

Annexure-B

User manual for API registration and data submission through API

Key Points:

1. Member have to first register for API on Member Portal (<https://uat.connect2nsccl.com/MemberPortal/>).
2. Password has to be encrypted with secret key. Secret key will be received in mail after successful registration.
3. The request and response will be in GZIP format.

Part A: Registration for API

1. Details of UAT member registration are as under:

URL : <https://uat.connect2nsccl.com/MemberPortal/>

User Id: ApiUser< Your 5 digit Member Code>

Member Code: Your 5 digit Member Code Password:

Password: Abcd@1234

If first time login, members will have to change the password and re-login with new password.

For first time login on member portal, OTP will be sent on registered mobile number or email id.

Once correct OTP is entered user will be logged in successfully.

If mobile number or email id needs to be updated, then mail us on DL-IT-ES-INSP-API@nse.co.in with the user id and new mobile number or email id.



Member TrainingExisting MembershipNSCCLProductsInvestorServices & Circulars



Sign In

User Id

Member Code

Password

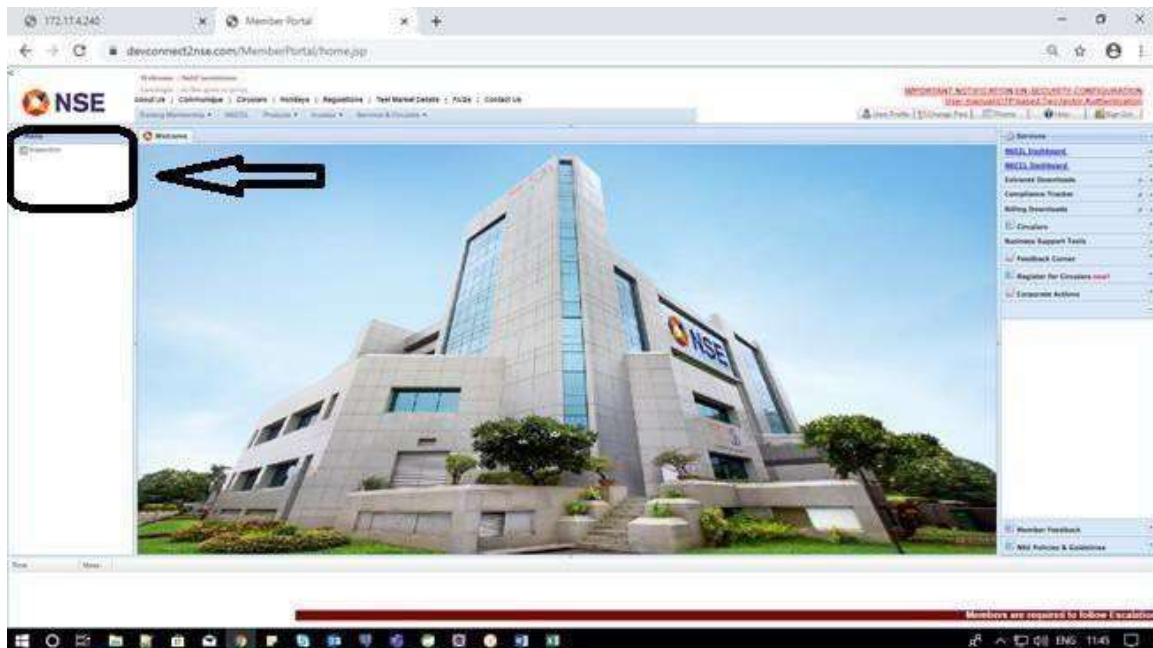
Enter Captcha - Case Sensitive

skVgkn

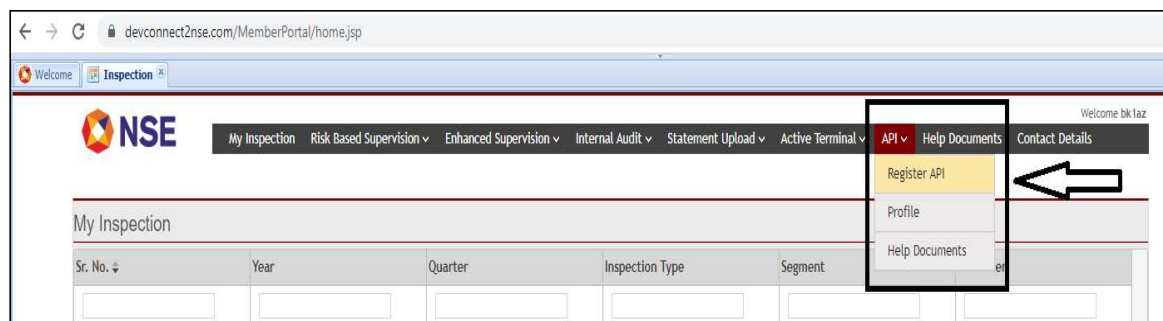
Forgot Password?

LoginCancel

- After successfully login, on left hand corner click on “Inspection” tab and proceed for API registration.



- Navigation: Inspection---> API---> Register API. Click “Register API”



- After clicking the “Register API” button and member will have to submit the following information in Registration form.

Sr. No.	Field	Validation points
1	Login Id	Login Id should be alphanumeric and 6 to 15 characters long. Special characters are not allowed.
2	Member PAN	Alpha-numeric Trading member PAN Character (10)
3	Email Id	Trading member’s Email Id
4	Phone No.	Mobile number Character (10) and should start with 9,8 or 7 only.
5	Password	Characters (12) (At least one capital character, one small character, one numeric and one special character from @,#,\$,%,^,&,* ,=)

After providing inputs for the information sought and accepting the declaration, click on “Register” button. If all details provided are valid then Registration will be successful.



Welcome 06000 dummy

My Inspection Risk Based Supervision Enhanced Supervision Internal Audit Statement Upload Active Terminal API Help Documents Contact Details Test Digital Signature

Register API

Login Id* Member PAN* Member's Email Id* Mobile No.*

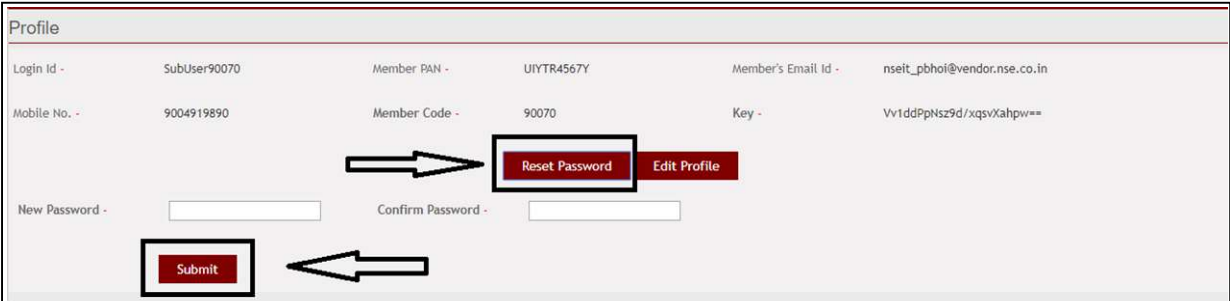
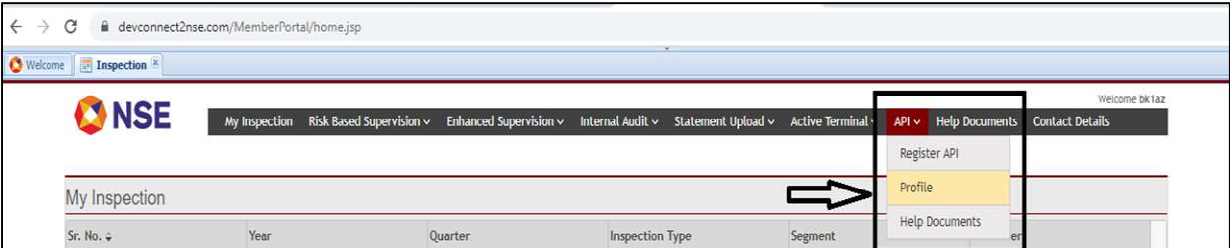
Password* Confirm Password*

Declaration* ☐ I/We hereby confirm that the information submitted to the Exchange will be accurate, correct and complete. I/we will not tamper / modify the data before submission in any manner. In case, if the same is found to be tampered / modified, I / we understand that my / our submission will be considered as rejected.

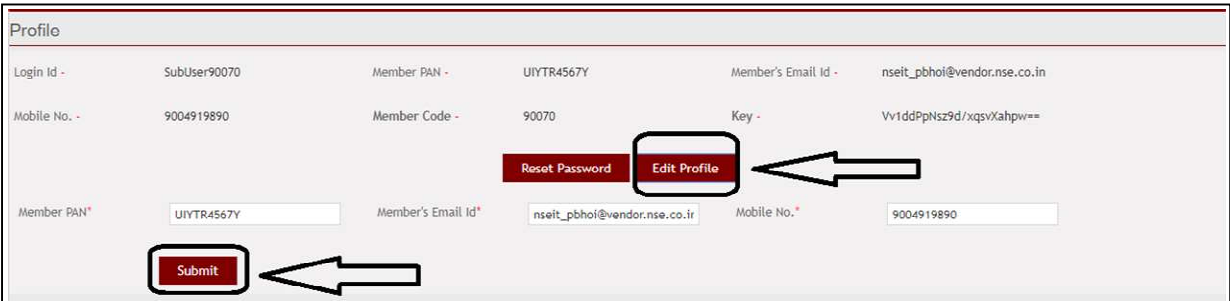
Register Reset

5. Member can view the profile by clicking “Profile” button.

Further, member can reset the Password, Email id and Phone no. by clicking “Reset” button.



6. Member can change “Member PAN”, “Member’s Email ID”, and “Mobile No.” on click of “Edit Profile” button.



After successful registration, member will receive an e-mail, on the e-mail address provided by

member while registration, which contains Login Id, Password, and Secret key.

Note: Secret key will change on every password reset.

7. Member have to encrypt its raw password by using secret key. This encrypted password need to be used while calling login API. Java and .Net Code Snippet of AES encryption is added at the end of the document.

Part B: Steps to consume API's

1. API endpoints:

Login:

<https://www.devconnect2nse.com/inspectionapi/login>

Holding Upload:

<https://www.devconnect2nse.com/inspectionapi/tradingHoldingUpload/{version}>

Cash and Cash Equivalent Upload:

<https://www.devconnect2nse.com/inspectionapi/tradingCashAndCashEquivalentUpload/{version}>

EOD Bank Balance Upload:

<https://www.devconnect2nse.com/inspectionapi/tradingEodBalanceUpload/{version}>

Logout:

<https://www.devconnect2nse.com/inspectionapi/logout>

Note:

1. Version is 1.0
2. API endpoint cannot be accessed from the browser (like google chrome, internet explorer, etc.).
3. Members may have to develop application to consume the API's.
4. Use 7Zip tool to generate .gzip.
5. For testing API's through Postmen, please refer point no. 2 to 5.
6. Snippet of AES and GZip is added at the end of this document.

2. How to login with /login API endpoint?

- Create **login text file** with below JSON code having details Member Code, Login ID, and Password.

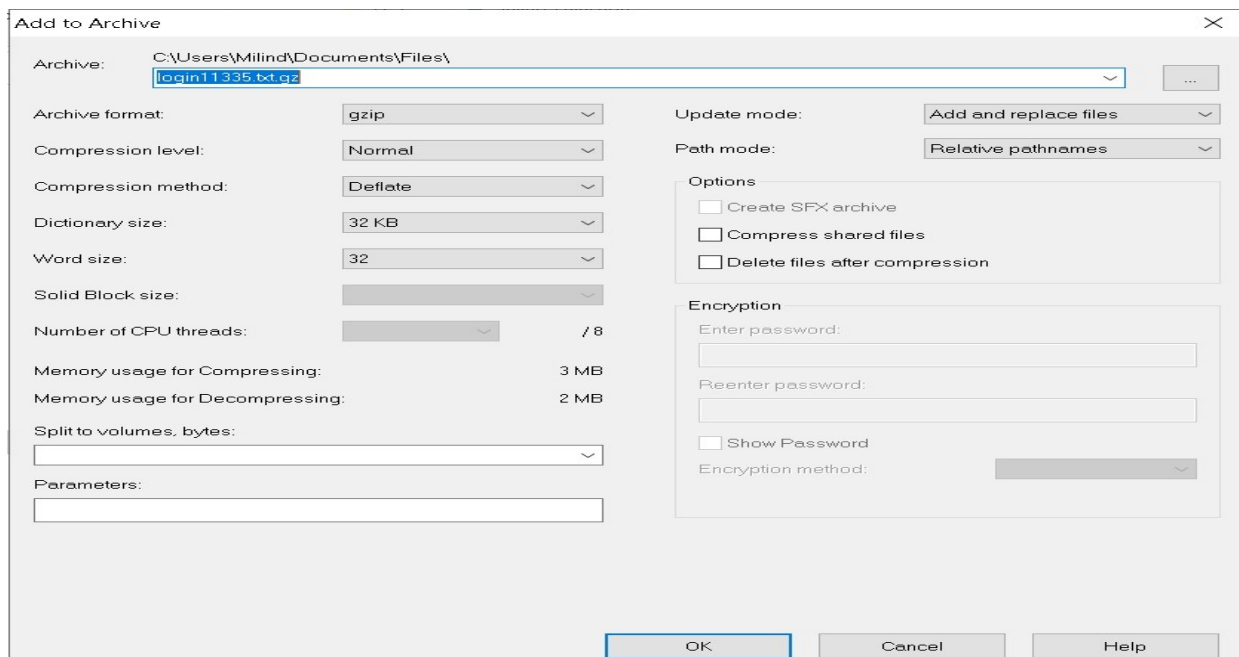
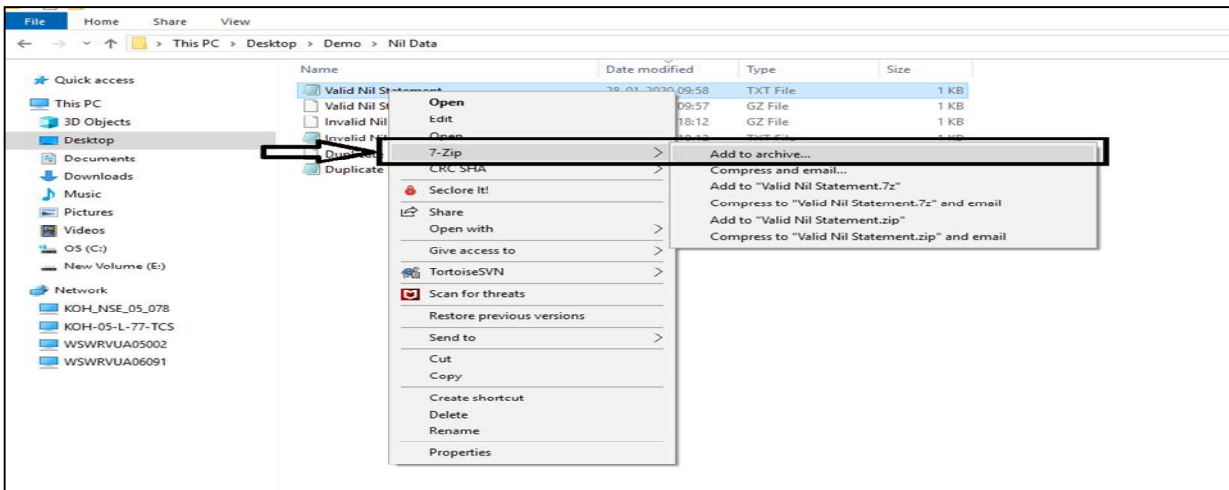
Login Request Object Structure (JSON)

```
{  
  "memberCode": "06000",  
  "loginId": "ApiUser06000",  
  "password": "qi8H8R7OM4xMUNMPuRAZxly=="  
}
```

Note: Use login Id and encrypted password.

- **How to convert above text file into gzip File format?**

Following is the diagrammatical representation of conversion of text file into .gzip file format using 7-Zip tool.



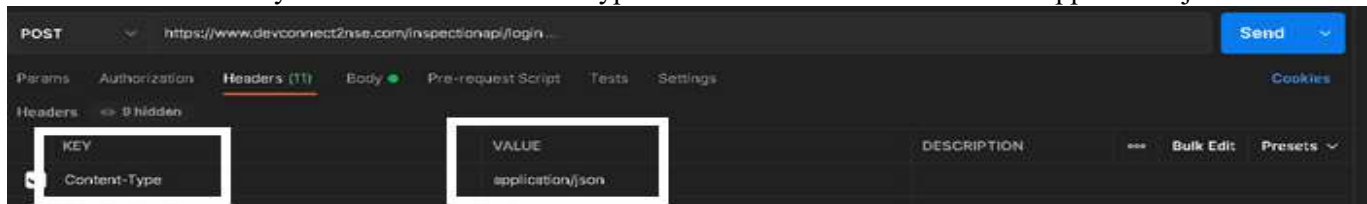
Steps to access API using postman tool:

- ❑ After opening the Postman tool/software, please select “POST” method & enter login URL in address bar.

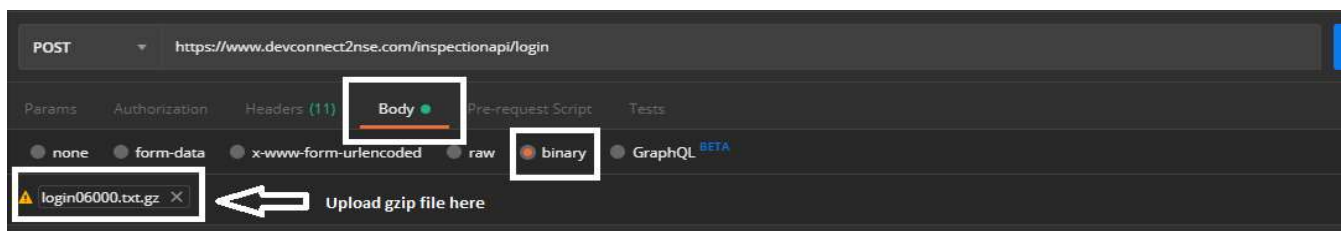
LOGIN: <https://www.devconnect2nse.com/inspectionapi/login>

Click on “Headers” tab [Check below image]

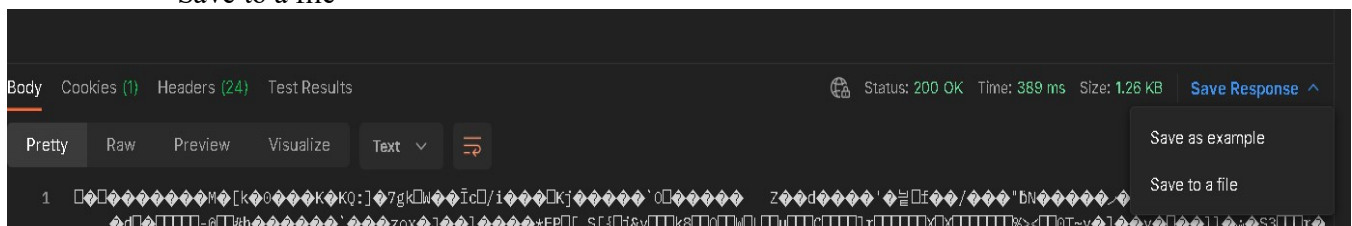
- In “Key” column enter Content-Type and in “VALUE” column enter ‘application/json’.



- Click on “Body” tab ==> Select “binary” radio button [Please check below image]
- Click on “Select File” button and upload your gzip File. [Please check below image]



- Click on “Send” button.
- The response will be in GZIP format. In postman tool click on Save Response and then click on Save to a file



- The saved response file need to be extracted with the 7zip tool and the below response file will be available.

Login Response Object Structure (JSON)

```
{
  "memberCode": "06000",
  "code": ["601"],
  "loginId": "ApiUser06000",
  "token": "eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIwOTUwMCIsImV4cCI6IjE2MjYzZmMjIiwiaWF0IjoxNjY0MDI4LCJqdGkiOiJhNWUyZiZiZiZTQwLTQ4NzItOGM4ZC0xNjE5ZjA2Nzk2NTEifQ.gGmBfpIbtP59X6gK28z4l61lOkFgOBOIGjAX-Whjky5DPJRxHJWtc-TOcDyPjLGAvnHNIvh-SZRRawKxFSNgYw",
  "status": "success"
}
```

3. How to upload holding statement data with /tradingHoldingUpload/{version} API endpoint?

- Create holding statement text file in JSON format.

Holding Statement Request Object Structure (JSON)

```
{
  "memberPan": "MBQAG3359A",
  "lastWeekDay": "26-03-2022",
  "weekDay": "25-03-2022",
  "holdingData": [{
    "recId": "1",
    "dmat": "IN30012600078641",
    "accountType": "POOL",
    "ucc": "271847",
    "clientName": "PRAVIN CHANDRA RATILAL MEHTA",
    "pan": "ACQPM0782E",
    "isin": "IN0020200286",
    "securityType": "BOND",
    "nameOfCommodity": "NA",
    "unitType": "NA",
    "totalPldgQty": "0",
    "freeBalQty": "1",
    "totalQty": "1"
  },
  {
    "recId": "2",
    "dmat": "IN30012600078641",
    "accountType": "POOL",
    "ucc": "271848",
    "clientName": "NAVIN CHANDRA",
    "pan": "ACVMM0782E",
    "isin": "IN0020200286",
    "securityType": "BOND",
    "nameOfCommodity": "NA",
    "unitType": "NA",
    "totalPldgQty": "0",
    "freeBalQty": "1",
    "totalQty": "1"
  }
]
```

- Convert above text file into gzip File format. [Check procedure shown for /login endpoint.]

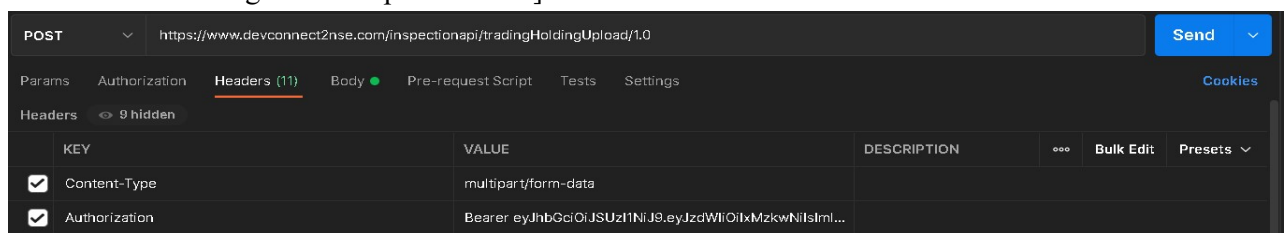
- Open Postmen select “POST” method & enter holding upload URL in address bar.

<https://www.devconnect2nse.com/inspectionapi/tradingHoldingUpload/{version}>

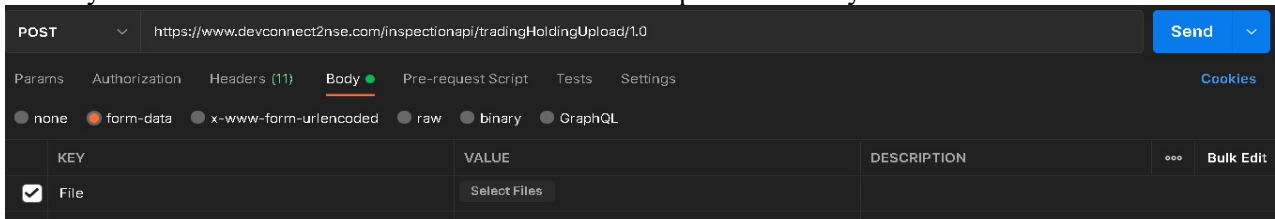
- Click on “Headers” tab [Check bellow image]

In “Key” column enter Content-Type and in “VALUE” column enter multipart/form-data.

In “Key” column enter Authorization and in “VALUE” column enter token which is generate in Login api.[Note: - You need to add a “Bearer” before token. It should have space between Bearer & Token. E.g. Bearer<space>Token]



- Click on “Body”_tab ==> Select “form-data” radio button [Please check below image]
In “Key” column enter File and also select file from dropdown in “Key”



- In “VALUE” column Click on “Select Files” button and upload your .gzip File.
- The response will be in GZIP format. In postman tool click on Save Response and then click on Save to a file



- The saved response file need to be extracted with the 7zip tool and the below response file will be available.
- Note: File Nomenclature should be in this format PAN_HS_DDMMYYYY_Seqno.gz

Holding Statement Response Object Structure (JSON)

```
{
  "hdrCode": ["601"],
  "ackId": "1133525032022213862333",
  "memberPan": "MBQAG3359A",
  "hdrStatus": "success",
  "weekDay": "25-03-2022",
  "holdingData": [{
    "recId": "1",
    "code": ["601"],
    "status": "success"
  }, {
    "recId": "2",
    "code": ["601"],
    "status": "success"
  }],
  "lastWeekDay": "26-03-2022"
}
```

Note:-

- "lastWeekDay" should be Saturday date only.
- "weekDay" should be dates from holding period only. i.e if week start from 09-12-2019 TO 14-12-2019 then "lastWeekDay" should be “14-12-2019” and "weekDay" should be from 09-12-2019 TO 13-12-2019, other dates result into the error.
- “weekDay” should have date for which data is submitted. If data is submitted for 09-12-2019 then it should be the weekday and if data is submitted for 10-12-2019 then it should be the weekday

5. How to upload EOD bank statement data with /tradingEodBalanceUpload/{version} API endpoint?

- Create holding statement text file in JSON format.

EOD Bank Balance Request Object Structure (JSON)

```
{
  "BankBalanceData": [{
    "Bank Account No": 00990610014806,
    "IFSC": "HDFC0000060",
    "Bank Account Type": ANY OTHER,
    "03-05-2021": 102007496,
    "04-05-2021": 477298321,
    "05-05-2021": 134428920,
    "06-05-2021": 167572707,
    "07-05-2021": 307590962,
    "08-05-2021": 298195332
  },
  {
    "Bank Account No": 00990630003430,
    "IFSC": "HDFC0000060",
    "Bank Account Type": "SETTLEMENT ACCOUNT",
    "03-05-2021": 196163955,
    "04-05-2021": 275308037,
    "05-05-2021": 262304676,
    "06-05-2021": 416129907,
    "07-05-2021": 212682960,
    "08-05-2021": 462731157
  }
]
}
```

- ☐ Convert above text file into gzip File format. [Check procedure shown for /login endpoint.]

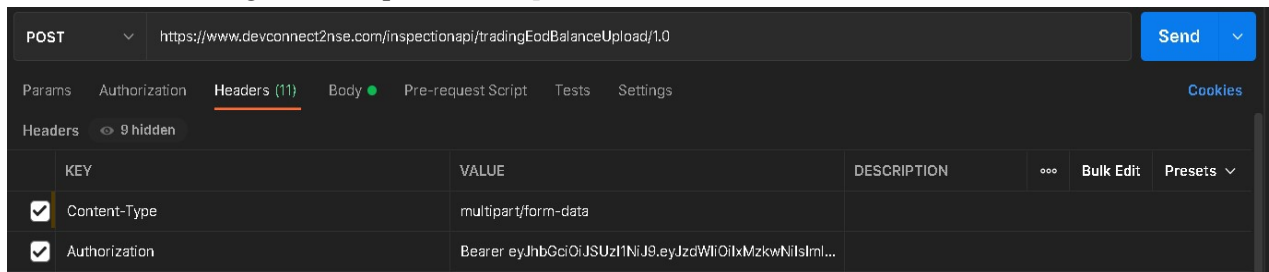
- ☐ Open Postmen select “POST” method & enter holding upload URL in address bar.

<https://www.devconnect2nse.com/inspectionapi/tradingEodBalanceUpload/{version}>

- Click on “Headers” tab [Check bellow image]

In “Key” column enter Content-Type and in “VALUE” column enter multipart/form-data.

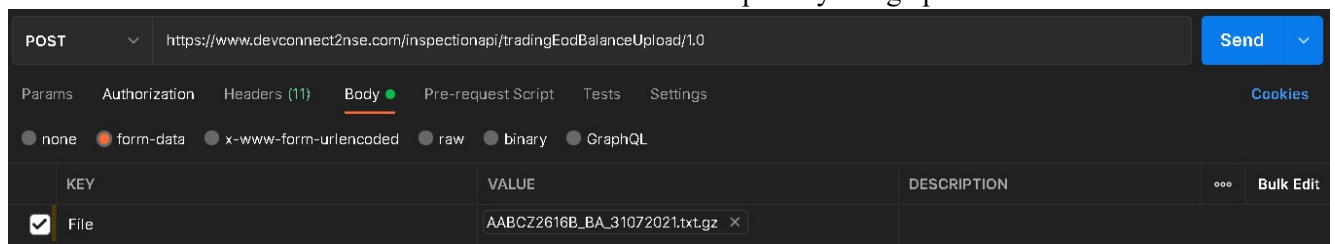
In “Key” column enter Authorization and in “VALUE” column enter token which is generate in Login api.[Note: - You need to add a “Bearer” before token. It should have space between Bearer & Token. E.g. Bearer<space>Token]



- Click on “Body”_tab ==> Select “form-data” radio button [Please check below image]

In “Key” column enter File and also select file from dropdown in “Key”

In “VALUE” column Click on “Select Files” button and upload your .gzip File.



- Click on “Send” button.
 - The response will be in GZIP format. In postman tool click on Save Response and then click on Save to a file
 - The saved response file need to be extracted with the 7zip tool and the below response file will be available.
- Note: CSV file Nomenclature should be in this format PAN_BA_DDMMYYYY_.gz

Eod Bank Balance Response Object Structure (JSON)

```
[ {
  "Bank Account No" : "00990610014806",
  "IFSC" : "HDFC0000060",
  "Bank Account Type" : "ANY OTHER",
  "03-05-2021" : "102007496",
  "04-05-2021" : "477298321",
  "05-05-2021" : "134428920",
  "06-05-2021" : "167572707",
  "07-05-2021" : "307590962",
  "08-05-2021" : "298195332",
  "Status" : "Success"
}, {
  "Bank Account No" : "00990630003430",
  "IFSC" : "HDFC0000060",
  "Bank Account Type" : "SETTLEMENT ACCOUNT",
  "03-05-2021" : "196163955",
  "04-05-2021" : "275308037",
  "05-05-2021" : "262304676",
  "06-05-2021" : "416129907",
  "07-05-2021" : "212682960",
  "08-05-2021" : "462731157",
  "Status" : "Success"
} ]
```

6. How to Logout with /logout API endpoint?

- ☐ **Create logout text file with below JSON code such as,**
Logout Request Object Structure (JSON)

```
{
  "memberCode": "12345",
  "loginId": "test"
}
```

- ☐ **Convert above text file into gzip File format [Check procedure shown for /loginendpoint.]**
- ☐ **Open Postmen select “POST” method & enter holding upload URL in address bar.**
<https://www.devconnect2nse.com/inspectionapi/logout>
- Click on “Headers” tab [Check bellow image]
In “Key” column enter Content-Type and in “VALUE” column enter application/json.
In “Key” column enter Authorization and in “VALUE” column enter token which is generate in Login api.[Note: - You need to add a “Bearer” before token. It should have space between Bearer & Token. E.g. Bearer<space>Token]

NSE Existing Membership NSCL Products Investor Services & Circulars Dashboards User Sign-Out APIUs

SERVICES Last Login: 05/04, Welcome Milind

Inspection

NSE My Inspection Risk Based Supervision Enhanced Supervision Internal Audit Statement Upload Active Terminal API Help Documents

My Inspection

Sr. No.	Year	Quarter	Inspection Type
1	2016-17	Q4	REG

Holding Statement Submit data View Submission

Cash & Cash Equivalent Balances

Bank Account Balances

Bank Statement

Collateral Segregation Submission

Yes Bank - Client Wise Submission

Monthly Regulatory Report

Collateral Shortage Submission

Reminder

NA

NSE Existing Membership NSCL Products Investor Services & Circulars Dashboards User Sign-Out APIUs

SERVICES Last Login: 05/0, Welcome Milind

Inspection

NSE My Inspection Risk Based Supervision Enhanced Supervision Internal Audit Statement Upload Active Terminal API Help Documents

MONTH* Select YEAR* Select

Search Reset

Showing latest 50 submissions. Kindly select the respective period's month and year to view old submissions.

Uploaded File	Activity	Created Time	Delete
1 CACAB3359A HS 02042022 01 SUCCESS.zip	Trading	Apr 5, 2022 12:44:34 AM	Delete
2 YXXYA3359A HS 02042022 01 FAILURE.zip	Trading	Apr 4, 2022 11:24:08 PM	
3 DWDWQ3359A HS 26032022 01 FAILURE.zip	Trading	Apr 4, 2022 7:13:37 AM	
4 DWDWQ3359A HS 26032022 01 FAILURE.zip	Trading	Apr 4, 2022 7:11:08 AM	
5 DWDWQ3359A HS 26032022 01 FAILURE.zip	Trading	Apr 4, 2022 7:07:11 AM	
6 ARDDQ3359A HS 25122021 01 SUCCESS.zip	Trading	Mar 31, 2022 8:46:38 AM	
7 ARCDQ3359A HS 25122021 01 FAILURE.zip	Trading	Mar 30, 2022 11:37:42 PM	
8 ARCDQ3359A HS 25122021 01 FAILURE.zip	Trading	Mar 30, 2022 11:31:53 PM	

Page 1 of 1 50 View 1 - 26 of 50

- As shown in images below, by clicking on file name you can download the file.
And by clicking on “Delete” button member can delete the file and re-upload the file through API.

NSE

Existing Membership

NSCCL

Products

Investor

Services & Circulars

Dashboards

User

Sign-Out

Api

SERVICES

Inspection

NSE

My Inspection

Risk Based Supervision

Enhanced Supervision

Internal Audit

Statement Upload

Active Terminal

API

Help Documents

Last Login: 05/0

Welcome Mitind

MONTH*

Select

YEAR*

Select

Search

Reset

Showing latest 50 submissions. Kindly select the respective period's month and year to view old submissions.

Uploaded File	Activity	Created Time	Delete
CACAB3359A_HS_02042022_01_SUCCESS.zip	Trading	Apr 5, 2022 12:44:34 AM	Delete
2 XYXIA3359A_HS_02042022_01_FAILURE.zip	Trading	Apr 4, 2022 11:24:08 PM	
3 DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:13:37 AM	
4 DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:11:08 AM	
5 DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:07:11 AM	
6 ARDDQ3359A_HS_25122021_01_SUCCESS.zip	Trading	Mar 31, 2022 8:46:38 AM	
7 ABCDQ3359A_HS_25122021_01_FAILURE.zip	Trading	Mar 30, 2022 11:37:42 PM	
8 ABCDQ3359A_HS_25122021_01_FAILURE.zip	Trading	Mar 30, 2022 11:31:53 PM	

Page 1 of 1

50

View 1 - 26 of 50

Download

NSE

Existing Membership

NSCCL

Products

Investor

Services & Circulars

Dashboards

User

Sign-Out

Api

SERVICES

Inspection

NSE

My Inspection

Risk Based Supervision

Enhanced Supervision

Internal Audit

Statement Upload

Active Terminal

API

Help Documents

Welcome Mitind

MONTH*

Select

YEAR*

Select

Search

Reset

Showing latest 50 submissions. Kindly select the respective period's month and year to view old submissions.

Uploaded File	Activity	Created Time	Delete
1 CACAB3359A_HS_02042022_01_SUCCESS.zip	Trading	Apr 5, 2022 12:44:34 AM	Delete
2 XYXIA3359A_HS_02042022_01_FAILURE.zip	Trading	Apr 4, 2022 11:24:08 PM	
3 DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:13:37 AM	
4 DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:11:08 AM	
5 DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:07:11 AM	
6 ARDDQ3359A_HS_25122021_01_SUCCESS.zip	Trading	Mar 31, 2022 8:46:38 AM	
7 ABCDQ3359A_HS_25122021_01_FAILURE.zip	Trading	Mar 30, 2022 11:37:42 PM	
8 ABCDQ3359A_HS_25122021_01_FAILURE.zip	Trading	Mar 30, 2022 11:31:53 PM	

Page 1 of 1

50

View 1 - 26 of 50

https://uat.connect2nsccl.com/inspectionhel8/openDocumentFile?id=UT0sg0+nB:8CMdl+8/zg==

CACAB3359A_HS_0...zip

Show all

Delete

SERVICES

Inspection



My Inspection ▾ Risk Based Supervision ▾ Enhanced Supervision ▾ Internal Audit ▾ Statement Upload ▾ Active Terminal ▾ API ▾ Help Doc

MONTH*

Select

YEAR*

Select

Showing latest 50 submissions. Kindly select the file to delete.

Showing old submissions.

	Uploaded File		Delete
1	XYXYA3359A_HS_02042022_01_FAILURE.zip	Trading	Apr 4, 2022 7:13:37 AM
2	DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:11:08 AM
3	DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:07:11 AM
4	DWDWQ3359A_HS_26032022_01_FAILURE.zip	Trading	Apr 4, 2022 7:07:11 AM
5	ABDDQ3359A_HS_25122021_01_SUCCESS.zip	Trading	Mar 31, 2022 8:46:38 AM
6	ABCDQ3359A_HS_25122021_01_FAILURE.zip	Trading	Mar 30, 2022 11:37:42 PM
7	ABCDQ3359A_HS_25122021_01_FAILURE.zip	Trading	Mar 30, 2022 11:31:53 PM
8	ABCDQ3359A_HS_25122021_01_FAILURE.zip	Trading	Mar 30, 2022 11:26:27 PM
9	ABCDQ3359A_HS_25122021_01_FAILURE.zip	Trading	Mar 30, 2022 11:20:54 PM

Success

✓

File CACAB3359A_HS_02042022_01_SUCCESS.zip deleted successfully

Ok

8. AES Code Snippet [Java]

```
public class CodeSnippet
{
    public static void main(String[] args) throws UnsupportedOperationException, NoSuchAlgorithmException,
    NoSuchPaddingException, InvalidKeyException, IllegalBlockSizeException, BadPaddingException
    {
        //Plain Text Password
        String password = "Nseitjan@123";

        //sample key is - "XBaNb0xmK2TNRIfcHA3F306Oi14HWAeYmtUd0qRheTc="
        String key = "XBaNb0xmK2TNRIfcHA3F306Oi14HWAeYmtUd0qRheTc=";
        //Key is converted to byte array
        byte[] keyByteArray = new Base64().decode(key.getBytes("UTF-8"));

        //SecretKeySpec is used to construct a SecretKey from a byte array
        SecretKeySpec secretKeySpec = new SecretKeySpec(keyByteArray, "AES");
        Cipher cipher = Cipher.getInstance("aes/ecb/pkcs5padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec);

        //pass plain text that is to be encrypt
        String encrypt = (new Base64()).encodeAsString(cipher.doFinal(password.getBytes()));

        //actual key in base64 format
        System.out.println("encrypted string:" + encrypt);
    }
}
```

9.Gzip Code Snippet [Java]

```
public class NetClientGet
{
    public static void main(String[] args) throws IOException
    {
        byte[] body = readTextFile();
        System.out.println("gzip bytes "+body);
    }
    public static byte[] readTextFile() throws IOException
    {
        //LoginJson.text contains json login Request example listed bellow
        //{
        // "memberCode": "06000",
        // "loginId": "SubUser06000",
        // "password":"sMzvV6fvENaQfRedWYd07w=="
        //}

        InputStream inputStream = new FileInputStream("<path>\\login06000.txt");
        @SuppressWarnings("resource")
        BufferedReader bufferReader = new BufferedReader(new InputStreamReader(inputStream));
        String line = bufferReader.readLine();
        StringBuilder stringBuffer = new StringBuilder(); while (line != null)
        {
            stringBuffer.append(line).append("\n"); line = bufferReader.readLine();
        }
        //Creates a gzip file compressString(stringBuffer.toString());
        //create gzip compress bytes.
        return compress(stringBuffer.toString());
    }

    public static byte[] compress(final String stringToCompress)
    {
        try (final ByteArrayOutputStream baos = new ByteArrayOutputStream();
            final GZIPOutputStream gzipOutput = new GZIPOutputStream(baos))
        {
            gzipOutput.write(stringToCompress.getBytes(StandardCharsets.UTF_8)); gzipOutput.finish();
            return baos.toByteArray();
        } catch (IOException e)
        {
            throw new UncheckedIOException("Error while compression!", e);}
    }

    public static String compressString(String str) throws IOException
    {
        if (str == null || str.length() == 0)
        {
            return str;
        }
        BufferedWriter writer = null;
        try{
            File file = new File("your.zip");
            GZIPOutputStream zip = new GZIPOutputStream(new FileOutputStream(file)); writer = new
            BufferedWriter(new OutputStreamWriter(zip, "UTF-8")); writer.append(str);
        }
        Finally{
            if(writer != null){ writer.close();
            }
            }
        return str;
    }
}
```

10. AES Code Snippet [.Net]

```
using System;
using System.Security.Cryptography; using System.Text;

namespace AES256{
class Program{
private static string GetString(byte[] b)
{
return Encoding.UTF8.GetString(b);
}
static void Main(string[] args){
byte[] data = Encoding.UTF8.GetBytes("NseitJan@201");
byte[]a = Convert.FromBase64String("AAECAwQFBgcICQoLDA0ODw==");
Console.WriteLine("Key : {0}", GetString(a));
byte[] enc = Encrypt(data, a);
string result = Convert.ToBase64String(enc);
Console.WriteLine("Encrypted text", result);
byte[] dec = Decrypt(enc, a);
Console.WriteLine("Encrypted : {0}", GetString(enc));
Console.WriteLine("Decrypted : {0}", GetString(dec));
// Console.ReadKey();
}
public static byte[] Encrypt(byte[] data, byte[] key){
using (RijndaelManaged csp = new RijndaelManaged())
{
csp.KeySize = 256;
csp.BlockSize = 128; csp.Key = key;
csp.Padding = PaddingMode.PKCS7;
csp.Mode = CipherMode.ECB;
CryptoTransform encrypter = csp.CreateEncryptor();
return encrypter.TransformFinalBlock(data, 0, data.Length);
}
}
private static byte[] Decrypt(byte[] data, byte[] key){
using (RijndaelManaged csp = new RijndaelManaged())
{
csp.KeySize = 256;
csp.BlockSize=128;
csp.Key = key;
csp.Padding = PaddingMode.PKCS7;
csp.Mode = CipherMode.ECB;
CryptoTransform decrypter = csp.CreateDecryptor();
return decrypter.TransformFinalBlock(data, 0, data.Length);
}
}
}
}
```

Technology Used:

Language: - java version "11.0.12" 2021-07-20 LTS

*****End of document*****