

EXTRANET API

User manual for API registration

Key Points:

1. Members have to first register for API on Member Portal

Production URL: (<https://ims.connect2nsccl.com/MemberPortal>)

Note: If the member has 10 users under him, then all the 10 users have to perform the steps mentioned in the user manual for self-user generation / maintenance for Extranet API in Production

2. Password has to be encrypted with secret key. Secret key will be received in mail after successful registration.

PART A: REGISTRATION FOR API

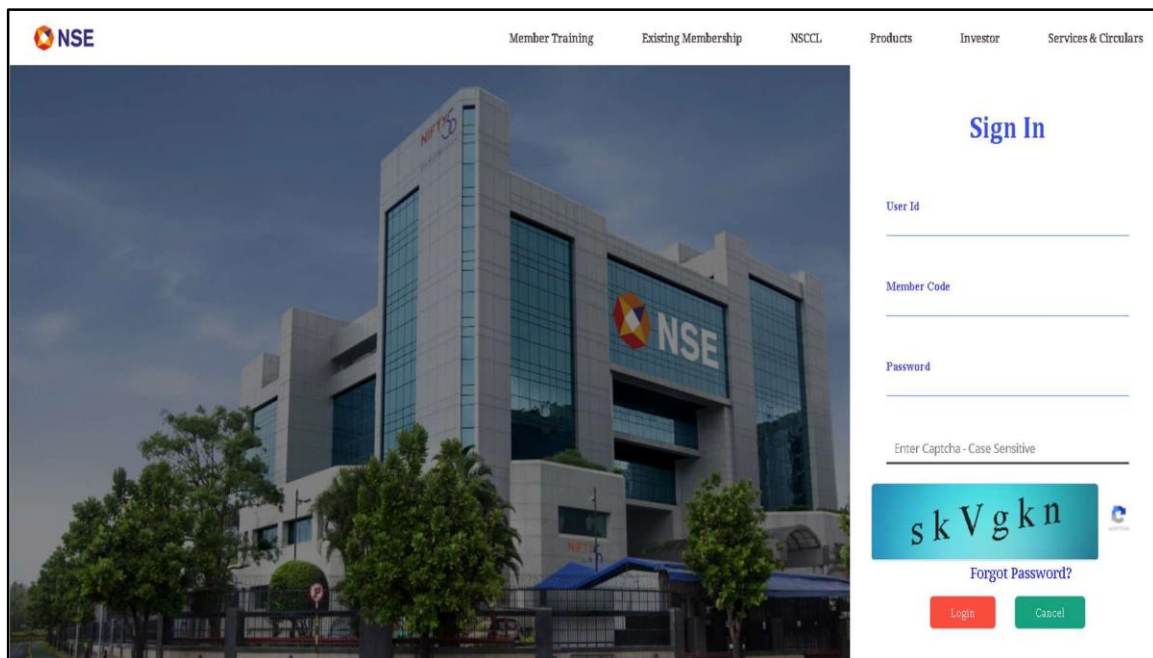
1. Details of member registration are as under:
URL: <https://ims.connect2nsccl.com/MemberPortal>

User Id: Member User ID

Member Code: 5 digit Member Code

Password: Member Password

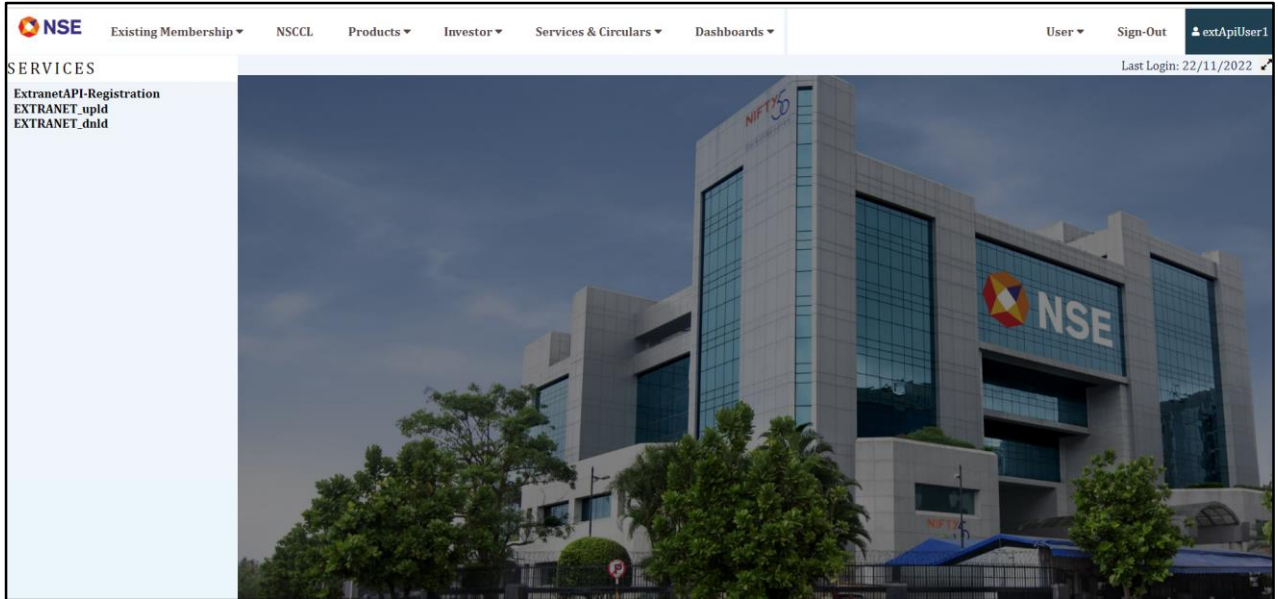
If first time login, members will have to change the password and re-login with new password. For first time login on member portal, OTP will be sent on registered mobile number or email id. Once correct OTP is entered user will be logged in successfully.



The screenshot shows the NSE Member Portal login interface. At the top, there is a navigation bar with links: Member Training, Existing Membership, NSCCL, Products, Investor, and Services & Circulars. The main content area is split into two sections. On the left, there is a large image of the NSE building. On the right, there is a login form with the following elements:

- A "Sign In" heading.
- Input fields for "User Id", "Member Code", and "Password".
- A captcha field with the text "Enter Captcha - Case Sensitive" and a visual captcha showing the characters "s k V g k n".
- A "Forgot Password?" link.
- Two buttons: "Login" (red) and "Cancel" (green).

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



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

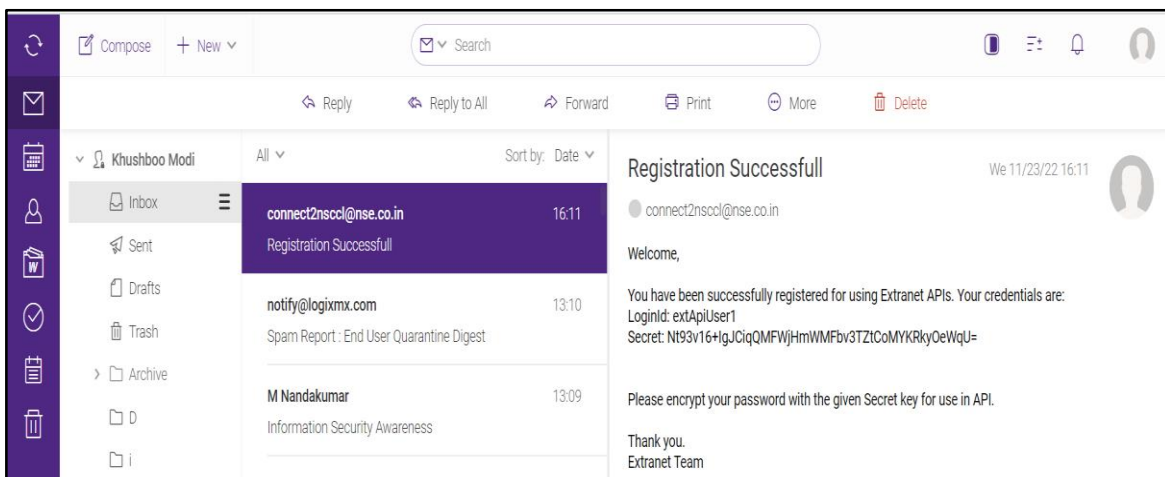
Sr. No.	Field	Validation points
1	Login Id	It will be auto-populated and will be same as the Member User ID
2	Member’s Email Id	Member’s Email Id and should be a valid email id.
3	Member’s Mobile No.	Mobile number Character (10) and should start with 9, 8 or 7 only.
4	Password	Characters (12) (At least one capital character, one small character, one numeric and one special character from @, #, \$, %, ^, &, *, =)
5	Confirm Password	Should be same as the Password

The screenshot shows the NSE Extranet API-Registration form. The top navigation bar is the same as in the previous image. The left sidebar shows the 'ExtranetAPI-Registration' tab selected. The main content area displays the 'Register Api' form. The form has five fields: Login ID (auto-populated with 'extApiUser1'), Member's Email Id (placeholder: 'i.e - example@mail.com'), Member's Mobile No. (placeholder: 'Mobile No.'), Password, and Confirm Password. A green 'Register' button is located at the bottom of the form.

After providing inputs for the information sought, click on “Register” button. If all details provided are valid then Registration will be successful and user will be directed to a page confirming the registration.



Also, 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.



4. In case the user is not registered then on clicking the “ExtranetAPI-Registration” button system will direct the user to the Registration page

ExtranetAPI-Registration → Register Api

A screenshot of the "Register Api" form in the NSE Extranet API interface. The form has the following fields: "Login ID*" (with value "extApiUser1"), "Member's Email Id*" (with value "i.e - example@mail.com"), "Member's Mobile No.*" (with value "Mobile No."), "Password*", and "Confirm Password*". There is a green "Register" button at the bottom. The top navigation bar includes "Existing Membership", "NSCL", "Products", "Investor", "Services & Circulars", "Dashboards", "User", "Sign-Out", and "extApiUser1". The left sidebar lists "SERVICES" with "ExtranetAPI-Registration", "EXTRANET_upld", and "EXTRANET_dnlld".

Once a member is registered the Member can view the profile by clicking “ExtranetAPI-Registration” button, which will redirect the member to the Profile page.

ExtranetAPI-Registration → Edit Profile

Edit Profile

Member Code

08081

Login ID

extApiUser1

Member's Email Id*

sringe@nseit.com

Member's Mobile No.*

9894561230

Key

AS(DJ_9812l@_ASDHHbsd723rbndv7uib345n

Submit

Change Password

Password*

Password

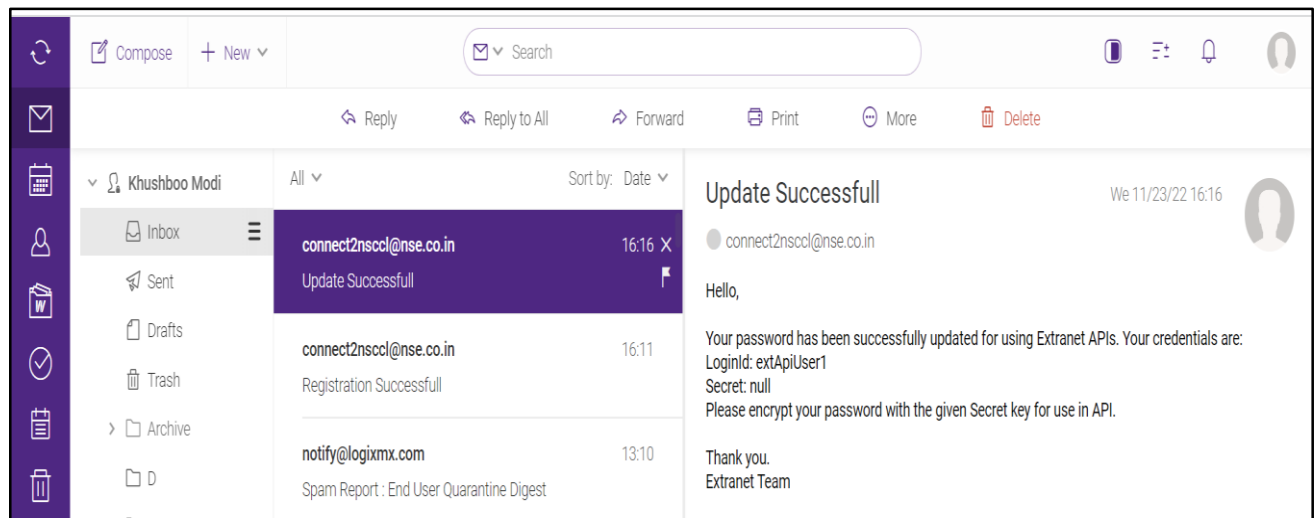
Confirm Password*

Confirm Password

Submit

- User can reset the Password, by entering the new Password and Confirm Password and clicking on Submit button.

If password reset then an email will go to the user intimating him that his password has been reset.



- User can update the Member's Email id and Member's Mobile no and click on Submit.
 On successfully editing the email id and mobile no system will display message as 'Update Successful'.

7. User has to encrypt its raw password by using secret key. This encrypted password needs 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: API End Points

1. Login:
<https://www.connect2nse.com/extranet-api/login/{version}>
 e.g.: <https://www.connect2nse.com/extranet-api/login/1.0>
2. Logout
<https://www.connect2nse.com/extranet-api/logout/{version}>
 e.g.: <https://www.connect2nse.com/extranet-api/logout/1.0>
3. Get Member Files/Folders
<https://www.connect2nse.com/extranet-api/member/content/{version}>
 e.g.: <https://www.connect2nse.com/extranet-api/member/content/1.0>
4. Download File
<https://www.connect2nse.com/extranet-api/member/file/download/{version}>
 e.g.: <https://www.connect2nse.com/extranet-api/member/file/download/1.0>
5. Get Common File/Folders
<https://www.connect2nse.com/extranet-api/common/content/{version}>
 e.g.: <https://www.connect2nse.com/extranet-api/common/content/1.0>
6. Download Common File
<https://www.connect2nse.com/extranet-api/common/file/download/{version}>
 e.g.: <https://www.connect2nse.com/extranet-api/common/file/download/1.0>

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 - "XBaNb0xmK2TNRIfcHA3F306Oi14HW AeYmtUd0qRheTc="
        String key = "XBaNb0xmK2TNRIfcHA3F306Oi14HW AeYmtUd0qRheTc=";
        //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. 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