# API Documentation for Online Trade Inquiry Service

Version 2.1

NOVEMBER 2024

National Stock Exchange of India Ltd
Exchange Plaza, Plot No. C/1, G Block,
Bandra-Kurla Complex, Bandra (E)
Mumbai - 400 051.

# Notice

Confidential

## Table of Contents

| Version | Date | Description |
|---|---|---|
| 1.5 | 4th June 2019 | Added Usage Guidelines |
| 1.6 | 29th Nov 2019 | Updated Interoperability Changes |
| 1.7 | 29th May 2020 | Added filter for Clearing Member to view trades as a Trading Member |
| 1.8 | 09th May 2022 | Modification in Usage Guidelines |
| 1.9 | 05th Sept 2022 | NOTIS FO API Download enhancements |
| 2.0 | 15th March 2023 | Added note for nonce header validation |
| 2.1 | 27th Nov 2024 | Uniqueness in Action inquiry response |

# 1 Background

Currently trading system transfers online trades data to NSEIL Online Trade Inquiry System (NOTIS) server.

NOTIS 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 NOTIS server is accessible only to the NOTIS client application.

It is proposed to expose API's to our esteemed empanelled members for trade & actions (any modifications performed on trade) inquiry for F&O segment.

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

- Following operations aspects are covered in this document:

| Sr. No. | Operation | Endpoints | Purpose |
|---|---|---|---|
| 1 | Login [ Handshake ] | /token | To authenticate the client |
| 2 | Trade Inquiry | /inquiry-fo/trades-inquiry | To disseminate trades information data/Client Modifications |
| 3 | Trade Actions Inquiry | /inquiry-fo/actions-inquiry | Approval/Rejection/Approve All Confirmation/SI Download/CP Modification |

- Technical Specifications
- Log-in Work flow
- Message Structure

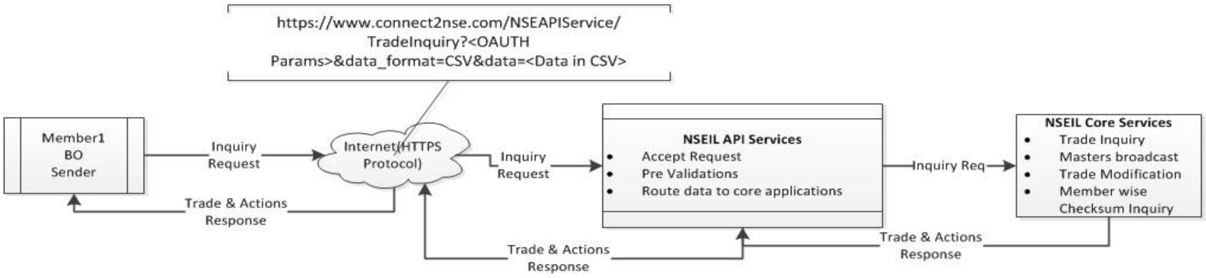
## 2 Data Flow Diagram

➢ API Registration

- For API registration, user need to request the NSE team to add entry for them in database.
- The entry for user will be added from backend manually and no frontend will be provided for this purpose.
- Once entry is added the user will be provided with username & password to access the token API in order to get token.
- Once token is obtained user can access the Trade Inquiry API using below mentioned diagram.

➢ API Call Work Flow

Member -> NSEIL



## 3 Technology Specification

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

Confidential

## 4 API Registration

- Members need to register through Member Portal → API Management.

- Member will need to provide information as described below:
    - Service → Trade Inquiry Service
    - Email Id →Email Address where the API credentials would be sent
    - IP Address → IP Address from which the member will communicate with NSEIL API Service

- Once this information is provided, admin at NSEIL will verify & generate Consumer Key & Consumer Secret. These values will be emailed to MEMBER using registered email Id.

- Once MEMBER receives Consumer Key & Consumer Secret, they can start using API.

## 5 Log-In Workflow

- Login Handshake (MEMBER → NSEIL)
- 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/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
```

➔ Request Structure

| API AUTHENTICATION REQUEST STRUCTURE (GET TOKEN) | | | | |
|---|---|---|---|---|
| **Sr. No.** | **Parameter Name** | **Data Type** | **Description** | **Sample Value** |
| 1 | Authorization | String | Will be of format:<br>Basic <member_credentials><br>Where, member_credentials is a base64 encoding of the following data:<br>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: ddMMyyyyHHmmssSSS:<6-digit random number> | MjAwMTIwMTcxNjEyMjE1OTE6<br>Note: A new unique nonce value will have to be sent for each subsequent request otherwise the member will face 401 error. |
| 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"
}
```

➔ Response Structure

**API AUTHENTICATION RESPONSE STRUCTURE (GET TOKEN)**

| Sr. No. | Parameter Name | Data Type | Description | Sample Value |
|---|---|---|---|---|
| 1 | access_token | String | The access token issued by the authorization server. | eyJhbGciOiJSUzI1NiIsInR5cCI |
| 2 | token_type | String | The type of the token issued | bearer |
| 3 | expires_in | int | The lifetime in seconds of the access token. For example, the value "3600" denotes that the access token will expire in one hour from the time the response was generated. | 3600 |
| 4 | scope | String | If identical to the scope requested by the client otherwise, REQUIRED. | api_scope |

**Note:**

a. The Access token is to be reused to access the NSE API Data till it expires.

b. An access token expires after ‘X’ minutes of inactivity.

Confidential

# 6 Request/Response Structure (JSON)

## 6.1 Trade Inquiry (ALL/TMTRADES Trades)

**UAT:** https://www.devconnect2nse.com/inquiry-fo/trades-inquiry

**Production/LIVE:** https://www.connect2nse.com/inquiry-fo/trades-inquiry

**Production Test URL for Enhancements**: https://www.connect2nse.com/inquiry-fo-2/trades-inquiry

https://www.connect2nse.com/inquiry-fo-2/actions-inquiry

NOTE: The following enhancements have been made available on the production test URL:

- Increase of number of records from 20000 to 100000 per request
- Sequence number range to decrease the number of member requests. For example, in case of no records returned in the first request/subsequent requests, we will provide the next sequence number from which the member should download the trades.
- No changes to existing member flow

➔ Sample request call (ALL Trades)

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

{
        "version": "1.0",
        "data": {
        "msgId": "00240201310140000001",
        "dataFormat": "CSV:CSV",
        "tradesInquiry": "0,ALL,,"
        }
}
```

➔ Sample request call (ALL Trades)

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

{
        "version": "1.0",

        "data": {
        "msgId": "00240201310140000001",
        "dataFormat": "CSV:CSV",
        "tradesInquiry": "0,TMTRADES,,"
        }
}
```

**NOTE**: This filter is only applicable to Clearing Members that want to view only the trades that they have performed as a Trading Member.

| 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<br><CODE><YYYYMMDD><nnnnnnn><br>• MEMBERCODE – Member code (Length : 5)<br>• YYYYMMDD – Date format<br>• nnnnnnn – Sequence no. starting from one i.e. For first request of the day, it should be (0000001). | 00240201310140000001 |
| 2 | data.dataformat | String | Request data format : Response data format | CSV:CSV |
| 3 | data.tradesInquiry | CSV | Data Structure specified below | 0,ALL,, |

| Trade Inquiry Request Packet Structure | | | | |
|---|---|---|---|---|
| **Field Name** | **Description** | **Data Type** | **Size (in bytes)** | **Sample** |
| seqNo | Trade Sequence number till where server had already sent data information In the previous request.<br>For first download request of the day, it should be 0. | long | 8 | 0 |
| srchFilter | Search Filter | String | 50 | • ALL – All Trades<br>• TMTRADES – Clearing Member to view trades as a Trading Member (Only applicable to Clearing Members) |
| fill1 | Filler | String | 10 | |
| fill2 | Filler | String | 10 | |

➔ Sample Response

🟩 → Control Record<br>
🟥 → Record Separator<br>
🟦 → Data Record(s)

```
{
"status": "success",
"messages": {
"code": "01010000"
},
"data": {
"msgId": "00240201310140000001",
```

Confidential

```
            "tradesInquiry":3,20120420,,,827587,3^827155,1,47964115,82609719241959,49102,800,267
05 ,2,1300000001946000,9,36956,1,SG18,SBISG0000081,,2,6001,1260524280,1,,40005100100400
0,P,08081,IBULHSGFIN,,FUTSTK,1261785600,-
1,XX,,,,,,,,^827201,1,89638308,82609719248043,49440,2200,4440,1,1500000001376851,9,8255
,1,ERROR,CITI00003543,,105,5445,1260523571,1,,400051001004012,P,08081,YESBANK,
,FUTSTK,1261837800,-
1,XX,,,,,,,,^827587,1,37527876,82609719325776,47859,250,403185,1,1200000001362256,9,364
11,1,KSAM12,CITI00003543,,2,6001,1260524281,1,,400051001002012,P,08081,BAJFINANCE,
,FUTSTK,1264809600,-1,XX,,,,,,,,

}
}
```

➔ 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. | 00240201310140000001 |
| 4 | data.tradesInquiry | CSV | Data Structure specified below | Refer Sample Response. |

| Trade Inquiry Response Packet Structure | | | | |
|---|---|---|---|---|
| **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 |
| trdResData | Trades Response Data Structure | Response Data Structure | Size of Response Data Structure | Structure details give below |

➔ System Info Response Structure

| System Info Response Structure | | | | |
|---|---|---|---|---|
| **Field Name** | **Description** | **Data Type** | **Size (in bytes)** | **Sample** |
| mktSts | Market Status<br>Refer Section “Transcodes”. | Short | 2 | 2 |
| currTrdDate | Current Trade Date (YYYYMMDD) | long | 8 | 20120420 |
| sfill1 | Filler | String | 10 | |
| sfill2 | Filler | String | 10 | |

Confidential

➔ Trade Response Data Structure

| Trades Response Data Structure | | | | |
|---|---|---|---|---|
| **Field Name** | **Description** | **Data Type** | **Size (in bytes)** | **Sample/Remarks** |
| maxSeqNo | Max sequence number sent in response | long | 8 | 827587 |
| noOfRec | Count of trades sent in the response | int | 4 | 3 |
| **Data Records** | | | | |
| tradesOutput | Array Of Trade Structure | Array Of Trade Structure | Size of (Array Of Trade Structure) | Structure details given below |

| | Array Of Trade Structure | | | | |
|---|---|---|---|---|---|
| **Sr No.** | **Field Name** | **Description** | **Data Type** | **Size (in bytes)** | **Sample** |
| 1 | seqNo | Unique Sequence Number | long | 8 | 827155 |
| 2 | mkt | Market Type. Refer Section “Transcodes” | String | 1 | 1 |
| 3 | trdNo | Trade Number | long | 8 | 47964115 |
| 4 | trdTm | Trade Time in Jiffy Format | long | 8 | 82609719241959 |
| 5 | tkn | Token | int | 4 | 49102 |
| 6 | trdQty | Trade Quantity | int | 4 | 800 |
| 7 | trdPrc | Trade Price in paise | int | 4 | 26705 |
| 8 | bsFlg | Buy Sell Flag. Refer Section “Transcodes” | String | 1 | 2 |
| 9 | ordNo | Order Number | double | 8 | 1300000001946000 |
| 10 | brnCd | Branch Code | int | 4 | 9 |

| | | | | | |
|---|---|---|---|---|---|
| 11 | usrId | User Id | int | 4 | 36956 |
| 12 | proCli | Client Type. Refer Section “Transcodes” | short | 2 | 1 |
| 13 | cliActNo | Client account number | String | 20 | SG18 |
| 14 | cpCd | Custodial participant Id | String | 12 | SBISG0000081 |
| 15 | remarks | Remarks | String | 25 | |
| 16 | actTyp | Activity Type. Refer Section “Transcodes” | short | 2 | 2 |
| 17 | TCd | Transaction Code. Refer Section “Transcodes” | short | 2 | 6001 |
| 18 | ordTm | Order Time in milliseconds from 1980 | long | 8 | 1260524280 |
| 19 | booktype | Book Type. Refer Section “Transcodes” | short | 2 | 1 |
| 20 | oppTmCd | Opposite Broker Id | String | 1 | |
| 21 | ctclId | CTCL code | double | 8 | 400051001004000 |
| 22 | status | Trade Status. Refer Section “Transcodes” | String | 1 | P |
| 23 | TmCd | Member Code | String | 5 | 08081 |
| 24 | sym | Symbol | String | 10 | IBULHSGFIN |
| 25 | ser | Series | String | 2 | |
| 26 | inst | Instrument | String | 6 | FUTSTK |
| 27 | expDt | Expiry Date (in milliseconds from 1980) | int | 4 | 1261785600 |
| 28 | strPrc | Strike Price in paise | int | 4 | -1 |
| 29 | optType | Option Type for Option Contract. Refer Section “Transcodes” | String | 2 | XX |
| 30 | fill1 | Filler | String | 10 | |
| 31 | fill2 | Filler | String | 10 | |
| 32 | fill3 | Filler | String | 10 | |
| 33 | fill4 | Filler | String | 10 | |
| 34 | fill5 | Filler | String | 10 | |

| | | | | | |
|---|---|---|---|---|---|
| 35 | fill6 | Filler | String | 10 | |
| 36 | fill7 | Filler | String | 10 | |
| 37 | fill8 | Filler | String | 10 | |

**Note:** Client modifications (Transaction Code: 5445) will be shown as part of the trade inquiry response.

### 6.2 Action Inquiry (Approval/Rejection/Approve ALL Response/CP Modification)

**UAT:** https://www.devconnect2nse.com/inquiry-fo/actions-inquiry

**Production/LIVE:** https://www.connect2nse.com/inquiry-fo/actions-inquiry

➔ Sample request call

```
GET /api/inquiry-fo/actions-inquiry HTTP/1.1
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3

{
  "version": "1.0",
  "data": {
    "msgId": " 06637202008280000001",
    "dataFormat": "CSV:CSV",
    "actionsInquiry": "0,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.msgId | String | Unique request number for the each request<br><CODE><YYYYMMDD><nnnnnnn><br>• MEMBERCODE – Member code (Length : 5)<br>• YYYYMMDD – Date format<br>• nnnnnnn – Sequence no. starting from one i.e. For first request of the day, it should be (0000001). | 06637202008280000001 |
| 3 | data.dataformat | String | Request data format : Response data format | CSV:CSV |
| 4 | data.actionsInquiry | CSV | Data Structure specified below | Refer “Sample Request” |

| Actions Download Request Packet Structure | | | | |
|---|---|---|---|---|
| **Field Name** | **Description** | **Data Type** | **Size (in bytes)** | **Remarks** |
| seqNo | Action Sequence number till where server had already sent data information In the previous request.<br>For first download request of the day, it should be 0. | long | 8 | 0 |
| srchFilter | Search Filter All (Default) | String | 50 | ALL |
| fill1 | Filler | String | 10 | |
| fill2 | Filler | String | 10 | |

➔ Sample Response

```
{
 "status": "success",
 "messages": {
   "code": "01010000"
 },
 "data": {
 "msgId": " 06637202008280000001",

 “actionsInquiry”:   0,20241023,,,26417,44^0,3863,2201,1416647701,5,ICICI0005163,43315^0,4726,3078,
 1416647701,4,ICICI0005163,43308^

}

}
```

➔ 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. | 06637202008280000001 |
| 4 | data.actionsInquiry | CSV | Data Structure specified below | Refer Sample Response |

Confidential

| Actions Download Response Packet Structure | | | | |
|---|---|---|---|---|
| **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 |
| actResData | Actions Response Data Structure | Response Data Structure | Size of Response Data Structure | Structure details give below |

➔ System Info Response Structure

| System Info Response Structure | | | | |
|---|---|---|---|---|
| **Field Name** | **Description** | **Data Type** | **Size (in bytes)** | **Sample** |
| mktSts | Market Status Refer Section "Transcodes". | Short | 2 | 0 |
| currTrdDate | Current Trade Date (YYYYMMDD) | long | 8 | 20241023 |
| sfill1 | Filler | String | 10 | |
| sfill2 | Filler | String | 10 | |

### ➔ Action Response Data Structure

| Actions Response Data Structure | | | | |
|---|---|---|---|---|
| **Field Name** | **Description** | **Data Type** | **Size (in bytes)** | **Sample** |
| maxSeqNo | Max sequence number sent in response | long | 8 | 26417 |
| noOfRec | Count of actions sent in response | int | 4 | 44 |
| **Data Records** | | | | |
| actionsOutput | Array Of AppRejAction Structure | Array Of Actions Structure | Size of (Array Of Actions Structure) | Structure details give below |

| AppRejAction Data Structure | | | | |
|---|---|---|---|---|
| **Field Name** | **Description** | **Data Type** | **Size (in bytes)** | **Sample** |
| errCd | Action Response error code | short | 2 | Refer Section “Async Response codes”. 0 |
| seqNo | Trade Sequence number | int | 4 | Sequence number of the trade for which action is received 3863 |
| actTrdNo | Trade Number | long | 8 | 2201 |
| actDtTm | Action Date Time | int | 4 | Date time in milliseconds from 1980<br>1416647701 |
| actId | Action Id | short | 2 | Refer Section “Transcodes”. 5 |
| cpCd | Custodial Participant Code | String | 10 | ICICI0005163 |
| actToken | Action Token | long | 8 | 43315 |

- 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. Thus every event occurred with respect to a particular trade/action will have a new sequence No.
  - On trades/actions download request, maximum trades/actions sequence no available should be sent. If there are no trades/actions present, it has to send the sequence as 0. API shall interpret the request, and will fetch "n" (Configurable) 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 has to again send the maximum trades/actions sequence no available with him.

# 7 Transcodes

## 7.1 Market Type

| | |
|---|---|
| 1 | Normal |
| 2 | Odd Lot |
| 3 | Spot |
| 4 | Auction |
| 5 | Call Auction 1 |
| 6 | Call Auction 2 |

## 7.2 Market Status

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

### 7.3 Transaction Code

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

### 7.4 Activity Type

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

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

Confidential

| | |
|---|---|
| 12 | Call Auction 2 |

### 7.6 Client Type

| | |
|---|---|
| 1 | Cli |
| 2 | Pro |

### 7.7 Buy Sell Flag

| | |
|---|---|
| 1 | BUY |
| 2 | SELL |

### 7.8 Trade Status

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

### 7.9 Option Type

| | |
|---|---|
| CA | Call American |
| PA | Put American |
| CE | Call Europian |
| PE | Put Europian |

### 7.10 Is Approval Flag

| | |
|---|---|
| 1 | Approve |
| 0 | Reject |

### 7.11 Action Id

| | |
|---|---|
| 2 | Buy SI Generated |
| 3 | Sell SI Generated |

Confidential

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

# 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 Codes |
|---|---|---|---|
| 1 | SUCCESS | Request was handled successfully | 200 |
| 2 | UNKNOWN_ERROR | Internal Server Error: Internal server error has occurred in our platform. | 500 |

Confidential

| | | | |
|---|---|---|---|
| 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:<br>* Unauthorized requestor IP address.<br>* API access disabled | 401 |
| 11 | TOKEN_EXPIRED | The TEMPORARY access token generated by the platform has expired and can no longer be used. | 572 |
| 12 | PERMISSION_DENIED | Subscriber has temporarily disallowed access to his private data. | 403 |
| 13 | REQUEST_NOT_FOUND | Registration 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.

### 8.2.1 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 | msgPrepDt | 0105 |
| 4 | Input Data Parameter | msgPrepTm | 0106 |
| 5 | Input Data Parameter | isApproval | 0109 |
| 6 | Input Data Parameter | seqNo | 0107 |
| 7 | Input Data Parameter | srchFilter | 0108 |
| 8 | Input Data Parameter | noOfRec | 0110 |

### 8.2.2 **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 | All HTTP status codes | HTTP error codes | HTTP Response codes. Refer section "HTTP Response Code". | Entire Message |
| 3 | Mismatch in control and data record | Message Level | 0200 | Entire Message |
| 4 | Minimum Required Length | Generic | 0201 | msgId |
| 5 | Maximum Required Length | Generic | 0202 | msgId |
| 6 | Mandatory field | Generic | 0204 | msgId, isApproval, noOfRec, seqNo, srchFilter, trdDate |
| 7 | Data Format like Msg Id / Date Format | Generic | 0206 | msgId, trdDate |
| 8 | Minimum allowed value | Generic | 0207 | seqNo, noOfRec |
| 9 | Maximum allowed value | Generic | 0208 | noOfRec |
| 10 | Invalid Value | Generic | 0209 | seqNo, isApproval, srchFilter, trdDate |
| 11 | System Error | Generic | 0241 | NA |
| 12 | Service Unavailable | Generic | 0242 | NA |
| 13 | Request Parsing Error : Invalid Request Structure | Generic | 0243 | NA |

Confidential

### 8.2.3 **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 BAD Request, error code that will be generated is as shown below:
Field Identifier: 0101 (which is the identifier of the entire message)
Validation Code: 400

code = combination of “Field Identifier” and “Validation Code” =0101400

### 8.2.4 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 |
| 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 has to 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 NOTIS 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.

Confidential