API Documentation for NSE IFSC Online Trade Inquiry Service

Version 1.2

February 2023

NSE IFSC Limited
1201, Brigade International Financial Centre
12th floor, Block-14, Road 1C, Zone -1,
GIFT SEZ, Gandhinagar, Gujarat – 382355

Notice

© Copyright NSE IFSC Limited. All rights reserved. Unpublished rights reserved under applicable copyright and trades secret laws.

The contents, ideas and concepts presented herein are proprietary and confidential. Duplication and disclosure to others in whole, or in part is prohibited

Version 1.2 February 2023	
Pages Changed	**Description**
8	Added secret_key_expiry to response of authentication url

Contents

Preface

1. Background

The Exchange is currently providing trade data to Trading Members on a real-time basis through the NSEIFSC Online Trade Inquiry System (NOTIS) client application provided by the Exchange.

Exchange is pleased to announce introduction of Application Program Interface (APIs) for NOTIS which shall enable Trading Members to receive trade information for NSE IFSC using the APIs and develop customized front-ends or interface it with their internal systems.

2. Purpose

This document describes the protocol to be used to communicate with the NOTIS server and download trade and action inquiries and thus serves as a development guide for the trading members.

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

3. Data Flow

➢ API Registration

1. For API registration, user needs to send the registration request to the IFSC team through email. (Email ID: notis@nseifsc.co.in)
2. Once registered, user will be provided with username & password to access the token API in order to get token.
3. Using this token user can access the Trade Inquiry API.

4. Technology Specification

4.1 Communication Protocol: HTTPS over Leased line/Internet.

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

4.3 Data Format: CSV (Comma Separated Values).

4.4 Security Framework: Security Framework should support OAuth 2.0 specifications.

5. API Registration

5.1 Member will need to provide information as described below:

5.1.1 Service → Trade Inquiry Service

5.1.2 Email Id → Email Address where the API credentials would be sent

5.1.3 IP Address → IP Address from which the member will communicate with NSEIFSC API Service

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

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

NOTE: Consumer Secret will expire after 1800 days (approx. 5 years) same would be sent as response in token api , 5 continuous failed token generation attempts will disable user.

6. Log-In Workflow

6.1 Login Handshake (MEMBER NSEIFSC)

6.1.1 Requesting a “Token”

A consumer application needs to send a HTTPS POST request to https://c2n.nseifsc.com/token

- Sample Request

```
POST /auth/token HTTP/1.1
Host: https://c2n.nseifsc.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": "b5a1321d-74c7-4261-948a-804b2cd09531",
   "secret_key_expiry": "10-11-2022",
   "scope": "api_scope",
   "token_type": "bearer",
   "expires_in": 933
}
```

- 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.	119
4	Scope	String	If identical to the scope requested by the client otherwise, REQUIRED.	api_scope
5	secret_key_expiry	String	The secret key expiry key is the date on which the consumer credentials (consumer secret and key) will expire	10-11-2022 Format(DD-MM-YYYY)

- **Note:**

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

b. An access token expires after 'X' minutes of inactivity.

6.2 Request/Response Structure (JSON)

6.2.1 Trade Inquiry (ALL Trades)

Endpoint → /trades-inquiry
URL → https://c2n.nseifsc.com/notis-gt/trades-inquiry

- Sample request call

```
POST/api/notis-gt/trades-inquiryHTTP/1.1 Host:
https://c2n.nseifsc.com

Authorization:Bearer3f64e56704f943b89d24e91

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
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. For first download request of the day, it should be 0.	long	8	0
srchFilter	Search Filter	String	50	ALL - All Trades

Field Name	Description	Data Type	Size (in bytes)	Sample
fill1	Filler	String	10	
fill2	Filler	String	10	

Sample Response

🟨 → Control Record
🟥 → Record Separator
🟦 → Data Record(s)

```
{
"status":
"success",
"messages": {
"code":
"01010000"
},
"data": {
"msgId": "00240201310140000001",
        "tradesInquiry":
        3,20120420,,,200,2^469520,1,226767,71522826125312,14732,500,19180,2,20140801000
        82090
        ,1,29125,1,88R26,07520,,2,6001,1091351717,1,12170,560002001009113,,0808
        2,NIFTY,EQ,OPTI
        DX,,910000,PE,,,,,,,,^469521,1,226768,71522826125312,14732,500,19180,2,20
        1408010008209
        0,1,29125,1,88R26,07520,,2,6001,1091351717,1,12170,560002001009113,,080
        82,NIFTY,EQ,OP TIDX,,910000,PE,,,,,,,,


 }
}
```

➢ 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

Field Name	Description	Data Type	Size (in bytes)	Sample
mktSts	Market Status Refer Section “Transcodes”.	Short	2	2
currTrdDate	Current Trade Date (YYYYMMDD)	long	8	20120420
sfill1	Filler	String	10	
sfill2	Filler	String	10	

- Trade Response Data Structure

Field Name	Description	Data Type	Size (in bytes)	Sample/Remarks
maxSeqNo	Max sequence number sent in response	Long	8	380
noOfRec	Count of trades sent in the response	Int	4	4000
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	469520
2	mkt	Market Type. Refer Section “Transcodes”	String	1	1
3	trdNo	Trade Number	long	8	226767
4	trdTm	Trade Time in Jiffy Format	long	8	71522826125312
5	Tkn	Token	int	4	14732
6	trdQty	Trade Quantity	int	4	500
7	trdPrc	Trade Price in USD	int	4	19180
8	bsFlg	Buy Sell Flag. Refer Section “Transcodes”	String	1	2
9	ordNo	Order Number	double	8	2014080100082090
10	brnCd	Branch Code	int	4	1
11	usrId	User Id	int	4	29125
12	proCli	Client Type. Refer Section “Transcodes”	short	2	1
13	cliActNo	Client account number	String	20	88R26
14	cpCd	Custodial participant Id	String	12	07520
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 Jiffy Format	long	8	1091351717
19	Booktype	Book Type. Refer Section “Transcodes”	short	2	1
20	oppTmCd	Opposite Broker Id	String	1	
21	ctclId	CTCL code	double	8	560002001009113
22	status	Trade Status. Refer Section	String	1	
		“Transcodes”			
23	TmCd	Member Code	String	5	08082
24	sym	Symbol	String	10	NIFTY
25	ser	Series	String	2	EQ
26	inst	Instrument	String	6	OPTIDX
27	expDt	Expiry Date (in msecs from	int	4	1151764200

Sr No.	Field Name	Description	Data Type	Size (in bytes)	Sample
		1980)			
28	strPrc	Strike Price in USD	int	4	910000
29	optType	Option Type for Option Contract. Refer Section "Transcodes"	String	2	PE
30	sessionID	Session ID	int	4	1(0=session1, 1=session2)
31	echoback	Echoback	String	16	0001
32	Fill1	Filler	String	10	
33	Fill2	Filler	String	10	
34	Fill3	Filler	String	10	
35	Fill4	Filler	String	10	
36	Fill5	Filler	String	10	
37	fill6	Filler	String	10	

NOTE:

For all contracts trading in USD, prices must be divided by 10000 while receiving in API responses

Example: If on a contract 'XX' precision is 2 then while receiving, if a price of 202300 is received in API response then this price should be displayed as 20.2300 after dividing the price by 10000 and converting to 2 digits after decimal

For all contracts trading in Cents, prices must be divided by 100 while receiving in API responses

Example: If on a contract 'XX' precision is 2 then while receiving, if a price of 202300 is received in API response then this price should be displayed as 2023.00 after dividing the price by 100 and converting to 2 digits after decimal

Details related to precision and currencies at contract level will be available in ni_contract.txt which is published by exchange on daily basis.

6.3 Action Inquiry (Approval/Rejection/Approve ALL Response)

Endpoint → /actions-inquiry

URL → https://c2n.nseifsc.com/notis-gt/actions-inquiry

➢ Sample request call

```
POST/api/notis-gt/actions-inquiry HTTP/1.1
Host: https://c2n.nseifsc.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3


{
  "version": "1.0",
  "data": {
    "msgId": "00240201310140000001",
    "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 <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.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. For first download request of the day, it should be 0.	long	8	0
srchFilter	Search Filter All (Default)	String	50	ALL

Field Name	Description	Data Type	Size (in bytes)	Remarks
fill1	Filler	String	10	
fill2	Filler	String	10	

- Sample Response

```
{
  "status": "success",
  "messages": {
    "code": "01010000"
  },
  "data": {
    "msgId": "00240201310140000001",
    "actionsInquiry":
“3,20120420,,,250,2^0,4560,226767,1080379404,4^0,4568,226768,1800379440,5"
  }
}
```

- Response Structure

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.actionsInquiry	CSV	Data Structure specified below	Refer Sample Response

- 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 Structur	Structure details give below

			e)	
actResData	Actions Response Data Structure	Response Data Structure	Size of Response Data Structure	Structure details give below

- System Info Response Structure

Field Name	Description	Data Type	Size (in bytes)	Sample
mktSts	Market Status Refer Section "Transcodes".	Short	2	2
currTrdDate	Current Trade Date (YYYYMMDD)	long	8	20120420
sfill1	Filler	String	10	
sfill2	Filler	String	10	

- Action Response Data Structure

Field Name	Description	Data Type	Size (in bytes)	Sample
maxSeqNo	Max sequence number sent in response	Long	8	250
noOfRec	Count of actions sent in response	Int	4	4000
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".
seqNo	Trade Sequence number	int	4	Sequence number of the trade for which action is received

actTrdNo	Trade Number	long	8	226767
actDtTm	Action Date Time	int	4	Date time in milliseconds from 1980
actId	Action Id	short	2	Refer Section “Transcodes”.
cpCd	Custodial Participant Code	String	12	EDEL00000007

- 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	NLT
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
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 (Not Used)
5	Odd Lot (Not Used)
6	Spot (Not Used)
7	Auction (Not Used)
11	Call Auction 1 (Not Used)
12	Call Auction 2 (Not Used)

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 European
PE	Put European

7.10 Is Approval Flag

1	Approve
0	Reject

7.11 Action Id

2	Buy SI Generated
3	Sell SI Generated
4	AppRej Buy Approval
5	AppRej Sell Approval
6	Buy SI Rejected
7	Sell SI Rejected
14	AppRej Buy Rejected
15	AppRej Sell Rejected
16	Buy SI Cancelled
17	Sell SI Cancelled
18	Approval All sent to trading

6 Response Codes

There can be three 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:

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 die 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	UNAUTHORISED: Failed to authenticate the request	CONSUMER_KEY_UNKNOWN: The provided Consumer Key (API key) is not registered in our system or service is not recognized and also for unauthenticated URI	401
9	UNAUTHORISED: Failed to authenticate the request	TOEKN_INVALID: The provided token is not registered in our system	401
10	UNAUTHORISED: Failed to authenticate the request	UNAUTHORISED: - Unauthorized requestor IP address - API access disabled	401
11	TOKEN_EXPIRED	The TEMPORARY access token generated by the platform has expired and can no	572

Sr. No.	Reason	Meaning	HTTP Response Code
		longer be used.	
12	PERMISSION_DENIED	Subscriber has temporarily disallowed access to his private data.	403
13	REQUEST_NOT_FOUND	Registration request not found.	570
14	CONSUMER_SECRET_EXPIRED	Consumer Secret expires after 1800 days (approx. 5 years) of generation	406

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	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

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

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

8.2.1 Sample example for success or failure code

8.2.1.1 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

8.2.1.2 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

8.2.1.3 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:

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

Validation Code: 0000

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

8.2.1.4 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

8.3 Async response code

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

Error	Error Code
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 IFSC or the MEMBER end, login workflow has to be re-initiated.