CionSystems

Cloud Identity Minder
Authentication & Self-Service

Cloud Identity Minder Authentication WebService Usage Guidelines

CionSystems

Table of Contents

User Authentication Service Functionality

User Authentication service is Web Service exposed as part of CIM Self-service application.

This web service is exposed to external clients to authenticate the users with CIM Self-service application. The external client just needs to provide User Name, password as first factor of Authentication and the Second factor to authenticate will be to validate using any one among the following options

1. Security Question with Answers
2. OTP to Email (`EmailPinNumber`)
3. OTP to Mobile (`SMSPinNumber`)

The web service will take this information validate against different data sources like Active Directory / Open Ldap / Office365 based on the Data Source mechanism chosen in CIM Self-service application.

WebService URL:

First thing is to add the WebService Reference to Client Application.

URL: http://IPAddress/SelfService/Resources/Services/UserAuthenticationService.asmx

Different Methods Exposed by the Service

1. AuthenticateUserAcct (New)
2. GetUserDetails
3. UserEvent
4. ValidateTwoFactorAuthentication
5. ValidateTwoFactorRequest (New)

Methods Exposed For Multifactor Authentication

1. AuthenticateUserAcct (New)
2. ValidateTwoFactorRequest (New)

AuthenticateUserAcct

This method will first validate the Username and password against Ldap Source as First level of authentication and once it is verified it will check if **Two Factor Authentication is enable** in CIM application if Yes then it will returns the options for the Second level of Authentication Which are 1.Security Questions configured for that user 2.EmailPinNumber 3.SMSPinNumber.User can select any one among these options as a second factor of authentication and validate.

Note : Password should be encrypted and send using Cion EncryptionManager dll explain later in this document.

Small sample code to make use of the Webservice Api methods

```
ServiceReference1.UserAuthenticationServiceSoapClient clnt = new
ServiceReference1.UserAuthenticationServiceSoapClient();
clnt.Endpoint.Address = new System.ServiceModel.EndpointAddress(txtWebServiceUrl.Text);
req = new ServiceReference1.UserAuthenticationRequest();
req.User = new ServiceReference1.User() { UserName = txtUserName, Password = pass };
resp = clnt.AuthenticateUserAcct(req);
```

Two factor is enable in CIM

If first level of authentication is Successful then response object will return the following properties

EnableTwoFactorAuthentication true

ResponseStatus Property will hold few more properties like AvailableTwoFactors , Exception, Message , StatusCode , TwoFactorExist , VerifiedTwoFactorResp

AvailableTwoFactors : If two factor is enable then this property will have values else it will be none

If SMTP and SMS Settings are configured it will return

SecretQuestions,EmailPinNumber,SMSPinNumber

If SMTP is configured and SMS is not Configured it will return

SecretQuestions,EmailPinNumber

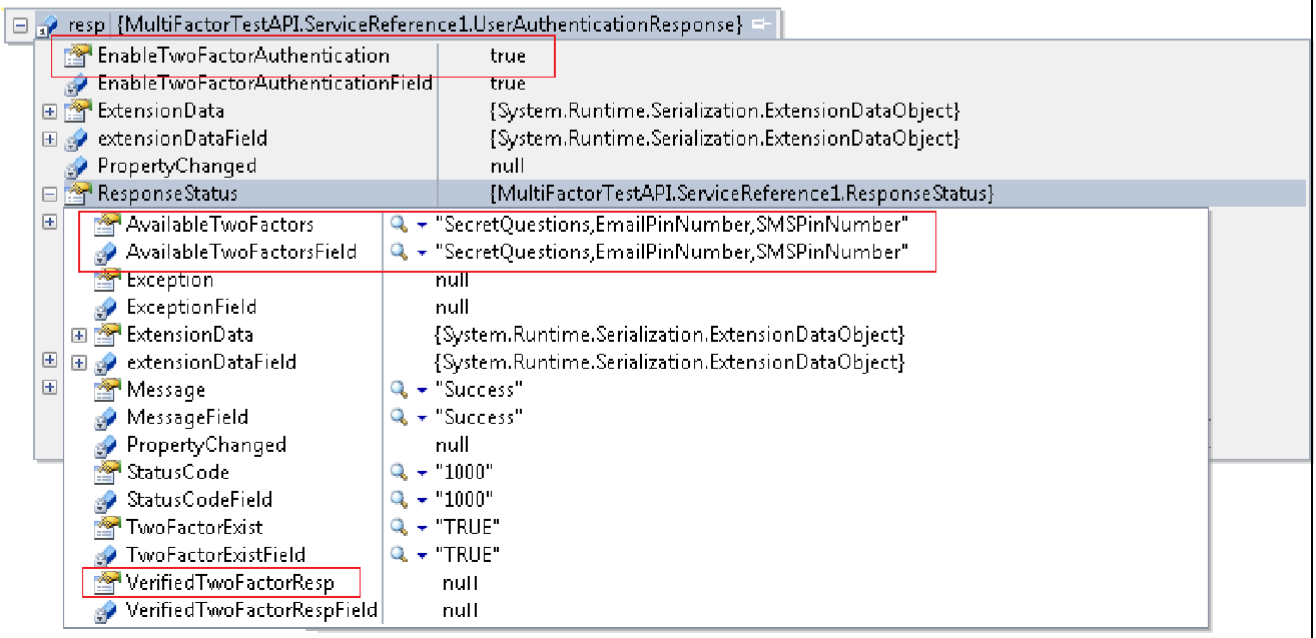
If SMS is configured and SMTP is not Configured it will return

SecretQuestions,,SMSPinNumber

If both SMS and SMTP are not Configured it will return

SecretQuestions

ResponseStatus Property in case of Successfull First Factor



Exception Property : This will be null in case of success and will hold exception details in case of failure screen shot attach below for reference.

StatusCode Property : This will return "1000" in case of Successful First factor Authentication and "1001" in case of failure.

Message Property : If StatusCode return "1000" ,This will have value as "Success" in case of "1001" this will have value "Fail" in case of failure.

TwoFactorExist Property : This will be "TRUE" if **Two Factor Authentication is enable** in CIM application and it will be null if two factor is not enabled in CIM.

VerifiedTwoFactorResp Property : This property will initially be null , it will have value once 2nd factor Authentication is success full. It will hold the value based on the 2nd factor authentication mechanism selected like SecurityQuestion or EmailPinNumber or SmsPinNumber.

After First level of Successful Authentication User will get 2nd Factor authentication options

1. SecurityQuestion
2. EmailPinNumber
3. SmsPinNumber

User can select any one and can go for the 2nd level of Authentication.This time **`AuthenticateUserAcct`** method is called with one extra parameter `SelectedTwoFactors (This property will hold the above options selected by the user.`

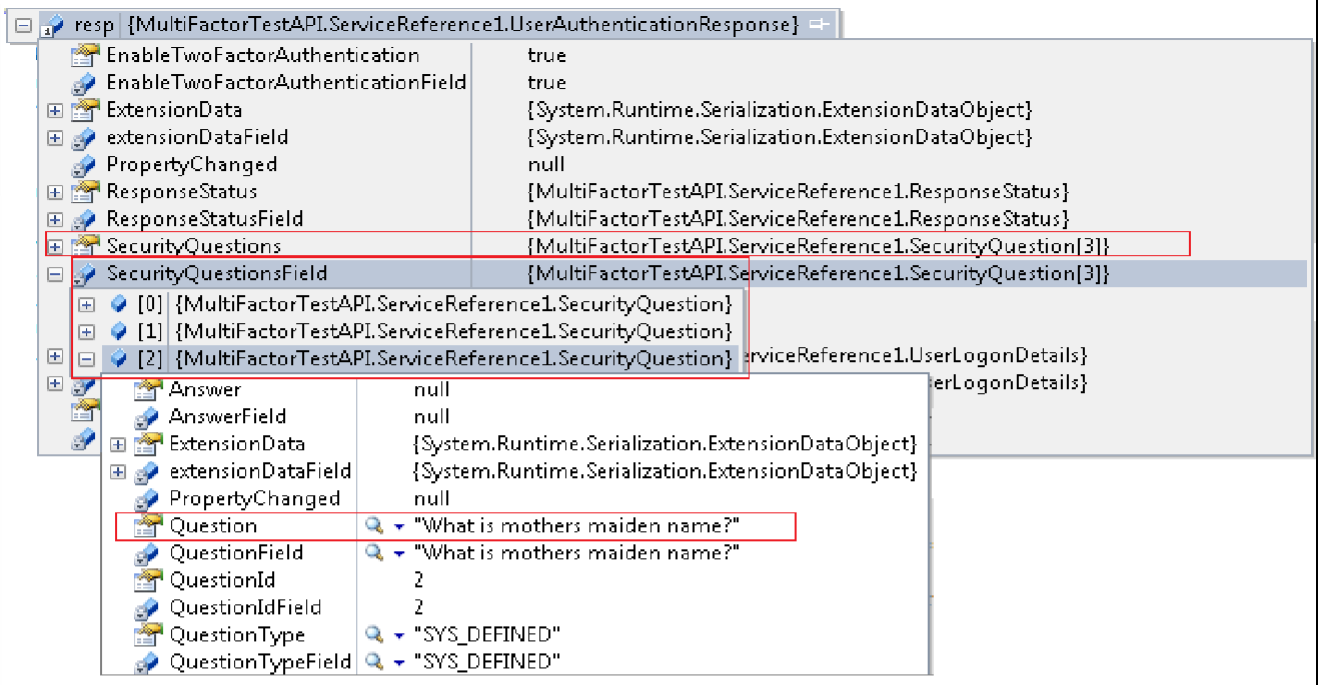
The Output of this method depends upon Configurations in Cloud Identity Minder Application. If Two Factor Authentication Type is

1. If Two Factor Authentication Type is SecurityQuestions then it will return Security Questions collection
2. If Two Factor Authentication Type is SMSPinNumber then OTP Code will generate and sends it to User mobile number.
3. If Two Factor Authentication Type is EmailPinNumber then OTP Code will generate and sends it to User registered email id.

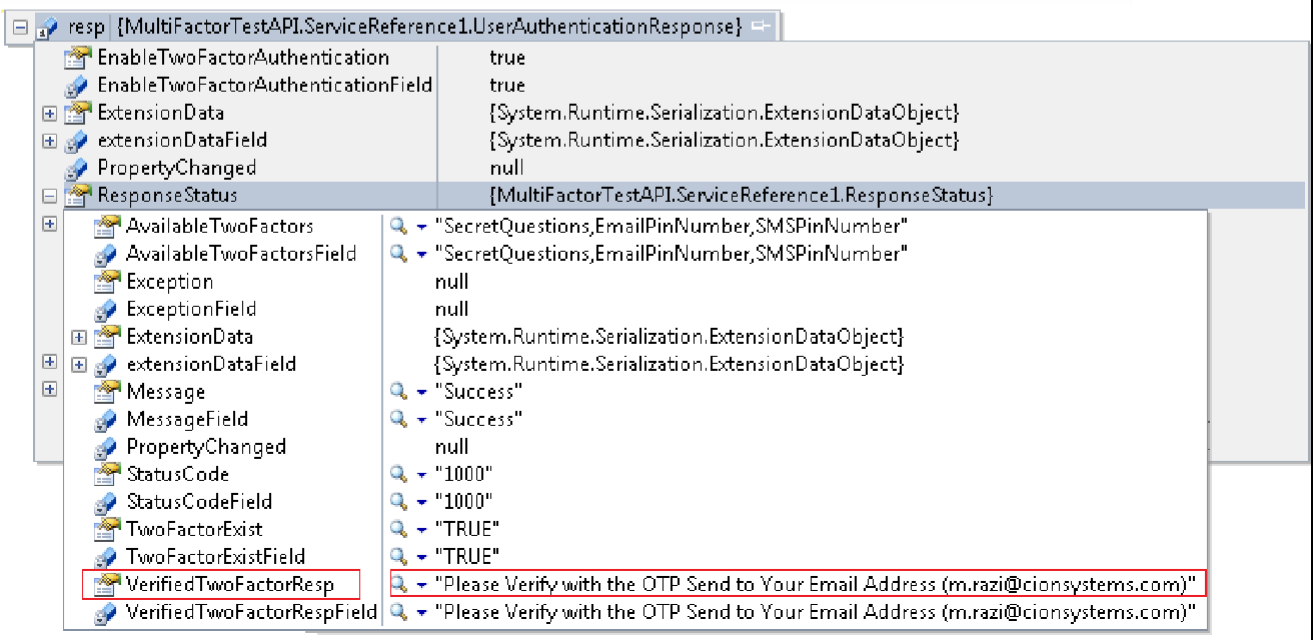
Piece of Sample Code

```
ServiceReference1.UserAuthenticationServiceSoapClient clnt = new
ServiceReference1.UserAuthenticationServiceSoapClient();
clnt.Endpoint.Address = new System.ServiceModel.EndpointAddress(txtWebServiceUrl.Text);
req = new ServiceReference1.UserAuthenticationRequest();
req.User = new ServiceReference1.User() { UserName = txtUserName.Text, Password = pass,
SelectedTwoFactors = selectedTwofactor };
resp = clnt.AuthenticateUserAcct(req);
```

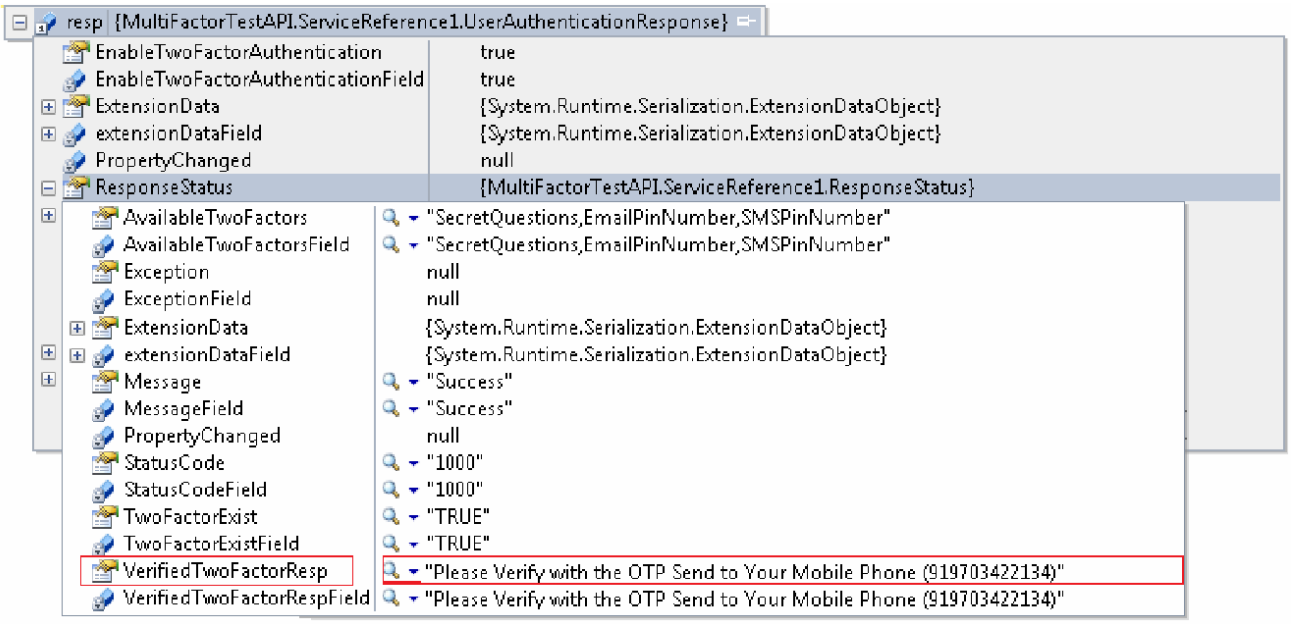
SecurityQuestion : If user selects this option , Webservice api will return the SecurityQuestions which are configured in the CIM application and corresponding **SecurityQuestions** property will hold the questions in an array.



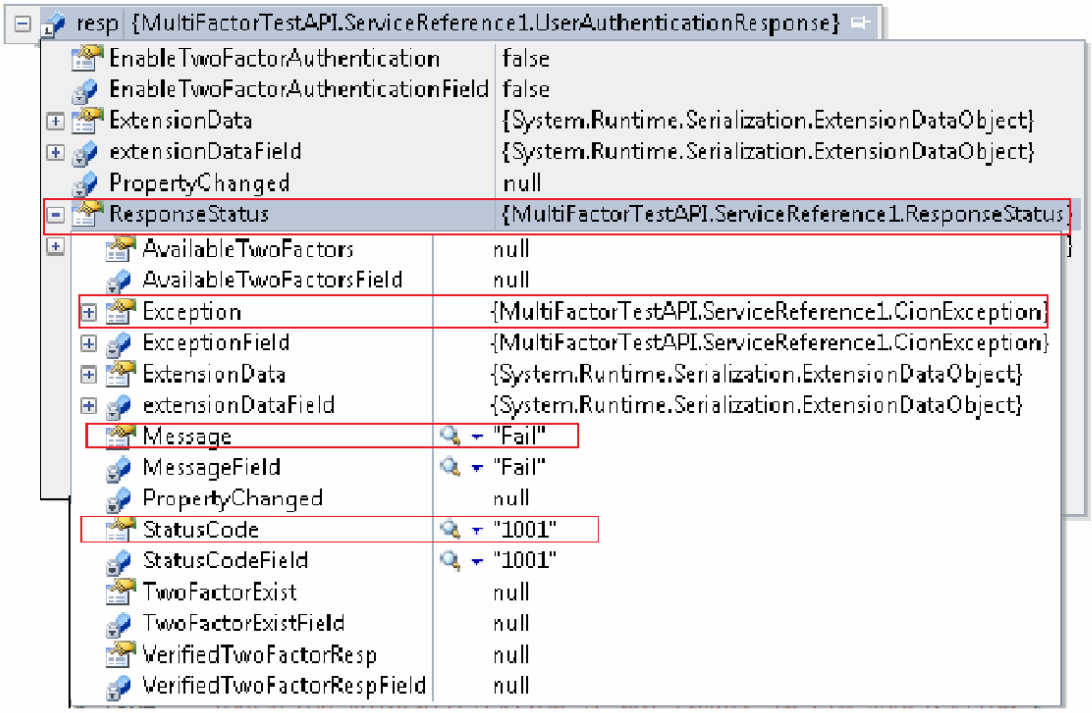
EmailPinNumber : If user selects this option , Webservice api will generate the otp pin and send to the user registered email address and **VerifiedTwoFactorResp property** will have the registered email address of the user as seen in the screen shot below



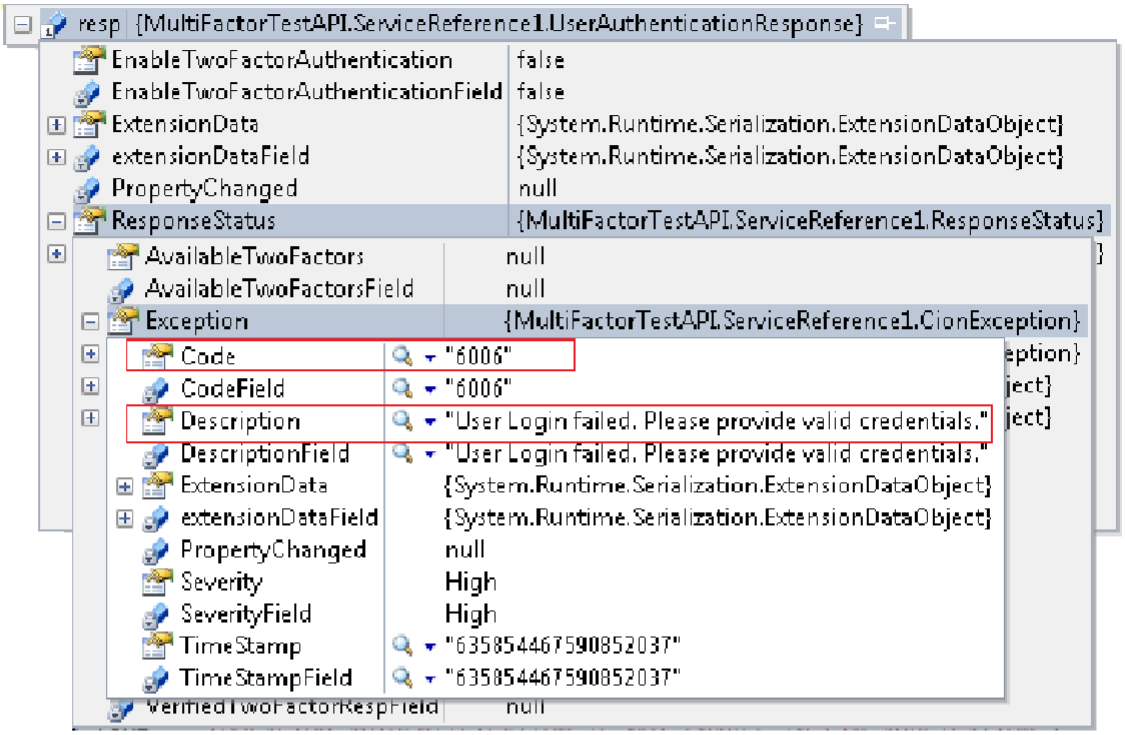
SMSPinNumber : If user selects this option , Webservice api will generate the otp pin and send to the user registered Mobile Phone and **VerifiedTwoFactorResp property** will have registered phone no of the user as seen in the screen shot below



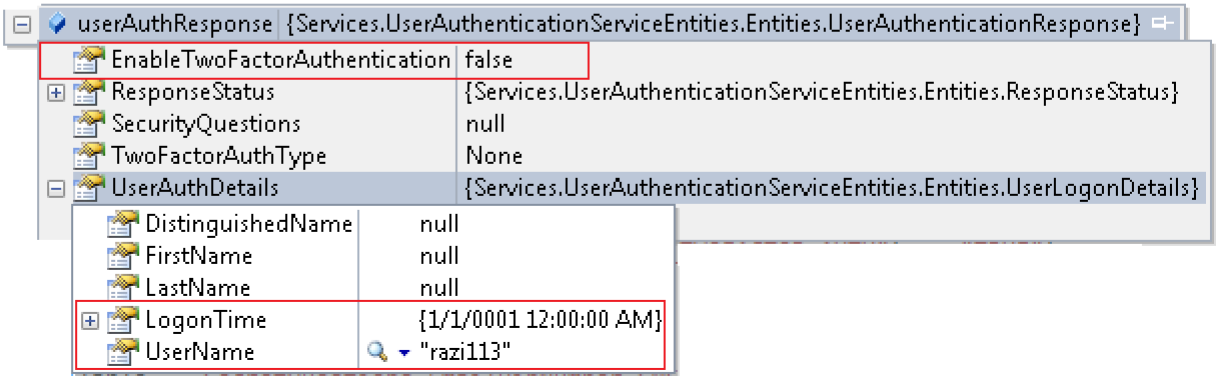
ResponseStatus Property in case of Failure



Exception Property in case of Failure



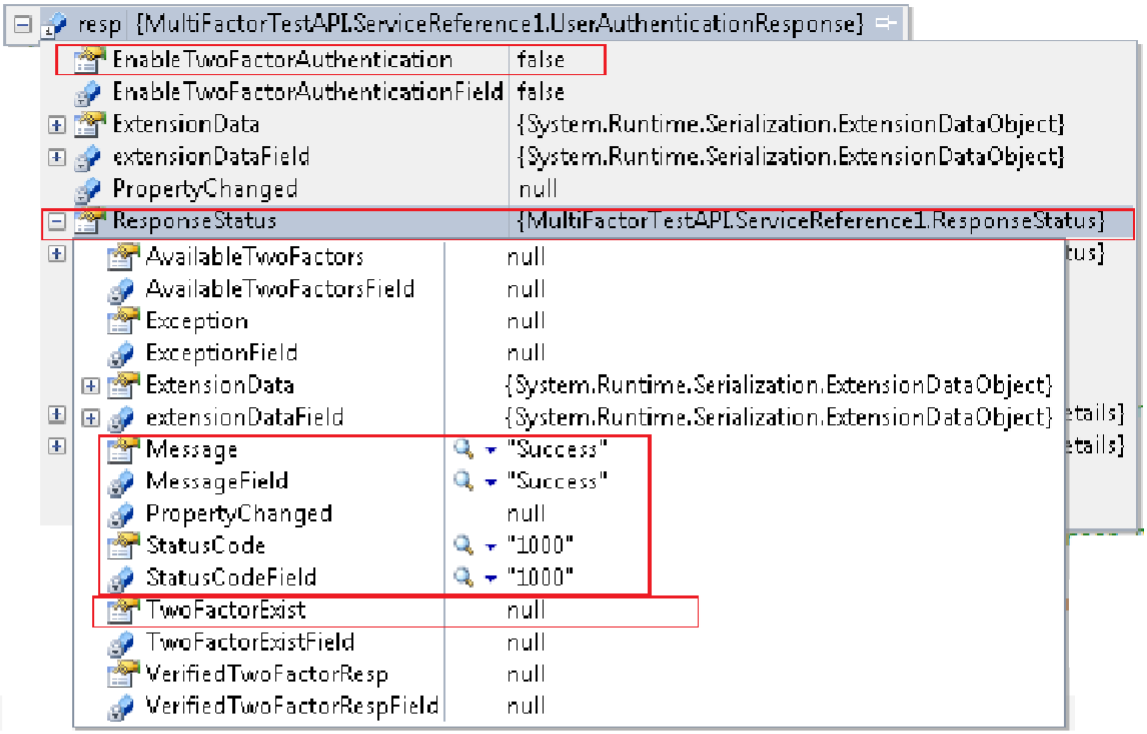
Two Factor Authentication Type is not enable in Cloud Identity Minder application, then this will only return UserName, LogonDateTime. Screen shots below for reference.



EnableTwoFactorAuthentication will be false

ResponseStatus : The Properties of Response Status like AvailableTwoFactors, TwoFactorExist , VerifiedTwoFactorResp , Exception will hold null values. Message will have “Success” or “Fail”, StatusCode will have “1000” or “1001”.

First Factor Verifed but Two Factor is not Enable



Code Usage to use AuthenticateUserAcct Webservice Api methods

Input:

AuthenticateUserAcct method will accept UserAuthenticationRequest entity as Input Parameter. The UserAuthenticationRequest entity has following properties.

`User, SecurityQuestion[], UserAuthenticationToken , SMSPinNumber. EmailPinNumber`

Note : SecurityQuestion [], UserAuthenticationToken , SMSPinNumber , EmailPinNumber these properties will not use in AuthenticateUserAcct method if send with 2 param (Username,Password), it will be used in `ValidateTwoFactorRequest` method once first factor gets verified.

User property contains UserName and Password. These two parameters are pass as input parameters for AuthenticateUserAcct method to validate first factor.

UserName and Password sends as Input Parameters with Encryption using CionEncryptionManager.

Output:

AuthenticateUserAcct method will return UserAuthenticationResponse entity as response. The UserAuthenticationResponse has following properties.

The Output of this method depends upon Configurations in Cloud Identity Minder Application. If Two Factor Authentication Type is
SecurityQuestions then it will return Security Questions
Two Factor AuthenticationType is SMSPinNumber then OTP Code will generate and sends it to User mobile number.
Two Factor Authentication Type is EmailPinNumber then OTP Code will generate and sends it to User registered email id.

If any of these two options(SecurityQuestions or SMSPinNumber or EmailPinNumber) are not selected in Cloud Identity Minder Two Factor Authentication then return Output as User Name, and LogonDateTime.

SecurityQuestion[ ] – This is Security Questions Collection. Once User name and password is validated, service will return the security questions which are configured by user in Cloud Identity Minder application. These question along with answers needs to attach in Request object and send to Service for Two Factor authentication.

EnableTwoFactorAuthentication – This is Boolean value. If Two Factor Authentication is enabled in Cloud Identity Minder application then returns true otherwise false.

UserAuthenticationToken – This is a unique token generated by server and sends to client. The same Token should send to Server as part of request for Two Factor authentication. If Two Factor Authentication is disabled in Cloud Identity Minder application then UserAuthenticationToken value will not return by the service.

TwoFactorAuthType – This property returns Type of TwoFactor Authentication configured in Cloud Identity Minder application. Possible values are Security Questions or SMSPinNumber and None if disabled two factor authentication.

UserLogonDetails – This property contains UserName, FirstName, LastName, LogonTime and DistinguishedName

UserName – This UserName is part of request for further method (ValidateTwoFactorAuthentication).

ResponseStatus –This property contains few Sub Properties. They are,

a). **StatusCode** - This will provide Status code as 1000, 1001 and 1003.
b). **Message** – This will provide Message for Success/Fail/Error.
c). **Exception** - This property contains few sub properties theyare,

i). Code - This property will provide error code. Ex: 6000, 6001... So on.
ii).Description – This will provide Error code message. Ex: "UserName should not be empty. Please enter username".
iii). Severity – This property provides error priority they are,
Critical, High, Medium, Low and Information.
iv). Timestamp - This will provide date time in ticks.

Note: If EnableTwoFactorAuthentication is set to false then client not require to call ValidateTwoFactorAuthentication method. The AuthenticateUser method itself will return sufficient information (Username, LogonDateTime to authenticate the client.

- **ValidateResponse:**
  If the Status Code of Response object is "1000" then the Request is "Success".
  If the Status Code of Response object is "1001" then the Request is "Fail".
  If the Status Code of Response object is "1003" then the Request is "Error".

Calling Service api:

Password should Pass with encryption as below.

```
ServiceReference1.UserAuthenticationServiceSoapClient clnt = new
ServiceReference1.UserAuthenticationServiceSoapClient();

ServiceReference1.UserAuthenticationRequest req = new
ServiceReference1.UserAuthenticationRequest();

string encPasssword = GetEncryptedPassword(txtUserName.Text,txtPassword.Text)
req.User = new UserSecurityTestApplication.ServiceReference1.User() { UserName =
txtUserName.Text, Password = encPasssword };

ServiceReference1.UserAuthenticationResponse resp = clnt.AuthenticateUserAcct(req);
```

Protecting Password With Encryption Library:

Here UserName and Password are Encrypting using CionEncryptionManager. For this add two dlls references (CionEncryptionManager and CionEncryptionHelper)

```
private string GetEncryptedPassword(string userName, string plainPassword)
{

     IPAddress[] ipList = Dns.GetHostAddresses(Dns.GetHostName());
     string ipAddress =string.Empty;

     foreach (IPAddress tmpAddress in ipList)
     {
        if (tmpAddress.AddressFamily == AddressFamily.InterNetwork)
        {
            ipAddress =tmpAddress.ToString();
            break;
        }
     }

      CionFramework.Utils.EncryptionManager encriptionManager = new
      CionFramework.Utils.EncryptionManager(ipAddress);
      string userMD5 = encriptionManager.MD5Encrypt(txtUserName.Text);
      CionFramework.Utils.EncryptionKeyGenerator encKeyGen = new
      CionFramework.Utils.EncryptionKeyGenerator();
```

```
            string encKeyForPwd = encKeyGen.GenerateKey(userMD5, IPAddress(), 24);
            CionFramework.Utils.EncryptionManager encManagerPwd = new
            CionFramework.Utils.EncryptionManager(encKeyForPwd);
             string encryptPasssword = encManagerPwd.TripleDESEncrypt(txtPassword.Text);
            return encryptPasssword;

        }
```

Validating Response :

Response Object will provide
UserName, FirstName, LastName, LogonTime, DistinguishedName (These are in UserAuthDetails property). SecurityQuestions, UserAuthenticationToken, ResponseStatus, EnableTwoFactorAuthentication and TwoFactorAuthType.
resp.ResponseStatus.StatusCode // 1000,1001 and 1003.
resp.ResponseStatus.Message //If error, it will show message (Success/Fail/Error).
resp.ResponseStatus.Exception.Code // If error, it will show Error code.
resp.ResponseStatus.Exception.Description // Error Code Description.

ValidateTwoFactorAuthenticationRequest

This will validate the Authentication Token which is issues for the same user in first request .If the Token is valid then it will validate the Two Factor Authentication (Security question Answers or SMSPinNumber or EmailPinNumber code).

Sample Code to use ValidateTwoFactorRequest Method

- **Input:**

  Here input parameter depends upon Two Factor Authentication Type. If Two Factor Authentication Type is SecurityQuestions then accepts **Answers** and **QuestionIds** as input parameters.
  If Two Factor Authentication Type is SMSPinNumber then accepts SMSPinNumber code as input parameter.
  If Two Factor Authentication Type is EmailPinNumber then accepts SMSPinNumber code as input parameter.

```
req.User = new ServiceReference1.User() { UserName =
resp.UserAuthDetails.UserName, SelectedTwoFactors = selectedTwofactor };
req.UserAuthenticationToken = resp.UserAuthenticationToken;
respAnswers = clnt.ValidateTwoFactorRequest(req);
```

  ValidateTwoFactorAuthenticationRequest method will accept UserAuthenticationRequest entity as Input Parameter. The UserAuthenticationRequest entity has following properties.

Answer – Its will take Security Answers for user Configured Security Questions. Here Answers should be sent with Encrypted using CionEncryptionManager dll.

UserAuthenticationToken – This is Unique token generated by server and sends to client. So that the Client should send to Server as part for request for further communication with Server.
UserName – It is for sending User name to validate Security Questions or SMSPinNumber to that user.
QuestionId - This will accept question Ids for those User Configured Security Questions while registration.
QuestionName – This property for show Question names.
SMSPinNumber –The Generated OTP Pin code which was sent to user mobile number has to be passed as input parameter to validate SMSPinNumber. This Generated Code will be valid before Twenty Four hours.
EmailPinNumber –The Generated OTP Pin code which was sent to user email address has to be passed as input parameter to validate EmailPinNumber. This Generated Code will be valid before Twenty Four hours.

- **Output:**

ValidateTwoFactorAuthenticationRequest method will return `UserAuthenticationResponse` entity as response. The `UserAuthenticationResponse` has following properties.

UserLogonDetails – This property contains UserName, FirstName, LastName, LogonTime

ResponseStatus –This property contains few Sub Properties. They are,

a). **StatusCode** - This will provide Statuscode as 1000, 1001 and 1003.
b). **Message** – This will provide Messagefor Success/Fail/Error.
c). **Exception** - This property contains few sub properties they are,

i). Code - This property will provide error code. Ex: 6000, 6001... So on.
ii).Description – This will provide Error code message. Ex: "UserName should not beempty. Please enterusername".
iii). Severity– This property provides error priority, possible priorities are,
Critical, High, Medium, Low and Information.
iv). Timestamp - This will provide date time in ticks.

- **ValidateResponse:**

If the Status Code of Response object is "1000" then the Request is "Success".
If the Status Code of Response object is "1001" then the Request is "Fail".
If the Status Code of Response object is "1003" then the Request is "Error".

Sample Code for Calling Service api:

Answers should with encryption.

Input parameters: UserName, QuestionIds, Answers(with Encrypted) and User UserAuthenticationToken.

```
ServiceReference1.UserAuthenticationRequest req = new
ServiceReference1.UserAuthenticationRequest();
ServiceReference1.SecurityQuestion[] securityQues = resp.SecurityQuestions;

        for (int j = 0; j < securityQues.Length; j++)
        {
            if (securityQues [j].QuestionId == 1)
                securityQues [j].Answer = "2WKUml+xQUk=;
            if (securityQues [j].QuestionId == 2)
                securityQues [j].Answer = "ZBwRidAP2Gg=";
            if (securityQues [j].QuestionId == 3)
                securityQues [j].Answer = "2WKUml+xQUk=";
        }

req.User = new UserSecurityTestApplication.ServiceReference1.User() { UserName =
userName };

req.SecurityQuestions = securityQues;
req.UserAuthenticationToken = resp.UserAuthenticationToken;

ServiceReference1.UserAuthenticationResponse resp =     clnt.
ValidateTwoFactorAuthenticationRequest (req);
```

Validating Response:

```
resp.UserName
resp.LogonDateTime
resp.ResponseStatus.StatusCode // 1000,1001 and 1003.
resp.ResponseStatus.Message //If error, it will show message (Success/Fail/Error).
resp.ResponseStatus.Exception.Code // If error, it will show Error code.
resp.ResponseStatus.Exception.Description // Error Code Description.
```

Error Codes

Status codes and Messages for all methods.

6000 - Username should not be empty. Please provide valid username
6001 – Client is not registered as a trusted party in Cloud Identity Minder.
6002 - User account is blocked. Please contact administrator.
6003 – User account is not registered in Cloud Identity Minder application. Please Register.
6004 - Please provide valid answers.
6005 - Answers are required for Authentication. Please enter.
6006 - User Login failed. Please provide valid credentials.
6007 - Please enter valid One Time Password.
6008 - User account is locked. Please contact administrator.
6009 - User Authentication Token is Invalid.
6010 - Please provide Two Factor Authentication Values.
6011 - User account is disabled. Please contact administrator.
6012 - Password should not be empty. Please provide valid password.
6013 - User must change password at next logon. Please login to Cloud Identity Minder.
6014 - Unable to perform operation at this time. Please retry after few minutes or Contact Administrator.
6015 - Invalid Distinguishedname.
6016 - Distinguishedname should not be empty. Please provide Distinguishedname.
6017 - Password is Expired please reset your password.
6018 - User Account is locked or disabled. Please contact administrator
6019 - User Authenticated Successfully -- First Factor.
6020 - User Authenticated Successfully -- Second Factor.
6021 - User Authentication Validation Success.
6022 - Challenge Questions Retrieved Successfully.
6023 - SMSPinNumber Generated Successfully.
6024 - SMSPinNumber Validation Success.
6025 - Exception Occured in SMSPinNumber Validation.
6026 - User Registration Validation Success.
6027 - Challenge Answers Validated Successfully.
6028 - The user name or password is incorrect. Verify your user name, and then type your password again.
6029 - Your Account is Blocked! Please contact your admin to unblock it!
6030 - User is not a licensed office365 account. Use a office365 licensed account.
6031 - Server is busy right now, please try after some time.

Note: Any Service connection failure cases should be handled by the Client Application.

Contact Notes:

For technical support or feature requests, please contact us at Support@CionSystems.com or 425.605.5325.

For sales or other business inquiries, we can be reached at Sales@CionSystems.com or 425.605.5325

If you'd like to view a complete list of our Active Directory Management solutions, please visit us online at www.CionSystems.com

Disclaimer

The information in this document is provided in connection with CionSystems products. No license, express or implied, to any intellectual property right is granted by this document or in connection with the sale of CionSystems products. EXCEPT AS SET FORTH IN CIONSYSTEMS'LICENSE AGREEMENT FOR THIS PRODUCT, CIONSYSTEMS INC. ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL CIONSYSTEMS INC. BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL,PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF CIONSYSTEMS INC. HAS BEEN ADVISED IN WRITING OF THE POSSIBILITY OF SUCH DAMAGES. CionSystems may update this document or the software application without notice.

CionSystems Inc
6640 185th Ave NE,
Redmond, WA-98052, USA
www.CionSystems.com
Ph: +1.425.605.5325