb*	T or B	Top or Bottom
bskt*		bskt
crt*		criteria

Sample Success Response :

[
{
"stat" : "Ok",
"request_time": "15:44:45 03-06-2020",
"values":[
{
"tsym": "AIRAN-EQ",
"lp": "950.00",
"c":"9i5.00",
"v": "42705",
"value": "40185405.00",
"oi":"0",
"Pc": "3.83"
k
{
"tsym": "SHRENIK-EQ",
"lp": "1850.00",
"c": "1785.00",
"V": "206846",
"value": "368806418.00",
"oi":"0",
"Pc": "3.64"
},
{
"tsyrn": "REMSONSIND-EQ",
"lp": "6000.00",
"C":" 5795.00",
"v" :"3948",
"value": "22752324.00",
"Oi":"0",
"pc": "3. 54"
L
{
"tsym": "AXISNIFTY-EQ",
"lp": "106700.00",
"C": "103301.00",
"V":"422",
"value": "43825544.00",
"oi":"0",
"Pc":"3.29"
}
]
}
]

Sample Failure Response :

{
"stat": "Not_0k",
"emsg": "Invalid Input : Missing uid or exch or bskt or tb or crt"
}

RESPONSE DETAILS

Response data will be in json format with below fields.

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

TOP / BOTTOM CONTRACTS OBJECT :

Json Fields	Possible value	Description
tsym		Trading symbol
IP		LTP
c		Previous Close price
V		volume
value		Total traded value
oi		Open interest
PC		LTP percentage change

GET TIME PRICE DATA (CHART DATA)

To get Time Price Data (Chart data) you need to make a POST call to the following url :
https://piconnect.flattrade.in/PiConnectTP/TPSeries

Here is a curl example IQ

```
curl --location 'https://BaseURL/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 Name	Possible value	Description
jData*		Should send json object with fields in below list
jKey*		Key Obtained on login success.

Json Fields	Possible value	Description
uid*		Logged in User Id
exch*		Exchange
token*		
st		Start time (seconds since 1 jan 1970)
et		End Time (seconds since 1 jan 1970)
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.

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

Sample Success Response :

[
{
"stat":"0k",
"time":"02-06-2020 15:46:23",
"into":"0.00",
"inth":"0.00",
"inti": "0.00",
"intc": "0.00",
"intvwap":"0. 00",
"intv": "0",
"intoi" :"0",
"v" :"980515",
"oi" :"128702"
h
{
"stat": "0k",
"time": "02-06-2020 15:45:23",
"into": "0.00",
"inth": "0.00",
"inti": "0.00",
"intc": "0.00",
"intvwap":"0.00",

inti	Interval low
intc	Interval close
intvwap	Interval vwap
intv	Interval volume
v	volume
intoi	Interval io change
oi	oi

```
"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

```
# Here is a curl example IQ
curl --location 'https://BaseURL/EODChartData' \
--header 'Content-Type: application/json' \
--data 'jData={
    "sym": "NSE:RELIANCE-EQ",
    "from": "1624838400",
    "to": "1663718400"}&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
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
into		Interval open
inth		Interval high
inti		Interval low
intc		Interval close
ssboe		Date.Seconds in 1970 format
intv		Interval volume

```
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":"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.

Json Fields	Possible value	Description
uid*		Logged in User Id
tsym*		Trading symbol of any of the option or future. Option chain forthat 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
ent*		Number of strike to return on one side of the mid price for PUT and CALL, (example ent is 4, total 16 contracts will be returned, if ent is is 5 total 20 contract will be returned)

```
# Here is a curl example IQ

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

RESPONSE DETAILS

Response data will be in json format with below fields.

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Market watch success or failure indication.
values		Array of json objects, (object fields given in below table)
emsg		This will be present only in case of errors. That is : 1) Invalid Input 2) Session Expired

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

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 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
exd		Expiry Date
strprc		Strike Price
sptprc		Spot Price
int_rate		Init Rate
volatility		Volatility
optt		Option Type

```
# Here is a curl example IQ

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

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.
caLprice		Cal Price

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_price	Put Price
caLdelta	Cal Delta
put_delta	Put Delta
caLgamma	Cal Gamma
put_gamma	Put Gamma
caLtheta	Cal Theta
put_theta	Put Theta
caLrho	Cal Rho
put_rho	Put Rho
caLvego	Cal Vego
put_vego	Put Vego

```
"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  IQ

curl --location ’https://BaseURL/ExchMsg" \
--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 (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  IQ

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.

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

Sample Success Response :

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

SPAN CALCULATOR

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 IQ
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 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

```
# Here is a curl example IQ
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'
```

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.

Sample Success Response :

[
{
"request_time":"11:22:26 08-04-2021",
"stat":"Oi created",
"al_id":"21040800000004"
}
1

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.
aljd		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	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.
aljd*		Alert Id

RESPONSE DETAILS

Response data will have below fields.

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

```
# Here is a curl example IQ

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

Sample Success Response :

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

Sample Failure Response :

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

MODIFY ALERT

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

RESPONSE DETAILS

Response data will have below fields.

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

```
# Here is a curl example IQ

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"
}
]
```

Sample Failure Response :

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

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

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

```
# Here is a curl example IQ

curl --location 'https://BaseURL/GetPendingAlerf \
--header 'Content-Type: application/json' \
--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",
```

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

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

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 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		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)

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

```
--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"
}
```

REQUEST DETAILS

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

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

RESPONSE DETAILS

Response data will have below fields.

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

FUNDS PAYOUT REQUEST

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

```
# Here is a curl example IQ

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_0k",
"emsg":"Session Expired : Invalid Session Key"
}
```

REQUEST DETAILS

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

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

RESPONSE DETAILS

Response data will have below fields.

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

GET PAYIN REPORT

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

```
# Here is a curl example IQ

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":"20211250000001",
"tran_status":"Complete",
"amt":"10000.00"
L
(
"stat":"Ok",
"actid":"GURURAJ",
"trans_ref_num":"20211250000002",
"tran_status":"Complete",
"amt":"10000.00"
}
]
```

Sample Failure Response :

```
{
"stat":"Not_0k",
"emsg":"Session Expired : Invalid Session Key"
}
```

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
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)
tran_status	ADD_FUND_ST_COMPLETE	This is used to indicate the status of transaction

	_STR	
		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 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
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)
tran_status	WITHDRAW.ST.COMPLETE_STR	This is used to indicate the status of transaction
amt		Amount

```
# Here is a curl example IQ

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":"20211270000002",
"tran.status":"Complete",
"amt":"-1000.00"
L
{
"stat":"Ok",
"actid":"GURURAJ",
"transrefnum":"20211270000003",
"transtatus":"Complete",
"amt":"-100.00"
h
{
"stat":"Ok",
"actid":"GURURAJ",
"trans_ref_num":"20211270000004",
"tran.status":"Complete",
"amt":"-1000.00"
L
(
"stat":"Ok",
"actid":"GURURAJ",
"trans_ref_num":"20211270000005",
"tran.status":"Complete",
"amt":"-100.00"
}
]
```

Sample Failure Response :

```
{
"stat":"Not_Ok",
"emsg"/'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
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.

```
# Here is a curl example IQ

curl --location 'https://BaseURL/CancelPayout' \
--header 'Content-Type: application/json' \
--data 'j'Data={
    "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_0k",
"request_time":"18:58:47 12-05-2021",
"emsg"/'Error Occurred : -103 20211300000033 is Already Canceled"
}
```

WEB SOCKET API

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

```
# Here is a curl example IQ

curl --location 'wss://piconnect.flattrade. in/PiConnectWSTp/'
```

GENERAL GUIDELINES

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

CONNECT

REQUEST

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

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
IP		LTP
pc		Percentage change
V		volume
o		Open price
h		High price
1		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
IP		LTP
PC		Percentage change
V		volume
o		Open price
h		High price
1		Low price
c		Close price
ap		Average trade price

UNSUBSCRIBE TOUCHLINE

REQUEST

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

RESPONSE

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

```
# Here is a curl example  IQ

curl --location 'wss://piconnect.flattrade.in/PiConnectWSTp/'
--data '{
    f :  "uk",
    "k": "NSE|22#BSE|508123#NSE|10#BSE|2879"
}'
```

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

```
# Here is a curl example  IQ

curl --location 'wss://piconnect.flattrade.in/PiConnectWSTp/'
--data '{
    "t": "d",
    "k": "NSEI22#BSE|508123#NSE|10#BSE|2879"
}'
```

SUBSCRIPTION DEPTH ACKNOWLEDGEMENT

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

Json Fields	Possible value	Description
t	dk	'dk' represents depth acknowledgement
e	NSE, BSE, NFO ..	Exchange name
tk	22	Scrip Token
lP		LTP
PC		Percentage change
V		volume
o		Open price
h		High price
		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
bql		Best Buy Quantity 1
bq2		Best Buy Quantity 2
bq3		Best Buy Quantity 3
bq4		Best Buy Quantity 4
bq5		Best Buy Quantity 5
bpl		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
bol		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
sq4		Best Sell Quantity 4
sq5		Best Sell Quantity 5
spl		Best Sell Price 1
sp2		Best Sell Price 2
sp2		Best Sell Price 2
sp3		Best Sell Price 3
sp4		Best Sell Price 4

Sample Message :


```
{
"t": "df",
"e": "NSE",
"tk": "22",
"o": "1166.00",
"h": "1179.00",
"l": "1145.35",
'1152.65",
 "1159.74",
'819881",
: "120952",
: "131730",
: "1156.00",
: "1156.50",
: "1155.80",
: "1156.55",
: "1155.75",
: "1156.65",
: "1155.70",
: "1156.70",
: "1155.65",
: "1156.75",
: "4",
: "10",
: "67",
: "63",
: "83",
: "1",
: "139",
: "53",
: "393",
: "94"
}
```

sp5		Best Sell Price 5
sol		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
521		52 week high low in other exchanges. Lifetime 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
v		volume
o		Open price
h		High price
1		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
bql		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
bol		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
sql		Best Sell Quantity 1
sq2		Best Sell Quantity 2
sq2		Best Sell Quantity 2
sq3		Best Sell Quantity 3
sq4		Best Sell Quantity 4
sq5		Best Sell Quantity 5
spl		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
sol		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, Lifetime high low in mcx
521		52 week high low in other exchanges, Lifetime high low in mcx

UNSUBSCRIBE DEPTH

REQUEST

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

RESPONSE

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

```
# Here is a curl example IQ

curl --location 'wss://piconnect.flattrade.in/PiConnectWSTp/'
--data ■{
    "t": "ud",
    "k": "NSE122#BSE|508123#NSE110#BSE12879"
}'
```

SUBSCRIBE ORDER UPDATE

REQUEST

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

```
# Here is a curl example IQ

curl --location 'wss://piconnect.flattrade.in/PiConnectWSTp/1
--data '{
    "t": "o",
    "actid": "FZ00000"
}'
```

SUBSCRIPTION ACKNOWLEDGEMENT

Json Fields	Possible value	Description
t	ok	'ok' represents order update subscription acknowledgement

ORDER UPDATE SUBSCRIPTION UPDATES

Json Fields	Possible value	Description
t	om	'om' represents touchline feed
norenordno		Noren Order Number
uid		User Id
actid		Account ID
exch		Exchange
tsym		Trading symbol
qty*		Order Quantity [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)
trantype*	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
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.
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.
exchjm		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 IQ

curl --location 'wss://piconnect.flattrade. in/PiConnectWSTp/1
--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 IQ

curl --location 1https ://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.

Jsonfeilds	Possible value	Description
uid*		Logged in User Id

RESPONSE DETAILS

Response data will be in json format with below fields

Json Fields	Possible value	Description
stat	Ok or Not_Ok	User details success or failure indication.
exarr		Json array of strings with enabled exchange names
orarr		Json array of strings with enabled price types for user
prarr		Json array of Product Obj with enabled products, as defined below.
brkname		Broker id
brnchid		Branch id
email		
actid		
m_num		Mobile Number
uprev		Always it will be an INVESTOR, other types of user not allowed to login using this API.
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.

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

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

```
# Here is a curl example IQ

curl --location 'https ://BaseURL/SearchScrip' \
--header 'Content-Type: application/json' \
--data 'jData={
    "uid": "FZ00000",
    "stext": "NIFTY",
    "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.

Jsonfeilds	Possible value	Description
uid*		Logged in User Id
stext*		Search Text
exch		Exchange (Select from 'exarr' Array provided in User Details response)

RESPONSE DETAILS

Response data will be in json format with below fields

Json Fields	Possible value	Description
stat	OkorNot_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	Possible value	Description
exch	NSE, BSE, NFO ...	Exchange
tsym		Trading symbol of the scrip (contract)
token		Token of the scrip (contract)
pp		Price precision
ti		Tick size
ls		Lot size

Sample Success Response :

{
"stat": "Ok",
"values": [
{
"exch": "NSE",
"token": "18069",
"tsym": "REL100NAV-EQ"
k
{
"exch": "NSE",
"token": "24225",
"tsym": "RELAXO-EQ"
k
{
"exch": "NSE",
"token": "4327",
"tsym": "RELAXOFOOT-EQ"
k
{
"exch": "NSE",
"token": "18068",
"tsym": "RELBANKNAV-EQ"
k
{
"exch": "NSE",
"token": "2882",
"tsym": "RELCAPITAL-EQ"
k
{
"exch": "NSE",
"token": "18070",
"tsym": "RELCONSNAV-EQ"
k
{
"exch": "NSE",
"token": "18071",
"tsym": "RELDIVNAV-EQ"
k
{
"exch": "NSE",
"token": "18072",
"tsym": "RELGOLDNAV-EQ"
k
{
"exch": "NSE",
"token": "2885",
"tsym": "RELIANCE-EQ"
k
{
"exch": "NSE",
"token": "15068",
"tsym": "RELIGARE-EQ"
k
{
"exch": "NSE",
"token": "553",
"tsym": "RELINFRA-EQ"
k
{
"exch": "NSE",
"token": "18074",
"tsym": "RELNV20NAV-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 IQ

```
curl \
jData={"uid" : "FZ00000", "exch":" NSE",
"token" : "22"}&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
token		Contract Token

RESPONSE DETAILS

Response data will be in json format with below fields.

Sample Success Response :

{
"request_time" : "12:05:21 18-05-2021",
"stat" : "Ok"

Json Fields	Possible value	Description
stat	Ok or Not_Ok	Watch list update success or failure indication.
request_time		It will be present only in a successful response.
exch	NSE, BSE, NFO ...	Exchange
tsym		Trading Symbol
cname		Company Name
symname		Symbol Name
seg		Segment
instname		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
		Day Low Price
v		Volume
ltq		Last trade quantity
ltt		Last trade time
ltd	dd-mm-yy	Last Trade Date
bpl		Best Buy Price 1
spl		Best Sell Price 1
bp2		Best Buy Price 2
sp2		Best Sell Price 2
bp3		Best Buy Price 3
sp3		Best Sell Price 3
bp4		Best Buy Price 4
sp4		Best Sell Price 4
bp5		Best Buy Price 5
sp5		Best Sell Price 5
bq1		Best Buy Quantity 1
sq!		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
bol		Best Buy Orders 1
sol		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 reciving 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 Order Number
uid		User Id
actid		Account ID
exch		Exchange
tsym		Trading symbol
qty*		Order Quantity [If qty is junk value other than numbers].
pre*		Order Price [If pre is junk value other than numbers] "Order price cannot be zero" [if pretyp = 'MKT/ SL-MKT' with price 'O'].
prd		Product
status		Order status (New, Replaced, Complete,Rejected etc)
reporttype		Order event for which this message is sent out. (Fill, Rejected, Canceled)
trantype*	B / S	B -> BUY, S -> SELL [transtype should be 'B' or 'S' else reject].
pretyp		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
avgpre		Average fill price
fltm		Fill Time(present only when reporttype is Fill)
flid		Fill ID (present only when reporttype is Fill)
flqty		Fill Qty(present only when reporttype is Fill)
flpre		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].
trgpre		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
blpre		This will be present for cover and bracket parent order. This is the differential stop loss trigger price to be entered.
bppre		This will be present for bracket parent order. This is the differential profit price to be entered.
trailpre		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 Format: dd-mm-YYYY hh:MM:SS
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.
tm		Timestamp
kidid		Kid Id
sno_fillid		BO Sequence Id
checksum	sha256 [noren_order_num +noren_time_stamp+vendor_ key]	Checksum, (Make sure checksum matches to avoid any third party sending false order updates to your url endpoint)

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

SCRIP MASTER

Scrip Groups	
NSE - Equity	DOWNLOAD
NSE - Equity Derivatives	DOWNLOAD
NSE - Index Derivatives	DOWNLOAD
NSE - Currency Derivatives	DOWNLOAD
MCX - Commodity	DOWNLOAD
BSE - Equity	DOWNLOAD
BSE - Index Derivatives	DOWNLOAD
BSE - Equity Derivatives	DOWNLOAD

API RATE LIMIT

Time Frame	Rate Limit
Per Second	40
Per Minute	200