Admin Member Management

Last updated by | Eric Ulrich | May 23, 2024 at 12:07 PM MDT

NOTE: The Base URL for all endpoints in this file are ServiceHost endpoints unless otherwise specified. Make sure to double check the Base URL when calling these endpoints.

Contents

How to determine membership user id

The admin teller's login record for the old admin tool is stored in the table udt_AdminTable. Get the Teller field from that table for the teller logged in to the low code admin tool. The membership user id for that teller is the id column of the MembershipUsers table for the entry where the LEGACY_udt_AdminTable_ID is set to the Teller field from the udt_AdminTable.

Example using SQL

```
select MembershipUserId from MembershipUsers
where LEGACY_udt_AdminTable_ID = (select Teller from udt_AdminTable where EmailAddress = '{replaceMe}')
```

Member Search

Search for a member by an identifier. The identifier type is optional. Teller id for this and other calls refers to the MembershipUserId for the teller performing the search.

Route

```
POST ~/api/admin-member-management/v1/search
```

Request

```
Body
{
    "Identifier": long,  REQUIRED
    "IdentifierType": AdminMemberSearchIdentifierType,
    "TellerId": long  REQUIRED
}

AdminMemberSearchIdentifierType Possible values
    AccountNumber
    PersonNumber
    OrgNumber
    MemberNumber
    CoreUserId
    Username
    TaxId
    Uuid
```

Response
200 - OK - Success

```
Body
{
    "members": [ (this is a list)
        {
            "accountNumber": string,
            "personNumber": string,
            "username": string,
            "uuid": string,
            "accountCategory": string
        }
    ],
    "demographics": { (only returned when there is exactly 1 member in the response)
        "firstName": string,
        "middleInitial": string,
        "lastName": string,
        "address": {
            "category": string,
            "addressLine1": string,
            "city": string,
            "state": {
                "name": string,
                "abbreviation": string
            },
            "zipCode": {
                "five": string,
                "plusFour": string
            },
            "isExpired": bool,
            "isDefault": bool,
            "residenceYears": int
        },
        "phoneNumber": {
            "type": string,
            "isDefault": bool,
            "areaCode": string,
            "exchangeCode": string,
            "subscriberNumber": string,
            "number": string
        },
        "dateOfBirth": datetime,
        "emailAddress": {
            "type": string,
            "isDefault": bool,
            "address": string
        },
        "preferredContactMethod": string,
        "onlineBankingEnrollmentDate": datetime,
        "taxId": string
    }
}
```

401 - Unauthorized - The necessary configs for Member Management API are not enabled
500 - Internal Server Error - Something went wrong

Get Member Demographics

Use this endpoint when multiple members were returned from the Member Search endpoint and the Teller has selected one to view. Pass in the member object exactly as it was returned from that endpoint. Teller id for this and other calls refers to the MembershipUserId for the teller performing the search.

Route

```
POST ~/api/admin-member-management/v1/demographics
```

Request

```
Body
{
    "TellerId": long, REQUIRED
    "Member": { REQUIRED
        "accountNumber": string,
        "personNumber": string,
        "username": string,
        "uuid": string,
        "accountCategory": string
    }
}
```

Response
200 - OK - Success

```
Body (this is the same object as the Member Search endpoint)
{
    "members": [
        {
            "accountNumber": string,
            "personNumber": string,
            "username": string,
            "uuid": string,
            "accountCategory": string
        }
    ],
    "demographics": {
        "firstName": string,
        "middleInitial": string,
        "lastName": string,
        "address": {
            "category": string,
            "addressLine1": string,
            "city": string,
            "state": {
                "name": string,
                "abbreviation": string
            },
            "zipCode": {
                "five": string,
                "plusFour": string
            },
            "isExpired": bool,
            "isDefault": bool,
            "residenceYears": int
        },
        "phoneNumber": {
            "type": string,
            "isDefault": bool,
            "areaCode": string,
            "exchangeCode": string,
            "subscriberNumber": string,
            "number": string
        },
        "dateOfBirth": datetime,
        "emailAddress": {
            "type": string,
            "isDefault": bool,
            "address": string
        },
        "preferredContactMethod": string,
        "onlineBankingEnrollmentDate": datetime,
        "taxId": string
    }
}
```

401 - Unauthorized - The necessary configs for Member Management API are not enabled
500 - Internal Server Error - Something went wrong

Reset Password

Resets a password for the specified member to a provided password or a generated one if not provided. There is also the option to send the password to the member's device (SMS, phone, or Email) if the device Id is provided.

Route

```
POST ~/api/admin-member-management/v1/reset-password
```

Request

```
Body
{
    "Uuid" : long,   REQUIRED
    "AccountNumber" : int,   REQUIRED
    "Password" : "string",
    "DeliveryAddress" : "string", (Phone Format: "1231231234 or Email Format "test@email.com")
    "SendAsVoice" : bool,
    "ClientIp" : "string",
    "WebId": "string",
    "TellerId" : long   REQUIRED
}
```

Response
200 - OK - Success

```
Body
{
    "passwordResetSuccessful": bool,
    "messageDeliverySuccessful": bool
}
```

401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Both the account and UUID were not provided
500 - Internal Server Error - Something went wrong

Send Username/Password

This endpoint will send the username to the provided phone number. If the type password is given then it will change the password without resetting the login counts

Route

```
POST ~/sms-security-code/send/v1
```

Request

```
Body
{
    "Uuid": long,
    "PhoneNumber": string, (No dashes or parentheses)
    "RequestType": int, (10-AdminSendTempPasswordToMember, 11-AdminSendUsernameToMember)
    "ShouldSendVoiceMessage": bool
}
```

Response
200 - OK - Success

```
Body
{
    "Successful" : bool,
    "RequestType" : int,
    "RequestId" : int
}
```

500 - Internal Server Error - Something went wrong

Get Phone Numbers

This endpoint will get the phone numbers for a specified member

Route

```
POST ~/sms-security-code/phone-numbers/v1
```

Request

```
Body
At least one of the following needs to be provided (Uuid, RequestId, AccountIdentifier, Username)
{
    "Uuid": long,
    "RequestId": GUID,
    "AccountIdentifier": string,
    "Username": string,
    "UserMetaData":
        {
        "UserAgentString": string,
        "UserIpAddress": string,
        "DeviceIdentifier": string,
        "SessionId": string,
        "WebIpAddress": string,
        "ClusterId": string,
        "UserApplication": "HomeBankingAdmin"
        }
}
```

Response
200 - OK - Success

```
Body
{
    "phoneNumbers": [
        {
            "type": "string",
            "isDefault": bool,
            "areaCode": "string",
            "exchangeCode": "string",
            "subscriberNumber": "string",
            "number": "string"
        },
        ...
    ]
}
```

500 - Internal Server Error - Something went wrong

Send Security Code

This endpoint will send a security code to the provided address

Route

```
POST ~/sms-security-code/one-time/v1
```

Request

```
Body
{
    "PhoneNumber": string, (No dashes or parentheses)
    "ShouldSendVoiceMessage": bool
}
```

Response
200 - OK - Success

```
Body
{
    "SecurityCode": "string",
    "Success" : bool
}
```

500 - Internal Server Error - Something went wrong

Unlock Member Account

Unlocks a member's account when the account has been locked.

Route

```
POST ~/api/admin-member-management/v1/unlock-member-account
```

Request

```
Body
{
    "Uuid" : long,   REQUIRED
    "AccountNumber" : int,   REQUIRED
    "TellerId" : long   REQUIRED
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Both the account and UUID were not provided
500 - Internal Server Error - Something went wrong

Force Member Info Validation

This call will force a member to validate either their email or address, or both, on the next login that happens.

Route

```
POST ~/api/admin-member-management/v1/force-info-validation
```

Request

```
Body
{
    "TellerId" : long,   REQUIRED
    "Member" :  REQUIRED (Member object from Member search)
    {
         "accountNumber": string,
         "personNumber": string,
         "username": string,
         "uuid": string,
         "accountCategory": string
    },
    "ForceEmailValidation" : bool,
    "ForceAddressValidation" : bool
}
```

Response
200 - OK - Success

```
Body
{
    "emailValidationSet": bool,
    "addressValidationSet": bool
}
```

401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request
500 - Internal Server Error - Something went wrong

Set Member Can Skip Captcha (login security code)

This call will set whether a member can skip the captcha check on login.

Route

```
POST ~/api/admin-member-management/v1/can-skip-captcha
```

Request

```
Body
{
    "TellerId" : long,   REQUIRED
    "Member" :  REQUIRED (Member object from Member search)
    {
         "accountNumber": string,
         "personNumber": string,
         "username": string,
         "uuid": string,
         "accountCategory": string
    },
    "CanSkipCaptcha" : bool
}
```

Response
200 - OK - Success (no body)
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Change Member MFA Challenge Settings

This call will set whether a member can skip the captcha check on login.

Route

```
POST ~/api/admin-member-management/v1/mfa-user-challenge-settings
```

Request

```
Body
{
    "TellerId" : long,   REQUIRED
    "Member" :  REQUIRED (Member object from Member search)
    {
         "accountNumber": string,
         "personNumber": string,
         "username": string,
         "uuid": string,
         "accountCategory": string
    },
    "NumberOfLogins" : int (nullable),
    "NumberOfDays" : int (nullable),
    "ForceChallengeSelection" : MfaForceChallengeSelectionEnum
}

MfaForceChallengeSelectionEnum Options
    Default
    AlwaysChallenge
    SkipChallenge
```

Response
200 - OK - Success (no body)
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Unlock Mobile Registration

This call will clear the block preventing mobile registration.

Route

```
POST ~/api/admin-member-management/v1/unlock-mobile-registration
```

Request

```
Body
{
    "TellerId" : long,   REQUIRED
    "uuid": string,   REQUIRED
}
```

Response
200 - OK - Success (no body)
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Get Cardlytics Information

This call will get a member's cardlytics status and information

Route

```
POST ~/api/admin-member-management/v1/cardlytics-information
```

Request

```
Body
{
    "TellerId" : long,   REQUIRED
    "Member" :  REQUIRED (Member object from Member search)
    {
         "accountNumber": string,
         "personNumber": string,
         "username": string,
         "uuid": string,
         "accountCategory": string
    }
}
```

Response
200 - OK - Success

```
Body
{
    "isActive": true,
    "postingSuffixes": [
        "string",
        "string"
    ],
    "currentPostingSuffix": "string",
    "cardlyticsCustomerId": "string"
}
```

401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request
500 - Internal Server Error - Something went wrong

Update Cardlytics

This call will update a member's status and the posting suffix for Cardlytics.

Route

```
POST ~/api/admin-member-management/v1/force-info-validation
```

Request

```
Body
{
    "TellerId" : long,   REQUIRED
    "Member" :  REQUIRED (Member object from Member search)
    {
         "accountNumber": string,
         "personNumber": string,
         "username": string,
         "uuid": string,
         "accountCategory": string
    },
    "IsActive" : bool,
    "PostingSuffix" : string
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request
500 - Internal Server Error - Something went wrong

Set Restrict Secure Message Attachments

This call will set whether a member is allowed to upload attachments to secure messages

Route

```
POST ~/api/admin-member-management/v1/restrict-secure-message-attachments
```

Request

```
Body
{
    "TellerId" : long,   REQUIRED
    "Member" :  REQUIRED (Member object from Member search)
    {
         "accountNumber": string,
         "personNumber": string,
         "username": string,
         "uuid": string,
         "accountCategory": string
    },
    "RestrictSendingAttachments" : bool
}
```

Response
200 - OK - Success (no body)
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Get Connect Native Login Redirect Choice

Get the current choice for a member on whether they should be redirected to connect native on login

Route

```
GET ~/api/User-options/connect-native-login-redirect-choice/v1/{uuid}
```

Url Parameter

```
uuid - long, the uuid of the member
```

Response
200 - OK - Success

```
{
  "ConnectNativeLoginRedirectChoice": enum
}

possible enum values:
None (use whatever the default is)
OptedOut (do not redirect)
OptedIn (redirect)
```

401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Set Connect Native Login Redirect Choice

Set the current choice for a member on whether they should be redirected to connect native on login

Route

```
POST ~/api/User-options/connect-native-login-redirect-choice/v1
```

Request

```
Body
{
    "Uuid" : long,   REQUIRED
    "ConnectNativeLoginRedirectChoice" : enum
}

possible enum values:
None (use whatever the default is)
OptedOut (do not redirect)
OptedIn (redirect)
```

Response
200 - OK - Success

```
{
  "ConnectNativeLoginRedirectChoice": enum
}
```

401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Unregister User Device

Will unregister a User Device using the uuid, and device guid

Route
POST {psiServiceHost}/api/admin-member-management/v1/unregister-user-device

Request

```
Body
{
    "deviceGuid": "string/guid",
    "uuid": long,
    "tellerId": int
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Lock User Device

Will lock a User Device using the uuid, and device guid

Route
POST {psiServiceHost}/api/admin-member-management/v1/lock-device

Request

```
Body
{
    "deviceGuid": "string/guid",
    "uuid": long,
    "tellerId": int
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Unlock User Device

Will unlock a User Device using the uuid, and device guid

Route
POST {psiServiceHost}/api/admin-member-management/v1/unlock-device

Request

```
Body
{
    "deviceGuid": "string/guid",
    "uuid": long,
    "tellerId": int
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Update Device Nickname

Will update a device nickname using the uuid, and device guid

Route
POST {psiServiceHost}/api/admin-member-management/v1/update-device-nickname

Request

```
Body
{
    "nickname": "string",
    "deviceGuid": "string/guid",
    "uuid": long,
    "tellerId": int
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Update Delivery Address

Will update a device delivery address using the uuid, and device guid

Route
POST {psiServiceHost}/api/admin-member-management/v1/update-delivery-address

Request

```
Body
{
    "address": "string",
    "deviceGuid": "string/guid",
    "uuid": long,
    "tellerId": int
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member
500 - Internal Server Error - Something went wrong

Delete User Device

Will change the status of a user device to Removed/Deleted using the uuid, and device guid

Route
POST {psiServiceHost}/api/admin-member-management/v1/delete-user-device

Request

```
Body
{
    "deviceGuid": "string/guid",
    "uuid": long,
    "tellerId": int
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled
400 - Bad Request - Necessary fields were missing from the request or there was no Device record for the given GUID
500 - Internal Server Error - Something went wrong

Is Employee

Checks if a provided Uuid or Account is for an employees account

Route
POST {psiServiceHost}/api/admin-member-management/v1/is-employee

Request

```
Body
{
    "Account": "string",
    "Uuid": long
}
```

Response
200 - OK - Success
401 - Unauthorized - The necessary configs for Member Management API are not enabled

400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member

500 - Internal Server Error - Something went wrong

Checks if a provided Uuid or Account is for an employees account

FedNow Get Preference

Route
Get {psiServiceHost}/api/admin-member-management/v1/fed-now-preference/{uuid}

Response

```
Body
{
    "wasSuccessful": bool,
    "optValue": "Enum" --will return the string representation of the value
}
```

200 - OK - Success

401 - Unauthorized - The necessary configs for Member Management API are not enabled, FedNow is not enabled or the config Fednow.GlobalAvailability setting is set to Off

400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member

500 - Internal Server Error - Something went wrong

FdNow Set Preference

Checks if a provided Uuid or Account is for an employees account

Route
POST {psiServiceHost}/api/admin-member-management/v1/is-employee

Request

```
Body
{
    "Member":{  REQUIRED  --This is the same object returned form the Member Search call
            "accountNumber": "string",
            "username": "string",
            "uuid": "string",
            "accountCategory": "string"
        },
    "OptValue": Enum --Can be either an int or string, possible values are:
                   0 or "Default"
                   1 or "OptIn"
                   2 or "OptOut"
}
```

Response

```
Body
{
    "wasSuccessful": bool,
    "optValue": "Enum" --will return the string representation of the value
}
```

200 - OK - Success

401 - Unauthorized - The necessary configs for Member Management API are not enabled, FedNow is not enabled or the config Fednow.GlobalAvailability setting is set to Off

400 - Bad Request - Necessary fields were missing from the request or there was no login record for the given member

500 - Internal Server Error - Something went wrong