

NSE Clearing Limited

Circular

| Department: FUTURES AND OPTIONS SEGMENT | |
|---|-------------------------------|
| Download Ref No: NCL/CMPT/66073 | Date: January 10, 2025 |
| Circular Ref. No: 007/2025 | |

All Members,

Sub: NCMS FO API Version 2.0 updated- Technical Specification

This is further to NCL/CMPT/63140 dated July 30, 2024, regarding version 2.0 API Facility in NCMS FO.

Kindly find attached the revised technical specification document as Annexure 1. The revisions are highlighted for ease of reference.

Members are requested to do testing for the same.

**For and on behalf of
NSE Clearing Limited**

Nisha Pillai
Vice President

| Telephone No | Email id |
|----------------------------|-------------------------|
| 18002660050 (IVR Option 2) | msd_clearing@nscl.co.in |

API Documentation for NCMS FO

Version 2.1

December 2024



The NSE Clearing Limited,
Exchange Plaza, C-1, Block G,
Bandra Kurla Complex,
Bandra (E)
Mumbai - 400 051

Statement of Confidentiality

This document contains information that is proprietary and confidential to NSE, which shall not be disclosed outside NSE, transmitted, or duplicated, used in whole or in part for any purpose other than its intended purpose. Any use or disclosure in whole or in part of this information without explicit written permission of NSE is prohibited.

© Copyright, NSE

Table of Contents

| | | |
|------|---|----|
| 1 | Background | 4 |
| 2 | Data Flow Diagram..... | 5 |
| 3 | Technology Specification | 5 |
| 4 | API Registration | 5 |
| 5 | Log-In Workflow | 6 |
| | Note: Field expires_in is the value in milliseconds for token expiry and it will be decrease in every request after time passes. | 7 |
| 6 | Request/Response Structure (JSON) | 7 |
| 6.1 | Combined Trades & Actions Inquiry (ALLTRDACT TMTRDACT CPTRDACT ERRORACT)..... | 7 |
| 6.2 | Approval/Rejection..... | 15 |
| 6.3 | CP Modification | 18 |
| 6.4 | Approve All | 22 |
| 7 | Reference Codes | 23 |
| 7.1 | Market Type | 23 |
| 7.2 | Market Status | 24 |
| 7.3 | Transaction Code..... | 24 |
| 7.4 | Activity Type | 24 |
| 7.5 | Book Type | 24 |
| 7.6 | Client Type..... | 25 |
| 7.7 | Buy Sell Flag..... | 25 |
| 7.8 | Trade Status..... | 25 |
| 7.9 | Option Type | 25 |
| 7.10 | Is Approval Flag..... | 25 |
| 7.11 | Action Type..... | 26 |
| 7.12 | Exchange Code..... | 26 |
| 7.13 | Clearing Corporation ID | 26 |
| 8 | Response Codes..... | 26 |

| | | |
|-----|--|----|
| 8.1 | HTTP response code | 27 |
| 8.2 | Message based response code..... | 28 |
| 8.3 | Sample example for success or failure code | 29 |
| 8.4 | Async response code | 30 |
| 9 | Contingency | 31 |
| 10 | Usage Guidelines | 31 |

Version Control

| Version | Date | Description |
|---------|-----------------------------|---|
| 1.0 | 15 th Dec 2022 | V1.0 |
| 1.1 | 24 th Feb 2023 | Changed token generation URL for production |
| 1.2 | 6 th Feb 2023 | CP Trade filter added |
| 1.3 | 29 th Aug 2023 | Action API Download Enhancement – Additional Fields added |
| 1.4 | 7 th Sept 2023 | NCMS FO API Phase 2 Enhancements |
| 1.5 | 15 th April 2024 | NCMS FO API updated message method, changed the search filter, changed the response of cp-modification and approval and rejection |
| 1.6 | 24 th April 2024 | Added new API to provide “Approve All” request. Added 0001 error code. |
| 2.0 | 12th Jun 2024 | <ul style="list-style-type: none"> • “Transcodes” section has been renamed to “Reference Codes”. • Added new API “Combined Trades & Actions Inquiry (ALL/TMTRADES/CPTRADES)” to download Trades & Actions in same request. • New Action Types added. • Subsection “Clearing Corporation ID” added under “Reference Codes” • Existing Trade Inquiry and Action Inquiry is being discontinued. |
| 2.1 | 27 th Dec 2024 | <ul style="list-style-type: none"> • New filter ERRORACT added in /trd-act-inquiry to download all the actions which have errors. • Description of transcode 9001 |

1 Background

Currently trading system transfers online trades data to NSE Clearing Management System (NCMS) server.

NCMS client application residing at the members end sends periodical request to pull the data from the server. Maximum 'N' number of records (parameterized at server) are sent to the client application for each request. Currently trade data available on the NCMS server is accessible only to the NCMS client application.

It is proposed to expose an API to our members for trade and actions (any modifications performed on trades) inquiry for FO segment.

This document covers the technical specifications for various operations involved at both NCL as well as at member end.

- Following operations aspects are covered in this document:

| # | Operation | Endpoints | Purpose |
|-----|---------------------------------|-----------------------------|--|
| 5.0 | Login [Handshake] | /token | To authenticate the client |
| 6.1 | Combined Trade & Action Inquiry | /ncms-fo/trd-act-inquiry | To disseminate trades & actions information data |
| 6.2 | Approval Rejection | /ncms-fo/approval-rejection | Endpoint for approval/rejection of trades |
| 6.3 | CP Modification | /ncms-fo/cp-modification | Endpoint for cp modification of trades |
| 6.4 | Approve All | /ncms-fo/approve-all | Endpoint for approval of all trades |

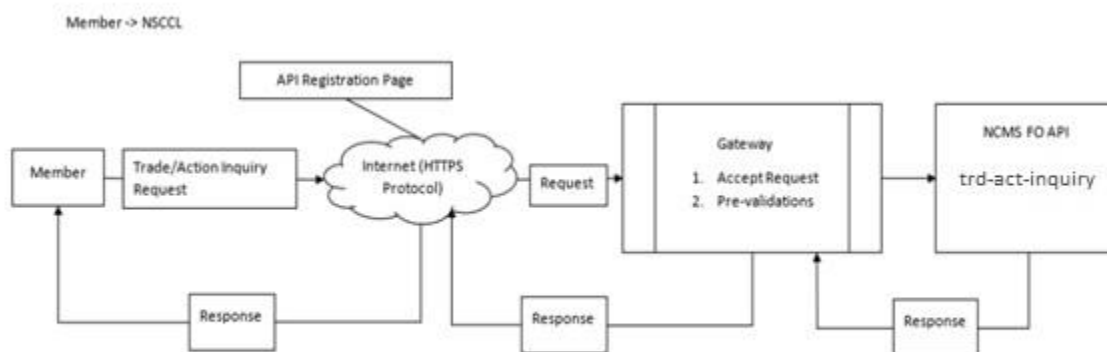
- Technical Specifications
- Log-in Workflow
- Message Structures

2 Data Flow Diagram

→ API Registration

- Members can register by providing the member code, member name, email address and IP from which they will be accessing the IP.
- On successful registration, an email will be sent to the email address provided with the Consumer Key and Consumer Secret.

→ API Call Workflow



3 Technology Specification

- Communication Protocol: HTTPS over internet.
- Request/Response Exchange Format: JSON (JavaScript Object Notation).
- Data Format: CSV (Comma Separated Values).
- Security Framework: Security Framework supports OAuth 2.0 specifications.

4 API Registration

- Member will need to provide information as described below for registration:
 - Member Code
 - Member Name
 - IP Address
 - Email Address
- Once this information is provided, the member specific Consumer Key and Consumer Secret will be generated and sent to the registered email address.
- Once the member receives the Consumer Key and Consumer Secret, they can start using the API.

5 Log-In Workflow

Requesting a “Token”

A consumer application needs to send a HTTPS POST request to the following URLs:

UAT: <https://www.devconnect2nse.com/token>

Production/LIVE: <https://www.connect2nse.com/ncms-token>

➔ Sample Request

```
POST /auth/token HTTP/1.1
Host: www.connect2nse.com

Content-Type: application/x-www-form-urlencoded
Authorization: Basic aGRmYzpoZGZjc2VjcmV0
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3

grant_type=client_credentials
```

➔ Response Structure

| API AUTHENTICATION REQUEST STRUCTURE (GET TOKEN) | | | | |
|--|----------------|-----------|--|-------------------------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 1 | Authorization | String | Will be of format: Basic <member_credentials> Where, member_credentials is a base64 encoding of the following data: cons_key:cons_secret | Basic aGRmYzpoZGZjc2VjcmV0 |
| 2 | nonce | String | An N-once value, that uniquely identifies each request sent to server. Has to be a base64 encoding of the following data: ddMMyyyyHHmsssSSS:<6 -digit random number> | MjAwMTIwMTcxNjEyMjE1OTE6 6 |
| 3 | grant_type | String | Value MUST be set to "client_credentials". | client_credentials |

➔ Sample Response


```

HTTP/1.1 200 OK
Content-Type: application/json
Pragma: no-cache{
  "access_token": "ee1073de-45d0-4040-b9c2-eddfa80280c0",
  "token_type": "bearer",
  "expires_in": "3600",
  "scope": "api_scope"
}

```

Note: Field expires_in is the value in milliseconds for token expiry and it will be decrease in every request after time passes.

6 Request/Response Structure (JSON)

As part of API Specifications Version 2.0 new end point for Combined Trades & Actions download is being introduced. Kindly refer to sub-section 6.1 below for the specifications of the same.

Kindly note that the following endpoints as available in Ver. 1.6 will be discontinued after Version 2.0 is rolled out,

UAT: <https://www.devconnect2nse.com/ncms-fo/trades-inquiry>

Production/LIVE: <https://www.connect2nse.com/ncms-fo/trades-inquiry>

UAT: <https://www.devconnect2nse.com/ncms-fo/actions-inquiry>

Production/LIVE: <https://www.connect2nse.com/ncms-fo/actions-inquiry>

6.1 Combined Trades & Actions Inquiry (ALLTRDACT|TMTRDACT|CPTRDACT|ERRORACT)

UAT: <https://www.devconnect2nse.com/ncms-fo/trd-act-inquiry>

Production/LIVE: <https://www.connect2nse.com/ncms-fo/trd-act-inquiry>

| Request Data Payload (JSON) | | | | |
|-----------------------------|----------------|-----------|---|--------------------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 1 | version | String | API version | 1.0 |
| 2 | data.msgld | String | Unique request number for the each request <CODE><YYYYMMDD><nnnnnnn> | 000012013101400 00001 |

| Request Data Payload (JSON) | | | | |
|-----------------------------|--------------------|-----------|--|---------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| | | | <ul style="list-style-type: none"> MEMBERCODE – Member code (Length : 5) YYYYMMDD – Date format nnnnnnn – Sequence no. starting from one i.e. For first request of the day, it should be (0000001). | |
| 2 | data.dataformat | String | Request data format : Response data format | CSV:CSV |
| 3 | data.trdactInquiry | CSV | Data Structure specified below | 0,ALLTRDACT,, |

| Trade + Action Inquiry Request Packet Structure (data.trdactInquiry) | | | | | |
|--|--|-----------|-----------------|---|--|
| Field Name | Description | Data Type | Size (in bytes) | Sample | |
| seqNo | Sequence till where server had sent the information in the previous request. For first download request of the day, it should be 0. | long | 8 | 0 | |
| srchFilter | Search Filter | String | 50 | <ul style="list-style-type: none"> ALLTRDACT – All Trades & Actions TMTRDACT – Clearing Member to view trades & actions as a Trading Member (Only applicable to Clearing Members) CPTRDACT – Clearing Member to view only CP Trades & Actions (Only applicable to Clearing Members) ERRORACT-All the failure action performed | |
| fill1 | Filler | String | 10 | | |
| fill2 | Filler | String | 10 | | |

| Sample request call (ALLTRDACT) |
|--|
| <pre>POST /api/ncms-fo/trade-action-inquiry HTTP/1.1 Host: www.connect2nse.com Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151 nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3 { "version": "1.0", "data": { "msgId": "00001201310140000001", "dataFormat": "CSV:CSV",</pre> |

| |
|---|
| <pre>"trdactInquiry": "0,ALLTRDACT,,", } }</pre> |
| <p>Sample request call (TMTRDACT Trades & Actions)</p> <pre>POST /api/ncms-fo/trd-act-inquiry HTTP/1.1 Host: www.connect2nse.com Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151 nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3 { "version": "1.0", "data": { "msgId": "00001201310140000001", "dataFormat": "CSV:CSV", " trdactInquiry": "0,TMTRDACT,,", } }</pre> <p><u>NOTE:</u> This filter is applicable to Clearing Members that want to view only the trades & actions that they have performed as a Trading Member.</p> |
| <p>Sample request call (CPTRDACT Trades & Actions)</p> <pre>POST /api/ncms-fo/trd-acts-inquiry HTTP/1.1 Host: www.connect2nse.com Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151 nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3 { "version": "1.0", "data": { "msgId": "00001201310140000001", "dataFormat": "CSV:CSV", " trdactInquiry": "0,CPTRDACT,,", } }</pre> <p><u>NOTE:</u> This filter is applicable to Clearing Members that want to view CP trades and actions only.</p> |
| <p>Sample request call (ERRORACT Actions)</p> <pre>POST /api/ncms-fo/trd-acts-inquiry HTTP/1.1 Host: www.connect2nse.com Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151 nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3 { "version": "1.0", "data": { "msgId": "00001201310140000001", "dataFormat": "CSV:CSV", " trdactInquiry": "0,ERRORACT,,", } }</pre> <p><u>NOTE:</u> This filter is applicable to identify the failure actions only.</p> |

➔ Response Structure

| Response Data Payload (JSON) | | | | |
|------------------------------|--------------------|-----------|--|--------------------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 1 | status | String | Response status | success/error |
| 2 | messages.code | String | Refer Section "Message based response code". | 01010000 |
| 3 | data.msgId | String | Unique request number sent in request. | 000012013101400000 01 |
| 4 | data.trdactInquiry | CSV | Data Structure specified below | Refer Sample Response. |

| Trade + Action Inquiry Response Packet Structure (data.trdactInquiry) | | | | |
|---|--|--------------------------------|--|------------------------------|
| Field Name | Description | Data Type | Size (in bytes) | Remarks |
| Control Record | | | | |
| sysinfoResData | System Info Response Structure | System Info Response Structure | Size of (System Info Response Structure) | Structure details give below |
| trdActResData | Trades + Actions Response Data Structure | Response Data Structure | Size of Response Data Structure | Structure details give below |

➔ Trade + Action Response Data Structure

| Trades Response Data Structure (ALLTRDACT TMTRDACT CPTRDACT) | | | | |
|--|--------------------------------------|--------------------------|------------------------------------|-------------------------------|
| Field Name | Description | Data Type | Size (in bytes) | Sample/Remarks |
| maxSeqNo | Max sequence number sent in response | long | 8 | 523764 |
| noOfRec | Count of trades sent in the response | int | 4 | 5 |
| Data Records | | | | |
| tradeActionOutput | Array Of Trade Structure | Array Of Trade Structure | Size of (Array Of Trade Structure) | Structure details given below |

| Array Of Trade Structure (tradeActionOutput ALLTRDACT TMTRDACT CPTRDACT) | | | | | |
|--|------------|------------------------|-----------|-----------------|---------|
| Sr No. | Field Name | Description | Data Type | Size (in bytes) | Sample |
| 1 | seqNo | Unique Sequence Number | long | 8 | 2014127 |

| Array Of Trade Structure (tradeActionOutput ALLTRDACT TMTRDACT CPTRDACT) | | | | | |
|--|-------------|--|-----------|-----------------|---------------------------------------|
| Sr No. | Field Name | Description | Data Type | Size (in bytes) | Sample |
| 2 | mkt | Market Type. Refer Section "Reference Codes" | string | 1 | 1 |
| 3 | trdNo | Trade Number | long | 8 | 91114327 |
| 4 | trdTm | Trade Time in Jiffy Format | long | 8 | 87841606323214 |
| 5 | tkn | Token | int | 4 | 42401 |
| 6 | trdQty | Trade Quantity | Int | 4 | 200 |
| 7 | trdPrc | Trade Price in paise | int | 4 | 1295 |
| 8 | bsFlg | Buy Sell Flag. Refer Section "Reference Codes" | string | 1 | 2 |
| 9 | ordNo | Order Number | double | 8 | 1100000000435542 |
| 10 | brnCd | Branch Code | int | 4 | 3 |
| 11 | usrId | User Id | int | 4 | 31908 |
| 12 | proCli | Client Type. Refer Section "Reference Codes" | short | 2 | 2 |
| 13 | cliActNo | Client account number | string | 20 | 07714 |
| 14 | cpCd | Custodial participant Id | string | 12 | 07714 |
| 15 | remarks | Remarks | string | 25 | |
| 16 | actTyp | Activity Type. Refer Section "Reference Codes" | short | 2 | 2 |
| 17 | TCd | Transaction Code. Refer Section "Reference Codes" | short | 2 | 6001 |
| 18 | ordTm | Order Time in milliseconds from 1980 | long | 8 | 1340356541 |
| 19 | booktype | Book Type. Refer Section "Reference Codes" | short | 2 | 1 |
| 20 | oppTmCd | Opposite Broker Id | string | 1 | |
| 21 | ctclld | CTCL code | double | 8 | 110005129066000 |
| 22 | status | Trade Status. Refer Section "Reference Codes" | string | 1 | P |
| 23 | TmCd | Member Code | string | 5 | 07714 |
| 24 | sym | Symbol | string | 10 | NIFTY |
| 25 | ser | Series | string | 2 | |
| 26 | inst | Instrument | string | 6 | OPTIDX |
| 27 | expDt | Expiry Date (in milliseconds from 1980) | int | 4 | 1340461800 |
| 28 | strPrc | Strike Price in paise | int | 4 | 1510000 |
| 29 | optType | Option Type for Option Contract. Refer Section "Reference Codes" | string | 2 | PE |
| 30 | exchangeID | Exchange Code. Refer section "Exchange Code" | int | 2 | 1 |
| 31 | tradeUniqID | Trade Unique ID | long | 20 | 42401911143271 |
| 32 | errCd | Action Response error code | short | 2 | Refer Section "Async Response codes". |
| 33 | actDtTm | Action Date Time | int | 4 | Date time in milliseconds from 1980 |
| 34 | actId | Action Type | short | 2 | Refer Section "Reference Codes" |

| Array Of Trade Structure (tradeActionOutput ALLTRDACT TMTRDACT CPTRDACT) | | | | | |
|--|------------|---|-----------|-----------------|-------------------------------------|
| Sr No. | Field Name | Description | Data Type | Size (in bytes) | Sample |
| 35 | trdTime | Trade time | int | 4 | Date time in milliseconds from 1980 |
| 36 | cmCd | Primary Member Code of the Clearing Member | string | 5 | 07714 |
| 37 | ccld | Clearing Corporation ID for Clearing Member of Trading Member | short | 2 | Refer Section "CC IDs" |

Note: 1. In addition to Original Trade (Transaction Code: 6001), all other actions including Approval-rejection and CP trade actions will be available as part of the trade & action inquiry response(Transaction Code: 9001). For list of transaction codes, kindly refer to the sub-section "Transaction Code" in the "Reference Codes" section.

2. Following fields will be populated blank for **transcode 9001**:

TA_ACCNO, TA_GIVEUP

3. Following fields will be populated zero for **transcode 9001**:

TA_EXPDT, TA_SPRC, TA_MKT, TA_ACTTYPE, TA_ORDERNO, TA_PRO_CLI, TA_DEALERID, TA_ORD_TIME, TA_BOOK_TYPE, TA_NNF, TA_BRNCD

Please refer above fields from original trade.

| Trades Response Data Structure(ERRORACT) | | | | |
|--|--------------------------------------|--------------------------|------------------------------------|-------------------------------|
| Field Name | Description | Data Type | Size (in bytes) | Sample/Remarks |
| maxSeqNo | Max sequence number sent in response | long | 8 | 523764 |
| noOfRec | Count of trades sent in the response | int | 4 | 5 |
| Data Records | | | | |
| tradeActionOutput | Array Of Trade Structure | Array Of Trade Structure | Size of (Array Of Trade Structure) | Structure details given below |

| Array Of Trade Structure (tradeActionOutput ERRORACT) | | | | | |
|---|------------|--|-----------|-----------------|-----------|
| Sr No. | Field Name | Description | Data Type | Size (in bytes) | Sample |
| 1 | seqNo | Unique Sequence Number | long | 8 | 145760 |
| 2 | mkt | Market Type. Refer Section "Reference Codes" | string | 1 | |
| 3 | trdNo | Trade Number | long | 8 | 630143511 |
| 4 | trdTm | Trade Time in Jiffy Format | long | 8 | 0 |
| 5 | tkn | Token | int | 4 | 44998 |

| Array Of Trade Structure (tradeActionOutput ERRORACT) | | | | | |
|---|-------------|--|-----------|-----------------|---------------------------------------|
| Sr No. | Field Name | Description | Data Type | Size (in bytes) | Sample |
| 6 | trdQty | Trade Quantity | Int | 4 | 60 |
| 7 | trdPrc | Trade Price in paise | int | 4 | 22500 |
| 8 | bsFlg | Buy Sell Flag. Refer Section "Reference Codes" | string | 1 | 1 |
| 9 | ordNo | Order Number | double | 8 | 0 |
| 10 | brnCd | Branch Code | int | 4 | 0 |
| 11 | usrId | User Id | int | 4 | 0 |
| 12 | proCli | Client Type. Refer Section "Reference Codes" | short | 2 | 0 |
| 13 | cliActNo | Client account number | string | 20 | |
| 14 | cpCd | Custodial participant Id | string | 12 | Z4760126I |
| 15 | remarks | Remarks | string | 25 | |
| 16 | actTyp | Activity Type. Refer Section "Reference Codes" | short | 2 | 0 |
| 17 | TCd | Transaction Code. Refer Section "Reference Codes" | short | 2 | 0 |
| 18 | ordTm | Order Time in milliseconds from 1980 | long | 8 | 0 |
| 19 | booktype | Book Type. Refer Section "Reference Codes" | short | 2 | 0 |
| 20 | oppTmCd | Opposite Broker Id | string | 1 | |
| 21 | ctclId | CTCL code | double | 8 | |
| 22 | status | Trade Status. Refer Section "Reference Codes" | string | 1 | |
| 23 | TmCd | Member Code | string | 5 | 90084 |
| 24 | sym | Symbol | string | 10 | |
| 25 | ser | Series | string | 2 | |
| 26 | inst | Instrument | string | 6 | |
| 27 | expDt | Expiry Date (in milliseconds from 1980) | int | 4 | 0 |
| 28 | strPrc | Strike Price in paise | int | 4 | 0 |
| 29 | optType | Option Type for Option Contract. Refer Section "Reference Codes" | string | 2 | |
| 30 | exchangeID | Exchange Code. Refer section "Exchange Code" | int | 2 | 1 |
| 31 | tradeUniqID | Trade Unique ID | long | 20 | 449986301435111 |
| 32 | errCd | Action Response error code | short | 2 | Refer Section "Async Response codes". |
| 33 | actDtTm | Action Date Time | int | 4 | Date time in milliseconds from 1980 |
| 34 | actId | Action Type | short | 2 | Refer Section "Reference Codes" |
| 35 | trdTime | Trade time | int | 4 | Date time in milliseconds from 1980 |
| 36 | cmCd | Primary Member Code of the Clearing Member | string | 5 | 90084 |

| Array Of Trade Structure (tradeActionOutput ERRORACT) | | | | | |
|---|------------|---|-----------|-----------------|------------------------|
| Sr No. | Field Name | Description | Data Type | Size (in bytes) | Sample |
| 37 | cclid | Clearing Corporation ID for Clearing Member of Trading Member | short | 2 | Refer Section "CC IDs" |
| 38 | msgId | Value of field data.msgId provided in request | string | 20 | 00001201310140000001 |

Note:

Following fields will be populated blank for **ERRORACT** filter:
Mkt, cliActNo, remarks, oppTmCd, ctclld, status, sym, ser, ser, optType

Following fields will be populated zero for **ERRORACT** filter:
trdTm, ordNo, brnCd, usrId, proCli, actTyp, TCd, ordTm, booktype, expDt, strPrc

➔ System Info Response Structure

| System Info Response Structure (sysinfoResData) | | | | |
|---|--|-----------|-----------------|----------|
| Field Name | Description | Data Type | Size (in bytes) | Sample |
| mktSts | Market Status Refer Section "Reference Codes" | Short | 2 | 1 |
| currTrdDate | Current Trade Date (YYYYMMDD) | long | 8 | 20220919 |
| sfill1 | Filler | String | 10 | |
| sfill2 | Filler | String | 10 | |

➔ Sample Response(ALLTRDACT|TMTRDACT|CPTRDACT)

Control Record Record Separator Trade Data Record Action Data Record

```
{
  "status": "success",
  "messages": {
    "code": "01010000"
  },
  "data": {
    "msgId": "00001201310140000001",
    "tradeActionInquiry": "1,20220919,,2513977,13197^523760,1,911143
27,91612702319650,42401,200,1295,2,2300000000056750,3,10000,2,CL
I1111,00001,,2,6001,1340356541,1,,400013021197777,P,00001,NIFTY,
,OPTIDX,1398522600,1510000,PE,1,42401911143271,0,1397898700,1,13
97898700,00001,1^523761,1,91114327,91612702319750,58945,400,1423
```



```

95,1,2300000000066157,3,10000,1,CPCLI001,CP0000000001,,2,6001,13
97898900,1,,400013021197777,P,00001,INFY,,FUTSTK,1398522600,-1,XX
X,1,589458830001411,0,1397898900,1,1397898900,00001,1523762,1,8
83000141,91612702319750,58945,400,142395,1,2300000000066157,10,3
1908,1,CLI001,CP0000000001,,2,,1397898900,1,,400013021196012,P,1
4665,INFY,,FUTSTK,1398522600,-1,XX,1,589458830001411,0,140196877
2,6,1397898900,14665,1523763,1,883000141,91612702319750,58945,4
00,142395,1,2300000000066157,10,31908,1,CLI001,CITI000000002,,101
,,1397898900,1,,400013021196012,P,14665,INFY,,FUTSTK,1398522600,
-1,XX,1,589458830001411,0,1401968772,8,1397898900,14665,1523764
,1,883000141,91612702319750,58945,400,142395,1,2300000000066157,
10,31908,1,CLI001,CITI000000002,,2,,1397898900,1,,400013021196012
,A,14665,INFY,,FUTSTK,1398522600,-1,XX,1,589458830001411,0,14019
68946,4,1397898900,14665,1"
}
}
}

```

➔ Sample Response(ERRORACT)

Control Record **Record Separator** **Failure action Record**

```

{
  "status": "success",
  "messages": {
    "code": "01010000"
  },
  "data": {
    "tradeActionInquiry": "0,20241113,,24774166,4327^
145760,,630143511,0,44998,60,22500,1,0,0,0,0,,Z4760126I,,0,0,0,0,,90
084,,,0,0,,1,449986301435111,0,1415954279,4,1411118140,90084,1^
145802,,630143529,0,44998,75,22500,1,0,0,0,0,,Z4760126I,,0,0,0,0,,90
084,,,0,0,,1,449986301435291,0,1415954279,4,1411118140,90084,1",
    "msgId": "90084202406030000005"
  }
}

```

6.2 Approval/Rejection

UAT: <https://www.devconnect2nse.com/ncms-fo/approval-rejection>

Production/LIVE: <https://www.connect2nse.com/ncms-fo/approval-rejection>

➔ Request Structure

| Request Data Payload (JSON) | | | | |
|-----------------------------|-----------------|-----------|---|---|
| Sr. No | Parameter Name | Data Type | Description | Sample Value |
| 1 | version | String | API version | 1.0 |
| 2 | data.msgId | String | Unique request number for the each request <CODE><YYYYMMDD><nnnnnnn> <ul style="list-style-type: none"> MEMBERCODE – Member code (Length : 5) YYYYMMDD – Date format nnnnnnn – Sequence no. starting from one i.e. For first request of the day, it should be (0000001). | 08081201310140000001 |
| 3 | data.isApproval | Char | Approval request or Rejection Request | Y – Approval N – Rejection |
| 4 | data.appRejData | JSON | Array of Approval or Rejection structure | Max 15000 records allowed per messageID. Structure details given below |

| Approval/Rejection Request Packet Structure | | | | |
|---|--|-----------|-----------------|-----------------|
| Field Name | Description | Data Type | Size (in bytes) | Remarks |
| seqNo | Sequence number of trade | long | 8 | 7760 |
| trdNo | Trade Number | long | 8 | 900000607 |
| bsFlag | Buy/Sell Flag. Refer section “Buy Sell Flag” | Short | 2 | 2 |
| uniqId | Trade Unique ID | String | 15 | 357009000006071 |

➔ Sample request call (Approval Request)

```
POST /api/ncms-fo/approval-rejection HTTP/1.1
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3

{
  "version": "1.0",
  "data": {
    "msgId": "08081201310140000001",
```

```
"isApproval": "Y",
"appRejData": [
  {
    "seqNo": 7760,
    "trdNo": 900000607,
    "bsFlag": 2,
    "uniqId": "357009000006071"
  },
  {
    "seqNo": 7767,
    "trdNo": 960000383,
    "bsFlag": 2,
    "uniqId": "351579600003831"
  }
]
}
```

➔ Sample request call (Rejection Request)

```
POST /api/ncms-fo/approval-rejection HTTP/1.1
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3
{
  "version": "1.0",
  "data": {
    "msgId": "08081201310140000001",
    "isApproval": "N",
    "appRejData": [
      {
        "seqNo": 7760,
        "trdNo": 900000607,
        "bsFlag": 2,
        "uniqId": "357009000006071"
      },
      {
        "seqNo": 7767,
        "trdNo": 960000383,
        "bsFlag": 2,
        "uniqId": "351579600003831"
      }
    ]
  }
}
```

➔ Response Structure

| Response Data Payload (JSON) | | | | |
|------------------------------|------------------|-----------|--|---------------------------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 1 | status | String | Response status | success/error |
| 2 | messages.success | String | message | Request submitted successfully. |
| 3 | data.code | String | Refer Section "Message based response code". | 01010000 |

➔ Sample Acknowledgement Response

```
{
  "status": "Success",
  "messages": {
    "success": " Request submitted successfully."
  },
  "data": {
    "code": "01010000"
  }
}
```

6.3 CP Modification

UAT: <https://www.devconnect2nse.com/ncms-fo/cp-modification>

Production/LIVE: <https://www.connect2nse.com/ncms-fo/cp-modification>

➔ Request Structure

| Request Data Payload (JSON) | | | | |
|-----------------------------|----------------|-----------|---|----------------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 1 | version | String | API version | 1.0 |
| 2 | data.msgId | String | Unique request number for the each request <CODE><YYYYMMDD><nnnnnnn> <ul style="list-style-type: none"> MEMBERCODE – Member code (Length : 5) YYYYMMDD – Date format | 08081201310140000001 |

| Request Data Payload (JSON) | | | | |
|-----------------------------|-----------------|-----------|---|---|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| | | | • nnnnnnn – Sequence no. starting from one i.e. For first request of the day, it should be (0000001). | |
| 4 | data. cpModData | JSON | Array of CP Modification structure | Max 15000 records allowed per messageID. Structure details given below |

| CP Modification Request Packet Structure | | | | |
|--|--|-----------|-----------------|------------------|
| Field Name | Description | Data Type | Size (in bytes) | Remarks |
| seqNo | Sequence number of trade | long | 8 | 421523 |
| orderNo | Order number | long | 12 | 2600000001529687 |
| trdNo | Trade Number | long | 8 | 960025162 |
| bsFlag | Buy/Sell Flag. Refer section "Buy Sell Flag" | Short | 2 | 1 |
| newCPCode | New CP code | String | 12 | AN074612I |
| oldCPCode | Old CP code | String | 12 | CITI00005680 |
| uniqId | Trade Unique ID | String | 15 | 1377619600251621 |

➔ Sample request call (cp trade to cp trade modification)

```
POST /api/ncms-fo/cp-modification HTTP/1.1
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3

{
  "version": "1.0",
  "data": {
```

```
"msgId": "08081202404180001016",
"dataFormat": "",
"cpModData": [
  {
    "seqNo": 421523,
    "orderNo": "2600000001529687",
    "trdNo": 960025162,
    "bsFlg": 1,
    "newCPCode": "AN074612I",
    "oldCPCode": "CITI00005680",
    "uniqId": "1377619600251621"
  }
]
```

➔ Sample request call (cp trade to client trade modification)

```
POST /api/ncms-fo/cp-modification HTTP/1.1
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3
```

```
{
  "version": "1.0",
  "data": {
    "msgId": "08081202404180001016",
    "dataFormat": "",
    "cpModData": [
      {
        "seqNo": 421523,
        "orderNo": "2600000001529687",
        "trdNo": 960025162,
        "bsFlg": 1,
        "newCPCode": "",
        "oldCPCode": "CITI00005680",
        "uniqId": "1377619600251621"
      }
    ]
  }
}
```

➔ Sample request call (client trade to cp trade modification)

```
POST /api/ncms-fo/cp-modification HTTP/1.1
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3
```

```
{
  "version": "1.0",
  "data": {
    "msgId": "08081202404180001016",
    "dataFormat": "",
    "cpModData": [
      {
        "seqNo": 421523,
        "orderNo": "2600000001529687",
        "trdNo": 960025162,
        "bsFlg": 1,
        "newCPCCode": "AN074612I",
        "oldCPCCode": "",
        "uniqId": "1377619600251621"
      }
    ]
  }
}
```

➔ Response Structure

| Response Data Payload (JSON) | | | | |
|------------------------------|------------------|-----------|--|--------------------------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 1 | status | String | Response status | success/error |
| 2 | messages.success | String | Message | Request submitted successfully |
| 3 | data.code | String | Refer Section "Message based response code". | 01010000 |

➔ Sample Acknowledgement Response

```
{
  "status": "Success",
  "messages": {
    "success": " Request submitted successfully."
  },
  "data": {
    "code": "01010000"
  }
}
```

6.4 Approve All

UAT: <https://www.devconnect2nse.com/ncms-fo/approve-all>

Production/LIVE: <https://www.connect2nse.com/ncms-fo/approve-all>

➔ Request Structure

| Request Data Payload (JSON) | | | | |
|-----------------------------|----------------|-----------|-------------|--------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 1 | version | String | API version | 1.0 |
| 2 | data | | | |

➔ Sample request call

```
POST /api/ncms-fo/approve-all HTTP/1.1
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3
{
  "version": "1.0",
  "data": {
    "msgId": "08081202404180001016",
    "memCode": "08081"
  }
}
```

➔ Response Structure

| Response Data Payload (JSON) | | | | |
|------------------------------|------------------|-----------|-----------------|--------------------------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 1 | status | String | Response status | success/error |
| 2 | messages.success | String | Message | Request submitted successfully |

| Response Data Payload (JSON) | | | | |
|------------------------------|----------------|-----------|--|--------------|
| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
| 3 | data.code | String | Refer Section "Message based response code". | 01010000 |

➔ Sample Acknowledgement Response

```
{
  "status": "Success",
  "messages": {
    "success": " Request submitted successfully."
  },
  "data": {
    "code": "01010000"
  }
}
```

- Workflow

- Trade/Action download works on sequence number basis present in individual trade/action response packet (*seqNo*).
- The sequence number signifies the sequence of events for a single trade/action lifecycle. Every event occurred with respect to a particular trade/action will have a new sequence number.
- On trades/actions download request, maximum trades/actions sequence number available should be sent. If there are no trades/actions present, the sequence number sent should be 0. API shall interpret the request and will fetch "n" number of trades/actions, whose trades/actions sequence number is greater than that sent by client. The fetched trades will be sent back to client in response.
- The trades/actions received by client in response packet are to be stored at client end. On subsequent trades/actions download request, client should again send the maximum trades/actions sequence number available with them that was returned as part of the control record in the previous request.

7 Reference Codes

7.1 Market Type

| Code | Description |
|------|-------------|
| 1 | Normal |
| 2 | Odd Lot |
| 3 | Spot |
| 4 | Auction |

| Code | Description |
|------|----------------|
| 5 | Call Auction 1 |
| 6 | Call Auction 2 |

7.2 Market Status

| Code | Description |
|------|-----------------------------|
| 1 | Preopen shutdown |
| 2 | Normal Market Preopen ended |
| 3 | Open Msg |
| 4 | Close Msg |
| 5 | Closing Start |
| 6 | Closing End |

7.3 Transaction Code

| Code | Description |
|------|---|
| 6001 | Original Trade |
| 5525 | Trade Modification Approval |
| 5565 | Control Trade Modification |
| 5520 | Trade Cancellation Approval |
| 5560 | Control Trade Cancellation |
| 5530 | Trade Cancellation Rejection |
| 5445 | Trade Modification (Client Modification) |
| 5440 | Trade Cancellation |
| 9001 | CP Actions (Approvals/Rejections/Modifications) |

7.4 Activity Type

| Code | Description |
|------|---|
| 2 | Original Trade |
| 7 | Trade Cancellation |
| 101 | Buy Participant modification |
| 102 | Sell Participant modification |
| 103 | Buy & Sell Participant modification |
| 104 | Quantity modification |
| 105 | Buy Account No. modification |
| 106 | Sell Account No. modification |
| 107 | Buy & Sell Account No. modification |
| 109 | Buy Trade Cancellation due to modification |
| 110 | Sell Participant Cancellation due to modification |
| 111 | Buy & Sell Trade Cancellation due to modification |

7.5 Book Type

| Code | Description |
|------|-------------|
| 1 | Regular Lot |

| Code | Description |
|------|------------------|
| 2 | Special Terms |
| 3 | Stop Loss / MIT |
| 4 | Negotiated Trade |
| 5 | Odd Lot |
| 6 | Spot |
| 7 | Auction |
| 11 | Call Auction 1 |
| 12 | Call Auction 2 |

7.6 Client Type

| Code | Description |
|------|-------------|
| 1 | Cli |
| 2 | Pro |

7.7 Buy Sell Flag

| Code | Description |
|------|-------------|
| 1 | BUY |
| 2 | SELL |

7.8 Trade Status

| Code | Description |
|------|-------------|
| P | Pending |
| R | Reject |
| A | Approve |

7.9 Option Type

| Code | Description |
|------|----------------------------|
| CA | Call American |
| PA | Put American |
| CE | Call European |
| PE | Put European |
| XX | Future Stocks/Future Index |
| FF | Future Stocks/Future Index |

7.10 Is Approval Flag

| Code | Description |
|------|-------------|
| 1 | Approve |
| 0 | Reject |

7.11 Action Type

| Code | Description |
|------|--|
| 1 | Original Trade |
| 2 | Buy SI Generated |
| 3 | Sell SI Generated |
| 4 | AppRej Buy Approval |
| 5 | AppRej Sell Approval |
| 6 | Buy Side CP Modification (Old CP) |
| 7 | Sell Side CP Modification (Old CP) |
| 8 | Buy Side CP Modification (New CP) |
| 9 | Sell Side CP Modification (New CP) |
| 14 | AppRej Buy Rejected |
| 15 | AppRej Sell Rejected |
| 16 | Buy SI Cancelled |
| 17 | Sell SI Cancelled |
| 20 | Approval/Rejection Window is closed |
| 22 | Approve All window is closed |
| 23 | Buy CM is not matching |
| 24 | Sell CM is not matching |
| 25 | AppRej action is already done |
| 26 | Approve All action is already done |
| 27 | Approve All process window is open |
| 28 | AppRej record is locked |
| 29 | Approve All process is already running |

7.12 Exchange Code

| Code | Description |
|------|-------------|
| 1 | NSE |
| 2 | BSE |
| 3 | MSE |

7.13 Clearing Corporation ID

| Code | Description |
|------|-------------|
| 1 | NCL |
| 2 | ICCL |
| 3 | MCCIL |

8 Response Codes

There can be two types of response codes

- HTTP response codes

- Message based response codes
- Async response codes

8.1 HTTP response code

- HTTP responses shall be generated during login with success or failure status
- HTTP response shall also be generated in case of any authentication/input validation failure of the message.

HTTP response codes are as follows:

| HTTP Response Codes | | | |
|---------------------|--|---|--------------------|
| Sr. No. | Reason | Meaning | HTTP Response Code |
| 1 | SUCCESS | Request was handled successfully | 200 |
| 2 | UNKNOWN_ERROR | Internal Server Error: Internal server error has occurred in our platform | 500 |
| 3 | SVC_UNAVAILABLE | The server is currently unable to handle the request due to a temporary overloading or maintenance of the server | 503 |
| 4 | METHOD_NOT_ALLOWED | Unsupported HTTP method: A request was made for a resource using a request method not supported by that resource (e.g. using POST instead of GET) | 405 |
| 5 | BAD REQUEST | PARAMETER_ABSENT – There’s a required parameter which is not present in the request | 400 |
| 6 | BAD REQUEST | DATA_INVALID – The data is not in correct format and not recognized by our system | 400 |
| 7 | BAD REQUEST | DATA_FORMAT_REJECTED – Unsupported Data format parameter value | 400 |
| 8 | UNAUTHORIZED: Failed to authenticate the request | CONSUMER_KEY_UNKNOWN – The provided Consumer Key (API key) is not registered in our system OR service is not registered | 401 |
| 9 | UNAUTHORIZED: Failed to authenticate the request | TOKEN_INVALID – The provided token is not registered in our system | 401 |
| 10 | UNAUTHORIZED: Failed to authenticate the request | UNAUTHORIZED: <ul style="list-style-type: none"> • Unauthorized requestor IP address • API access disabled | 401 |
| 11 | PERMISSION_DENIED | Subscriber has temporarily disallowed access to his private data | 403 |
| 12 | The requested URL was not found | The requested URL was not found | 404 |

| HTTP Response Codes | | | |
|---------------------|-------------------|------------------------------|--------------------|
| Sr. No. | Reason | Meaning | HTTP Response Code |
| 13 | REQUEST_NOT_FOUND | Registered request not found | 570 |

8.2 Message based response code

- Message based response code shall be populated in the field “code” of the JSON response message
- It shall be of below format
 - First four characters (Field Identifier): refers to specific field or the entire message
 - Next characters (Validation code): refers to specific validation failure or success. Success code shall be populated only on successful acceptance of the message.

Field Identifier is as follows:

| Sr. No. | Module | Field Name | Field Identifier |
|---------|----------------------|------------|------------------|
| 1 | Entire Message | NA | 0101 |
| 2 | Input Data Parameter | msgId | 0102 |
| 3 | Input Data Parameter | isApproval | 0109 |
| 4 | Input Data Parameter | seqNo | 0107 |
| 5 | Input Data Parameter | srchFilter | 0108 |
| 6 | Input Data Parameter | noOfRec | 0110 |
| 7 | Input Data Parameter | appRejData | 0111 |
| 8 | Input Data Parameter | trdNo | 0112 |
| 9 | Input Data Parameter | bsFlag | 0113 |
| 10 | Input Data Parameter | uniqId | 0114 |

Validation codes are as follows:

| Sr. No. | Validation | Validation Type | Validation Code | Validation performed on Field |
|---------|----------------------------------|------------------|--|-------------------------------|
| 1 | Submitted to server successfully | Message Level | 0000 | Entire Message |
| 2 | Duplicate request received | Message Level | 0001 | Entire Message |
| 3 | All HTTP status codes | HTTP error codes | HTTP Response codes. Refer section “HTTP Response Code”. | Entire Message |

| Sr. No. | Validation | Validation Type | Validation Code | Validation performed on Field |
|---------|---|-----------------|-----------------|--|
| 4 | Mismatch in control and data record | Message Level | 0200 | Entire Message |
| 5 | Minimum Required Length | Generic | 0201 | msgId |
| 6 | Maximum Required Length | Generic | 0202 | msgId, max records count for CP modification and approval rejection |
| 7 | Mandatory field | Generic | 0204 | msgId, isApproval, noOfRec, seqNo, srchFilter, trdDate, appRejData, trdNo, seqNo, isApproval, uniqId, bsFlag, dataFormat |
| 8 | Data Format like Message Id / Date Format | Generic | 0206 | msgId, trdDate |
| 9 | Minimum allowed value | Generic | 0207 | seqNo, noOfRec, trdno, bsFlag |
| 10 | Maximum allowed value | Generic | 0208 | noOfRec, bsFlag |
| 11 | Invalid Value | Generic | 0209 | seqNo, isApproval, srchFilter, trdDate, trdno |
| 12 | Not Authorized Request | Generic | 0210 | App/Rej request by TM, CP mod request by PCM |
| 12 | System Error | Generic | 0241 | NA |
| 13 | Service Unavailable | Generic | 0242 | NA |
| 14 | Request Parsing Error: Invalid Request Structure | Generic | 0243 | NA |
| 15 | New cp code and old cp code equal while CP modification | Generic | 0244 | oldCPCode, newCPCpde |

8.3 Sample example for success or failure code

Example for Generic Error Code

Let's assume that msgId field holds value ABCD201340402132165, which turns out to be an error "Invalid Data Format". Error Code that will be generated is as shown below:

Field Identifier: 0102

Validation Code: 0206

code = combination of "Field Identifier" and "Validation Code" = 01020206

Example for Field Error Code

Let’s assume that seqNo field holds value -1, which turns out to be an error “Minimum allowed value”.

Error Code that will be generated is as shown below:

Field Identifier: 0107

Validation Code: 0207

code = combination of “Field Identifier” and “Validation Code” =01070207

Example for Success code (Submitted to server successfully)

Let’s assume that message for approval/rejection is successful, success code that will be generated is as shown below:

Field Identifier: 0101 (which is the identifier of the entire message)

Validation Code: 0000

code = combination of “Field Identifier” and “Validation Code” =01010000

Example for HTTP error code

Let’s assume that the invalid request scenario due to UNAUTHORIZED Access Request, error code that will be generated is as shown below:

Field Identifier: 0101 (which is the identifier of the entire message)

Validation Code: 401

code = combination of “Field Identifier” and “Validation Code” =0101401

Note:

For HTTP error code the above code will not be valid for every response. It will be valid only if there is an error in request header other than this it will populate HTTP code only.

8.4 Async response code

Async response code shall be populated in the field “errCd” of the message

| Error | Error Code |
|-------------------------|------------|
| Success | 0 |
| System in wrong state | 1 |
| Invalid Contract | 2 |
| Invalid Participant | 3 |
| Trade not found | 4 |
| Trade already cancelled | 5 |
| System Error | 6 |
| Trade already approved | 7 |
| Trade already rejected | 8 |
| Outstanding alert | 9 |
| Invalid user | 10 |

| Error | Error Code |
|--|------------|
| Invalid data | 11 |
| Clearing Member is in VC mode. Trade Approval/Rejection not allowed. | 12 |
| Clearing Member is Disabled. Trade Approval/Rejection not allowed. | 13 |
| Not Latest Trade | -12 |
| Approve All request rejected-Invalid market status | -19 |
| Invalid Seq No | -20 |
| Invalid Clearing Member | -21 |
| Invalid CP code | -22 |
| Invalid buy/sell flag | -23 |
| Invalid instrument | -24 |
| Invalid symbol | -25 |
| Invalid strike price | -26 |
| Invalid expiry date | -27 |
| Invalid option type | -28 |
| Invalid trade quantity | -29 |
| Invalid trade price | -30 |
| Invalid order number | -31 |
| Invalid trade number | -32 |
| Invalid broker id | -33 |
| Already submitted | -50 |
| Already approved | -51 |
| Already rejected | -52 |

9 Contingency

In case of any failure such as network, application, high bandwidth utilization at NSE or the MEMBER end, login workflow should be re-initiated.

10 Usage Guidelines

- a) Members should limit requests to 15 seconds between each request.
- b) Members can send requests to the API between 6:30 AM to 5 AM next day. Kindly note that NCMS services shall not be available between 5 AM to 6:30 AM due to maintenance activity.
- c) Failure to adhere to the above guidelines will result in removal of IP from whitelist which means that member will not be able to access the API until IP is re-added to the whitelist.