API Documentation for
Online Trade Inquiry Service

Version 1.0

May 2022

National Stock Exchange of India Ltd

Exchange Plaza, Plot No. C/1, G Block,

Bandra-Kurla Complex, Bandra (E)

Mumbai - 400 051.

Notice

Table of Contents

Version	Date	Description
1.1	13th June 2022	API Changes for SLB

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 APIs to our esteemed empanelled members for trade inquiry for SLB segment.

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

1.1 Following operations aspects are covered in this document:

Sr. No.	Operation	Endpoints	Purpose
1	Login [Handshake]	/token	To authenticate the client
2	Trade Inquiry	/notis-slb/trades-inquiry	To disseminate trades information data

1.2 Technical Specifications

1.3 Log-in Work flow

1.4 Message Structures

2   Data Flow Diagram

➔ API Registration

1) For API registration, user need to request msm@nse.co.in to add entry for them in database with relevant details such as below:

- Member Code
- Member Name
- Contact Person Email Address, Name and Mobile Number
- Segment for which access is required
- Static IP which will be used to access the API
- UAT or LIVE

2) The entry for user will be added from backend manually and no frontend will be provided for this purpose.

3) Once entry is added the user will be provided with Consumer Key and Consumer Secret to access the token API in order to get token.

4) Once token is obtained user can access the Trade Inquiry API using below mentioned diagram.

➔ API Call Work Flow

Member -> NSEIL
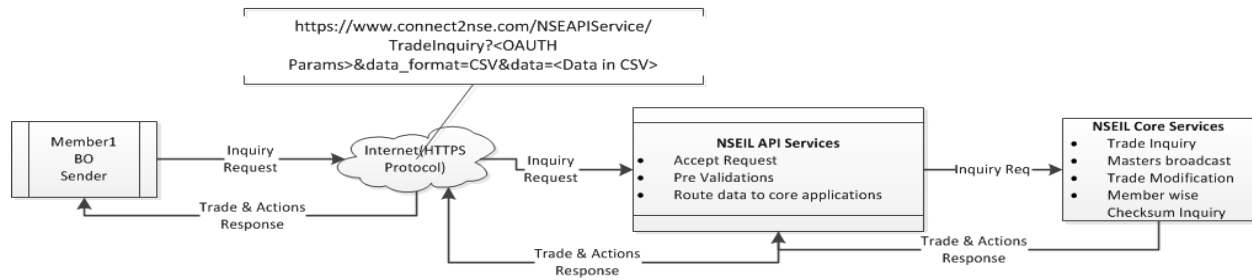
3 Technology Specification

1) Communication Protocol: HTTPS over internet/Leased Line

2) Request/Response Exchange Format: JSON (JavaScript Object Notation).

3) Data Format: CSV (Comma Separated Values).

4) Security Framework: Security Framework should support OAuth 2.0 specifications.

4 API Registration

1) Members need to register by contacting MSD (Member Service Department) for creating the user account details in the database.

2) Member will need to provide information as described below:
   - Member Code
   - Member Name
   - Contact Person Email Address, Name and Mobile Number
   - Segment for which access is required
   - Static IP which will be used to access the API
   - UAT or LIVE

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

4) Once MEMBER receives Consumer Key and Consumer Secret, they can start using API.

5   Log-In Workflow

- Login Handshake (MEMBER → NSEIL)

- Requesting a "Token"

NOTE: When sending the token request, member should request the token from one platform and share the generated token across multiple platforms to avoid the 500 error code that occurs when more than one token has been generated for a consumer key/secret.

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

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

Production/LIVE: https://www.connect2nse.com/token

➔ Sample Request

```
POST /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: 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: ddMMyyyyHHmmssSSS:<6-digit random number>	MjAwMTIwMTcxNjEyMjE1OTE6
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.	ee1073de-45d0-4040-b9c2-eddfa80280c0
2	token_type	String	The type of the token issued	bearer

3	expires_in	int	The lifetime in seconds of the access token. From the first generation, the token is valid for 32400 seconds (approx. 9 hours) 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.

6   Accessing the NSE Data using the API

Trade Inquiry (ALL)

UAT: https://www.devconnect2nse.com/notis-slb/trades-inquiry

Production/LIVE: https://www.connect2nse.com/notis-cm/trades-inquiry

➔   Sample Request (ALL Trades)

```
POST /notis-slb/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,,"
        }
}
```

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> • 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).	00240201310140000001
3	data.dataformat	String	Request data format : Response data format	CSV:CSV
4	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. For first download request of the day, it should be 0.	long	8	0
srchFilter	Search Filter	String	50	• ALL – All Trades
fill1	Filler	String	10	
fill2	Filler	String	10	

➔ Sample Response

➔ Control Record
➔ Record Separator
➔ Data Record(s)

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

 "msgId": "08081202008280000001",

"tradesInquiry":"6,20220725,,,95502,108^2022070500000000,6001,HDFC,X8,88082900582400,HDFC LT
D,EQ,1,N,914,14,1,91,50,2,10978,10978,N,0,7,87915245862912,100000000000010,NIL,1341480069,1
6814,169532,W,N,10978,2,0.0^"
}

}
```

➔ 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

9

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 Refer Section "Transcodes".	Short	2	6
currTrdDate	Current Trade Date (YYYYMMDD)	long	8	20220725
sfill1	Filler	String	10	
sfill2	Filler	String	10	

➔ 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	1360320
noOfRec	Count of trades sent in the response	Int	4	10000
Data Records				
tradesOutput	Array Of Trade Structure	Array Of Trade Structure	Size of (Array Of Trade Structure)	Structure details given below

10

		Array Of Trade Structure			
Sr No.	Field Name	Description	Data Type	Size (in bytes)	Sample
1	trdNo	Trade Number	Long	8	2022070500000000
2	TCd	Trans Code. Refer Section "Transcodes"	Short	2	6001
3	Sym	Symbol	String	10	HDFC
4	Ser	Series	String	2	X8
5	settlementDate	Settlement date in Jiffy Format	Long	8	88082900582400
6	secName	Security Name	String	2	HDFC LTD
7	Inst	Instrument Type	String	30	EQ
8	mktTyp	Market Book Type. Refer Section "Transcodes"	Short	2	1
9	Mkt	Market Type. Refer Section "Transcodes"	String	1	N
10	usrId	User Id	Int	4	914
11	brnCd	Branch Code	Int	4	14
12	bsFlg	Buy Sell Flag. Refer Section "Transcodes"	String	1	1
13	trdQty	Trade Quantity	Int	4	91
14	trdPrc	Trade Price in paise	Int	4	50
15	proCli	Client Type. Refer Section "Transcodes"	Short	2	2
16	cliActNo	Client account number	String	20	10978
17	cpCd	Custodial participant Id	String	12	10978
18	aucPartType	Auction Part Type	Char	1	N
19	aucNo	Auction Number (This field is not applicable)	Short	2	0
20	Settlement	settlement	Int	4	7
21	trdTm	Trade Time in Jiffy Format	Long	8	87915245862912
22	ordNo	Order Number	String	16	1000000000000010
23	oppBrokerCd	Opposite Broker Code (Blank value since members should not be able to view this value)	String	1	NIL
24	ordTm	Order Time in milliseconds from 1980	Long	8	1341480069

25	seqNo	Unique Sequence Number	Long	8	16814
26	Tkn	Token	Int	4	169532
27	stpTyp	Trade Settlement Type	String	1	W
28	secIdentifier	Trade Secondary Identifier	String	1	N
29	TMcode	TMcode	String	12	10978
30	actTyp	actTyp	short	1	2
30	ctclId	CTCL code	String	20	0.0

7   Workflow

1) Trade download works on sequence number basis present in individual trade response packet (*seqNo*).
2) The sequence number signifies the sequence of events for a single trade lifecycle. Thus every event occurred with respect to a particular trade will have a new sequence No.
3) On trades download request, maximum trades sequence no available should be sent. If there are no trades present, it has to send the sequence as 0. API shall interpret the request, and will fetch "n" (Configurable) number of trades, whose trades sequence number is greater than that sent by client. The fetched trades will be sent back to client in response.
4) The trades received by client in response packet are to be stored at client end. On subsequent trades download request, client has to again send the maximum trades sequence no available with him.

8   Transcodes

8.1   Market Type

1	Normal

8.2   Trans 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

8.3   Book Type

1	Regular Lot
3	Stop Loss / MIT

8.4   Client Type

1	Cli
2	Pro

8.5   Buy Sell Flag

1	Borrow
2	Lend
3	Recall
4	Repay

8.6   Instrument Type

EQ	Equities

8.7   Market Status

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

8.8 Activity Type

2	Original Trade
7	Trade Cancellation
5	Client Modification Pre-Image
27	Client Modification Post-Image
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

9 Response Codes

There can be two types of response codes

1) HTTP response codes
2) Message based response codes
3) Async response codes

9.1 HTTP response code

1) HTTP responses shall be generated during login with success or failure status
2) HTTP response shall also be generated in case of any authentication/input validation failure of the message
3) 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
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: * Unauthorized requestor IP address. * 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

9.2 Message based response code

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

9.3   Field Identifier codes

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

9.4   Validation codes

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

16

Sr. No.	Validation	Validation Type	Validation Code	Validation performed on Field
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

9.5 Example for success or failure code

9.5.1 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

9.5.2 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

9.5.3 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:

17

Field Identifier: 0101 (which is the identifier of the entire message)
Validation Code: 0000

code = combination of "Field Identifier" and "Validation Code" =01010000

9.5.4 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

9.6 Async response code

1) Async response code shall be populated in the field "errCd" of the message
2) Below are the list of codes:

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

10 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.

11 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.