

Protocol for WEB API for Members

UPIX System



The NSE Clearing Limited (National Clearing)
Exchange Plaza, Plot No. C/1, G Block,
Bandra-Kurla Complex, Bandra (E),
Mumbai - 400 051

Notice

© Copyright NSE Clearing Ltd (NCL). 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 Unpublished rights reserved under applicable copyright and trades secret laws.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

Table of Contents

Introduction	3
General Instructions.....	3
HTTP Status Codes	4
Common Error Response JSON	4
Environment Details.....	4
API Security	4
Generating JWT.....	5
Validating JWT.....	6
Clearing Corporation APIs.....	8
POST /<version>/request/blockrelease	8
POST /<version>/request/transfer	9
POST /<version>/inquire/blockrelease.....	10
POST /<version>/inquire/transfer	12
POST /<version>/inquire/blockcreate	13
POST /rest/<version>/inquire/blockdebitlist.....	15
POST /rest/<version>/inquire/blockdebit	16
POST /<version>/inquire/regdereg.....	18
POST /<version>/inquire/segblockbal	20
Member APIs.....	21
POST /<version>/notify/blockrelease	21
POST /<version>/notify/transfer	21
POST /<version>/notify/blockcreate	22
POST /<version>/notify/blockdebit	22
POST /<version>/notify/regdereg.....	23
Appendix A - Status Codes	24

Introduction

This document describes the Web API's to programmatically access and exchange data with UPIX Platform and Members..

The document outlines the messaging protocols and structures for developing such interface.

General Instructions

1. Following headers need to be provided in all API calls made to clearing corporation.
 - **Content-Type:** This header should be provided in all requests with method as "POST". Header value should be "application/json".
 - **User-Agent:** All requests should contain this header. The value of "User-Agent" header can be "/".
 - **Accept-Encoding:** This header is required in all API calls to CC. The value of this header should be blank.
 - **Accept:** This header value should be "application/json"
2. Following headers will be added in all API calls made to member APIs
 - **Content-Type:** This header will be provided in all requests with method as "POST". Header value will be "application/json".
 - **Accept:** This header value will be "application/json"
3. Path parameters and query parameters in the URL's must be encoded using percent encoding. (Refer http://www.w3schools.com/tags/ref_urlencode.asp for details)
4. All request and response messages are in JSON (Javascript Object Notation) format. (Refer <http://www.json.org/> for details).
5. Some of the key specifications related to JSON and standards followed for the API's are as follows
 - JSON is built on 2 structures. Map containing key value pairs and an ordered list of values.
 - A value could be boolean (true / false), number, decimal, String or a structure (List or Object).
 - Object or key value pair structure consists of keys which are strings and values of any of the above types. E.g. {"name":"Amit", "age":25}
 - List contains list of values. E.g. ["Amit", "Ajay", "Vikas"]
 - A Boolean has only 2 values true or false.
 - String values are enclosed in double quotes. e.g. "name", "Amit", "Pending"
 - Numbers and decimals are represented without any thousand - separator character. Decimal indicator is dot (".")
 - Numbers have an optional maximum number of digits. If not specified then it is defaulted to 18.
 - Decimals have 2 mandatory length parameters. The first length parameter indicates number of digits in the whole part (before decimal place) and the second length parameter indicates number of digits in the decimal part (after decimal place).
 - All dates, times and datetimes are represented as strings and in Indian standard time. Dates are formatted using format "dd-MMM-yyyy" (E.g. 01-Jan-2023 or 01-JAN-2023). Time are formatted as "hh24:mm:ss". Date times are formatted as "dd-MMM-yyyy hh24:mm:ss" (E.g. 01-Jan-2016 15:30:00).
6. All URLs for API will be always in lower case.

7. All JSON field names will follow camel-hump style of naming. A field with multiple words would be concatenated without spaces. All characters will be in lower case. First characters of words other than the first word in the field name will be in upper case. For e.g. field for "Order Number" could be represented by field name "orderNumber". Other examples are "firstName", "lastName".
8. In case of JSONs representing an object or a key-value pair, keys with null values could be omitted from the JSON.

HTTP Status Codes

All API's will respond with an HTTP status code. A status code of 200 would indicate successful execution of the API and the response body would be as defined in the API specification.

In case of an error a HTTP status code other than 200 will be returned. The API may or may not return an error response JSON depending upon the type of error encountered. Following are the HTTP status codes that could be returned by the APIs

Status Code	Description	JSON Response
400	Indicates a validation / business logic error / json parsing errors	Yes
401	Indicates that the JWT token shared for authentication is invalid or expired.	Yes
404	Resource does not exist	Yes
405	Method not allowed for the resource.	No
500	Any other application error. Such errors are to be reported to the support desk.	Yes
503	Service unavailable	No

Common Error Response JSON

Field	Type	Mandatory	Description
code	Number	Yes	Http Status Code. See above
messages	List<String>	Yes	One or more error messages

Sample Response

```
{
  "code": 400,
  "messages": [
    "Invalid JSON."
  ]
}
```

Environment Details

Base URL for all APIs will be as follows

Testing Environment: Details will be provided subsequently.

Live Environment: Details will be provided subsequently.

API Security

Client generated JWT will be used for authenticating clients. In case of Clearing Corporation APIs, the member acts as a client (API Invoker) and CC acts as server (API Provider). In case of Member APIs, the

clearing corporation acts as a client and the Member acts as server. Refer <https://jwt.io/> for more details on JWT.

Generating JWT

For every API call, the client will generate a new JWT. JWT will contain 3 parts.

Header

The header of JWT will have following claims

Sr No	Claim	Claim Description	Value
1	typ	Type of Token	Always "JWT"
2	alg	Signature Algorithm	Always "HS256"

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

Payload

The payload of JWT will have following claims

Sr No	Claim	Claim Description	Value
1	sub	Subject	Identifies the client invoking the API. In case of Clearing corporation APIs where member acts as a client the value should be member identifier in the format <MEMTYPE>_<MEMCODE> Where MEMTYPE = Member Type. CM or TM And MEMCODE = 5 digit member code e.g. CM_12345, TM_56565 In case of Member APIs where clearing corporation acts as a client the value will be "NCL"
2	aud	Audience	Identifies the server whose API is being invoked by the client. In case of Clearing corporation APIs the value should be "NCL" In case of Member APIs the value should be member identifier in the format <MEMTYPE>_<MEMCODE> Where MEMTYPE = Member Type. CM or TM And MEMCODE = 5 digit member code e.g. CM_12345, TM_56565
3	iat	Issues At	The time at which the token was self generated by the client. It should be a integer number and should be computed as milliseconds since the Unix epoch (January 1, 1970, 00:00:00 UTC). For example timestamp of "29-Jun-2023 13:12:48,404 IST" gets converted to 1688024568404. Reference https://currentmillis.com

4	upixApi	Private Claim - UPIX system API	The API for which the JWT token was generated in the format <METHOD>_<APIURL> Where METHOD = API method. POST / GET APIURL = URL of the API. For example : "POST_/v1/request/blockrelease"
---	---------	---------------------------------	---

```
{
  "sub": "CM_12345",
  "aud": "NCL",
  "iat": 1688024568404,
  "upixApi": "POST_/v1/request/blockrelease"
}
```

Signature

The client and server will share a common 32 byte secret key which will be base64 encoded for sharing and storing at both ends. Signature will be generated in following steps

- The signing input will be created by concatenating the Base64 URL-safe encoded header and payload, separated by a period ('.'). For all string to bytes conversion and vice-versa UTF-8 character encoding will be used.
- The HMAC-SHA256 algorithm is applied to the signing input using the 32 byte secret key. This produces a cryptographic hash, which is then Base64 URL-safe encoded to form the signature.

Transmitting Token

The JWT will then be constructed by concatenating the signing input and the signature separated by a period ('.')

- signingInput = base64UrlEncode(header) + "." + base64UrlEncode(payload)
- signature = base64UrlEncode(HMACSHA256(signingInput), secretKeyBytes)
- JWT = signingInput + "." + signature

The token will be transmitted in "Authorization" header using the "Bearer" scheme.

```
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJDTV8xMjM0NSIsImF1ZCI6Ikd5DTCIsImhhdCI6MTY4ODAyNDU2ODQwNCwidXBpeEFwaSI6IjBPU1RfL3YxL3JlcXVlc3QvYmxxY2tyZWxlYXNlIn0.Wn7faCsPwXqBytKH9ItD_kg7-nllFRDTSLL6Iyi0glo
```

Validating JWT

The server should perform following steps to validate the JWT

- Split JWT into 3 encoded parts using period (".")
- Base 64 URL decode header, payload and signature from respective parts.
- Validate the header claims. "typ" should be "JWT" and "alg" should be "HS256"

- Validate the payload claims “sub”, “aud”, “upixApi”. Value in field “iat” should match the server time with a tolerance of plus or minus 1 minute I.e. 60000 milli-seconds.
- Get secret key corresponding to client identified using “sub” claim. Generate signature original JWT parts 1 and 2 concatenated by period “.” and using the common secret key using HMAC-SHA256 algorithm. Base64 URL-safe encode the signature and compare it with the 3rd part of JWT token. A match will indicate a success.

Clearing Corporation APIs

This chapter gives details of the API's exposed by clearing corporation and to be consumed by members.

POST /<version>/request/blockrelease

This API is exclusive for clearing members and will allow clearing members to submit request to host system for releasing block on one or more UMN's. A maximum of 1000 release request can be submitted in a single API call. Each release request should be assigned a unique request id. The host system in its response will acknowledge the receipt of each and every release request and would then asynchronously process all the requests in the list. The host system will notify the status of release requests to the clearing member via corresponding notify API. Alternatively the clearing member should be able to query for block release request status using "reqId" field , using corresponding inquire API.

Request	JSON
Response	JSON

Request JSON

List of maps containing following fields

Field	Type	Mandatory	Description
reqId	String (40)	Yes	Unique request id assigned for the request by clearing member
seg	String (3)	Yes	Segment
cmCd	String (5)	Yes	Primary member code
tmCd	String (5)	Yes	Trading member code
ucc	String (10)	Yes	UCC code
tmRefId	String (40)	Yes	Unique reference id assigned by trading member to the block request.
umn	String (100)	Yes	Unique mandate number as received from UPI.

Sample Request

```
[
  {
    "reqId": "9876520230116R0000000001",
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",
    "tmRefId": "TEST05555130123000001",
    "umn": "TESTUMN1234"
  },
  {
    "reqId": "9876520230116R0000000002",
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001002",
    "tmRefId": "TEST05555130123000002",
    "umn": "TESTUMN1235"
  }
]
```


Response JSON

List of maps containing acknowledging the requests with following fields.

Field	Type	Mandatory	Description
reqId	String (40)	Yes	Same as input
status	String (8)	Yes	01010000 = Submitted Successfully See Appendix B for other status codes.

Sample Response

```
[
  {
    "reqId": "9876520230116R0000000001",
    "status": "01010000"
  },
  {
    "reqId": "9876520230116R0000000002",
    "status": "01010000"
  }
]
```

POST /<version>/request/transfer

This API is exclusive for clearing members and will allow clearing members to submit a list of transfer collateral requests to the host system. A maximum of 1000 transfer request can be submitted in a single list. A unique request id should be assigned for each transfer request in the list. The host system in its response will acknowledge the receipt of each and every transfer request and would then asynchronously process all the transfers in the request. The host system will notify the status of the transfer requests to the clearing member via corresponding notify API. Alternatively the clearing member should be able to query the status of individual transfer requests using corresponding inquire API.

Request	JSON
Response	JSON

Request JSON

Field	Type	Mandatory	Description
reqId	String (40)	Yes	Unique request no at clearing member
cmCd	String (5)	Yes	Primary member code
tmCd	String (5)	Yes	Trading member code
ucc	String (10)	Yes	UCC code
srcSeg	String (3)	Yes	Source segment from which the amount will be transferred
trgSeg	String (3)	Yes	Target segment to which the amount will be transferred
amtTfr	Decimal (15,2)	Yes	Amount to be transferred

Sample Request

```
[
  {
    "reqId": "9876520230116T0000000001",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",

```

```

    "srcSeg": "CM",
    "trgSeg": "FO",
    "amtTfr": 200
  },
  {
    "reqId": "9876520230116T0000000002",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001004",
    "srcSeg": "CM",
    "trgSeg": "FO",
    "amtTfr": 1500
  }
]

```

Response JSON

List of maps containing acknowledging the requests with following fields.

Field	Type	Mandatory	Description
reqId	String (40)	Yes	Same as input
status	String(8)	Yes	01010000 = Submitted Successfully See Appendix B for other status codes.

Sample Response

```

[
  {
    "reqId": "9876520230116R0000000001",
    "status": "01010000"
  },
  {
    "reqId": "9876520230116R0000000002",
    "status": "01010000"
  }
]

```

POST /<version>/inquire/blockrelease

This API will allow clearing members to inquire status of one or more previously submitted block release requests using API POST /<version>/request/blockrelease. A maximum of 1000 release requests can be inquired in a single API call.

Request	JSON
Response	JSON

Request JSON

List of block release request ids.

Field	Type	Mandatory	Description
reqId	List<String (40)>	Yes	List of Request id of previously submitted block release request.

Sample Request

```

[
  "9876520230116R0000000001",
  "9876520230116R0000000002"
]

```

Response JSON

List of maps containing following fields.

Block Release Structure

Field	Type	Mandatory	Description
reqId	String (40)	Yes	Same as in Request JSON of API POST /<version>/request/blockrelease
seg	String (3)	Yes	
cmCd	String (5)	Yes	
tmCd	String (5)	Yes	
ucc	String (10)	Yes	
tmRefId	String (40)	Yes	
umn	String (100)	No	
ifsc	String (11)	No	Details of the block as received in the block create message. If block identified by umn does not exist then null values will be returned in these fields.
accNo	String (30)	No	
amtBlocked	Decimal (15,2)	No	
totAmtDr	Decimal (15,2)	No	
ccRefId	String (40)	Yes	Unique reference id assigned by CC for the release request
rc	Number (4)	No	Status code as received from UPI. It will be null in case of rejection by NCL or response pending from UPI.
status	String(8)	Yes	Status of individual block release 01010100 = Success See Appendix B for status codes for failures.
amtRelease	Decimal (15,2)	Conditional	Mandatory if status = success. Amount released back to client
relTime	Date Time	Conditional	Mandatory if status = success. Time when release was done. In "DD-MON-YYYY HH24:MI:SS" format (IST)

Sample Response

```
[
  {
    "reqId": "9876520230116R0000000001",
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",
    "tmRefId": "TEST05555130123000001",
    "umn": "TESTUMN1234",
    "ifsc": "BANK0000111",
    "accNo": "ACCNO000001",
    "amtBlocked": 1500,
    "totAmtDr": 500,
    "ccRefId": "9876520230116R00000000100",
    "rc": 100,
    "status": "01010100",
    "amtRelease": 1000,
    "relTime": "16-JAN-2023 16:15:01"
  },
  {
    "reqId": "9876520230116R0000000002",
    "seg": "FO",
    "cmCd": "98765",
```

```

    "tmCd": "12345",
    "ucc": "40001004",
    "tmRefId": "TEST05555130123000101",
    "umn": "TESTUMN1543",
    "ifsc": "BANK0000122",
    "accNo": "ACCNO000021",
    "amtBlocked": 100,
    "totAmtDr": 0,
    "ccRefId": "9876520230116R00000000101",
    "rc": 100,
    "status": "01010100",
    "amtRelease": 100,
    "relTime": "16-JAN-2023 16:15:02"
  }
]

```

POST /<version>/inquire/transfer

This API will allow clearing members to inquire status of one or more previously submitted transfer requests using API POST /<version>/request/transfer. A maximum of 1000 transfer requests can be inquired in a single API call.

Request	JSON
Response	JSON

Request JSON

List of transfer request ids.

Field	Type	Mandatory	Description
reqId	List<String (40)>	Yes	List of Request id of previously submitted block release request.

Sample Request

```

[
  "9876520230116R0000000001",
  "9876520230116R0000000002"
]

```

Response JSON

List of maps containing following fields.

Transfer Structure

Field	Type	Mandatory	Description
reqId	String (40)	Yes	Same as in Request JSON of API POST /<version>/request/transfer
cmCd	String (5)	Yes	
tmCd	String (5)	Yes	
ucc	String (10)	Yes	
srcSeg	String (3)	Yes	
trgSeg	String (3)	Yes	
amtTfr	Decimal (15,2)	Yes	Unique reference id assigned by CC to the transfer request.
ccRefId	String (40)	Yes	
status	String(8)	Yes	Status of individual transfer 01010100 = Success See Appendix B for status codes for failures.

Sample Response

```
[
  {
    "reqId": "9876520230116T000000001 ",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",
    "srcSeg": "CM",
    "trgSeg": "FO",
    "amtTfr": 200,
    "ccRefId": "9876520230116R00000000100",
    "status": "01010100"
  },
  {
    "reqId": "9876520230116T000000002",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001004",
    "srcSeg": "CM",
    "trgSeg": "FO",
    "amtTfr": 1500,
    "ccRefId": "9876520230116R00000000101",
    "status": "01010100"
  }
]
```

POST /<version>/inquire/blockcreate

This API will allow members (both CM & TM) to inquire one or more block creations using tmRefIds. A maximum of 1000 inquiries can be submitted in a single API call.

Request	JSON
Response	JSON

Request JSON

List of Map with each map representing a block create request initiated by TM using tmRefId.

Field	Type	Mandatory	Description
seg	String (3)	Yes	Original segment in which the block was created.
cmCd	String (5)	Yes	Primary member code
tmCd	String (5)	Yes	Trading member code
ucc	String (10)	Yes	UCC code
tmRefId	String (40)	Yes	Unique reference id assigned by trading member to the block request.

Sample Request

```
[
  {
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",
    "tmRefId": "TEST05555130123000001"
  },
  {
    "seg": "FO",
    "cmCd": "98765",
  }
]
```

```

    "tmCd": "12345",
    "ucc": "40001004",
    "tmRefId": "TEST05555130123000101"
  },
  {
    "seg": "FO",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001005",
    "tmRefId": "TEST05555130123000201"
  }
]

```

Response JSON

List of Map. Each Map corresponds to block creation record with details. The Map will contain following fields in case record is found.

Field	Type	Mandatory	Description
seg	String (3)	Yes	Same as input.
cmCd	String (5)	Yes	
tmCd	String (5)	Yes	
ucc	String (10)	Yes	
tmRefId	String (40)	Yes	
rc	Number (4)	Yes	Status code as received from UPI
umn	String (100)	Conditional	Unique mandate number assigned to the block by UPI system
ifsc	String (11)	Conditional	Client account IFSC code (as received from UPI)
accNo	String (30)	Conditional	Client account number (as received from UPI)
amtBlocked	Decimal (15,2)	Conditional	Original block amount
txnTime	Date Time	Conditional	Block creation time. In "DD-MON-YYYY HH24:MI:SS" format (IST)
matTime	Date Time	Conditional	Maturity/Expiry date of the block. In "DD-MON-YYYY HH24:MI:SS" format (IST)
totAmtDr	Decimal (15,2)	Conditional	Total amount debited in favour of CC
amtRelease	Decimal (15,2)	Conditional	Total amount released back to client.
blockStatus	String (1)	Conditional	Mandatory if rc = 100. Status of block V = Valid R = Released X = Auto released by CC M = Matured
ccReleaseReason	Number (2)	Conditional	Applicable only if rcCode = 100 and blockStatus = X (Auto release by CC). 1 - UCC not registered 2 - Bank account not registered for UCC 3 - Bank limit exceeded. 99 - Other reason

Sample Response

```

[
  {
    "seg": "CM",

```

```

    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",
    "tmRefId": "TEST05555130123000001",
    "rc": 100,
    "umn": "TESTUMN1234",
    "ifsc": "BANK0000001",
    "accNo": "0123456789",
    "amtBlocked": 500000,
    "txnTime": "16-JAN-2023 16:15:01",
    "matTime": "16-JAN-2063 16:15:01",
    "totAmtDr": 0,
    "amtRelease": 0,
    "blockStatus": "V"
  },
  {
    "seg": "FO",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001004",
    "tmRefId": "TEST05555130123000101",
    "rc": 100,
    "umn": "TESTUMN1235",
    "ifsc": "BANK0000001",
    "accNo": "0123456790",
    "amtBlocked": 500000,
    "txnTime": "16-JAN-2023 16:15:02",
    "matTime": "16-JAN-2063 16:15:02",
    "totAmtDr": 0,
    "amtRelease": 500000,
    "blockStatus": "X",
    "ccReleaseReason": 2
  },
  {
    "seg": "FO",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001005",
    "tmRefId": "TEST05555130123000201",
    "rc": 22
  }
]

```

POST /rest/<version>/inquire/blockdebitlist

This API will allow members (both CM & TM) to inquire list of all debits invoked by CC on client blocks during a given period. This API will be used by members (both CM & TM) for reconciliation and subsequently downloading missed POST /notify/blockdebit API notifications.

Request	JSON
Response	JSON

Request JSON

Map containing following fields.

Field	Type	Mandatory	Description
fromTxnTime	Date Time	Yes	In "DD-MON-YYYY HH24:MI:SS" format (IST)
toTxnTime	Date Time	Yes	In "DD-MON-YYYY HH24:MI:SS" format (IST)

Sample Request

```
{
  "fromTxnTime": "16-JAN-2023 16:00:00",
  "toTxnTime": "16-JAN-2023 17:00:00"
}
```

Response JSON

The response will contain a list of strings with each string representing a ccRefId.

Field	Type	Mandatory	Description
ccRefId	String (40)	Yes	Unique reference id assigned by CC for the debit instruction.

Sample Response

```
[
  "9876520230116R00000000100",
  "9876520230116R00000000101"
]
```

POST /rest/<version>/inquire/blockdebit

This API will allow members (both CM & TM) to inquire one or more debits identified by ccRefId invoked by CC on client blocks. The API should be used for reconciliation in conjunction with the API POST /rest/<version>/inquire/blockdebitlist. A maximum of 1000 inquiries can be submitted in a single API call.

Request	JSON
Response	JSON

Request JSON

List of ccRefId assigned by CC for the block debit.

Field	Type	Mand.	Description
ccRefId	List<String (40)>	Yes	List of reference ids ccRefId assigned by CC to the debit instruction

Sample Request

```
[
  "9876520230116R00000000100",
  "9876520230116R00000000101"
]
```

Response JSON

List of Map each representing a block debit details. Map will contain following fields.

Field	Type	Mandatory	Description
seg	String (3)	Yes	Original segment in which the block was created.
cmCd	String (5)	Yes	Primary member code
tmCd	String (5)	Yes	Trading member code
ucc	String (10)	Yes	UCC code
tmRefId	String (40)	Yes	Unique reference id assigned by trading member to the block request.
umn	String (100)	Yes	Unique mandate number as received from UPI

ifsc	String (11)	Yes	Client account IFSC code (as received from UPI)
accNo	String (30)	Yes	Client account number (as received from UPI)
amtBlocked	Decimal (15,2)	Yes	Original block amount
txnTime	Date Time	Yes	Block creation time. In "DD-MON-YYYY HH24:MI:SS" format (IST)
matTime	Date Time	Yes	Maturity/Expiry date of the block. In "DD-MON-YYYY HH24:MI:SS" format (IST)
ccRefId	String (40)	Yes	Unique reference id assigned by CC to the debit.
rc	Number (4)	Yes	Status code as received from UPI.
drAmt	Decimal (15,2)	Yes	Actual amount debited from the block.
drTime	Date Time	Yes	Actual Debit time. In "DD-MON-YYYY HH24:MI:SS" format (IST)
desc	String (100)	No	Narration for the debit transaction
totAmtDr	Decimal (15,2)	Yes	Total amount debited in favour of CC after this debit.
balAmount	Decimal (15,2)	Yes	Balance block amount against the UMN after this debit = amtBlocked - totAmtDr

Sample Response

```
[
  {
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",
    "tmRefId": "TEST05555130123000001",
    "umn": "TESTUMN1234",
    "ifsc": "BANK0000001",
    "accNo": "0123456789",
    "amtBlocked": 500000,
    "txnTime": "16-JAN-2023 16:15:01",
    "matTime": "16-JAN-2063 16:15:01",
    "ccRefId": "9876520230116R0000000100",
    "rc": 100,
    "drAmt": 1000,
    "drTime": "26-JAN-2023 19:15:01",
    "desc": "Debit for obligation",
    "totAmtDr": 2000,
    "balAmount": 480000
  },
  {
    "seg": "FO",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001004",
    "tmRefId": "TEST05555130123000101",
    "umn": "TESTUMN1235",
    "ifsc": "BANK0000001",
    "accNo": "0123456790",
    "amtBlocked": 500000,
    "txnTime": "16-JAN-2023 16:15:02",
    "matTime": "16-JAN-2063 16:15:02",
    "ccRefId": "9876520230116R0000000101",
    "rc": 100,
```

```

    "drAmt": 25000,
    "drTime": "26-JAN-2023 19:15:02",
    "totAmtDr": 50000,
    "balAmount": 0
  }
]

```

POST /<version>/inquire/regdereg

This API will allow members (both TM as well as CM) to inquire registration status of one or more UCCs in UPIX for a segment. A maximum of 1000 requests can be submitted in a single API call.

Request	JSON
Response	JSON

Request JSON

List of Map with each map representing a UCC.

Field	Type	Mandatory	Description
seg	String (3)	Yes	Segment
cmCd	String (5)	Yes	Primary member code
tmCd	String (5)	Yes	Trading member code
ucc	String (10)	Yes	UCC code

Sample Request

```

[
  {
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001"
  },
  {
    "seg": "FO",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001004"
  }
]

```

Response JSON

List of Map. Each Map corresponds to individual query record in the request JSON. In case registration details exist for the UCC then the map will contain following fields

Field	Type	Mandatory	Description
seg	String (3)	Yes	Same as input.
cmCd	String (5)	Yes	
tmCd	String (5)	Yes	
ucc	String (10)	Yes	
status	String(8)	Yes	01010100 - Successful
regStatus	String(1)	Yes	Registration Status R = Registered D = De-registered

regTime	Date Time	Yes	Registration Time. In “DD-MON-YYYY HH24:MI:SS” format (IST)
lastActivityTime	Date Time	Yes	Last Registration Activity Time. In “DD-MON-YYYY HH24:MI:SS” format (IST)
exch	String(3)	Yes	Exchange at which registered. NSE/BSE/MSE
bankList	List<Bank Account>	Yes	List of bank accounts. Minimum 1.
depo	String (8)	Yes	IN000018 = NSDL IN000026 = CDSL
dpId	String (8)	Conditional	Primary dp account DP id. Mandatory in case of NSDL
clId	String (16)	Yes	Primary dp acc client or ben id. In case of NSDL exactly 8 characters. In case CDSL exactly 16 characters.

Bank Account Structure

Field	Type	Mandatory	Description
ifsc	String (11)	Yes	Bank IFSC
accNo	String (30)	Yes	Bank acc no.
pri	String (1)	No	Y = Primary Bank Account

In case of error such as registration request not found or any other error individual map will contain following fields.

Field	Type	Mandatory	Description
seg	String (3)	Yes	Same as input.
cmCd	String (5)	Yes	
tmCd	String (5)	Yes	
ucc	String (10)	Yes	
status	String(8)	Yes	01010104 - Not found See Appendix B for other status codes.

Sample Response

```
[
  {
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",
    "status": "01010100",
    "regStatus": "R",
    "regTime": "15-Jun-2023 10:00:15",
    "lastActivityTime": "15-Jun-2023 10:00:15",
    "exch": "NSE",
    "bankList": [
      {
        "ifsc": "BANK0000001",
        "accNo": "ACCNO000001",
        "pri": "Y"
      },
      {
        "ifsc": "BANK0000002",
        "accNo": "ACCNO000002"
      }
    ]
  }
]
```

```

    ],
    "depo": "IN000018",
    "dpId": "IN000001",
    "cliId": "12121212"
  },
  {
    "seg": "FO",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001004",
    "status": "01010102"
  }
]

```

POST /<version>/inquire/segblockbal

This API will allow members (both TM as well as CM) to inquire segment wise UCC wise balance block amounts. A maximum of 1000 queries can be submitted in a single request.

Request	JSON
Response	JSON

Request JSON

List of Map with each map representing a segment UCC combination to be inquired.

Field	Type	Mandatory	Description
seg	String (3)	Yes	Segment
cmCd	String (5)	Yes	Primary member code
tmCd	String (5)	Yes	Trading member code
ucc	String (10)	Yes	UCC code

Sample Request

```

[
  {
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001"
  },
  {
    "seg": "FO",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001004"
  }
]

```

Response JSON

List of Map. Each Map corresponds to corresponding map in the request json. The Map will contain following fields.

Field	Type	Mandatory	Description
seg	String (3)	Yes	Same as input.
cmCd	String (5)	Yes	
tmCd	String (5)	Yes	
ucc	String (10)	Yes	

status	String (8)	Yes	01010100 - Success 01010104 - Not found See Appendix B for other status codes.
balAmount	Decimal(15, 2)	Conditional	Balance block amount. Applicable only if status = 01010100

Sample Response

```
[
  {
    "seg": "CM",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "10001001",
    "status": "01010100",
    "balAmount": 50000
  },
  {
    "seg": "FO",
    "cmCd": "98765",
    "tmCd": "12345",
    "ucc": "40001004",
    "status": "01010104"
  }
]
```

Member APIs

This chapter gives details of the call back API's which the members can provide. These APIs are optional and allow CCs to send various notifications to members.

POST /<version>/notify/blockrelease

This API exposed by CM as well as TM will allow exchange to optionally notify the status of previously submitted block release batch using API POST /<version>/request/blockrelease. A maximum of 1000 notifications can be submitted in a single API call.

Request	JSON
Response	JSON

Request JSON

Structure same as Response JSON of API POST /<version>/inquire/blockrelease.

Response JSON

Map acknowledging the request with following fields.

Field	Type	Mandatory	Description
status	String(8)	Yes	01010000 = Submitted Successfully

Sample Response

```
{
  "status": "01010000"
}
```

POST /<version>/notify/transfer

This API exposed by CM as well as TM will allow exchange to notify the status of previously submitted transfer batch using API POST /<version>/request/transfer. A maximum of 1000 notifications can be submitted in a single API call.

Request	JSON
Response	JSON

Request JSON

Structure same as Response JSON of API POST /<version>/inquire/transfer.

Response JSON

Map acknowledging the request with following fields.

Field	Type	Mandatory	Description
status	String(8)	Yes	01010000 = Submitted Successfully

Sample Response

```
{
  "status": "01010000"
}
```

POST /<version>/notify/blockcreate

This API exposed by CM as well as TM will allow exchange to notify the status of one or more block creations. A maximum of 1000 notifications can be clubbed in a single API call. A maximum of 1000 notifications can be submitted in a single API call.

Request	JSON
Response	JSON

Request JSON

Structure same as Response JSON of API POST /<version>/inquire/blockcreate.

Response JSON

Map acknowledging the request with following fields.

Field	Type	Mandatory	Description
status	String(8)	Yes	01010000 = Submitted Successfully.

Sample Response

```
{
  "status": "01010000"
}
```

POST /<version>/notify/blockdebit

This API exposed by CM as well as TM will allow exchange to notify whenever a debit is performed in favour of CC on client block. A maximum of 1000 notifications can be submitted in a single API call.

Request	JSON
Response	JSON

Request JSON

Structure same as Response JSON of API POST /<version>/inquire/blockdebit.

Response JSON

Map acknowledging the request with following fields.

Field	Type	Mandatory	Description
status	String(8)	Yes	01010000 = Submitted Successfully

Sample Response

```
{
  "status": "01010000"
}
```

POST /<version>/notify/regdereg

This API exposed by CM as well as TM will allow exchange to notify the new UCC registrations, modifications and de-registrations of one or more UCCs for segment. A maximum of 1000 notifications can be submitted in a single API call.

Request	JSON
Response	JSON

Request JSON

Structure same as Response JSON of API POST /<version>/inquire/regdereg.

Response JSON

Map acknowledging the request with following fields.

Field	Type	Mandatory	Description
status	String(8)	Yes	01010000 = Submitted Successfully.

Sample Response

```
{
  "status": "01010000"
}
```

Appendix A - Status Codes

- Status code shall be populated in the field “**status**” in the response. It will comprise of exactly 8 characters all of which will be numeric (0 to 9).
 - 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 field validation failure or success.
- Success code shall be populated only on successful acceptance of the request.

1. Field Identifier

Sr. No.	Field Name	Field Identifier
1	Entire Record	0101
2	Transaction Time	0116
3	Source Segment	0108
4	Target Segment	0119
5	TM Code	0104
6	UCC Client Code	0105
7	TM Reference ID	0102
8	UMN No	0103
9	IFSC Code	0109
10	Account No	0110
11	Original Amount	0111
12	Residual Amount	0114
13	Maturity Date	0118
14	Transfer Request Amount	0120

2. Validation Codes

Sr. No.	Validation	Validation Type	Validation Code	Validation performed on Field
1	Submitted successfully	Message Level	0000	NA
2	Status Request accepted successfully - 0100 Request Rejected due to higher margin utilization - 0101 Processing – 0102 Pending - 0103 Not Found - 0104	Request level	0100 0101 0102 0103 0104	Entire request

3	Minimum Required Length	Generic	0201	All Fields *
4	Maximum Required Length	Generic	0202	All Fields *
5	Range validation	Generic	0203	All Fields *
6	Mandatory Field	Generic	0204	All Fields *
7	Special Characters not allowed	Generic	0205	All Fields *
8	Data Format like Msg Id or File Name/ Date Format	Generic	0206	All Fields *
9	Minimum allowed value	Generic	0207	All Fields *
10	Maximum allowed value	Generic	0208	All Fields *
11	Invalid Value	Generic	0209	All Fields*
12	Invalid Header Record	Generic	0210	All Fields*
13	Service Unavailable	Generic	0211	NA
14	System Error	Generic	0212	NA
15	Number of records	Generic	0213	Number of records submitted is greater than configured allowed records per request
16	Duplicate Message Id	Generic	0214	All Fields *
17	Current Date	Field	0215	Current Date
18	Segment Valid segment CM FO CD SL CO	Field	0216	Segment
19	Valid Trading Member Code Should be the part of valid CM-TM link of specified segment	Field	0217	TM Code
20	Valid UCC CLI Code. Should be part of valid TM-UCC CLI Code combination of specified segment	Field	0218	Incorrect UCC CLI Code
21	Bank limit exceeded. Auto release	Request level	0219	Entire request
22	Business Date	Field	0220	Should be in the format ddmmyyyy

23	Clearing Member Code Should be valid CM Code	Field	0221	CM Code
24	Value should be blank In case of future use fields	Field	0222	Value should be blank/null
25	Custodial Participant Code Should be valid CP Code	Field	0223	CP Code
26	'TM-UCC-Segment-IFSC- Account No' combination should be matched with registration details. In case of mismatch, block will be auto release.	Request Level	0226	UCC details mismatch in registration and block creation
27	Maturity Date Maturity Date should be equal to or greater than min allowable years	Request Level	0227	Lesser Maturity Date

***** End of Document *****