API Documentation for CIM Allocation Service

Version 1.3

April 2022

Exchange Plaza, C-1, Block G,
Bandra Kurla Complex, Bandra (E), Mumbai,
Maharashtra 400051

1 Contents

2 Version History

Version	Page No	Description
1.0	-	Initial Version
1.1	7, 9, 19, 4	Acceptance of allocation/un-allocation EOD requests and added GZIP compression information.
1.2	10,11,13,14,16,17	Inclusion of CSV data format
1.3	6	Change in Registration process

3 Background

This document is to aid members/business to access the CIM Allocation API.

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

- Following operations aspects are covered in this document:

Sr. No.	Operation	Endpoints	Purpose
1	Login [Handshake]	/coll-token	To authenticate the client consumer key and secret and generate token to access the API.
2	Allocation	/coll-allocation/allocation	Addition and reduction for CASH/BG/FD.
3	Allocation Inquiry	/coll-allocation/allocation-statusInquiry	Status inquiry

- Technical Specifications
- Log-in Workflow
- Message Structures

4 Technology Specification

- Communication Protocol: HTTPS over Leased line/Internet.
- Request/Response Exchange Format: JSON (JavaScript Object Notation).
- Security Framework: Security Framework should support OAuth 2.0 specifications.
- Request/Response will be compressed in GZIP.

5 API Registration

- Members need to register through sending a mail to collaterals_ops@nsccl.co.in.

- Member will need to provide information as described below:
  1. IP Address: IP Address from which the member will communicate with API Service
  2. Email Id

- Once this information is provided, admin at NCL will verify and generate the Consumer Key/Consumer Secret.

- Once member receives the Consumer Key/Consumer Secret through email, they can start using the API.

6 Log-In Workflow

- Login Handshake (MEMBER --> NCL)
- Requesting a "Token"

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

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

➔ Sample Request

```
POST /coll-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": "3f64e567-04f9-43b8-9d24-e99856b24151",
   "token_type": "bearer",
   "expires_in": "32400",
   "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.	3f64e567-04f9-43b8-9d24-e99856b24151
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.	32400
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.

7 Request/Response Structure (JSON)

NOTE: The member will be allowed to submit the request for a maximum of 1000 records per request.

The amount mentioned in the structure would be the final allocation requested amount for the mentioned combination.

7.1 Allocation Request

Production/LIVE: https://www.connect2nse.com/coll-allocation/allocation

Set Authorization key value as Bearer 'Token value' obtained from token login request.

➔ Sample request call

```
POST /coll-allocation/allocation HTTP/1.1
Content-Type: application/json
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3
```

```
{
  "version": "1.0",
  "data": {
    "msgId": "22222202110140000001",
    "requestType": "I",
    "allocationRequest": [
      {
        "curDate": "14-OCT-2021",
        "segment": "CM",
        "cmCode": "22222",
        " tmCode": "11223",
        "cpCode": "",
        "cliCode": "",
        "accType": "P",
        "amt": 10000,
        "filler1": "",
        "filler2": "",
        "filler3": "",
        "filler4": "",
        "filler5": "",
        "filler6": "",
        "action": "D"
      },
      {
        "curDate": "14-OCT-2021",
```

```
    "segment": "CM",
    "cmCode": "22222",
    " tmCode": "11111",
    "cpCode": "",
    "cliCode": "",
    "accType": "P",
    "amt": 5000,
    "filler1": "",
    "filler2": "",
    "filler3": "",
    "filler4": "",
    "filler5": "",
    "filler6": "",
    "action": "D"
  },
  {
    "curDate": "14-OCT-2021",
    "segment": "CM",
    "cmCode": "22222",
    " tmCode": "",
    "cpCode": "",
    "cliCode": "",
    "accType": "P",
    "amt": 50000,
    "filler1": "",
    "filler2": "",
    "filler3": "",
    "filler4": "",
    "filler5": "",
    "filler6": "",
    "action": "D"
  },
  {
    "curDate": "14-OCT-2021",
    "segment": "FO",
    "cmCode": "22222",
    " tmCode": "11111",
    "cpCode": "",
    "cliCode": "ABCD",
    "accType": "C",
    "amt": 50000,
    "filler1": "",
    "filler2": "",
    "filler3": "",
    "filler4": "",
    "filler5": "",
    "filler6": "",
    "action": "U"
```

```
        },
        {
          "curDate": "14-OCT-2021",
          "segment": "CD",
          "cmCode": "22222",
          " tmCode": "",
          "cpCode": "CP12345678",
          "cliCode": "",
          "accType": "C",
          "amt": 10000,
          "filler1": "",
          "filler2": "",
          "filler3": "",
          "filler4": "",
          "filler5": "",
          "filler6": "",
          "action": "U"
        }
      ]
   }
}
```


OR

```
POST /coll-allocation/allocation HTTP/1.1
Content-Type: application/json
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3
```


```
{
   "version": "1.0",
   "data": {
      "msgId": "22222202110140000001",
      "requestType": "I",
      "allocationRequest":"14-OCT-2021,CM,22222,11223,,,P,10000,,,,,,,D^14-OCT-
2021,CM,22222,11111,,,P,5000,,,,,,,D^14-OCT-2021,CM,22222,,,,P,50000,,,,,,,D^14-OCT-
2021,FO,22222,11111,,ABCD,C,50000,,,,,,,U^14-OCT-2021,CD,22222,,CP12345678,,C,10000,,,,,,,U"
   }
}
```

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 each request <CODE><YYYYMMDD><nnnnnnn> • MEMBERCODE – Member code (Max Length: 5) • YYYYMMDD – Date format • nnnnnnn – Unique number	22222202110140000001
3	data.requestType	String	I – Immediate E – EOD	I
4	data.allocationRequest	JSON/CSV	Data Structure specified below	Array of records

Request Data Payload (JSON/CSV) : allocationRequest					
Sr. No.	**Field Details**	**Data Type**	**Length**	**Description**	**Sample Value**
1.	Current Date	String	11	Trade Date with DD-MON-YYYY format	14-OCT-2021
2.	Segment	String	3	CM – Cash Market FO – Future & Options CD– Currency derivatives DT– Debt CO– Commodity SLB– Securities Lending & Borrowing TPR– Triparty OFS– Offer for sale	CM
3.	CM Code	String	6	Primary Member Code	22222
4.	TM Code	String	5	TM Code	
5.	CP Code	String	12	CP Code	

6.	Cli Code	String	10	Cli Code	
7.	Account Type	String	1	CLI – C and PRO - P	P
8.	Amount	Number	(15,2)	Amount	50000
9.	Filler1	String	20	Filler. Reserved for future.	
10.	Filler2	String	20	Filler. Reserved for future.	
11.	Filler3	String	20	Filler. Reserved for future.	
12.	Filler4	String	20	Filler. Reserved for future.	
13.	Filler5	String	20	Filler. Reserved for future.	
14.	Filler6	String	20	Filler. Reserved for future.	
15.	Action	String	1	Upward Direction – U Downward Direction - D	D

➔ Sample Response - Acknowledgment

HTTP/1.1 200 OK

Content-Type: application/json


```
{
  "status": "success",
  "messages": {},
  "data": {
    "response": "Request accepted for Message ID: 22222202110140000001"
  }
}
```


Response shall be sent at message level and not for individual records within a message.

➔ Response Structure

Response Data Payload (JSON) - Acknowledgment				
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.response	String	Request status	Request accepted for MessageID: 22222202110140000001

7.2 Inquiry Request

Allocation Inquiry

Production/LIVE: https://www.connect2nse.com/coll-allocation/allocation-statusinquiry

Inquiry Request shall be sent at message level.

➔ Sample request call

```
POST /coll-allocation/allocation-statusInquiry HTTP/1.1
Content-Type: application/json
Host: www.connect2nse.com
Authorization: Bearer 3f64e567-04f9-43b8-9d24-e99856b24151
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3
{
  "version": "1.0",
  "data": {
    "msgId": "22222202110140000001"
    "dataFormat": "CSV"
  }
}
```

Request Data Payload (JSON)				
Sr. No.	**Parameter Name**	**Data Type**	**Description**	**Sample Value**
1	version	String	API version	1.0
2	data.msgId	String	Message Id of already requested transaction.	22222202110140000001
3	data.dataformat	String	Response data format CSV JSON	JSON

Inquiry Response shall be provided for individual records within a message after processing of all records in the message. If any of the records are in processing state within a message, then the response shall be provided at message level.

➔ Sample Response (JSON)


```
HTTP/1.1 200 OK
Content-Type: application/json
{
   "status": "success",
   "messages": {
      "code": "01010000"

   },
   "data": {
      "inquiryResponse": [
         {
            "curDate": "14-OCT-2021",
            "segment": "CM",
            "cmCode": "22222",
            " tmCode": "11223",
            "cpCode": "",
            "cliCode": "",
            "accType": "P",
            "amt": 10000,
            "filler1": "",
            "filler2": "",
            "filler3": "",
            "filler4": "",
            "filler5": "",
            "filler6": "",
            "action": "D",
            "errCd": "01050100"
         },
         {
            "curDate": "14-OCT-2021",
            "segment": "CM",
            "cmCode": "22222",
            " tmCode": "11111",
            "cpCode": "",
            "cliCode": "",
            "accType": "P",
            "amt": 5000,
            "filler1": "",
            "filler2": "",
            "filler3": "",
            "filler4": "",
```

```
    "filler5": "",
    "filler6": "",
    "action": "D",
    "errCd": "01050100"
  },
  {
    "curDate": "14-OCT-2021",
    "segment": "CM",
    "cmCode": "22222",
    " tmCode": "",
    "cpCode": "",
    "cliCode": "",
    "accType": "P",
    "amt": 50000,
    "filler1": "",
    "filler2": "",
    "filler3": "",
    "filler4": "",
    "filler5": "",
    "filler6": "",
    "action": "D",
    "errCd": "01050100"
  },
  {
    "curDate": "14-OCT-2021",
    "segment": "FO",
    "cmCode": "22222",
    " tmCode": "11111",
    "cpCode": "",
    "cliCode": "ABCD",
    "accType": "C",
    "amt": 50000,
    "filler1": "",
    "filler2": "",
    "filler3": "",
    "filler4": "",
    "filler5": "",
    "filler6": "",
    "action": "U",
    "errCd": "01050100"
  },
  {
    "curDate": "14-OCT-2021",
    "segment": "CD",
    "cmCode": "22222",
    " tmCode": "",
    "cpCode": "CP12345678",
    "cliCode": "",
```

```
            "accType": "C",
            "amt": 10000,
            "filler1": "",
            "filler2": "",
            "filler3": "",
            "filler4": "",
            "filler5": "",
            "filler6": "",
            "action": "U",
            "errCd": "01050100"
         }
      ]
   }
}
```


➔ Sample Response (CSV)


```
HTTP/1.1 200 OK
Content-Type: application/json
{
   "status": "success",
   "messages": {
      "code": "01010000"

   },
   "data": {
      "inquiryResponse":"14-OCT-2021,CM,22222,11223,,,P,10000,,,,,,,D,01050100^14-OCT-
2021,CM,22222,11111,,,P,5000,,,,,,,D,01050100^14-OCT-2021,CM,22222,,,,P,50000,,,,,,,D,01050100^14-
OCT-2021,FO,22222,11111,,ABCD,C,50000,,,,,,,U,01050100^14-OCT-
2021,CD,22222,,CP12345678,,C,10000,,,,,,,U,01050100"
   }
}
```


➔ 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.inquiryResponse	JSON/CSV	Data Structure specified below	Array of records

Response Data Payload (JSON/CSV)					
Sr. No.	**Field Details**	**Data Type**	**Length**	**Description**	**Sample Value**
1	Current Date	String	11	DD-MON-YYYY	14-OCT-2021
2	Segment	String	3	CM – Cash Market FO – Future & Options CD– Currency derivatives DT– Debt CO– Commodity SLB– Securities Lending & Borrowing TPR– Triparty OFS– Offer for sale	CM
3	CM Code	String	6	Primary Member Code	22222
4	TM Code	String	5	TM Code	
5	CP Code	String	12	CP Code	
6	Cli Code	String	10	Cli Code	
7	Account Type	String	1	Prop – P, Cli - C	P
8	Amount	Number	(15,2)	Amount	10000
9	Filler1	String	20	Filler. Reserved for future.	
10	Filler2	String	20	Filler. Reserved for future.	
11	Filler3	String	20	Filler. Reserved for future.	
12	Filler4	String	20	Filler. Reserved for future.	
13	Filler5	String	20	Filler. Reserved for future.	
14	Filler6	String	20	Filler. Reserved for future.	
15	Action	String	1	U- Upward, D-Downward	D
16	Error Code	Number	8	Error code	01050100

8 Response Codes

There can be two types of response codes

- HTTP response codes
- Message level response codes
- Record level 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
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

8.2 Message level 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.

8.3 Record level Response Codes

- Async response code shall be populated in the field "**errCd**" of the JSON response message for each record.
- 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.

8.4 Field Identifier

Sr. No.	Module	Field Name	Field Identifier
1	Entire Message		0101
2	Header	msgId	0102
3	Header	allocationRequest	0103
4	Entire Record		0105
5	RecordDetails	curDate	0107
6	RecordDetails	segment	0108
7	RecordDetails	cmCode	0109
8	RecordDetails	tmCode	0110
9	RecordDetails	cpCode	0111
10	RecordDetails	cliCode	0112
11	RecordDetails	accType	0113

12	RecordDetails	amt	0114
13	RecordDetails	action	0115
14	RecordDetails	Filler1	0116
15	RecordDetails	Filler2	0117
16	RecordDetails	Filler3	0118
17	RecordDetails	Filler4	0119
18	RecordDetails	Filler5	0120
19	RecordDetails	Filler6	0121
20	Header	requestType	0122

8.5 Validation Code

Sr. No.	Validation	Validation Type	Validation Code	Validation performed on Field
1	Submitted to server successfully	Message Level	0000	Entire Message
2	Status Request Under process – 0102	Message Level	0102	Entire Message
3	Status Request Rejected due to higher margin utilization - 0103 Request partially accepted due to higher margin utilization – 0104	Record level	0103 0104 0100	Entire record

	Request accepted successfully - 0100			
4	Minimum Required Length	Generic	0201	All Fields *
5	Maximum Required Length	Generic	0202	All Fields *
6	Range validation	Generic	0203	All Fields *
7	Mandatory Field	Generic	0204	All Fields *
8	Special Characters not allowed	Generic	0205	All Fields *
9	Data Format like Msg Id or File Name/ Date Format	Generic	0206	All Fields *
10	Minimum allowed value	Generic	0207	All Fields *
11	Maximum allowed value	Generic	0208	All Fields *
12	Invalid Value	Generic	0209	All Fields*
13	Duplicate MsgID/File Name	Generic	0210	All Fields *
14	Service Unavailable/ Timing window over	Generic	0212	NA
15	System Error	Generic	0213	NA
16	Number of records	Generic	0214	Number of records submitted is greater than configured allowed records per request
17	Current Date	Field	0217	curDate
18	Segment	Field	0218	segment

	Valid segment CM FO CD DT SLB OFS TPR			
19	Primary Member Code Should be available in Clearing Member Master	Field	0219	cmCode
20	Trading Member Code Should be the part of valid CM-TM link of specified segment	Field	0220	tmCode
21	CP code Should be the part of valid CM-TM link of specified segment	Field	0221	cpCode
22	Account Type P – Prop C - Cli	Field	0222	accType
23	Upward or downward	Field	0224	action
24	All HTTP status codes	HTTP response codes	HTTP Response codes. Refer section “HTTP Response Code”.	Entire Message

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.