Profile Management

New Stack Business Services

Interface Design Document

Rick Foster

May 27, 2015

Version: 3.91

Contents

[Introduction 3](#_Toc390764905)

[Profile Management "New Stack" 3](#_Toc390764906)

[Architecture 3](#_Toc390764907)

[Environment 4](#_Toc390764908)

[Error handling 4](#_Toc390764909)

[Versioning 4](#_Toc390764910)

[Interface Specification Approach 5](#_Toc390764911)

[JSON Response Format 5](#_Toc390764912)

[Profile Management REST Services 6](#_Toc390764913)

[registration-preconditions 6](#_Toc390764914)

[service-association 7](#_Toc390764915)

[credential-validation 9](#_Toc390764916)

[credential-validation 10](#_Toc390764917)

[token-validation 11](#_Toc390764918)

[account-validation 13](#_Toc390764919)

[service-disassociation 14](#_Toc390764920)

[email 16](#_Toc390764921)

[Appendix 18](#_Toc390764922)

[Error Codes 18](#_Toc390764923)

# Introduction

## Profile Management "New Stack"

Profile management is largely the domain of several platforms:

1. Open SSO aka “Client Identity”- Open SSO based single sign on platform for managing portal user profiles (UUID and associated roles and personalization parameters)
2. Your Account Portal – Weblogic portal based webapp that contains portlets allowing users to associate business objects (billing accounts, subscriptions) with their portal user profiles.
3. Portal EJB Client – backend services which Your Account Portal uses to read and modify data in CODS (ARDB for user profile associations, CRDB for read only billing data copied regularly from Knowbility)
4. OneStop - Tool for client care agents providing comprehensive suite of user profile management functions in order for agents to manage profiles in behalf of clients.

In order to align with new priorities including the ability to make and deploy changes quickly, the existing applications will be refactored into a set of business services to be consumed by the next generation TELUS consumer portal (with other lines of business as potential consumers in the future). We refer to this suite of services as "Profile Management New Stack" (PMNS).

This document is an interface specification of these services.

## Architecture

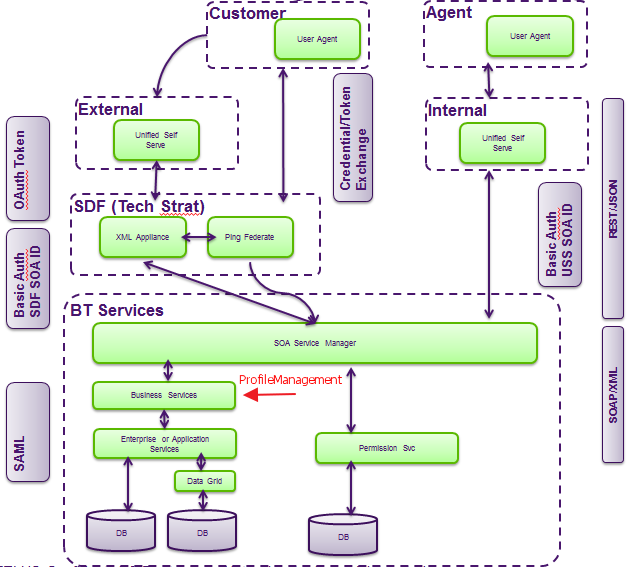
PMNS services are business-level services. They provide functionality suitable web tier applications to consume directly. These services aggregate information from and orchestrate functionality from lower level domain web services. Simply wrapping an existing domain service doesn't add any value and we would generally refrain from doing so.

The approach we have taken with respect to granularity and interface is to use the existing SelfServe flows and the Struts actions which tie UI pages together as a starting point, with a view to making them general enough to be usable by other potential consumers.

Keeping in mind the generally stateless nature of REST, we provide for optional parameters which would be the type of information a UI would cache, and which when provided to the service allow for bypassing certain lower level calls thereby increasing performance.

The currently identified business requirement is for REST services returning a JSON payload. Technical requirements include governance via the SOA Policy Manager. Version 6.1 of SOA, which is to be promoted to production in Q3 2013, has support for REST services. The SOA Competency Centre team has recommended that our services be built as SOAP services (due to the use of last-mile SAML security which a pure REST service would not support). A REST binding and XML-to-JSON transformation, features of 6.1, will allow SOAP services to behave like REST/JSON. The one limitation to this approach discovered thus far is that a SOAP service masquerading as REST in this way cannot control the HTTP status code, which is the canonical way to communicate response status in REST. The next section describes how we deal with this.

The "RESTification" feature of SOA 6.1 allows lower level domain SOAP services to be easily exposed without the need for a REST wrapper service, should that be required.



## Environment

The Profile Management services is driven by the following webapp in PROD:

BuildServer=btln001124.corp.ads  
WAR=/work/staging/SSNS/ProfileManagement/SSNS.PROFILE.MANAGEMENT.1.0.90/war/profileManagement-SOAP.war  
Domain=Currently active PRCustomerManagementBusSvc  
Targets=AccountManagement

## Error handling

REST design normally maps error conditions to existing well-established HTTP error codes, e.g. 200 ("OK"), 404 ("Resource not found"). Both for reasons of design flexibility (the predefined codes often seem ill-suited to specific conditions) and technical limitations (the SOA Policy Manager's implementation of exposing SOAP services as REST as described above), we are taking a different approach. Generally, as long as the call to the SOA-bound service succeeds, the HTTP response status will be 200 (success), even when there are underlying failures, and a "status" object, always part of the response payload, is used to indicate any such failures. (This is the approach Facebook takes.)

The **status** object is defined as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Datatype** | **Description** | **Possible/typical values** |
| statusCd | string | HTTP result code | 200, 400, 500 |
| statusSubCd | String | Our internal status code | UM1, UM2 |
| statusTxt | string | Short description of status | Invalid URL; CustomerRatingInquiry call failed |
| systemErrorTimeStamp | string | Time of underlying error (to help in locating error in logs) | 2013-05-23T11:07:48 |
| systemErrorCd | string | If the underlying service we call returns its own error code, we'll include it here, e.g. Contents of faultString in error response | "error: java.lang.IllegalArgumentException";  " BAN [90653242] Not Found" |
| systemErrorTxt | string | Description of system error; messageId, errorCode from PolicyException | "messageId: APP10004; errorCode CMB\_AIS\_0001" |

## Versioning

The strategy for this is still being finalized as of this writing. However, broadly speaking the major version number will become part of the service name.

## Interface Specification Approach

As the primary (or at least initial) consumer of our services requires REST/JSON, while our underlying implementation is SOAP/XML, this document describes the interface with REST-style requests and JSON-format responses. Our SOAP implementation will be done in a way to make the SOA transformation to JSON achieve the desired result.

## JSON Response Format

The approach here is to always provide all fields in the spec, even if they have no assigned value. While omitting such fields can make for a smaller payload, keeping the response format consistent tends to make client-side coding to consume such responses more straightforward. Fields of string type will have a value of "null" rather than empty string "", e.g.

"middleInitial" : null,

rather than

"middleInitial" : "",

# Profile Management REST Services

This service provides operations in support of user profile and business services account association. Supports linking and registration front end flows for registration.

## registration-preconditions

|  |  |
| --- | --- |
| **OPERATION**  sub/{sub}/registration-preconditions | |
| **Method** | GET |
| **Description** | Used to establish capabilities of the subscriber device preconditions necessary for on-network auto-detection registration. |
| **Input** | Inputs:  **sub** = subscription MDN, ie “4169762323”  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | brand | 1 = Telus  3 = Koodo | Yes | Indicates company brand |   Examples:  profilemanagement/sub/4169762323/registration-preconditions?brand=1 |
| **Status** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IS | Invalid Subscriber (Portal DAL call failed) | Subscriber [4169762323] Not Found | \*\* Rob to provide via DAL | DAL fails due to invalid subscriber | | 500 | DAL | Application level error (DAL call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  | Pass through error code | Any caught exception not captured elsewhere | |
| **Output** | “registrationPreconditionsResponse” :  {  "singleSubscriberInd": true,  “smsCapableInd” : true,  “doesOwnerExistInd” : true,  “doesManagerExistInd” : true,  “doesMemberApproverExistInd” : true,  "accountTypeCd" : "I",  "status" :  { "statusCd" : "200",  "statusSubCd" : null,  "statusTxt" : "OK",  "systemErrorTimeStamp": null,  "systemErrorCd" : 0,  "systemErrorTxt" : null  }  }  Where fields are returned:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | singleSubscriberInd | Boolean | Condition for is single subscriber on BAN | **true**, or **false** | | smsCapableInd | Boolean | Determines if device is capable of receiving SMS  (ie, device type is not internet key) | **true**, or **false** | | doesOwnerExistInd | Boolean | Determines if an owner already exists on BAN associated to subscriber. | **true**, or **false** | | doesManagerExistInd | Boolean | Determines if an Manager exists already exists on BAN associated to subscriber. | **true**, or **false** | | doesMemberApproverExistInd | Boolean | Determines if a user on BAN exists that is allowed to approve member access. True if for TCS BANs Owner or Manager exists, or if Business/Corp BAN has a Manager. Otherwise false. | **true**, or **false** | | accountTypeCd | string | Mobility account type as defined in CODS given that only *mobility* phone numbers are input into registration. | "I" for Consumer, "B" for SMB, "C" for Corporate | |
| **SLA** | Services/APIs called:   |  |  | | --- | --- | | **Serviced/operation** | **SLA** | | **AccountMgmtDataService v 1.0**  getAccountbySubscriber(subscriber,brand) | 1000 ms average; 3000 ms peak | | **WirelessPermissionSvc3.0**  getRolePermissionListfromObject | 1000 ms average; 3000 ms peak |   Expected average response time: 1500 – 5000 ms (depending on which optional parameters are passed in).  Note: SLAs for services above are based estimations. |
| **Sample GUI** |  |

## service-association

|  |  |
| --- | --- |
| **OPERATION**  profile/{uuid}/service-association | |
| **Method** | POST |
| **Description** | TO associate either a BAN (as owner), BAN (as manager) or Subscriber (as member) to a given portal profile (UUID) |
| **Input** | Inputs:  **uuid** = Indicates the user profile ID to associate service to  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | ban | “123456789” | Yes if sub is null | Indicates the account to associate to user profile as either owner or manager | | sub | “4169762323” | Yes if BAN is null | Indicates the subscriber to assign as member to user profile | | pin | “1234” | Yes if sub is null and ban is type “B” or “C” | Personal Identification Number for validation purposes (TBS only) | | brand | 1 = Telus  3 = Koodo | Yes | Indicates company brand | | uuid | ie “21EC2020-3AEA-1069-A2DD-08002B30309D” | Y | Client Identity unified user profile identifier | | postalcd | “L3S3S8” | Yes if sub is null | Postal code for validation purposes. Value will be trimmed of spaces and symbols, made compared ignoring case |   Example:  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/service-association |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **Status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN | BAN [90653242] Not Found | messageId=xxxxx errorCode=xxxxx | Call fails due to invalid Ban | | 400 | IS | Invalid Sub | Subscriber [416-97623232] Not Found | messageId=xxxxxxxx, errorCode=xxxxxxxxxx | Call fails due to invalid Sub | | 400 | IC | Invalid Credentials |  |  | Call fails due to invalid BAN validation criteria being passed in. | | 400 | AE | Already Exists |  | User: e3f30e70-420d-45b5-8bc0-2575aeebc182 has an existing active link to business object + urn:x-telus-business:account:knowbility:70076120 | Account to profile association already exists. | | 400 | OAE | Owner Already Exists |  | User: An owner already exists for business object + urn:x-telus-business:account:knowbility:70076120 | Account has another Owner already | | 400 | AWLNO | Already Wireline Owner |  |  | Profile already has Owner permission on passed Customer ID (Wireline only) | | 400 | AMAEWLN | Another Manager Already Exists for this wireline BAN |  |  | To disallow others taking over an active link that already exists against active UUID | | 400 | PR | Pin is required |  |  | Call fails because PIN is required for SMB and Corp BANs | | 400 | WLNPR | Pin is required |  |  | Call fails because PIN is required for Wireline FFH Customers | | 500 | WPS | WirelessPermissionSvc Down | underlying error code | underlying error message | Any other Policy or Service Exception | | 500 | DAL | Application level error (DAL call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |
| **Output** | “associateServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {  "clientTypeTxt" : "Mobility Consumer",  “clientRoleTypeTxt” : “Owner”,  “accountTypeCd” : “I”,  “accountSubTypeCd” : “R”,  “customerTypeCd” : “R”,  “customerSubTypeCd” : “I”,  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |   Where fields are returned:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | clientTypeTxt | string | Type of association made | **Mobility Consumer**  **Mobility SMB**  **Mobility SMB Business Connect**  **Mobility Corporate**  **FFH Consumer**  **FFH SMB** | | clientRoleTypeTxt | string | Type of role obtained | **Owner (Consumer only)**  **Manager**  **Member** | | accountTypeCd | String | BAN type from source billing system ODS | **“I” for Consumer “B” for Business**  **“C” for Corporate** | | accountSubTypeCd | String | BAN type from source billing system ODS | **“R” for Regular** | | customerTypeCd | String | Customer type for FFH (hard coded to “R” if BAN provided was FFH) | **“R” for Residential** | | customerSubTypeCd | String | Customer sub type for FFH (hard coded to “I” if BAN provided was FFH) | **“I”** | |
| **SLA** | Services/APIs called:  **AccountMgmtDataService**  getClientAccountBillingContactIndividual  (returns IndividualValueObject containing Address object containing a postal code)  **SubscriberInformationService v3.2 (BVoip MDN cross reference)** getSubscriberByPhoneNumber  **ClientIdentityProfileSvc**  getProfiles (to see what role UUID has already)  updateAuthorization (to set wireless role if not already set)  **WirelessPermissionSvc**  getRolePermissionList  **UserRoleManagementSvc**  approveUserRole  ClientIdentityProfileSvc    **PermissionService** (aka enterprise permission svc)  getPermissions  **ConsumerCustomerManagementService**  *searchCustomer*  *getCustomer*  *getCustomerWithContacts*  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## credential-validation

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/credential-validation | |
| **Method** | GET |
| **Description** | TO validate requester as a client that owns the a BAN (as owner) |
| **Input** | Inputs:  **ban** = Billing Account Number as defined in billing system, ie “123456789”  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | brand | 1 = Telus  3 = Koodo | Yes | Indicates company brand | | postalcd | “L3S3S8” | Yes | Postal code for validation purposes. Value will be trimmed of spaces and symbols, made compared ignoring case | | pin | “1234” | Yes ban is type “B” or “C” | Personal Identification Number for validation purposes (TBS only) |   Example:  profilemanagement/account/123456789/credential-validation?brand=1&postalcd=l3s3s8 |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN | BAN [90653242] Not Found | messageId=xxxxx errorCode=xxxxx | Call fails due to invalid Ban | | 400 | IC | Invalid Credentials |  |  | Call fails due to invalid BAN validation criteria being passed in. | | 400 | ANE | Approver Does Not Exist  (v1.1 and up only) |  |  |  | | 500 | DAL | Application level error (DAL call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |
| **Output** | “credentialValidationServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {  "clientTypeTxt" : "Mobility Consumer",  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |   Where fields are returned:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | clientTypeTxt | string | Type of association made | **Mobility Consumer**  **Mobility SMB**  **Mobility SMB Business Connect**  **Mobility Corporate**  **FFH Consumer**  **FFH SMB** | |
| **SLA** | Services/APIs called:  **AccountMgmtDataService**  getClientAccountBillingContactIndividual  (returns IndividualValueObject containing Address object containing a postal code)  **ConsumerCustomerManagementService**  *searchCustomer*  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## credential-validation

|  |  |
| --- | --- |
| **OPERATION**  credential-validation | |
| **Method** | GET |
| **Description** | To validate a Consumer’s User profile First Name, Last name when associated with a BAN or Subscriber number provided. Steps executed:  If BAN provided:   * Query CODS (via account mgmt DAL) for BAN or SUB provided, if not found query CustomerODS (FFH) via ConsumerCustomerManagementService (getCustomer) with BAN (only) provided. * Query WPS for UUID associated with the BAN or SUB, if not found query EPS using BAN   + URNs are returned, either as owner, manager, or subscriber roles.   + If SUB is provided as input and no member role found, query for mobility BAN associated to SUB and re-query WPS for association. * For each URN returned by WPS/EPS, get UUID and query CIPS (getProfiles) and compare first/lastname * If single matching UUID is found, call CIPS (getIdentityToken) to set and retrieve identityToken and return in response. Otherwise:   + Throw error if no BAN found   + Throw error if no SUB found   + Throw error if BAN or SUB found but no UUID matching in CIPS   Return either a success (200) or an error (400 if not matching or 500 if a system error) |
| **Input** | Inputs:  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | ban | 9 digit string ‘987654321’ | Yes if sub is null |  | | sub | Phone number ‘4166843443’ | Yes if ban is null |  | | firstname | String “Rick” case insensitive | yes | firstName of **user profile** associated to the BAN or sub. | | lastname | String “Foster” case insensitive | yes | firstName of **user profile** associated to the BAN or sub. |   Example:  ban only:  profilemanagement/credential-validation?ban=987654321&firstname=Rick&lastname=Foster  sub only:  profilemanagement/credential-validation?sub=4166843443&firstname=Rick&lastname=Foster |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NMU | No Matching User |  | Ban or Sub found, but no user profile identified with matching first/last name |  | | 400 | NMA | No Match Account |  | Name provided does not match ban or sub |  | | 500 | DAL | Application level error (DAL call fails) |  | Pass through error code | Any other Service Exception | | 500 | CCMS | Application level error (Consumer customer Management Servoice fall fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |
| **Output** | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | “credentialValidationServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {  "clientTypeTxt" : "Mobility Consumer",  "identityTokenId" : "123kjh123kh123lk1jh321kj3",  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |   Where fields are returned:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | clientTypeTxt | string | Type of association made | **Mobility Consumer**  **Mobility SMB**  **Mobility SMB Business Connect**  **Mobility Corporate**  **FFH Consumer**  **FFH SMB** | | identityTokenId | String | Key to determine identity of user |  | | |
| **SLA** | Services/APIs called:  **AccountMgmtDataService**  getAccount  getAccountByPhoneNumber  (FFH)  **ConsumerCustomerManagementService**  *getCustomer*  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## token-validation

|  |  |
| --- | --- |
| **OPERATION**  registrationtoken/{token}/token-validation | |
| **Method** | GET |
| **Description** | To validate requester has a valid validation token before proceeding to challenge the user on an account attachment. Returns whether or not token is valid and basic information about current attachments in order for front end to determine an appropriate challenge question for access. Internally performs these order of operations:   * Using token is input, retrieves a UUID from ClientIdentity Profile Service. * Looks up wireless permissions to see if mobility (via UUID) * Looks up enterprise permissions to see if FFH customer ID is associated (via UUID) |
| **Input** | Inputs:  **registrationtoken** = Client Identity user profile token as issued by client identity system.  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | brand | 1 = Telus  3 = Koodo | Yes | Indicates company brand |   Example:  profilemanagement/registrationtoken/2280bbbcx5b52b4429bb98bb09dcc64ed50/token-validation?brand=1 |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NP | No Permissions |  | User is has no associated service permissions | Call fails due to permitted business objects associated with UUID represented by token input | | 400 | NPR | No Profile |  | no profile exists |  | | 400 | MPR | More Than One Profile Found |  | More than one profile found for token: <registrationtoken> |  | | 400 | MLE | Multiple Linkage Exists |  | More than account association exists |  | | 400 | PNP | Profile not pending |  | profile status is not pending as expected. Status is: Active: <status> |  | | 400 | AA | profile status is already in active state |  | profile status is not pending as expected. Status is: Active |  | | 400 | EXP | profile status is expired |  | profile status is not pending as expected. Status is: Expired |  | | 500 | CIPS | Application level error (Client Identity Profile Service call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |
| **Output** | “tokenValidationServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {  "clientTypeTxt" : "Mobility Consumer",  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |   Where fields are returned:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | clientTypeTxt | string | Type of account to be matched for challenge question. | **Mobility Consumer**  **Mobility SMB**  **Mobility SMB Business Connect**  **Mobility Corporate**  **FFH Consumer**  **FFH SMB** | |
| **SLA** | Services/APIs called:  **ClientIdentityProfileSvc**  searchProfile  **SubscriberInformationService v3.2 (BVoip MDN cross reference)** getSubscriberByPhoneNumber  **WirelessPermissionSvc**  getRolePermissionList  ---- for wireline support ----  **PermissionService** (aka enterprise permission svc)  getPermissions  Expected average response time: 1500 ms |
| **Sample GUI** |  |

## account-validation

|  |  |
| --- | --- |
| **OPERATION**  registrationtoken/{token}/account-validation | |
| **Method** | GET |
| **Description** | To validate a subscriber belongs to any BAN associated with a user profile provided or directly to any user profile.   * Using token is input, retrieves a UUID from ClientIdentity Profile Service. * Looks up wireless permissions to see if mobility (via UUID)   + For mobility BAN (as a list) obtains client\_account\_id’s via calling getAccountList   + For client\_account\_id, calls getSubscriptionsByClientAccountId and checks for first matching subscriber   + If matching subscriber, returns 200 success. * Looks up enterprise permissions to see if FFH customer ID is associated (via UUID)   Upon successful subscriber validation a client identity role is issued by sending Client Identity one of several roles to associate to the user:  "teluswirelessconsumer"  "teluswirelessbusiness"  "teluswirelesscorporate"  "koodoWirelessConsumer"  “telusffh” |
| **Input** | Inputs:  **registrationtoken** = Client Identity user profile token as issued by client identity system.  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | brand | 1 = Telus  3 = Koodo | Yes | Indicates company brand | | sub | “4165551234” | Yes | Indicates the subscriber to match to any associated |   Example:  profilemanagement/registrationtoken/2280bbbcx5b52b4429bb98bb09dcc64ed502/account-validation?brand=1&sub=4165551234 |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NP | No Permissions |  | User is has no associated service permissions | Call fails due to permitted business objects associated with UUID represented by token input | | 400 | NPR | No Profile |  | no profile exists |  | | 400 | IC | Invalid Credentials |  | subscriber: <sub> is different from what was found on the account | if BVOIP and v1.1 up, systemErrorTxt could potentually be:   * brand: <brand> is different from what was found on the account * the account is not active. * the account is not linked. | | 400 | AA | profile status is already in active state |  | profile status is not pending as expected. Status is: Active |  | | 400 | PNP | Profile not pending |  | profile status is not pending as expected. Status is: Active: <status> |  | | 400 | EXP | profile status is expired |  | profile status is not pending as expected. Status is: Expired |  | | 500 | CIPS | Application level error (Client Identity Profile Service call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |
| **Output** | “accountValidationServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {  "clientTypeTxt" : "Mobility Consumer",  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |   Where fields are returned:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | clientTypeTxt | string | Type of account matched for challenge question. | **Mobility Consumer**  **Mobility SMB**  **Mobility SMB Business Connect**  **Mobility Corporate**  **FFH Consumer**  **FFH SMB** | |
| **SLA** | Services/APIs called:  **AccountMgmtDataService**  getAccountList  getSubscriptionsByClientAccountId  **SubscriberInformationService v3.2 (BVoip MDN cross reference)** getSubscriberByPhoneNumber  **WirelessPermissionSvc**  getRolePermissionList  **ClientIdentityProfileSvc**  searchProfile  createInitialProfile  --- To support FFH later ---  **PermissionService** (aka enterprise permission svc)  getPermissions  **ConsumerCustomerManagementService**  *searchCustomer*  *getCustomer*  *getCustomerWithContacts*  No method today to “activate” an existing link independently of the CII user profile..  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## service-disassociation

|  |  |
| --- | --- |
| **OPERATION**  profile/{uuid}/service-disassociation | |
| **Method** | POST |
| **Description** | TO disassociate (unlink) either a BAN (as owner), BAN (as manager) or Subscriber (as member) to a given portal profile (UUID) |
| **Input** | Inputs:  **uuid** = Indicates the user profile ID to associate service to  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | banList | list of bans | optional | Indicates the account to associate to user profile as either owner or manager | | subList | List of subscribers | optional | Indicates the subscriber to assign as member to user profile | | brand | 1 = Telus  3 = Koodo | Yes | Indicates company brand |   Example:  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/service-disassociation |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **Status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | IB | Invalid BAN | BAN [90653242] Not Found | messageId=xxxxx errorCode=xxxxx | Call fails due to invalid Ban | | 400 | IS | Invalid Sub | Subscriber [416-97623232] Not Found | messageId=xxxxxxxx, errorCode=xxxxxxxxxx | Call fails due to invalid Sub | | 500 | WPS | WirelessPermissionSvc Down | underlying error code | underlying error message | Any other Policy or Service Exception | | 500 | DAL | Application level error (DAL call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |
| **Output** | “disassociateServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } | |
| **SLA** | Services/APIs called:  **UserRoleManagementSvc**  deleteUserRoles  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## email

|  |  |
| --- | --- |
| **OPERATION**  profile/{uuid}/email | |
| **Method** | PUT |
| **Description** | This operation takes sets an email address as belonging to a user profile (uuid).  **Version 1.0, 1.1:**   * Call CustomerEmailSyncService **changeUserEmailAddress** operation sending in ServiceType=”consumer”, uuid and email, and self serve CMDB appid ‘11776’   **Reminder:** The URI parameter profile/{uuid} input will not be used as input, rather, everything is read from POST body. URI pattern is used as a security mechanism for SDF solution. The uuid parameter in both URI and in POST body will be populated by SDF based on access token issued, sent by calling application.  **Version 1.2:**  Returns a list describing which systems are updated by changeUserEmailAddress operation. |
| **Input** | Inputs:  **uuid** = Indicates the user profile ID to associate service to  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | uuid | ie “21EC2020-3AEA-1069-A2DD-08002B30309D” | Y | Client Identity unified user profile identifier | | email | ie “robisawesome@gmail.com” | Y | Email to apply to user profile |   Example:  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/email |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NPR | No Profile |  | no profile exists |  | | 400 | MPR | More Than One Profile Found |  | More than one profile found for token: <registrationtoken> |  | | 400 | CESS | Already Exists |  | Email already exists |  | | 400 | EESSERR | EnterpriseEmailSyncService returned error |  |  |  | | 400 | PNA | Profile status is not active |  | Profile status is not active |  | | 500 | CIPS | Application level error (Client Identity Profile Service call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |  |  |  |  |  |  | |  |  |  |  |  |  | |
| **Output** | “accountValidationServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {    "systemNameList" : [ // v1.2 and up  {  "KB",  "Ebill",  "WCPMS",  "LDAP",  "Primary email",  "FFH ebill email"  }  ],  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } | |  |   Where fields are returned:   |  |  |  |  | | --- | --- | --- | --- | | **Field** | **Datatype** | **Description** | **Possible/typical values** | | systemNameList | string | Downstream systems with emails updated by this call. | KB  Ebill  WCPMS (Contact sync DB for Keystone)  LDAP  Primary email  FFH ebill email | |
| **SLA** | Services/APIs called:  **ClientIdentityProfileSvc**  getProfiles  **ConsumerEmailSyncService (v1.0 – 1.1 only)** changeUserEmailAddress  **EnterpriseCustomerEmailSyncService\_v1\_0 (v1.2 onwards)** changeUserEmailAddress  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## role – WORK IN PROGRESS

|  |  |
| --- | --- |
| **OPERATION**  profile/{uuid}/account/{ban}/role | |
| **Method** | PUT |
| **Description** | This operation will update roles assigned to a user profile as it pertains to existing associated products/services. Uses of this operation include:   * Upgrading Manager to Owner role for given profile and mobility BAN * Upgrading Member to Manager role for given profile at mobility sub (to manager of BAN for the sub) * Downgrading Manager to Member role for given UUID, BAN and SUB. * Downgrading Owner to Manager role for given profile and BAN.   **Reminder:** The URI parameter profile/{uuid} input will not be used as input, rather, everything is read from POST body. URI pattern is used as a security mechanism for SDF solution. The uuid parameter in both URI and in POST body will be populated by SDF based on access token issued, sent by calling application. |
| **Input** | Inputs:  **uuid** = Indicates the user profile ID to associate service to  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | uuid | ie “21EC2020-3AEA-1069-A2DD-08002B30309D” | Y | Client Identity unified user profile identifier of user permitted to alter roles on BAN. | | profile\_id | “rick.foster@telus.com | N | Profile email address with role to be managed. | | user\_id | “fosterri” | N | Profile user id with role to be edited. | | ban | “987654321” | Y | BAN associated to uuid to have role updated | | currentsub | “4166843634” | N | If current role is member, the subscriber to update to manager or owner role | | newsub | “4166843634” | N | If current role is Manager or Owner and new role is Member, the subscriber to associate to the profile. The subscriber must belong to ban provided. | | currentrole | Manager, Owner, Member | Y | Role currently assigned to given product/service for given profile | | newrole | Manager, Owner, Member | Y | Role to assign to given product/service for given profile |   **Example – Upgrade Manager to Owner:**  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/account/987654321/role  Payload:  *{*  *“uuid”: ”21EC2020-3AEA-1069-A2DD-08002B30309D”,*  *“ban”: ”987654321”,*  *“currentrole”: “Manager”,*  *“newrole”: ”Owner”,*  *"currentsub": "",*  *"newsub": ""*  *}*  **Example – Upgrade Member to Manager: \*\***  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/account/987654321/role  Payload:  {  “uuid”: ”21EC2020-3AEA-1069-A2DD-08002B30309D”,  “ban”: ”987654321”,  “currentrole”: “Member”,  “newrole”: ”Manager”,  "currentsub": "4166843634",  "newsub": ""  }    **Example: Downgrade Owner to Manager:**  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/account/987654321/role  Payload:  *{*  *“uuid”: ”21EC2020-3AEA-1069-A2DD-08002B30309D”,*  *“ban”: ”987654321”,*  *“currentrole”: “Owner”,*  *“newrole”: ”Manager”,*  *"currentsub": "",*  *"newsub": ""*  *}*    **Example: Downgrade Manager to Member:**  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/account/987654321/role  Payload:  *{*  *“uuid”: ”21EC2020-3AEA-1069-A2DD-08002B30309D”,*  *“ban”: ”987654321”,*  *“currentrole”: “Manager”,*  *“newrole”: ”Member”,*  *"currentsub": "",*  *"newsub": "4166843634"*  *}*  **Example: Member swap** (maybe not used right away)  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/account/987654321/role  {  “uuid”: ”21EC2020-3AEA-1069-A2DD-08002B30309D”,  “ban”: ”987654321”,  “currentrole”: “Member”,  “newrole”: ”Member”,  "currentsub": "4166844321",  "newsub": "4166841234"  }  \*\* Note – Post ICM the “Upgrading Member to Owner or Member to Manager” should be accomplished using service-association to add Owner or Manager role, leaving Member role in place. So this action is a net-add. |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **Status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NPR | No Profile |  | no profile exists |  | | 400 | IB | Invalid Ban |  | BAN <ban> not found | BAN not found in CODS | | 400 | IS | Invalid Sub |  | SUB <sub> not found | Subscriber not found in CODS | | 400 | IR | Invalid Role |  | Role <role> is not a valid role | Invalid role name was used or was null. | | 400 | IU | Invalid User |  | UUID was not valid. | CIPS returns no UUID | | 400 | CRNE | Current Role Not Exist |  | Current role provided is not associated to user. | Linkage does not exist for user provided. | | 400 | IAM | Illegal Account Mixing |  | New role conflicts with existing role an account. | Triggered by URMS when ILLEGAL\_ACCOUNT\_MIXING error code is returned. | | 400 | IR | Illegal Role |  | New role is not allowed: <urn> | New role fails URMS error ILLEGAL\_ROLE | | 400 | OAE | Owner Already Exists |  | An owner already exists for BAN: <urn> | When trying to create an owner on a BAN that already has an owner. | | 400 | ONE | Owner Does Not Exist |  | An owner does not yet exist on this BAN <ban> | Returned when URMS returns OWNER\_DOES\_NOT\_EXIST | | 400 | MRNA | Member Role Not Allowed |  | New member role cannot be created because subscriber <newsub> does not belong to the BAN <ban> |  | | 400 | OE | Owner Exists |  | Owner role <urn> is already associated to this user. |  | | 400 | ME | Member Exists |  | Member role <urn> is already associated to this user. | Happens when currentsub = newsub and currentsub is a member already. | | 400 | AE | Already Exists |  | Linkage <urn> already exists | Catch all ‘already exists’ condition, so for manager case since ME and OE were not triggered. | | 500 | CIPS | Application level error (Client Identity Profile Service call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |  |  |  |  |  |  | |  |  |  |  |  |  | |
| **Output** | “accountValidationServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {    "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |   Where fields are returned: |
| **SLA** | Services/APIs called:  **ClientIdentityProfileSvc**  getProfiles  **ConsumerEmailSyncService** changeUserEmailAddress  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## account-association-list – WORK IN PROGRESS

|  |  |
| --- | --- |
| **OPERATION**  account/{ban}/account-profile-association-list | |
| **Method** | GET |
| **Description** | This operation returns all profiles associated to a given account (ban) and subscribers belonging to provided BAN. |
| **Input** | Inputs:  **ban** = wireless or wireline account  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | ban | “70144488” | N  (if customerid not given) | Billing Account Number | | language | "en", "fr", determines language of decoration | N | Default is “en” |   Example:  profilemanagement/account/70144488/account-association-list?language=en |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NPR | No Profile |  | no profile exists for account <ban> |  | | 500 | CIPS | Application level error (Client Identity Profile Service call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |  |  |  |  |  |  | |  |  |  |  |  |  | |
| **Output** | “accountAssociationServiceResponse” : { < accountAssociationList> }  “accountAssociationServiceResponse” :  {  “accountAssociationList” : [  {  "permissionId" : "11137",  "firstNameTxt" : "Richard",  "lastNameTxt" : "Dawkins",  "roleTxt" : "Account Owner",  “billingAccountNum” : “”,  “phoneNumberTxt” : “”,  “userNameTxt” : “”,  “emailAddressTxt” : “fbk@telusinternal.com”,  “emailValidationStatusTxt” : “Assumed Valid”,  “profileCreationDt” : “2013-06-24T13:45:03-4:00”,  “profileStatusTxt” : “Active”,  “lastLoggedInDt” : “2013-08-24T13:45:03-4:00”,  “emailValidationDt” : “2013-08-24T13:45:03-4:00”,  “tempPasswordInd” : “false”,  }    {  "permissionId" : " 7395 ",  "firstNameTxt" : "Richard",  "lastNameTxt" : "Hitchens",  "roleTxt" : "Member",  “phoneNumberTxt” : “4160608283”,  “userNameTxt” : “”,  “emailAddressTxt” : “fbk@telusinternal.com”,  “emailValidationStatusTxt” : “Assumed Valid”,  “profileCreationDt” : “2013-06-24T13:45:03-4:00”,  “profileStatusTxt” : “Active”,  “associationStatusTxt” : “PENDING”,  “lastLoggedInDt” : “2013-08-24T13:45:03-4:00”,  “emailValidationDt” : “2013-08-24T13:45:03-4:00”,  “tempPasswordInd” : “false”,  }  ]    "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  }  Where fields are returned: |
| **SLA** | Services/APIs called:  **AccountMgmtDataService**  getAccount  **ClientIdentityProfileSvc**  getProfiles  **WirelessPermissionSvc**  getRolePermissionList  **PermissionService** (aka enterprise permission svc)  getPermissionsList  **ConsumerCustomerManagementService**  *getCustomer*  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## profile-association-list – WORK IN PROGRESS

|  |  |
| --- | --- |
| **OPERATION**  profile/{uuid}/profile-association-list | |
| **Method** | GET |
| **Description** | This operation returns all accounts associated to a given user (uuid). |
| **Input** | Inputs:  **uuid** = Indicates the user profile ID to associate service to  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | uuid | ie “21EC2020-3AEA-1069-A2DD-08002B30309D” | Y | Client Identity unified user profile identifier |   Example:  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/profile-association-list |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NPR | No Profile |  | no profile exists for account <ban> |  | | 500 | CIPS | Application level error (Client Identity Profile Service call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |  |  |  |  |  |  | |  |  |  |  |  |  | |
| **Output** | “userAssociationServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {    "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } |   Where fields are returned: |
| **SLA** | Services/APIs called:  **AccountMgmtDataService**  getClientAccountBillingContactIndividual  (returns IndividualValueObject containing Address object containing a postal code)  **ClientIdentityProfileSvc**  getProfiles  **WirelessPermissionSvc**  getRolePermissionList  **PermissionService** (aka enterprise permission svc)  getPermissionsList  **ConsumerCustomerManagementService**  *getCustomer*  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## profile-detail – WORK IN PROGRESS

|  |  |
| --- | --- |
| **OPERATION**  profile/{uuid}/profile-detail | |
| **Method** | GET |
| **Description** | This operation returns detailed attributes about a given profile (uuid) |
| **Input** | Inputs:  **uuid** = unified user profile identifer  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | language | "en", "fr", determines language of decoration | N | Default is “en” |   Example:  profilemanagement/profile/21EC2020-3AEA-1069-A2DD-08002B30309D/profile-detail?language=en |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | NPR | No Profile |  | no profile exists for account <ban> |  | | 500 | CIPS | Application level error (Client Identity Profile Service call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |  |  |  |  |  |  | |  |  |  |  |  |  | |
| **Output** | “profileDetailServiceResponse” : { < profileDetail > }  Example:  “profileDetailServiceResponse” : {  “profileDetail” : {  "firstNameTxt" : "Richard",  "lastNameTxt" : "Dawkins",  “emailAddressTxt” : “fbk@telusinternal.com”,  “userNameTxt” : “”,  “emailValidationStatusTxt” : “Assumed Valid”,  “emailValidationSentDt” : “2013-09-24T13:45:03-4:00”,  “profileCreationDt” : “2013-06-24T13:45:03-4:00”,  “profileStatusTxt” : “Active”,  “profileLanguageTxt” : “English”,  “lastLoggedInDt” : “2013-08-24T13:45:03-4:00”,  “emailValidationDt” : “2013-08-24T13:45:03-4:00”,  “tempPasswordInd” : “false”,  “passwordStatusTxt” : “Unlocked”  “profileStatusExpiryDate ” : "2099-08-24T13:45:03-4:00”,  “identityRoleList” : [  {  “identityRoleTxt“ : “teluswirelessconsumer“  }  ],  “deltaEnrolInd”, “true”,  “profileLanguageTxt” : “English”  “lastProfileUpdateDt” : “2013-09-24T13:45:03-4:00”,  “registeredByCd” : “self”,  “convergenceInd” : “false”  },    "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  }  Where fields are returned: |
| **SLA** | Services/APIs called:  **ClientIdentityProfileSvc**  getProfiles  Expected average response time: 2500 ms |
| **Sample GUI** |  |

## profile-disassociation– WORK IN PROGRESS

|  |  |
| --- | --- |
| **OPERATION**  /profile-disassociation | |
| **Method** | POST |
| **Description** | To disassociate (unlink) a list of personas from a given billing account. |
| **Input** | Inputs:  **ban** = wireless billing account number  Parameters   |  |  |  |  | | --- | --- | --- | --- | | **Name** | **Values** | **Mandatory?** | **Comments** | | ban | “12345678” | Y | Wireless BAN | | tid | “t819202” | Y | Employee ID of agent making request | | permissionList | List of permissionIds | Y | Indicates the permissionIds (ie “persona’s”) to expire, all must be associated to the ban passed in. |   **Example:**  profilemanagement/profile-disassociation |
| **Status Codes** | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **status**  **Cd** | **Status**  **SubCd** | **statusTxt** | **systemErrorCd** | **systemErrorTxt** | ***Notes*** | | 200 |  | OK |  |  |  | | 400 | TIDR | TID is required |  |  |  | | 400 | IPID | Invalid Sub | Subscriber [416-97623232] Not Found | messageId=xxxxxxxx, errorCode=xxxxxxxxxx | Call fails due to invalid Sub | | 500 | WPS | WirelessPermissionSvc Down | underlying error code | underlying error message | Any other Policy or Service Exception | | 500 | DAL | Application level error (DAL call fails) |  | Pass through error code | Any other Service Exception | | 500 |  | general error |  |  | Any caught exception not handled elsewhere | |
| **Output** | “disassociateServiceResponse” : {  "status” : <status>  Example:   |  | | --- | | Undecorated | | {  "status": {  "statusCd": "200",  "statusSubCd": null,  "statusTxt": "OK",  "systemErrorTimeStamp": null,  "systemErrorCd": 0,  "systemErrorTxt": null  }  } | |
| **SLA** | Services/APIs called:  **UserRoleManagementSvc**  deleteUserRoles  Expected average response time: 2500 ms |
| **Sample GUI** |  |

# Appendix

## Error Codes

|  |  |
| --- | --- |
| IB | Invalid BAN |
| IS | Invalid Subscriber Number |
| DAL | Account Management DAL level application error |
| WPS | Wireless Permission Svc level application error |
| IC | Invalid Credentials |